SEETHROUGH: A REAL-TIME SURVEILLANCE DEHAZING SYSTEM USING DARK CHANNEL PRIOR ALGORITHM WITH GAUSSIAN FILTERING METHOD

An Undergraduate Thesis Project
Submitted to the Faculty of the
College of Engineering, Computer Studies, and Architecture Lyceum
of the Philippines University – Cavite

In Partial Fulfillment
of the Requirements of the Degree Bachelor of
Science in Computer Science

Andre M. Nazareth

Alexandre F. Basiño

Ferdinand Carlo M. Palo

May 2024

APPROVAL SHEET

This undergraduate thesis project entitled **SEETHROUGH: A REAL-TIME SURVEILLANCE DEHAZING SYSTEM USING DARK CHANNEL PRIOR ALGORITHM WITH GAUSSIAN FILTERING METHOD** prepared and submitted by **ANDRE M. NAZARETH, ALEXANDRE F. BASIÑO, AND FERDINAND CARLO M. PALO** has been reviewed and recommended as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

JERIAN R. PEREN, PhD, LPT
Research Adviser

Accepted and approved by the panel of examiners following a successful Final Defense on May 18, 2024, with a grade of _______.

GENSON P. MENDOZA, MIT, LPT
Panel Member

RAYMUND P. CONSTANTE, MBA, LPT
Panel Member

AMANDA JANE C. MENTA, PhD
Panel Chair

ARNEL M. AVELINO, PhD
Dean, College of Engineering, Computer Studies, and Architecture

Accepted and approved in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

MARK IRVIN C. CELIS, DBA
Vice President for Academic Affairs

Date: _________

Certificate of Originality

We hereby declare that the research paper entitled:

SEETHROUGH: A REAL-TIME SURVEILLANCE DEHAZING SYSTEM USING DARK CHANNEL PRIOR ALGORITHM WITH GAUSSIAN FILTERING METHOD

is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material to which to a substantial extent has been accepted for award of any other degree or diploma of a university or other institute of higher learning, except where due acknowledgement is made in the text.

We also declare that the intellectual content of this thesis is the product of our work, even though we may have received assistance from others on style, presentation and language expression.

Andre M. Nazareth

Alexandre F. Basiño

Ferdinand Carlo M. Palo
Proponent's Signature

Date of Completion: ___________
(mm/dd/yyyy)

Attested by:

JERIAN R. PEREN, Ph.D.
Research Adviser's Signature

Date of Completion: ___________
(mm/dd/yyyy)

ACKNOWLEDGEMENT

The researchers would like to thank their research adviser, Dr. Jerian Peren and our research professors, Dr. Elmer Matel and Mr. Joven Cajigas, for lending their valuable experience as an academic professional that greatly contributed to the improvement of this study.

Also, to their class adviser and professor, Mr. Sean Charlston Gono, for equipping the researchers with the necessary skills to commit themselves into the development of this thesis project.

To the statistician, Engr. Jelyn Rodriguez for guiding the researchers throughout the evaluation process and giving them advice on our statistical inquiries.

Lastly, their utmost appreciation to the Lord our God for giving them the capacity to learn and understand the ideas behind this research topic.

Andre M. Nazareth
Alexandre F. Basiño
Ferdinand Carlo M. Palo

ABSTRACT

The study focused on dehazing solutions for surveillance systems and mobile phones, specifically on Windows and Android platforms. To dehaze the input hazy image, the researchers utilized the Dark Channel Prior (DCP) algorithm approach in combination with a Gaussian Filtering method to reduce the halo residue of the DCP-based dehazing method. Forty-eight (48) respondents evaluated the desktop and mobile applications based on the ISO/IEC 25010:2023 standard and Android Core App Quality criteria, respectively. The evaluation results show that both the desktop and mobile applications garnered an “Highly Acceptable” rating, which means that the applications met the intended users’ standards. SeeThrough was validated against the Foggy Road Image Dataset (FRIDA) with positive results on dehazing homogenous and heterogeneous fogs, showing that the dehazing system improved the image quality (visibility of objects) in terms of Peak-Signal-To-Noise Ratio evaluation. However, the system fell short on SSIM (Structural Similarity Index Measure) evaluation, which means that the dehazed output is not structurally similar to the ground truth/haze-free image. The dehazing method’s real-time performance was also evaluated on desktop and mobile versions. At 480p video resolution, the desktop application could dehaze 30 frames per second while the mobile version at 12 frames per second. The researchers recommend that future developers should explore adaptive brightness correction to reduce the darkening of images after using DCP, improve the performance of mobile and desktop applications by exploring GPU (Graphical Processing Unit) utilization, and explore the application of dehazing on vehicle dashboard cameras.

Keywords—dehazing, video dehazing, surveillance footage dehazing, mobile dehazing

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF APPENDICES

LIST OF ABBREVIATIONS

Abbreviation/Symbol	Definition
CPU	Computer Processing Unit
DCP	Dark Channel Prior
FRIDA	Foggy Road Image Database
GPU	Graphical Processing Unit
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
PSNR	Peak Signal-to-Noise Ratio
RTSP	Real Time Streaming Protocol
SSIM	Structural Similarity Index Measure
URL	Uniform Resource Locator

INTRODUCTION

Background of the Study

Fog or haze is a phenomenon caused by the particles stranded in the air such as water-droplets, dust, etc. wherein some light is absorbed by the particles while the others are scattered away before reaching the observer that results in a reduction on visibility of objects. To produce a quality image, digital cameras, much like the human eye, depend on the amount of light its image sensors collect through the lens to gather information about the subject (Malav et al., 2019).

More particles mean more absorption and scattering of light, hence the low visibility of objects ("Basic Information about Visibility," 2023). According to Rahman (2022) the haze phenomenon often occurs due to natural or man-made factors resulting in high concentration of the particles in the air. One of the biggest causes of road accident-related fatal injuries across the world is visibility degradation (Babu et al., 2021) caused by different weather conditions including haze, increasing accidents at night and doubling the fatality per 100 accidents (Duthon et al., 2020). The solution to the problem of haze phenomenon is the term "dehazing" where it is a process to improve the degraded visibility caused by the atmospheric conditions (Zahra et al., 2021). According to Zahra et al. (2021), the importance of removing haze can be used in many purposes as in surveillance and object identification among other possible fields. For Yang et al. (2023), improving the visibility of images affected by haze and fog is crucial for enhancing the performance of real-time outdoor computer vision applications.

Locally, Baguio City Senior Inspector Cabintoy (2023), stated in an interview that the traffic enforcement unit or the public safety division of Baguio does not have any means of dehazing their surveillance footage in real-time. If any crime or incident occurs during foggy conditions, an IT or multimedia expert is called to enhance a frame to progress a case. This means that a dehazing system would prove to be helpful during these conditions to allow the public safety division to be proactive while monitoring activities through their surveillance systems, despite the limitations posed by the hazy condition.

One of the current solutions to achieve removal of haze on an image is Dark Channel Prior Algorithm (DCP), which was proposed by He et al., based on a key observation that certain pixels in any color channel exhibit exceptionally low intensity. The DCP-based dehazing algorithm is composed of four steps; these are: atmospheric light estimation, transmission map estimation, transmission map refinement, and image reconstruction. For the transmission map refinement stage, the original implementation of the DCP uses a soft matting method. However, the research made by Lee at al. (2016) shows that although soft matting performs the best in terms of transmission map accuracy, it is the worst in terms of computational complexity. In terms of computational complexity, however, guided filter and gaussian filter performed the best. The study of Liu et al. (2021) on the comparison of different dehazing methods concludes that the current dehazing algorithms can effectively remove haze but a common problem is the lack of real-time performance. The existing studies focused on reducing the computational complexity of the algorithm itself through optimization, where in fact, it may be more effective to use hardware acceleration instead (Liu et al., 2021).

Statement of the Problem

Weather phenomena like haze, fog, and volcanic smog can significantly impair image quality, resulting in a loss of vital information in surveillance recordings. This issue is especially problematic in areas prone to fog, where current surveillance systems do not feature real-time dehazing capabilities. An interview with a senior police inspector indicated that the public safety division is unable to dehaze footage, which hampers their ability to effectively monitor during foggy conditions. Furthermore, research conducted by Liu et al. (2021) points out a deficiency in utilizing hardware acceleration techniques for dehazing, which could potentially yield better results than relying solely on algorithmic improvements. Commercial dehazing camera modules, such as Panasonic's GP-MH330 and Hitachi's KP-D5001, range in cost from $500 (around ₱27,776) to $5,000 (roughly ₱277,760), creating a notable obstacle for small enterprises and individuals. This study seeks to tackle these issues by creating a budget-friendly dehazing solution that employs the Dark Channel Prior with Gaussian Filtering, designed for integration with current surveillance systems and implementation on Android mobile platforms, thereby offering a practical and economical option for real-time dehazing. Generally, this study aims to address the limitations of surveillance systems in performing under different intensities of fog which limits its vision and to provide the general public a means to dehaze the area with an android device.

Specifically, this study sought to answer the following questions;

1. How was the dehazing algorithm be optimized to process footage in real-time while maintaining high image quality?
2. What specific development steps were followed to create the desktop application using Microsoft Visual Studio Code with Python and the mobile application using Android Studio with Java?
3. How was the system's dehazing performance be measured in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) using the Foggy Road Image Dataset (FRIDA)?
4. What functional and compatibility tests were conducted to ensure the reliability and usability of both the desktop and mobile applications?
5. How was the developed system be evaluated for acceptability in terms of Functional Suitability, Performance Efficiency, Compatibility, Interaction Capability, Reliability, Maintainability, and Flexibility?
6. What criteria were used to assess the mobile application's visual experience, performance and stability, and privacy and security to ensure it meets Android core app quality standards?

Objectives of the Study

This study aimed to design a "Real-Time Surveillance Dehazing System using Dark Channel Prior with Gaussian Filtering Method".

Specifically, the study aimed to:

1. Design a specific dehazing method that can be used by researchers to dehaze footage in real-time and allow users to dehaze an image or a stored video.

2. Create a desktop and mobile application using Microsoft Visual Studio Code with Python programming language for the desktop application and Android Studio with Java programming language for the mobile application. The OpenCV library will be used for developing the backend of the system. The desktop application will run on Windows 10 and above while the mobile application will run on Android 10 and above.

3. Evaluate the performance of the dehazing implementation in terms of the following:
    a) Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) results on Foggy Road Image Dataset (FRIDA);
    b) Frames processed per second on actual captured video.

4. Evaluate the developed desktop and android applications through Functional

Testing and Compatibility Testing.

5. Evaluate the acceptability of the developed system based on ISO/IEC 25010:2023 software quality standard with the following criteria:
    a) Functional Suitability
    b) Performance Efficiency
    c) Compatibility
    d) Interaction Capability
    e) Reliability
    f) Maintainability
    g) Flexibility
6. Evaluate the mobile application acceptability based on android core app quality with the following criteria on:
    a) Visual Experience
        - Navigation
        - UI and Graphics
        - Visual Quality
        - Accessibility
    b) Performance and Stability
        - Stability
        - Performance
    c) Privacy and Security

- Permissions
- Data & Files

Scope and delimitations of the Study

The study used Dark Channel Prior with Gaussian Filtering method to dehaze images or videos in real-time. To ensure that the testing phase was not dependent on real-world weather conditions, the testing of the algorithm was first done on still images from readily available datasets online such as FRIDA (Foggy Road Image DAtaset). After several optimizations, the final testing was conducted in a controlled environment with artificially produced haze/fog produced by a fog machine. The surveillance dehazing system acted as a standalone system and does not include any integration features to any existing surveillance applications. The respondents of this study were individuals who has been to, is a frequent visitor, or is a resident of foggy areas in the Philippines (i.e., Tagaytay, Baguio, etc.), who are the determinants of whether our algorithm performs at an acceptable level. Ideally, there were at least 30 respondents to increase variance in responses. The duration of the study was until the end of the second semester of the academic year 2023-2024.

Due to hardware limitations, the desktop system can only dehaze one footage up to 720p resolution in real-time and the mobile application does not include a real-time recording feature. Due to financial limitations, the system was tested with USB web cameras and IP surveillance cameras.

Significance of the Study

This study is beneficial in many ways, due to the possible applications of a real-time dehazing system. Although the literature on image and video dehazing is very rich, real-time dehazing is very rare on mobile platforms (Çimtay, 2020). According to Chawla et al. (2020), *"Dehazing is vital for many computer-vision algorithms used in remote sensing, agricultural monitoring, intelligent vehicles, object recognition, and surveillance that assume the hazy image as original scene radiance and hence suffer from bias."*

These were the users who could use the application. In this way, the accompanying would profit by the undertaking:

Public Safety Divisions

The expected output of this study is surveillance operators designated in public safety divisions in their ability to observe and detect objects, people, and events within the hazy video footage.

The Community

The general public will benefit from the improved capabilities of their local public safety division in ensuring their safety. The general public may also use the results of this study for themselves as an aid during foggy weather conditions.

The Researchers

The researchers will enrich themselves from conducting this study by analyzing and applying the Dark Channel Prior and Gaussian Filtering algorithms in a real-time system.

Future Researchers

The research community in the field of computer vision, image processing, and transportation safety can benefit from the findings of this study. The findings of this study will contribute to the existing body of knowledge by exploring the effectiveness of the Dark Channel Prior with Gaussian Filtering Method in a real-time dehazing system. The study also contributes to the study of applications of real-time dehazing on Android platform, providing insights for future research designs in similar contexts.

REVIEW OF RELATED LITERATURE AND STUDIES

In this chapter, the researchers gathered literature and studies related to the study. These will help the readers to fully understand the content of the research in the later part of the study. In the review of literature, the terms that are strongly connected to the study are explained while the review of related studies discusses the researchers' findings and recommendations that are highly relevant to this study. A synthesis is also presented to show how this literature and studies correlate to the researchers' research and how the algorithm was implemented in a real-time system.

Related Literatures

The researchers have included in this section a review of literature that is relevant to the manuscript.

Foggy and Hazy Conditions

According to Zahra et al. (2021), images that are taken during hazy and foggy conditions for video surveillance applications reduce the reliability of the footage because it reduces visibility. The image under foggy or hazy conditions decreases the color contrast and surface color of the object with deference to the distance from the object. Due to this poor contrast and low visibility, degraded images create difficulty in various real time applications. Therefore, to remove fog, haze, and enhance the visibility of an image is very important because an image can be used for many purposes such as surveillance.

Application of the dehazing in relation to the study

Computer vision is a simulation of human vision that is meant to train the computer to understand and interpret vision of the world using several algorithms and applications. It is used in object tracking, face recognition, self-driving, and such (Hammoudeh et al., 2022). In the context of this study, the algorithm will fall under computer vision, specifically on image processing to remove or mitigate the effect of haze in the input video or image.

The application of dehazing is vital for many scenarios including surveillance (Chawla et al., 2020). According to Senstar (2023), a video surveillance system consists of interconnected cameras, display units or monitors and recorders. Its purpose is to monitor or record activities in a specific area for security purposes and commonly used in public spaces and in private buildings (Isarsoft, 2023). The researchers utilized the surveillance system's devices to test the capability of real-time dehazing.

The researchers utilized the use of RTSP (Real-Time Streaming Protocol) along with a surveillance camera to apply dehazing in the context of the study. According to Lenihan (2022), RTSP has the ability to transmit audio/video from one source to another. RTSP is mostly used in IP Cameras where it can help transport low-latency content across the internet so it can be played back on a device.

Surveillance Image Enhancement Techniques

Image dehazing is a technique used in image enhancement to improve image quality and visibility by removing haze, thereby increasing the color and contrast of scenes (Baig, 2021). Fan et al. (2019) highlight denoising as a method to remove noise that distorts image information during acquisition, compression, and transmission. Kaur et al. (2021) discusses image fusion, which combines multi-temporal, multi-view, and multi-sensor information into a single image, enhancing quality while retaining crucial features. Ai and Kwon (2020) address the challenges of low-light image enhancement, which makes buried details in dark and noisy images more visible. Mane et al. (2022) emphasizes the importance of color image enhancement, involving HDR compression and contrast enhancement to improve image visibility and quality. The researchers utilized image dehazing as the image enhancement technique for the study as the study deals with removal of the haze of the scenes of the surveillance camera.

Real-Time Systems

According to Dahiya (2023), systems subjected to real-time means that the response or the output of the system should be guaranteed within a specified timing constraint. There are different types of real-time systems that researchers can use to implement an algorithm on, depending on their project task’s deadline restrictions. A real-time system falls under two types, which is hard real-time system or soft real-time system. Missing a task deadline on a hard real-time system is catastrophic, while it can merely lead to heavy losses in a soft real-time system.

In a soft real time system, deadlines are not compulsory meaning missed deadlines may have a small impact on the quality of service provided but not catastrophic (Koren & Krishna, 2021). As the project’s purpose is only to dehaze surveillance footage in real-time and does not require any immediate notifications on certain scenarios, the project falls under a soft real-time system.

Dark Channel Prior as the dehazing method of the study

Table 1

Overview of Dehazing Methods: Advantages, Disadvantages, and Practical Implementations

Method	Advantages	Disadvantages	Implementation
Contrast Maximizing Method	Enhances contrast	Color Distortion and brightness saturation, halo effect, blocking artifact	Image Dehazing using Laplacian Approach (Kumar et al., 2021)
Generative Adversarial Network Method	Yields specific training algorithm	Requires supervision	Remote sensing image dehazing using generative adversarial network (Shen et al., 2024)
Deep Learning-based Method	Outperforms many traditional methods	Tends to be larger in size, have more computational parameters, often slower than traditional methods, and requires supervision May not be suitable for real-world conditions	Cycle-Dehaze: Enhanced CycleGAN for Single Image Dehazing (Engin et al., 2018)
Dark Channel Prior-based	Simple theory and clear	Computationally complex, halo effect	Single Image Haze Removal Using Dark

Method	restoration result		Channel Prior (He et al., 2009)
Polarization Method	Estimated Accurate Transmission Light	Noise may occur during dehazing	Polarization-based image dehazing (Sun et al., 2022)
Color Attenuation Prior-based Method	Reduces amplified noise	Cannot be used for dense fog, requires complex training	Improved Color Attenuation Prior for Single-Image Haze Removal (Ngo et al., 2019)

The table provides a side-by-side comparison of different dehazing methods. It points out their pros, cons, and specific uses. For example, the Contrast Maximizing Method used in the Image Dehazing with Laplacian Approach (Kumar et al., 2021) is recognized for improving contrast quite well. Still, it has some issues color distortion, brightness saturation, halo effects, & blocking artifacts, which can hurt overall image quality.

On the other hand, Generative Adversarial Network (GAN) techniques are highlighted by the Remote Sensing Image Dehazing using GAN (Shen et al., 2024). These methods have the advantage of specific training algorithms designed for their tasks. However, they do need a lot of supervision. This demand can be resource-intensive and limits their practical uses in situations where labeled training data is hard to find.

Deep Learning approaches like Cycle-Dehaze (Engin et al., 2018) show better performance than many older methods because they use advanced neural network structures. Yet, these techniques usually have larger sizes with more computing parameters. This often leads to slower processing speeds when compared to traditional methods. Additionally, they require high-performance devices & substantial supervision, which may restrict their use in real-world settings where computing resources are limited.

The Dark Channel Prior (DCP) method explained by He et al. (2009) has a simple theory & gives clear restoration results. But it is also computationally complex and can create halo effects that reduce the visual quality of the dehazed images.

Meanwhile, the Polarization Method discussed by Sun et al. (2022) gets precise transmission light values that help improve dehazing. Still, it might add noise during the process, which could lower image quality.

Lastly, the Color Attenuation Prior-based method enhanced by Ngo et al. (2019) successfully reduces amplified noise. But it isn't great for heavy fog conditions & requires complicated training, making it less adaptable than some other methods.

In conclusion, every dehazing method has its own set of strengths & weaknesses. Choosing the right dehazing technique depends on different factors like the need for real-time processing, how much computing power is available, and the type of fog or haze conditions that need addressing.

.

Gaussian Filter as the transmission map refinement of the dehazing method

According to Aksoy and Salman (2020), Gaussian filter is a low-pass filter to remove detail and noise in the image. It is commonly used in image processing applications to smoothen and blur the noises especially the edges to improve the image quality. In the context of the main algorithm, the Dark Channel Prior is prone to halos and artifacts. The main reason why the researchers will utilize the use of Gaussian Filter is because it is useful in removing color textures in the transmission map (Koranga & Kumawat, 2020) and to smoothen the edges of the halo artifacts brought by the Dark Channel Prior to better blend it with the surrounding scene in the image.

Mobile Dehazing

According to Cimtay (2021), real-time haze removal on mobile devices like in Android and iOS is uncommon. While there has been a successful attempt to implement haze removal on Windows Phone using GPU rendering, the efficiency of the algorithms remains low. However, despite this, mobile haze removal has met its intended goals in terms of effectiveness. The author proposes a new real-time haze removal method that utilizes the orientation sensor of a mobile device, comparing rotation levels to a set threshold. This approach achieves a low processing time for 360p imagery, demonstrating the feasibility of real-time implementation on mobile platforms.

Programming Languages

Python is a high-level interpreted programming language created by Guido van Rossum. Python is widely adopted due to its clear and readable syntax, and is often used in data analytics, machine learning, image processing, and scientific applications (Zola, 2021). Python is used as the main programming language that the researchers will use upon developing the desktop application.

Android applications can be developed using programming languages such as C, C++, Scala, and Java. However, Java is the most preferred programming language in android application development and can run on over 6 billion devices (Admin, 2022). In this study, the mobile dehazing application is developed using Java.

In programming, libraries are a collection of reusable prewritten codes that developers can use to quickly develop applications and optimize tasks, designed to provide a specific solution for a project (Meltzer, 2023). Open-source Computer Vision or OpenCV was initially developed by Intel as a free cross-platform computer vision library for real-time image processing (Boesch, 2023). This library can be used in Python, Java, and C/C++ programming languages. In this study, both the desktop and mobile implementations of the dehazing system utilized the OpenCV library to perform video and image processing. Numerical Python or NumPy is a Python library for performing numerical calculations fast and efficiently. NumPy takes advantage of the computational efficiency of the C programming language by being written in it. This library is an important library for data science, machine learning, image processing, scientific applications, and more (Python Tutorial, 2023). In the context of the project, NumPy is used as the library for the most calculations of the algorithms implemented.

Evaluation Metrics for Dehazing Algorithms

Several metrics to assess image quality are available, falling under two categories, namely a full-reference approach and a no-reference approach (Sara et al., 2019). A full reference approach assesses the quality of a test image with a reference image. The reference image, also called as the ground truth, is considered as a pristine quality image with no distortion or degradation. Examples of a full-reference quality measurement technique are Mean Square Error, Root Mean Square Error, Peak Signal to Noise Ratio, and Structural Similarity Index Measure. The no reference approach assesses the quality of a test image using only the test image itself, without any reference to a pristine image.

Peak Signal to Noise Ratio or PSNR is the ratio of the highest value of a pixel to the noise, which has an impact on the pixel quality (Paul and Anitha, 2019). The error is expressed in terms of a logarithmic decibel scale, and the higher the PSNR means smaller the error meaning it has a better denoised result in an image (Shahin et al., 2019).

Structural Similarity Index Measure (SSIM) is a FR-IQA (Full-Reference Image Quality Assessment) measure that is adapted for extracting the structural information from the scenes. An SSIM value close to 1 indicates that the images are similar to one another (Bakurov, 2022).

Ethical Considerations in Surveillance Image Processing

In the case of computer vision algorithms that require large sets of data, researchers scrape the web to gather large data sets of facial images to enhance commercial and military surveillance algorithms. Users are unaware that the images they post online are later used by researchers as training data (Innodata, 2022).

According to Saini (2023), it is necessary to maintain ethical standards by being transparent with the purpose, scope, and possible risks of data collection and utilization. Human oversight and decision-making was also discussed, stating that the algorithm's output should not be the only consideration when making critical decisions, that human intervention and judgment is necessary.

In the context of this study, the sample datasets mainly came from synthetic images provided by RESIDE and FRIDA. Real-world testing was done with the respondents' and local authorities' consent.

Related Studies

Discussed in this section are the studies that play a significant role in planning the manuscript project.

Dark Channel Prior as a Dehazing Algorithm

The research made by Liu et al. (2023) shows that an improved dark channel prior has certain application value. Their experiment shows that their implementation of dark channel prior can effectively remove haze in a road scene image with a result that is close to the ground truth or the non-hazy image.

The study of Liu et al. (2021), compared different types of prior-based and deep learning-based dehazing methods side-by-side with their distinct categories (Image Enhancement, Image Fusion & Image Restoration). Their study aims to evaluate the HE, Retinex, DCP, Non-Local, DehazeNet, All-in-One Dehazing Network (AOD-Net) and Gated Context Aggregation Network (GCANet) quantitative performance in remote sensing hazy images (See Appendix 13).

Liu et al. (2021) concludes that a prior-based dehazing method extracts the parameters by imposing latent prior knowledge on atmospheric scattering model, thereby it can restore a haze-free scene from the hazy image physically. Data driven dehazing methods can produce a high-quality haze-free scene, but they may fail in cases with heavy haze. Based on their study, they tested other different hazed images and the results confirmed that prior-based and data-driven dehazing methods could remove haze and produce a good number of outputs however, they do not work well on the images with various scenes.

Prior-based dehazing methods

The dark channel prior algorithm is prone to halos and artifacts over the sky region. In the non-sky regions, the dark channel prior performs very well and so the author proposed a prior-based algorithm approach to reduce the halos over the sky region that has an original implementation of dark channel prior algorithm called “Bright Channel Prior”. The results demonstrate that the proposed method can well preserve sky regions, reduce color distortion and oversaturation, and provide higher PSNR and SSIM scores however, the proposed method cannot fully achieve the effect of dehazing non-uniform thick hazy images, which may be since the physical model and prior knowledge are not fully practical in non-uniform haze scenes (Li et al., 2023).

Hybrid dark channel prior (HDCP) method addresses issues with dark images and incomplete defogging caused by inaccurate atmospheric transmittance estimation. Initially, the Retinex approach is used to eliminate illumination interference and enhance image brightness. Atmospheric light intensity is then iteratively refined, followed by the application of a tolerance-improved dark channel prior (DCP) to produce the dehazed image. A variant genetic algorithm (VGA) enhances the grayscale of the original image, which is used as a guided filtering image to optimize transmittance. Experimental results using O-HAZE and NYU2 datasets show that the proposed method reduces the mean square error (MSE) by 26.98%, increasing the structural similarity index (SSIM) by 10.298. Compared to conventional DCP, the HDCP method results in higher brightness and more thorough fog removal (Wu et al., 2023).

Colour Attenuation Prior (CAP) is based on the observation that hazy regions of an image exhibit more significant variation in brightness intensity compared to haze-free areas. The study highlights that CAP outperforms DCP by integrating depth information for estimating transmission. However, it sometimes provides limited enhancement. Since, it uses hard thresholding to limit the scene radiance. The result does not suffer from excess saturation and brings out the real colors Parihar et al. (2020).

In conclusion, according to Li et al. (2023), Prior-based image dehazing methods estimate the parameters of the atmospheric scattering model and recover sharp images. Although it relies on the estimation of parameters, it is prone to problems such as distortion. The accuracy and rationality for prior selection are crucial for image dehazing. Some methods utilize certain features of local pixels in hazy images as priors for transmission estimation, while others adopt the geometry of pixel cluster distributions for transmission estimation. However, these priors are significantly influenced by outliers or noises in hazy images, resulting in unsatisfactory dehazed results in some cases (Yuan et al., 2021).

Deep Learning-based dehazing methods

The research of Zhang and Tao (2020) introduced FAMED-Net or Fast and Accurate Multi-Scale Dehazing Network, which improves the efficiency and accuracy of dehazing in single images. The traditional dehazing methods often suffer from high complexity and limited capability in handling various hazy conditions. FAMED-Net is a multi-scale architecture that integrates three encoders and a fusion module, and this allows the network to capture haze at different scales effectively. FAMED-Net use of point-wise convolutional layers and pooling layers within each encoder, foregoing larger convolutional kernels that typically increase computational load. By focusing on lightweight and efficient layers, FAMED-Net manages to achieve faster processing times while maintaining high accuracy. The network is trained to directly learn the mapping from hazy to haze-free images, thereby streamlining the dehazing process and reducing the need for extensive pre-processing. Empirical evaluations on both synthetic datasets and real-world hazy images showed that FAMED-Net manages to outperform the state-of-the-art dehazing methods. It offers a balanced trade-off between computational efficiency and restoration accuracy, making it particularly suitable for applications requiring real-time dehazing. The study shows that FAMED-Net achieves great performance in terms of model complexity, processing speed, and generalization across different datasets, making it a significant contribution to the field of image dehazing.

Zhao et al. (2023) proposes an unsupervised generative adversarial network (GAN) specifically designed for dehazing remote sensing images. This network includes two identical generators and two identical discriminators. One generator focuses on image dehazing, while the other adds haze to images. The discriminators distinguish between real and generated images. The generator utilizes an encoder-decoder architecture with multi-scale feature-extraction and attention modules. The multi-scale feature-extraction module has three branches with dilated convolutions and attention modules to extract features at different receptive fields. The proposed method outperforms state-of-the-art methods, achieving the highest peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) metrics, demonstrating its superior performance in dehazing remote sensing images.

Park et al. (2020) proposes a novel image dehazing method using heterogeneous Generative Adversarial Networks (GANs), combining CycleGAN for producing haze-clear images and conditional GAN (cGAN) for preserving textural details. A novel loss function is introduced to minimize GAN-generated artifacts, recover fine details, and preserve color components. These networks are fused via a convolutional neural network (CNN) to generate the final dehazed image. Extensive experiments demonstrate that the proposed method significantly outperforms state-of-the-art techniques on both synthetic and real-world hazy images.

Mehta et al. (2020) addresses the challenging problem of haze removal in images from various scenarios. Mehta et al. stated that traditional dehazing methods use old methods such as reconstructing the transmission map and estimating values through the RGB channels. Therefore, Mehta et al. proposes a deep learning-based dehazing method called HIDeGAN or Hyperspectral-guided Image Dehazing Generative Adversarial Network. The proposed method utilizes hyperspectral images (HSI) in combination with cycle-consistency and skeleton losses to improve the quality of information recovery by analyzing the entire spectrum for dehazing. The method generates the final dehazed output based on the RGB-HSI mapping learned from CycleGAN.

In conclusion, deep learning-based defogging methods are effective but have several drawbacks. These models can struggle under challenging lighting conditions, such as strong sunlight and shadows. They need a lot of training data, and a smaller dataset can lead to overfitting and poor generalization. Additionally, these methods require significant computational resources and data, making them expensive to deploy. The most significant issue is the incomplete fog removal, especially in situations with varying or high haze densities (Wu et al., 2023).

Time Complexity

Wang et al. (2021) discussed that the traditional method of Dark Channel Prior of Kaiming He has a high time complexity due to the soft matting technology. Their study shows that their method on changing Soft Matting technology to Guided Filtering with a four-point algorithm effectively improved the time complexity and edge effect.

Abed et al. (2020) performed a study on edge detection using a parallel implementation in the GPU and a sequential execution in the CPU. The results of Abed et al's study shows that their image processing algorithm performed up to 37 times faster on the GPU and 25 times faster on the CPU with a shared memory approach.

Lee et al. (2016) discussed different methods of transmission map refinement by comparing their computational complexity and dehazing quality against each other. Their research shows that in terms of computational complexity, gaussian and guided filters perform the best. In terms of transmission map estimation accuracy, soft-matting performs best, followed by a cross-bilateral filter.

Synthesis

The combination of the existing literature and relevant studies provides a thorough basis for creating an effective dehazing system designed for surveillance purposes. By examining both prior-based methods and deep learning techniques, valuable insights into their advantages and limitations are gained, which aid in shaping the development of the SeeThrough system. Research conducted by Liu et al. (2023) and Liu et al. (2021) highlights the effectiveness of the Dark Channel Prior (DCP) algorithm for dehazing, specifically in cases involving road scene images and hazy remote sensing images. The DCP approach utilizes inherent prior knowledge to restore scenes devoid of haze, offering a physically grounded method. However, it faces challenges when dealing with heavy haze and complex scenes. Efforts to tackle prevalent issues such as halos, artifacts, and incomplete defogging are illustrated in the studies on the "Bright Channel Prior" by Li et al. (2023) and the Hybrid Dark Channel Prior (HDCP) by Wu et al. (2023). These enhancements improve the visual clarity and brightness of the dehazed images but struggle with the unevenness of thick haze. The time complexity concerns associated with traditional DCP algorithms, as noted by Wang et al. (2021) and Lee et al. (2016), emphasize the necessity for efficient computational methods. Transitioning from Soft Matting to Guided Filtering, along with the adoption of hardware acceleration techniques as proposed by Abed et al. (2020), notably enhances the real-time performance of DCP-based dehazing systems.

In terms of deep learning approaches to dehazing, research by Zhang & Tao (2020), Zhao et al. (2023), and Park et al. (2020) illustrates the sophisticated capabilities of such methods, including FAMED-Net and GANs, in addressing various hazy scenarios. Though these techniques achieve high-quality dehazing, they demand significant computational resources and extensive datasets, which can be impractical for real-time surveillance. The use of hyperspectral images in HIDeGAN by Mehta et al. (2020) and the combination of CycleGAN with cGAN by Park et al. (2020) present novel strategies for improving dehazing outcomes. Nonetheless, these approaches still encounter obstacles in completely removing fog under differing haze thicknesses and lighting situations, as indicated by Wu et al. (2023).

Focusing on real-time surveillance, the system utilizes the straightforward and effective DCP algorithm while tackling its computational limitations through hardware acceleration and efficient refinement of transmission maps. Implementing Gaussian filtering, as suggested by Lee et al. (2016), enhances image quality by minimizing halos and artifacts, making it an appropriate option for real-time dehazing. Choosing to develop the system using Python, OpenCV, NumPy, and SciPy corresponds with the demand for reliable and scalable image processing tools. Prioritizing real-time performance is crucial, as highlighted by Liu et al. (2021), who point out the prevalent issue regarding the time efficiency of existing dehazing methods.

The dehazing system, grounded in the Dark Channel Prior algorithm and enhanced through Gaussian filtering and hardware acceleration, aims to provide an efficient and effective real-time dehazing solution tailored for surveillance. The synthesis of prior-based and deep learning techniques informs the design of the system, balancing the need for computational efficiency and high-quality outputs in dehazing. The literature reinforces the conceptual foundation of the system, underscoring the significance of addressing real-time performance challenges to fulfill the specific demands of surveillance systems.

Theoretical Background

This study is anchored on the theory of the atmospheric optic model. Salazar-Colores et al. (2019) discussed the atmospheric optic model or the formation of haze in the observed image. In this model, recovering the non-hazy image is difficult when the transmission map and atmospheric light parameters are unknown. Adding onto this difficulty, Lee et al. (2016) mentioned that the number of unknowns is equivalent to the number of image pixels.

Two dehazing methods were identified and discussed in this study, namely deep learning-based and prior-based methods. Wu et al. (2023) stated that although deep learning-based dehazing methods are effective at removing haze, they require significant training data and computational resources, making these methods expensive to deploy in a real-time system. On the other hand, prior-based dehazing methods are prone to distortion, influenced by the noise that is present in the image. Despite this demerit, however, Yang et al. (2021) stated that the Dark Channel Prior (DCP) is a simple and effective solution that results in a good dehazing output. Lee et al. (2016) also studied different transmission refinement methods commonly used with a DCP-based dehazing method, showing that guided and Gaussian filters perform the best in terms of computational complexity. The guided filter performs slightly faster than the Gaussian filter but requires a significant amount of memory.

The study of He et al. (2009) on Single Image Dehazing introduced the Dark Channel Prior algorithm as a method of determining the transmission map from the observed hazy image with the dark channel as the “prior” knowledge which directly answers the unknowns of the atmospheric optic model to recover the non-hazy image. However, using Dark Channel Prior has difficulty on preserving the brightness of the sky and often introduces halos and artifacts in the dehazed image, and therefore the study of Lee et al. (2016) on evaluating the performance and integrations of other algorithms with the Dark Channel Prior algorithm introduced the researchers to the Gaussian Filtering method which is a technique to refine transmission maps to reduce the artifacts and halos improving the image quality.

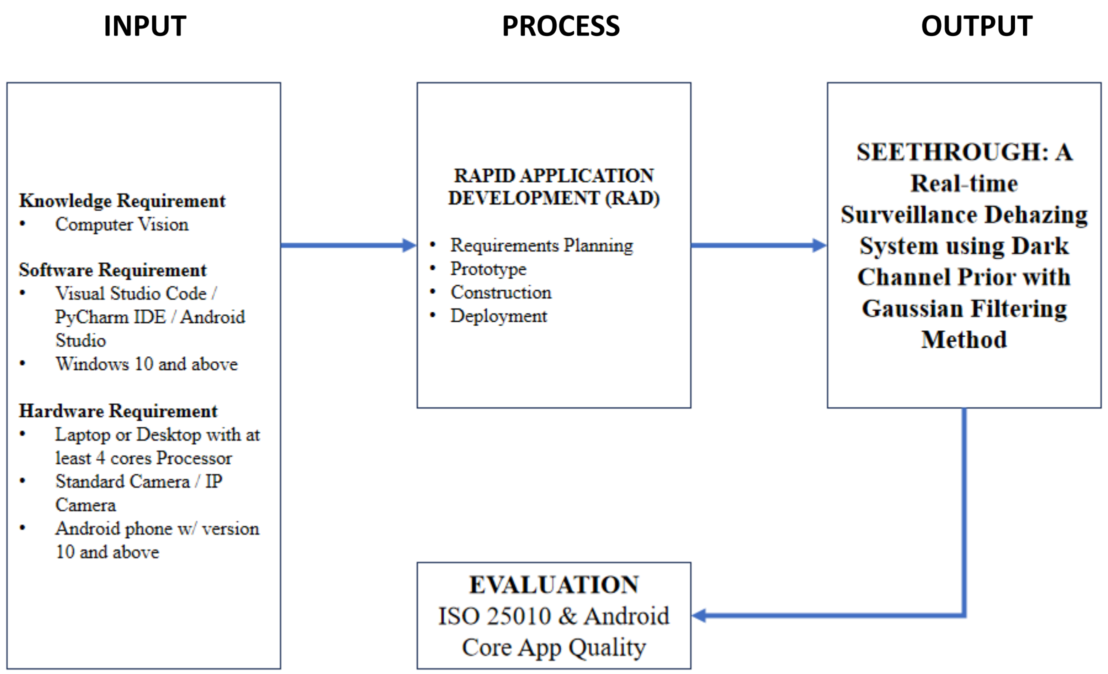
Based on the discussed merits and demerits of the identified dehazing algorithms, the researchers have decided that the system will be a DCP-based dehazing system that will work alongside a surveillance system, which is expected to run in real-time. To improve the dehazing capabilities and speed of the dehazing process, the gaussian filtering method will be used in the transmission map refinement stage of the DCP.

Conceptual Framework of the Study

The Conceptual Framework of the Study serves as a roadmap for accomplishing the study's objectives. It visualizes the Input, Process, and Output and eventually the evaluation of the finished system.

Figure 1

Conceptual Framework of the Study



Shown in Figure 1, the researchers were required to learn the fundamentals that are needed to create real-time surveillance dehazing using Dark Channel Prior with Gaussian Filtering method system. The researchers must be knowledgeable in using Java for mobile and Python for desktop together with mathematical libraries that are built-in on Python as it will be the foundation of Computer Vision. The researchers used synthetic datasets such as FRIDA to test the effectiveness of the algorithms used in the study.

The Rapid Application Development software development life cycle is the chosen methodology by the researchers since the project has to be refined based on the requirements gathered until the prototype is finally in an acceptable state. Since the study is descriptive, the researchers aim to make this study contribute to the real-time dehazing field. Below are the actual steps that the researchers did in order to build the project system using Rapid Application Development software development life cycle:

Requirements Planning

The researchers conducted an analysis on surveillance systems to determine the necessary features for the dehazing system.

Prototype

Once the requirements were gathered, the researchers developed the prototype of the product and had it tested by IT experts, subject matter experts, and end-users. After testing, the researchers gathered their feedback and made improvements on the prototype, until it is ready to enter the construction phase.

Construction

After prototyping the product, the researchers constructed the product in a live environment based on the suggestions of the users from the prototyping stage.

Deployment

The product is deployed on a website for the download links and Google Play to be officially used by the users.

Once the system was built, the researchers evaluated its acceptability based on ISO 25010:2023 standard for the desktop application and Android Core App Quality standard for the mobile application on the respondents.

Technical Definition of Terms

The technical definition of terms provides explanations of the terminologies of the area of the study. This is where it explains the technical terms found in the study.

Atmospheric Light

It refers to the light that is scattered and reflected by atmospheric particles.

Computer Vision

A field of Artificial Intelligence that trains computers to understand the real-world.

CPU

Stands for Central Processing Unit. It acts as the central brain of the computer which runs the machine's operating system and applications.

Dark Channel Prior

An algorithm that reads the low intensity channels of the RGB (Red, Green, Blue) on an image. Often referred to as "DCP" in this paper.

Dehazing

A process to improve image visibility caused by Hazed Phenomenon.

Gaussian Filtering

An algorithm to preserve or smoothen the edges on an image.

Haze

An event caused by atmospheric phenomena which obscure visibility and clarity of the sky.

Noise

It refers to the random color information that is not a feature or present in the scene.

SDK

Stands for Software Development Kit and it provides tools that allow developers to create software applications on a specific platform.

Transmission Map

It describes the portion of the light that is not scattered and reaches the camera.

RESEARCH DESIGN AND METHODOLOGY

This section explains the research design and methodology employed in the study. The research design provides the diagrams needed for the output of the system and provides explanation for data collection and analysis for the study.

Research Design

The research design that was used for this study is a quantitative research design. A quantitative research design helped the researchers in achieving the objectives of the study due to its suitable approach for this manuscript in assessing the dehazing algorithm's effectiveness through a quantitative metric such as PSNR and SSIM. The users' opinion or attitude towards the application can also be collected and analyzed through a Likert scale.

The researchers utilized Rapid Application Development (RAD) Software Development Life Cycle for the development of the manuscript project entitled "Real-Time Surveillance Dehazing using Dark Channel Prior with Gaussian Filtering Method". There are (4) components of RAD software development that the researchers utilized. These include Requirements Planning, Prototype, Construction, and Deployment.

Figure 2

Rapid Application Development Software Development Life Cycle

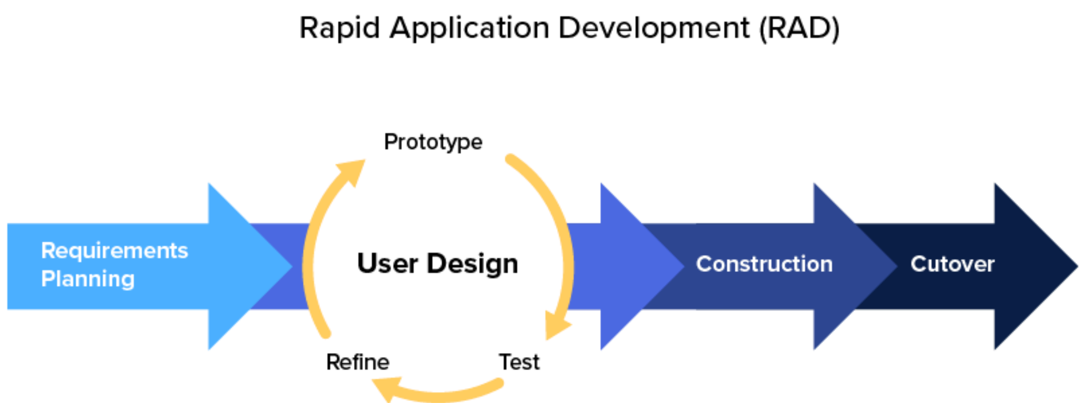


Figure 2 illustrates the process of the Rapid Application Development (RAD) Life Cycle. Below discusses each step pertaining to the actions of the life cycle:

1. **Requirements Planning**: The researchers identified the requirements needed in a dehazing system. This laid a foundation on designing and prototyping the system.
2. **Prototype**: The researchers developed the system with the intended functionalities during requirements planning phase as fast as the researchers can.
3. **Test**: After the researchers developed the prototype of the system, it was tested for its functionality and evaluated by the end-users.

4. **Refine**: Once the testing is over, the prototype was refined or had its functionality tweaked, depending if there is a module not implemented or lacking its intended use.
5. **Construction**: In this phase, the development of the system by the researchers was intensive in finding bugs. Feedbacks were also addressed during this stage.
6. **Deployment**: In this phase, the system is ready to deploy and involves intensive scale testing, technical documentation, issue tracking, final customization, and system simulation.

Fundamental Algorithm Used

The Dark Channel Prior algorithm was introduced by Kaiming He, Jian Sung, and Xiaoou Tang in their paper entitled "Single Image Haze Removal Using Dark Channel Prior" and it is later then used in the field of computer vision to address the problems in hazed images. The DCP-based dehazing algorithm composes four major steps to successfully dehaze an image, these are 1) Atmospheric Light Estimation 2) Transmission Map Estimation 3) Transmission Map Refinement and 4) Image Reconstruction (Lee et al, 2016). The advantages of using Dark Channel Prior as a dehazing algorithm has been recognized as an effective algorithm by many studies to eliminate haze from hazy images (Dwivedi, 2023).

Figure 3

Dark Channel Prior Dehazing Algorithm

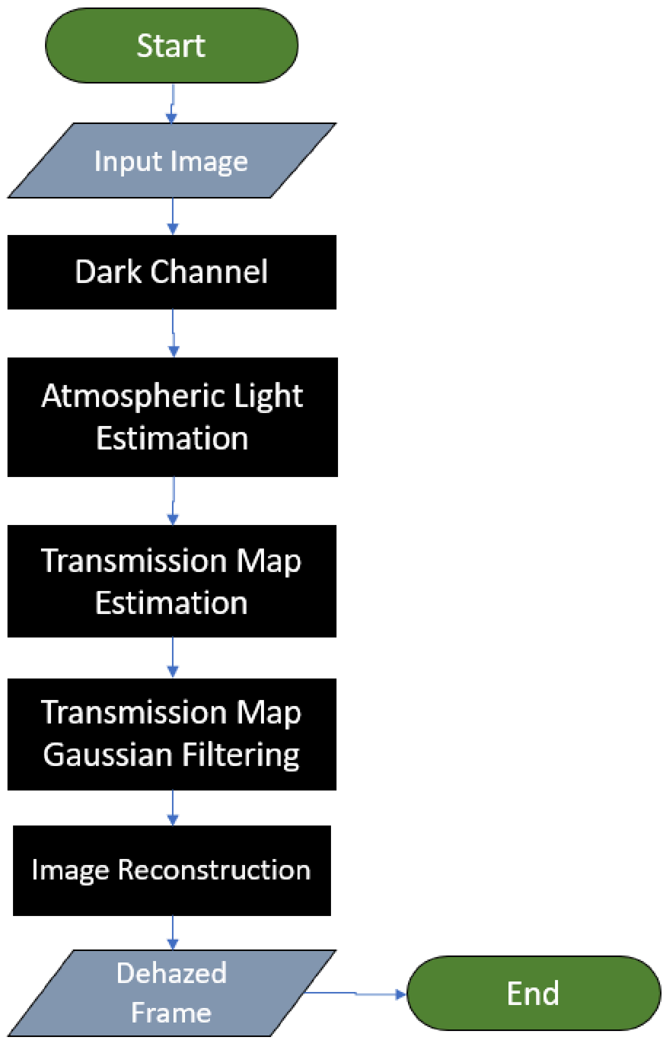


Figure 3 illustrates the flowchart of the study's Dark Channel Prior algorithm of the researchers referenced by the original Dark Channel Implementation of He et al. In the diagram, the image or frame acts as the input on the system. From this input, the dark channel is estimated based on the lowest pixel value per color channel.

Next is the atmospheric light estimation, where it finds the location of the high intensity pixels on the image. Then, the transmission map will be calculated wherein it estimates how much fog is present in the image. Afterwards, the transmission map will be refined using the Gaussian Filtering algorithm to reduce the noise and detect the edges on the image. Finally, the non-hazy image is reconstructed through the image reconstruction step, wherein the input image is now dehazed.

Strengths and Weaknesses of the Algorithm

The strengths and weaknesses of the dark channel prior and gaussian filter are discussed in this section.

Table 2

Strengths and Weaknesses of DCP

Strengths	Weaknesses
• Simple and effective. • Easy to implement. • Good defogging effects for landscape photos. • Ability to restore image with high quality.	• Computationally complex. • Color distortion in areas with massive gray or white colors. • Over darkening of image.

Table 2 shows the strengths and weaknesses of the dark channel prior algorithm. According to Wu et al. (2023), Yang et al. (2021), and Koranga and Kumawat (2020), the strengths of a DCP-based algorithm is primarily its simple implementation yet effective dehazing output. However, the dark channel prior also struggles with color distortion and over darkening of the image. Its computational complexity is the most pointed out weakness.

Table 3

Strengths and Weaknesses of Gaussian Filter

Strengths	Weaknesses
• Useful in removing color textures in the transmission map.	• Ineffective in sharpening a blurry transmission map.

Table 3 shows the strengths and weaknesses of the Gaussian filter. According to Koranga and Kumawat (2020), the gaussian filtering method is useful in removing any residual color textures in the transmission map which contributes to an inaccuracy when reconstructing the non-hazy image. However, it also struggles in sharpening a blurry transmission map which may leave out some halo residue from the dark channel estimation.

Dark Channel Prior with Gaussian Filtering Method Algorithm

Line Algorithm: Dark Channel Prior with Gaussian Filtering Method

1. *Initialize input variable for video feed*
2. *while True:*
3. *Read input frame*
4. *Is frame successfully read?*

```
if no:
    Output error message
    End loop
else
    Convert input frame to floating-point value [0.0 to 1.0]
    Split image color channels [R, G, B]
    Initialize dark_channel (DC) variable
    for i in range(pixels):
        Is R[i] < G[i]?
        if yes:
            Is R[i] < B[i]?
            if yes:
                Store R[i] to DC[i]
            else
                Store B[i] to DC[i]
        else
            Is G[i] < B[i]?
            if yes:
                Store G[i] to DC[i]
            else
                Store B[i] to DC[i]
    Enhance dark regions through erosion
    Return dark_channel of image
    Estimate Atmospheric Light (AL) using input frame and its DC
    Estimate Transmission Map (T) using input frame and its AL
    Refine T using Gaussian Filter
    Reconstruct image to acquire dehazed frame
    Fill recovered_image (R) variable with zeros (R = Frame[W , H])
    Loop through R, G, and B color channels
```

```
34.         for i in range (3):
35.             R = (frame[:, :, i] – AL[0, i]) / T + AL[0, i]
36.         Return dehazed frame
37.     Output dehazed frame
```

This pseudo-code provides the foundation in developing the SeeThrough Dehazing System as illustrated in Figure 4. The system has the following capabilities:

a) Take prerecorded video, image, or live video as input

b) Dark Channel computation

c) Atmospheric Light and Transmission Map Estimation

d) Reconstructing the non-hazy image based on the estimated parameters

e) Displaying the dehazed frame in real-time

Figure 4

Desktop Stream Input Process

Figure 4 illustrates how the desktop system acquires its input image in real-time for lines 1 to 4 of the dark channel prior with Gaussian filtering method algorithm.

Through the IP camera connected to the computer through its RTSP address, the dehazing system is provided with a live video feed. As the input device feeds the system with its video feed, the dehazing system treats this stream input as a series of images, called frames.

Figure 5

Mobile Stream Input Process

Figure 5 illustrates how the mobile application acquires its input image in real-time for lines 1 to 4 of the dark channel prior with Gaussian filtering method algorithm. Through the smartphone's camera, the mobile dehazing application is provided with a live video feed. The mobile application uses this live video feed as its input stream.

Figure 6

Dark Channel Prior Estimation Demonstration

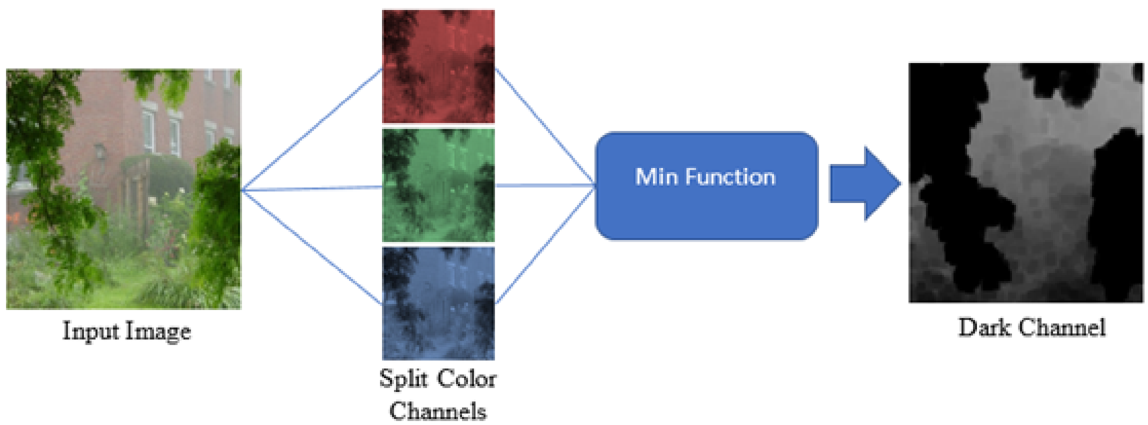
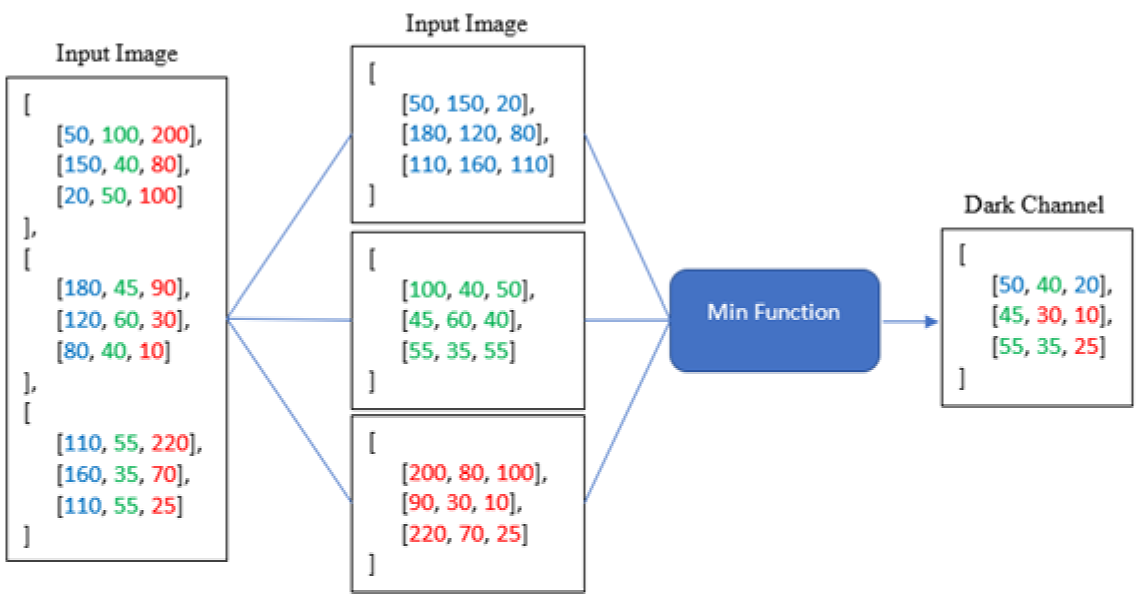


Figure 6 illustrates the dark channel prior estimation demonstration for lines 10 to 25 of the dark channel prior with Gaussian filtering method algorithm. After the dehazing system receives an input image or frame, the first step of the dehazing process, the DCP estimation, begins. Given an input frame, the dark channel prior function returns the dark channel values of the hazy image.

Figure 7

Dark Channel Prior Estimation Demonstration Pixel Value



To better understand how the dark channel values are acquired, it is easier to visualize an image as a series of pixel values. Figure 7 shows the pixel value of a 3x3 image, its red, green, and blue color channel values, and the output dark channel value after processing the three color channels into a min function to get the lowest pixel value among the three color channels.

Equation 1

Mathematical Model of Dark Channel Prior

The dark channel is computed as follows:

$$J^{dark}(x) = min_{y\in\Omega(x)}(min_{c\in\{r,\, g,\, b\}} J^{c}(y))$$

where:

J^{c} – is the intensity of the color channel $c\in\{r, g, b\}$.

$\Omega(x)$ – is a local patch centered at x.

min – is a minimum value function.

y – is an element of the local patch $\Omega(x)$.

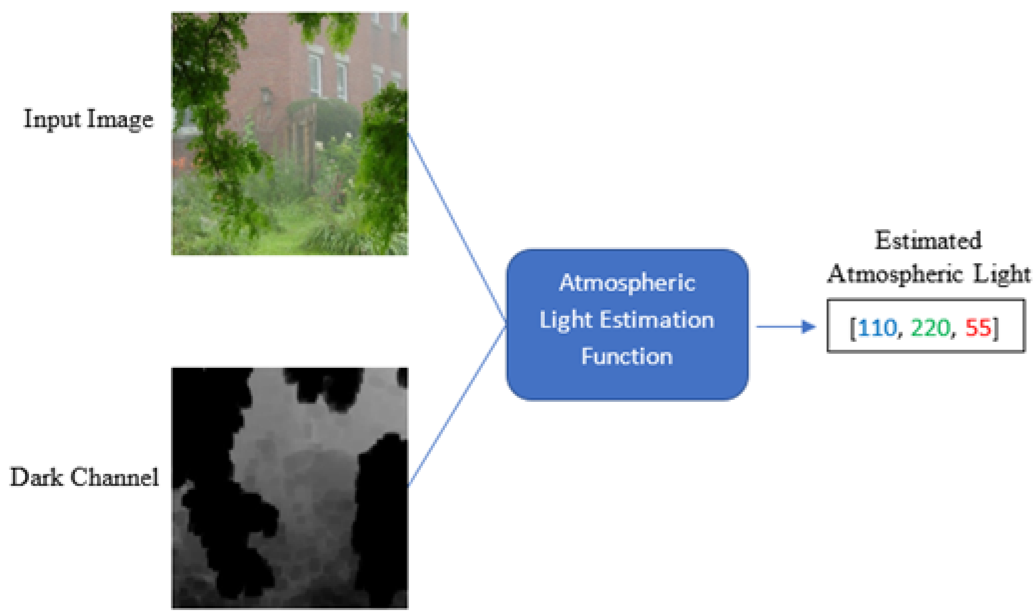
The Source of Equation 1 is retrieved from the study of Lee et al. (2016) from SpringerOpen: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-016-0104-y

Equation 1 represents the mathematical model of the dark channel prior, where min represents an element-wise minimum value function, which means that the algorithm will compute for the minimum pixel value “per element” of the image array. *y* is a pixel element of the image's local patch (denoted as Ω) centered at index x. c represents the color channel of the image, namely the red (R), green (G), and blue (B) channels. J^c is any of the three color channels, and $J^c(y)$ represents a pixel value in a color channel.

The output of this computation is a new color channel called J^{dark}, or simply the dark channel, which contains the lowest intensity value of the RGB color channels at each pixel in the image.

Figure 8

Atmospheric Light Estimation Demonstration



Equation 2

Mathematical Model of Atmospheric Light Estimation

The atmospheric light is computed as:

$$A = I(argmax_x(I^{dark}(x)))$$

where:

I – image

argmax – maximum value operation

x – pixel value

The Source of Equation 2 is retrieved from the study of Lee et al. (2016) from SpringerOpen: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-016-0104-y

Figure 8 and Equation 2 illustrates the atmospheric light estimation for line 28 of the dark channel prior with Gaussian filtering method algorithm. In this process, the function takes the input image and its dark channel value and computes for the highest intensity dark channel value of the image per color channel, producing an array with a (1, 3) shape, which will be used as the global atmospheric light.

Figure 9

Transmission Map Estimation Demonstration

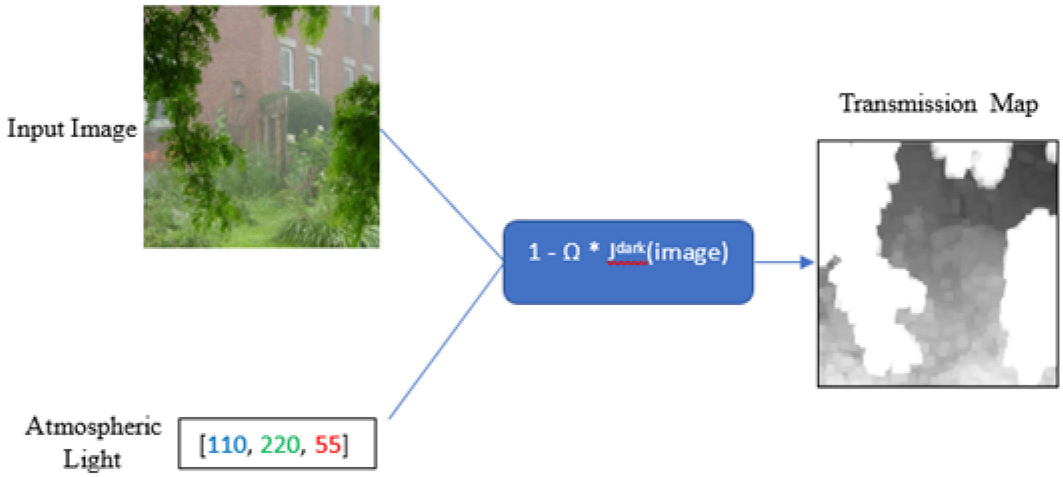


Figure 9 illustrates the transmission map estimation function demonstration for line 29 of the dark channel prior with Gaussian filtering method algorithm. ω (omega) represents a constant value, which is a commonly used parameter value for the transmission map. The transmission map is estimated by dividing the image value by the estimated atmospheric light, plugging the result into the dark channel function, multiplied by the omega constant, and subtracting these from a constant 1.

Equation 3

Mathematical Model of Transmission Map Estimation

The transmission map is computed as:

$$\tilde{t} = 1 - \omega \, min_{y \in \Omega(x)} (min_{c \in \{r,g,b\}} \frac{I^c(y)}{A^c})$$

where:

ω – a constant (0 < ω < 1).

min – minimum value function.

$\Omega(x)$ – Local patch centered at x.

c – a color channel.

y – element of local patch $\Omega(x)$.

A – Estimated airlight

The Source of Equation 3 is retrieved from the study of Lee et al. (2016) from SpringerOpen: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-016-0104-y

Equation 3 illustrates the mathematical model of the transmission map estimation that was used in this study. A constant ω where 0 < ω < 1 is set to 0.95 in the context of the SeeThrough system, which determines the strength of the dark channel prior. A high omega constant gives more emphasis on the dark channel, while a lower omega constant lowers emphasis. The minimum function in the equation is simply the reusage of the dark channel prior equation on the given variables.

In the context of the proposed system, following the results of Lee et al. (2016), the local image patch is set to 3x3 and the omega constant is set to 0.90.

Figure 10

Transmission Map Refinement Demonstration

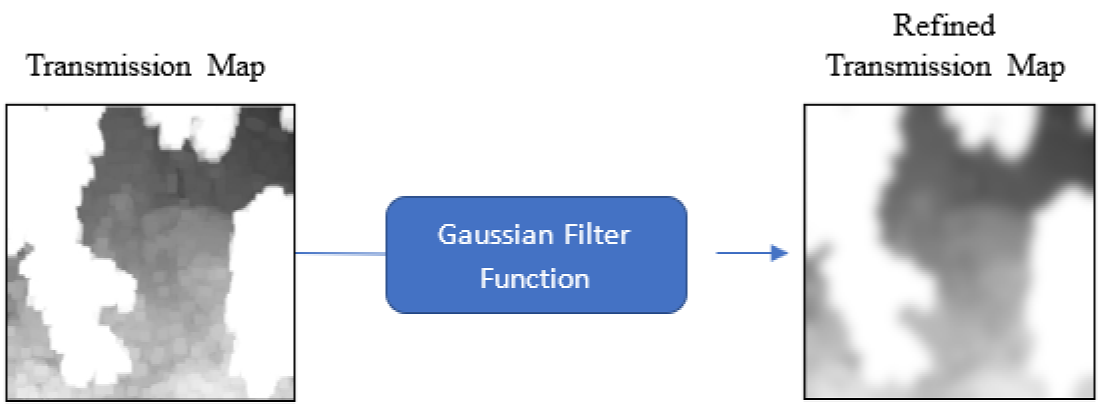


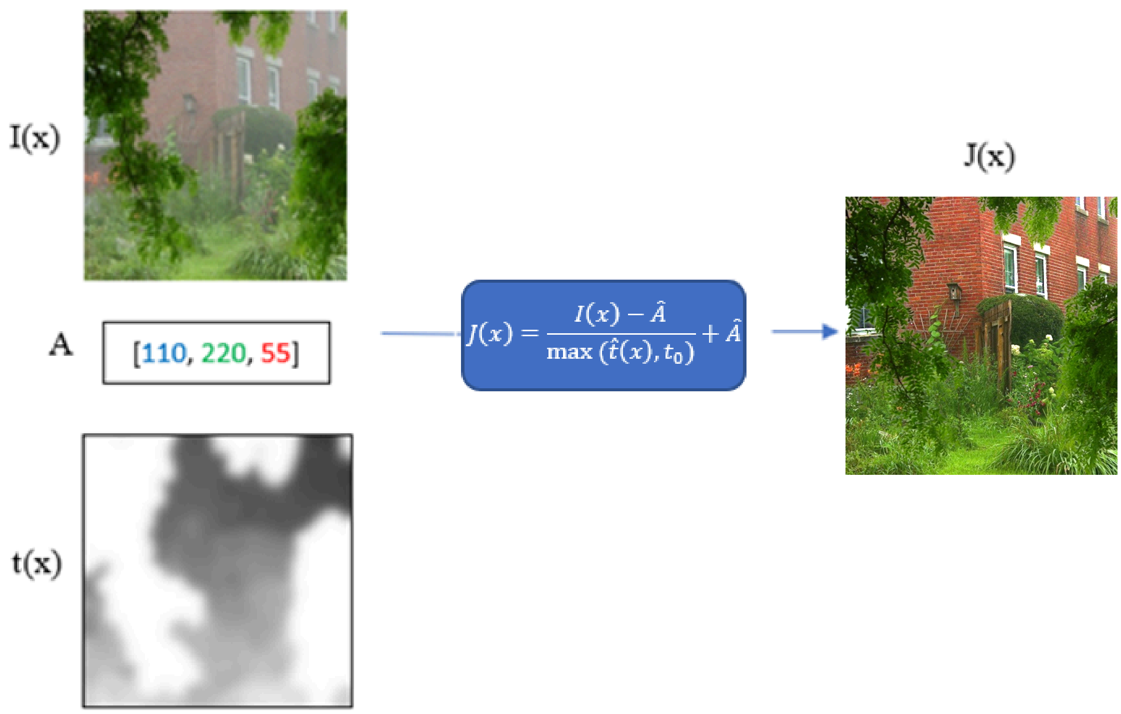
Figure 10 illustrates the transmission map refinement stage of the dehazing process for line 30 of the dark channel prior with Gaussian filtering method algorithm. By smoothing out the transmission map's edges, the resulting halo of the dehazing process is reduced. In the context of this paper, the gaussian filter is taken from SciPy's library. The system simply supplies their implementation with the estimated transmission map and a radius value to determine the gaussian filter's strength.

Image Reconstruction

In this stage, the haze-free image is finally recovered using the following equation

Figure 11

Image Reconstruction Demonstration



Equation 4

Mathematical Model of Image Reconstruction

The image is reconstructed using the following equation:

$$J(x) = \frac{I(x) - \hat{A}}{max(\hat{t}(x), t_0)} + \hat{A}$$

where:

I – is the hazy input image

$\hat{A}$ – is the estimated atmospheric light

max – is a maximum value function

$\hat{t}$ – is the estimated transmission map

t_0 – is a constant value

x – pixel value

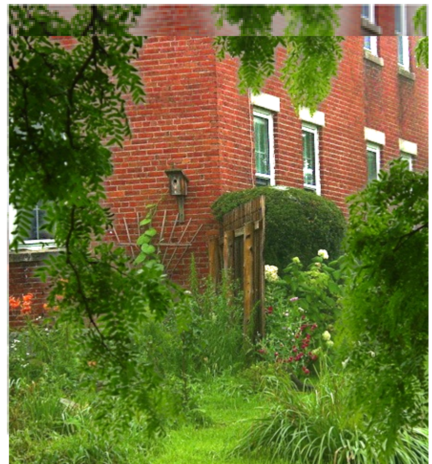
The Source of Equation 4 is retrieved from the study of Lee et al. (2016) from SpringerOpen: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-016-0104-y

Equation 4 shows the image reconstruction computation for each pixel *x* for lines 31 to 36 of the dark channel prior with Gaussian filtering method algorithm. The *max* function is an element-wise maximum function that takes a transmission map $\hat{t}$ and a constant t_0. The constant value t_0 is a typical value to avoid having a denominator with a low value. Most DCP-based dehazing methods initializes t_0 as 0.1. For simplicity, the SeeThrough system also initialized t_0 to 0.1.

Figure 12

Image Reconstruction Demonstration cont.

Input Image



Output Image

Figure 11 and 12 illustrates the final step of the dehazing process on line 37 of the dark channel prior with Gaussian filtering method algorithm, where I(x) represent the input image, A represents the estimated atmospheric light, t(x) represents the refined transmission map, and J(x) represents the reconstructed non-hazy image.

Mathematical Models

Equation 5

Mathematical Model of Dark Channel Prior

$$J^{dark}(x) = min_{y \in \Omega(x)}(min_{c \in \{r,g,b)}(J^{c}(y)))$$

where:

J^c = Intensity of the color channel $c \in \{r, g, b\}$.

$\Omega(x)$ = Local patch centered at x.

min = Minimum value function.

y = An element of $\Omega(x)$ / Local Patch.

The Source of Equation 5 is retrieved from the study of Lee et al. (2016) from SpringerOpen: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-016-0104-y

Equation 5 represents the mathematical model of the dark channel prior. The result of this model will be used as prior knowledge to estimate the atmospheric light and transmission map. For simplicity, assume that *J* is a 3x3 image. This means that there are a total of 9 pixels. A pixel contains three values, representing the color intensity of the three color channels red, green, and blue at its position.

Figure 13

A Visual Representation of J

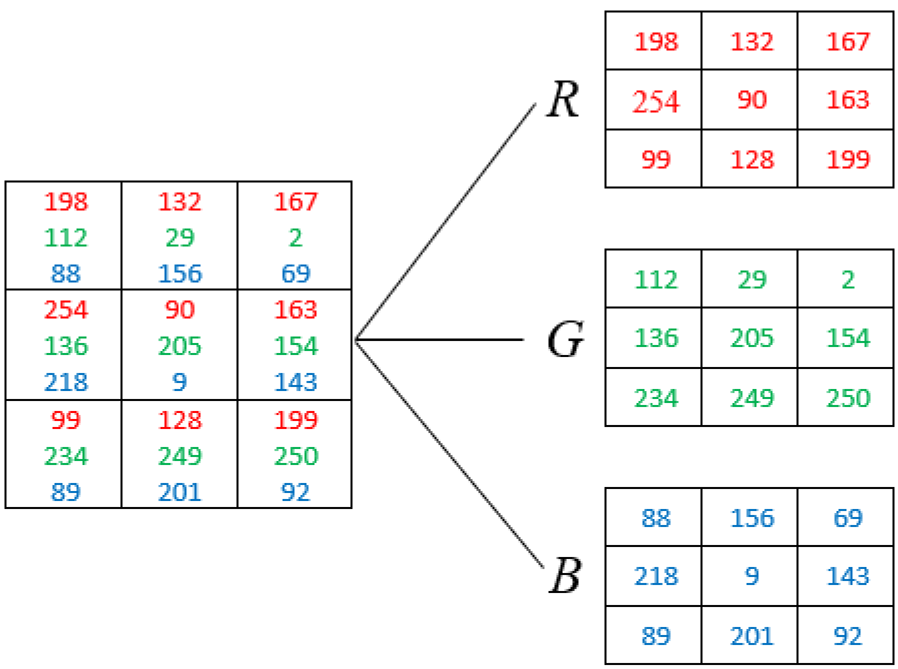
=

198 112 88	132 29 156	167 2 69
254 136 218	90 205 9	163 154 143
99 234 89	128 249 201	199 250 92

Figure 13 shows the RGB values of the image per pixel. Before going through the min function, the image's color channels must first be split into three separate arrays.

Figure 14

Split Color Channel Values



198 112 88	132 29 156	167 2 69
254 136 218	90 205 9	163 154 143
99 234 89	128 249 201	199 250 92

R

198	132	167
254	90	163
99	128	199

G

112	29	2
136	205	154
234	249	250

B

88	156	69
218	9	143
89	201	92

Figure 14 shows the image being split into three separate arrays. These arrays represent a color channel and the pixel's color intensity at that position.

Then, the dark channel prior will be computed using an element-wise min function, traversing through each pixel position via a for-loop. The dark channel is an array with the same shape as the image. For example, if the image is 3 x 3, then the dark channel will also consist of 3 x 3 pixels.

Figure 15

Dark Channel Prior Min Function Demonstration

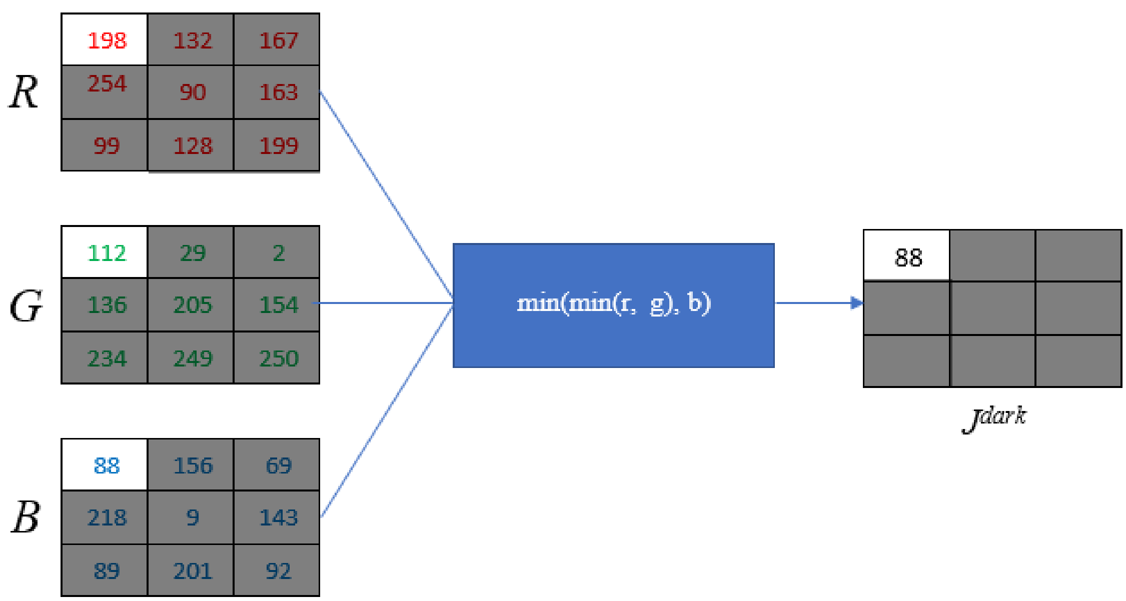


Figure 15 shows the first iteration. The index (0, 0) is checked across all color channels. Between the three values being checked, the smallest value is stored into the J^{dark} array. In this case, between 198, 112, and 88, the blue channel's value "88" is the lowest. Therefore, 88 is stored into the dark channel array. This type of process reiterates through all pixel positions.

Figure 16

Dark Channel Prior Min Function Demonstration cont.

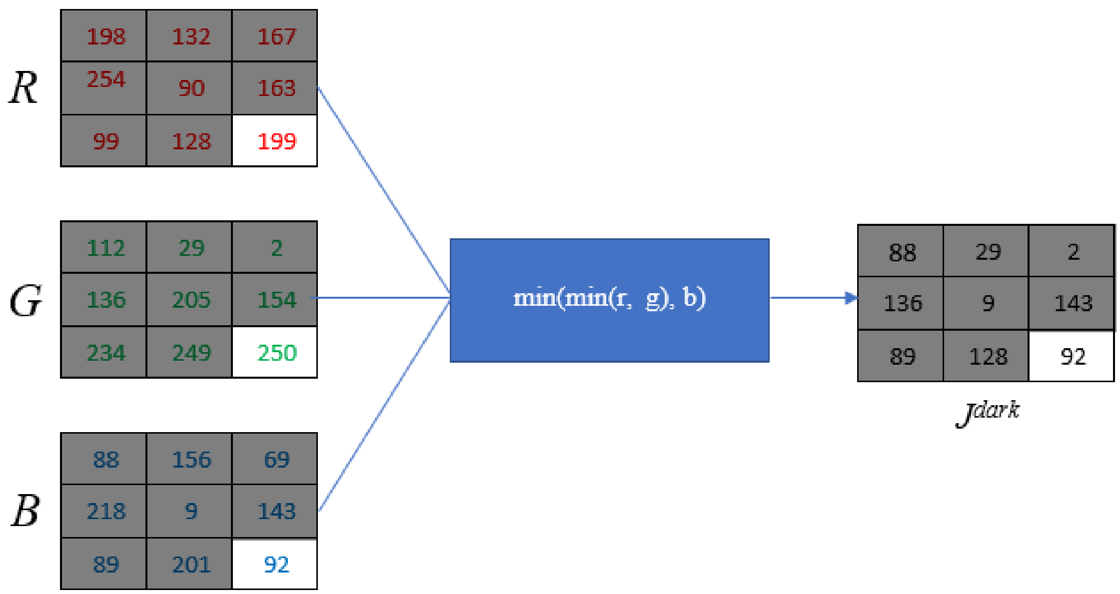


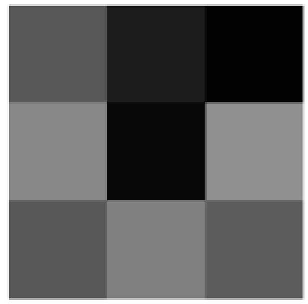
Figure 16 shows the last iteration of the dark channel function. Between the three values, 92 is the lowest. Therefore, 92 will be stored into the dark channel array.

Figure 17

The Dark Channel Prior

88	29	2
136	9	143
89	128	92

=



J^{dark}

To better visualize the dark channel prior, figure 17 shows the image equivalent of the J^{dark} array.

To further enhance the dark channel, the array will undergo a morphological operation called erosion with the use of the OpenCV library. In the context of the proposed system, the erosion patch size is set to 15x15 for the dark channel prior and 3x3 for the transmission map estimation. To simplify the explanation, the demonstration will use a 3x3 kernel over the 3x3 dark channel array.

Figure 18

Dark Channel Erosion

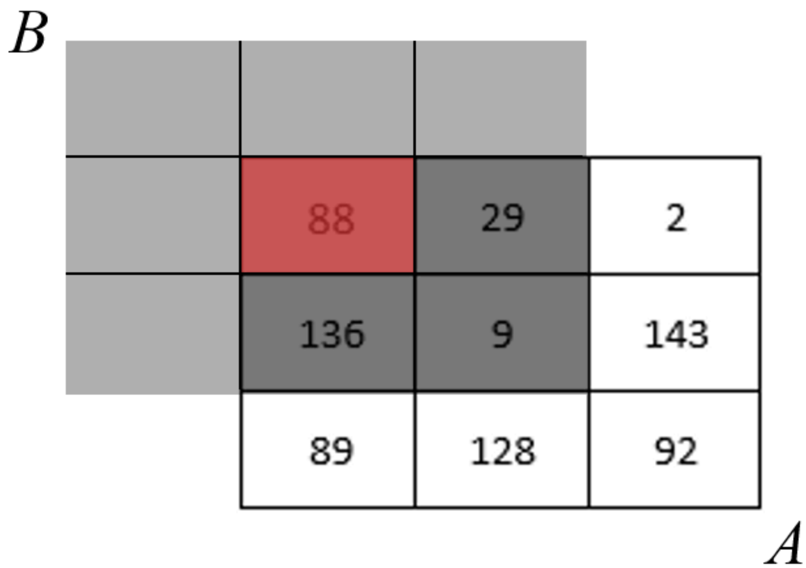


Figure 18 shows the convolution of a kernel *B* and the dark channel *A*. According to the documentation of OpenCV, erosion convolves an image *A* with some kernel *B* of a specified patch size.

Figure 19

Dark Channel Erosion cont.

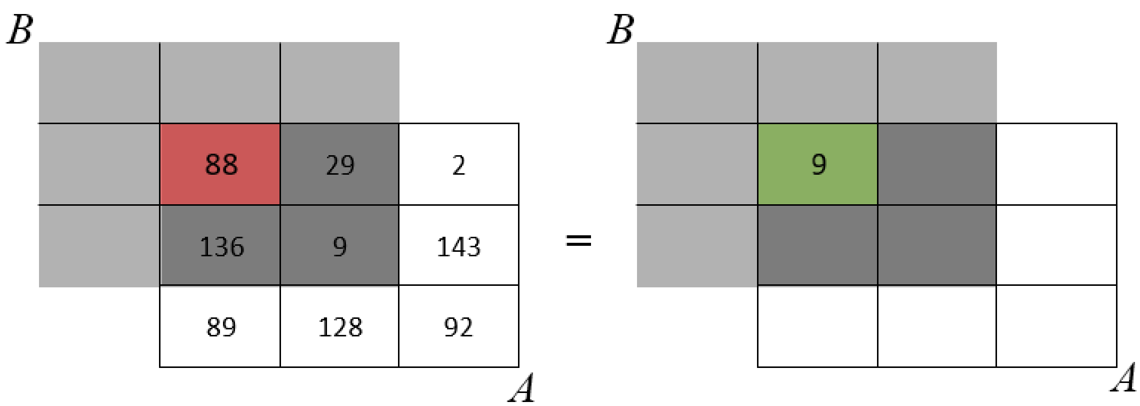


Figure 19 shows the selection process of the erosion operation at index (0, 0). The erosion operation computes the minimal pixel value overlapped by *B* and replaces the dark channel's pixel under the anchor point (center of the kernel) with that minimal value. In this example, the local minima overlapped by *B* is 9. Therefore, the image's pixel at the anchor point is set to 9, replacing 88. This is done iteratively across all image positions.

Figure 20

Dark Channel Erosion

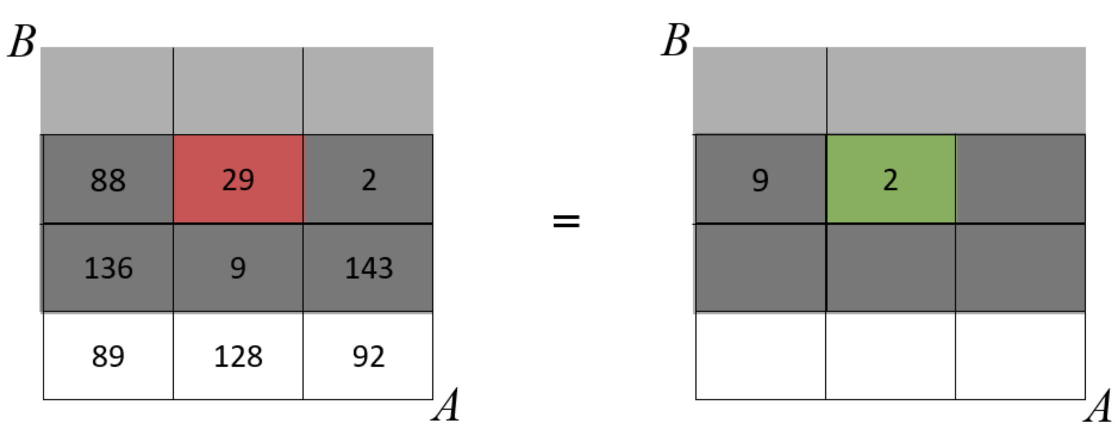


Figure 20 shows the selection process of the erosion operation at index (0, 1). In this example, the local minima overlapped by *B* is 2. Therefore, the image's pixel at the anchor point is set to 2, replacing 29.

Figure 21

Dark Channel Erosion

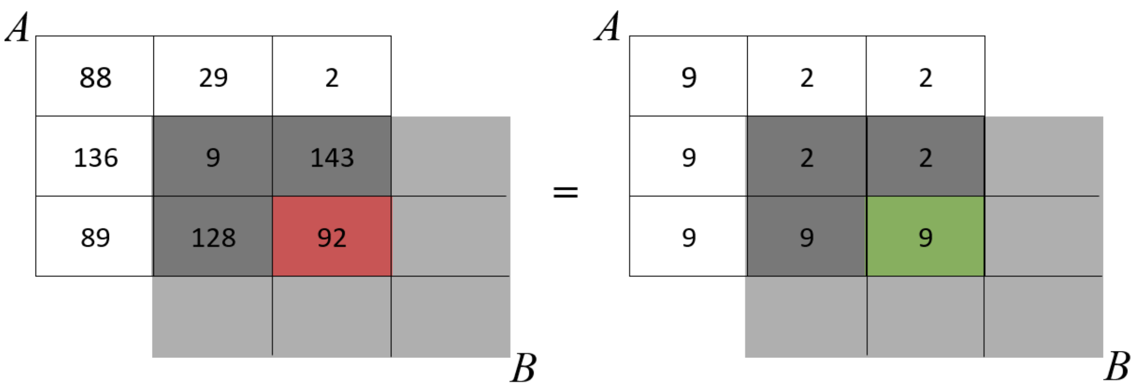


Figure 21 shows the last iteration of erosion at index (2, 2). In this example, the local minima overlapped by *B* is 9. Therefore, the image's pixel at the anchor point is set to 9, replacing 92.

Figure 22

Eroded Dark Channel

9	2	2
9	2	2
9	9	9

J^{dark}

=

J^{dark}

To better visualize the eroded dark channel, figure 22 shows the image equivalent of the eroded J^{dark} array.

Equation 6

Mathematical Model of Atmospheric Light Estimation

$$A = \frac{\sum I(argmax_x(I^{dark}(x))}{n}$$

where:

$\sum$ – summation of

I – image

I^{dark} – image dark channel

$argmax$ – maximum value operation

x – pixel value

n – number of estimated brightest pixels

The Source of Equation 6 is retrieved from the study of Lee et al. (2016) from SpringerOpen: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-016-0104-y

Equation 6 shows how the atmospheric light is estimated. The atmospheric light is used to estimate the global illumination of the frame by taking the $p\%$ brightest pixels. In the context of this study, $p = 0.0001$. To avoid having a low amount of brightest pixels, a max function is applied as $max(p\%, 1)$. This means that if the number of brightest pixels is estimated to be less than 1, the system will automatically select at least one bright pixel.

In this example, the number of brightest pixels is determined to be $max(0.0009, 1)$. Therefore, the amount of pixels that will be selected is $n = 1$.

Figure 23

Brightest Pixel Selection

9	2	2
9	2	2
9	9	9

$argmax_x(I^{dark}) = (2, 2)$

198 112 88	132 29 156	167 2 69
254 136 218	90 205 9	163 154 143
99 234 89	128 249 201	199 250 92

$I(2, 2)$

With the 3x3 image and its dark channel prior in mind, the brightest pixel is selected as shown in figure 23 through an *argmax* function. The algorithm has determined that the position of the pixel with the brightest intensity dark channel is at index (2, 2). Since $n = 1$, the estimated atmospheric light A remains as is after division. Therefore, the atmospheric light A is [199, 250, 92].

Equation 7

Mathematical Model of Transmission Map Estimation

$$\tilde{t} = 1 - \omega \, min_{y \in \Omega(x)} (min_{c \in \{r,g,b\}} \frac{I^c(y)}{A^c})$$

where:

I – hazy image

ω – a constant (0 < ω < 1).

min – minimum value function.

$\Omega(x)$ – Local patch centered at x.

c – a color channel.

y – element of local patch $\Omega(x)$.

A – Estimated airlight

The Source of Equation 7 is retrieved from the study of Lee et al. (2016) from SpringerOpen: https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-016-0104-y

In the context of the developed system, the transmission map was used to estimate the amount of haze present at each pixel position. Given a hazy image's color channel I^c at position y and its atmosphere light A^c, $\frac{I^c(y)}{A^c}$ is calculated as follows:

Figure 24

Estimating the Transmission Map

198 112 88	132 29 156	167 2 69
254 136 218	90 205 9	163 154 143
99 234 89	128 249 201	199 250 92

$$I = I^c \begin{bmatrix} 198 & 132 & 167 \\ 254 & 90 & 163 \\ 99 & 128 & 199 \end{bmatrix}$$

Figure 24 shows I^c where $c = 0$. After selecting the first color channel in the hazy image array, the algorithm divides it by the atmospheric light at position c. Using numpy, an element-wise division operation is performed.

Figure 25

Estimating the Transmission Map cont.

$$I^0 \begin{bmatrix} 198 & 132 & 167 \\ 254 & 90 & 163 \\ 99 & 128 & 199 \end{bmatrix} \;/\; A^0 \;[199] = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$I^1 \begin{bmatrix} 112 & 29 & 2 \\ 136 & 205 & 154 \\ 234 & 249 & 250 \end{bmatrix} \;/\; A^1 \;[250] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$I^2 \begin{bmatrix} 88 & 156 & 69 \\ 218 & 9 & 143 \\ 89 & 201 & 92 \end{bmatrix} \;/\; A^2 \;[92] = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}$$

Figure 25 shows all three iterations of $\frac{I^c(y)}{A^c}$. Notice that the quotients are whole numbers. To simplify the demonstration, assume that all inputs and outputs are integers.

Figure 26

Estimating the Transmission Map cont.

$$I^0 \begin{bmatrix} 198 & 132 & 167 \\ 254 & 90 & 163 \\ 99 & 128 & 199 \end{bmatrix} \;/\; A^0 \;[199] = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$I^1 \begin{bmatrix} 112 & 29 & 2 \\ 136 & 205 & 154 \\ 234 & 249 & 250 \end{bmatrix} \;/\; A^1 \;[250] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$I^2 \begin{bmatrix} 88 & 156 & 69 \\ 218 & 9 & 143 \\ 89 & 201 & 92 \end{bmatrix} \;/\; A^2 \;[92] = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}$$

0 0 0	0 0 1	0 0 0
1 0 2	0 0 0	0 0 1
0 0 0	0 0 2	1 1 1

$$\frac{I^C(y)}{A^C}$$

Figure 26 shows the output result of $\frac{I^c(y)}{A^c}$. After the computation, the color channels are recombined into one image. The next step in the equation is $min_{y\in\Omega(x)}(min_{c\in\{r,g,b\}}\frac{I^c(y)}{A^c})$, which simply means to get the dark channel value of $\frac{I^c(y)}{A^c}$ using equation 1. Therefore, the result is as shown in figure 27:

Figure 27

Estimating the Transmission Map cont.

0 0 0	0 0 1	0 0 0
1 0 2	0 0 0	0 0 1
0 0 0	0 0 2	1 1 1

$\frac{I^C(y)}{A^C}$

0	0	0
0	0	0
0	0	0

$min_{y\in\Omega(x)}\,(min_{c\in\{r,g,b\}}\frac{I^C(y)}{A^C})$

After computing the dark channel of $\frac{I^c(y)}{A^c}$, the next step is to multiply it by the constant ω. In the context of this study, ω is set to 0.90.

Figure 28

Estimating the Transmission Map cont.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times 0.90 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 28 shows the result of multiplying the dark channel of $\frac{I^c(y)}{A^c}$ by the ω value *0.90.* The algorithm then performs one last element-wise operation, which is to subtract this output from a constant 1.

Figure 29

Estimating the Transmission Map cont.

$$1 - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}_{\tilde{t}(x)}$$

Figure 29 shows the returned transmission map. In this example, the transmission map is an array filled with ones.

Project Requirements Specification

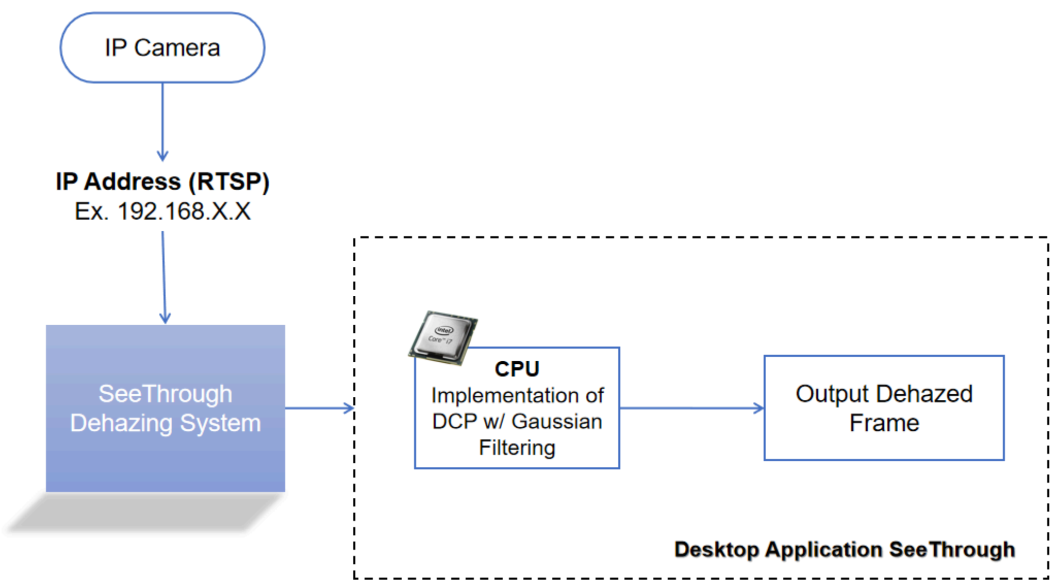
This section tackles the resources upon making the project system. It covers the architecture, specifications, and the communication of the system to the users.

Systems Architecture

The system architecture model shows the flow of communication and interaction between the different components of the system, starting from the input device up to the output device.

Figure 30

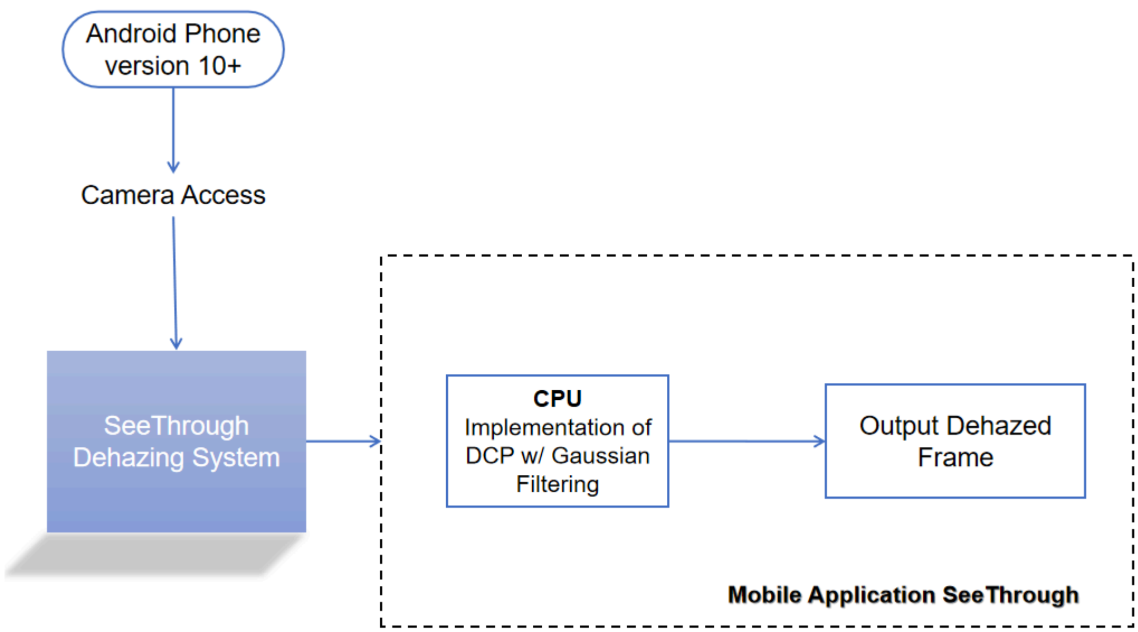
Systems Architecture of the Project - Desktop Version



As shown in Figure 30, the IP Camera is the main device to be used to capture the frames but in order to see the captured frames by the computer, the user is required to input the RTSP (Real-Time Streaming Protocol) address on the system. Once the RTSP address has been read by the system, the computer would be able to connect with the IP camera and therefore every frame recorded by the IP Camera would undergo the process of Dark Channel Prior and Gaussian Filtering methods to produce a haze-free frames in real-time.

Figure 31

System Architecture of the Project - Mobile Version



As shown in figure 31 above, the Android phone with versions 10 and above is the main device to be used, but in order to capture frames from the device's camera, the user is required to grant SeeThrough permission to use the camera on the device. Once the user allowed the permissions on the device's camera, the camera would capture every frame that would undergo the process of Dark Channel Prior and Gaussian Filtering methods to produce haze-free frames in real-time.

Software Specifications

In this section, the list of software used in developing the project system on their respective devices is discussed.

Desktop Application

The software requirements for the desktop application of the SeeThrough dehazing system include Microsoft Visual Studio Code & PyCharm as the primary code editor, and a Windows 10 or above operating system for compatibility. The system uses Python 3.11.4 and its various libraries to function such as OpenCV 4.8.0 for image and video processing, threading to utilize CPU multithreading, NumPy to perform mathematical operations, and PyQt5 for the graphical user interface. These software specifications ensure that the system can effectively and efficiently operate, while also providing the necessary tools and resources for the developers to customize and enhance the system's features.

Mobile Application

The software requirements for the mobile application of the SeeThrough dehazing system include Android Studio as the primary code editor, and an Android versions 10 and above operating system for compatibility. The system uses Java and OpenCV mobile for image and video processing. These software specifications ensure that the system can effectively and efficiently operate, while also providing the necessary tools and resources for the developers to customize and enhance the system's features.

OpenCV 4.8.0

To process images and videos, the developers used the OpenCV library.

Python 3.11.4

Python is an interpreted, high-level, and general-purpose programming language. Python is the programming language that was used to develop the SeeThrough Surveillance Dehazing System.

Java

Java is a high-level, class-based, object-oriented programming language. Java is the programming language that was used to develop the mobile version of SeeThrough Dehazing System.

Microsoft Visual Studio Code

Visual Studio Code is a multipurpose IDE built by Microsoft which includes support for hundreds of programming languages, including Python, offering a range of extensions and a customizable user interface.

Android Studio

Android Studio is the official integrated development environment for Google's Android operating system. Android Studio is used as the main IDE for the mobile application development of SeeThrough Dehazing System.

Windows 10 or above versions

The operating system for the desktop application used in running and testing SeeThrough using Dark Channel Prior with Gaussian Filtering Method.

Android 10 and above versions

The operating system for the mobile application used in running and testing SeeThrough using Dark Channel Prior with Gaussian Filtering Method.

Hardware Specifications

The system is recommended to run on a 4-core CPU. These hardware specifications can enhance the performance of the system and effectively run the real-time dehazing with Gaussian Filtering method. The system is also intended to work with IP Cameras to take live input videos. To test the algorithm, the researchers used TC70 Camera as their input device.

Laptop

The laptop used to develop the system is powered by Intel Core i7-7820HQ 2.9Ghz 4 Cores for the processor and NVIDIA QUADRO M2200 with 4GB of GDDR5 Memory and 128-bit bus Graphics Card. The Operating System used was a 64-bit paired with 16GB of RAM and a 500GB hard drive.

Camera

The camera used in Real-Time surveillance is the Tapo TC70 which has RTSP Support and can connect via wireless connectivity of IEEE 802.11b/g/n, 2.4 GHz Wi-Fi. It has a frame rate of 15 FPS and a video compression of H.264 with a maximum resolution of 1080p HD.

Android Device

The android device used for mobile app development was a Redmi Note 12 4G powered by the Qualcomm's 6nm Snapdragon 685 chipset. It had 6GB RAM with RAM speed of 2133 MHz, it runs on Xiaomi's HyperOS 1.0.7.0 based on Android 14. The chipset consists of 8 CPU Threads, 4 x 2.8 GHz and 4 x 1.9 GHz. The GPU is a Qualcomm Adreno 610 GPU clocked at 1260MHz.

Object Modeling

This portion of the design of the application demonstrates the expected client of the application and the capacities that are accessible for the clients to utilize.

Use Case Diagram

The Use Case Diagram is a visual representation of the interactions between a system and its actors, visualizing the flow on how the system must perform and communicate with the user.

Figure 32

Use Case Diagram of Desktop System

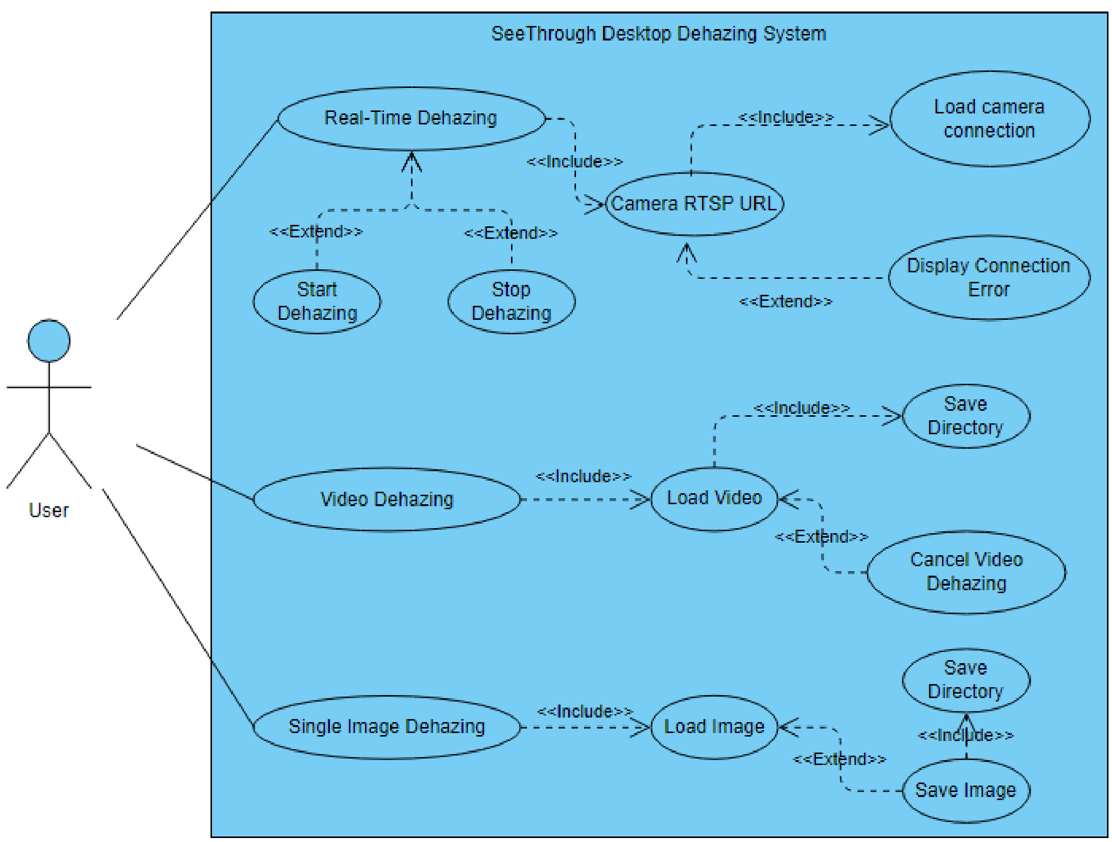


Figure 32 illustrates the use case diagram which represents the high-level scope of the SeeThrough desktop system that the user has direct interaction with. These high-level scopes are: Real-Time Dehazing, Video Dehazing, and Single Image Dehazing. In Real-Time Dehazing, the user can choose an IP camera as an input device and the system processes its input into the dehazing algorithm and shows the dehazed footage in real-time. Real-time dehazing has an “include” relationship with Camera Selection, where the user can enter their existing IP camera’s Real Time Streaming Protocol (RTSP) address. It loads the system’s connection to the IP camera, and if a connection error occurs, an error prompt will be displayed. After the connection has been established, the user has the option to either start or stop the real-time dehazing process. In Video Dehazing, the user can select a video file as an input and the system outputs a dehazed video file at the user-specified save directory. During the dehazing process, the user can choose to cancel the process. In Single Image Dehazing, the user can input an image file and the system outputs a dehazed image. The user can choose to save the dehazed image file in the user-specified save directory.

Figure 33

Use Case Diagram of Mobile Application

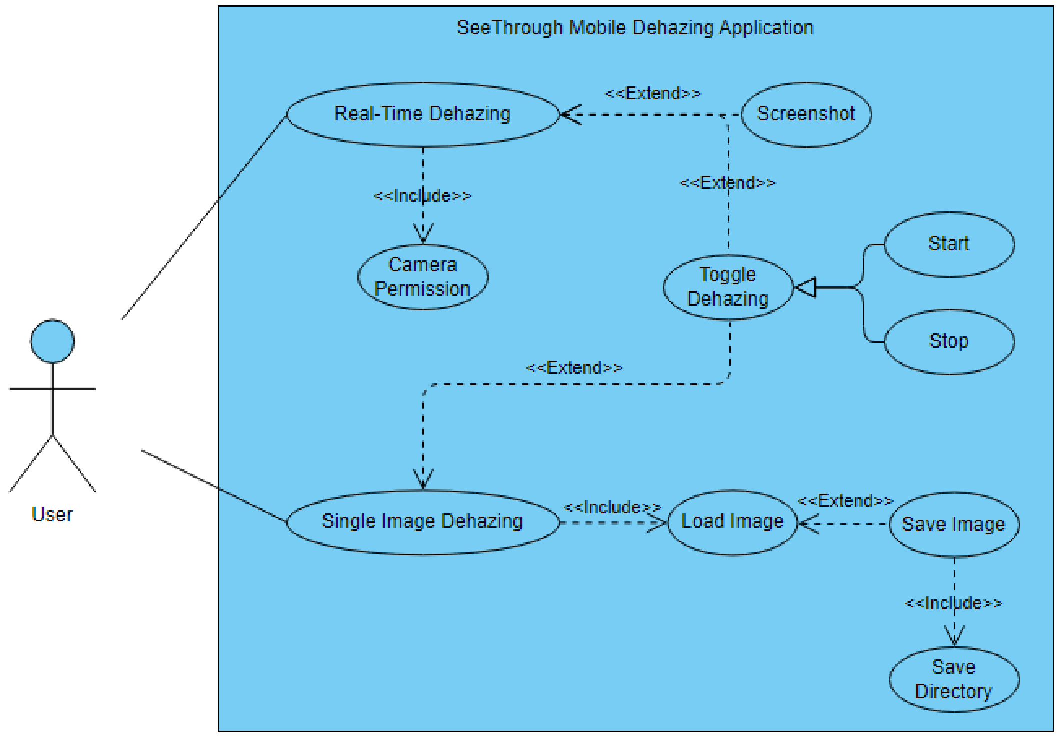


Figure 33 illustrates the use case diagram which represents the high-level scope of the SeeThrough mobile application that the user has direct interaction with. These high-level scopes are: Real-Time Dehazing and Single Image Dehazing. The real-time dehazing use case requires camera permission from the user. The user can then toggle the dehazing feature on or off through a toggle switch. The user can also choose to capture a screenshot during the dehazing process. In the single image dehazing use case, the user can load a hazy image and toggle the dehazing switch on or off. The user can also choose to save the dehazed image in the save image directory.

Sequence Diagram

The sequence diagram represents the SeeThrough Dehazing System and the Mobile application. The SeeThrough Dehazing System has three options to choose from: Real-Time, Single Image, and Video Input. The SeeThrough Mobile Dehazing Application has two options Single Image Dehazing and Real-time Dehazing.

Figure 34

Real-Time Dehazing Module Sequence Diagram

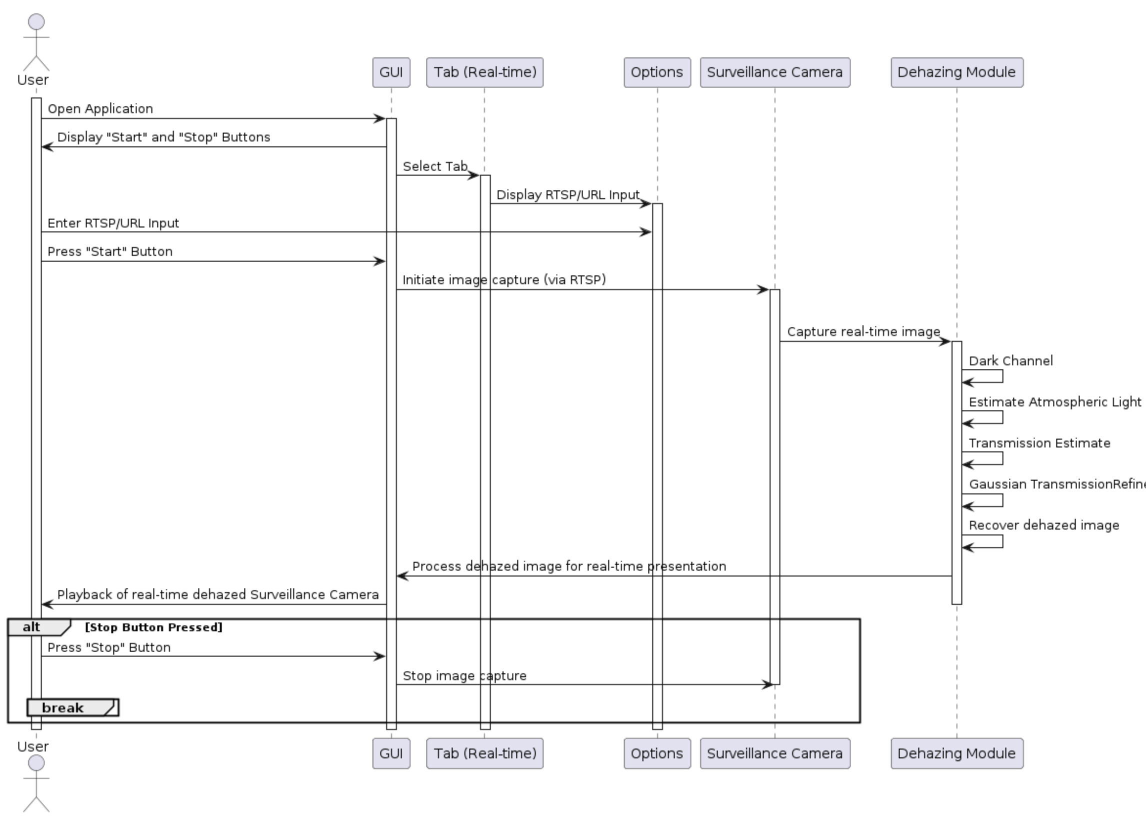


Figure 34 shows the process of a real-time surveillance dehazing module. The user opens the application and is greeted in the real-time tab with start and stop buttons.

After entering the RTSP/URL input in the options and clicking start, it initiates image capture via RTSP from the surveillance camera. The captured real-time images are then processed by the dehazing module, which performs tasks like estimating atmospheric light, transmission estimation, Gaussian transmission refinement, and recovering the dehaze image. The dehazed real-time surveillance camera feed is then played back to the user. If the stop button is pressed, the image capture is stopped.

Figure 35

Image Dehazing Module Sequence Diagram

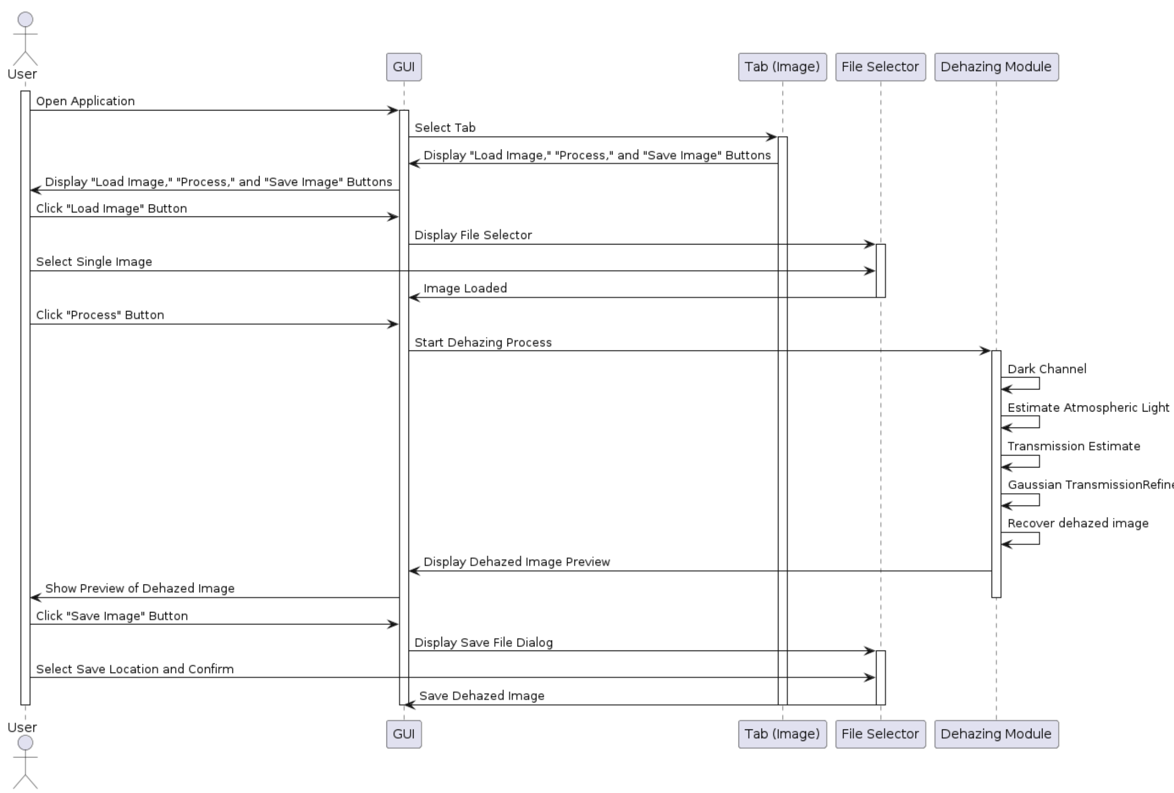


Figure 35 shows the process of a single image dehazing module. Once the user opens the application, it is greeted with load image, process, and save Image buttons on the Image tab. After clicking the load image, the user is asked to select a single image file from the file selector dialog. Once the image is loaded, the user clicks the process button to initiate the dehazing process. The dehazing module performs tasks like dark channel prior, estimating atmospheric light, transmission estimation, Gaussian transmission refinement, and recovering the dehaze image. The dehaze image preview is then displayed to the user. To save the dehaze image, the user can click the save image button and then the file dialog for saving the dehaze will show and it will be saved on that specified location.

Figure 36

Video Dehazing Module Sequence Diagram

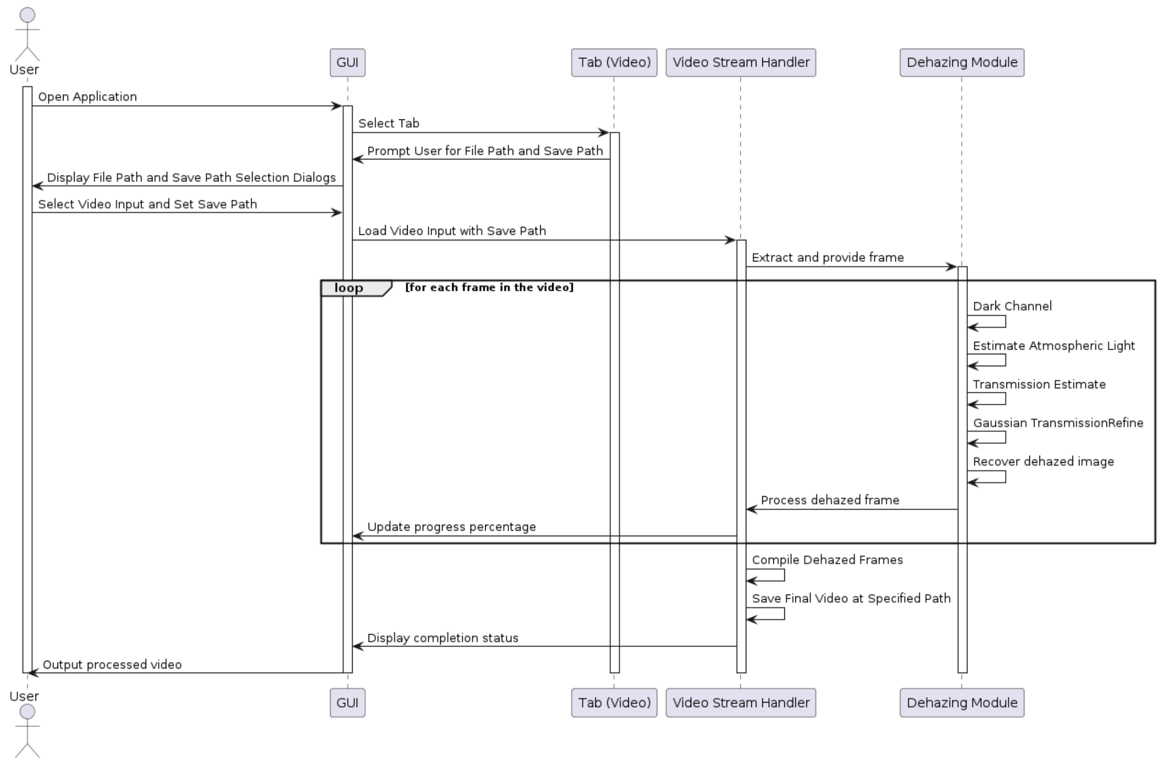


Figure 36 shows the process of a video dehazing module. The user opens the application and selects the Video tab. Is immediately prompted to select the video input file path and the save path for the processed video. Once the user provides these paths, the video input is loaded along with the save path. For each frame in the video, the Video Stream Handler extracts and feeds the frame to the Dehazing Module then it performs the tasks like dark channel prior, estimating atmospheric light, transmission estimation, Gaussian transmission refine, and recovering the dehaze image on each frame. The processed dehazed frames are then sent back to the Video Stream Handler and compiles all the dehazed frames, saves the final processed video at the specified save path, and displays the completion status to the user through the GUI. The user can then access the output processed video.

Figure 37

SeeThrough Mobile Real Time Camera Dehazing Sequence Diagram

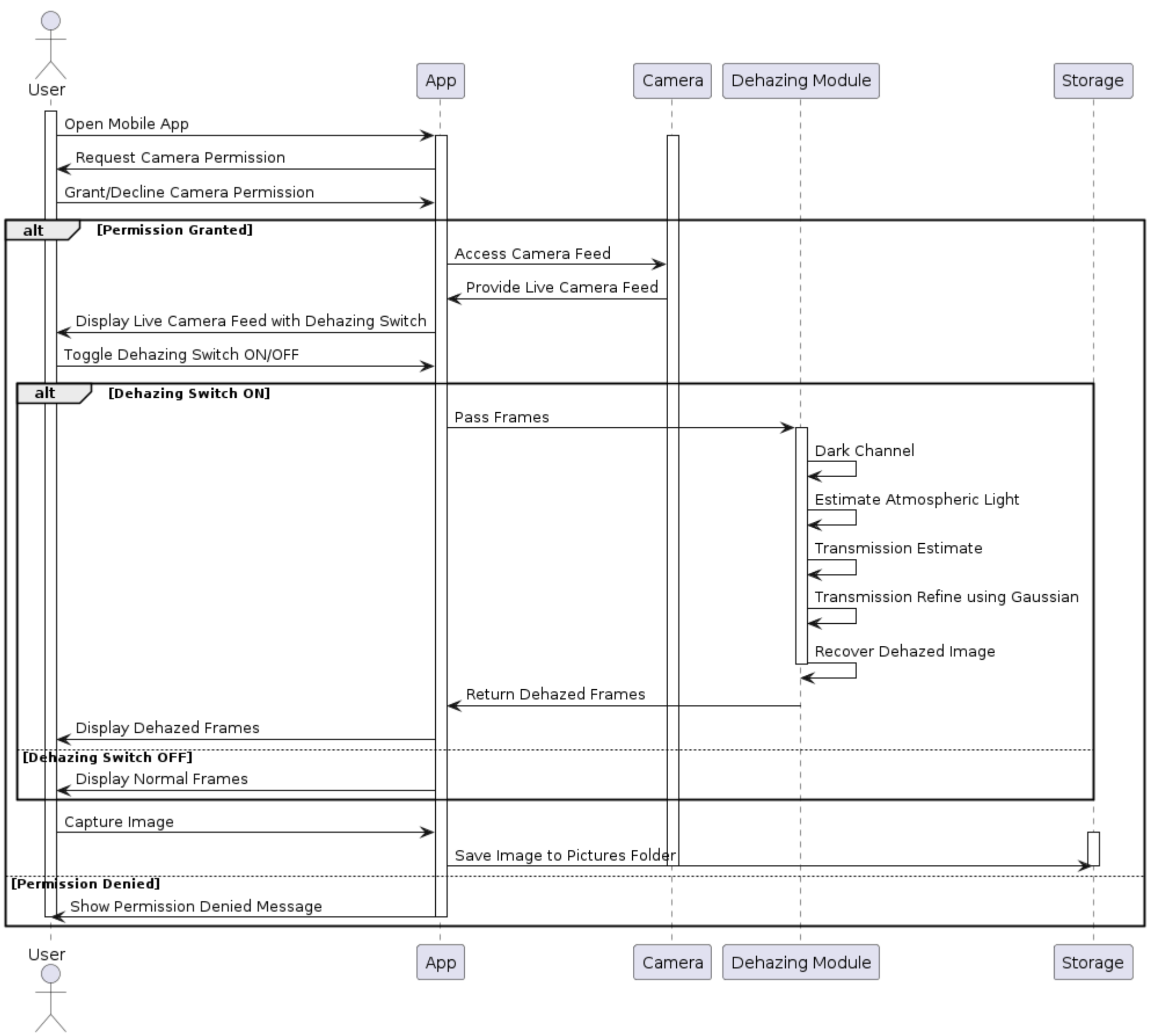


Figure 37 shows the mobile app process for real time camera dehazing. On first launch, it requests a camera permission. If granted, then it displays the live camera feed with a dehazing switch option. When the dehazing switch is on, frames are sent to the Dehazing Module for processing (dark channel, atmospheric light estimation, transmission estimation, Gaussian transmission refinement, image recovery). Then the processed frames are displayed. With dehazing off, normal frames are shown. The user can capture images and will save them to the Pictures folder. If camera permission is denied, a message is displayed.

Figure 38

SeeThrough Mobile Image Dehazing Sequence Diagram

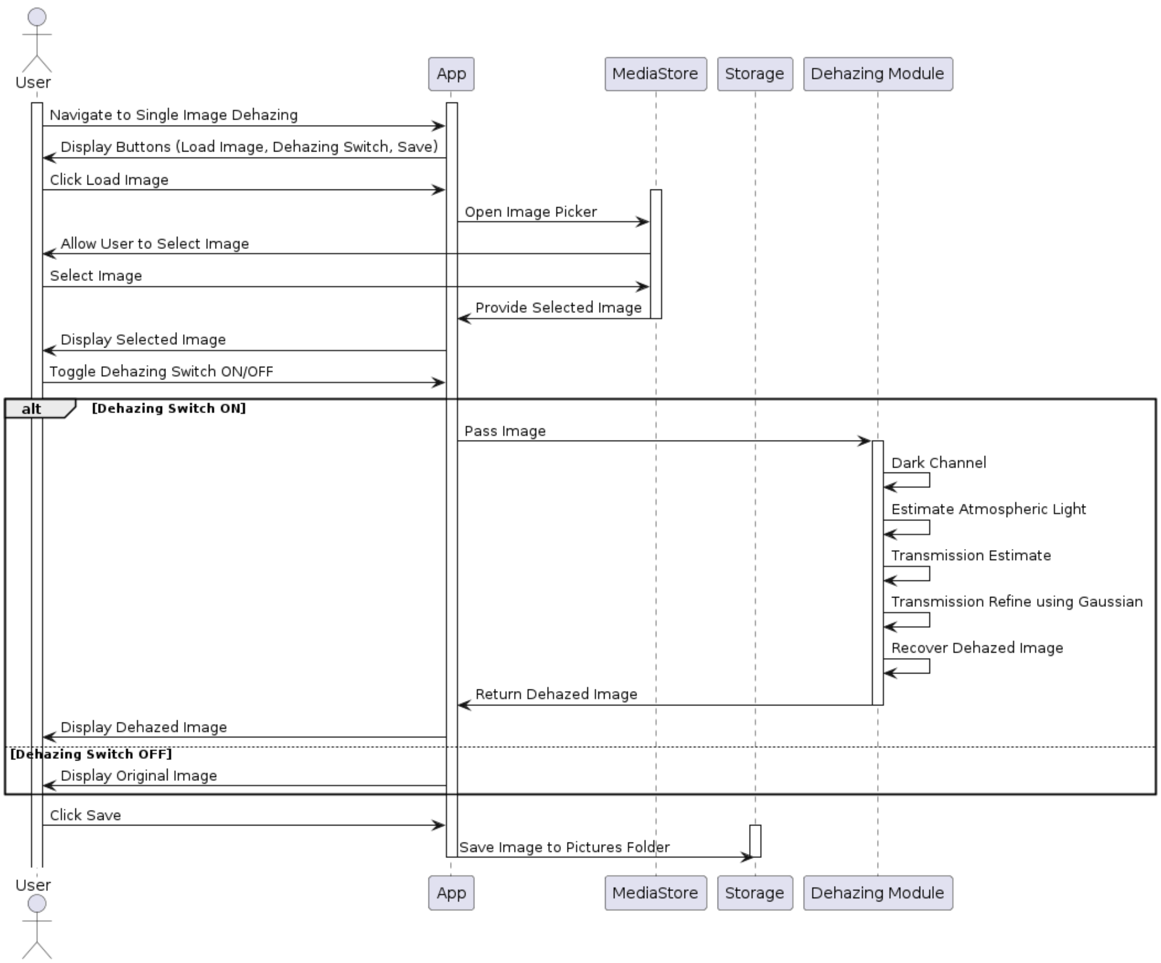


Figure 38 shows the mobile app process for single image dehazing. The user loads an image from the device's storage and is displayed to the user. The user can toggle dehazing on/off. If on, the app passes the image to the Dehazing Module for processing (dark channel, atmospheric light estimation, transmission estimation, Gaussian transmission refine, dehaze image recovery) and then dehaze image is displayed. If dehazing is off, the original image is shown. The user can save the displayed image to the device's Pictures folder.

Activity Diagram

Activity Diagram illustrates the various activities and actions involved in executing a process to understand the aspects of the system.

Figure 39

Activity Diagram of SeeThrough Desktop System

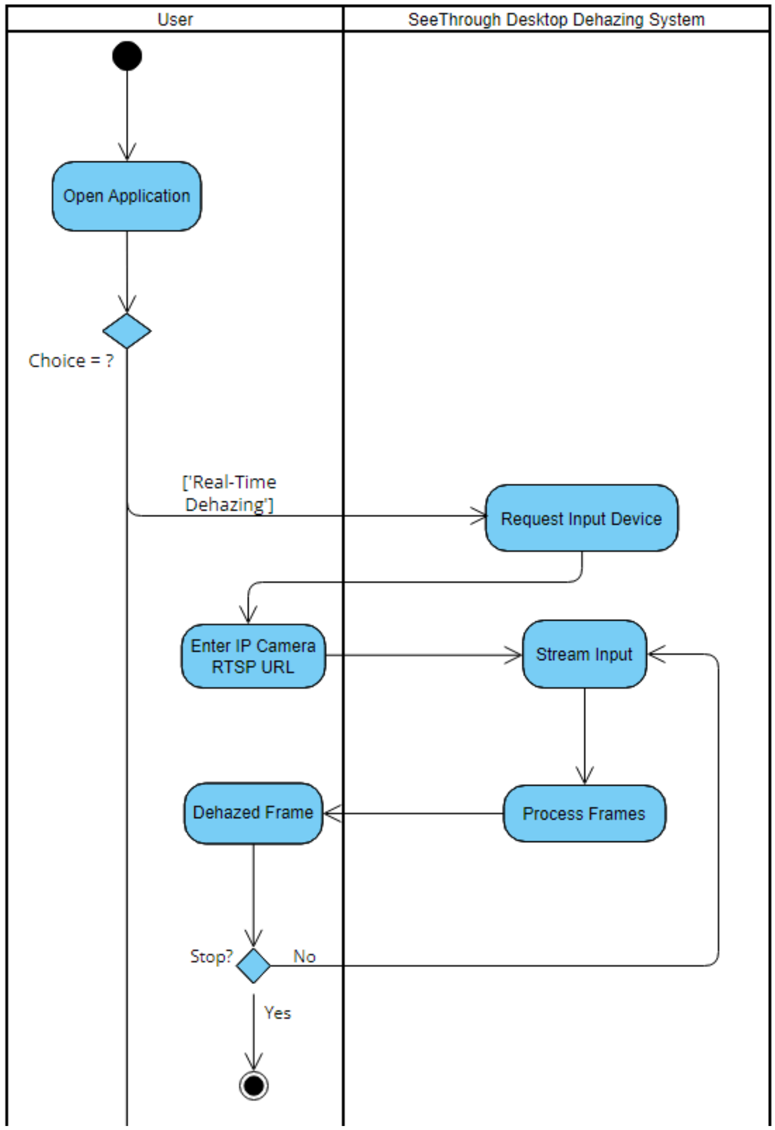


Figure 39 represents the activity diagram or the flow of the SeeThrough Desktop Dehazing System's process, from one activity to another. Upon application startup, the user has three options: Real-Time Dehazing, Video Dehazing, and Single Image Dehazing. If the user selects real-time dehazing, the system will require the user to first establish the desktop system's connection with their IP camera by entering their IP camera's RTSP URL in the camera settings. After establishing the connection, the system will continuously take input from the IP camera, process each frame into the dehazing algorithm, and output the dehazed frame until the user chooses to stop the real-time process.

Figure 40

Activity Diagram of SeeThrough Desktop System Cont.

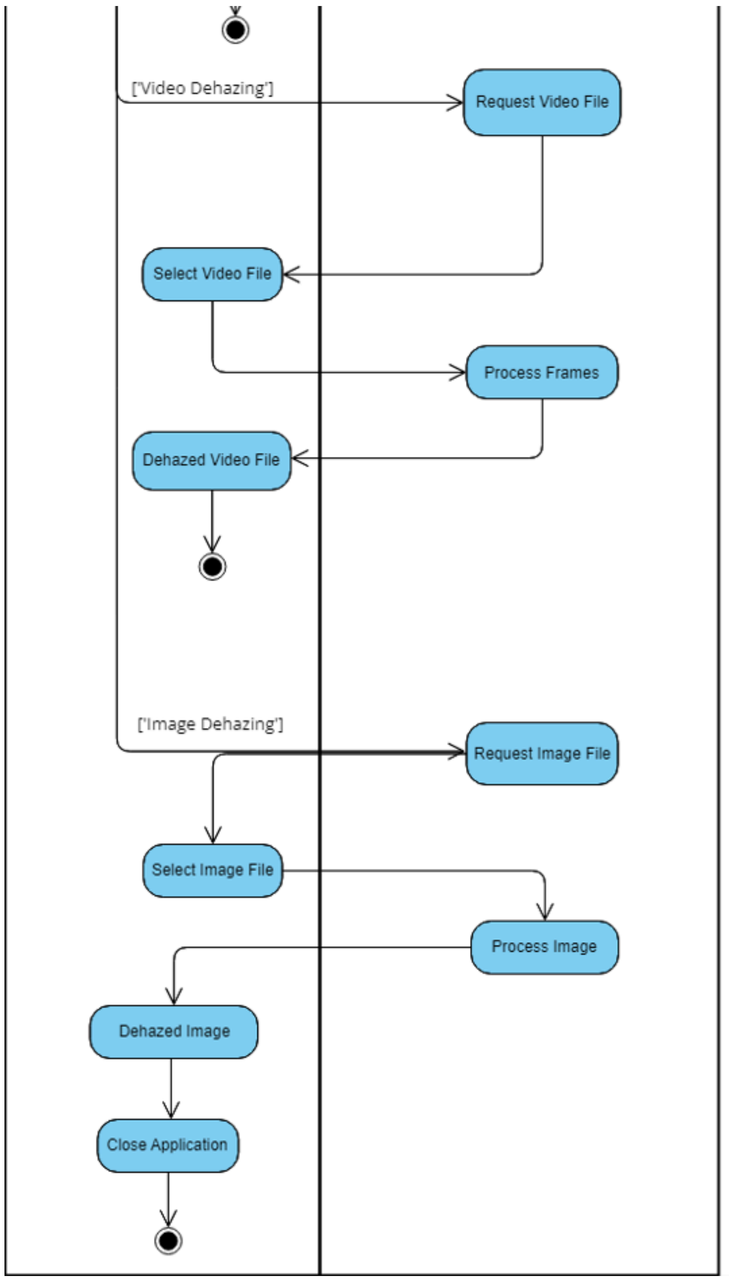


Figure 40 shows that if the user chooses video dehazing, the system will ask the user for an input video file. The system will process each frame in the video and create a new file which contains the input's dehazed version. Lastly, if the user chooses image dehazing, the system will ask the user for an input image file. The system will run the image into the dehazing algorithm and output a dehazed version of the image onto the system window. The system will terminate if the user closes the application at any point of activity.

Figure 41

Activity Diagram of SeeThrough Mobile Application.

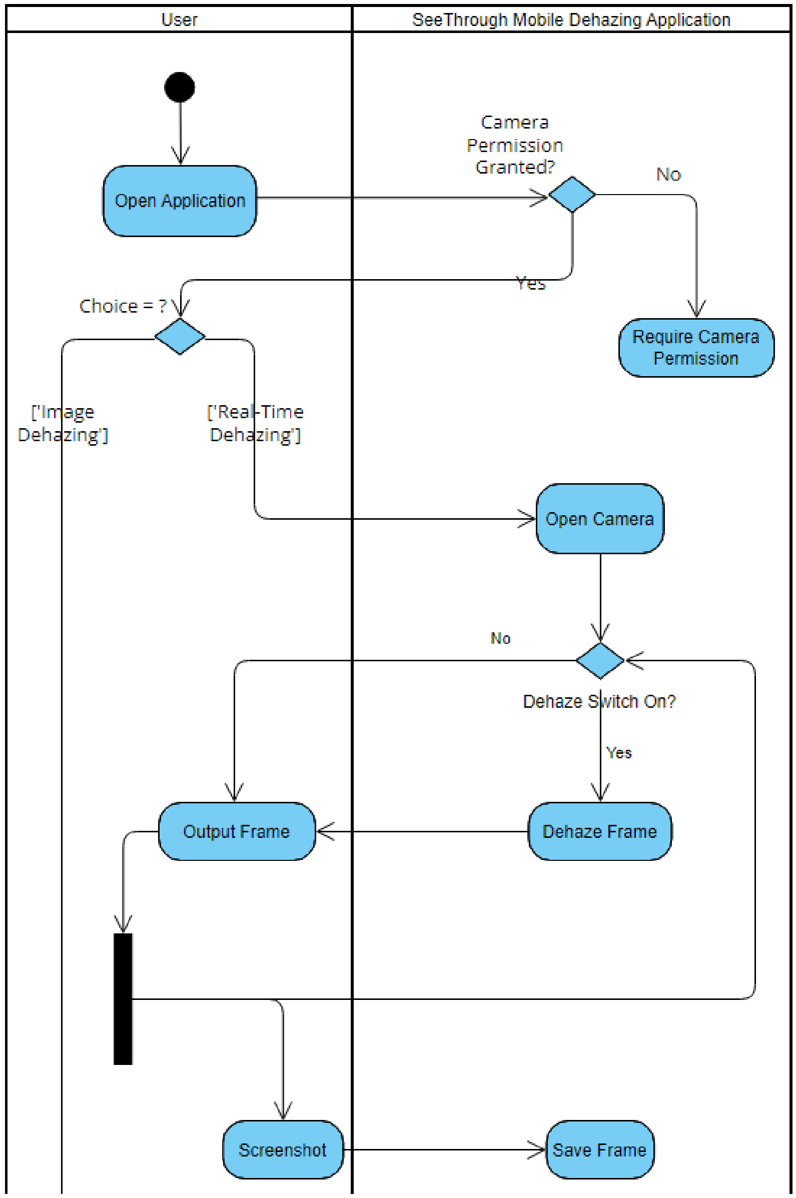


Figure 41 represents the activity diagram or the flow of the SeeThrough Mobile Dehazing Application's process, from one activity to another. Upon application startup, the application will request the user's permission to use the android phone's camera. Without the user's permission, the real-time dehazing feature will not be accessible. After permitting the application to use the camera, the user has two options to choose from: real-time dehazing and image dehazing. If the user selects real-time dehazing, the application will open the phone's back camera and start taking input frames. If the dehazing toggle switch is turned off, the system will only output the non-processed frames. If the dehazing toggle switch is turned on, the input frame will be processed through the dehazing function and the dehazed frame will be outputted. Throughout the process, the user has the option to take a screenshot within the application. If the user presses the screenshot button, the system will save the current frame being shown into the screenshot directory of the phone.

Figure 42

Activity Diagram of SeeThrough Mobile Application Cont.

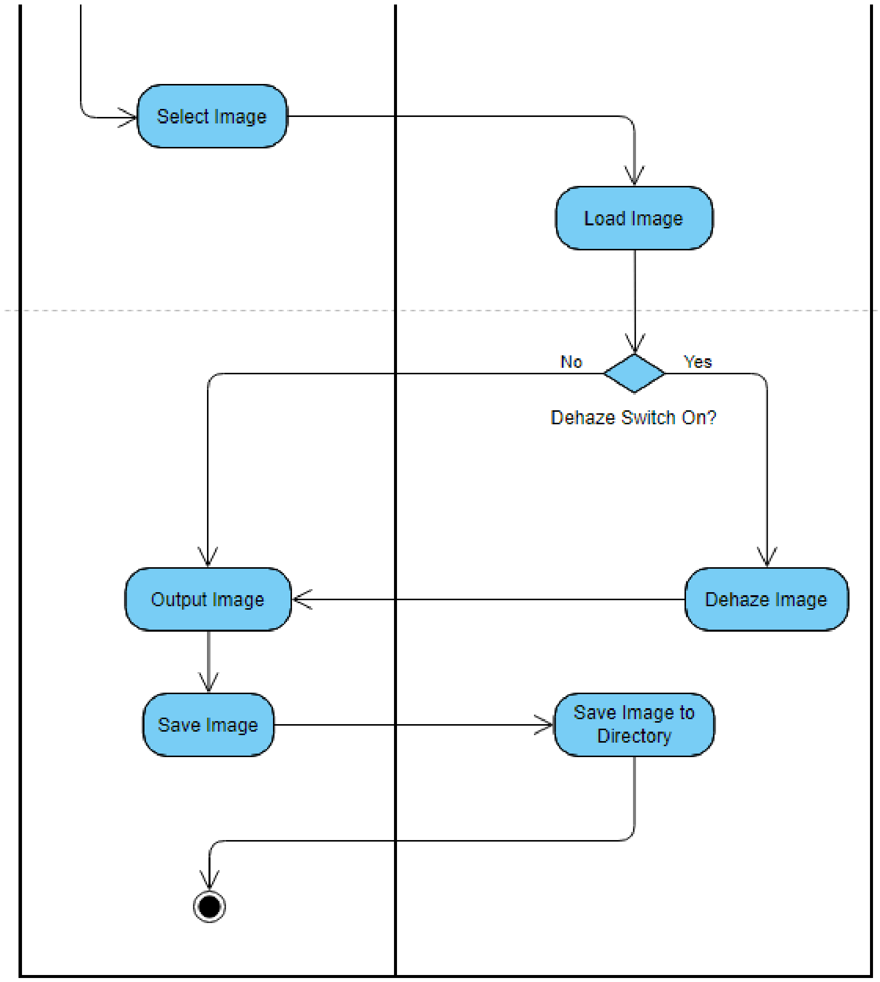


Figure 42 shows that if the user selects the image dehazing feature instead, the user will be able to select an image from their gallery. After loading the image into the application, if the dehazing toggle switch is turned off, no processing will be done on the image. If the toggle switch is turned on, the image will undergo the dehazing process and the dehazed image will be displayed instead. The user can also choose to save the image.

Project Testing and Evaluation

In this section, the researchers discuss the actions taken to ensure the quality of the project system and evaluate the system through known standards.

Functional Testing

Functional testing was performed to determine how well some aspects of the system perform under a workload. It serves different purposes like demonstrating if the system meets a standard criterion.

The researchers conducted this testing by inviting subject matter experts and IT experts to evaluate the dehazing application. The experts were provided an installer, a video demonstration of how to install and use the application, and a list of unit test cases to be done on the application to capture all use case scenarios. During the testing process, the experts were asked to observe the application's behavior and to write any remarks on the provided test case sheet.

Compatibility Testing

Compatibility testing was performed to determine if the system is compatible with different device environments.

As with functional testing, the compatibility testing was also done by the same subject matter and IT experts that were invited for the functional testing. The experts were provided an installer, a video demonstration of how to install and use the application, and a list of compatibility test cases to capture different system environments that the application should be compatible with. During the testing process, the experts were asked to observe the application's behavior and to write any remarks on the provided test case sheet.

Evaluations

The evaluations were used to determine if the system meets the standard to be used as an application. The evaluation process was done by inviting end-users and IT experts to rate the dehazing applications' acceptability and to gather feedback for future improvements. Depending on the respondent's availability, the survey was conducted either in-person or online. For in-person surveys, the researchers demonstrated the application to the respondent, and afterwards, the respondent was provided a physical survey questionnaire for them to rate and make necessary comments on the application. For online surveys, the respondents were provided a digital form made with Microsoft forms which includes a video demonstration of the desktop or mobile application.

There were two standards used to determine the acceptability of the system: ISO/IEC 25010:2023 for the desktop and Android Core App Quality for the mobile application. The standards used to evaluate for this project were:

ISO/IEC 25010:2023

- Functional Suitability: capability of a product to provide functions that meet stated and implied needs of intended users when it is used under specified conditions.

- Performance Efficiency: capability of a product to perform its functions within specified time and throughput parameters and be efficient in the use of resources under specified conditions.
- Compatibility: capability of a product to exchange information with other products, and/or to perform its required functions while sharing the same common environment and resources.
- Interaction Capability: capability of a product to be interacted with by specified users to exchange information between a user and a system via the user interface to complete the intended task.
- Reliability: capability of a product to perform specified functions under specified conditions for a specified period of time without interruptions and failures
- Maintainability: capability of a product to be modified by the intended maintainers with effectiveness and efficiency.
- Flexibility: capability of a product to be adapted to changes in its requirements, contexts of use, or system environment.

Android Core App Quality

- **Visual Experience:** The app should provide standard Android visual design and interaction patterns where appropriate, for a consistent and intuitive user experience.
    - Navigation
    - UI and Graphics

- Visual Quality
- Accessibility

- **Performance and Stability:** The app should provide the performance, stability, compatibility, and responsiveness expected by users.
  - Stability
  - Performance
- **Privacy and Security:** The app should handle user data and personal information safely, with the appropriate level of permission.
  - Permissions
  - Data & Files

Research Methodology

This section discusses steps and procedures to gather, analyze, and interpret data for the project.

Research Method Used

The study titled “Real-Time Surveillance Dehazing using Dark Channel Prior Algorithm with Gaussian Filtering Method” employs a quantitative research method.

A quantitative method was applied to the study as the numbers will determine the effectiveness of a real-time dehazing system using quantitative evaluations on image processing such as PSNR (Peak Signal to Noise Ratio) and SSIM (Structural Similarity Index Measure), and the combination of Likert Scale in software evaluation adhering to the standard of ISO/IEC 25010:2023 and Android Core App Quality criteria.

By using the quantitative method, the results of the study provided numerical data that was not biased by subjective opinions. This could evaluate the properties of the dehazing results which could be helpful for future references.

Research Instrument

The data was collected through a survey questionnaire with the use of Microsoft Forms to determine the effectiveness of the dehazing system. The participants were asked to participate in the software evaluation of the desktop system, mobile application, and the dehazing quality. The survey questionnaire contains five sections. Section 1 contains an introduction, informing the participants about the purpose of the evaluation form. Section 2 contains the data privacy policy, acquiring the respondent’s consent. Section 3 contains personal information fields that are relevant to the evaluation. Section 4 is the main part of the survey questionnaire which contains a Likert scale question and answer portion based on the ISO/IEC 25010:2023 software standard for the desktop system and the Android Core App Quality criteria for the mobile application. Lastly, section 5 contains a suggestion and feedback field for respondents to add remarks on the system for further improvements or consideration.

Table 4

Verbal Interpretation of the Mean Interval for the Acceptability of SeeThrough Dehazing System

Point	Weighted Mean	Verbal Interpretation
5	4.20 - 5.00	Highly Acceptable
4	3.40 - 4.19	Acceptable
3	2.60 - 3.39	Moderately Acceptable
2	1.80 - 2.59	Fairly Acceptable
1	1.00 - 1.79	Poorly Acceptable

Table 4 shows the scoring criteria for the Likert scale survey questionnaire, where 5 is the highest and 1 is the lowest, and its corresponding weighted mean and verbal interpretation. A grand mean ranging from “1.00” to “1.79” is verbally interpreted as “Poorly Acceptable”, which means the application might have failed to execute its main functions; a grand mean from “1.80” to “2.59” is interpreted as “Fairly Acceptable” where, somehow, the application have executed but lack of consistency to function well; “2.60” to “3.39” is “Moderately Acceptable”, which means that the features of the application almost meet its basic requirements but could be further improved; “3.40” to

"4.19" is "Acceptable", which means that the application meets expectations and enhances the quality of the application; and "4.20" to "5.00" is "Highly Acceptable" where the overall functions and objectives of the application exceeds expectations and works smoothly.

Population and Sampling

For the ISO/IEC 25010:2023 and Android Core App Quality survey, a total of 48 participants were needed to gather enough data for this research. The 48 participants consisted of 36 end-users and 12 IT experts. For the functional and compatibility testing, a total of 15 participants were needed which consisted of subject matter experts and IT experts only. The inclusion criteria for the end-users are individuals who are barangay kagawads or officials, a resident, or a frequent tourist of a fog-prone area such as Tagaytay City. The inclusion criteria for subject matter experts are individuals who are responsible for their area's surveillance system such as surveillance technicians or barangay kagawads and officials. The inclusion criteria for IT experts are individuals who are currently working in the IT industry. A convenience sampling technique was used to select the participants for this study. This method allows for the recruitment of participants who meet the inclusion criteria and are readily available and willing to participate in the study.

Participants of the Study

The participants of this study consists of locals, frequent tourists, surveillance technicians, and barangay kagawads or officials of Tagaytay. These individuals are familiar with the challenges posed by the reduced visibility due to haze and can provide valuable insights into the effectiveness of the real-time video dehazing system.

Validation of Instrument

To make sure that the data to be collected is accurate and reliable, a pilot test was conducted to validate the research instrument. The questionnaire was handed out to a small group of individuals who are not part of the actual study.

Based on the feedback that the pilot test would acquire, the questionnaire was revised to ensure that the questions and writing method instructions are understandable and clear. These steps were taken to increase the validity and reliability of the research instruments.

Data Gathering Procedure

The researchers of this study relied on online surveys with the selected respondents. The online questionnaire was sent to the respondents through their email address or through their preferred platform.

The questions for the survey were developed based on the research objectives and the results of the initial pilot test to ensure accuracy. The researchers provided the participants a clear guideline on how to accomplish the survey. In accordance with the data privacy act, the collected data from the respondents were securely stored and analyzed using statistical tools to identify patterns and trends The respondents' privacy and confidentiality was strictly maintained throughout the data gathering and analysis process.

Statistical Treatment of Data

The data was gathered from a total of forty-eight (48) evaluation respondents. The collected data was computed and underwent data interpretation and validation based on the weighted mean and standard deviation. The following are the formulas used by the researchers.

Weighted mean. It is a kind of average, rather than every datum point contributing similarly to the final mean.

Equation 8

Weighted Mean Formula

$$\bar{x} = \frac{\sum x}{n}$$

where:

$\bar{x}$ = mean

$\sum$ = "summation of"

x = Score of proper weight

n = Total number of respondents

$\sqrt{n}$ = Square Root of Sample Size

The mean is the arithmetic average of the scores that were given by the test respondents. X is the summation of the total scores and N is the total number of respondents. In this study, the mean is used to determine the acceptability rating of the applications based on the gathered responses.

Standard deviation. This is the proportion of dispersion of a set of data from its mean. It quantifies the absolute variability of distribution; the higher the dispersion or variability, the more prominent is the standard deviation and more prominent the extent of the deviation of the incentive from their mean.

First, input the raw data in the excel table prepared by the researchers to easily separate each criterion from one another. After that, excel has the function to extract SD from the data gathered from the evaluators.

Likert scale. The researchers used the Likert scale in measuring the respondents' acceptability level from the questionnaires guided by the criteria presented in the recommended evaluation instrument.

Data Analysis

In this study, the collected data using the Likert-scale questionnaire was analyzed using the appropriate statistical methods. After gathering the results, the treated data was interpreted based on the software acceptability metric shown in table 4.

Ethical Consideration

Ensuring the participants' confidentiality and anonymity is a crucial ethical consideration for this study. The researchers obtained informed consent from the participants, explaining the purpose of the study and how the data is collected and used in the researchers' study. The participants' personal information, including their name and demographics was kept confidential and were only accessible to the researchers. Any personally identifying information was removed or anonymized before the data was analyzed or published. The researchers also ensure that the participants were not subjected to any harm or discomfort during the study, and they were given the option to withdraw at any point in time from the study without penalty or delay.

RESULTS AND DISCUSSION

The results of the study are shown in this section and are discussed in detail. It includes the objectives of the research that were met, the performance evaluation of the algorithm based on image quality metrics, test results of the functional and compatibility test cases, and the presentation and interpretation of the data gathered.

SOP #1: How was the dehazing algorithm be optimized to process footage in real-time while maintaining high image quality?

Initially, SeeThrough used a single thread to manage its dehazing process. This thread captured footage from the surveillance camera, processed the input stream for dehazing, and displayed the cleared frame instantly. However, during testing, this single-threaded approach averaged only about 3 to 6 frames per second. Problems like frame skipping & delays of up to 10 seconds were common.

To fix these problems and improve the dehazing algorithm for quick processing without sacrificing image quality, several strategies were adopted:

1. **Efficient Computational Techniques:**
    - A multithreading method was implemented. This spreads the dehazing tasks across all available CPU threads. The main thread collects input from the surveillance camera & creates a pool of images. Meanwhile, other threads work on dehazing these images from this pool. As a result, SeeThrough's real-time processing speed rose to an impressive average of 30 frames per second.
2. **Adaptive Filtering & Enhancement:**
    - **Dark Channel Prior:** This method estimates haze thickness to enhance clarity in images. It works well to reduce haze and boost both contrast & color accuracy.
    - **Gaussian Filtering:** Adaptive Gaussian filtering adjusts dehazing parameters according to current lighting conditions and scenes. This ensures the algorithm operates effectively across various footage types. Additionally, noise reduction techniques help remove artifacts while keeping important details intact.
3. **Hardware Acceleration:**
    - Using optimized software libraries like OpenCV improves processing speed significantly. Also, harnessing specialized hardware such as GPUs can speed up computations, ensuring performance remains real-time.

4. **Hybrid Approach:**
    - A hybrid dehazing method combines various strengths of traditional approaches—like Dark Channel Prior with Gaussian filtering. This allows quick handling of simpler cases using traditional methods while still ensuring top image quality through advanced filtering for complex scenarios.

5. **Real-World Testing & Feedback:**
    - Conducting thorough real-world testing helps gather valuable performance data & user feedback. This ongoing process aids in refining and optimizing the algorithm, ensuring it performs well under different conditions.

By applying these strategies, the dehazing algorithm achieves optimization for real-time footage processing without compromising on image quality. The integration of multithreading, adaptive filtering with Dark Channel Prior & Gaussian methods, hardware acceleration, plus a hybrid approach makes this algorithm suitable for research & everyday practical use by end-users. The multithreading initiative initiated by SeeThrough marks a significant step toward better performance in real-time processing & enhanced speed overall.

SOP #2: What specific development steps were followed to create the desktop application using Microsoft Visual Studio Code with Python and the mobile application using Android Studio with Java?

A desktop and mobile application were developed by the researchers and are available in the official SeeThrough website. In developing the desktop and mobile applications, a Rapid Application Development (RAD) software development lifecycle model was used, allowing the researchers to design a prototype and gather immediate feedback which helped improve SeeThrough to be user centric.

The next figures show the developed desktop and mobile application along with the discussion of its capabilities and limitations.

Figure 43

SeeThrough Desktop Version: Realtime Dehazing Tab

Figure 43 shows the desktop version of SeeThrough. Upon opening the application, the user is met with the realtime dehazing interface tab. In this tab, the user can interact with three buttons, namely “Start”, “Settings”, and “Screenshot”. The settings button is the first button that the user should interact with as this is where they can establish the camera connection through its RTSP URL. After establishing a connection, the user can click on the start button to begin dehazing in real time. While the dehazing process is happening, the user can press the screenshot button to capture an image frame from the footage. The screenshots are located in the same folder where SeeThrough was installed.

Figure 44

SeeThrough Desktop Version: Realtime Dehazing Tab

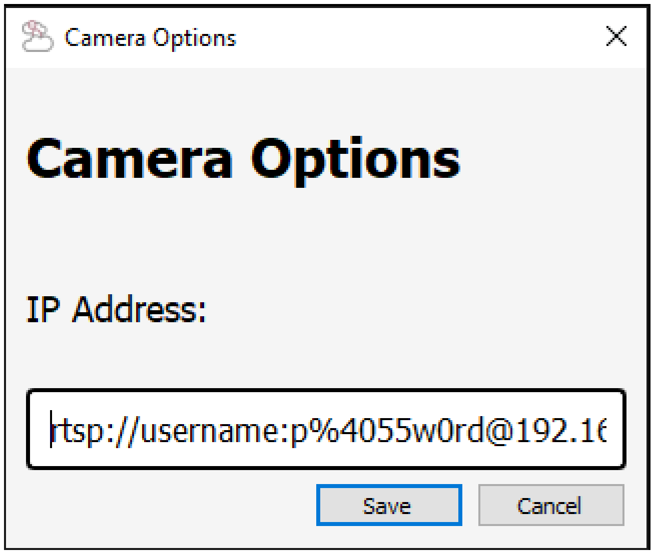


Figure 44 shows the camera options popup when the user selects the settings button in the realtime dehazing tab. In this popup, the user can enter the RTSP URL of their surveillance camera. After saving, the URL is saved into the configuration file of SeeThrough.

Figure 45

SeeThrough Desktop Version: Realtime Dehazing Tab

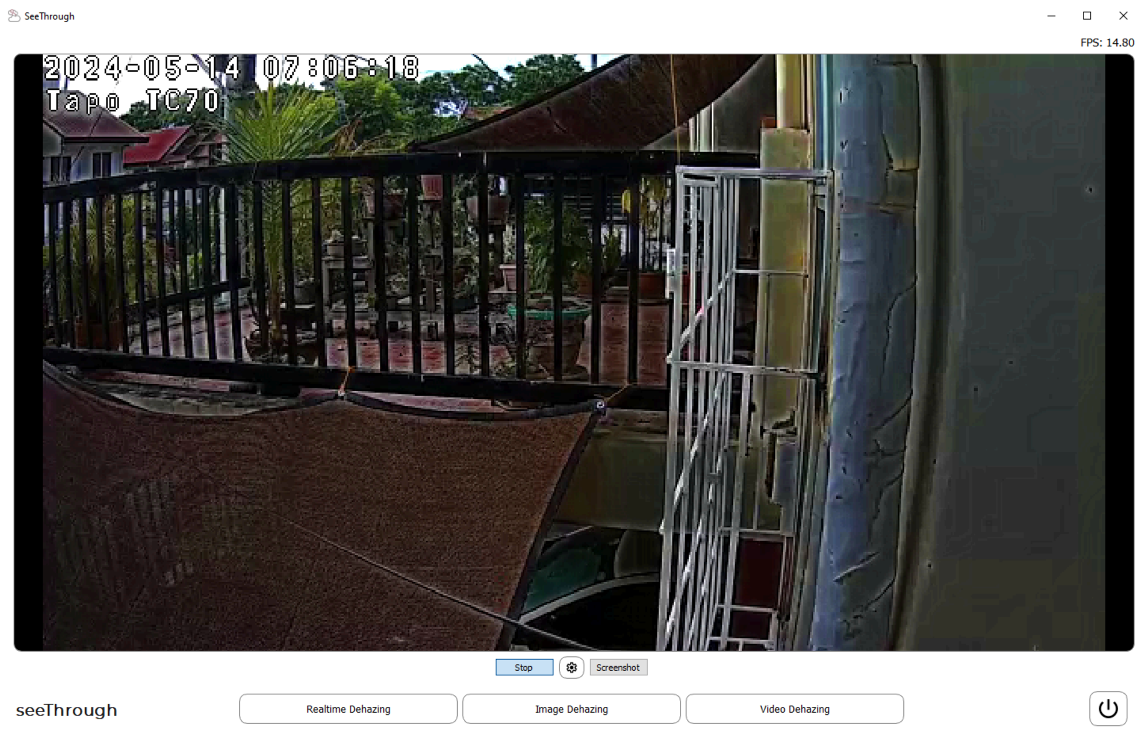


Figure 45 shows the real time dehazing feature of SeeThrough's Desktop Version. The system acquires live feed input from the surveillance camera and displays the dehazed version of what the camera sees.

Figure 46

SeeThrough Desktop Version: Image Dehazing Tab

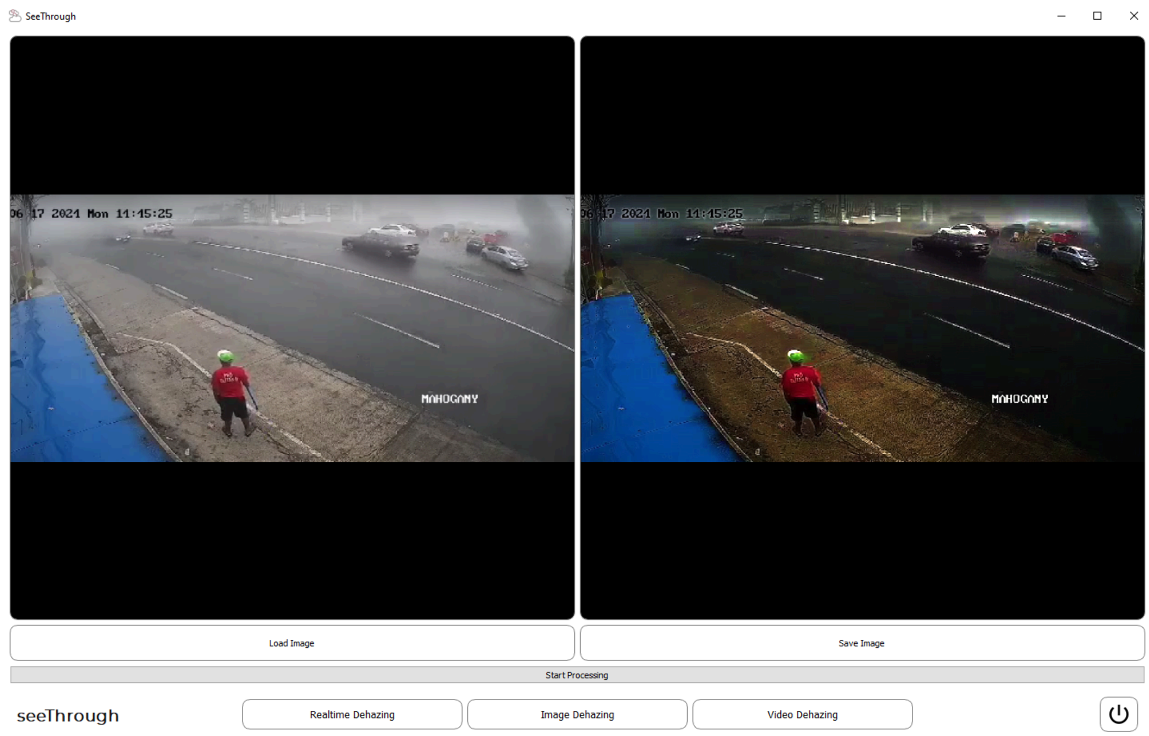


Figure 46 shows the image dehazing feature of SeeThrough's Desktop Version. In the image dehazing tab, the user is met with three buttons, namely "Load Image", "Start Processing", and "Save Image". The load image button opens a file dialog that allows the user to select an image file. After choosing an image, the system displays the raw input on the left-hand side of the screen as shown in the figure above. After loading an image, the user can click the start processing button, after which a dehazed version of the image is shown on the right-hand side of the screen as shown in the figure above. Lastly, the user has the option to save the dehazed image by clicking the save image button directly below the dehazed output.

Figure 47

SeeThrough Desktop Version: Video Dehazing

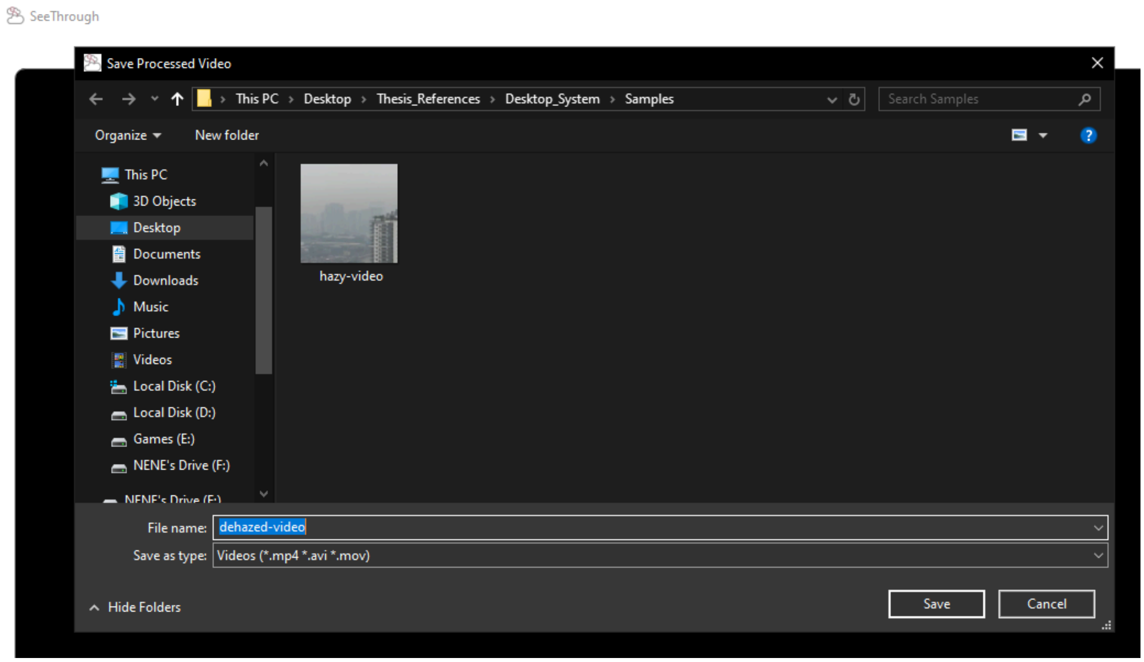


Figure 47 shows the video dehazing feature of SeeThrough's desktop version. Upon clicking the "Video Dehazing" button, a file dialog pops up. The user is requested to select the hazy video as input. After selecting the hazy video, a second file dialog pops up which requests the user to select the save directory and name the output file.

Figure 48

SeeThrough Mobile Version: Camera Permission

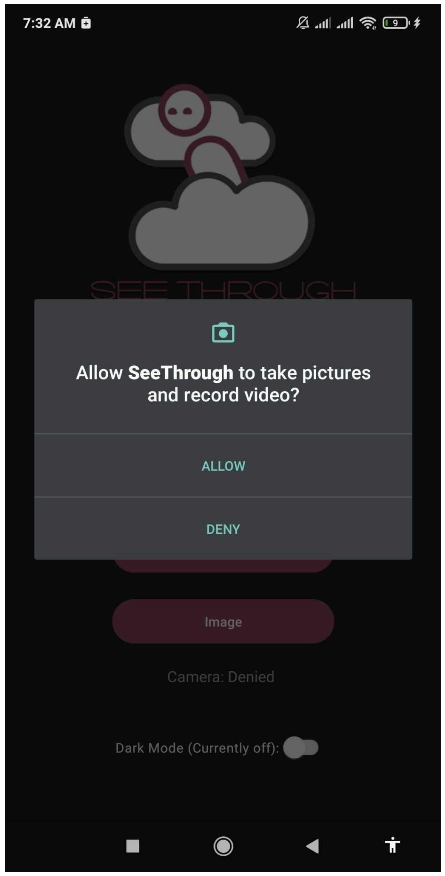


Figure 48 shows the mobile version of SeeThrough. Upon first time usage of the app, SeeThrough requests the user for permission to use the device's camera. If permission is allowed, the user can use the real time dehazing feature of SeeThrough mobile. Otherwise, the user will not be able to use the real time dehazing feature of SeeThrough Mobile.

Figure 49

SeeThrough Mobile Version: First Look

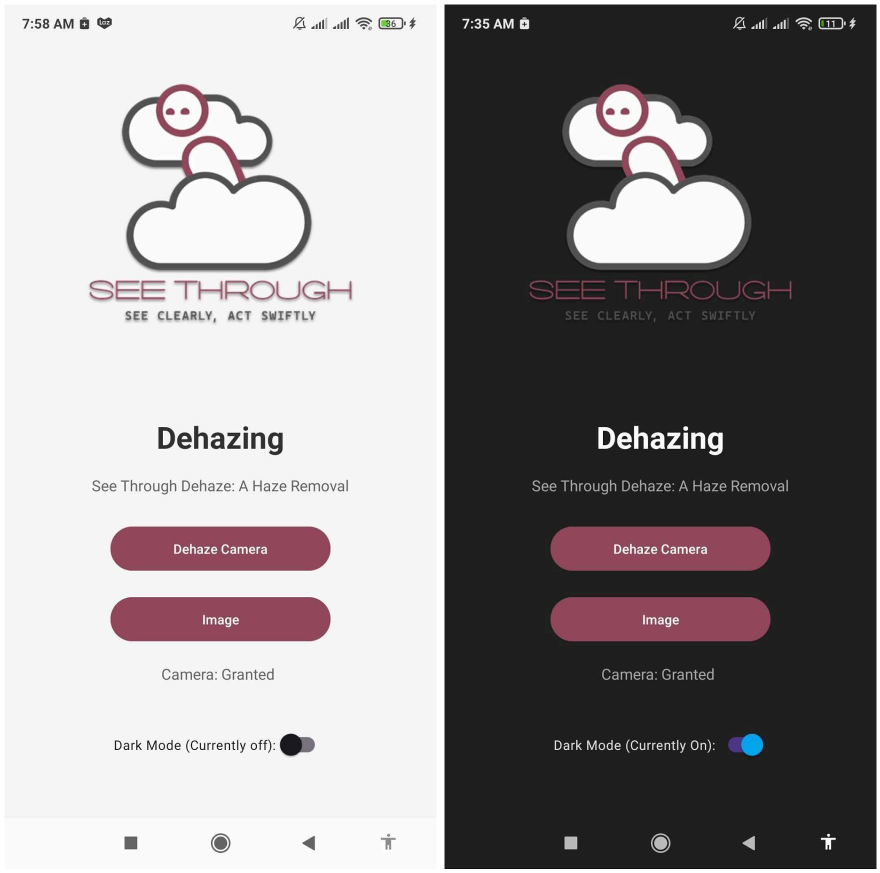


Figure 49 shows the interface that greets the user every time they open the application if camera permission is granted. In this interface, the user can interact with three buttons, namely “Dehaze Camera”, “Image”, and “Dark Mode” toggle switch. The dehaze camera opens the device’s back camera to start taking live video input, the image button directs the user to the image dehazing tab, and the dark mode toggle switch enables the user to reverse the color scheme of the application from light to dark or vice versa.

Figure 50

SeeThrough Mobile Version: Realtime Dehazing

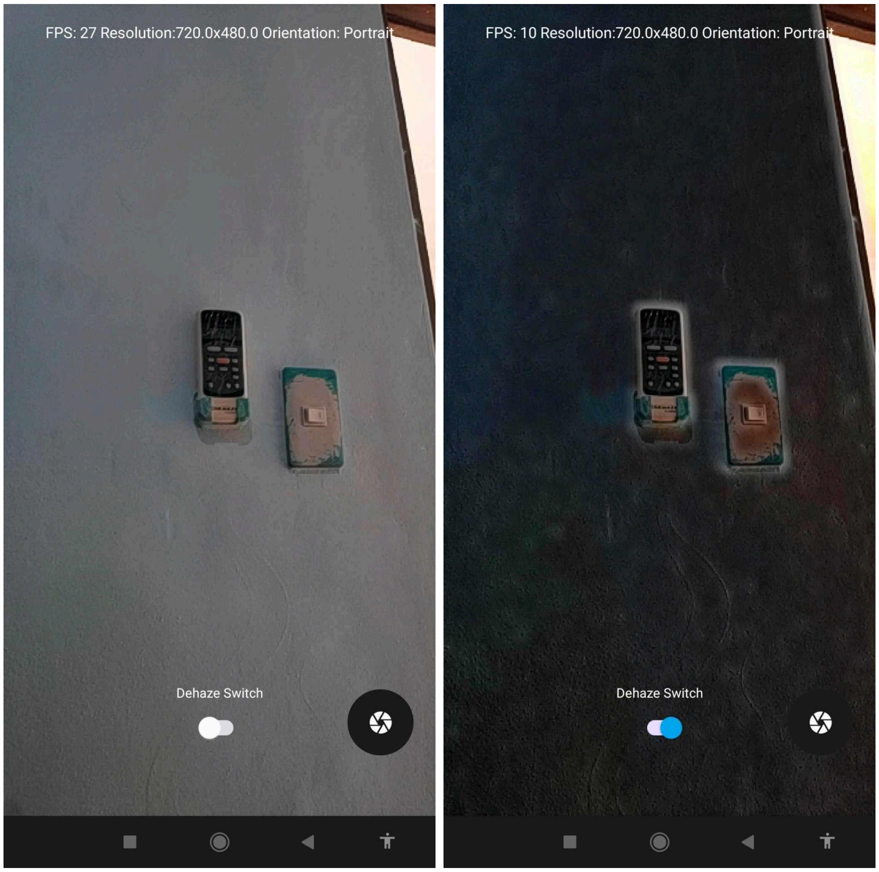


Figure 50 shows the real time dehazing feature of SeeThrough's mobile version. In this interface, the user can interact with the dehaze toggle switch and the screenshot button on the lower right part of the screen. If the dehazing switch is toggled off, the application only displays a raw video feed. If the dehazing switch is toggled on, the application displays the dehazed version of what the back camera sees. As shown in the figure above, no haze is currently present, hence the unnatural look of the sample. In addition, an fps counter, resolution, and camera orientation is displayed at the top of the screen. The orientation can be set to either portrait mode or landscape mode simply by tilting the device into the desired orientation.

Figure 51

SeeThrough Mobile Version: Image Dehazing

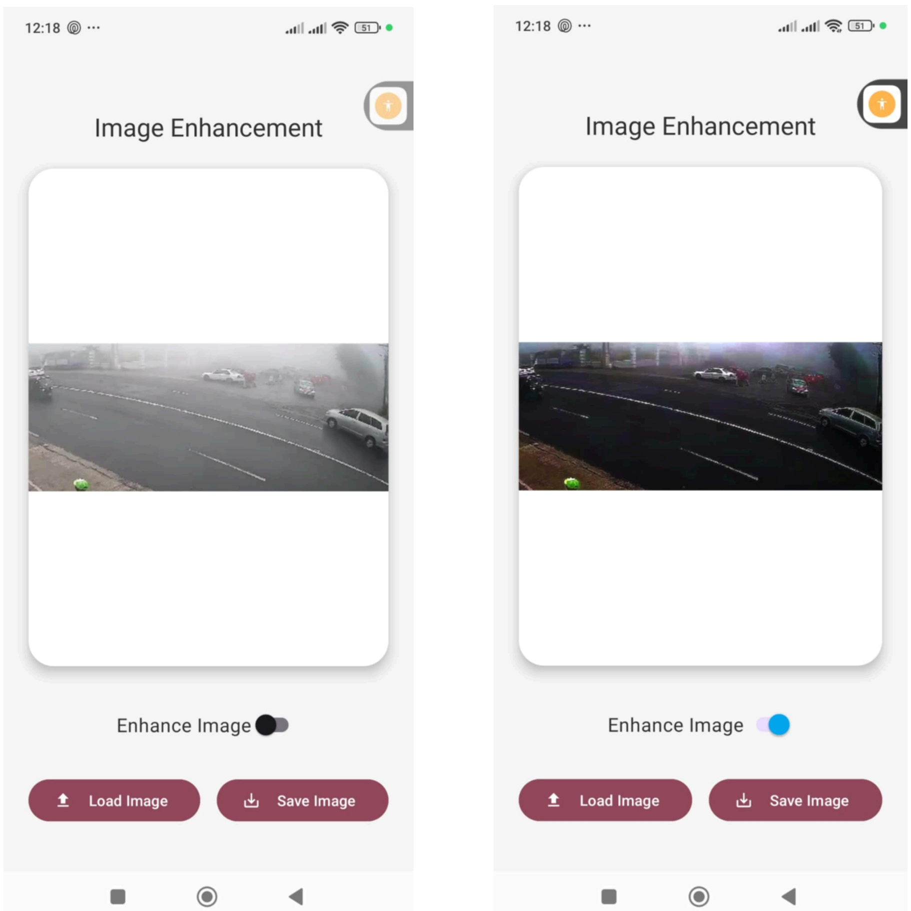


Figure 51 shows the image dehazing feature of SeeThrough's mobile version. In this interface, the user can interact with three buttons, namely "Load,", "Save", and "Dehaze " toggle switch. By pressing the load button, the application requests the user to select an image from their gallery. The selected image is displayed on the screen, as shown on the left in figure 51. By toggling the dehaze switch to on, the application displays the dehazed version of the selected image. Additionally, the user can revert the dehazed image back into its raw/unprocessed version by simply toggling the dehaze switch off. Lastly, the user has the option to save the dehazed image.

Project Capabilities and Limitations

The following are the capabilities of the project:

1. The desktop system is capable of dehazing images, videos, or live footage in real-time.
2. Establishing a connection with the surveillance camera is possible through the camera's RTSP URL.
3. The desktop system allows the user to save the dehazed image or video to the user specified directory.
4. The mobile application is capable of dehazing images or in real time.
5. The mobile application enables the user to toggle the dehazing feature on or off.
6. The mobile system is capable of respecting the user's decision on camera permission.

The following are the limitations of the project:

1. The surveillance camera can only be connected to the desktop system through its RTSP URL.

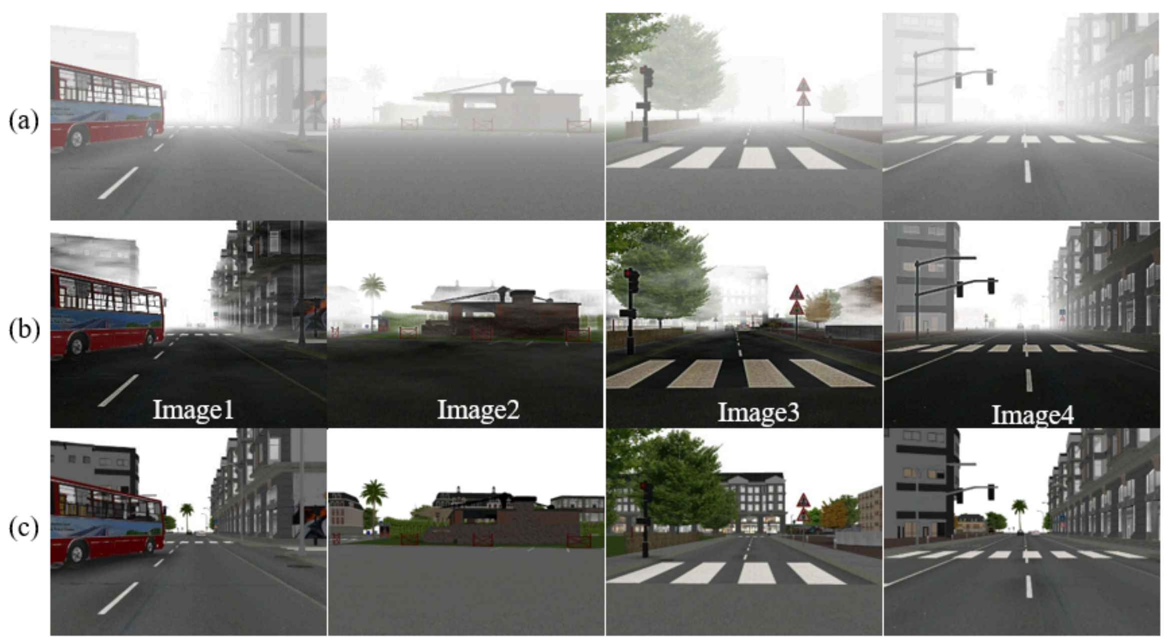
2. The desktop system can only connect to and dehaze one surveillance live footage at a time.
3. The desktop system does not support a live recording feature. Due to the computational complexity of the algorithm alone, making a recording function to run simultaneously with the dehazing process results in a bigger load on the CPU, significantly reducing the frames processed per second.
4. The mobile application does not support a live recording feature. Not only is this due to hardware limitations, but also due to the limitations of the Android OpenCV library being unmaintained by its developers.

SOP #3: How was the system's dehazing performance be measured in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) using the Foggy Road Image Dataset (FRIDA)?

To evaluate SeeThrough's dehazing performance in terms of the PSNR and SSIM metrics, a pristine/non-hazy image is required as a reference image. Tarel et al. (2012) created FRIDA with four (4) fog categories containing sixty-six (66) hazy images each, along with its corresponding ground truth or the pristine image as reference. Through FRIDA, the PSNR and SSIM calculation was made possible.

Figure 52

Visual Comparison of the Dehazed Output against the Ground Truth



Note. (a) Hazy Input, (b) Dehazed Output, (c) Ground Truth.

Figure 52 shows SeeThrough's dehazing output (image1, image2, image3, and image4) against the ground truth (non-hazy/clear image) from the Foggy Road Image Dataset (FRIDA) made by Tarel et al. (2012). It shows that the system can remove a certain amount of haze and can regain color textures that were lost from the input image. However, the dehazed output produces an unnaturally dark scene.

Table 5

PSNR and SSIM Results of the Dehazed Output in Figure 52

Metric	Image	Before Dehazing	After Dehazing
PSNR	image1	9.113	13.521
	image2	9.562	12.309
	image3	8.502	11.221
	image4	8.509	13.432
SSIM	image1	0.694	0.692
	image2	0.769	0.704
	image3	0.690	0.636
	image4	0.690	0.691

Note: For PSNR - The higher the value, the better. For SSIM - The value closer to 1, the better.

Table 5 shows the PSNR and SSIM results of the images shown in figure 52, before and after the dehazing process. To interpret the values, a high PSNR value is better and an SSIM value closer to 1 is better. The results show that in terms of the PSNR value, the system has enhanced the hazy image. The SSIM value shows that there is no significant difference in the image similarity. The researchers assume that the SSIM results can be further improved if the brightness of the dehazing output is corrected.

Table 6

Average PSNR and SSIM Results of FRIDA, Before and After Dehazing

Metric	Image	Before Dehazing	After Dehazing
PSNR	Homogeneous	9.979	12.440
	Heterogeneous	9.778	12.444
	Cloudy Homogeneous	10.970	10.863
	Cloudy Heterogeneous	11.162	10.813
SSIM	Homogenous	0.823	0.691
	Heterogenous	0.729	0.675
	Cloudy Homogenous	0.741	0.595
	Cloudy Heterogenous	0.740	0.587

Note: For PSNR - The higher the value, the better. For SSIM - The value closer to 1, the better.

Table 6 shows the average results of the 66 images from FRIDA, before and after the dehazing process. In terms of PSNR results, the table shows that the dehazing system performed the best on homogeneous and heterogeneous fog. However, the system fell short on cloudy homogeneous and cloudy heterogeneous fog. In terms of SSIM results, the system's dehazed images produced a low similarity index with its ground truth counterpart. The researchers assume that this is due to the darkening of images as a result of the DCP algorithm.

SOP #4: What functional and compatibility tests were conducted to ensure the reliability and usability of both the desktop and mobile applications?

In conducting the functional test, a unit testing approach was used to ensure that every button, text, display, etc., works as intended. A total of 24 functional test cases were written by the researchers, which encompasses both the desktop and mobile versions. This is done to ensure the reliability of SeeThrough in performing its tasks.

The compatibility test was conducted to ensure the usability of SeeThrough across different system environments. For this, a total of 10 test cases were written by the researchers, encompassing both mobile and desktop versions in terms of operating system, screen resolution, and platform version.

Functional Test Result

The researchers visualized the summary of the functionality test cases and gave an overview of the total passed and failed scenarios according to the subject matter experts and IT experts respondents.

Table 7

Functionality Test Results

Test Respondent	Pass	Fail	Test Criteria	Percentage
Subject Matter Expert 1	24	0	24	100.00%
Subject Matter Expert 2	24	0	24	100.00%
Subject Matter Expert 3	24	0	24	100.00%
Subject Matter Expert 4	24	0	24	100.00%
Subject Matter Expert 5	24	0	24	100.00%
Subject Matter Expert 6	24	0	24	100.00%
IT Expert 1	24	0	24	100.00%
IT Expert 2	22	2	24	91.67%
IT Expert 3	24	0	24	100.00%
IT Expert 4	24	0	24	100.00%
IT Expert 5	24	0	24	100.00%
IT Expert 6	24	0	24	100.00%
IT Expert 7	24	0	24	100.00%
IT Expert 8	24	0	24	100.00%
IT Expert 9	24	0	24	100.00%
Overall Result	**358**	**2**	**360**	**99.44%**

Note: The value of test criteria is the total test case scenarios in the functionality test cases.

As shown in table 7, all six (6) subject matter experts found no faults on the system but out of nine (9) IT experts who participated in the functionality testing, one reported finding two errors in the mobile application. The second IT expert's first identified error is related to the camera permission prompt. The second identified error is related to the image dehazing feature of the mobile application. The system's developers were able to quickly verify the errors and have immediately resolved the issue through a step-by-step debugging procedure to replicate the identified error, which ensured that the application remained operational and that the users were able to access and utilize the application as intended.

Compatibility Test Result

In this section, the researchers visualized the summary of the compatibility test cases and gave an overview of the total passed and failed scenarios according to the Subject matter expert and IT Expert respondents.

Table 8

Compatibility Test Results

Test Respondent	Pass	Fail	Test Criteria	Percentage
Subject Matter Expert 1	10	0	10	100.00%
Subject Matter Expert 2	10	0	10	100.00%
Subject Matter Expert 3	10	0	10	100.00%
Subject Matter Expert 4	10	0	10	100.00%
Subject Matter Expert 5	10	0	10	100.00%
Subject Matter Expert 6	10	0	10	100.00%
IT Expert 1	10	0	10	100.00%
IT Expert 2	10	0	10	100.00%
IT Expert 3	10	0	10	100.00%
IT Expert 4	10	0	10	100.00%
IT Expert 5	10	0	10	100.00%
IT Expert 6	10	0	10	100.00%
IT Expert 7	10	0	10	100.00%
IT Expert 8	10	0	10	100.00%
IT Expert 9	10	0	10	100.00%
Overall Result	**150**	**0**	**150**	**100.00%**

Note: The value of test criteria is the total test case scenarios in the compatibility test cases.

As shown in table 8, the fifteen (15) respondents including subject matter experts and IT experts of the compatibility testing did not encounter any faults, bugs, or issues with the specified screen sizes and operating system versions. The screen sizes specified were 720p, 1080p, and 1440p screen resolutions. The operating systems included in the test case were Windows 10 and 11, and Android 10 to 14.

Functionality and Compatibility Test Conclusion

The testing phase was done to ensure that the desktop system and mobile application are functioning the way the researchers have intended for it to be. The test cases used are functionality and compatibility. The functionality test is done to test the system's features and performance. The functionality test utilized a unit testing method to ensure that every module in the system is thoroughly checked for faults or errors. The compatibility test is done to test the availability of the desktop system and mobile application in various screen sizes and operating system versions (Win10, Win11, and Android 10 to 14).

The researchers tested the processing performance of the desktop system and the mobile application. The desktop system was tested on an intel core i7-7820HQ 2.9Ghz 4-core CPU and 16GB RAM environment and was able to process 30 frames per second on average. The mobile application was tested on a QUALCOMM SM6225 snapdragon 685 (6nm) environment and was able to process 12 frames per second on average.

During the IT experts' testing, two (2) of the IT experts left comments for the future improvement of the system. The first IT expert advised that the developers should explicitly mention that the application is free of malware prior to installation, to highlight the navigation tab that the user is currently at, and other comments to improve user experience revolving around ease of use. The second IT Expert noted that the dehazing feature should mind the orange and blue/teal tones for future image quality improvement.

Overall, after the six (6) subject matter experts and nine (9) IT experts' testing, the functionality test results came out with a 99.44% passing rate while the compatibility testing achieved a 100% passing rate.

SOP #5: How was the developed system be evaluated for acceptability in terms of Functional Suitability, Performance Efficiency, Compatibility, Interaction Capability, Reliability, Maintainability, and Flexibility?

To evaluate the desktop system's acceptability, a survey questionnaire was created following the ISO/IEC 25010:2023 software quality standard as a guideline. A detailed result of this evaluation is discussed at the Data Analysis and Interpretation section.

SOP #6: What criteria were used to assess the mobile application's visual experience, performance and stability, and privacy and security to ensure it meets Android core app quality standards?

To evaluate the mobile application's visual experience, the researchers constructed questions that focused on sub-criteria including navigation, UI and graphics, visual quality, and accessibility. The respondents assessed the mobile application's performance and stability as well in terms of its loading times and error prevention respectively. For privacy and security, the application is evaluated through permissions and data and files access.

A detailed result of this evaluation is discussed at the Data Analysis and Interpretation section.

Data Analysis and Interpretation

The data gathered from the ISO 25010, Android Core App Quality, and Dehazing Quality evaluation forms are analyzed and interpreted in this section.

Desktop System Evaluation

The desktop system's acceptability rating was assessed through the ISO/IEC 25010:2023 software quality standard, which is used to evaluate the application's functional suitability, performance efficiency, compatibility, interaction capability, reliability, maintainability, and flexibility. The system's functional suitability is evaluated through functional completeness, functional correctness, and functional appropriateness criteria. The system's performance efficiency is evaluated through time behavior, resource utilization, and capacity criteria. The system's compatibility is evaluated through co-existence and interoperability criteria. For the system's interaction capability, its appropriateness recognizability, learnability, operability, user error protection, user engagement, inclusivity, user assistance, and self-descriptiveness are evaluated. To assess the system's reliability, the criteria used includes the system's faultlessness, availability, fault tolerance, and recoverability. For maintainability, the criteria are as follows:

modularity, analyzability, modifiability, and testability. Lastly, flexibility is evaluated through the adaptability, scalability, and installability criteria.

Table 9

Evaluation Result of End-Users in terms of Functional Suitability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Functional Completeness – The dehazing system provides a set of functions that covers all the specified tasks and intended users' objectives	4.61	0.77	Highly Acceptable
Functional Correctness – The dehazing system provides accurate results when used	4.61	0.60	Highly Acceptable
Functional Appropriateness – The dehazing system provide functions that facilitate the accomplishment of specified tasks and objectives	4.69	0.52	Highly Acceptable
Grand Mean	**4.64**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 9 shows the mean scores of the functional suitability criteria and its verbal interpretation from the end-users' evaluation. The functional suitability criteria garnered a grand mean score of "4.69" with a verbal interpretation of "Highly Acceptable". This result indicates that the desktop system provided functions that exceeded the needs and standards of its intended users.

Table 10

Evaluation Result of IT Experts in terms of Functional Suitability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Functional Completeness – The dehazing system provides a set of functions that covers all the specified tasks and intended users' objectives	4.67	0.49	Highly Acceptable
Functional Correctness – The dehazing system provides accurate results when used	4.17	0.72	Acceptable
Functional Appropriateness – The dehazing system provide functions that facilitate the accomplishment of specified tasks and objectives	4.50	0.52	Highly Acceptable
Grand Mean	**4.44**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 10 shows the mean scores of the functional suitability criteria and its verbal interpretation from the IT experts' evaluation. The functional suitability criteria garnered a grand mean score of "4.44" with a verbal interpretation of "Highly Acceptable". This result indicates that the desktop system's functions are suitable from an IT expert's perspective.

Table 11

Evaluation Result of End-Users in terms of Performance Efficiency

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Time Behavior – The dehazing system performs its functions under specified conditions so that the response time and throughput rates meet the requirements	4.56	0.61	Highly Acceptable
Resource Utilization – The dehazing system uses no more than the specified amount of resources to perform its function under specified conditions	4.61	0.55	Highly Acceptable
Capacity – The dehazing system meets requirements for the maximum limits of a product parameter (e.g. number of image/footages that can be dehazed)	4.61	0.49	Highly Acceptable
Grand Mean		**4.59**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 11 shows the mean scores of the performance efficiency criteria and its verbal interpretation from the end-users' evaluation. The performance efficiency criteria garnered a grand mean score of "4.59" with a verbal interpretation of "Highly Acceptable". This result indicates that the desktop system is efficient in utilizing the resources of the end-user's computer under specified conditions.

Table 12

Evaluation Result of IT Experts in terms of Performance Efficiency

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Time Behavior – The dehazing system performs its functions under specified conditions so that the response time and throughput rates meet the requirements	4.42	0.67	Highly Acceptable
Resource Utilization – The dehazing system uses no more than the specified amount of resources to perform its function under specified conditions	4.58	0.67	Highly Acceptable
Capacity – The dehazing system meets requirements for the maximum limits of a product parameter (e.g. number of image/footages that can be dehazed)	4.42	0.51	Highly Acceptable
Grand Mean	**4.47**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 12 shows the mean scores of the performance efficiency criteria and its verbal interpretation from the IT experts' evaluation. The performance efficiency criteria garnered a grand mean score of "4.47" with a verbal interpretation of "Highly Acceptable". This result indicates that the desktop system's performance is highly acceptable from an IT expert's perspective.

Table 13

Evaluation Result of End-Users in terms of Compatibility

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Co-existence – The dehazing system performs its required function efficiently while sharing a common environment and resources with other existing systems, without detrimental impact on any other existing systems	4.69	0.47	Highly Acceptable
Interoperability – The dehazing system can exchange information with other products and use the information that has been acquired	4.67	0.48	Highly Acceptable
Grand Mean		**4.68**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 13 shows the mean scores of the compatibility criteria and its verbal interpretation from the end-users' evaluation. The compatibility criteria garnered a grand mean score of "4.68" with a verbal interpretation of "Highly Acceptable". This means that the desktop system is capable of using the end-user's IP camera as an input feed and can perform its functions while sharing the same environment and resources with other products or systems.

Table 14

Evaluation Result of IT Experts in terms of Compatibility

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Co-existence – The dehazing system performs its required function efficiently while sharing a common environment and resources with other existing systems, without detrimental impact on any other existing systems	4.50	0.52	Highly Acceptable
Interoperability – The dehazing system can exchange information with other products and use the information that has been acquired	4.25	0.75	Highly Acceptable
Grand Mean		**4.38**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 14 shows the mean scores of the compatibility criteria and its verbal interpretation from the end-users' evaluation. The compatibility criteria garnered a grand mean score of "4.38" with a verbal interpretation of "Highly Acceptable". This indicates that the desktop system has high compatibility with the target system environments.

Table 15

Evaluation Result of End-Users in terms of Interaction Capability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Appropriateness Recognizability – The dehazing system is appropriate for the user's needs	4.61	0.55	Highly Acceptable
Learnability – The dehazing system is easy to learn	4.75	0.50	Highly Acceptable
Operability – The dehazing system is easy to operate and control	4.78	0.42	Highly Acceptable
User Error Protection – The dehazing system protects users against making errors	4.44	0.65	Highly Acceptable
User Engagement – The dehazing system presents information in an inviting and motivating manner, encouraging continued interaction	4.53	0.56	Highly Acceptable
Inclusivity – The dehazing system is appropriate for the user's needs	4.58	0.65	Highly Acceptable
User Assistance – The dehazing system is easy to learn	4.58	0.60	Highly Acceptable
Self-Descriptiveness – The dehazing system is easy to operate and control	4.61	0.55	Highly Acceptable
Grand Mean	**4.61**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 15 shows the mean scores of the interaction capability criteria and its verbal interpretation based on the end-users' evaluation. All criteria obtained a grand mean score of "4.61" with a verbal interpretation of "Highly Acceptable", which means that the desktop system is engaging, inclusive, and can be easily learned and operated by the intended users to complete a task.

Table 16

Evaluation Result of IT Experts in terms of Interaction Capability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Appropriateness Recognizability – The dehazing system is appropriate for the user's needs	4.50	0.67	Highly Acceptable
Learnability – The dehazing system is easy to learn	4.33	0.98	Highly Acceptable
Operability – The dehazing system is easy to operate and control	4.42	1.00	Highly Acceptable
User Error Protection – The dehazing system protects users against making errors	4.08	0.67	Acceptable
User Engagement – The dehazing system presents information in an inviting and motivating manner, encouraging continued interaction	4.33	0.65	Highly Acceptable
Inclusivity – The dehazing system is appropriate for the user's needs	4.58	0.51	Highly Acceptable
User Assistance – The dehazing system is easy to learn	4.25	1.06	Highly Acceptable
Self-Descriptiveness – The dehazing system is easy to operate and control	4.25	0.87	Highly Acceptable
Grand Mean	**4.34**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 16 shows the mean scores of the compatibility criteria and its verbal interpretation from the IT experts' evaluation. The interaction capability criteria obtained a grand mean score of "4.34" with a verbal interpretation of "Highly Acceptable", which means that the desktop system's user interface is deemed as highly acceptable by the IT experts.

Table 17

Evaluation Result of End-Users in terms of Reliability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Faultlessness – The dehazing system performs its functions without fault/error under normal operation	4.42	0.77	Highly Acceptable
Availability – The dehazing system is operational and accessible when needed	4.72	0.45	Highly Acceptable
Fault Tolerance – The dehazing system operates as intended despite the presence of hardware or software faults	4.58	0.60	Highly Acceptable
Recoverability – In the event of an interruption or failure, the dehazing system can recover the data directly affected and re-establish the desired state of the system	4.53	0.74	Highly Acceptable
Grand Mean	**4.56**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 17 shows the mean scores of the reliability criteria and its verbal interpretation based on the end-users' evaluation. All criteria obtained a grand mean score of "4.56" with a verbal interpretation of "Highly Acceptable", which means that the end-user can perform tasks for a specified period of time without interruptions and failures.

Table 18

Evaluation Result of IT Experts in terms of Reliability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Faultlessness – The dehazing system performs its functions without fault/error under normal operation	4.33	0.65	Highly Acceptable
Availability – The dehazing system is operational and accessible when needed	4.33	0.65	Highly Acceptable
Fault Tolerance – The dehazing system operates as intended despite the presence of hardware or software faults	4.17	0.72	Acceptable
Recoverability – In the event of an interruption or failure, the dehazing system can recover the data directly affected and re-establish the desired state of the system	4.00	0.85	Acceptable
Grand Mean		**4.21**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 18 shows the mean scores of the reliability criteria and its verbal interpretation based on the IT experts' evaluation. All criteria obtained a grand mean score of "4.21" with a verbal interpretation of "Highly Acceptable", which means that the desktop system can perform its functions for a specified period of time without interruptions and failures. The recoverability criteria, however, achieved the lowest rating among the four criteria under reliability. This is due to the system having no support for data recovery in the event of a power interruption or similar. The IT experts deemed this as "Acceptable", with a mean score of 4.00, which means that despite the lack of recovery support, it does not affect the system's functionality.

Table 19

Evaluation Result of End-Users in terms of Maintainability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Modularity – Changes to one component does not affect other components of the dehazing system	4.58	0.55	Highly Acceptable
Analyzability – The data from the dehazing system can be easily assessed for system changes or diagnosed for causes of failure	4.67	0.53	Highly Acceptable
Modifiability – The dehazing system can be easily modified without introducing defects or degrading the product quality	4.31	0.98	Highly Acceptable
Testability – The dehazing system enables an objective and feasible test to be designed and performed to determine whether a requirement is met	4.69	0.47	Highly Acceptable
Grand Mean	**4.56**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 19 shows the mean scores of the maintainability criteria and its verbal interpretation from the end-users' evaluation. Overall, the grand mean score of the maintainability criteria is "4.56" with a verbal interpretation of "Highly Acceptable", which means that the desktop system can be easily maintained or modified by the intended maintainers.

Table 20

Evaluation Result of IT Experts in terms of Maintainability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Modularity – Changes to one component does not affect other components of the dehazing system	4.50	0.52	Highly Acceptable
Analyzability – The data from the dehazing system can be easily assessed for system changes or diagnosed for causes of failure	4.17	0.83	Acceptable
Modifiability – The dehazing system can be easily modified without introducing defects or degrading the product quality	4.42	0.67	Highly Acceptable
Testability – The dehazing system enables an objective and feasible test to be designed and performed to determine whether a requirement is met	4.50	0.67	Highly Acceptable
Grand Mean	**4.40**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 20 shows the mean scores of the maintainability criteria and its verbal interpretation from the IT experts' evaluation. Overall, the grand mean score of the maintainability criteria is "4.40" with a verbal interpretation of "Highly Acceptable", which means that the desktop system's functions are modular enough to be modified without negatively affecting other functions. The analyzability criteria, however, achieved the lowest rating among the four criterias under maintainability. The IT experts deemed this as "Acceptable", with a mean score of 4.17, which means that despite the lack of system modules' analyzability, the system's maintainability is still manageable to a certain extent.

Table 21

Evaluation Result of End-Users in terms of Flexibility

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Adaptability – The dehazing can be used in different system environments	4.64	0.59	Highly Acceptable
Scalability – The dehazing system can handle growing or shrinking workloads, or adapt its capacity to handle variability	4.64	0.54	Highly Acceptable
Installability – The dehazing system can be successfully installed or uninstalled	4.86	0.35	Highly Acceptable
Grand Mean	**4.71**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 21 shows the mean scores of the flexibility criteria and its verbal interpretation from the end-users' evaluation. Overall, the grand mean score of the flexibility criteria is "4.71" with a verbal interpretation of "Highly Acceptable", which means that the intended users can install the desktop system on their system environment without fail.

Table 22

Evaluation Result of IT Experts in terms of Flexibility

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Adaptability – The dehazing can be used in different system environments	4.42	0.90	Highly Acceptable
Scalability – The dehazing system can handle growing or shrinking workloads, or adapt its capacity to handle variability	4.17	0.72	Acceptable
Installability – The dehazing system can be successfully installed or uninstalled	4.17	0.94	Acceptable
Grand Mean	**4.25**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 22 shows the mean scores of the flexibility criteria and its verbal interpretation from the IT experts' evaluation. Overall, the grand mean score of the flexibility criteria is "4.25" with a verbal interpretation of "Highly Acceptable", which means that the desktop system can be installed on different environments provided that it complies with the minimum requirements of the system. The installability and scalability criterias achieved the lowest rating among the three criterias under flexibility. The IT experts deemed this as "Acceptable", with both criterias having a mean score of 4.17, which means that there is a room for improvement when it comes to enabling an easier installation process and having the system a lot more scalable in terms of workloads.

Table 23

Overall Result of ISO/IEC 25010:2023 Evaluation by End-Users

Factor	Mean	Standard Deviation	Verbal Interpretation
Functional Suitability	4.64	0.63	Highly Acceptable
Performance Efficiency	4.59	0.55	Highly Acceptable
Compatibility	4.68	0.47	Highly Acceptable
Interaction Capability	4.61	0.57	Highly Acceptable
Reliability	4.56	0.66	Highly Acceptable
Maintainability	4.56	0.68	Highly Acceptable
Flexibility	4.71	0.51	Highly Acceptable
Overall	**4.61**	**0.59**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 23 shows the overall result of the end-users' evaluation of the desktop system. The results show that the desktop system garnered a grand mean score of "4.61" with a verbal interpretation of "Highly Acceptable" which indicates a high satisfaction rating for the end-users.

Table 24

Overall Result of Evaluation by IT Experts

Factor	Mean	Standard Deviation	Verbal Interpretation
Functional Suitability	4.44	0.61	Highly Acceptable
Performance Efficiency	4.47	0.61	Highly Acceptable
Compatibility	4.38	0.65	Highly Acceptable
Interaction Capability	4.34	0.81	Highly Acceptable
Reliability	4.21	0.71	Highly Acceptable
Maintainability	4.40	0.68	Highly Acceptable
Flexibility	4.25	0.84	Highly Acceptable
Overall	**4.35**	**0.72**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 24 shows the overall result of the IT experts evaluation of the desktop system. The results show that the desktop system garnered a grand mean score of “4.35” with a verbal interpretation of “Highly Acceptable” which indicates a high acceptance rating by the IT experts.

Mobile Application Evaluation

To assess the mobile application’s acceptability rating, the Android core app quality standards are used to evaluate the application’s visual experience, performance and stability, and privacy and security. The visual experience is evaluated through the mobile application’s navigation, user interface and graphics, visual quality, and accessibility criterias. The mobile application’s performance, stability, privacy, and security are evaluated as well.

Table 25

Evaluation Result of End-Users in terms of Navigation

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application support back button navigation and functions correctly	4.75	0.44	Highly Acceptable
The application supports gesture navigation for going back or going to the home screen.	4.72	0.45	Highly Acceptable
The application preserves information when minimizing the app or changing the states.	4.61	0.49	Highly Acceptable
Grand Mean	**4.69**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 25 shows the mean scores of the navigation criteria and their verbal interpretation for the end-user’s evaluation. The grand mean score of the navigation criteria is “4.69” with a verbal interpretation of “Highly Acceptable”. The result indicates that the mobile application’s navigation support met end-user standards.

Table 26

Evaluation Result of IT Experts in terms of Navigation

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application support back button navigation and functions correctly	4.58	0.51	Highly Acceptable
The application supports gesture navigation for going back or going to the home screen.	4.33	0.89	Highly Acceptable
The application preserves information when minimizing the app or changing the states.	4.50	0.52	Highly Acceptable
Grand Mean	**4.47**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 26 shows the mean scores of the navigation criteria and their verbal interpretation for the IT Expert's evaluation. The grand mean score of the IT Expert's evaluation is "4.47" with a verbal interpretation of "Highly Acceptable". Although it has a slightly lower rating compared to end-users, the navigation features are still considered highly acceptable by IT experts.

Table 27

Evaluation Result of End-Users in terms of UI & Graphics

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application supports landscape and portrait orientations. Orientations and fold states expose essentially the same features and actions and preserve functional parity.	4.53	0.51	Highly Acceptable
The application fills the whole section of the device.	4.67	0.48	Highly Acceptable
Grand Mean	**4.60**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 27 shows the mean scores of the UI & Graphics criteria and their verbal interpretation for the end-user's evaluation. The grand mean score of the end-user's evaluation is "4.60" with a verbal interpretation of "Highly Acceptable". The features evaluated include support for both landscape and portrait orientations and screen utilization.

Table 28

Evaluation Result of IT Experts in terms of UI & Graphics

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application supports landscape and portrait orientations. Orientations and fold states expose essentially the same features and actions and preserve functional parity.	3.83	1.19	Acceptable
The application fills the whole section of the device.	4.58	0.51	Highly Acceptable
Grand Mean	**4.21**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 28 shows the mean scores of the UI & Graphics criteria and their verbal interpretation for the IT Expert's evaluation. IT Experts had a more mixed evaluation for the UI & graphics criteria, with mean scores of "3.83" and "4.58". The grand mean score of "4.21" suggests that, while screen utilization is highly acceptable, support for multiple orientations could be improved.

Table 29

Evaluation Result of End-Users in terms of Visual Quality

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application displays graphics, text, images, and other UI elements without noticeable distortion, blurring, or pixelation.	4.64	0.54	Highly Acceptable
The application displays text and text blocks in an acceptable manner for each	4.53	0.51	Highly Acceptable

of the app's supported languages.			
The application supports a dark theme.	4.64	0.59	Highly Acceptable
Grand Mean	**4.60**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 29 shows the mean scores of the Visual Quality criteria and their verbal interpretation for the end-user's evaluation. The grand mean score of the end-user's evaluation is "4.60" with a verbal interpretation of "Highly Acceptable". Key indicators such as text and graphics quality and dark theme support are all highly rated.

Table 30

Evaluation Result of IT Experts in terms of Visual Quality

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application displays graphics, text, images, and other UI elements without noticeable distortion, blurring, or pixelation.	4.08	0.90	Acceptable
The application displays text and text	4.42	0.67	Highly Acceptable

blocks in an acceptable manner for each of the app's supported languages.			
The application supports a dark theme.	4.50	0.52	Highly Acceptable
Grand Mean	**4.33**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 30 shows the mean scores of the Visual Quality criteria and their verbal interpretation for the IT Expert's evaluation. The grand mean score of the IT Expert's evaluation is "4.33" with a verbal interpretation of "Highly Acceptable". While generally positive, the quality of graphics and text received a lower score, indicating room for improvement.

Table 31

Evaluation Result of End-Users in terms of Accessibility

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Touch targets like buttons/switches are readable, clickable, and pleasing to look at.	4.69	0.47	Highly Acceptable
The application's texts and background	4.67	0.48	Highly Acceptable

color maintain enough color contrast ratio to avoid eye strains and fatigue.		
Grand Mean	**4.68**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 31 shows the mean scores of the Accessibility criteria and their verbal interpretation for the end-user's evaluation. The grand mean score of the end-user's evaluation is "4.68" with a verbal interpretation of "Highly Acceptable". The application's touch targets and color contrast are considered highly acceptable, making it easier for users to access the visual elements of the application.

Table 32

Evaluation Result of IT Experts in terms of Accessibility

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
Touch targets like buttons/switches are readable, clickable, and pleasing to look at.	4.58	0.51	Highly Acceptable

The application's texts and background color maintain enough color contrast ratio to avoid eye strains and fatigue.	4.33	0.49	Highly Acceptable
Grand Mean	**4.46**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 32 shows the mean scores of the Accessibility criteria and their verbal interpretation for the IT Expert's evaluation. The grand mean score of the IT Expert's evaluation is "4.46" with a verbal interpretation of "Highly Acceptable". Although slightly lower than end-user ratings, the accessibility aspects still meet the high standards evaluated by the IT Experts.

Table 33

Evaluation Result of End-Users in terms of Stability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application does not crash or prompt "Android Not Responding" errors.	4.75	0.50	Highly Acceptable

Grand Mean	**4.75**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 33 shows the mean scores of the Stability criteria and their verbal interpretation for the end-user's evaluation. The grand mean score of the end-user's evaluation is "4.75" with a verbal interpretation of "Highly Acceptable". The result indicates that the application rarely crashes or encounters errors.

Table 34

Evaluation Result of IT Experts in terms of Stability

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application does not crash or prompt	4.58	0.51	Highly Acceptable

"Android Not Responding" errors.		
Grand Mean	**4.58**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 34 shows the mean scores of the Stability criteria and their verbal interpretation for the IT Expert's evaluation. The grand mean score of the IT Expert's evaluation is "4.58" with a verbal interpretation of "Highly Acceptable". The result confirms that the application maintains high reliability and stability.

Table 35

Evaluation Result of End-Users in terms of Performance

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation

The application loads quickly upon opening.	4.61	0.49	Highly Acceptable
The application loads pages or layout in a timely manner.	4.61	0.55	Highly Acceptable
Grand Mean		**4.61**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 35 shows the mean scores of the Performance criteria and their verbal interpretation for the end-user's evaluation. The grand mean score of the end-user's evaluation is "4.61" with a verbal interpretation of "Highly Acceptable". The evaluation includes quick loading times and timely page loading. The result indicates satisfaction with the application's performance as evaluated by the end-users.

Table 36

Evaluation Result of IT Experts in terms of Performance

Constructed Statement	Mean	Standard	Verbal Interpretation

		Deviation	
The application loads quickly upon opening.	4.50	0.52	Highly Acceptable
The application loads pages or layout in a timely manner.	4.58	0.51	Highly Acceptable
Grand Mean	**4.54**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 36 shows the mean scores of the Performance criteria and their verbal interpretation for the IT Expert's evaluation. The grand mean score of the IT Expert's evaluation is "4.54" with a verbal interpretation of "Highly Acceptable". The result is consistent with the end-users results, and therefore the application is perceived to perform well under various conditions.

Table 37

Evaluation Result of End-Users in terms of Permissions

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application does request for permission to access the Camera.	4.75	0.44	Highly Acceptable
The application clearly conveys why certain permissions are needed.	4.67	0.48	Highly Acceptable
The application respects the decision if the user denied such permissions.	4.67	0.48	Highly Acceptable
Grand Mean	**4.69**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 37 shows the mean scores of the Permissions criteria and their verbal interpretation for the end-user’s evaluation. The grand mean score of the end-user’s evaluation is “4.69” with a verbal interpretation of “Highly Acceptable”. The result indicates that users appreciate clear permission communication before using the application's live dehazing feature.

Table 38

Evaluation Result of IT Experts in terms of Permissions

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The application does request for permission to access the Camera.	4.67	0.49	Highly Acceptable
The application clearly conveys why certain permissions are needed.	3.75	1.22	Acceptable
The application respects the decision if the user denied such permissions.	4.25	0.62	Highly Acceptable
Grand Mean	**4.22**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 38 shows the mean scores of the Permissions criteria and their verbal interpretation for the IT Expert's evaluation. The grand mean score of the IT Expert's evaluation is "4.22" with a verbal interpretation of "Highly Acceptable". The result indicates that while camera permissions and user respect are highly acceptable, clarity on why permissions are needed could be improved, as evaluated by the IT Experts.

Table 39

Evaluation Result of End-Users in terms of Data and Files

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The image is saved in the screenshot folder found on the device's internal storage.	4.64	0.49	Highly Acceptable
The application did not request for the user's sensitive data.	4.61	0.77	Highly Acceptable
Grand Mean	**4.63**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 39 shows the mean scores of the Data and Files criteria and their verbal interpretation for the end-user's evaluation. The grand mean score of the end-user's evaluation is "4.63" with a verbal interpretation of "Highly Acceptable". The results indicate that the application effectively manages image saving and does not request user's sensitive data.

Table 40

Evaluation Result of IT Experts in terms of Data and Files

Constructed Statement	Mean	Standard Deviation	Verbal Interpretation
The image is saved in the screenshot folder found on the device's internal storage.	4.42	0.51	Highly Acceptable
The application did not request for the user's sensitive data.	4.25	1.14	Highly Acceptable
Grand Mean	**4.33**		**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 40 shows the mean scores of the Data and Files criteria and their verbal interpretation for the IT Expert's evaluation. The grand mean score of the IT Expert's evaluation is "4.33" with a verbal interpretation of "Highly Acceptable". The result indicates that both factors are highly rated by the IT Experts.

Table 41

Overall Result of Android Core App Quality Evaluation by End-Users

Factor	Mean	Standard Deviation	Verbal Interpretation
Navigation	4.69	0.46	Highly Acceptable
UI & Graphics	4.60	0.49	Highly Acceptable
Visual Quality	4.60	0.55	Highly Acceptable
Accessibility	4.68	0.47	Highly Acceptable
Stability	4.75	0.50	Highly Acceptable
Performance	4.61	0.52	Highly Acceptable
Permission	4.69	0.46	Highly Acceptable
Data and Files	4.63	0.64	Highly Acceptable
Overall	**4.66**	**0.51**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 41 shows the overall mean for end-user evaluations across all factors and received a grand mean of "4.66", with all factors rated as "Highly Acceptable". This indicates strong overall user satisfaction with the application.

Table 42

Overall Result of Android Core App Quality Evaluation by IT Experts

Factor	Mean	Standard Deviation	Verbal Interpretation
Navigation	4.47	0.65	Highly Acceptable
UI & Graphics	4.21	0.98	Highly Acceptable
Visual Quality	4.33	0.72	Highly Acceptable
Accessibility	4.46	0.51	Highly Acceptable
Stability	4.58	0.51	Highly Acceptable
Performance	4.54	0.51	Highly Acceptable
Permission	4.22	0.90	Highly Acceptable
Data and Files	4.33	0.87	Highly Acceptable
Overall	**4.39**	**0.71**	**Highly Acceptable**

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 42 shows the overall mean for IT expert evaluations across all factors, which received a grand mean of “4.39”, also with all factors rated as “Highly Acceptable”. Although slightly lower than the grand mean of the end-user, this still reflects a high level of acceptance and satisfaction from IT professionals.

Table 43

Evaluation Result of Desktop System

Criteria	Mean	Interpretation	Rank
Functional Suitability	4.59	Highly Acceptable	3
Performance Efficiency	4.56	Highly Acceptable	4
Compatibility	4.60	Highly Acceptable	1
Interaction Capability	4.54	Highly Acceptable	5
Reliability	4.47	Highly Acceptable	7
Maintainability	4.52	Highly Acceptable	6
Flexibility	4.60	Highly Acceptable	2
Average Mean	**4.55**	**Highly Acceptable**	

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 43 shows the evaluation result of the forty-eight (48) respondents towards the SeeThrough desktop system. It shows that Compatibility ranked first, followed by Flexibility, Functional Suitability, Performance Efficiency, Interaction Capability, and Maintainability, with Reliability ranking last. Based on the ISO/IEC 25010:2023 criteria, all aspects of the desktop system have been deemed as “Highly Acceptable” with an average mean score of “4.55”.

Table 44

Evaluation Result of Mobile Application

Criteria	Mean	Interpretation	Rank
Navigation	4.64	Highly Acceptable	2
UI & Graphics	4.50	Highly Acceptable	8
Visual Quality	4.53	Highly Acceptable	7
Accessibility	4.63	Highly Acceptable	3
Stability	4.71	Highly Acceptable	1
Performance	4.59	Highly Acceptable	4
Permissions	4.58	Highly Acceptable	5
Data and Files	4.55	Highly Acceptable	6
Average Mean	**4.59**	**Highly Acceptable**	

Note: For interpretation, the following remarks apply to the mean interval: 5.00 – 4.20 (Highly Acceptable), 4.19-3.40 (Acceptable), 3.39 – 2.60 (Moderately Acceptable), 2.59-1.80 (Fairly Acceptable), and 1.79 – 1.00 (Poorly Acceptable).

Table 44 shows the evaluation result of the forty-eight (48) respondents towards the SeeThrough mobile application. It shows that Stability ranked first, followed by Navigation, Accessibility, Performance, Permissions, Data and Files, Visual Quality, and with UI & Graphics ranking last. Most aspects are deemed as “Highly Acceptable” by the evaluators. Based on the Android Core App Quality criteria, all aspects of the mobile application have been deemed as “Highly Acceptable” with an average mean score of “4.59”

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

This chapter contains the summary of findings which discusses the actual output, test results, evaluation results, the researchers' conclusion which explains how the objectives were met, and the recommendations which offer suggestions for the project's future improvement.

Summary of Findings

1. The development and evaluation of the SeeThrough Desktop and Mobile applications highlight their robust performance and high acceptability across various criteria. The desktop application was designed to provide real-time dehazing for surveillance footage, videos, and images. It connects to surveillance cameras and saves dehazed footage or images to user-specified directories. However, due to the computational complexity of the dehazing method, it can handle only one surveillance feed at a time and does not support recording features. Similarly, the mobile application offers real-time and image dehazing capabilities with the option to toggle the dehazing feature on or off. It also respects user decisions regarding camera use but lacks recording functionality due to computational constraints.

2. The evaluation of the SeeThrough system revealed several key findings about its performance. Notably, it significantly improved the clarity of objects in images derived from the FRIDA dataset. This demonstrates a clear enhancement in visual quality. However, there were some issues. Under cloudy homogeneous and heterogeneous fog conditions, the system showed lowered SSIM scores. This can be linked to the Dark Channel Prior (DCP) algorithm, which tends to darken images in these situations.

   When looking at video processing, the desktop application performed admirably. It managed to handle 480p video at a rate of 30 frames per second. This indicates that it possesses strong real-time dehazing capabilities. On the other hand, the mobile application only managed 480p video at a slower pace—12 frames per second. This reflects the limitations that come with mobile hardware when compared to its desktop counterpart. In summary, the system does excel in enhancing image clarity but faces challenges with certain fog conditions. Additionally, performance varies across different platforms as well.

3. The assessment of SeeThrough's performance in removing haze, conducted with the FRIDA dataset, indicated that while there was a notable increase in PSNR values—pointing to better image quality—the SSIM values did not exhibit any improvements. This suggests that the structural similarity to the original images was limited, potentially due to the darkening effect of the

DCP algorithm. Nevertheless, the system demonstrated effective performance in both uniform and varying fog conditions, although it faced challenges with fog types characterized as cloudy. The functional and compatibility evaluations underscored a high level of reliability and user-friendliness, achieving a pass rate of 99.44% in functionality tests and 100% in compatibility assessments, confirming the application's effective operation in diverse environments. Any minor issues highlighted by IT specialists were swiftly addressed, preserving the application's integrity.

4. The SeeThrough system, which includes both desktop and mobile applications, exhibited impressive reliability and user-friendliness during thorough functionality and compatibility assessments. The functional testing, utilizing a unit testing strategy across 24 distinct test scenarios, achieved a 99.44% success rate, with only two issues reported and swiftly addressed by IT specialists. Compatibility tests conducted on various operating systems and screen resolutions resulted in a flawless 100% success rate, confirming the system's durability across different environments. Performance evaluations showed that the desktop application processed videos at an average of 30 frames per second, whereas the mobile app handled 12 frames per second, reflecting the constraints of mobile hardware. Insights from IT professionals pointed out the necessity for malware assurance and improved user navigation, as well as recommendations for enhancing the dehazing feature. In summary, the testing confirmed that SeeThrough maintains strong

functionality and compatibility, making it suitable for real-time dehazing tasks on both platforms.

5. The SeeThrough application has received high commendation from both end-users and IT experts, achieving an overall mean score of 4.66 from end-users and 4.39 from IT experts, indicating high acceptability. End-users particularly appreciate the application's navigation (4.69), stability (4.75), and permissions management (4.69), reflecting its intuitive design and reliability. Visual quality, accessibility, and performance also received high ratings, demonstrating that the application is user-friendly, visually appealing, and efficient in performance.

   IT experts share a positive view, highlighting the application's stability (4.58) and performance (4.54) while noting some room for improvement in UI & graphics (4.21) and clarity of permission requests (4.22). Both desktop and mobile versions of the application were rated highly acceptable, with desktop achieving a mean score of 4.55 and mobile 4.59, emphasizing compatibility, flexibility, and stability as key strengths. Overall, the SeeThrough application is deemed effective, reliable, and satisfactory by both user groups.

Conclusions

1. The SeeThrough Desktop and Mobile applications exhibit strong performance in the real-time dehazing of surveillance footage, videos, and images. By designing the desktop system to utilize the available CPU cores of the

computer, the dehazing method was able to perform in real-time, closing the research gap that was identified by Liu et al. (2021). Specifically, a real-time performance was achieved by using a multithreaded approach to handle video capture and frame processing concurrently. By closing this gap, future researchers are provided a framework to explore other dehazing methods for surveillance systems without a lack of real-time performance. However, they are restricted to processing one feed at a time and do not have recording capabilities due to limitations in computational power. Overall, the system's effectiveness and high user acceptance underscore its potential as a significant resource for improving visual clarity in surveillance settings.

2. The assessment of the SeeThrough system indicated notable enhancements in image sharpness, especially for images sourced from the FRIDA dataset, showcasing its capability in visual improvement. Nonetheless, the system encounters difficulties with cloudy homogeneous and cloudy heterogeneous fog conditions because the DCP algorithm tends to darken images. Additionally, while the desktop application processes 480p video at 30 frames per second effectively, the mobile version is restricted to 12 frames per second due to limitations in hardware.
3. SeeThrough demonstrates effective dehazing capabilities, as indicated by improved PSNR values, though the increase in SSIM is limited due to the darkening effect associated with the DCP algorithm. It manages uniform fog

types well and offers strong reliability and ease of use, achieving high success in both functionality and assessments.

4. The SeeThrough system showcased remarkable reliability and ease of use, attaining a success rate of 99.44% during functional testing and a perfect 100% in compatibility tests. This confirms that it is well-suited for real-time dehazing tasks on both desktop and mobile devices.
5. The SeeThrough app is highly esteemed by both users and IT professionals, with average ratings of 4.66 and 4.39, respectively. It receives favorable feedback for its navigation, stability, and permissions management, reflecting its user-friendly design and dependability. Users commend both the desktop and mobile versions for their compatibility, flexibility, and stability, reinforcing the application's efficiency, trustworthiness, and high level of user contentment.

Recommendations

The following recommendations are based on the suggestions and comments of the respondents who participated in the study and the researchers' own findings to help guide future researchers that may develop a similar project and make potential improvements to the system.

1. Enhancement of Dehazing Algorithm by refining the DCP Algorithm to address darkening, improve brightness and contrast, and enhance SSIM values.

2. Future researchers may explore dashboard camera applications of the dehazing system for vehicles as well as to explore the use of application of a dehazing method on eyewear devices.
3. Extend the mobile application’s platform. Enable the use of the dehazing app on iOS platforms.
4. Improve mobile application’s dehazing performance. This will enable simultaneous dehazing and recording and lower the minimum mobile specification requirements to run the real-time dehazing feature.
5. Improve UI design. Add tooltips to guide the user and highlight the tab that the user is currently at in the desktop system.
6. Implementation of hardware acceleration by exploring GPU utilization to improve the desktop system’s dehazing performance to enable simultaneous dehazing of more than one live feed.

REFERENCES

Abed, S., Ali, M. H., & Alshayeji, M. H. (2020). Enhanced GPU-Based Anti-Noise Hybrid Edge Detection Method. Computer Systems Science and Engineering, 35(1), 21–37. https://doi.org/10.32604/csse.2020.35.021

Admin. (2022, July 14). Benefits of Java for Mobile Application Development - NCube. nCube. https://ncube.com/benefits-of-java-for-mobile-application-development/

Ai, S., & Kwon, J. (2020). Extreme low-light image enhancement for surveillance cameras using attention U-Net. Sensors, 20(2), 495. https://doi.org/10.3390/s20020495

Aksoy, Bekir & Salman, Osamah. (2020). A New Hybrid Filter Approach for Image Processing. Sakarya University Journal of Computer and Information Sciences. 3. 334-342. 10.35377/saucis.03.03.785749.

Baig, N. (2021). Development of Enhanced Image Dehazing Techniques for Security and Surveillance. National University of Science & Technology, Islamabad. Retrieved from http://prr.hec.gov.pk/jspui/handle/123456789/18250

Bakurov, I., Buzzelli, M., Schettini, R., Castelli, M., Vanneschi, L. (2022). Structural similarity index (SSIM) revisited: A data-driven approach. Expert Systems with Applications, 189, 116087. https://doi.org/10.1016/j.eswa.2021.116087.

Basic Information about Visibility | US EPA. (2023, February 14). US EPA. https://www.epa.gov/visibility/basic-information-about-visibility#:~:text=Haze%20is%20caused%20when%20sunlight,color%20of%20what%20we%20see.

Boesch, G. (2023). What is OpenCV? The Complete Guide (2024). viso.ai. https://viso.ai/computer-vision/opencv/

Chawla, P., Kumar, S., Sharma, A. M., Goyal, B., & Dogra, A. (2020, January 28). Image Processing: Real-time image dehazing clarifies digital images. Laser Focus World. https://www.laserfocusworld.com/detectors-imaging/article/14074042/realtime-image-dehazing-clarifies-digital-images

Çimtay, Y. (2020). TOWARDS REAL TIME IMAGE DEHAZING ON ANDROID OPERATING SYSTEM. Communications Faculty of Sciences University of Ankara.

Series A2-A3: Physics, Engineerigng Physics, Electronic Engineering and Astronomy, 62(2), 177–188. https://doi.org/10.33769/aupse.811810

Cimtay, Y. (2021). Smart and real-time image dehazing on mobile devices. Journal of Real-Time Image Processing, 18(6), 2063–2072. https://doi.org/10.1007/s11554-021-01085-z

Dahiya, P. (2023). Real Time Systems. GeeksforGeeks. https://www.geeksforgeeks.org/real-time-systems/

Duthon, P., Colomb, M., & Bernardin, F. (2020). Fog classification by their droplet size distributions: application to the characterization of Cerema's platform. Atmosphere (Basel), 11(6), 596. https://doi.org/10.3390/atmos11060596

Engin, D., Genc, A., & Ekenel, H. K. (2018). Cycle-Dehaze: Enhanced CycleGAN for single image dehazing. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (pp. 938-9388). IEEE. https://doi.org/10.1109/CVPRW.2018.00127

Fan, L., Zhang, F., Fan, H., Wu, G., & Lu, H. (2019). Brief review of image denoising techniques. Visual Computing for Industry, Biomedicine, and Art, 2(1), 7. https://doi.org/10.1186/s42492-019-0016-7

Guo, F., Yang, J., Liu, Z., & Tang, J. (2023, June 7). Haze removal for single image: A comprehensive review. Neurocomputing, 537, 85–109. https://doi.org/10.1016/j.neucom.2023.03.061

Hammoudeh, M., Alsaykhan, M., Alsalameh, R., & Althwaibi, N. (2022). Computer vision: A review of detecting objects in videos – Challenges and techniques. International Journal of Online and Biomedical Engineering (iJOE), 18, 15-27. https://doi.org/10.3991/ijoe.v18i01.27577[1]

Harish Babu, G., & Venkatram, N. (2021). Density based Traffic Monitoring system in Haze and FOG Conditions. World Academy of Research in Science and Engineering, 9(3), 2. https://doi.org/10.30534/ijeter/2021/22932021

He, K., Sun, J., & Tang, X. (2009, June 1). *Single image haze removal using dark channel prior*. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5206515

Innodata. (2022, January 5). Ethical issues in computer vision and strategies for success. Innodata Inc. https://innodata.com/ethical-issues-in-computer-vision-and-strategies-for-success/

Jiao, W., Jia, X., Liu, Y., Jiang, Q., & Sun, Z. (2021). Single image mixed dehazing method based on numerical iterative model and DehazeNet. PloS one, 16(7), e0254664. https://doi.org/10.1371/journal.pone.0254664

Kaur, H., Koundal, D., & Kadyan, V. (2021). Image Fusion Techniques: A Survey. Archives of Computational Methods in Engineering, 28(3), 4425–4447. https://doi.org/10.1007/s11831-021-09540-7

Koranga, P., & Kumawat, S. (2020, December 1). A review on comparison of different techniques of image dehazing. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9358387

Koren, I., & Krishna, C. M. (2021). Checkpointing. In I. Koren & C. M. Krishna (Eds.), Fault-Tolerant Systems (Second Edition) (pp. 203-235). Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-818105-8.00016-4

Kumar, R. C., Steffina, I. D., Kamali, D., & Selciya, J. J. (2021). Image Dehazing by using Laplacian Approach with Contrast Adjustment. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, 9(5), 55-61. Retrieved from https://ijireeice.com/wp-content/uploads/2021/05/IJIREEICE.2021.9510.pdf

Lee, S., Yun, S. H., Nam, J., Won, C. S., & Jung, S. (2016). A review on dark channel prior based image dehazing algorithms. Eurasip Journal on Image and Video Processing, 2016(1). https://doi.org/10.1186/s13640-016-0104-y

Lenihan, A. (2022, October 4). *What is Real Time Streaming Protocol (RTSP)*. Dacast. Retrieved October 18, 2023, from https://www.dacast.com/blog/what-is-real-time-streaming-protocol/

Li, C., Yuan, C., Pan, H., Yang, Y., Wang, Z., Zhou, H., & Xiong, H. (2023). Single-image dehazing based on improved bright channel prior and dark channel prior. Electronics, 12(2), 299. https://doi.org/10.3390/electronics12020299

Liu, J., Wang, S., Wang, X., Ju, M., & Zhang, D. (2021). A Review of Remote Sensing Image Dehazing. *Sensors*, *21*(11), 3926. https://doi.org/10.3390/s21113926

Liu, P., Zhang, H., & Mi, Y. (2023). Research on Road Scene Image Dehazing Method based on Improved Dark Channel Prior. www.airitilibrary.com. https://doi.org/10.6919/ICJE.202304_9(4).0018

Malav, R., Kim, A., Sahoo, S. R., & Pandey, G. (2019). DHSGAN: an end to end dehazing network for fog and smoke. In Lecture Notes in Computer Science (pp. 593–608). https://doi.org/10.1007/978-3-030-20873-8_38

Mane, T., Mane, P., Jadhav, S., Ingale, N., & Rane, P. (2022). Image Enhancement for Surveillance Videos using Quick Browsing system. In Proceedings of the 7th International Conference on Innovations and Research in Technology and Engineering (ICIRTE-2022), organized by VPPCOE & VA, Mumbai-22, India. Retrieved from SSRN: https://ssrn.com/abstract=4113703 or http://dx.doi.org/10.2139/ssrn.4113703

Mehta, A., Sinha, H., Narang, P., & Mandal, M. (2020). HIDeGan: A Hyperspectral-guided Image Dehazing GAN. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). doi:10.1109/cvprw50498.2020.00114

Meltzer, R. (2023, January 3). What is a Programming Library? A Beginner's Guide. CareerFoundry. https://careerfoundry.com/en/blog/web-development/programming-library-guide/

Microsoft. (2021, November 3). *Why Visual Studio code?* Visual Studio Code. https://code.visualstudio.com/docs/editor/whyvscode

Ngo, D., Lee, G.-D., & Kang, B. (2019). Improved color attenuation prior for single-image haze removal. Applied Sciences, 9(19), 4011. https://doi.org/10.3390/app9194011

Parihar, A. S., Gupta, Y. K., Singodia, Y., Singh, V., & Singh, K. (2020). A comparative study of image dehazing algorithms. In 2020 5th International Conference on Communication and Electronics Systems (ICCES) (pp. 766-771). IEEE. https://doi.org/10.1109/ICCES48766.2020.9138037

Park, J., Han, D. K., & Ko, H. (2020). Fusion of heterogeneous adversarial networks for single image dehazing. IEEE Transactions on Image Processing, 29, 4721-4732. https://doi.org/10.1109/TIP.2020.2975986

Paul, E. S., & Anitha, J. (2019). Analysis of Transform-Based Compression Techniques for MRI and CT Images. In D. Jude Hemanth, D. Gupta, V. E. Balas (Eds.), Intelligent Data Analysis for Biomedical Applications (pp. 103-120). Academic Press. ISBN 9780128155530. https://doi.org/10.1016/B978-0-12-815553-0.00005-7.

Peak Signal-to-Noise Ratio as an Image Quality Metric. (2023, March 23). NI. https://www.ni.com/en-ph/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/peak-signal-to-noise-ratio-as-an-image-quality-metric.html

Python Tutorial. (2022, August 16). What is NumPy. Python Tutorial - Master Python Programming for Beginners From Scratch. https://www.pythontutorial.net/python-numpy/what-is-numpy/

Rahman, S. H. A. ., & Ngadiran, R. . (2022). A Design of Dark Channel Prior for Image Dehazing. Indonesian Journal of Advanced Computer Science and Information Technology, 3(3). Retrieved from https://www.amcs-press.com/index.php/ijacsit/article/view/500

Saini, T. K. (2023). Re: What are some of the ethical considerations that arise when using image classification models in real-world applications? ResearchGate. https://www.researchgate.net/post/What_are_some_of_the_ethical_considerations_that_arise_when_using_image_classification_models_in_real-world_applications

Salazar-Colores, S., Pedraza-Ortega, J., & Rodríguez‑Reséndiz, J. (2019). Efficient single image dehazing by modifying the dark channel prior. Eurasip Journal on Image and Video Processing, 2019(1). https://doi.org/10.1186/s13640-019-0447-2

Sara, U. , Akter, M. and Uddin, M. (2019) Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. Journal of Computer and Communications, 7, 8-18. doi: 10.4236/jcc.2019.73002.

Shen, H., Zhong, T., Jia, Y., & Wu, C. (2024). Remote sensing image dehazing using generative adversarial network with texture and color space enhancement. Scientific Reports, 14. https://doi.org/10.1038/s41598-024-63259-6

Sun, C., Ding, Z., & Ma, L. (2023). Optimized method for polarization-based image dehazing. Heliyon, 9(5), e15849. https://doi.org/10.1016/j.heliyon.2023.e15849

Tarel, J.-P., Hautière, N., Caraffa, L., Cord, A., Halmaoui, H., & Gruyer, D. (2012). Vision enhancement in homogeneous and heterogeneous fog. *IEEE Intelligent Transportation Systems Magazine, 4*(2), 6-20.

Wang, C., Ding, M., Zhang, Y., & Wang, L. (2021). *A Single Image Enhancement Technique Using Dark Channel Prior. Applied Sciences,* 11(6), 2712. https://doi.org/10.3390/app11062712

What is Video Surveillance? (2023, October 6). Isarsoft. Retrieved November 15, 2023, from https://www.isarsoft.com/knowledge-hub/video-surveillance

What is a Video Surveillance System? | Senstar. (2023, January 27). Senstar. https://senstar.com/senstarpedia/what-is-a-video-surveillance-system/

Wu, L., Chen, J., Chen, S., Yang, X., Xu, L., Zhang, Y., & Zhang, J. (2023). Hybrid Dark Channel Prior for Image Dehazing Based on Transmittance Estimation by Variant Genetic Algorithm. Applied Sciences, 13(8), 4825. https://doi.org/10.3390/app13084825

Yang, Y., Zhang, J., Wang, Z., & Zhang, H. (2023). Single image fast dehazing based on haze density classification prior. Expert Systems with Applications, 232(C), December 2023, 120777. https://doi.org/10.1016/j.eswa.2023.120777.

Yang, G., & Evans, A. (2021). Improved single image dehazing methods for resource-constrained platforms. *Journal of Real-time Image Processing*, *18*(6), 2511–2525. https://doi.org/10.1007/s11554-021-01143-6

Yang, Y., Long, W., Li, Y., Shi, X., & Gao, L. (2021). Image defogging based on amended dark channel prior and 4‐directional L 1 regularisation. Iet Image Processing, 15(11), 2454–2477. https://doi.org/10.1049/ipr2.12233

Yuan, F., Zhou, Y., Xia, X., Qian, X., & Huang, J. (2021). A confidence prior for image dehazing. Pattern Recognition, 119, 108076. https://doi.org/10.1016/j.patcog.2021.108076

Zahra, G., Imran, M., Qahtani, A. M., Alsufyani, A., Almutiry, O., Mahmood, A., & Alazemi, F. (2021). Visibility Enhancement of scene images degraded by foggy weather condition: an application to video surveillance. Computers, Materials & Continua, 68(3), 3465–3481. https://doi.org/10.32604/cmc.2021.017454

Zhang, J., & Tao, D. (2020). FAMED-Net: A fast and accurate multi-scale end-to-end dehazing network. IEEE Transactions on Image Processing, 29, 72-84. https://doi.org/10.1109/TIP.2019.2922837

Zhao, L., Yin, Y., Zhong, T., & Jia, Y. (2023). Remote sensing image dehazing through an unsupervised generative adversarial network. Sensors, 23(17), 7484. https://doi.org/10.3390/s23177484

Zola, A. (2021). Python. WhatIs.com. https://www.techtarget.com/whatis/definition/Python

APPENDICES

Appendix 1

Gantt Chart

Activities	Months												
	2023								2024				
	May	June	July	August	September	October	November	December	January	February	March	April	May
Project Conceptualization													
Requirements Gathering													
Wireframing of Design													
Data Gathering													
Project Development (Desktop)													
Project Development (Mobile)													
Refining of Desktop and Mobile Applications													
Designing of Assessment and Evaluation													
Quality Assurance Testing													
Respondents Evaluation													
Application of Recommendations													
Deployment for Production													

Appendix 2

Budget Allocation

Particulars	Cost	Quantity	Amount
Manuscript	3399	6x	3399
Transportation	83	6x	500
TP-Link TC70 IP Camera	956	1	956
Mile Tech 800W Fog Machine	1,899	1	1,899
100mL Fog Liquid Solution	20	5	100
Lumber	210	1	210
Black Spray Paint	110	1	110
		TOTAL	**7174.00php**

Appendix 3

Test Instrument

Project Name:	SeeThrough: A Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method		
Written By:	Andre M. Nazareth	Alexandre F. Basiño	Ferdinand Carlo M. Palo
Written Date:	Sunday, April 7, 2024		
Test Case Description:	Compatibility Test		
Testers:	Test Respondent's Full Name with Signature		

Test Conditions		Date:	
Total Test Case:		Rating:	
Pass:			
Fail:			

Test Case Scenario ID	Test Case Scenario	Action	Expected Result	Pass	Fail	Comments/Suggestions
DESKTOP APPLICATION COMPATIBILITY						
STDA-WIN-10	The SeeThrough application can be installed and compatible in a Windows 10 version.	Install and Run the SeeThrough application that was installed to the computer on Windows 10 version.	SeeThrough Desktop Application should be accessible and well functioned, such as buttons & camera			
STDA-WIN-11	The SeeThrough application can be installed and compatible in a Windows 11 version.	Install and Run the SeeThrough application that was installed to the computer on Windows 11 version.	SeeThrough Desktop Application should be accessible and well functioned, such as buttons & camera			
STDA-RES	Check if the SeeThrough if the application in desktop is compatible in Resizing windows.	Resize the application to any desired screen size.	Layouts should be responsive and accessible			
MOBILE APPLICATION COMPATIBILITY						
STMA-VER-10	The SeeThrough Mobile application can be played on the device with the android version 10.	Tap the SeeThrough mobile application that was installed to the provided android phone.	SeeThrough Mobile Application should be accessible and well functioned, such as buttons & camera			
STMA-VER-11	The SeeThrough Mobile application can be played on the device with the android version 11.	Tap the SeeThrough mobile application that was installed to the provided android phone.	SeeThrough Mobile Application should be accessible and well functioned, such as buttons & camera			
STMA-VER-12	The SeeThrough Mobile application can be played on the device with the android version 12.	Tap the SeeThrough mobile application that was installed to the provided android phone.	SeeThrough Mobile Application should be accessible and well functioned, such as buttons & camera			
STMA-VER-13	The SeeThrough Mobile application can be played on the device with the android version 13.	Tap the SeeThrough mobile application that was installed to the provided android phone.	SeeThrough Mobile Application should be accessible and well functioned, such as buttons & camera			
STMA-SR 720x1280 (HD)	Check if the application is fit and exact in mobile device with 720x1280 screen resolution	Load the installed SeeThrough Mobile application	SeeThrough Mobile Application should be displayed smoothly in a 720x1280 screen resolution			
STMA-SR 1080x1920 (FHD)	Check if the application is fit and exact in mobile device with 1080x1920 screen resolution	Load the installed SeeThrough Mobile application	SeeThrough Mobile Application should be displayed smoothly in a 1080x1920 screen resolution			
STMA-SR 1440x2560 (QHD)	Check if the application is fit and exact in mobile device with 1440x2560 screen resolution	Load the installed SeeThrough Mobile application	SeeThrough Mobile Application should be displayed smoothly in a 1440x2560 screen resolution			

Project Name:	SeeThrough: A Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method
Written By:	Andre M. Nazareth · Alexandre F. Basiño · Ferdinand Carlo M. Palo
Written Date:	Sunday, April 7, 2024
Test Case Description:	Functionality Test
Testers:	Test Respondent's Full Name with Signature

Test Conditions		Date:	
Total Test Case:		Rating:	
Pass:			
Fail:			

Test Case Scenario ID	Name of the Module Function	Test Case Scenario	Action	Actual Input	Pass	Fail	Comments/Suggestions
DESKTOP APPLICATION							
STDA-INS	**Installation**	The installer will pop-up in the desktop and will be asked on where to install the application.	The user will open the installer file.	Install the SeeThrough Application			
DESKTOP APPLICATION NAVIGATION							
STDA-NAV-1	Real-Time Module	The application will popup and there will be Navigation for Real-Time Module on the bottom center of the application.	The user will navigate through Real-Time dehazing page.	Click the Real-Time Dehazing button			
STDA-NAV-2	Image Module	The application will popup and there will be Navigation for Image Module on the bottom center of the application.	The user will navigate through Image dehazing module.	Click the Image Dehazing button			
STDA-NAV-3	Video Input Module	The application will popup and there will be Navigation for Video Input Module on the bottom center of the application.	The user will receive a pop-up for video input.	Click the Video Input button			
REAL-TIME MODULE							
STDA-RT-1A	RTSP	The user will be shown an RTSP input popup	Under the Real-Time navigation, the user will click the settings icon.	Click Settings icon button			
STDA-RT-1B	RTSP	The user can input an RTSP URL	Upon showing the popup, the user will type the RTSP URL	Input RTSP URL			
STDA-RT-1C	RTSP	The user will be shown an alert box upon confirming the input of RTSP URL	Upon entering the RTSP URL, a popup will say that the RTSP URL is saved.	Click "OK" button			
STDA-RT-2	Start Live	The user will be able to connect the RTSP address to start the Live Feed on the camera through the application.	The user will click the Start button under the Real-Time page.	Click Start Button			
STDA-RT-3	Stop Live	The user will be able to stop the live through the application.	The user will click the Stop button under Real-Time page.	Click Stop Button			
IMAGE DEHAZING MODULE							
STDA-IMG-1	Load Image	The user will be able to upload an image on the application	The user will click the Load Image button under the Image Dehazing Tab	Upload a hazy Image			
STDA-IMG-2	Save Image	The user will be able to save the dehazed image.	The user will click the Save Image button under the Image Dehazing Tab	Save the Image by clicking the Save Button			
STDA-IMG-3	Dehazed Image	The user will be able to see the dehazed image after the user loads the image.	The user will go to the file path where the image is saved.	Navigate to the saved path.			
VIDEO DEHAZING MODULE							
STDA-VID-1	Upload Video	The user will be able to upload a video for dehazing	The user will click the Video Module on the	Upload a hazy video			

			bottom center of the application.				
STDA-VID-2	Save Video	The user will be able to save the video after the pop-up prompt.	The user will be asked where to save the video.	Save the dehazed video			
STDA-VID-3	Dehazed Video	The user will be able to see the saved dehazed file from the computer	The user will go to the location where the video is saved.	Play the saved Dehazed Video			
MOBILE DEHAZING PERMISSION							
STMA-PERM-1	Camera Permission	Upon opening the Live Page of the application, it will ask permission to use the Camera.	The application will ask permission to use the Camera	The user will tap "OK" option			
MOBILE DEHAZING NAVIGATION							
STMA-NAV-1	Dehaze Camera	The mobile application will run and there will be button for Live Module.	The user will tap the Dehaze Camera button.	Tap Dehaze Camera button			
STMA-NAV-2	Image Module	The mobile application will run and there will be button for Image Module.	The user will tap the Image button.	Tap Image button			
MOBILE DEHAZING DEHAZING CAMERA							
STMA-DC-1	Dehaze Camera Switch	The application will navigate to the Dehaze Camera page and will turn the dehazing switch "ON" or "OFF"	The user will tap the Dehaze switch at the bottom of the application to turn ON and OFF the dehazing enhancement.	Tap Dehazing Switch button			
STMA-DC-2	Capture Image while the Dehaze Switch is OFF	The user would be able to capture an image while the Dehaze switch is OFF.	The user will tap the capture button and go to the Gallery folder on the phone.	Tap Capture Image button while Dehaze Enhancement is OFF			
STMA-DC-3	Capture Image while the Dehaze Switch is ON	The user would be able to capture an image while the Dehaze switch is ON.	The user will tap the capture button and go to the Gallery folder on the phone.	Tap Capture Image button while Dehaze Enhancement is ON			

MOBILE DEHAZING IMAGE MODULE							
STMA-IMG-1	Load Image	The user should be able load an image upon tapping the "Load" button on Image Dehazing module.	The user will tap the Load button and would redirect the user in File Input window.	Tap Load Image button			
STMA-IMG-2	Dehaze Switch	Upon loading an image, the user should be able to turn ON/OFF the dehazing switch	The user will tap the Dehaze Switch button to apply dehazing enhancement.	Tap Dehaze Switch			
STMA-IMG-3	Save Image	The saved image file should have a dehazing enhancement applied	The user will click the save button and go to the gallery where the dehazed image is saved.	Tap Save Image button			

Appendix 4

Evaluation Instrument

SeeThrough Dehazing System – ISO/IEC 25010:2023
Systems and Software Quality Requirements and Evaluation (SQuaRE)

Part 1. Introduction

Greetings! We are computer science students from Lyceum of the Philippines University Cavite, currently undertaking our undergraduate thesis titled "SeeThrough: A Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method".

The purpose of this form is for respondents to evaluate the dehazing system's acceptability based on the ISO/IEC 25010:2023 criteria. Data obtained from this survey will be used for academic purposes only and any personal details will be kept confidential. If you agree to participate in the survey, you may withdraw at any time.

Part 2. Consent

Do you give your consent to participate in the research study titled "SeeThrough: A Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method"?

◯ *"I consent to participate in this study, and I understand that I may withdraw at any time."*

Part 3. Personal Information

Direction: Please provide the information requested below by writing on the space or by checking the appropriate box ( ) which corresponds to your profile.

First Name	
Last Name	
Age	
Gender	
Occupation	
City	
Type of Respondent	IT Expert End-User

Part 4. Evaluation

Direction: Please accomplish this questionnaire carefully, honestly, and accurately as you can. Indicate your assessment on the software by checking the box ( ) based on the given rating scale:

5 = Highly Acceptable 4 = Acceptable 3 = Moderately Acceptable 2 = Fairly Acceptable

1 = Poorly Acceptable

A. Functional Suitability	**5**	**4**	**3**	**2**	**1**
1. Functional Completeness – The dehazing system provides a set of dehazing features that covers all the specified tasks and intended users' objectives					
2. Functional Correctness – The dehazing system provides accurate dehazing results when used					
3. Functional Appropriateness – The dehazing system provide functions that facilitate the accomplishment of specified tasks and objectives					
B. Performance Efficiency					
4. Time Behavior – The dehazing system performs its functions with acceptable response time and frames per second (FPS)					
5. Resource Utilization – The dehazing system effectively utilizes the computer system's resources					
6. Capacity – The dehazing system has the ability to set and identify the limit of a function (e.g., number of image/footages that can be dehazed)					
C. Compatibility					
7. Co-Existence – The dehazing system performs efficiently while sharing a common environment and resources with other existing systems					
8. Interoperability – The dehazing system can exchange information with other products and use the information that has been acquired					
D. Interaction Capability					
9. Appropriateness Recognizability – The dehazing system is appropriate for the user's needs					
10. Learnability – The dehazing system is easy to learn					
11. Operability – The dehazing system is easy to operate and control					
12. User Error Protection – The dehazing system protects users against making errors					
13. User Engagement – The dehazing system presents information in an inviting and motivating manner, encouraging continued interaction					
14. Inclusivity – The dehazing system can be used by people of various backgrounds (e.g., age, educational attainment, culture, ethnicity, gender, etc.)					
15. User Assistance – The dehazing system can be used by people with different characteristics and capabilities					
16. Self-Descriptiveness – The dehazing system presents appropriate information to make its capabilities and use immediately obvious					

E. Reliability	**5**	**4**	**3**	**2**	**1**
17. Faultlessness – The dehazing system performs its functions without fault/error under normal operation					
18. Availability – The dehazing system is operational and accessible when needed					
19. Fault Tolerance – The dehazing system operates as intended despite the presence of hardware or software faults					
20. Recoverability – In the event of an interruption or failure, the dehazing system can recover the data directly affected and re-establish the desired state of the system					
F. Maintainability					
21. Modularity – Changes to one component does not affect other components					
22. Analyzability – The data from the dehazing system can be easily assessed for system changes or diagnosed for causes of failure.					
23. Modifiability – The dehazing system can be easily modified without introducing defects or degrading the product quality					
24. Testability – The dehazing system enables an objective and feasible test to be designed and performed to determine whether a requirement is met					
G. Flexibility					
25. Adaptability – The dehazing system can be used in different system environments					
26. Scalability – The dehazing system can handle growing or shrinking workloads, or adapt its capacity to handle variability					
27. Installability – The dehazing system can be successfully installed or uninstalled					

Part 5. Suggestions/Recommendations

Suggestions and recommendations for further improvement of the project:

SeeThrough Dehazing System – Core App Evaluation

Android Core App Quality Standard

Part 1. Introduction

Greetings! We are computer science students from Lyceum of the Philippines University Cavite, currently undertaking our undergraduate thesis titled "SeeThrough: A Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method".

The purpose of this form is for respondents to evaluate the dehazing system's acceptability based on the Android Core App Quality criteria. Data obtained from this survey will be used for academic purposes only and any personal details will be kept confidential. If you agree to participate in the survey, you may withdraw at any time.

Part 2. Consent

Do you give your consent to participate in the research study titled "SeeThrough: A Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method"?

◯ *"I consent to participate in this study, and I understand that I may withdraw at any time."*

Part 3. Personal Information

Direction: Please provide the information requested below by writing on the space or by checking the appropriate box ( ) which corresponds to your profile.

First Name	
Last Name	
Age	
Gender	
Occupation	
City	
Type of Respondent	IT Expert End-User

Part 4. Evaluation

Direction: Please accomplish this questionnaire carefully, honestly, and accurately as you can. Indicate your assessment on the software by checking the box ( ) based on the given rating scale:

5 = Highly Acceptable 4 = Acceptable 3 = Moderately Acceptable 2 = Fairly Acceptable

1 = Poorly Acceptable

A. Navigation	**5**	**4**	**3**	**2**	**1**
1. The application support back button navigations and functions correctly.					
2. The application support gesture navigation for going back or going to the home screen.					
3. The application preserves information when minimizing the app or changing the states.					
B. UI & Graphics					
4. The supports landscape and portrait orientations. Orientations and fold states expose essentially the same features and actions and preserve functional parity.					
5. The application fills the whole section of the device.					
C. Visual Quality					
6. The application displays graphics, text, images, and other UI elements without noticeable distortion, blurring, or pixelation.					
7. The application displays text and text blocks in an acceptable manner for each of the app's supported languages.					
8. The application supports dark theme.					
D. Accessibility					
9. Touch targets like buttons/switches are readable, clickable, and pleasing to look.					
10. The application's texts and background color maintain enough color contrast ratio to avoid eye strains and fatigue.					
E. Stability	**5**	**4**	**3**	**2**	**1**
11. The application does not crash or prompt "Android Not Responding" errors.					
F. Performance					
12. The application loads quickly upon opening.					
13. The application loads pages or layout in a timely manner.					
G. Permissions					
14. The application does request for permission to access the Camera.					

15. The application clearly conveys why certain permissions are needed.					
16. The application respects the decision if the user denied such permissions.					
H. Data and Files					
17. The image is saved on the screenshot folder found on device's internal storage.					
18. The application did not request for the user's sensitive data.					

Part 5. Suggestions/Recommendations

Suggestions and recommendations for further improvement of the project:

____

____

____

Appendix 5

Summary of Test and Evaluation Results

SeeThrough: A Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method

SeeThrough Dehazing System – Core App Evaluation
Android Core App Quality Standard

Table 1.1
Type of User

TYPE OF USER	FREQUENCY	PERCENTAGE
IT Expert	12	25.00
End-user	36	75.00
TOTAL	**48**	**100.00**

Table 1.2.1
Mean Response on Core App Evaluation in terms of Navigation (End-User)

FACTORS/INDICATORS	MEAN	STANDARD DEVIATION	INTERPRETATION
Navigation			
The application support back button navigations and functions correctly.	4.75	0.44	Highly Acceptable
The application support gesture navigation for going back or going to the home screen.	4.72	0.45	Highly Acceptable
The application preserves information when minimizing the app or changing the states.	4.61	0.49	Highly Acceptable
GRAND MEAN	**4.69**	**Highly Acceptable**	

Table 1.2.2
Mean Response on Core App Evaluation in terms of Navigation (IT Expert)

FACTORS/INDICATORS	MEAN	STANDARD DEVIATION	INTERPRETATION
Navigation			
The application support back button navigations and functions correctly.	4.58	0.51	Highly Acceptable
The application support gesture navigation for going back or going to the home screen.	4.33	0.89	Highly Acceptable
The application preserves information when minimizing the app or changing the states.	4.50	0.52	Highly Acceptable
GRAND MEAN	**4.47**	**Highly Acceptable**	

Table 1.3.1
Mean Response on Core App Evaluation in terms of UI & Graphics (End-User)

FACTORS/INDICATORS	MEAN	STANDARD DEVIATION	INTERPRETATION
UI & Graphics			
The application supports landscape and portrait orientations. Orientations and fold states expose essentially the same features and actions and preserve functional parity.	4.53	0.51	Highly Acceptable
The application fills the whole section of the device.	4.67	0.48	Highly Acceptable
GRAND MEAN	**4.60**	**Highly Acceptable**	

Table 1.3.2

Mean Response on Core App Evaluation in terms of UI & Graphics (IT Expert)

FACTORS/INDICATORS UI & Graphics	MEAN	STANDARD DEVIATION	INTERPRETATION
The application supports landscape and portrait orientations. Orientations and fold states expose essentially the same features and actions and preserve functional parity.	3.83	1.19	Acceptable
The application fills the whole section of the device.	4.58	0.51	Highly Acceptable
GRAND MEAN	**4.21**	**Highly Acceptable**	

Table 1.4.1

Mean Response on Core App Evaluation in terms of Visual Quality (End-User)

FACTORS/INDICATORS Visual Quality	MEAN	STANDARD DEVIATION	INTERPRETATION
The application displays graphics, text, images, and other UI elements without noticeable distortion, blurring, or pixelation.	4.64	0.54	Highly Acceptable
The application displays text and text blocks in an 3 manner for each of the app's supported languages.	4.53	0.51	Highly Acceptable
The application supports dark theme.	4.64	0.59	Highly Acceptable
GRAND MEAN	**4.60**	**Highly Acceptable**	

Table 1.4.2

Mean Response on Core App Evaluation in terms of Visual Quality (IT Expert)

FACTORS/INDICATORS Visual Quality	MEAN	STANDARD DEVIATION	INTERPRETATION
The application displays graphics, text, images, and other UI elements without noticeable distortion, blurring, or pixelation.	4.08	0.90	Acceptable
The application displays text and text blocks in an 3 manner for each of the app's supported languages.	4.42	0.67	Highly Acceptable
The application supports dark theme.	4.50	0.52	Highly Acceptable
GRAND MEAN	**4.33**	**Highly Acceptable**	

Table 1.5.1

Mean Response on Core App Evaluation in terms of Accessibility (End-User)

FACTORS/INDICATORS Accessibility	MEAN	STANDARD DEVIATION	INTERPRETATION
Touch targets like buttons/switches are readable, clickable, and pleasing to look.	4.69	0.47	Highly Acceptable
The application's texts and background color maintain enough color contrast ratio to avoid eye strains and fatigue.	4.67	0.48	Highly Acceptable
GRAND MEAN	**4.68**	**Highly Acceptable**	

Table 1.5.2

Mean Response on Core App Evaluation in terms of Accessibility (IT Expert)

FACTORS/INDICATORS Accessibility	MEAN	STANDARD DEVIATION	INTERPRETATION
Touch targets like buttons/switches are readable, clickable, and pleasing to look.	4.58	0.51	Highly Acceptable
The application's texts and background color maintain enough color contrast ratio to avoid eye strains and fatigue.	4.33	0.49	Highly Acceptable
GRAND MEAN	**4.46**	**Highly Acceptable**	

Table 1.6.1

Mean Response on Core App Evaluation in terms of Stability (End-User)

FACTORS/INDICATORS Stability	MEAN	STANDARD DEVIATION	INTERPRETATION
The application does not crash or prompt "Android Not Responding" errors.	4.75	0.50	Highly Acceptable

Table 1.6.2

Mean Response on Core App Evaluation in terms of Stability (IT Expert)

FACTORS/INDICATORS Stability	MEAN	STANDARD DEVIATION	INTERPRETATION
The application does not crash or prompt "Android Not Responding" errors.	4.58	0.51	Highly Acceptable

Table 1.7.1

Mean Response on Core App Evaluation in terms of Performance (End-User)

FACTORS/INDICATORS Performance	MEAN	STANDARD DEVIATION	INTERPRETATION
The application loads quickly upon opening.	4.61	0.49	Highly Acceptable
The application loads pages or layout in a timely manner.	4.61	0.55	Highly Acceptable
GRAND MEAN	**4.61**	**Highly Acceptable**	

Table 1.7.2

Mean Response on Core App Evaluation in terms of Performance (IT Expert)

FACTORS/INDICATORS Performance	MEAN	STANDARD DEVIATION	INTERPRETATION
The application loads quickly upon opening.	4.50	0.52	Highly Acceptable
The application loads pages or layout in a timely manner.	4.58	0.51	Highly Acceptable
GRAND MEAN	**4.54**	**Highly Acceptable**	

Table 1.8.1
Mean Response on Core App Evaluation in terms of Permissions (End-User)

FACTORS/INDICATORS Permissions	MEAN	STANDARD DEVIATION	INTERPRETATION
The application does request for permission to access the Camera.	4.75	0.44	Highly Acceptable
The application clearly conveys why certain permissions are needed.	4.67	0.48	Highly Acceptable
The application respects the decision if the user denied such permissions.	4.67	0.48	Highly Acceptable
GRAND MEAN	**4.69**	**Highly Acceptable**	

Table 1.8.2
Mean Response on Core App Evaluation in terms of Permissions (IT Expert)

FACTORS/INDICATORS Permissions	MEAN	STANDARD DEVIATION	INTERPRETATION
The application does request for permission to access the Camera.	4.67	0.49	Highly Acceptable
The application clearly conveys why certain permissions are needed.	3.75	1.22	Acceptable
The application respects the decision if the user denied such permissions.	4.25	0.62	Highly Acceptable
GRAND MEAN	**4.22**	**Highly Acceptable**	

Table 1.9.1
Mean Response on Core App Evaluation in terms of Data and Files (End-User)

FACTORS/INDICATORS Data and Files	MEAN	STANDARD DEVIATION	INTERPRETATION
The image is saved on the screenshot folder found on device's internal storage.	4.64	0.49	Highly Acceptable
The application did not request for the user's sensitive data.	4.61	0.77	Highly Acceptable
GRAND MEAN	**4.63**	**Highly Acceptable**	

Table 1.9.2
Mean Response on Core App Evaluation in terms of Data and Files (IT Expert)

FACTORS/INDICATORS Data and Files	MEAN	STANDARD DEVIATION	INTERPRETATION
The image is saved on the screenshot folder found on device's internal storage.	4.42	0.51	Highly Acceptable
The application did not request for the user's sensitive data.	4.25	1.14	Highly Acceptable
GRAND MEAN	**4.33**	**Highly Acceptable**	

SeeThrough Dehazing System – ISO/IEC 25010:2023
Systems and Software Quality Requirements and Evaluation (SQuaRE)

Table 2.1.1
Mean Response on Systems and Software Quality Requirements Evaluation in terms of Functional Suitability (End-User)

FACTORS/INDICATORS Functional Suitability	MEAN	STANDARD DEVIATION	INTERPRETATION
Functional Completeness - The dehazing system provides a set of functions that covers all the specified tasks and intended users' objectives	4.61	0.77	Highly Acceptable
Functional Correctness - The dehazing system provides accurate results when used	4.61	0.60	Highly Acceptable
Functional Appropriateness - The dehazing system provide functions that facilitate the accomplishment of specified tasks and objectives	4.69	0.52	Highly Acceptable
GRAND MEAN	**4.64**	**Highly Acceptable**	

Table 2.1.2
Mean Response on Systems and Software Quality Requirements Evaluation in terms of Functional Suitability (IT Expert)

FACTORS/INDICATORS Functional Suitability	MEAN	STANDARD DEVIATION	INTERPRETATION
Functional Completeness - The dehazing system provides a set of functions that covers all the specified tasks and intended users' objectives	4.67	0.49	Highly Acceptable
Functional Correctness - The dehazing system provides accurate results when used	4.17	0.72	Acceptable
Functional Appropriateness - The dehazing system provide functions that facilitate the accomplishment of specified tasks and objectives	4.50	0.52	Highly Acceptable
GRAND MEAN	**4.44**	**Highly Acceptable**	

Table 2.2.1
Mean Response on Systems and Software Quality Requirements Evaluation in terms of Performance Efficiency (End-User)

FACTORS/INDICATORS Performance Efficiency	MEAN	STANDARD DEVIATION	INTERPRETATION
Time Behavior - The dehazing system performs its functions under specified conditions so that the response time and throughput rates meet the requirements.	4.56	0.61	Highly Acceptable
Resource Utilization - The dehazing system uses no more than the specified amount of resources to perform its function under specified conditions.	4.61	0.55	Highly Acceptable
Capacity - The dehazing system meets requirements for the maximum limits of a product parameter (e.g. number of image/footage that can be dehazed)	4.61	0.49	Highly Acceptable
GRAND MEAN	**4.59**	**Highly Acceptable**	

Table 2.2.2

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Performance Efficiency (IT Expert)

FACTORS/INDICATORS Performance Efficiency	MEAN	STANDARD DEVIATION	INTERPRETATION
Time Behavior - The dehazing system performs its functions under specified conditions so that the response time and throughput rates meet the requirements.	4.42	0.67	Highly Acceptable
Resource Utilization - The dehazing system uses no more than the specified amount of resources to perform its function under specified conditions.	4.58	0.67	Highly Acceptable
Capacity - The dehazing system meets requirements for the maximum limits of a product parameter (e.g. number of image/footage that can be dehazed)	4.42	0.51	Highly Acceptable
GRAND MEAN	**4.47**	**Highly Acceptable**	

Table 2.3.1

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Compatibility (End-User)

FACTORS/INDICATORS Compatibility	MEAN	STANDARD DEVIATION	INTERPRETATION
Co-existence - The dehazing system performs its required function efficiently while sharing a common environment and resources with other existing systems, without detrimental impact on any other ...	4.69	0.47	Highly Acceptable
Interoperability - The dehazing system exchanges information with other products and mutually use the information that has been acquired.	4.67	0.48	Highly Acceptable
GRAND MEAN	**4.68**	**Highly Acceptable**	

Table 2.3.2

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Compatibility (IT Expert)

FACTORS/INDICATORS Compatibility	MEAN	STANDARD DEVIATION	INTERPRETATION
Co-existence - The dehazing system performs its required function efficiently while sharing a common environment and resources with other existing systems, without detrimental impact on any other ...	4.50	0.52	Highly Acceptable
Interoperability - The dehazing system exchanges information with other products and mutually use the information that has been acquired.	4.25	0.75	Highly Acceptable
GRAND MEAN	**4.38**	**Highly Acceptable**	

Table 2.4.1

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Interaction Capability (End-user)

FACTORS/INDICATORS Interaction Capability	MEAN	STANDARD DEVIATION	INTERPRETATION
Appropriateness Recognizability - The dehazing system is recognized as appropriate for the user's needs	4.61	0.55	Highly Acceptable
Learnability - The user can learn the dehazing system's functions in an appropriate amount of time	4.75	0.50	Highly Acceptable
Operability - The dehazing system has functions and attributes that makes it easy to operate and control	4.78	0.42	Highly Acceptable
User Error Protection - The dehazing system prevents operation errors	4.44	0.65	Highly Acceptable
User Engagement - The dehazing system presents functions and information in an inviting and motivating manner, encouraging continued interaction	4.53	0.56	Highly Acceptable
Inclusivity - The dehazing system can be utilized by people of various backgrounds (e.g. age, education, culture, ethnicity, gender, etc.)	4.58	0.65	Highly Acceptable
User Assistance - The dehazing system can be used by people with the widest range of characteristics and capabilities to achieve specified goals in a specified context of use	4.58	0.60	Highly Acceptable
Self-descriptiveness - The dehazing system presents appropriate information, where needed by the user, to make its capabilities and use immediately obvious to the user without excessive interactio...	4.61	0.55	Highly Acceptable
GRAND MEAN	**4.61**	**Highly Acceptable**	

Table 2.4.2

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Interaction Capability (IT Expert)

FACTORS/INDICATORS Interaction Capability	MEAN	STANDARD DEVIATION	INTERPRETATION
Appropriateness Recognizability - The dehazing system is recognized as appropriate for the user's needs	4.50	0.67	Highly Acceptable
Learnability - The user can learn the dehazing system's functions in an appropriate amount of time	4.33	0.98	Highly Acceptable
Operability - The dehazing system has functions and attributes that makes it easy to operate and control	4.42	1.00	Highly Acceptable
User Error Protection - The dehazing system prevents operation errors	4.08	0.67	Acceptable
User Engagement - The dehazing system presents functions and information in an inviting and motivating manner, encouraging continued interaction	4.33	0.65	Highly Acceptable
Inclusivity - The dehazing system can be utilized by people of various backgrounds (e.g. age, education, culture, ethnicity, gender, etc.)	4.58	0.51	Highly Acceptable
User Assistance - The dehazing system can be used by people with the widest range of characteristics and capabilities to achieve specified goals in a specified context of use	4.25	1.06	Highly Acceptable
Self-descriptiveness - The dehazing system presents appropriate information, where needed by the user, to make its capabilities and use immediately obvious to the user without excessive interactio...	4.25	0.87	Highly Acceptable
GRAND MEAN	**4.34**	**Highly Acceptable**	

Table 2.5.1

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Reliability (End-User)

FACTORS/INDICATORS Reliability	MEAN	STANDARD DEVIATION	INTERPRETATION
Faultlessness - The dehazing system performs specified functions without fault/error under normal operation	4.42	0.77	Highly Acceptable
Availability - The dehazing system is operational and accessible when required for use	4.72	0.45	Highly Acceptable
Fault Tolerance - The dehazing system operates as intended despite the presence of hardware or software faults	4.58	0.60	Highly Acceptable
Recoverability - In the event of an interruption or a failure, the dehazing system can recover the data directly affected and re-establish the desired state of the system	4.53	0.74	Highly Acceptable
GRAND MEAN	**4.56**	**Highly Acceptable**	

Table 2.5.2

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Reliability (IT Expert)

FACTORS/INDICATORS Reliability	MEAN	STANDARD DEVIATION	INTERPRETATION
Faultlessness - The dehazing system performs specified functions without fault/error under normal operation	4.33	0.65	Highly Acceptable
Availability - The dehazing system is operational and accessible when required for use	4.33	0.65	Highly Acceptable
Fault Tolerance - The dehazing system operates as intended despite the presence of hardware or software faults	4.17	0.72	Acceptable
Recoverability - In the event of an interruption or a failure, the dehazing system can recover the data directly affected and re-establish the desired state of the system	4.00	0.85	Acceptable
GRAND MEAN	**4.21**	**Highly Acceptable**	

Table 2.6.1

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Maintainability (End-User)

FACTORS/INDICATORS Maintainability	MEAN	STANDARD DEVIATION	INTERPRETATION
Modularity - Changes to one component does not affect other components	4.58	0.55	Highly Acceptable
Analyzability - The data from the dehazing system can be easily assessed for system changes or diagnosed for causes of failure	4.67	0.53	Highly Acceptable
Modifiability - The dehazing system can be easily modified without introducing defects or degrading the product quality	4.31	0.98	Highly Acceptable
Testability - The dehazing system enables an objective and feasible test to be designed and performed to determine whether a requirement is met	4.69	0.47	Highly Acceptable
GRAND MEAN	**4.56**	**Highly Acceptable**	

Table 2.6.2

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Maintainability (IT Expert)

FACTORS/INDICATORS Maintainability	MEAN	STANDARD DEVIATION	INTERPRETATION
Modularity - Changes to one component does not affect other components	4.50	0.52	Highly Acceptable
Analyzability - The data from the dehazing system can be easily assessed for system changes or diagnosed for causes of failure	4.17	0.83	Acceptable
Modifiability - The dehazing system can be easily modified without introducing defects or degrading the product quality	4.42	0.67	Highly Acceptable
Testability - The dehazing system enables an objective and feasible test to be designed and performed to determine whether a requirement is met	4.50	0.67	Highly Acceptable
GRAND MEAN	**4.40**	**Highly Acceptable**	

Table 2.7.1

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Flexibility (End-User)

FACTORS/INDICATORS Flexibility	MEAN	STANDARD DEVIATION	INTERPRETATION
Adaptability - The dehazing system can be effectively and efficiently adapted for or transferred to different hardware, software or other operational or usage environments	4.64	0.59	Highly Acceptable
Scalability - The dehazing system can handle growing of shrinking workloads or can adapt its capacity to handle variability	4.64	0.54	Highly Acceptable
Installability - The dehazing system can be effectively and efficiently installed successfully and/or uninstalled in a specified environment	4.86	0.35	Highly Acceptable
GRAND MEAN	**4.71**	**Highly Acceptable**	

Table 2.7.2

Mean Response on Systems and Software Quality Requirements Evaluation in terms of Flexibility (IT Expert)

FACTORS/INDICATORS Flexibility	MEAN	STANDARD DEVIATION	INTERPRETATION
Adaptability - The dehazing system can be effectively and efficiently adapted for or transferred to different hardware, software or other operational or usage environments	4.42	0.90	Highly Acceptable
Scalability - The dehazing system can handle growing of shrinking workloads or can adapt its capacity to handle variability	4.17	0.72	Acceptable
Installability - The dehazing system can be effectively and efficiently installed successfully and/or uninstalled in a specified environment	4.17	0.94	Acceptable
GRAND MEAN	**4.25**	**Highly Acceptable**	

ENGR. JELYN M. RODRIGUEZ
Statistician
COECSA-DOE

Appendix 6

Endorsement Letter

April 25, 2024

Eulogio A. Malabanan
Barangay Hall of Kaybagal Central
Tagaytay City

Dear Sir Malabanan,

Greetings!
We, the 4th year students of Bachelor of Science in Computer Science from Lyceum of the Philippines University - Cavite, are currently working on a study entitled ***"SeeThrough: A Real-Time Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method"*** as part of our academic requirements for Thesis 2 subject.

Our software project can be used to enhance footages or images captured by a surveillance camera amidst low-visibility situations such as foggy or hazy weather conditions. In connection with this, our group would like to ask for 10-25 minutes of your time to allow us to present and demonstrate our software project with you and some residents from your barangay for your evaluation. Part of our study is to find out the acceptability of the developed software on the part of the end-user.

We are also willing to donate the system to your barangay if you would allow it.

If you have any questions or concerns, please contact me at mobile number 0927 854 1392.

Looking forward to your favorable response to this request.

Sincerely,

Mr. Andre Nazareth
Project Lead
BSCS401 Student

Jerian R. Peren
Research Adviser

April 25, 2024

Victor S. Bergado
Barangay Hall of Kaybagal North
Tagaytay City

Dear Sir Bergado,

Greetings!
We, the 4th year students of Bachelor of Science in Computer Science from Lyceum of the Philippines University - Cavite, are currently working on a study entitled ***"SeeThrough: A Real-Time Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method"*** as part of our academic requirements for Thesis 2 subject.

Our software project can be used to enhance footages or images captured by a surveillance camera amidst low-visibility situations such as foggy or hazy weather conditions. In connection with this, our group would like to ask for 10-25 minutes of your time to allow us to present and demonstrate our software project with you and some residents from your barangay for your evaluation. Part of our study is to find out the acceptability of the developed software on the part of the end-user.

We are also willing to donate the system to your barangay if you would allow it.

If you have any questions or concerns, please contact me at mobile number 0927 854 1392.

Looking forward to your favorable response to this request.

Sincerely,

Mr. Andre Nazareth
Project Lead
BSCS401 Student

Jerian R. Peren
Research Adviser

April 25, 2024

Fatima G. Guadaña
Barangay Hall of Kaybagal South
Tagaytay City

Dear Ma'am Guadaña,

Greetings!
We, the 4th year students of Bachelor of Science in Computer Science from Lyceum of the Philippines University - Cavite, are currently working on a study entitled ***"SeeThrough: A Real-Time Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method"*** as part of our academic requirements for Thesis 2 subject.

Our software project can be used to enhance footages or images captured by a surveillance camera amidst low-visibility situations such as foggy or hazy weather conditions. In connection with this, our group would like to ask for 10-25 minutes of your time to allow us to present and demonstrate our software project with you and some residents from your barangay for your evaluation. Part of our study is to find out the acceptability of the developed software on the part of the end-user.

We are also willing to donate the system to your barangay if you would allow it.

If you have any questions or concerns, please contact me at mobile number 0927 854 1392.

Looking forward to your favorable response to this request.

Sincerely,

Mr. Andre Nazareth
Project Lead
BSCS401 Student

Jerian R. Peren
Research Adviser

April 25, 2024

Bernardo B. Cabasi
Barangay Hall of Silang Crossing East
Tagaytay City

Dear Sir Cabasi,

Greetings!
We, the 4th year students of Bachelor of Science in Computer Science from Lyceum of the Philippines University - Cavite, are currently working on a study entitled ***"SeeThrough: A Real-Time Surveillance Dehazing System Using Dark Channel Prior Algorithm with Gaussian Filtering Method"*** as part of our academic requirements for Thesis 2 subject.

Our software project can be used to enhance footages or images captured by a surveillance camera amidst low-visibility situations such as foggy or hazy weather conditions. In connection with this, our group would like to ask for 10-25 minutes of your time to allow us to present and demonstrate our software project with you and some residents from your barangay for your evaluation. Part of our study is to find out the acceptability of the developed software on the part of the end-user.

We are also willing to donate the system to your barangay if you would allow it.

If you have any questions or concerns, please contact me at mobile number 0927 854 1392.

Looking forward to your favorable response to this request.

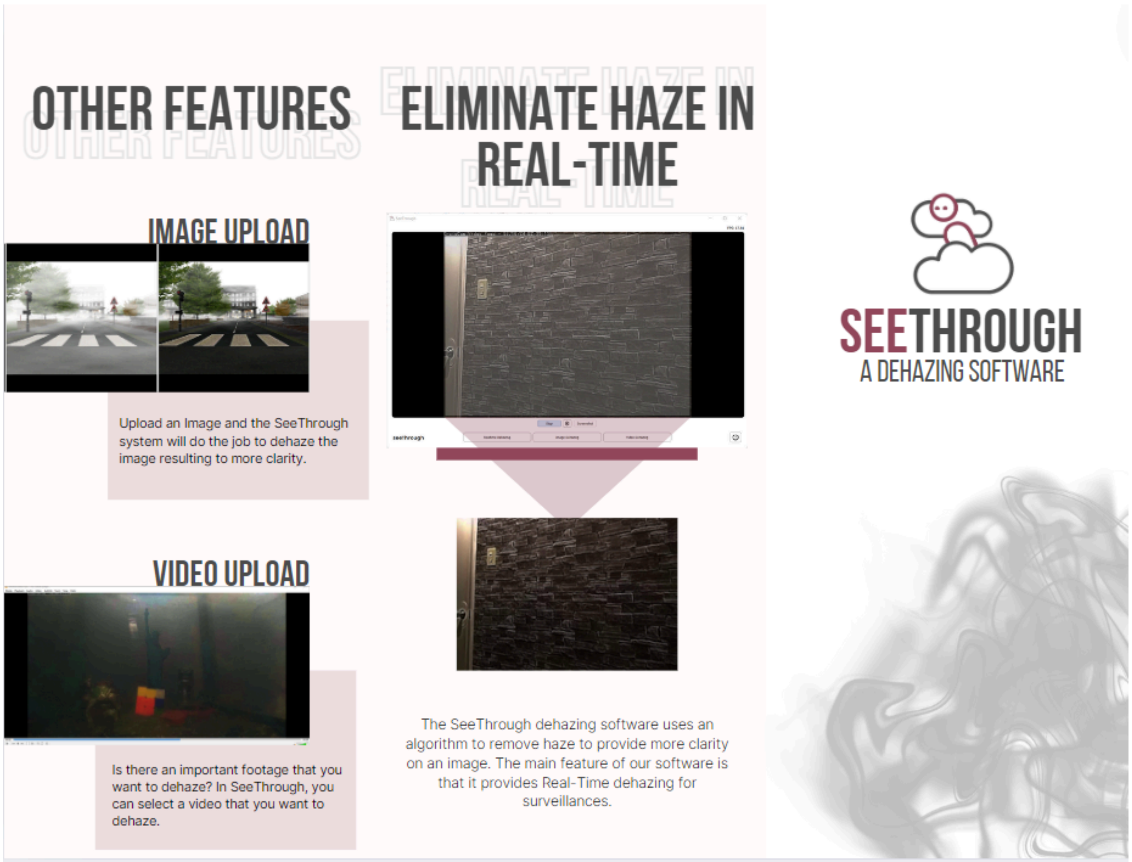
Sincerely,

Mr. Andre Nazareth
Project Lead
BSCS401 Student

Jerian R. Peren
Research Adviser

Appendix 7

Brochure


OTHER FEATURES
ELIMINATE HAZE IN REAL-TIME
IMAGE UPLOAD
Upload an Image and the SeeThrough system will do the job to dehaze the image resulting to more clarity.
VIDEO UPLOAD
Is there an important footage that you want to dehaze? In SeeThrough, you can select a video that you want to dehaze.
The SeeThrough dehazing software uses an algorithm to remove haze to provide more clarity on an image. The main feature of our software is that it provides Real-Time dehazing for surveillances.
SEETHROUGH
A DEHAZING SOFTWARE


ALSO IN MOBILE!
MOBILE FEATURES
LIVE DEHAZING
instead of surveillance, in the mobile version you can just use your camera to dehaze in real-time, how amazing that is!
IMAGE DEHAZING
things are more easier now that you can dehaze an image on the go with just your smartphone!
*The mobile version is only available in Android 10 above versions
ANDRE NAZARETH
LEADER
Andre Nazareth is the leader of SeeThrough Thesis project. He analyzed how the algorithm works for our system and oversees the progress of the project.
ALEXANDRE BASIÑO
BACKEND
Alexandre Basiño is the backend developer of the project. He is in-charge of how the algorithm integrates on our system and the process behind every feature of the system.
FERDINAND CARLO PALO
FRONTEND
Ferdinand Carlo Palo is the frontend developer of the project. He designed the system flow on how the user will interact on the system.
WHO ARE WE?
WHO ARE WE?
WHO ARE WE?
CONTACT US
(+63) 905+379+4008
ferdinand.palo@lpunetwork.edu.ph

Appendix 8

Appointment Of Research Advisor

College of Engineering, Computer Science and Architecture

Appointment of Research Advisor

October 27, 2023

Ms. Peren
Lyceum of the Philippines University
Gen. Trias, Cavite

Dear Ms. Peren,

Greetings in *Veritas et Fortitudo, Pro Deo et Patria!*

I am pleased to inform you that you have been appointed as Research Advisor of the research study/project entitled: SeeThrough: A Real-Time Surveillance Dehazing using Dark Channel Prior Algorithm with Gaussian Filtering Method with Andre M. Nazareth, Alexandre F. Basiño, and Ferdinand Carlo M. Palo as your advisee(s).

For your guidance, find attached list of duties and responsibilities.

Sincerely,

JERIAN R. PEREN

Chairperson's Signature over Encoded Name

Accepted:

JERIAN R. PEREN

Research Advisor's Signature over Encoded Name

Cc: Research, Publications and Innovation Center

Appendix 9

Relevant Source Code

Desktop Dark Channel Prior Source Code in Python

```
# DESKTOP DARK CHANNEL PRIOR ALGORITHM SOURCE CODE

class DehazingCPU(object):

    def DarkChannel(self, im, sz):

        b, g, r = cv2.split(im)

        dc = np.min(np.stack([r, g, b]), axis=0)

        kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (sz, sz))

        dark = cv2.erode(dc, kernel)

        return dark



    def EstimateA(self, img, darkChannel):

        h, w, _ = img.shape

        length = h * w

        num = max(int(length * 0.0001), 1)

        darkChannVec = darkChannel.reshape(length)

        index = np.argpartition(darkChannVec, -num)[-num:]

        coords = np.column_stack(np.unravel_index(index, (h, w)))
```

```
        A = np.mean(img[coords[:, 0], coords[:, 1], :], axis=0, keepdims=True)

        return A



    def TransmissionEstimate(self, im, A, sz):

        omega = 0.90

        im3 = im / A[0, :]

        transmission = 1 - omega * self.DarkChannel(im3, sz)

        return transmission



    def GaussianTransmissionRefine(self, et, sigma=2):

        return gaussian_filter(et, sigma=sigma)



    def Recover(self, im, t, A, tx=0.1):

        t = np.maximum(t, tx)

        t_broadcasted = np.expand_dims(t, axis=-1)

        res = (im - A) / t_broadcasted + A

        return res
```

```
    def image_processing(self, frame):
        I = frame.astype('float64') / 255
        dark = self.DarkChannel(I, 15)
        A = self.EstimateA(I, dark)
        te = self.TransmissionEstimate(I, A, 3)
        t = self.GaussianTransmissionRefine(te)
        J = self.Recover(I, t, A, 0.1)

        return J
# END OF DESKTOP DARK CHANNEL PRIOR CODE
```

Mobile Application Dark Channel Prior Source Code in Java

```
package com.seethrough.dehazing;



import org.opencv.core.*;

import org.opencv.imgproc.Imgproc;

import org.opencv.imgcodecs.Imgcodecs;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;



public class DarkChannelPrior {

    private static final int DARK_CHANNEL_SIZE = 12;

    private static final int TRANSMISSION_MAP_SIZE = 3;

    private static final double OMEGA = 0.90;

    private static final double MIN_TRANSMISSION = 0.1;

    private static final double MAX_TRANSMISSION = 1.0;

    private static final double ATMOSPHERIC_LIGHT_PERCENTILE = 0.001;
```

```
    public static Mat enhance(Mat image) {

        if (image == null) {

            throw new IllegalArgumentException("Input image cannot be null");

        }

        image.convertTo(image, CvType.CV_32F);

        Mat DarkChannel = darkChannel(image, DARK_CHANNEL_SIZE);

        double atmosphericLight = estimateAtmosphericLight(DarkChannel);

        Mat transmissionMap = estimateTransmissionMap(image, atmosphericLight);

        Mat refinedTransmissionMap = transmissionMapRefine(transmissionMap);

        Mat enhancedImage = recoverImage(image, refinedTransmissionMap,
atmosphericLight);



        enhancedImage.convertTo(enhancedImage, CvType.CV_8UC3);

        DarkChannel.release();

        transmissionMap.release();



        return enhancedImage;
```

```
    }



    private static Mat darkChannel(Mat image, int size) {
        List<Mat> channels = new ArrayList<>();
        Core.split(image, channels);
        Mat darkChannel = new Mat();
        Core.min(channels.get(0), channels.get(1), darkChannel);
        Core.min(darkChannel, channels.get(2), darkChannel);
        Mat kernel = Imgproc.getStructuringElement(Imgproc.MORPH_RECT, new
Size(size, size));
        Imgproc.erode(darkChannel, darkChannel, kernel);


        kernel.release();
        channels.forEach(Mat::release);
        return darkChannel;
    }
```

```
    private static double estimateAtmosphericLight(Mat darkChannel) {
        Mat sortedMinRGB = new Mat();
        Core.sort(darkChannel.reshape(1, 1), sortedMinRGB, Core.SORT_EVERY_ROW +
Core.SORT_ASCENDING);
        int topN = Math.max((int)(sortedMinRGB.total() *
ATMOSPHERIC_LIGHT_PERCENTILE), 1); // Top 0.1% brightest pixels, at least 1
        double sum = 0.0;
        for (int i = sortedMinRGB.cols() - topN; i < sortedMinRGB.cols(); i++) {
            sum += sortedMinRGB.get(0, i)[0];
        }
        double avgAtmosLight = sum / topN;
        sortedMinRGB.release();
        return avgAtmosLight;
    }


    private static Mat estimateTransmissionMap(Mat image, double A) {
        Mat im3 = new Mat();
```

```
        Core.divide(image, new Scalar(A, A, A, 0), im3); // Normalize each channel by A

        Mat darkChannel = darkChannel(im3, TRANSMISSION_MAP_SIZE);

        Mat transmission = new Mat();

        Core.multiply(darkChannel, new Scalar(-OMEGA), transmission);

        Core.add(transmission, new Scalar(MAX_TRANSMISSION), transmission);

        // Ensure transmission values are within a reasonable range

        Core.max(transmission, new Scalar(MIN_TRANSMISSION), transmission); //
Avoid too low values that could cause black pixels

        Core.min(transmission, new Scalar(MAX_TRANSMISSION), transmission); //
Ensure no value exceeds 1.0



        //clean up

        darkChannel.release();

        im3.release();

        return transmission;

    }

    private static Mat transmissionMapRefine(Mat transmission) {
```

```
        Mat refinedTransmission = new Mat();
        Imgproc.GaussianBlur(transmission, refinedTransmission, new
Size(TRANSMISSION_MAP_SIZE * 2 + 1, TRANSMISSION_MAP_SIZE * 2 + 1), 2);
        transmission.release();
        return refinedTransmission;
    }
    private static Mat recoverImage(Mat image, Mat t, double AtmosphericLight) {
        double tx = 0.1;
        Mat maxT = new Mat();
        Core.max(t, new Scalar(tx), maxT);
        List<Mat> channels = new ArrayList<>();
        Core.split(image, channels);
        List<Mat> recoveredChannels = new ArrayList<>();
        for (Mat channel : channels) {
            Mat recoveredChannel = new Mat();
            Core.subtract(channel, new Scalar(AtmosphericLight), recoveredChannel);
            Core.divide(recoveredChannel, maxT, recoveredChannel);
```

```
            Core.add(recoveredChannel, new Scalar(AtmosphericLight), recoveredChannel);
            recoveredChannels.add(recoveredChannel);
        }
        Mat result = new Mat();
        Core.merge(recoveredChannels, result);


        maxT.release();
        for (Mat channel : channels) {
            channel.release();
        }
        for (Mat channel : recoveredChannels) {
            channel.release();
        }
        return result;
    }
}
// END OF MOBILE DARK CHANNEL PRIOR CODE
```

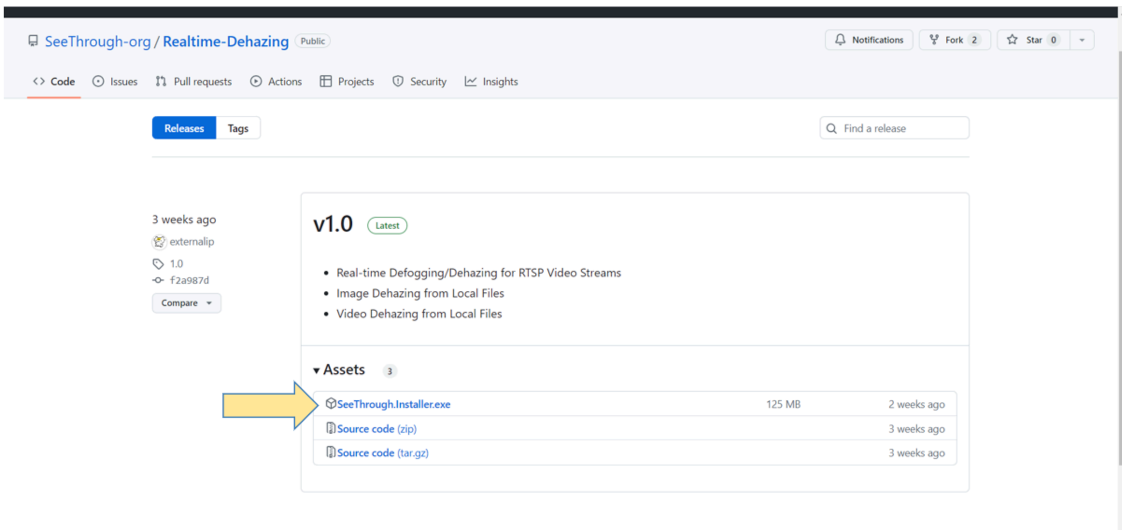
Appendix 10

User Guide

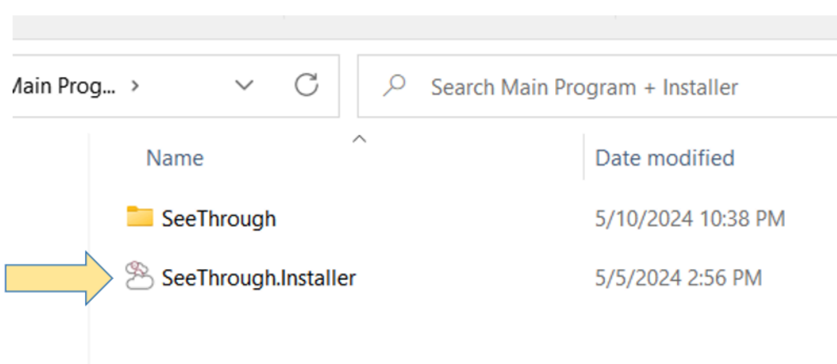
Desktop Application User Guides

Part 1. Installation

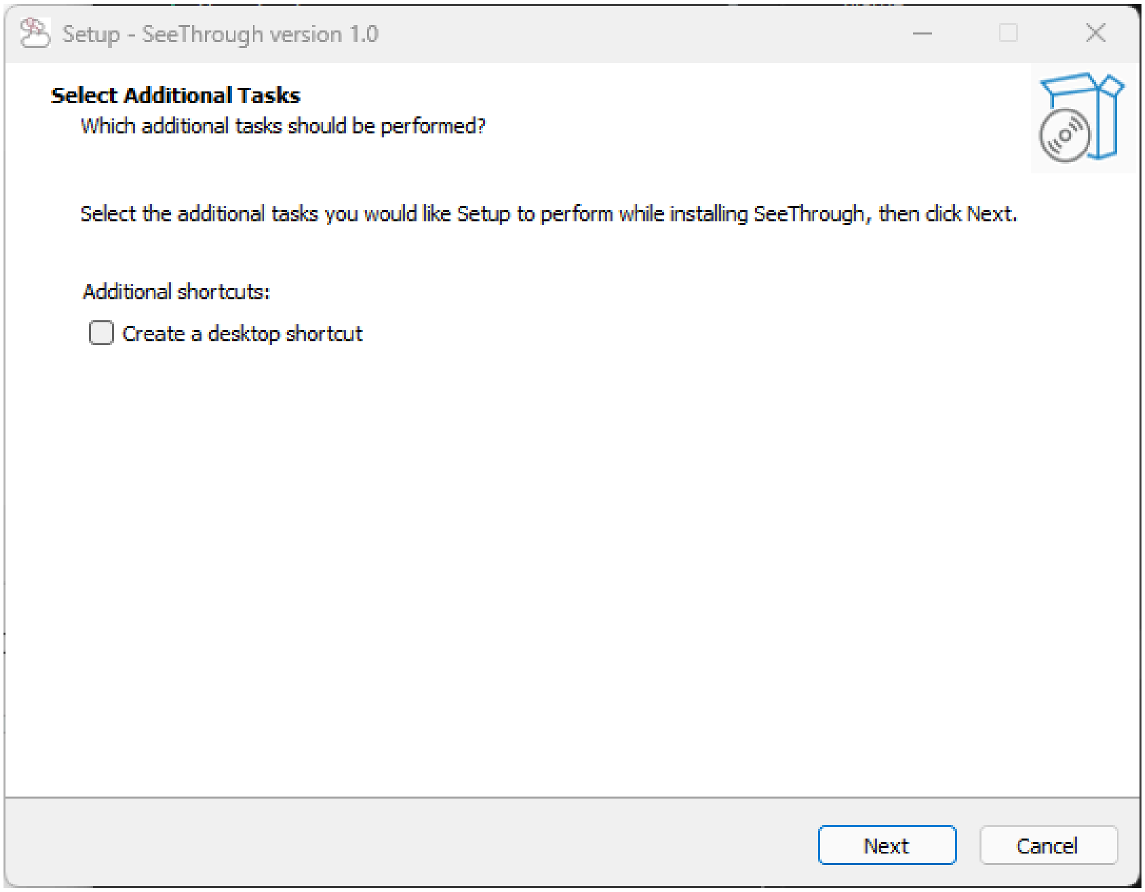
Go to this website https://github.com/SeeThrough-org/Realtime-Dehazing/releases and Download the .exe file installer where the orange arrow points in the below image and save it anywhere on the computer.

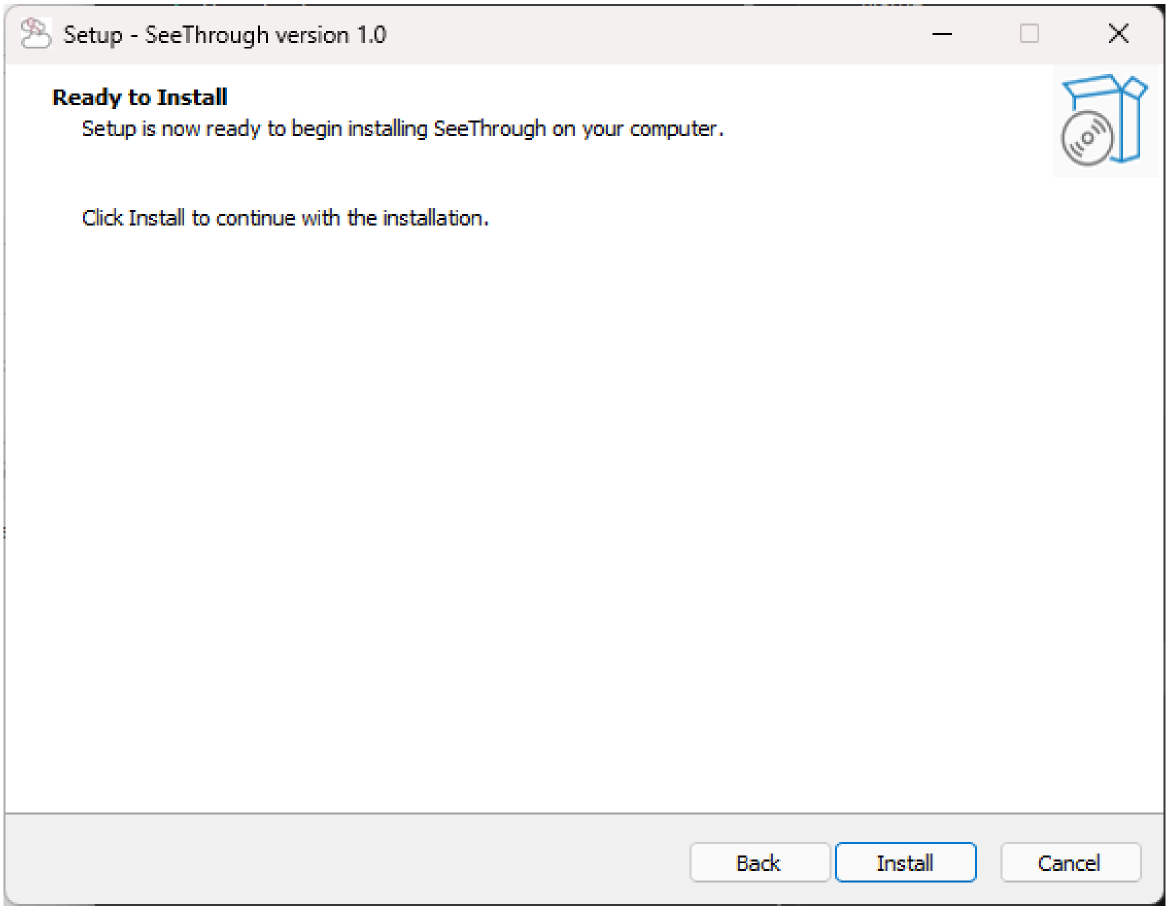


Once finished downloading the .exe file, click the installer.

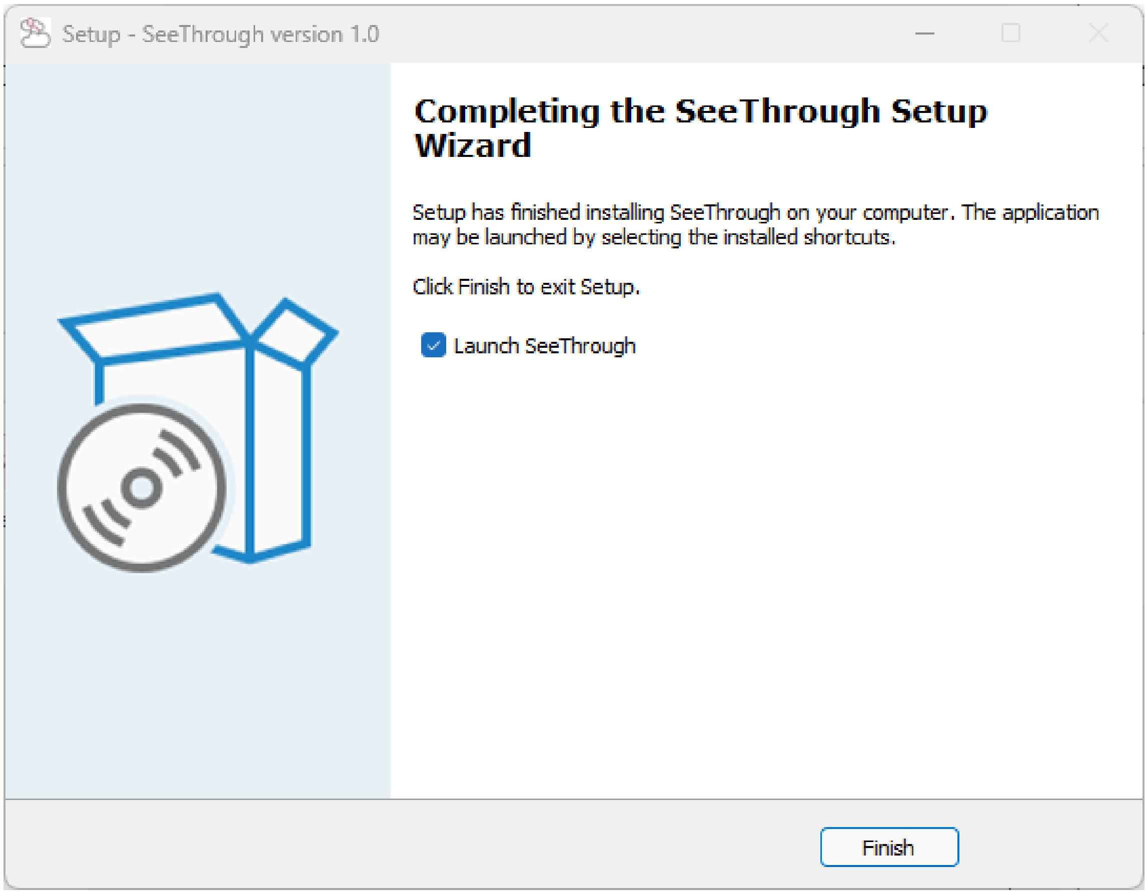


An installation pop-up window will appear and click “Next” button and “Install”.



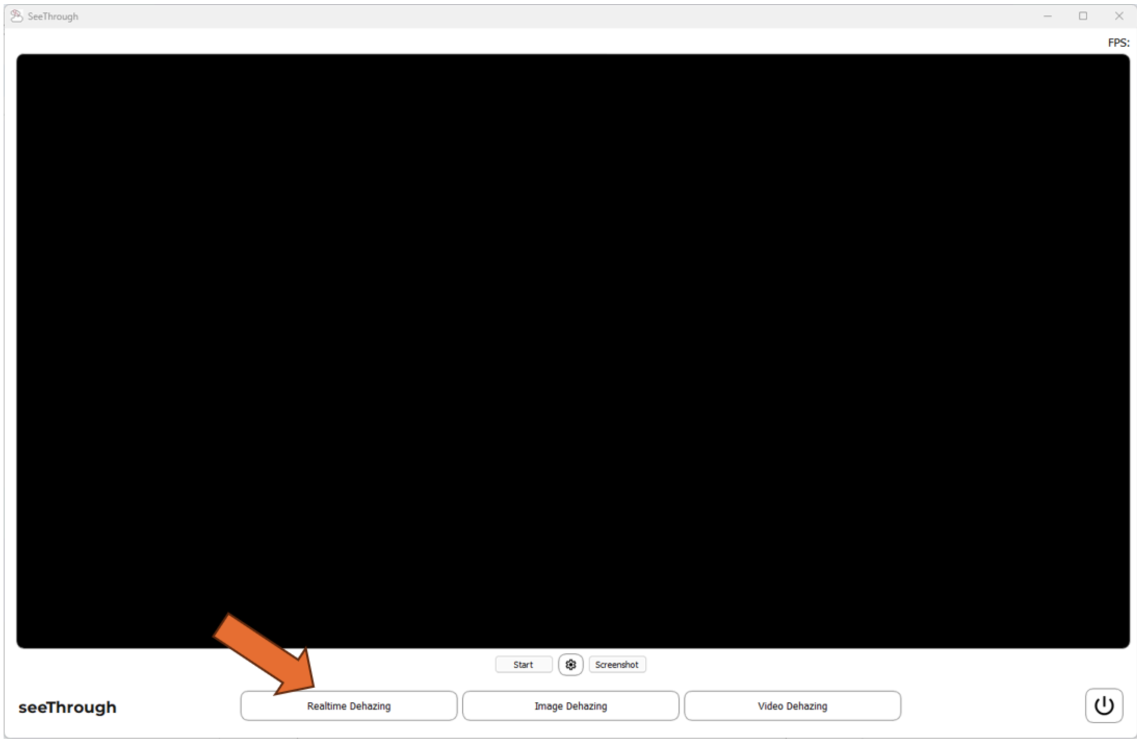

Setup - SeeThrough version 1.0
Ready to Install
Setup is now ready to begin installing SeeThrough on your computer.
Click Install to continue with the installation.
Back
Install
Cancel

Once done, click “Finish” and SeeThrough is now installed on the computer.

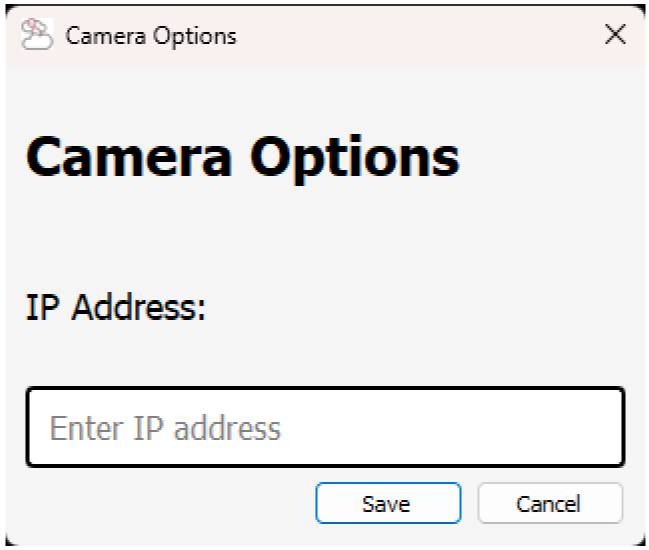


Part 2: Real-Time Dehazing

On the navigation tab on the bottom center, there are 3 navigations “Realtime Dehazing”, “Image Dehazing”, “Video Dehazing”. Click on Realtime dehazing button where the orange arrow points in below image:

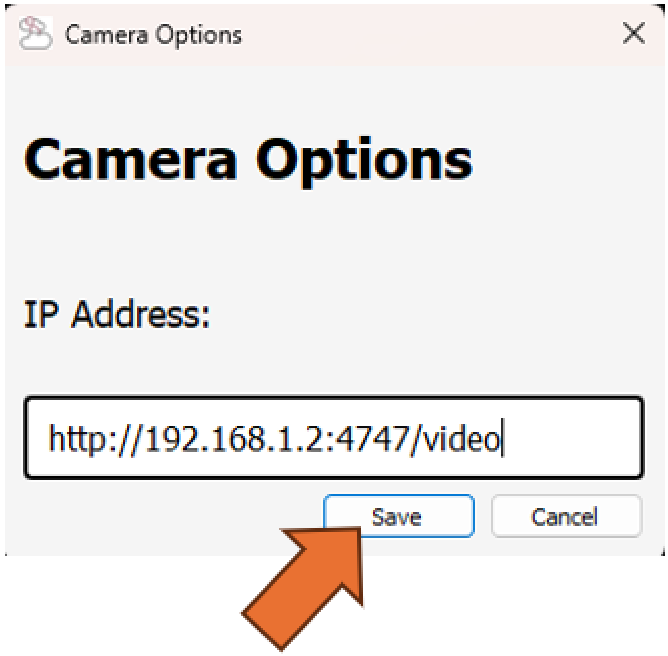


Once clicked, above the navigations, there 3 buttons “Start”, the settings button, and the “Screenshot” button. To utilize the real-time, click on the settings button and a pop-window will appear.

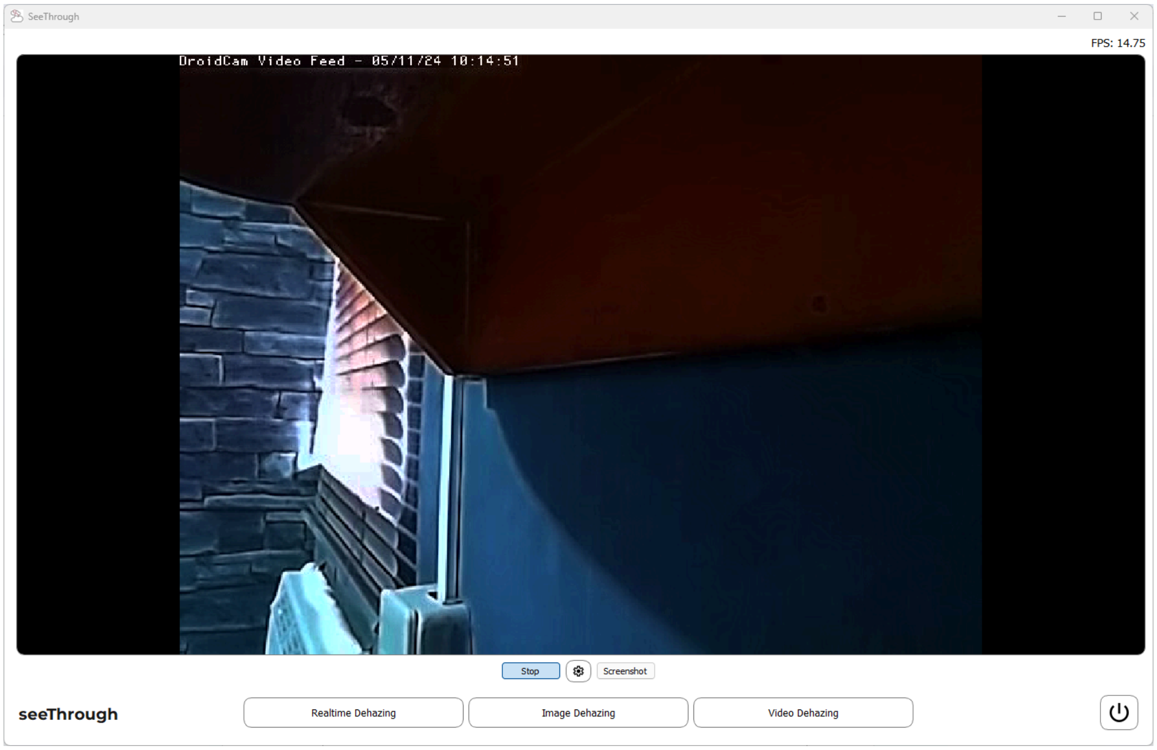


Here, the user might need to enter the IP Camera's RTSP (Real-Time Streaming Protocol) address like for example in the image below and click save.

NOTE: Not all RTSP is the same as the address below, please refer to your IP Camera's RTSP Address.



Once done, on the Realtime Dehazing tab, click the “Start” button to establish connection between the software and IP Camera to see the output.

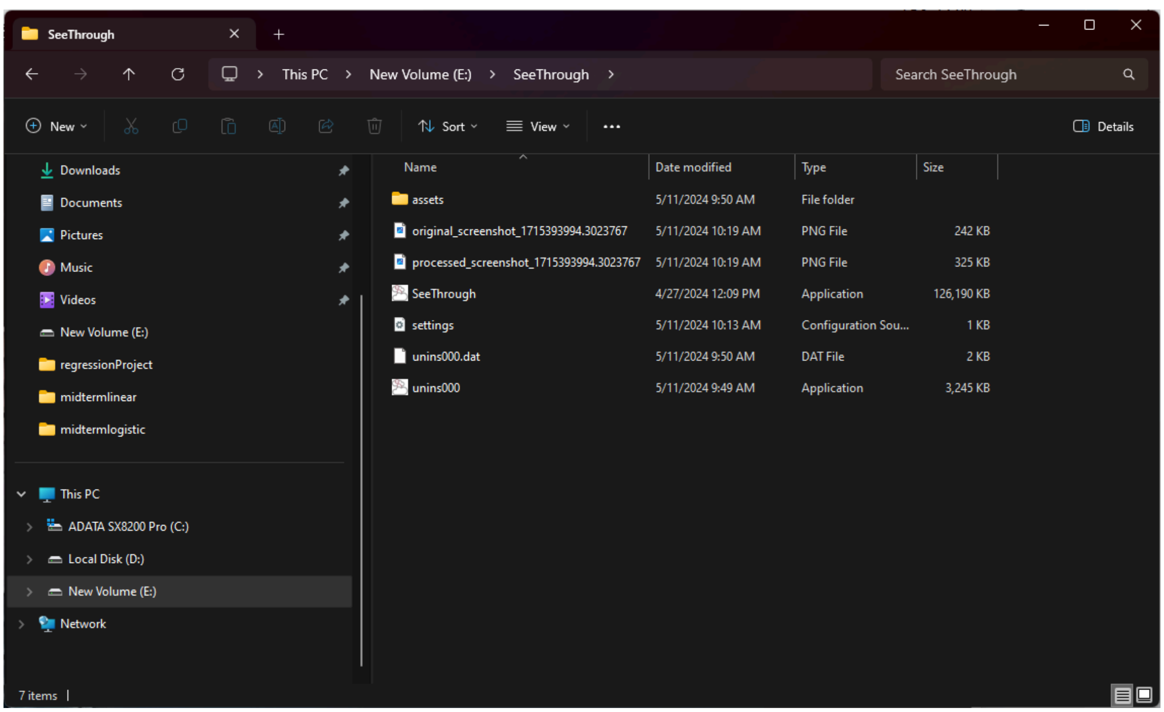


To stop the feed, click on the “Stop” button like on the image below.

Optional:

To screenshot while running, click on the “Screenshot” button and the screenshots will be found in the SeeThrough folder. The screenshot contains the original (unprocessed) image and a dehazed (processed) image. Please refer to the image below:

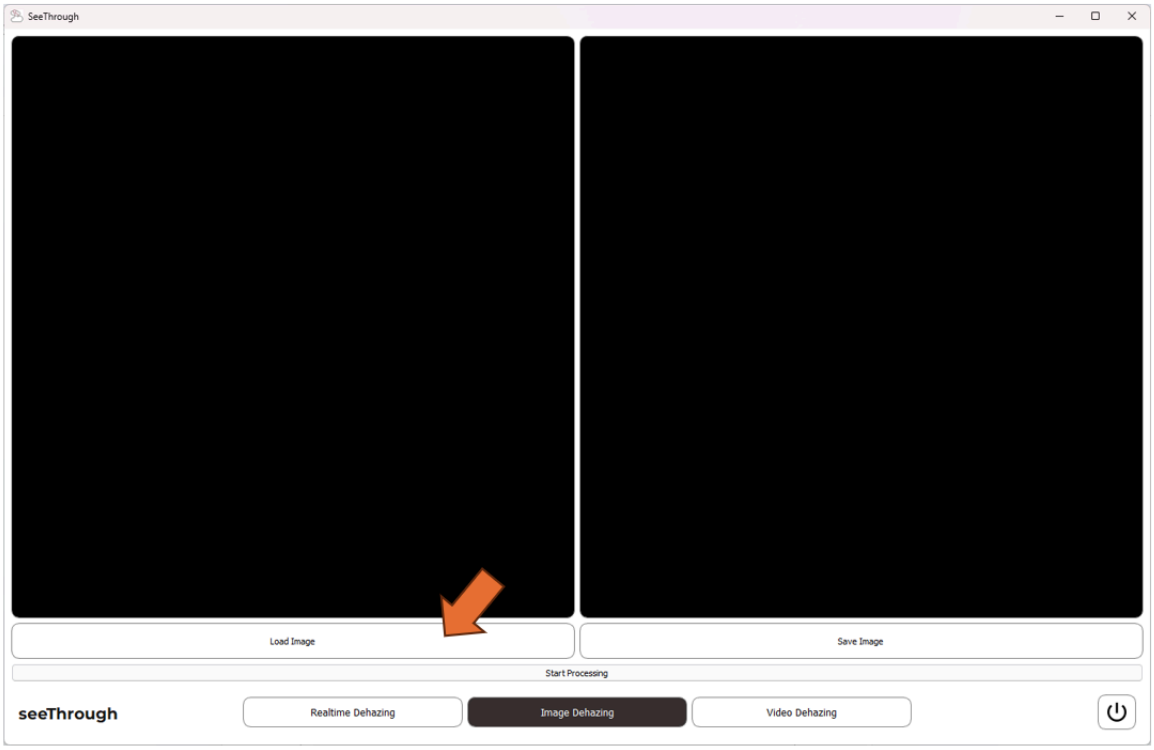

SeeThrough
FPS: 14.75
DroidCam Video Feed - 05/11/24 10:14:51
Stop
Screenshot
seeThrough
Realtime Dehazing
Image Dehazing
Video Dehazing


SeeThrough
This PC > New Volume (E:) > SeeThrough
Search SeeThrough
New
Sort
View
Details
Downloads
Documents
Pictures
Music
Videos
New Volume (E:)
regressionProject
midtermlinear
midtermlogistic
This PC
ADATA SX8200 Pro (C:)
Local Disk (D:)
New Volume (E:)
Network
Name
Date modified
Type
Size
assets 5/11/2024 9:50 AM File folder
original_screenshot_1715393994.3023767 5/11/2024 10:19 AM PNG File 242 KB
processed_screenshot_1715393994.3023767 5/11/2024 10:19 AM PNG File 325 KB
SeeThrough 4/27/2024 12:09 PM Application 126,190 KB
settings 5/11/2024 10:13 AM Configuration Sou... 1 KB
unins000.dat 5/11/2024 9:50 AM DAT File 2 KB
unins000 5/11/2024 9:49 AM Application 3,245 KB
7 items

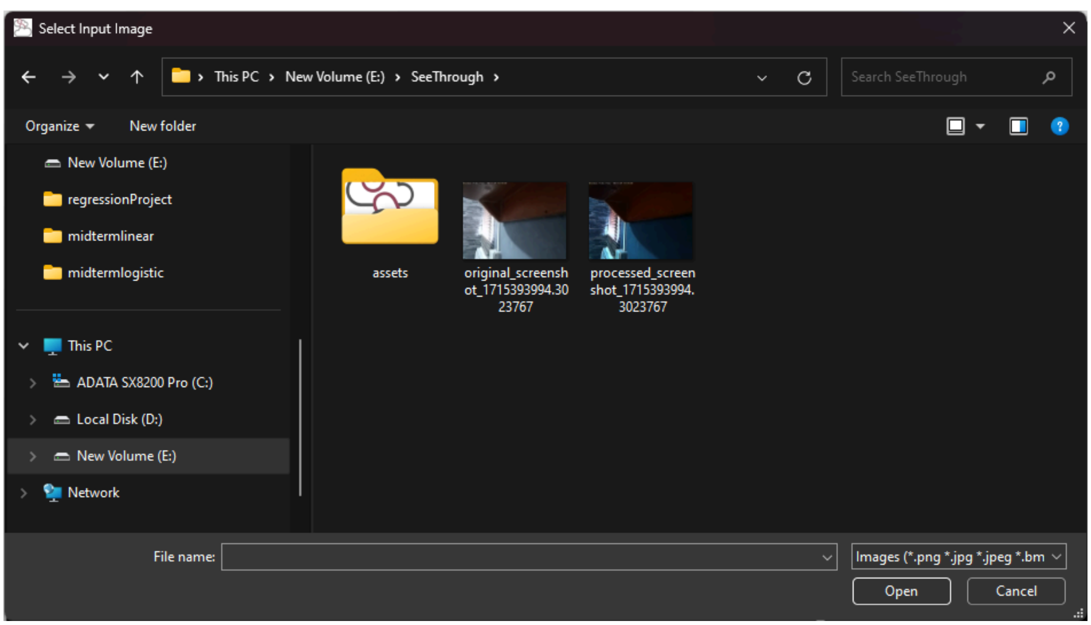
Part 3. Image Dehazing

On the navigation tab on the bottom center, click on Image dehazing button where the orange arrow points in below image:

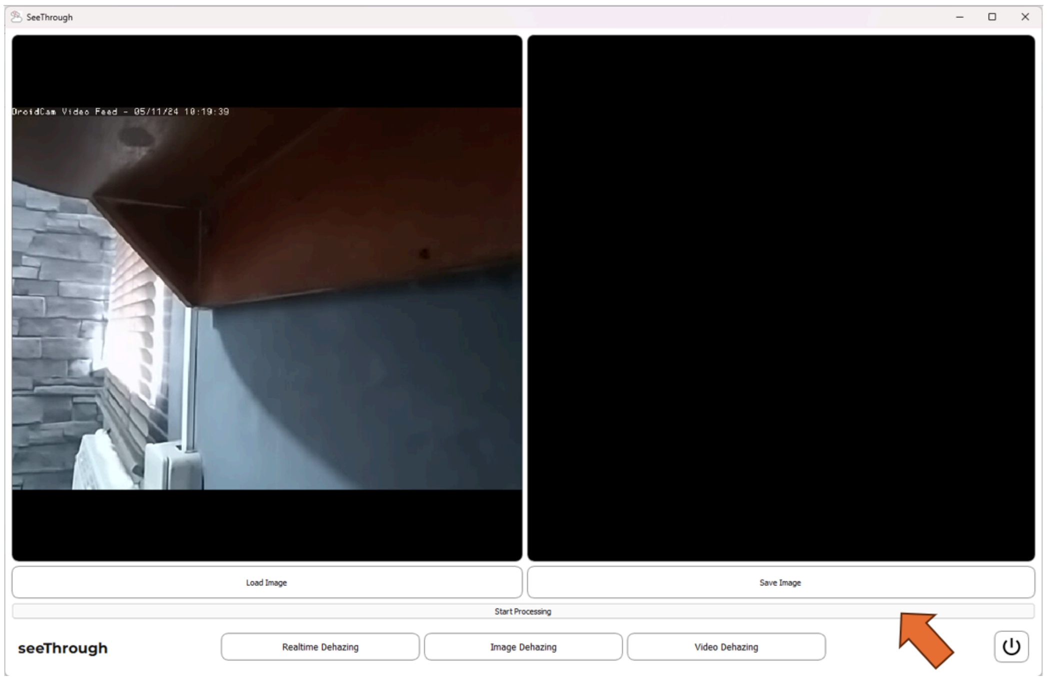
On the Image dehazing tab, there are 3 buttons, "Load Image", "Save Image", and "Start Processing". The user will load an image by clicking the "Load Image" button where the orange arrow points on the image below:

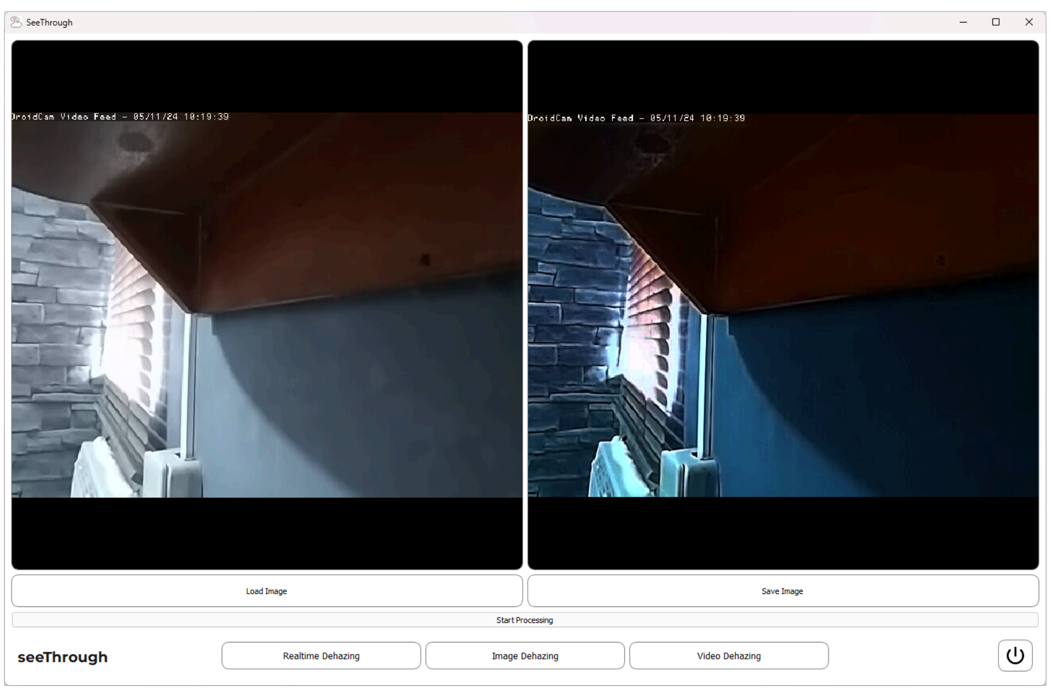


A pop-up file input for image will appear. Select any image that is applicable on user's end.

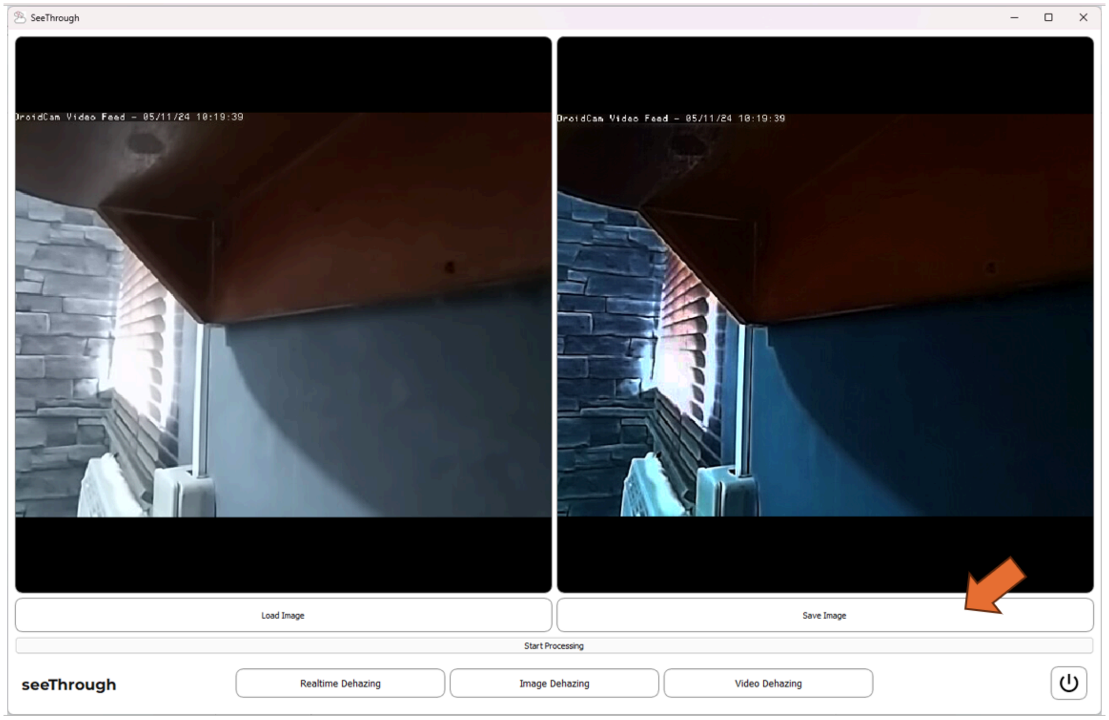


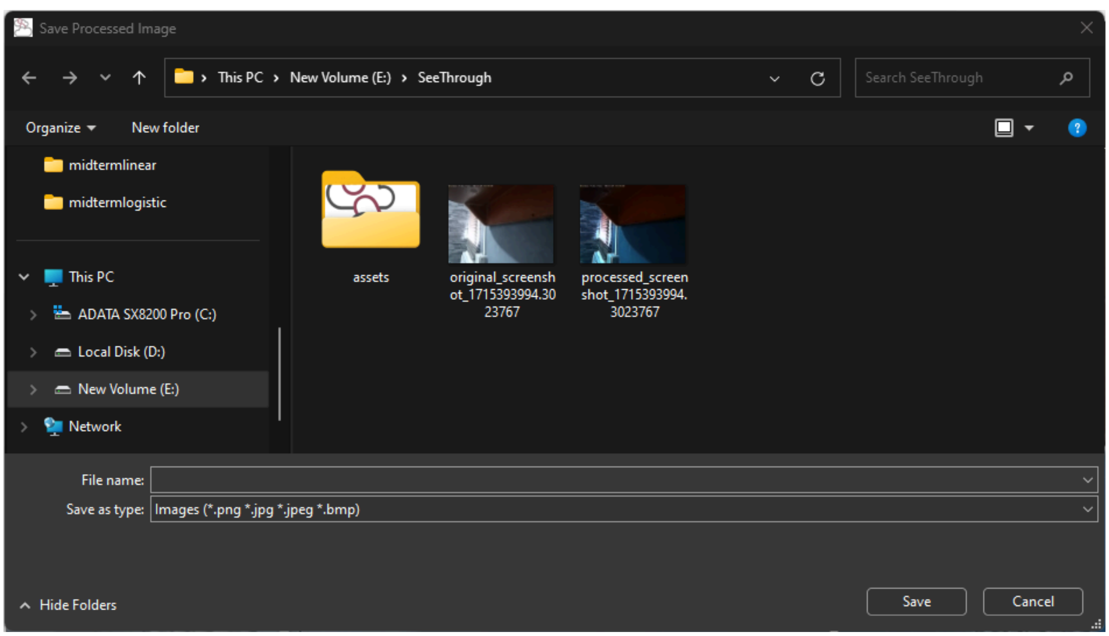
Once the image is uploaded, click on the "Start Processing" button to see the dehazed output of the image.





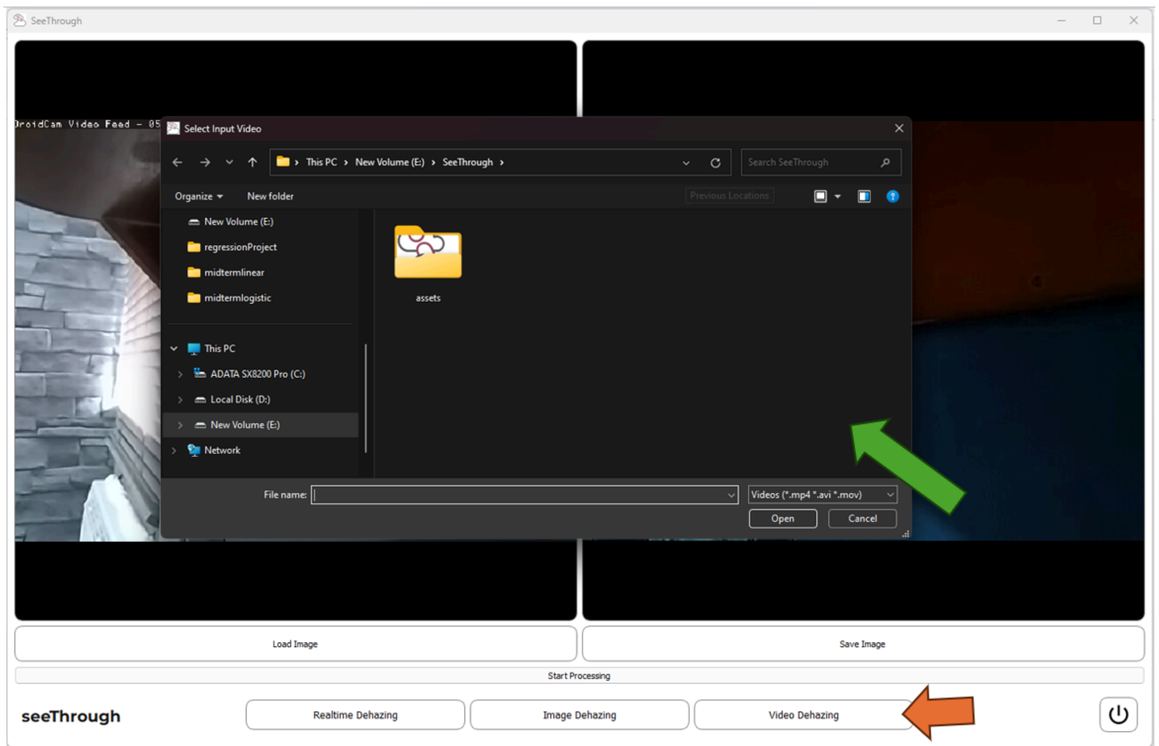
Once done, click on the "Save Image" button and a pop-window will appear again to ask where to save the dehazed output of the image.



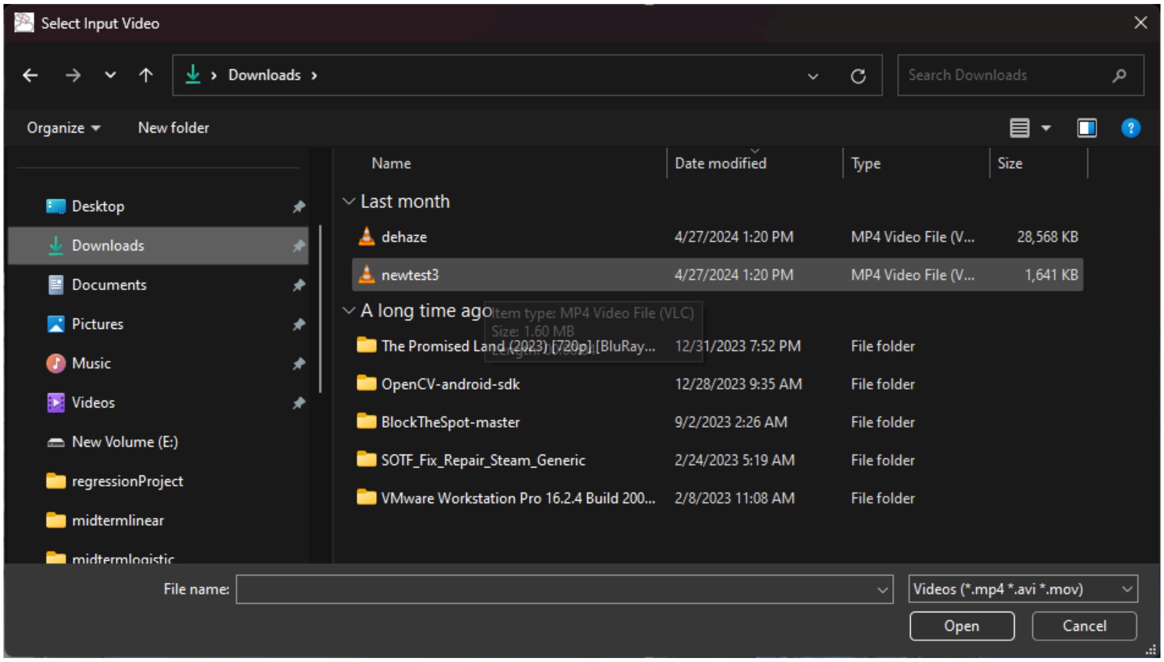


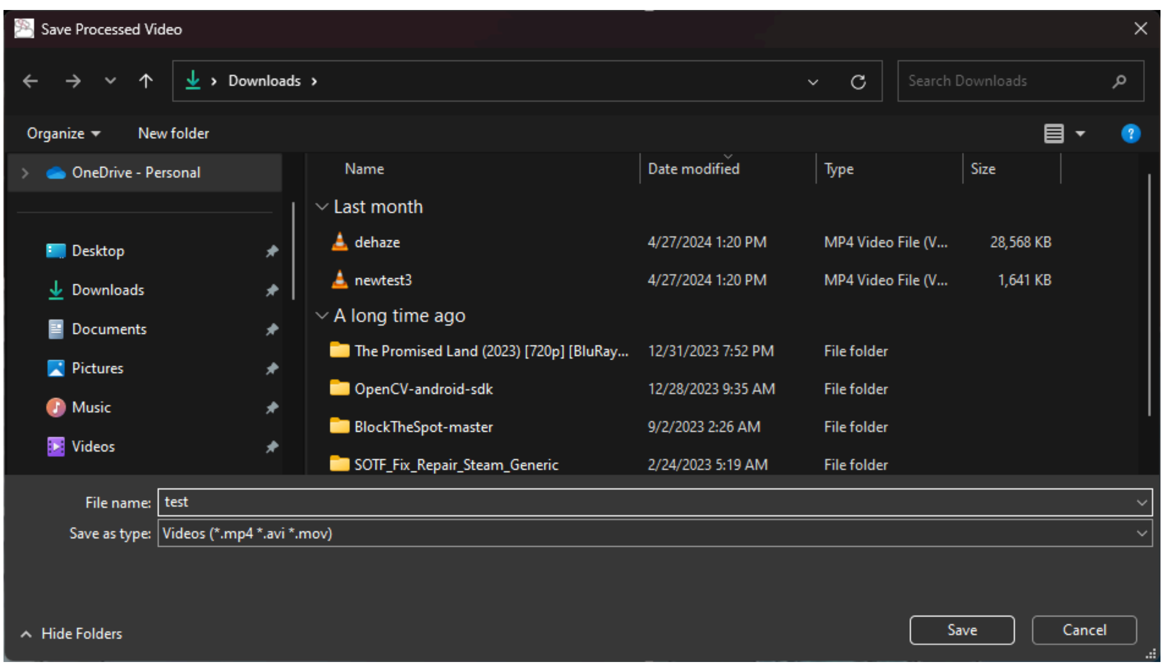
Part 4. Video Dehazing

On the navigation tab on the bottom center, click on Video dehazing button where the orange arrow points in below image, and a video file-input pop-up will appear on the screen where the green arrow points in below image:

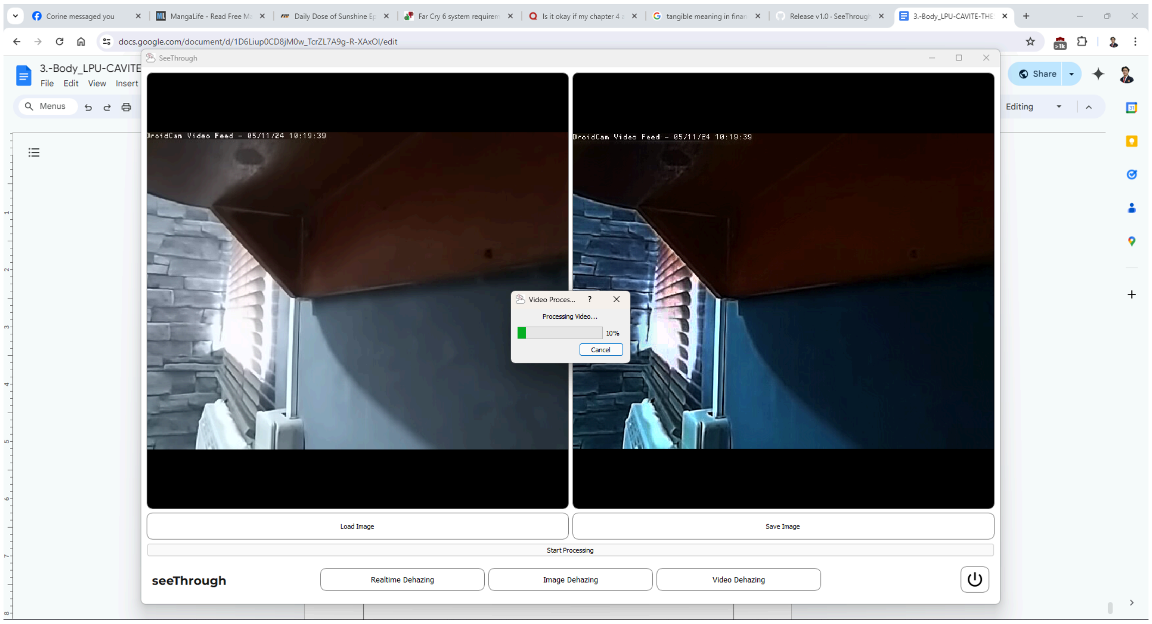


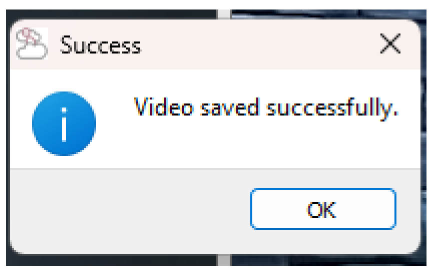
Select a video file on the computer, and another pop-up will appear asking where to save the file.





Once done, a progress bar will show up determining the progress of the dehazed video output and once the progress is done a message box will appear.





Optional:

To exit the application, click this button on the right bottom side of the application.

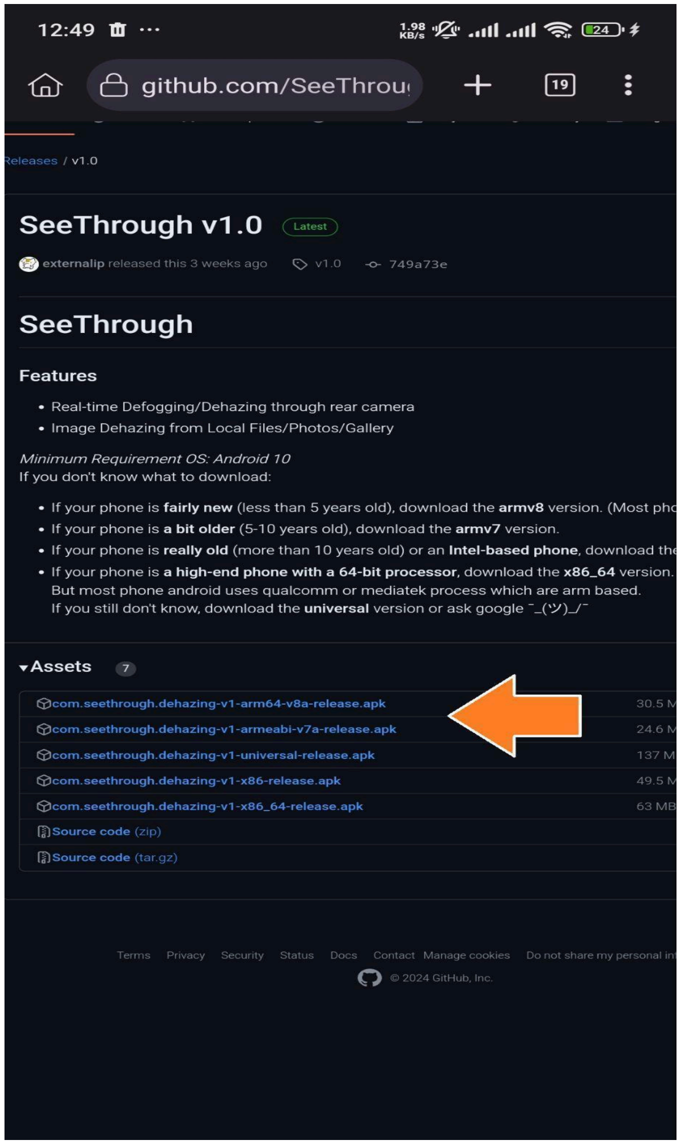
Mobile Application (Android only) User Guides

Part 1. Installation

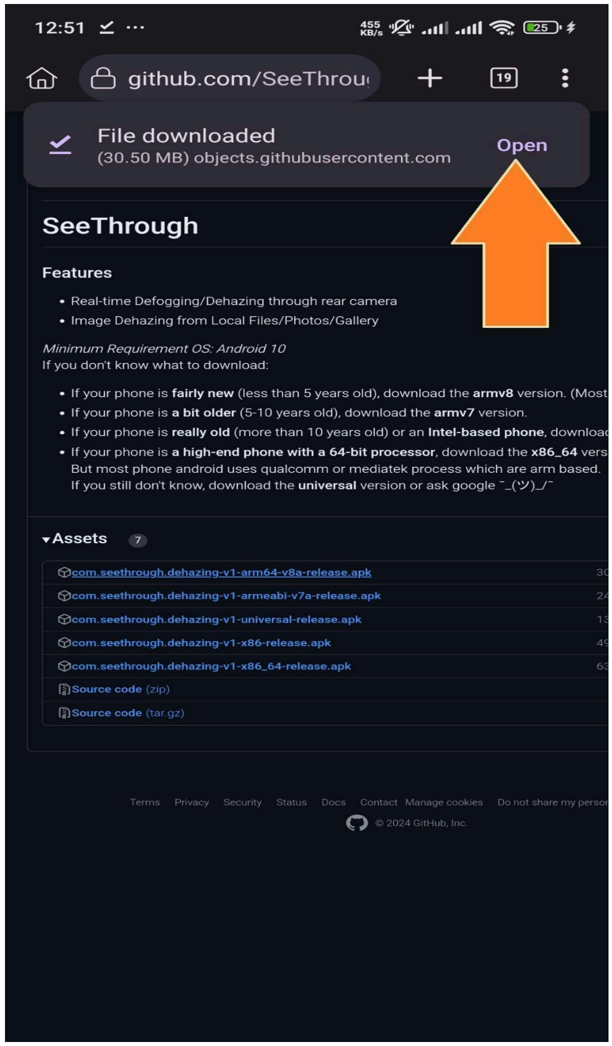
Go to this website https://github.com/SeeThrough-org/android-dehazing/releases/tag/v1.0 to download the APK file needed to install the mobile application however the user must refer to this instructions before selecting an APK file:

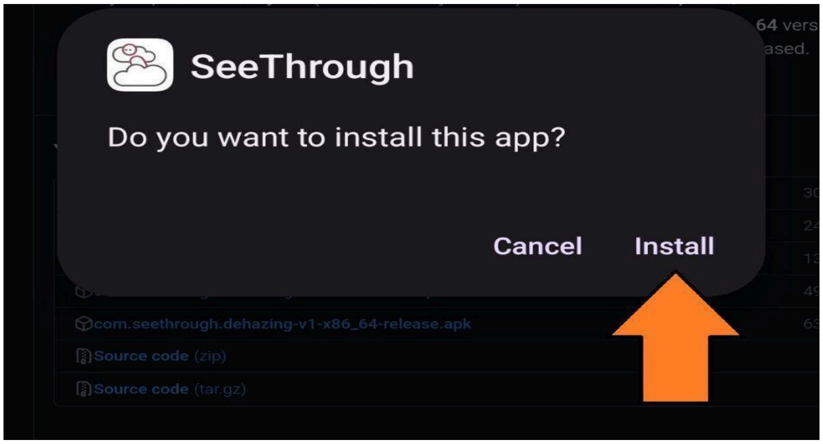
Instructions:

Minimum android version requirement **ANDROID 10**	
Requirements	**What to download**
If the user's phone is fairly new (less than 5 years old)	Download the **v8a** version
If the user's phone is a bit older (5-10 years old)	Download the **v7a** version
If the user's phone is really old (more than 10 years old) or an Intel-based phone	Download the **x86** version
If your phone is a high-end phone with a 64-bit processor	Download the **x86_64** version
If the user is not sure what to download then,	Download the **Universal** version


12:49
github.com/SeeThrou
Releases / v1.0
SeeThrough v1.0
Latest
externalip released this 3 weeks ago
v1.0
749a73e
SeeThrough
Features
Real-time Defogging/Dehazing through rear camera
Image Dehazing from Local Files/Photos/Gallery
Minimum Requirement OS: Android 10
If you don't know what to download:
If your phone is fairly new (less than 5 years old), download the armv8 version. (Most pho
If your phone is a bit older (5-10 years old), download the armv7 version.
If your phone is really old (more than 10 years old) or an Intel-based phone, download the
If your phone is a high-end phone with a 64-bit processor, download the x86_64 version.
But most phone android uses qualcomm or mediatek process which are arm based.
If you still don't know, download the universal version or ask google ¯_(ツ)_/¯
Assets 7
com.seethrough.dehazing-v1-arm64-v8a-release.apk
com.seethrough.dehazing-v1-armeabi-v7a-release.apk
com.seethrough.dehazing-v1-universal-release.apk
com.seethrough.dehazing-v1-x86-release.apk
com.seethrough.dehazing-v1-x86_64-release.apk
Source code (zip)
Source code (tar.gz)
Terms Privacy Security Status Docs Contact Manage cookies
© 2024 GitHub, Inc.

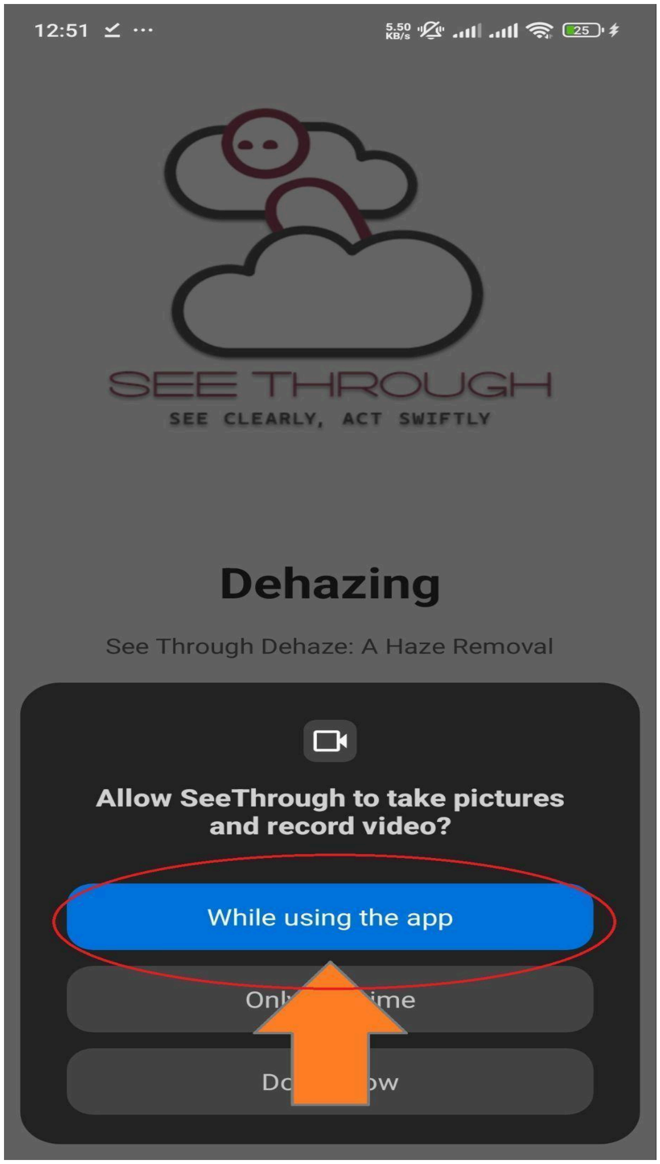
Once you have successfully downloaded the APK, open the APK and press the install button to install the app.



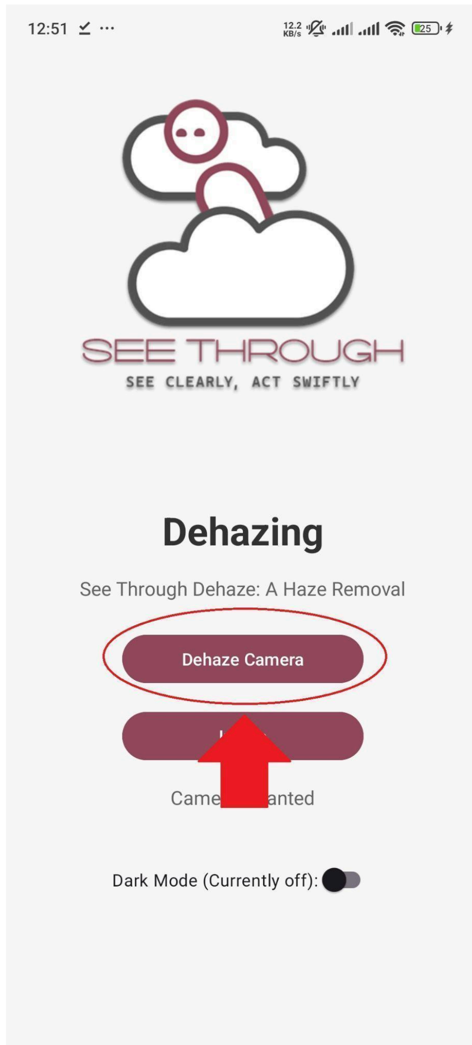


Part 2. Dehaze Camera

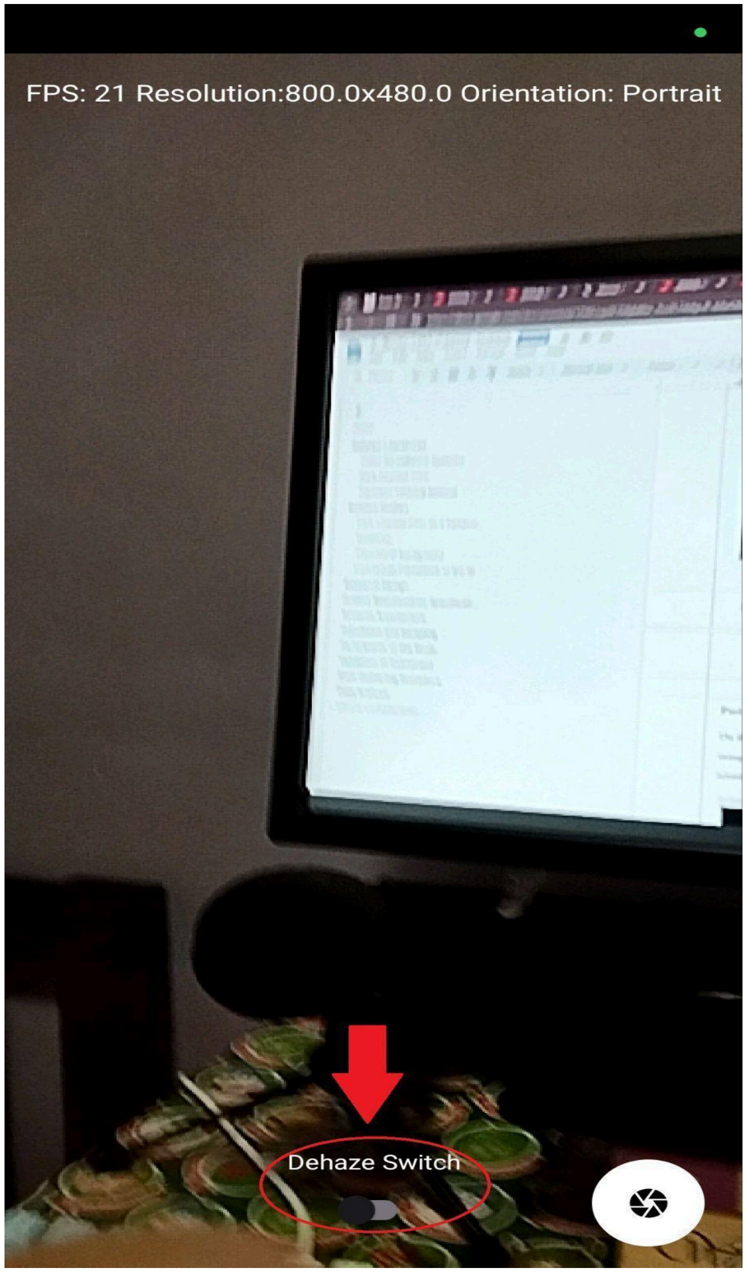
When the user opens the application for the first time, the application will request a camera permission. Make sure to allow/accept the camera permission request in order for the application to fully function.

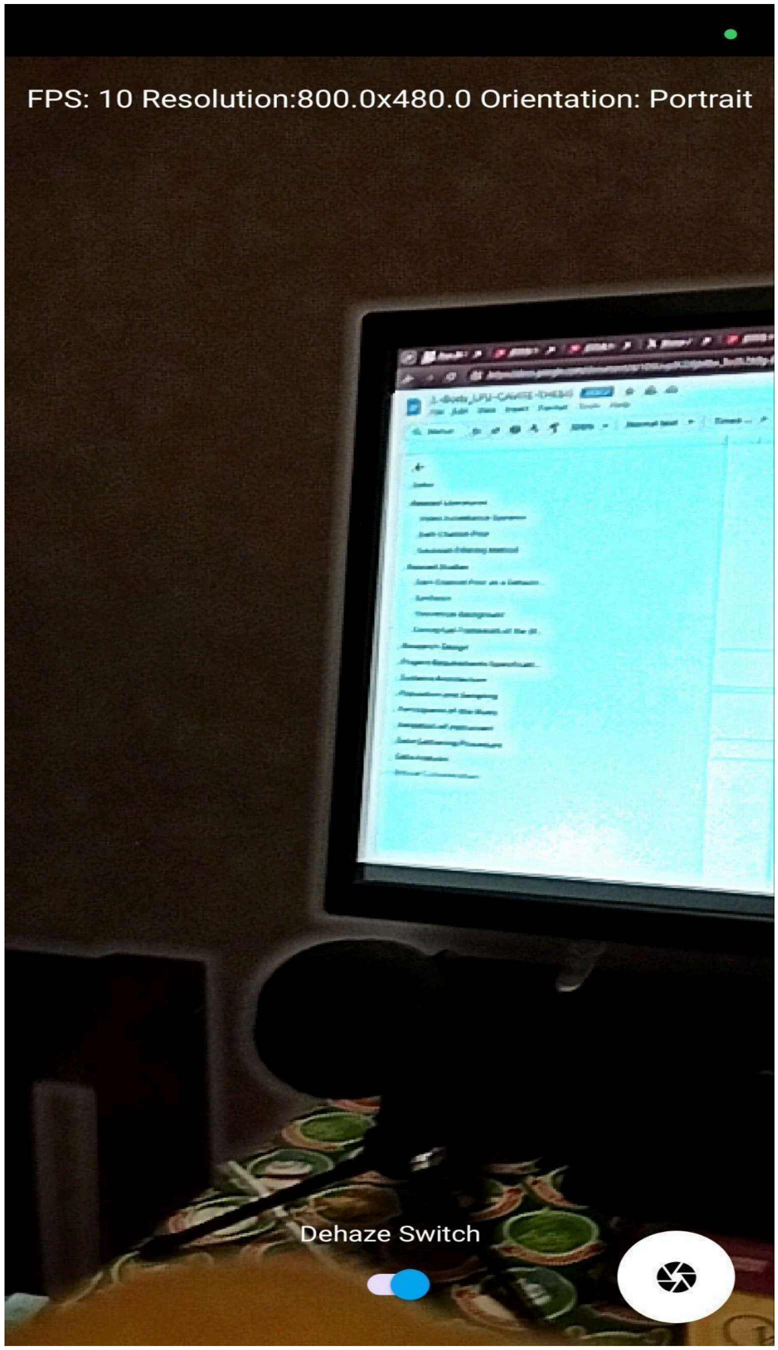


In the main menu of the app, the user has two options: Dehaze Camera and Image. Tap the Dehaze Camera button.



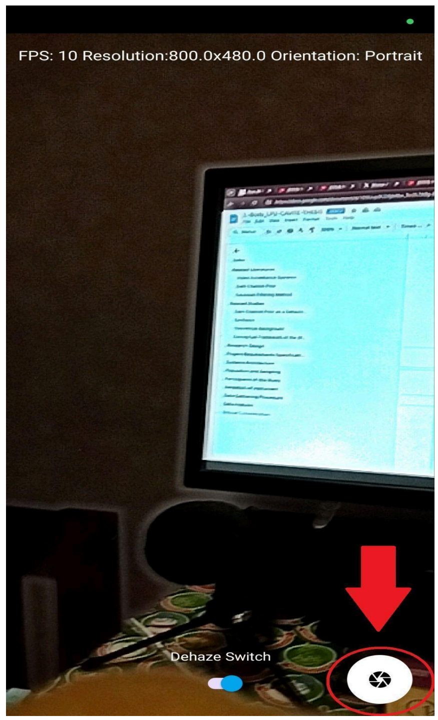
In the Dehaze Camera, the user can see the live feed and there is a switch to turn on/off the dehazing and a capture button.

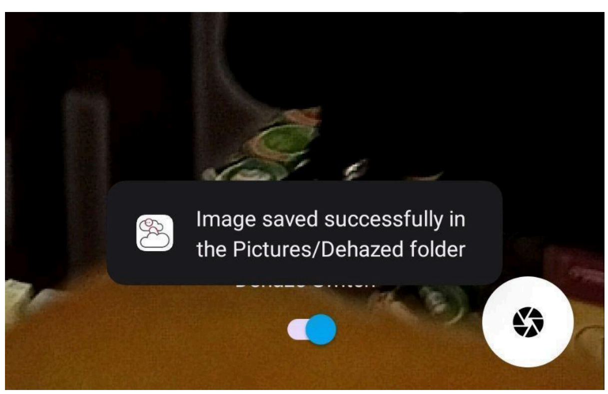

FPS: 21 Resolution:800.0x480.0 Orientation: Portrait
Dehaze Switch

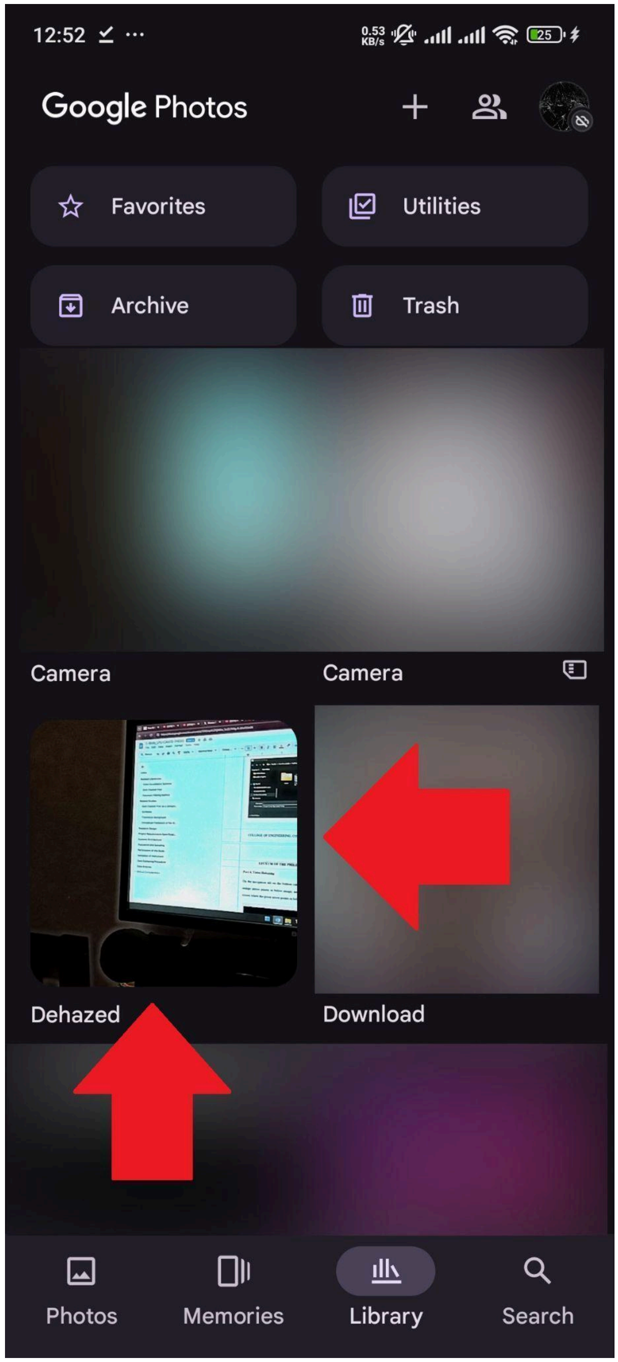

FPS: 10 Resolution:800.0x480.0 Orientation: Portrait
Dehaze Switch

Optional:

Tap this button on the bottom right side of the screen to screenshot the current imagery. This will save the image found on the phone's Pictures/Dehazed Folder depending if the dehaze switch is turned on/off.

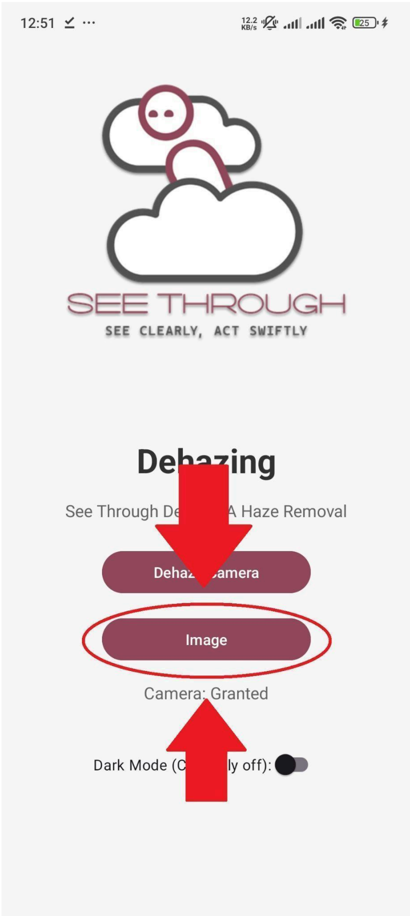




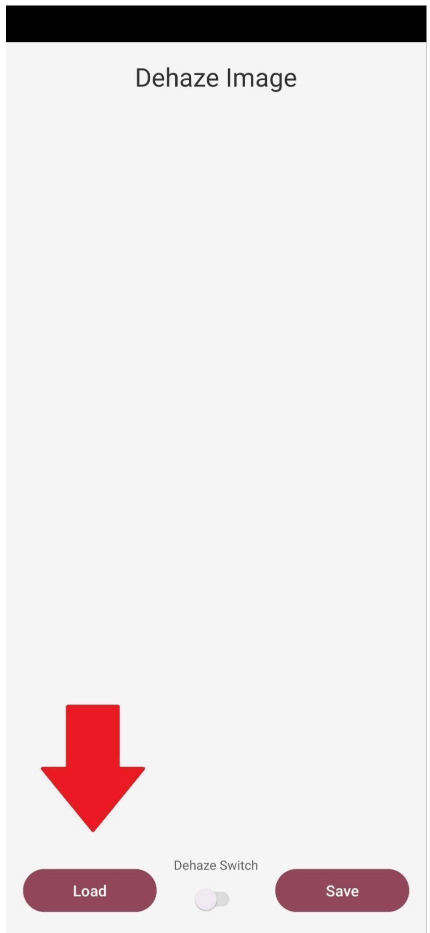

12:52
Google Photos
Favorites
Utilities
Archive
Trash
Camera
Camera
Dehazed
Download
Photos
Memories
Library
Search

Part 3. Image Dehazing

On the main menu of the application, click the Image Button.

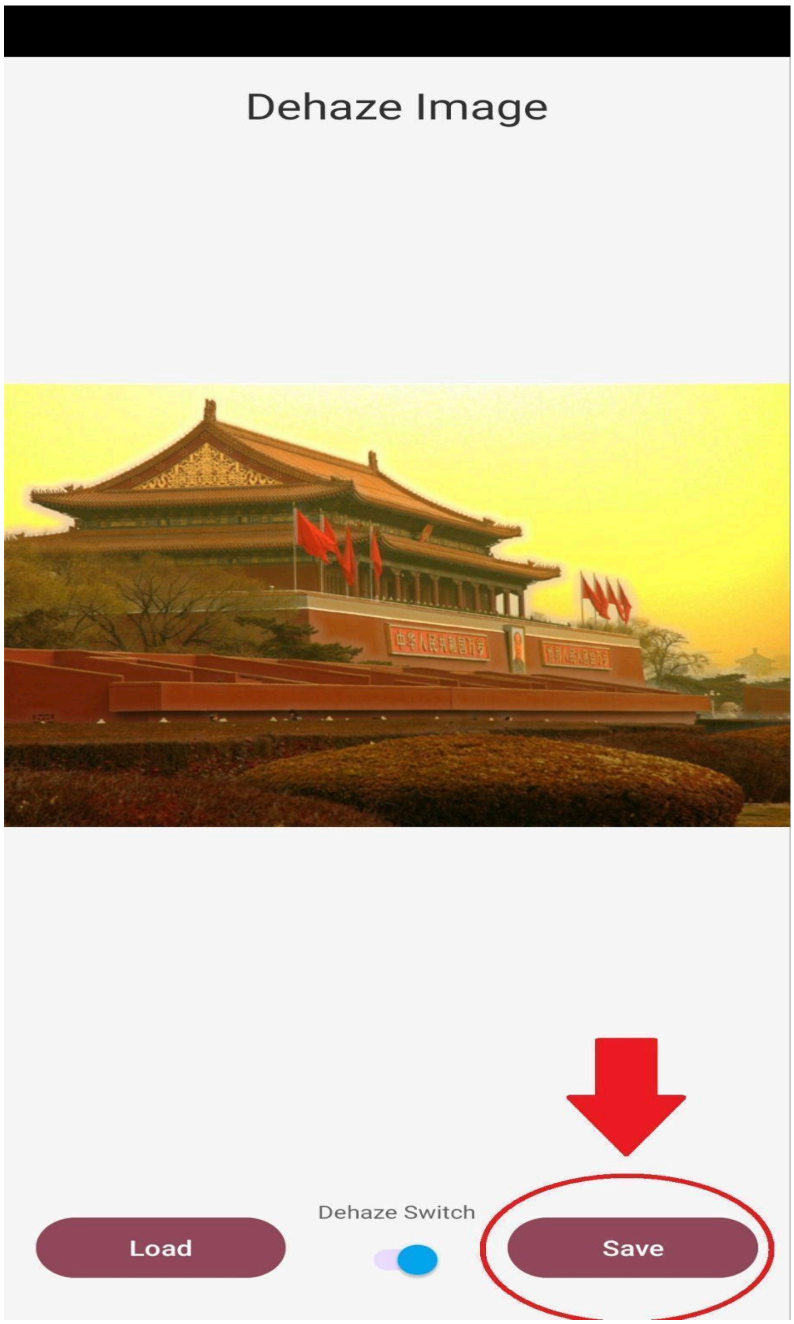


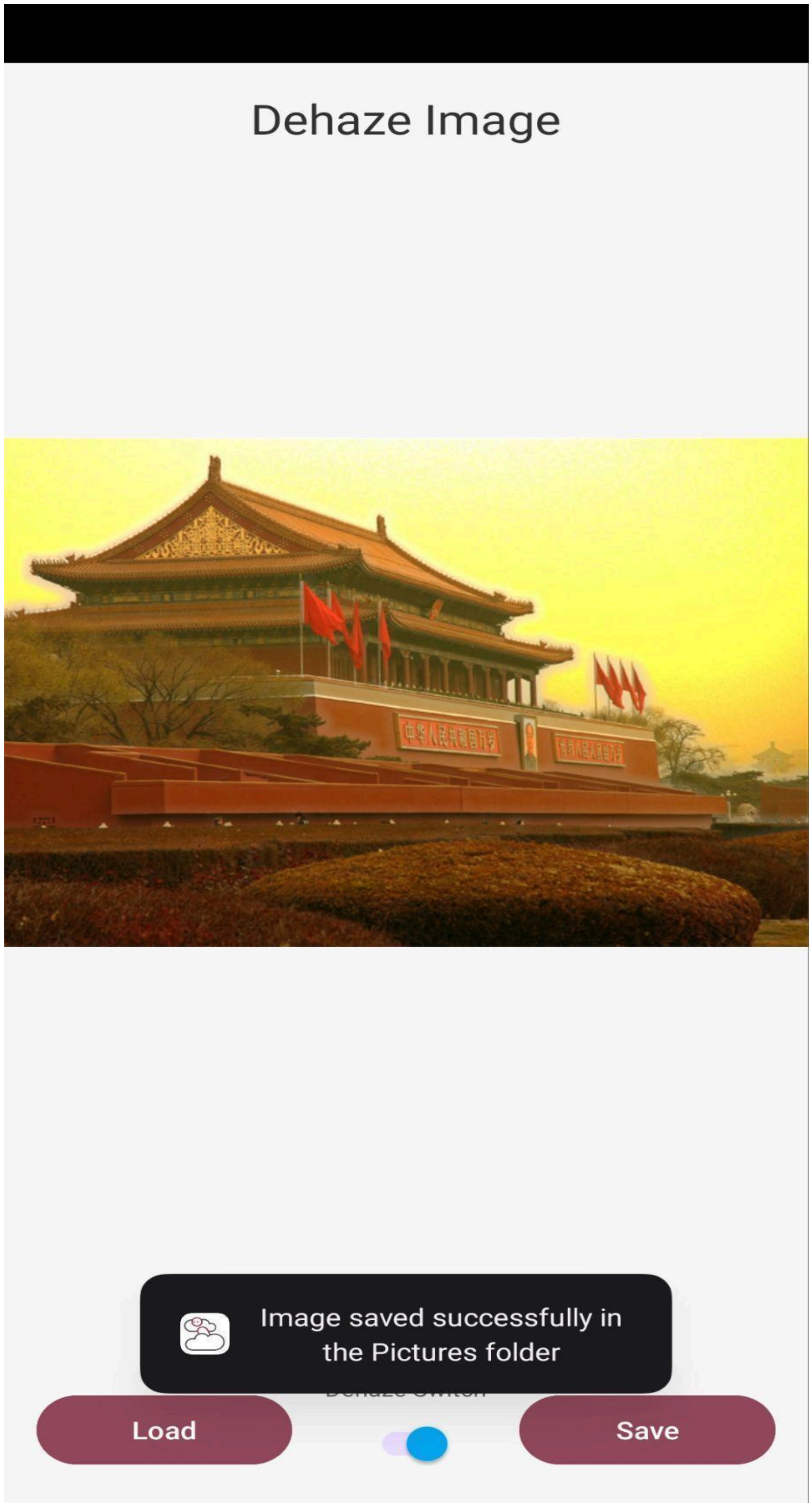
On the image dehazing page, select image files from the device by tapping the “Load” button to pop the image selection page from the smartphone.

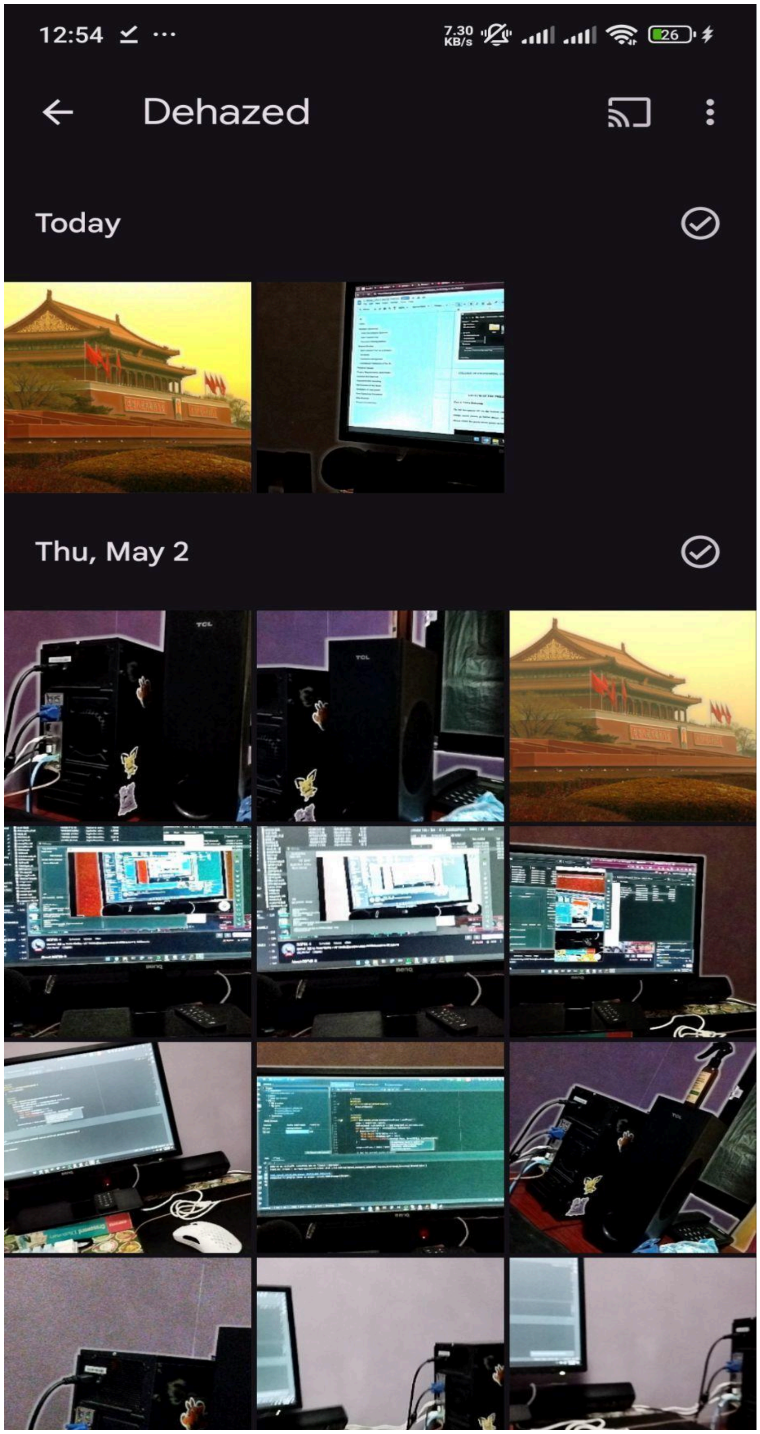


Once an image is uploaded on the application, turn on the dehaze switch in order for the user to see the dehazed preview of the image.

To save the image, tap “Save” button and it will be saved on the Pictures folder on the device.




Dehaze Image
Image saved successfully in
the Pictures folder
Load
Save


12:54
Dehazed
Today
Thu, May 2

Appendix 11

Turnitin Originality Report and Receipt

SUPER UPDATED SeeThrough Official Manuscript July 31, 2024.pdf

ORIGINALITY REPORT

6%	2%	6%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Juping Liu, Shiju Wang, Xin Wang, Mingye Ju, Dengyin Zhang. "A Review of Remote Sensing Image Dehazing", Sensors, 2021 Publication	1%
2	Chuan Li, Changjiu Yuan, Hongbo Pan, Yue Yang, Ziyan Wang, Hao Zhou, Hailing Xiong. "Single-Image Dehazing Based on Improved Bright Channel Prior and Dark Channel Prior", Electronics, 2023 Publication	<1%
3	Long Wu, Jie Chen, Shuyu Chen, Xu Yang, Lu Xu, Yong Zhang, Jianlong Zhang. "Hybrid Dark Channel Prior for Image Dehazing Based on Transmittance Estimation by Variant Genetic Algorithm", Applied Sciences, 2023 Publication	<1%
4	ouci.dntb.gov.ua Internet Source	<1%
5	Feiniu Yuan, Yu Zhou, Xue Xia, Xueming Qian, Jian Huang. "A confidence prior for image	<1%

turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

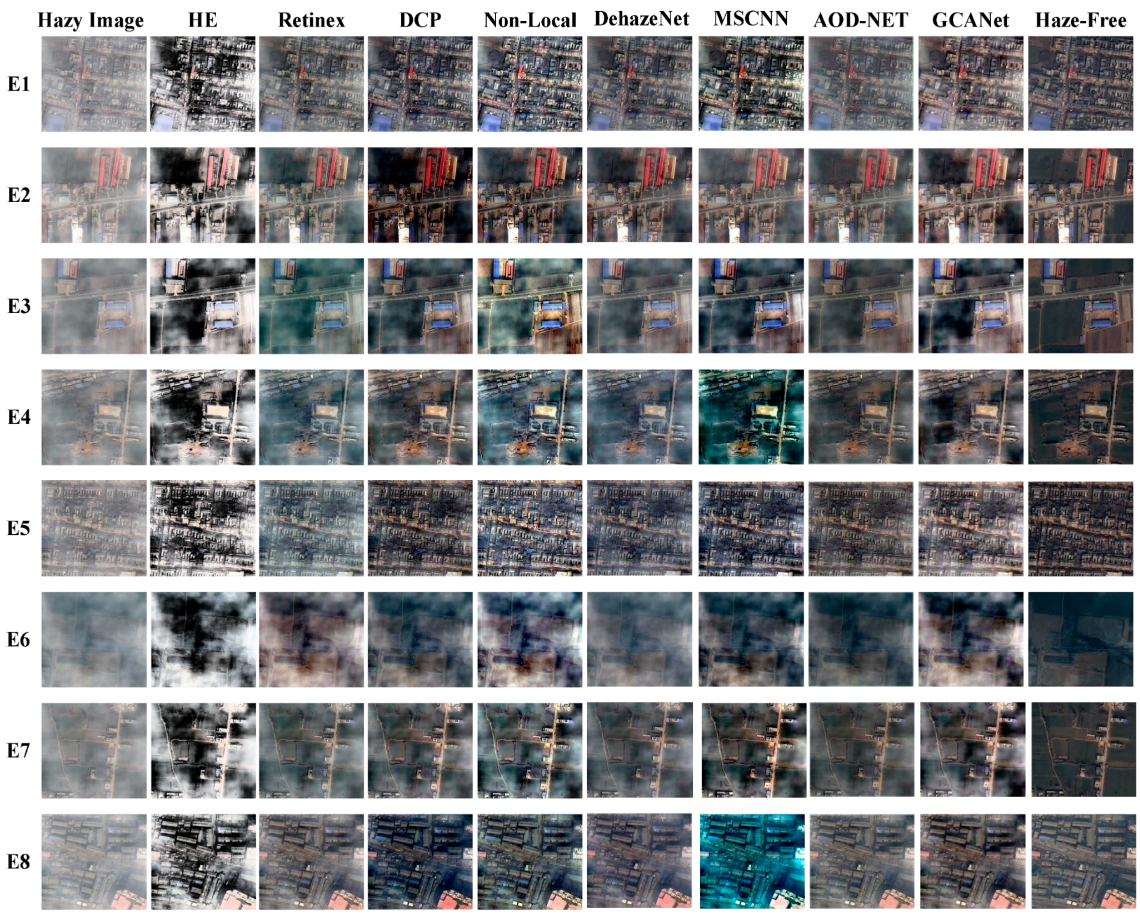
Submission author:	Ferdinand Carlo Palo
Assignment title:	Turnitin check of Final Manuscript
Submission title:	SUPER UPDATED SeeThrough Official Manuscript July 31, 202...
File name:	SUPER_UPDATED_SeeThrough_Official_Manuscript_July_31_2...
File size:	9.55M
Page count:	281
Word count:	35,828
Character count:	201,016
Submission date:	05-Aug-2024 09:42PM (UTC+0800)
Submission ID:	2427681569

Copyright 2024 Turnitin. All rights reserved.

Appendix 12

Liu et al's comparison of different dehazing algorithms

Liu et al's comparison of different dehazing algorithms



Source: https://www.mdpi.com/1424-8220/21/11/3926

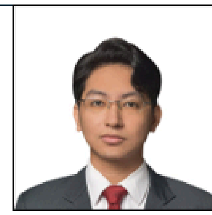
The results of above figure's metrics in PSNR and SSIM

Metric	Image	HE	Retinex	DCP	Non-Local	Dehaze Net	MSCNN	AOD-Net	GCA Net
SSIM	E1	0.6244	0.8533	0.8687	0.7670	0.8870	0.7742	0.8429	0.8416
	E2	0.6290	0.8297	0.6681	0.8649	0.8900	0.8723	0.8373	0.8581
	E3	0.5179	0.6285	0.7921	0.6042	0.7909	0.7427	0.8061	0.7321
	E4	0.4934	0.6150	0.7554	0.6082	0.6434	0.5457	0.7807	0.7662
	E5	0.6196	0.6911	0.8652	0.7251	0.7991	0.8230	0.8231	0.7827
	E6	0.3996	0.3357	0.7244	0.5104	0.6864	0.7246	0.7937	0.4873
	E7	0.4193	0.7345	0.7478	0.6231	0.7745	0.7045	0.7783	0.6874
	E8	0.5479	0.8689	0.7142	0.8267	0.8776	0.3633	0.8445	0.8437
PSNR	E1	11.3018	17.3161	19.9218	16.1124	20.5306	16.7291	19.1775	16.4637
	E2	12.4495	17.3834	16.0288	21.2499	22.6673	19.2980	19.1143	20.1044
	E3	10.0178	15.0239	17.2843	12.5169	15.3483	16.9267	18.1787	14.5837
	E4	9.1166	12.3160	14.7143	12.9381	14.2402	16.4077	17.4782	15.5258
	E5	9.9918	12.0322	19.2096	13.7839	16.9787	17.3780	17.9371	13.0886
	E6	8.2387	10.8556	14.9927	10.4802	12.7805	16.1591	17.7590	12.0327
	E7	8.6274	14.0277	14.9921	13.4883	15.6290	15.9782	17.6301	14.5927
	E8	12.1672	19.9542	17.5611	21.0496	22.7793	15.2877	20.5717	21.1517

Note: For SSIM - The value closer to 1, the better and for PSNR - The higher the value, the better

Appendix 13

Curriculum Vitae



Andre M. Nazareth

andre_nazareth@ymail.com

0927 854 1392

SS5 Block 2 Lot 7, Lancaster New City, Barangay Navarro, General Trias, Cavite

Bio

Andre is a 4th year BS Computer Science student of Lyceum of the Philippines University Cavite. He is a certified IT Specialist on Databases, has foundational knowledge in programming languages such as Java, PHP, Python, C#, and JavaScript, and has demonstrated some proficiency with programming frameworks such as Laravel, Vue.js, React.js, and .Net.

His interests lean towards full-stack web development, cybersecurity, and software testing. As a lifelong learner, he is driven to expand his technical skills to solve real world problems through code.

Alexandre F. Basiño

alexandrebasino05@gmail.com

0915 415 5091

Block 9 Lot 10 Somerset Phase-4 Lancaster New City, Navarro, General Trias, Cavite

Bio

Alexandre Basiño is a 4th year BS Computer Science student of Lyceum of the Philippines – Cavite with proficiency in various technologies including Android Studio, Git, JavaScript, ReactJS, PHP, Java, and Python. He has a strong foundation in both front-end and back-end development, with solid experience in version control systems.

His interests lean toward full-stack development, mobile app creation, and software engineering. He is eager to improve his knowledge and skills in these computing fields to pursue a career in innovative technology solutions. He is passionate about building cutting-edge applications and continuously expanding his technical proficiency.

Ferdinand Carlo M. Palo

ferdinand.palo@lpunetwork.edu.ph

0905 379 4008

BLK 31 LT 15 Grandriverside Subdivision, Pasong Camachile I, General Trias City, Cavite

Bio

Ferdinand Carlo Palo is a 4th year BS in Computer Science student of Lyceum of the Philippines University – Cavite. He has experience with designing and development of desktop and mobile applications and excels at doing UI/UX and Frontend development. He has foundational knowledge in designing software such as Figma, development software such as Microsoft Visual Studio Code and Android Studio, programming languages such as JavaScript and Python, and frameworks like VueJS and Tailwind CSS.

His interest leans toward Frontend Designing and Development for web applications and desktop/mobile systems. On the project he designed how the user will interact with the system and how the system communicates with the users to ensure that the project delivers the expected output of the dehazing system.