

Beyond Security: Leveraging Vision Solutions in Building Surveillance

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

by

Prayushi Mathur

2021701034

prayushi.m@research.iiit.ac.in



International Institute of Information Technology
Hyderabad - 500032, INDIA
January 2024

Copyright © Prayushi Mathur, 2024
All Rights Reserved

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Beyond Security: Leveraging Vision Solutions in Building Surveillance” by Prayushi Mathur, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Dr. Syed Azeemuddin

Co-Advisor: Dr. Charu Sharma

To UNIVERSE

Acknowledgments

I would like to express my sincere gratitude to my advisors Dr. Syed Azeemuddin and Dr. Charu Sharma for their unwavering support, expert guidance, and boundless patience. Their mentorship has been instrumental in shaping the direction of my research and in my personal and academic growth. I have greatly benefited from my time at IIIT Hyderabad, where the vibrant culture of innovation has nurtured my passion for research, making it an incredibly enriching experience. My overwhelming gratitude extends to my mom, dad and Prakarsh for their unwavering belief in me and their constant encouragement. Your love, understanding, and sacrifices have been the driving force behind my pursuit of higher education. To my colleague and friend Kajal, your camaraderie and understanding have been a constant source of motivation. The daily conversations, shared experiences, and occasional juice/shake breaks not only eased the stress of work but also enriched my perspective. Your encouragement during late nights and busy days made this endeavor more manageable. To my human diary and human alarm clock Shraiya, who stood by me through thick and thin, your unwavering belief in my abilities and your unconditional support have been the bedrock of my strength. Whether it was lending an empathetic ear during challenging times or celebrating the small victories, your presence has made this academic journey far more meaningful. I also wish to express my heartfelt appreciation to Shreyansh, you have been more than just a friend, you've been my confidant and my cheerleader throughout the journey. Thank you Anu, Ganesh, Aditya, Shambhavi, Bhumika, Apoorva, Geetika, Sahish for being a vital part of my life and for cheering me on in this academic pursuit.

Abstract

In the infrastructure industry, the focus is shifting from constructing new suburban buildings to maintaining and rehabilitating existing structures, particularly high-rise buildings. These buildings are prone to various failures due to age, density, and altitude, making regular maintenance vital for safety and longevity. However, traditional inspection methods using heavy machinery and risky rappelling are time-consuming and costly. They often fail to provide comprehensive details for inaccessible surfaces. Finding safer and more efficient inspection approaches is essential to ensure building safety and durability.

We introduce an innovative end-to-end pipeline designed specifically for high-rise building inspection. Our proposed method incorporates several key components to ensure effective inspection processes. Firstly, we develop a trajectory generation system for an unmanned aerial vehicle (UAV). This system optimizes the UAV's trajectory, enabling it to reach the desired destination even in the presence of obstacles. The trajectory can be dynamically adjusted in real-time to ensure efficient navigation. During the UAV's flight, it captures images of the high-rise building using predetermined camera and drone parameters. These images serve as the basis for building inspections, particularly in detecting cracks. Moreover, the collected pool of images is utilized to construct a detailed 3D mesh model of the high-rise building. This model allows for a comprehensive representation of the structure, facilitating the identification and visualization of detected cracks. By combining trajectory optimization, image capture, crack detection, and 3D mesh modeling, our proposed pipeline offers a comprehensive approach to high-rise building inspection. It presents a promising solution to enhance the efficiency, accuracy, and safety of inspection processes in the field of structural engineering.

Our work explores the underexplored domain of deep learning in thermalimaging. It focuses on the classical problem of generating high-resolution images from low-resolution counterparts using Super-resolution (SR) techniques. The objective is to extract more details from the original scene by increasing the pixel density, which is particularly valuable in computer vision applications, including pattern recognition and medical imaging. The proposed pipeline aims to achieve real-time video super-resolution using a thermal camera on an embedded edge device.

Contents

Chapter	Page
1 Introduction	1
1.1 Inspection using UAVs	1
1.2 Depth vision	3
1.3 Motivation	3
1.4 Contribution	4
1.5 Thesis Organization	4
2 Related Works	6
2.1 Building Inspection	6
2.1.1 Traditional Methods	6
2.1.2 Modern Methods	7
3 Autonomous Inspection	9
3.1 Introduction	9
3.2 Theory	9
3.2.1 Health Inspection of buildings	9
3.2.2 Inspection Methods	10
3.3 Autonomous Drone Navigation	11
3.3.1 Trajectory Planning	11
3.3.2 Collision Avoidance	13
3.3.2.1 Obstacle Detection	13
3.3.2.2 Trajectory Optimization	14
3.3.3 Image Capturing	14
3.4 Model Construction	15
3.4.1 3D Reconstruction	16
3.4.1.1 Camera Parameter Extraction	17
3.4.1.2 Reconstruction	17
3.4.2 2D to 3D Mapping	18
3.4.2.1 Image correction	18
3.4.2.2 Ray Tracing	19
4 Façade Detection	21
4.1 Introduction	21
4.2 Façade Defects and Inspection Practices	21
4.2.1 Types of Façades	21

4.2.2	Façade Defects and Anomalies	23
4.3	Crack Detection	23
4.3.1	Challenges	23
4.4	Façade Inspection	24
4.4.1	Visual Inspection	24
4.4.2	Deep Learning Method	25
5	Experiments and Results	27
5.1	Façade Detection	27
5.1.1	Pipeline	27
5.1.2	Implementation details	27
5.1.3	Analysis	29
6	Superresolution	33
6.1	Introduction	33
6.2	Theory	34
6.3	Related Work	34
6.4	Dataset	35
6.5	Deep Learning Models	36
6.5.1	ESPCN	36
6.5.2	FSRCNN	37
6.5.3	LapSRN	38
6.5.4	EDSR	39
6.6	Evaluation Metrics	40
6.6.1	Peak Signal-to-Noise Ratio	40
6.6.2	Structural Similarity	42
6.7	Experiments and Results	43
6.7.1	Implementation Details	43
6.7.1.1	Model Training	44
6.7.1.2	Inference Pipeline	44
6.7.2	Results	46
6.7.3	Contribution	49
7	Conclusion and Future Work	51
	Bibliography	53

List of Figures

Figure	Page
1.1 Building inspection using UAVs	2
3.1 Our reconstructed 3D mesh model (left) of the high-rise building (right). The proposed pipeline reconstructs the 3D model from the real-world images captured by UAV, where the red spiral path marks the trajectory followed by the UAV.	12
3.2 2D projection of Trajectory Planning on the XY plane. The top view of the building is represented by the inner rectangle and ellipse trajectory followed by the UAV maintaining a safe distance, s , from the building.	13
3.3 A random sample of images captured of the high-rise building by the UAV.	15
3.4 3D mesh model of the building (left) using 2D to 3D mapping shown using the red ray and red ring; Captured image of the building (right) using UAV showing 2D image of it. Correspondences of all the three images are shown using blue lines.	17
3.5 Ray Tracing	19
5.1 Proposed Building Inspection Pipeline consists of three modules: (1) Autonomous Drone Navigation containing (a) Trajectory Planning, (b) Collision Avoidance and (c) Image Capturing; (2) Façade Detection; (3) Model Construction containing (a) 3D Reconstruction and (b) 2D to 3D Mapping	28
5.2 Evaluation metrics of the crack detection model.	29
5.3 Our reconstructed 3D mesh model (left) of the high-rise building (right) with the detected crack (bottom right). The proposed pipeline reconstructs the 3D model from the real-world images captured by UAV, where the red spiral path marks the trajectory followed by the UAV.	30
5.4 3D mesh model of the building (left) with marked crack using 2D to 3D mapping shown using the red ray and red ring; Captured image of the building (top-right) using UAV showing the crack predicted using YOLOv5 [1]; Zoomed-in Crack (bottom-right); Correspondences of all the three images are shown using blue lines.	31
5.5 The final simulation of the autonomous drone navigation on our high-rise building 3D reconstruction (left); the RGB camera view of the drone (top right); the depth camera view of the drone (bottom right).	32
6.1 ESPCN network architecture	37
6.2 FSRCNN network architecture	38
6.3 LapSRN network architecture	39
6.4 EDSR network architecture	40

6.5	Inference Pipeline.	46
6.6	Sample target image (HR) and predicted output of various deep learning models using various scaling factors (2x, 3x, 4x, 8x) on test set.	50

Chapter 1

Introduction

Building inspection is a crucial process for assessing the condition, safety, and compliance of structures. With advancements in technology, unmanned aerial vehicles (UAVs), commonly known as drones, have emerged as a powerful tool for conducting building inspections [2]. Drones equipped with high-resolution cameras, thermal imaging, and other sensors offer a safer, more efficient, and cost-effective alternative to traditional inspection methods. By integrating UAVs into building inspection, the aim is to optimize efficiency, accuracy, and effectiveness in evaluating the condition and integrity of structures. Drones provide a safer, more accessible, cost-effective, and data-rich alternative to traditional inspection methods, leading to improved outcomes in terms of safety, compliance, maintenance, and overall building quality.

1.1 Inspection using UAVs

Drones have become increasingly valuable tools for building inspection and surveillance due to several advantages they offer over traditional methods. Here are some reasons why drones are beneficial and a comparison with traditional inspection methods:

1. *Accessible and Efficient:* Drones can easily access areas that are difficult or dangerous for humans to reach, such as rooftops, high-rise buildings, or confined spaces. They provide an efficient way to inspect these areas without the need for scaffolding, ladders, or specialized equipment, saving time and reducing potential risks to inspectors.
2. *Aerial Perspective:* Drones provide an aerial perspective, allowing inspectors to capture high-resolution images, videos, and thermal data of the entire building envelope. This comprehensive view helps detect defects, such as cracks or water damage, that may not be visible from the ground. It also enables inspectors to assess the overall condition and identify potential issues that may require further investigation.
3. *Rapid and Cost-Effective:* Drones can cover larger areas quickly, resulting in faster inspection times compared to traditional methods. This efficiency translates into cost savings for building



Figure 1.1 Building inspection using UAVs

owners or inspection agencies, as fewer resources and manpower are required to complete the inspections.

4. *Enhanced Data Collection:* Drones equipped with various sensors and cameras can capture detailed and accurate data [3], including high-resolution imagery, thermal imaging, and 3D mapping. This data can be analyzed and processed to generate detailed reports, identify defects, measure dimensions, and track changes over time. It provides a more comprehensive and objective assessment compared to visual inspections alone.
5. *Safety and Risk Mitigation:* By deploying drones, inspectors can minimize the risks associated with working at heights, navigating hazardous areas, or accessing inaccessible spaces. This improves overall safety for the inspection team and reduces potential accidents or injuries.
6. *Documentation and Reporting:* Drone inspections produce visual evidence in the form of images, videos, or thermal data, which can be documented and included in inspection reports. These visual records provide a clear and visual representation of the building's condition, facilitating better communication with stakeholders, such as building owners, contractors, or insurance companies.

Overall, drones offer significant advantages in terms of accessibility, efficiency, data collection, safety, and documentation, making them valuable tools for building inspection and surveillance when combined with traditional methods, ultimately enhancing the effectiveness and accuracy of the inspection process.

1.2 Depth vision

The motivation behind using depth cameras [4] in UAVs for building surveillance stems from the desire to enhance the effectiveness and capabilities of aerial inspections. These specialized camera technologies offer unique advantages in capturing valuable data for building assessments.

1. *Structural Analysis*: Depth cameras, such as LiDAR (Light Detection and Ranging), provide detailed 3D information about the building's geometry and structure. This data allows for precise measurements, identification of deformations, and assessment of structural integrity [5]. It enables inspectors to detect subtle changes, identify potential weaknesses, and make informed decisions regarding maintenance or repairs.
2. *Point Cloud Generation*: Depth cameras generate point cloud data, representing the spatial distribution of points in the building's environment. This data can be utilized for advanced analysis, modeling, and simulation purposes. Point clouds facilitate the creation of accurate 3D models [6] of the building, aiding in planning, design, and visualization.
3. *Obstacle Detection and Avoidance*: Depth cameras enable real-time obstacle detection and avoidance during UAV flights. By perceiving the surrounding environment in 3D, drones equipped with depth cameras can navigate complex spaces, avoiding collisions with structures or other objects. This enhances the safety and maneuverability of the UAV during building surveillance missions.

The inspiration behind using depth cameras in UAVs for building surveillance is to augment inspection capabilities, enabling more comprehensive assessments, early anomaly detection, and enhanced safety measures. These camera technologies empower inspectors and decision-makers with valuable data for effective maintenance, energy efficiency improvements, and the overall well-being of buildings and their occupants.

1.3 Motivation

The motivation behind using UAVs for building inspection stems from the need to enhance inspection processes and overcome the limitations of traditional methods. Drones provide several advantages, such as improved safety by eliminating the need for inspectors to physically access hazardous or hard-to-reach areas. They also offer accessibility to remote or challenging locations and reduce operational costs through streamlined workflows. Moreover, drones enable the collection of accurate and detailed data, aiding in comprehensive analysis and decision-making. The integration of UAVs in building inspection aligns with the goal of optimizing efficiency, accuracy, and overall effectiveness in assessing the condition and integrity of structures.

Our research work aims to solve the problem of inspection of the buildings using Unmanned Aerial Vehicle (UAVs). The major limitation of traditional techniques are that they are time consuming, costly

and not safe for people climbing the buildings for inspection. We propose a UAV based solution to solve the inspection problem. In this, autonomous drone navigation module is developed for path planning with obstacle avoidance. The images captured are processed using computer vision and deep learning based methods. We detect façades in the RGB images to understand the quality of the building. We perform superresolution on thermal images which can help in the jight inspection of the building and zoom in the images to analyze the scenario more. Also, the images are used to build a virtual 3D model of the complete building using 3D graphics and computer vision. The detected façades on the 2D images are mapped on the constructed 3D model to understand it in a 3D space with better visualization. In a nutshell, a complete pipeline with various methods involved in the modules is developed.

1.4 Contribution

The significant contributions of our work are as follows:

1. We provide an automated, periodic, accurate and economical solution for the inspection of such buildings on real-world images.
2. We propose a novel end-to-end integrated autonomous pipeline for building inspection which consists of three modules: i) Autonomous Drone Navigation, ii) Façade Detection, and iii) Model Construction.
3. Our experimental analysis shows the promising performance of i) our crack detection model with a precision and recall of 0.95 and mAP score of 0.96; ii) our 3D reconstruction method includes finer details of the building without having additional information on the sequence of images; and iii) our 2D-3D mapping to compute the original location/world coordinates of cracks for a building.
4. A comparative study of selected deep learning super-resolution models and a real-time performance was achieved using less data. Constructing and optimizing an end-to-end inference pipeline using cutting edge technology to integrate the whole workflow. We have also experimented the entire pipeline on our custom thermal dataset.
5. As a consequence, the chosen superresolution model was able to achieve a real-time speed of over 29, 36 and 45 high FPS; 32.9dB/0.889, 31.86dB/0.801 and 30.94dB/0.728 PSNR/SSIM values for 2x, 3x and 4x scaling factors respectively.

1.5 Thesis Organization

The thesis is organized as follows:

1. Chapter 2 details the literature review for building inspection and superresolution for surveillance. Further the existing techniques and proposed techniques for the same are discussed.
2. Chapter 3 details about the Autonomous Inspection.
3. Chapter 4 details about the Façade Detection used to detect cracks in the building. This chapter gives the detection module of the building inspection pipeline.
4. Chapter 5 presents the Experiments and Results.
5. Chapter 6 details about the superresolution techniques.
6. Chapter 7 provides the Conclusion and Future Works.

Chapter 2

Related Works

Traditional inspection methods refer to the conventional techniques used to assess the condition, quality, or integrity of various structures or systems. These methods often involve physical presence and manual examination by human inspectors. Some common traditional inspection methods include visual inspection, manual measurements, non-destructive testing (NDT) techniques, and structural analysis. While traditional inspection methods have been effective to a certain extent, they have limitations that can be addressed by drone inspection. So, we shift to drone inspection as a smart solution for building inspection.

2.1 Building Inspection

Building inspections are essential for ensuring the safety of occupants, identifying code violations, and detecting potential hazards. They help maintain compliance with regulations, prevent costly repairs by addressing maintenance needs early on, and provide valuable information for property valuation, legal purposes, and renovation projects. They can be further divided classified as:

1. Traditional methods
2. Modern methods

2.1.1 Traditional Methods

For decades in the infrastructure industry, the economic setting has promoted new construction favouring suburban growth. Now, the main focus is on the maintenance and rehabilitation of existing structures [2]. Building structures suffer from bending, buckling, compressive and tensile failures. There are high chances of such failures in buildings due to their age, density and altitude. To prevent structural collapse, casualties and economic loss, periodic maintenance is essential for the safety of buildings to increase their durability and lifespan. The inspection requires heavy machinery, lifts, field professionals and people rappelling from dangerous heights, which is labour-intensive and time-consuming. Additionally, professionals traditionally inspect the buildings using climbing gear, swing

stages and access tools. They further spend on insurance and labour, which increases the inspection cost. Even after all these efforts and expenses, some surfaces' comprehensive details are unavailable due to the inaccessibility to some parts of surfaces.

Traditionally, building inspections are carried out through a systematic and visual examination of the building's components. A building inspector visits the site and inspects the structural elements, electrical systems, plumbing, HVAC, fire safety features, and accessibility to ensure compliance with building codes and standards. They document their findings, identify any violations or deficiencies, and provide recommendations for corrective actions. The process involves scheduling, site visits, thorough examinations, documentation, and follow-up inspections if needed, with the aim of ensuring the safety, functionality, and compliance of the building.

Need for UAV inspection Although traditional inspection methods have shown effectiveness to some degree, they possess limitations that can be overcome through the utilization of drone inspection. Some reasons why drone inspection is gaining popularity as a replacement for traditional methods are safety, accessibility, cost-efficiency, speed, data accuracy, repeatable comparable result, tedious documentation and reporting. While drone inspection offers numerous advantages, it is important to note that it may not completely replace all traditional methods in every scenario. Some inspections may still require human intervention, tactile inspections, or specialized equipment. Nonetheless, the integration of drone technology in inspection processes has proven to be a valuable addition, enhancing safety, efficiency, and accuracy in various industries.

2.1.2 Modern Methods

In the current era, Unmanned Aerial Vehicles (UAVs) can accurately, efficiently and economically perform a wide range of surveying applications [2] such as transmission-line inspection [7], power-line inspection [8], underwater area inspection [9], tree cavity inspection [10], industrial plants [11], bridge inspection [12], and dam inspection [13] for Structural Health Monitoring (SHM). To keep professionals out of danger, UAVs can provide aid for building inspection. This can reduce operational costs, human error, safety risks and time taken for the inspection. UAVs can fly and scan the entire structure for evaluation, owing to their mobility and ability to capture footage by applying appropriate path planning for the whole structure [14].

Classical methods [15] comprise detecting cracks manually which is painstakingly time-consuming and is biased by the subjective judgment of the inspectors. With the advent of computer vision in the past decade, the traditional surveying task can be aided by such techniques for insightful and accurate inspections [16]. Crack detection [17] has gained popularity due to the adverse effects of buildings not being monitored periodically. Computer Vision methods [18] [19] and physical interaction methods [11] [20] have been assisting in inspection to have economic and periodic maintenance. The footage captured by the UAV can also be used in Building Information Modeling (BIM) [21] for 3D rendering [14]. This serves as an aid to the Architecture, Engineering and Construction (AEC) industry. The benefit of

working on a 3D model over traditional drafting is to allow the building inspectors to understand the condition of the façades.

The aforementioned methods implement a particular stage or a few stages of the building inspection problem. We propose a novel end-to-end pipeline for high-rise building inspection, as shown in Fig. 5.1. Our first module generates the trajectory for the UAV. Then, collision avoidance ensures that the UAV reaches the destination despite encountering obstacles by recomputing its trajectory on the fly. During the flight, the UAV captures images of the building based on camera and UAV parameters. The images captured are used for crack detection and to reconstruct a 3D mesh model of the building and mark the detected crack correspondences on the model. Fig. 5.3 shows an example of a 3D mesh model of the building constructed with our proposed pipeline using real-world images, the trajectory of drone navigation and a detected crack. Our major contributions are as follows: i) To the best of our knowledge, we are the first to propose an automated pipeline for building inspection; ii) We propose a novel end-to-end framework for high-rise infrastructure inspection for façade detection; iii) We also integrate the entire pipeline for autonomous inspection using UAVs; iv) We create a dataset and empirically study real-world building data for crack detection, 3D reconstruction and respective mapping.

Chapter 3

Autonomous Inspection

3.1 Introduction

The need for autonomous inspection of buildings has arisen as a response to the limitations of traditional manual inspection methods. Autonomous inspection refers to the use of advanced technologies, such as robotics, artificial intelligence, and computer vision, to perform inspections without direct human intervention. This approach offers several advantages, including increased safety, efficiency, and accuracy in assessing the condition, maintenance needs, and compliance of buildings. By harnessing the power of automation, autonomous inspection is transforming the way we evaluate and manage the built environment.

3.2 Theory

3.2.1 Health Inspection of buildings

Health inspections of buildings, particularly regarding issues like cracks and delamination, are crucial for assessing the structural integrity and identifying potential hazards.

1. *Visual Inspection:* Building inspectors conduct visual inspections to identify visible signs of cracks and delamination. They carefully examine building components such as walls, floors, ceilings, and foundations to look for cracks, separation, or peeling of materials. Inspectors may use flashlights, magnifying glasses, or other tools to aid in the examination.
2. *Crack Detection:* Inspectors look for cracks in various building elements, including concrete, masonry, plaster, and wood. They assess the size, length, width, and pattern of the cracks. Common types of cracks include vertical, horizontal, diagonal, or stair-step patterns, which can indicate different underlying issues such as foundation settlement, structural movement, or moisture-related problems.

3. *Non-Destructive Testing*: In some cases, non-destructive testing methods may be employed to assess the extent and severity of cracks and delamination. Techniques like ground-penetrating radar (GPR) or infrared thermography can provide additional insights into hidden defects or structural abnormalities.
4. *Material Delamination Assessment*: Delamination refers to the separation or detachment of layers in building materials, such as concrete, stucco, or coatings. Inspectors examine surfaces for signs of flaking, spalling, or peeling, which can indicate delamination. They may also use tapping or sounding techniques to identify hollow or delaminated areas within materials.
5. *Structural Assessment*: Inspectors evaluate the impact of cracks and delamination on the overall structural stability of the building. They consider factors such as the location, size, and progression of the issues. Structural calculations, load testing, or consulting with structural engineers may be necessary to assess the severity of the problem and determine appropriate corrective actions.
6. *Documentation and Reporting*: Inspectors document their findings, including descriptions, photographs, measurements, and recommended actions. They provide detailed reports to building owners, managers, or relevant authorities, outlining the identified issues and suggesting necessary repairs, maintenance, or further investigations.

It's important to involve qualified professionals, such as structural engineers or certified building inspectors, for comprehensive and accurate assessments of cracks, delamination, and other potential structural issues in buildings. They have the expertise to analyze and provide appropriate recommendations based on the specific conditions of the building.

3.2.2 Inspection Methods

Building inspection methods utilizing computer vision and machine learning techniques are revolutionizing the field of inspection, enabling automated analysis and decision-making. Here are several examples of these methods:

1. *Object Recognition*: Computer vision algorithms can be employed to identify and classify objects within building environments. For instance, they can detect and categorize structural components, such as beams, columns, or walls, aiding in the assessment of their condition and identifying any defects or abnormalities.
2. *Defect Detection*: Computer vision algorithms combined with machine learning techniques can automatically detect and localize defects or anomalies in buildings. By analyzing visual data, such as images or videos captured during inspections, these algorithms can identify cracks, corrosion, water damage, or other structural issues that may require attention.

3. **Semantic Segmentation:** Semantic segmentation algorithms enable the partitioning of images into meaningful regions or segments. In building inspections, this technique can be used to segment different building elements, such as roofs, facades, or windows. It helps in quantifying and analyzing specific areas of interest and facilitates condition assessment or maintenance planning.
4. **Structural Health Monitoring:** Machine learning algorithms can be utilized for structural health monitoring, where sensors installed within buildings collect data on vibrations, strain, or other structural parameters. Machine learning algorithms process and analyze this data to detect patterns or deviations that indicate potential structural issues, allowing for proactive maintenance and avoiding catastrophic failures.
5. **Change Detection:** By comparing images or 3D point clouds acquired from different time points, machine learning algorithms can detect and quantify changes in the building's condition over time. This approach helps monitor deterioration, track maintenance progress, or identify unauthorized modifications in real estate or construction projects.
6. **Automated Reporting:** Machine learning techniques can assist in automating the generation of inspection reports. By analyzing inspection data, images, and sensor readings, algorithms can extract relevant information, highlight critical findings, and generate comprehensive reports with minimal human intervention. This streamlines the reporting process, facilitates decision-making, and improves communication among stakeholders.

These building inspection methods employing computer vision and machine learning leverage the power of data analysis, pattern recognition, and automation to enhance the speed, accuracy, and efficiency of inspections. By automating certain tasks and augmenting human capabilities, these methods contribute to better building management, maintenance planning, and overall safety. In this work, we focus on inspecting the crack on the building.

3.3 Autonomous Drone Navigation

Autonomous drone navigation refers to the ability of drones to navigate and fly without direct human control, utilizing onboard sensors, algorithms, and artificial intelligence. It enables drones to autonomously plan their flight paths, avoid obstacles, and make real-time adjustments to ensure safe and efficient navigation. This technology empowers drones to operate independently, opening up a wide range of applications in areas such as surveillance, mapping, delivery, and inspection. Our autonomous drone navigation system is explained below.

3.3.1 Trajectory Planning

For inspection, we develop an autonomous trajectory plan to scan the building under inspection from bottom to top. Our algorithm takes the corner coordinates, $\{(x_i, y_i) | i = 1, 2, 3, 4\}$ and the height, h_b

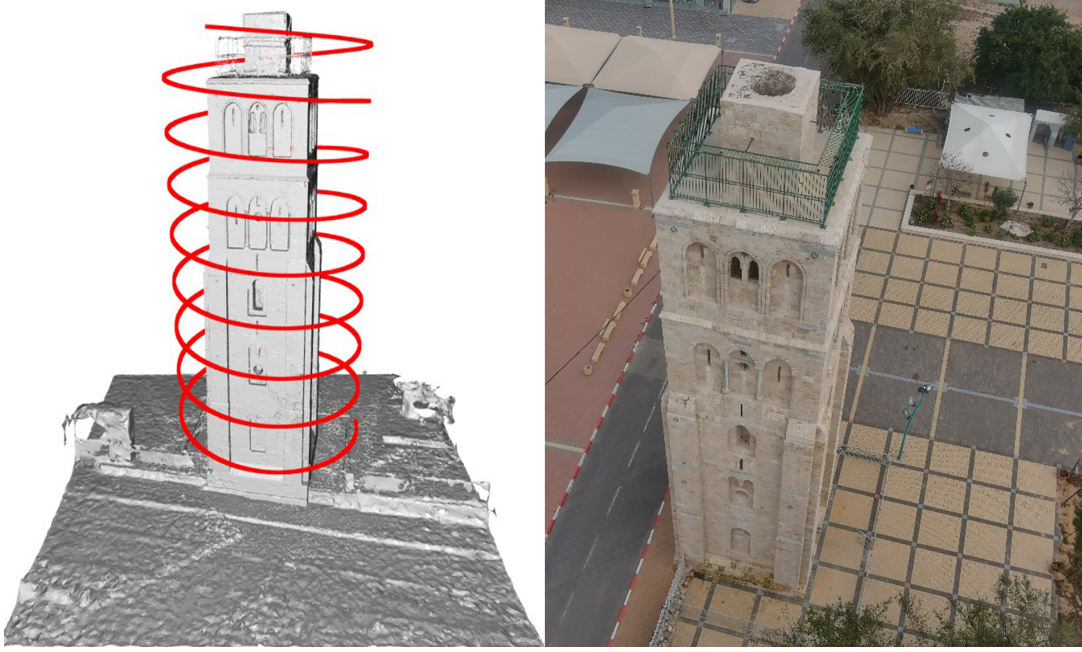


Figure 3.1 Our reconstructed 3D mesh model (left) of the high-rise building (right). The proposed pipeline reconstructs the 3D model from the real-world images captured by UAV, where the red spiral path marks the trajectory followed by the UAV.

of the building as input to plan the trajectory. The UAV trajectory is planned in an elliptical spiral path around the building. In 3.2, the inner rectangle represents the building ground plane. We take a distance, s from the building, as the drone should be at a minimum safety distance to capture images. The rectangle $ABCD$ maintains this distance, s from the building. For minimum distance, we plan an ellipse spiral trajectory passing through $ABCD$. 3.2 represents the 2D projection of the trajectory. Equation (3.1) shows the equation of an ellipse with center (h, k) and radii a and b .

Consider a circle around a square, where the circle touches all four corners of the square. Here, the radius, r of the circle, is the same as the semi-diagonal of the square. If we squeeze the square from either of the sides, we get a rectangle. As an effect, the circle gets squeezed into an ellipse keeping the ratios the same. Therefore, in equation (3.2), we get the radii, a and b , of the base ellipse for the spiral trajectory path.

$$\frac{(x - h)^2}{a^2} + \frac{(y - k)^2}{b^2} = 1 \quad (3.1)$$

$$a = \frac{x_{AB}}{\sqrt{2}} \quad b = \frac{y_{BC}}{\sqrt{2}} \quad (3.2)$$

$$x_{AB} = (x_2 - x_1) + 2s \quad y_{BC} = (y_3 - y_2) + 2s \quad (3.3)$$

$$a = \frac{(x_2 - x_1 + 2s)}{\sqrt{2}} \quad b = \frac{(y_3 - y_2 + 2s)}{\sqrt{2}} \quad (3.4)$$

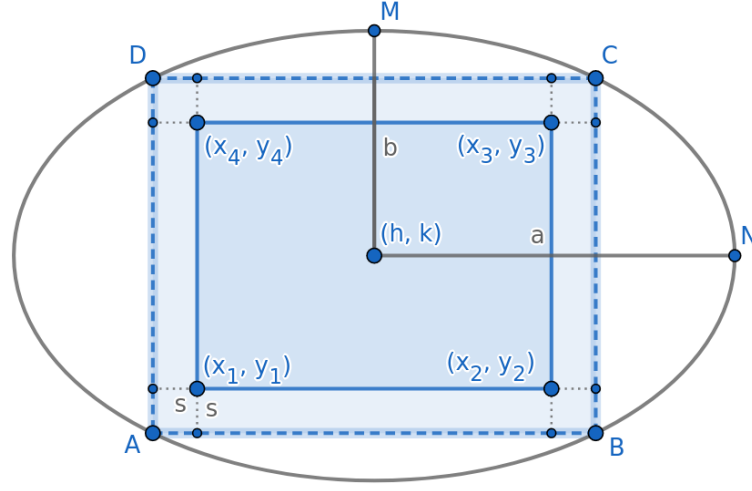


Figure 3.2 2D projection of Trajectory Planning on the XY plane. The top view of the building is represented by the inner rectangle and ellipse trajectory followed by the UAV maintaining a safe distance, s , from the building.

Combining equations (3.1) and (3.4), we get the equation of an ellipse which is used by the UAV to form a spiral path till the height, h_b . Along with the radii of the ellipse, the pitch of the spiral path is needed to decide the final trajectory around the building. The pitch is decided based on the distance, s and camera parameters, which include the field of view, aperture, focal length, image resolution and ISO of the camera.

3.3.2 Collision Avoidance

To consider different types of protrusions of a building, like balconies, antennas and other possible obstacles, we need collision avoidance [22]. Such obstacles must first be detected for complete autonomous drone navigation, and the UAV needs to be guided through a new optimized path to reach its goal. Thus, collision avoidance can be divided into Obstacle Detection and Trajectory Optimization.

3.3.2.1 Obstacle Detection

A depth camera is attached to the UAV to detect upcoming obstacles [22]. We capture the positions of the obstacles in the environment as point clouds obtained from a depth camera. The point clouds are then down-sampled to reduce the number of points for processing and to denoise the data. Object clusters are extracted from the reduced point cloud, and their positions are determined.

3.3.2.2 Trajectory Optimization

This module is responsible for computing a collision-free trajectory towards a given goal position. The motion model of the UAV is given as follows:

$$\dot{x}_t = v_{x_t} \quad \dot{y}_t = v_{y_t} \quad \dot{z}_t = v_{z_t} \quad (3.5)$$

$$\ddot{x}_t = a_{x_t} \quad \ddot{y}_t = a_{y_t} \quad \ddot{z}_t = a_{z_t} \quad (3.6)$$

Here, $(v_{x_t}, v_{y_t}, v_{z_t})$ represents the velocity of the UAV in x, y, and z directions and $(a_{x_t}, a_{y_t}, a_{z_t})$ represents the acceleration of the UAV in x, y, and z directions. Given the motion model, we assume that there is a low-level controller for the UAV, which takes the next waypoint, velocity and acceleration as input and generates control commands for the UAV to reach that position. Our trajectory optimizer can be defined as follows:

$$\min_{a_{x_t}, a_{y_t}, a_{z_t}} \sum C_g + C_a \quad (3.7)$$

$$a_{min} \leq a_{x_t}, a_{y_t}, a_{z_t} \leq a_{max} \quad (3.8)$$

$$(x_t - x_{o,t})^2 + (y_t - y_{o,t})^2 + (z_t - z_{o,t})^2 > (\lambda_o)^2 \quad (3.9)$$

$$f_l(x_t, y_t, z_t) > 0 \quad (3.10)$$

$$C_g = (x_{t_f} - x_{goal})^2 + (y_{t_f} - y_{goal})^2 + (z_{t_f} - z_{goal})^2 \quad (3.11)$$

$$C_a = (a_{x_t})^2 + (a_{y_t})^2 + (a_{z_t})^2 \quad (3.12)$$

Here, the cost function, C_g given by equation (3.11), minimizes the distance of the trajectory end position to the goal, and the cost function, C_a given by equation (3.12), minimizes the acceleration magnitude at each time instant, for maintaining the smoothness of the trajectory. The inequality constraint in equation (3.8) limits the accelerations to their maximum and minimum bounds, equation (3.9) enforces collision avoidance of the i^{th} obstacle with the UAV by enforcing the euclidean distance between them to be greater than the threshold λ_o and equation (3.10) ensures that the trajectory does not cross the lane boundaries defined by the building walls, which is represented by the function, f_l . Equation (3.10) ensures that the UAV remains at a safe distance from the building. Here, f_l is an equation representing the shape of the building which can be a linear equation, quadratic equation, etc parametrized by x_t , y_t and z_t .

3.3.3 Image Capturing

While the UAV moves around the building, as shown in Fig. 5.3, a camera facing the building is dedicated for capturing images. These images are captured by keeping the camera settings constant throughout the flight of the UAV. This helps in keeping the intrinsic parameters of the camera identical for all the images of a particular building. While capturing the images, the frame rate of the camera is set such that the overlap in the consecutive images is at least 60% to 70%. A sample of the images used



Figure 3.3 A random sample of images captured of the high-rise building by the UAV.

for simulation is shown in Fig. 3.3. The total number of images captured of the building is 2123 for the 3D model construction, of which 175 images contain cracks.

The resolution of the images is chosen such that the façades of the building are clearly visible. The visibility is based on the distance of the UAV from the building, which is considered during the trajectory planning of the UAV (Section 3.3.1). The camera used for this purpose should not have an active auto-focus facility as the camera's intrinsic parameters change when the camera auto-focuses separately for all the images. Our pipeline requires the camera intrinsics for all the images of a particular building to be the same for 3D construction, elaborated in Section 3.4.1.

3.4 Model Construction

Constructing a 3D model for building inspection tasks is essential for several reasons. Firstly, it provides a comprehensive representation of the building's geometry, enabling accurate spatial analysis and identification of structural components. Secondly, it facilitates virtual exploration of inaccessible areas, aiding in the detection of defects or anomalies. Lastly, a 3D model serves as a valuable reference for documentation, collaboration, and future maintenance planning, enhancing the overall efficiency and effectiveness of building inspections.

3.4.1 3D Reconstruction

Now, we have the images captured from the UAV. We use these images for constructing a 3D model of the building. This model helps in the visualisation of the building. We use COLMAP [23] [24] for 3D reconstruction.

COLMAP (Structure-from-Motion and Multi-View Stereo COLonel MAPper) is an open-source software package used for computer vision tasks, specifically for 3D reconstruction from images. It provides a collection of algorithms and tools to reconstruct the 3D structure of a scene from a set of 2D images. The main functionalities of COLMAP include:

1. *Structure-from-Motion (SfM)*: COLMAP performs the estimation of camera poses and 3D structure of a scene from a collection of images. It uses a bundle adjustment algorithm to refine the camera poses and optimize the 3D point positions based on feature correspondences between images. SfM is the process of recovering the camera poses and sparse 3D structure from a set of unordered images.
2. *Dense Reconstruction*: COLMAP can perform dense reconstruction, also known as Multi-View Stereo (MVS), to estimate the depth and surface geometry of the scene. It uses the initial camera poses and sparse 3D structure obtained from the SfM stage to generate a dense 3D point cloud or mesh representation of the scene.
3. *Image Undistortion*: COLMAP includes tools to handle lens distortion in images and correct it for more accurate reconstruction. Lens distortion, caused by imperfections in camera lenses, can introduce geometric distortions in images that can affect the accuracy of the reconstruction process. COLMAP provides methods to estimate and remove lens distortion effects.
4. *Image Alignment and Matching*: COLMAP offers feature detection, description, and matching algorithms to find corresponding points across images. These correspondences are essential for the SfM and MVS stages to establish correspondences between images and estimate camera poses and 3D structure.
5. *Texturing*: After the reconstruction process, COLMAP allows for the texturing of the generated 3D models. It can project the images onto the 3D geometry, creating a textured representation of the scene.

COLMAP is widely used in academic research, computer vision, and photogrammetry applications. Its open-source nature and extensive feature set make it a popular choice for 3D reconstruction tasks. It supports various image formats, camera models, and export formats for 3D models. Additionally, COLMAP provides a command-line interface as well as a graphical user interface (GUI) for ease of use and accessibility.



Figure 3.4 3D mesh model of the building (left) using 2D to 3D mapping shown using the red ray and red ring; Captured image of the building (right) using UAV showing 2D image of it. Correspondences of all the three images are shown using blue lines.

3.4.1.1 Camera Parameter Extraction

For constructing the 3D model, the intrinsic and extrinsic camera parameters are required. These are calculated using a standard traditional method called COLMAP. COLMAP is a Structure-from-Motion (SfM) and Multi-View Stereo (MVS) method which takes multi-view images as input. It uses the overlap patches in distinct unordered images for feature matching. This gives the camera settings while capturing the images (intrinsic camera parameters) and the camera locations for all the images (extrinsic camera parameters). These camera parameters are used for the reconstruction of the building.

3.4.1.2 Reconstruction

Firstly, the camera parameters generated above are used to reconstruct a dense point cloud using COLMAP. Then, Poisson-disk sampling [25] is applied to sub-sample the generated point cloud. Normals are then computed for each of the remaining points for constructing the 3D mesh model of the object using Marching Cubes APSS algorithm [26] [27]. This gives a fine triangular mesh of the building. This fine high-density mesh is used to produce accurate results as it has more number of faces per

unit area. Different views of the reconstructed 3D mesh model are shown in Fig. 3.4, Fig. 5.3, Fig. 5.4 and Fig. 5.5.

3.4.2 2D to 3D Mapping

The pixels in the 2D image can be mapped on the 3D model generated in Section 3.4.1

for easier understanding and better visualization of the 2D image patches on the 3D model which is described as follows:

3.4.2.1 Image correction

The 2D images captured in section 3.3.3 as shown in Fig. 3.3 have image distortion where images appear to be curved or deformed due to lens aperture. This is known as image distortion, which is majorly of two kinds: tangential distortion and radial distortion. Tangential distortion is caused due to misalignment of the camera lens with respect to the parallel image plane. This results some areas to appear nearer than expected and can be corrected using equations (3.13) and (3.14). In radial distortion, the straight lines in an image appear to be curved and as the point in the real world is farther from the image center, the radial distortion increases. This needs to be corrected before working on images, and can be corrected using equations (3.15) and (3.16).

$$x_{tangential} = x + [2p_1xy + p_2(r^2 + 2x^2)] \quad (3.13)$$

$$y_{tangential} = y + [p_1(r^2 + 2y^2)2p_2xy] \quad (3.14)$$

$$x_{radial} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (3.15)$$

$$y_{radial} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (3.16)$$

So, here we find that five parameters are required for distortion correction. Distortion coefficients are given by $coe f_{distortion}$ as follows:

$$coe f_{distortion} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) \quad (3.17)$$

The distortion coefficients are obtained from camera intrinsic parameters, which we extract in Section 3.4.1 from the images captured in Section 3.3.3. Camera intrinsic parameters also include information like focal length (f_x, f_y) and optical centers (c_x, c_y) given in equation (3.18). There are camera extrinsic parameters which provide the rotation parameters, r_{ij} and translation parameters, t_i of the images as shown in equation (3.18). Thus, the final projection of the image after correcting the image distortion is given by λ as follows:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.18)$$

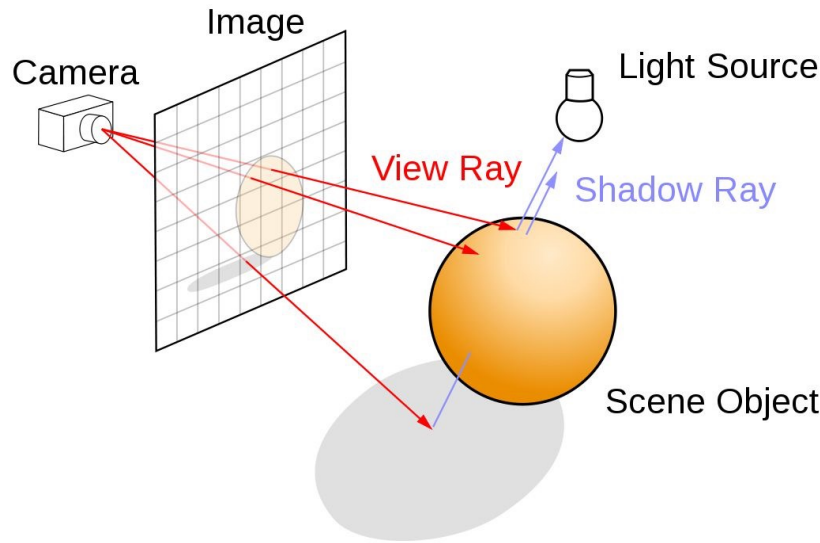


Figure 3.5 Ray Tracing

In stereo applications, it is necessary to correct image distortions before proceeding with further processing. To achieve this, a set of sample images featuring a clearly defined pattern, such as a chessboard, is required. By identifying specific points with known relative positions, such as the corners of the squares on the chessboard, it becomes possible to determine the distortion coefficients. These coefficients are derived by comparing the known real-world coordinates of the points with their corresponding coordinates in the distorted image. To ensure accurate calibration, it is recommended to use a minimum of 10 test patterns.

3.4.2.2 Ray Tracing

Ray tracing is a rendering technique used in computer graphics to simulate the behavior of light in a virtual scene as shown in Figure. 3.5. It models how light rays interact with objects, surfaces, and the environment to produce realistic and visually accurate images. Traditional rendering techniques, such as rasterization, calculate the color of each pixel on a screen by projecting objects onto a 2D plane and determining their visibility. While rasterization is fast and widely used, it has limitations in accurately simulating complex lighting effects and realistic reflections. In contrast, ray tracing works by tracing the path of light rays backwards from the viewer's eye or camera through the virtual scene. It simulates the physical properties of light, including reflection, refraction, and shading. By accurately modeling how light interacts with objects and surfaces, ray tracing can create more lifelike and visually appealing images. Here's a simplified overview of how ray tracing works:

1. *Ray Generation:* Rays are cast from the camera or eye position through each pixel on the image plane. These rays represent the path that light would take if it were emitted from the viewer's eye.

2. *Intersection Tracing*: Each ray is traced for intersection with objects in the scene, such as polygons or primitives. This is typically done using bounding volumes or acceleration structures like bounding boxes or BVH (Bounding Volume Hierarchy) to speed up the process.
3. *Surface Interaction*: When a ray intersects an object, it computes the interaction between the ray and the object's surface. This involves calculating lighting effects, such as shadows, reflections, and refractions, based on the properties of the material and the scene's lighting.
4. *Recursive Ray Tracing*: For reflective or refractive surfaces, additional rays are generated to simulate the reflected or refracted light. These rays follow the same process of intersection testing and surface interaction, potentially generating more rays recursively.
5. *Shading and Illumination*: The color of each pixel is determined by combining the contributions from different light sources, reflections, and other factors. Complex lighting effects, such as soft shadows, global illumination, and caustics, can be simulated through ray tracing.

Ray tracing is computationally intensive and requires significant processing power, especially for scenes with a high level of complexity and realistic lighting effects. However, advancements in hardware and algorithms, including the use of dedicated GPUs and acceleration techniques, have made real-time ray tracing possible in some applications. Ray tracing has become increasingly popular in computer graphics, gaming, and visual effects industries, as it offers improved realism, lighting accuracy, and the ability to create visually stunning images and animations. It is commonly used in applications such as movie rendering, architectural visualization, product design, and virtual reality experiences.

The 2D image patches can be visualized and mapped on the 3D model generated for a better understanding of its locations. This is implemented using Ray Tracing. This method requires the camera intrinsic and extrinsic parameters extracted using COLMAP in Section 3.4.1 and the corresponding images. Considering the origin of the rays to be at the location of the camera, the rays are extrapolated to the points in the 3D model the desired pixel of the 2D image. Multiple rays are projected from the camera location towards the scene. The ones which pass through our desired pixel in the image are visualised from the camera to the the 3D model of the building.

Chapter 4

Façade Detection

4.1 Introduction

Regular inspections and condition assessments are essential to ensure the functionality and structural integrity of buildings and civil infrastructures. With a growing number of older buildings, the continuous exposure of façades to harsh outdoor environmental conditions accelerates their deterioration [28]. Anticipated consequences include an increase in façade defects and incidents involving falling objects from heights, posing significant public safety concerns [29]. Consequently, the inclusion of structural health monitoring has become a crucial component of façade condition assessments. This is due to the potential risks posed by falling objects from tall buildings, which can cause harm to the public and necessitate the evaluation of structural safety measures [30]. Regular monitoring and inspections of buildings are essential to ensure the proper safety and security of their structural components [31]. Consequently, there is a need for acquiring new insights regarding the types and properties of falling objects from façades, the key factors influencing their occurrence, and the efficacy of different inspection methods.

4.2 Façade Defects and Inspection Practices

Various types of façades can exhibit a range of common defects and anomalies. Inspection methods are selected based on the specific type of façade defect detected.

4.2.1 Types of Façades

In the world of construction and architecture, defects in buildings can encompass a wide range of issues, from minor cosmetic imperfections to structural weaknesses that pose significant risks. Some of the types of façades are mentioned in Table 4.1.




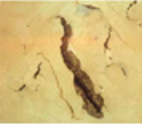











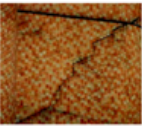




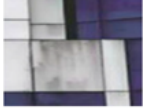


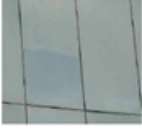




Type of Facade	Example			
Concrete				
Brick Masonry				
Plaster				
Tile				
Stone Cladding				
Metal Cladding				
Glass Cladding				

Table 4.1 Common defects and anomalies from different types of façades

4.2.2 Façade Defects and Anomalies

The condition and functionality of building façades are influenced by both the physical properties of the construction materials and the surrounding environment. In Table 4.1, various types of façades are examined, highlighting the common defects and anomalies that can lead to falling objects from tall buildings. These issues encompass cracking, water penetration, misalignment, discolouration, efflorescence, corrosion, and more. Concrete is frequently used in façade construction, and it is prone to surface defects such as cracking, spalling, biological growth, drying shrinkage, and delamination, all of which can result in falling objects. Researchers have explored different sensing techniques to localize and quantify concrete cracking and spalling defects [32, 33].

Additional façade materials like brick masonry, plaster, and tiling can also contribute to falling object hazards. Specifically, in tropical climates characterized by elevated temperatures and humidity, these materials are prone to defects such as cracking, rising dampness, biological growth, efflorescence, and delamination, which further exacerbate the risk of falling objects[34]. However, there is a notable gap in the existing literature regarding the inclusion of design and maintenance considerations for façade components during the initial planning phase. Another significant source of potential fatal falling objects is cladding, which encompasses stone cladding, metal cladding, and glass cladding. The primary causes for such incidents involve damage and cracking of façade materials, failures in joints or connections, and insufficient design and maintenance of the support system. Research investigations have revealed that casement windows account for 80% of fallen windows, primarily due to factors such as corrosion of aluminium rivets, as well as improper design, installation, and maintenance practices [29]. Therefore, there is a research necessity to enhance the identification and categorization of prevalent façade defects and anomalies.

4.3 Crack Detection

Crack detection refers to the process of identifying and locating cracks or fractures in various materials or structures. Cracks can occur in a wide range of materials, such as metals, concrete, ceramics, and composites, and they can pose significant risks, including structural failures, reduced performance, and safety hazards.

4.3.1 Challenges

Crack detection can present various challenges depending on the material, the type and size of cracks, and the specific detection method being used. Following are some common challenges in crack detection.

1. *Crack Size and Visibility:* Cracks can vary in size, from microscopic cracks to larger, more visible ones. Detecting small or hairline cracks can be particularly challenging, as they may be difficult to spot visually or require specialized equipment for detection.

2. *Surface Condition*: The condition of the material's surface can impact crack detection. Surface roughness, coatings, paint, or corrosion can obscure or hide cracks, making them harder to detect. Preparing the surface properly before inspection is crucial to ensure accurate crack detection.
3. *Crack Orientation and Geometry*: Cracks can occur in different orientations and have various geometries, such as straight, curved, or branching cracks. Detecting and characterizing cracks with complex shapes can be more challenging, as the crack may not be easily visible or may require multiple inspection angles.
4. *Material and Environmental Factors*: Different materials have different properties and response to crack detection methods. Some materials may be more prone to false positives or false negatives. Additionally, environmental factors, such as temperature, humidity, and electromagnetic interference, can affect crack detection accuracy and reliability.
5. *Depth and Subsurface Cracks*: Some cracks may be located below the surface or within the material, making them more difficult to detect using conventional methods. Specialized techniques like ultrasonic testing or radiographic testing may be required to identify subsurface cracks accurately.
6. *False Positives and False Negatives*: Crack detection methods may occasionally produce false results. False positives occur when a defect is wrongly identified as a crack, leading to unnecessary repairs or maintenance. False negatives happen when cracks go undetected, potentially leading to safety risks or structural failures.

Addressing these challenges often involves a combination of different crack detection techniques, optimization of inspection procedures, continuous research and development of advanced inspection methods, and ongoing training and expertise of inspectors.

4.4 Façade Inspection

Various façade inspection methods are chosen depending on the type of façade found on the high-rise building. The various methods for the same are explained below.

4.4.1 Visual Inspection

The existing approach involves certified inspectors visually examining the building and documenting surface defects through photographs and sketches. However, these conventional inspection methods are inadequate for gaining a comprehensive understanding of the building's condition during the review stage. To address this issue, researchers have explored the utilization of unmanned aerial vehicles (UAVs) for facilitating automated visual inspections [35].

Due to their lower payload capacity, unmanned aerial vehicles (UAVs) are less suited for carrying advanced sensing devices like Light Detection and Ranging (LiDAR) laser scanners for point cloud acquisition. As a result, unmanned ground vehicles (UGVs) or ground robots offer greater stability and are more readily equipped with such advanced sensing capabilities [36]. The obtained 2D images or 3D point clouds were subsequently employed for the purpose of detecting building defects and analyzing potential damages. This encompassed the utilization of laser scanning to identify concrete spalling defects [32], quantification of concrete surface defects using laser point clouds obtained from UAV-based systems [37] and change detection and monitoring of deformations are also performed [38]. As an example, UAV-acquired image data was employed to identify various forms of concrete cracks on buildings [33]. Research was conducted to explore the application of infrared thermography in capturing delamination defects prior to the occurrence of cracks [39]. Recent research has placed emphasis on utilizing point clouds for the purpose of quantifying building defects. Moreover, the potential of image-based 3D reconstruction was investigated to facilitate the evaluation of building conditions and assessment of damages.

4.4.2 Deep Learning Method

A building has various types of façades on the walls. These façades can be cracks, de-laminations, paint deteriorations, moulds and stains. In our work, we focus on detecting cracks using our model so that those which need attention don't get neglected due to probable human error. To build an autonomous crack detection system for high-rise buildings, various issues occur, such as the texture of the building, lighting conditions and severity of the crack. We use YOLOv5 [1] deep learning model for object detection to detect cracks with bounding boxes in our pipeline.

YOLOv5 is a deep learning model that stands for "You Only Look Once version 5." It is a state-of-the-art object detection algorithm and architecture designed for real-time object detection tasks. YOLOv5 builds upon its predecessors (YOLOv1, YOLOv2, YOLOv3) by introducing improvements and advancements in terms of accuracy, speed, and versatility. The YOLOv5 model follows a single-shot detection approach, which means it processes the entire image in a single pass to identify and localize objects. This differs from other object detection algorithms that use region proposal methods, such as selective search or region-based convolutional neural networks.

YOLOv5 employs a deep convolutional neural network architecture, typically based on a backbone network like Darknet, CSPDarknet, or EfficientNet. It consists of multiple convolutional layers, down-sampling layers, and up-sampling layers, allowing the model to extract features at different scales and levels of abstraction. The model is trained on large annotated datasets, where bounding box annotations are provided for various object classes. During training, YOLOv5 optimizes its parameters to minimize the difference between predicted bounding boxes and ground truth annotations, utilizing techniques like backpropagation and gradient descent. Once trained, YOLOv5 can be used for real-time object detection on images or videos, providing bounding box coordinates and class labels for detected objects. It offers a balance between accuracy and speed, making it suitable for applications that require fast and efficient

object detection, such as autonomous driving, surveillance, and robotics. YOLOv5 has gained popularity in the computer vision community due to its simplicity, effectiveness, and open-source nature, allowing researchers and developers to adapt and extend the model for various tasks and domains.

Chapter 5

Experiments and Results

In this chapter we will explain the various experiments which were conducted for façade detection and superresolution during autonomous drone navigation. In the below section we will first detail the pipeline built for façade detection along with autonomous drone navigation. Then we will discuss about training and inference pipeline used to deploy the complete technique on the edge device.

5.1 Façade Detection

Façade detection plays a crucial role in building inspection tasks for several reasons. Firstly, it allows for targeted assessment of the building’s exterior, enabling the identification of specific areas that require closer inspection. Secondly, it aids in detecting façade-related issues such as cracks, stains, or deterioration, ensuring timely maintenance and preservation. Lastly, accurate façade detection enhances the overall efficiency and accuracy of inspections, providing valuable insights for building condition assessment and decision-making.

5.1.1 Pipeline

This section combines multiple modules to create a unified and cohesive solution for high-rise building inspection. Firstly, we identify various modules involved in the building inspection, such as autonomous drone navigation, façade detection and model construction. Our proposed pipeline integrates these modules as shown in 5.1 and explains each framework in the following sections.

5.1.2 Implementation details

We use the Gazebo framework for simulation on a 2.25 GHz system with 16 GB RAM, an Intel i7 processor and Nvidia RTX 3060 GPU. The 3D mesh model constructed in Section 3.4.1 from the images captured using UAV, is imported into Gazebo [40].

In the simulation environment, a Pelican Quadrotor is spawned with two cameras to follow the trajectory planned in Section 3.3.1. The specifications of the both the cameras are mentioned in Table

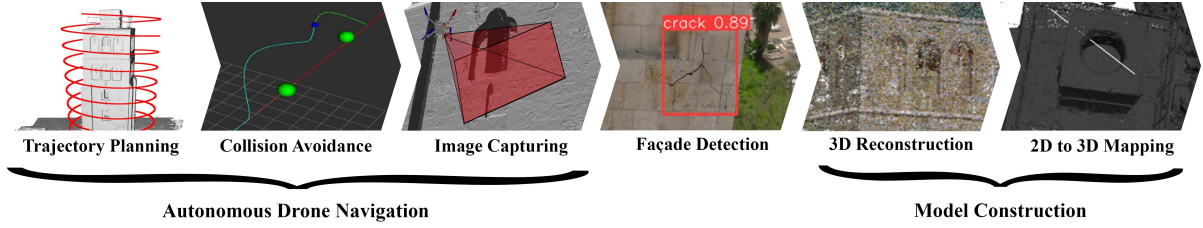


Figure 5.1 Proposed Building Inspection Pipeline consists of three modules: (1) Autonomous Drone Navigation containing (a) Trajectory Planning, (b) Collision Avoidance and (c) Image Capturing; (2) Façade Detection; (3) Model Construction containing (a) 3D Reconstruction and (b) 2D to 3D Mapping

Specifications	
Image Resolution	752 × 480
Frame Rate	30 fps
FOV	80°
Min. Depth	0.02m
Max. Depth	30m

Table 5.1 Specifications of RGB and depth camera mounted on Pelican Quadrotor used for simulation.

5.1. Both the cameras are mounted on the Pelican, orthogonal to each other where the RGB camera faces the building model to capture images and the depth camera faces towards the path to optimize its path for collision-free flight. An example of the view is shown in Fig. 5.5. Optimal control solver ACADO [41] is used for trajectory optimization in Eq. (3.7). We use Point Cloud Library (PCL) to obtain obstacle positions from the point cloud.

We annotate the cracks in the captured images for façade detection. The YOLOv5-Large model is fine-tuned on the images with cracks. For the YOLOv5-Large model, all these images are resized to 640×640 pixels. The fine-tuning takes approximately 55 minutes to train till 233 epochs with the model size of 92.8MB. The model, initially supposed to train for 500 epochs stops training early at 233 epochs for the best model. This happens as no improvement is observed in the model post 132 epochs.

The captured images are also used for 3D model reconstruction of the high-rise building as explained in Section 3.4.1. Initially, a dense point cloud of 1,384,433 points taking 20.8MB is generated. The generated triangular 3D mesh of the high-rise building contains 864,584 vertices and 871,137 triangles with a size of 126.8MB, which contains triangular faces, vertices, vertex normals and vertex textures of the generated mesh. The detected cracks are mapped on the 3D model as explained in Section 3.4.2 and shown in Fig. 5.4. Hence, the entire proposed pipeline is integrated in an end-to-end manner for high-rise building inspection.

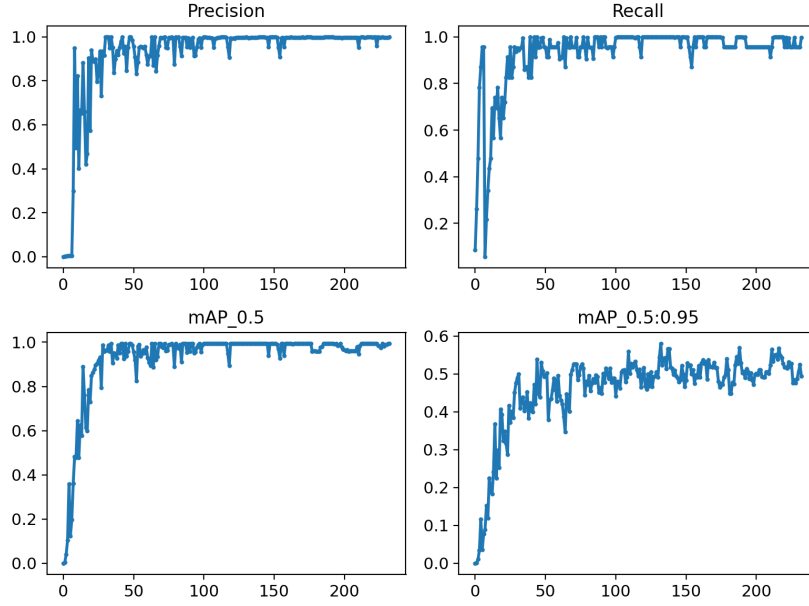


Figure 5.2 Evaluation metrics of the crack detection model.

5.1.3 Analysis

The first step in our pipeline is collision-free drone navigation. The depth camera used here has a maximum depth range of 30 metres. This gives the UAV enough visibility to avoid upcoming obstacles. The UAV takes 200-300 milliseconds to compute the distance and the size of the obstacle to avoid them in real time for safety.

The results of façade detection as shown in Fig. 3.4 show a bounding box around the cracks detected by the YOLOv5 model. Prediction results with evaluation metrics such as precision, recall and mean average precision (mAP) are shown in Fig. 5.2 and Table 5.2. Since our detection model is learning from the annotated cracks on the building, it predicts cracks with a recall of 95.7% and an average precision of 96% for unseen images. This gives confidence to the pipeline that our model performs well on a real-world high-rise building.

Finally, our pipeline integrated with all the modules helps to autonomously monitor building inspection, as shown in Fig. 5.5.

Metrics	Train	Validation	Test
Number of images	139	18	18
Number of cracks	172	23	23
Precision	1.00	0.995	0.95
Recall	0.998	1.00	0.957
mAP50	0.995	0.995	0.96

Table 5.2 Results of the crack detection model.

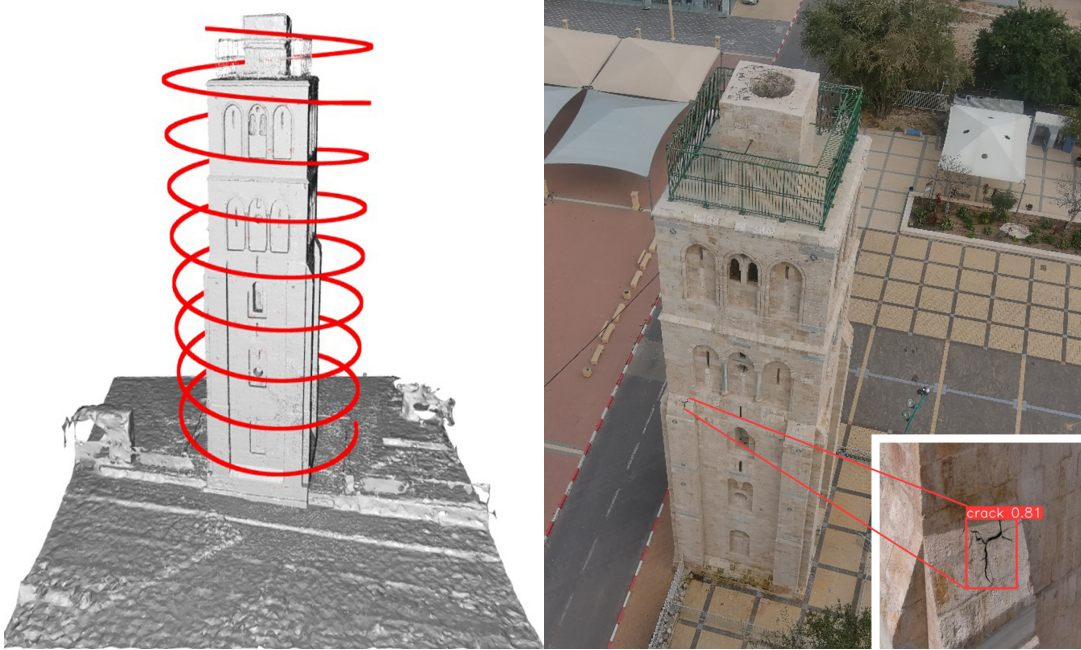


Figure 5.3 Our reconstructed 3D mesh model (left) of the high-rise building (right) with the detected crack (bottom right). The proposed pipeline reconstructs the 3D model from the real-world images captured by UAV, where the red spiral path marks the trajectory followed by the UAV.



Figure 5.4 3D mesh model of the building (left) with marked crack using 2D to 3D mapping shown using the red ray and red ring; Captured image of the building (top-right) using UAV showing the crack predicted using YOLOv5 [1]; Zoomed-in Crack (bottom-right); Correspondences of all the three images are shown using blue lines.

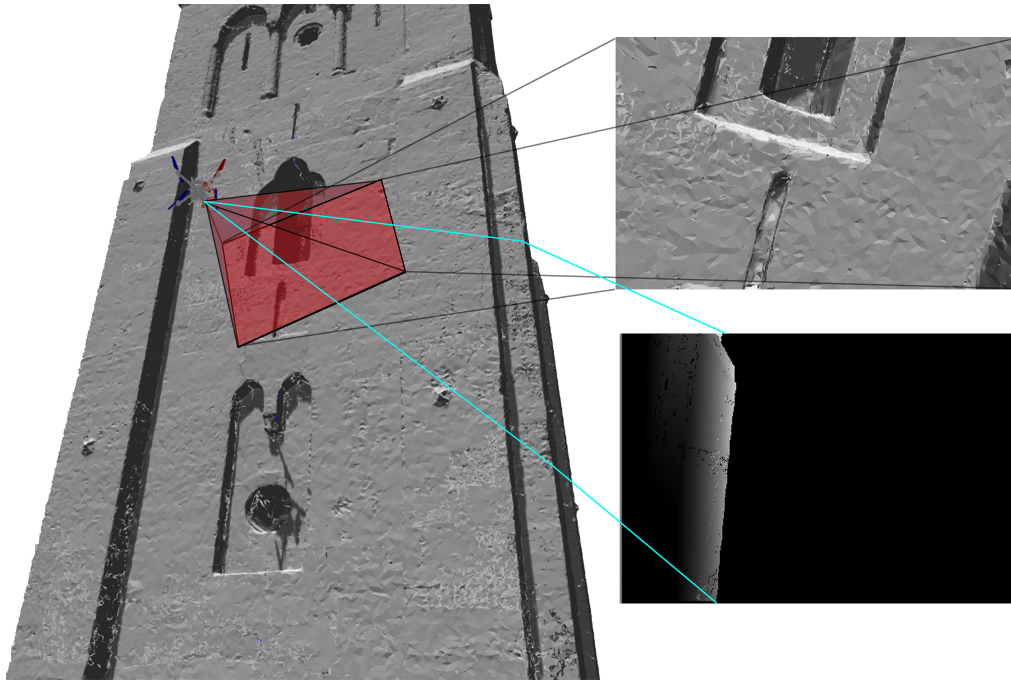


Figure 5.5 The final simulation of the autonomous drone navigation on our high-rise building 3D reconstruction (left); the RGB camera view of the drone (top right); the depth camera view of the drone (bottom right).

Chapter 6

Superresolution

6.1 Introduction

The central theme of Super-resolution (SR) techniques are primarily focused on the task of generating higher resolution images from lower resolution counterparts. The main objective behind this is to extract more details about the original scene by increasing the pixel density of the image. High-resolution images are particularly valuable in computer vision applications as they contribute to improved performance in tasks like pattern recognition and image analysis. In the field of medical imaging, high resolution is crucial for accurate diagnosis. Various applications, such as surveillance, forensic analysis, and satellite imaging, often necessitate the ability to zoom in on specific areas of interest within an image, making high resolution an essential requirement.

High-resolution images are not always readily available due to factors such as cost and limitations in sensor and optics manufacturing technology. Creating a setup for high-resolution imaging can be expensive and impractical in certain situations. However, image processing algorithms, such as super-resolution, offer a solution to this problem. By utilizing these algorithms, it becomes possible to enhance the resolution of existing low-resolution images without the need for expensive hardware upgrades. This advantage of super-resolution lies in its cost-effectiveness and the ability to make use of existing low-resolution imaging systems. It provides an accessible and affordable alternative for obtaining higher resolution images.

Thermal Cameras:

1. *Detection of Anomalies:* Thermal cameras capture the infrared radiation emitted by objects and display temperature variations. This enables the identification of anomalies such as heat leaks, insulation deficiencies, electrical faults, or water intrusions that may not be visible to the naked eye. Detecting these issues early on helps prevent energy waste, potential hazards, or further damage.
2. *Fire Detection:* Thermal cameras are effective in detecting heat signatures associated with fires. They can quickly identify hotspots or areas of increased temperature, allowing for rapid response

and timely fire prevention measures. UAVs equipped with thermal cameras can provide early fire detection and aid firefighters in assessing the situation from a safe distance.

3. *Energy Efficiency Assessments*: Thermal cameras provide insights into the energy efficiency of buildings. By visualizing heat patterns, they help identify areas of heat loss or excessive energy consumption, leading to targeted improvements and cost savings. Thermal imaging assists in assessing the effectiveness of insulation, HVAC systems, or renewable energy installations.

6.2 Theory

The core concept of super-resolution is to leverage a series of low-resolution images, which may contain noise, of a particular scene to generate a high-resolution image or image sequence. The objective is to reconstruct the original scene image with enhanced resolution based on the observed images at lower resolution. By utilizing information from multiple low-resolution images, super-resolution algorithms aim to recover finer details and improve the overall visual quality of the final high-resolution image. This process involves sophisticated techniques to estimate and align the low-resolution images, enhance their spatial resolution, and mitigate the effects of noise to generate a more detailed and visually pleasing high-resolution representation of the scene.

The general approach in super-resolution considers that the low-resolution images are obtained by resampling a high-resolution image. The objective is to reconstruct the original high-resolution image, which, when resampled using the input images and the imaging model, will generate the observed low-resolution images. Therefore, the accuracy of the imaging model plays a crucial role in super-resolution techniques. If the modeling of factors like motion (for example, camera or object motion) is incorrect, it can actually worsen the image quality rather than improve it. A precise and accurate modeling of various factors involved in the imaging process is essential to achieve successful super-resolution results. By ensuring an accurate understanding and representation of these factors, super-resolution algorithms can effectively recover the high-resolution details and enhance the overall image quality.

Single image super-resolution (SISR) is a significant aspect of image restoration, focusing on the enhancement of low-resolution (LR) images to generate high-resolution (HR) versions. This technique finds applications in various domains, such as camera surveillance systems where it can improve the recognition of individuals with low-resolution faces. Additionally, super-resolution (SR) methods are utilized in areas like HDTV, medical imaging, and satellite imaging to enhance image quality and detail.

6.3 Related Work

The super-resolution (SR) is a technique used in Computer Vision to obtain High Resolution (HR) image from its corresponding Low Resolution image (LR). It is an image enhancing task used to get

refined details and better quality from the coarse details. This technique is widely used in medical imaging, security and surveillance systems, satellite imaging and other imaging systems.

In the modern era, there is an abundance of digital images with the increase in demand of technology. The quality of images and videos cannot be compromised with this demand. Further, this gives a rise to super-resolution technique which is also known by other names such as zooming, enlargement, up-sampling, image scaling and interpolation. There are many Image Processing techniques used to perform this task but they have a limitation due to fixed mathematical formulation. On the other hand, with the advent of Machine Learning and Deep Learning (DL), machines have got the ability to learn from varied data to generate better quality results.

There are various traditional approaches used for enhancing the quality of images. Some of them are prediction-based [42, 43, 44], statistical approach based [45, 46], edge-based [47, 48], patch-based [47, 49, 50] and representation based approaches [51, 52]. In Digital Image Processing, image interpolation is a technique used to resize or distort the image from one pixel grid to another. Interpolation is used to estimate values at unknown points by using the previously known points. The most famous and effective interpolation techniques are nearest neighbour [53], bilinear [54] and bicubic [55] interpolation.

It has been challenging to perform real-time application on the resource limited embedded edge device in computationally efficient manner. Jetson Nano is the latest edition to NVIDIA's Jetson line of computing boards. It is used for artificial intelligence based application to bring the power of GPUs to a small single board computer with a low-power envelope. Thus, we used Jetson Nano board for experimentation and inference purpose in our paper.

In this work, we have had experimented various super-resolution models and selected a few which can be used for video super-resolution task. A comparative study of the selected models has been done. Those models were trained on RGB dataset as well as custom thermal datasets prepared by us. We have implemented an end-to-end SR pipeline to perform real-time up-scaling depending on the scaling factors provided. We optimized the pipeline to achieve high performance measured on the basis various metrics such as FPS, latency, PSNR, SSIM, GPU/CPU utilization. High performance was achieved using less thermal data for training. The whole system was deployed on constrained embedded edge device NVIDIA Jetson Nano board to perform the task in real-time which was successfully done.

6.4 Dataset

In this work, we used three types of datasets which are as follows: (1) DIVERse 2K resolution high quality images (DIV2K) dataset [56] consisting of 800 RGB training images and 200 test images, (2) a random thermal dataset prepared using our thermal camera consisting of 231 training images 70 test images, (3) a contrast-variant and wide view/angle thermal dataset also prepared using our thermal camera consisting of 1187 training images 250 test images.

Model	Model size (No. of parameters)	Inference time
ESPCN	Small model (20K)	Good inference, real-time video up-scaling (depending on image size)
FSRCNN	Small model (12K)	Accurate inference, real-time video up-scaling
LapSRN	Medium sized model (812K)	Good inference
EDSR	Big model (43M)	Slow inference

Table 6.1 Comparison study of studied models.

6.5 Deep Learning Models

Various models have been developed for super-resolution task. Here, we focus on exploiting them for thermal imaging purpose. For the real-time SR task, there are many factors which need to be considered. Some of them are: 1) Model size: The size of a deep learning model is determined by the number of parameters present in it. The smaller the size of the model, the lesser space it occupies on the processor board. The model chosen should have small size; 2) Inference time: As the task has to be performed in the real-time environment, inference time plays a major role. The model chosen should have more frames per second; 3) Accuracy: There should be decent accuracy with a correct balance of model size and inference time.

The super-resolution models Table 6.1 used in this paper for comparison study are:

6.5.1 ESPCN

Efficient Sub-Pixel Convolutional Neural Network [57] (ESPCN) is a CNN-based architecture proposed for super-resolution tasks. It introduces an efficient sub-pixel convolutional layer to the network, enabling resolution enhancement at the final stage. Unlike traditional methods that use interpolation for upscaling, ESPCN directly takes the smaller low-resolution (LR) image as input, eliminating the need for interpolation. By handling upscaling in the last layer, ESPCN allows the network to learn a superior LR-to-HR mapping compared to interpolation-based upscaling performed prior to feeding the image

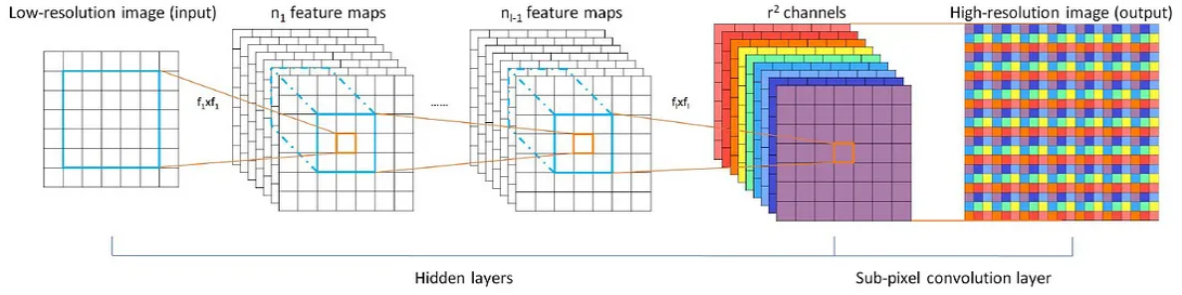


Figure 6.1 ESPCN network architecture

into the network. This approach enhances the capability of the network to generate high-resolution (HR) images with better details.

ESPCN benefits from the reduced input image size, which allows the use of smaller filter sizes to extract features. This leads to a reduction in computational complexity and memory requirements, significantly improving efficiency. These advantages make ESPCN an ideal choice for real-time super-resolution of high-definition (HD) videos, where the architecture's efficiency enables rapid processing and delivery of enhanced video quality. Overall, ESPCN's incorporation of the efficient sub-pixel convolutional layer, its ability to learn improved LR-to-HR mappings, reduced computational complexity, and suitability for real-time video processing make it a favorable option for super-resolution tasks, particularly in the context of HD video enhancement.

6.5.2 FSRCNN

Fast Super-Resolution Convolutional Neural Network [58] (FSRCNN) is an imaging technique specifically designed to enhance the resolution of digital images. It employs a shallow network architecture that offers faster and clearer results compared to previous methods. FSRCNN has gained attention for its potential in real-world applications within the field of digital imaging. It demonstrates the ability to generate high-quality images that have been upsampled from their original size, a process often referred to as "super-scaling." By leveraging the power of convolutional neural networks, FSRCNN achieves improved image quality while maintaining shorter processing times.

While Bicubic Interpolation is a common baseline method for upscaling images, SRCNN [59, 60] surpasses it by learning more complex mappings and utilizing the power of deep convolutional neural networks. SRCNN has demonstrated improved performance and image quality compared to traditional interpolation techniques like Bicubic Interpolation. FSRCNN is a deep learning model based on shallow network design which is clearer and faster as compared to its predecessors in super-resolving the images. FSRCNN involves the following steps (1) feature extraction which replaces bicubic interpolation with 5x5 convolutions; (2) shrinking which reduces the feature maps; (3) non-linear mapping where

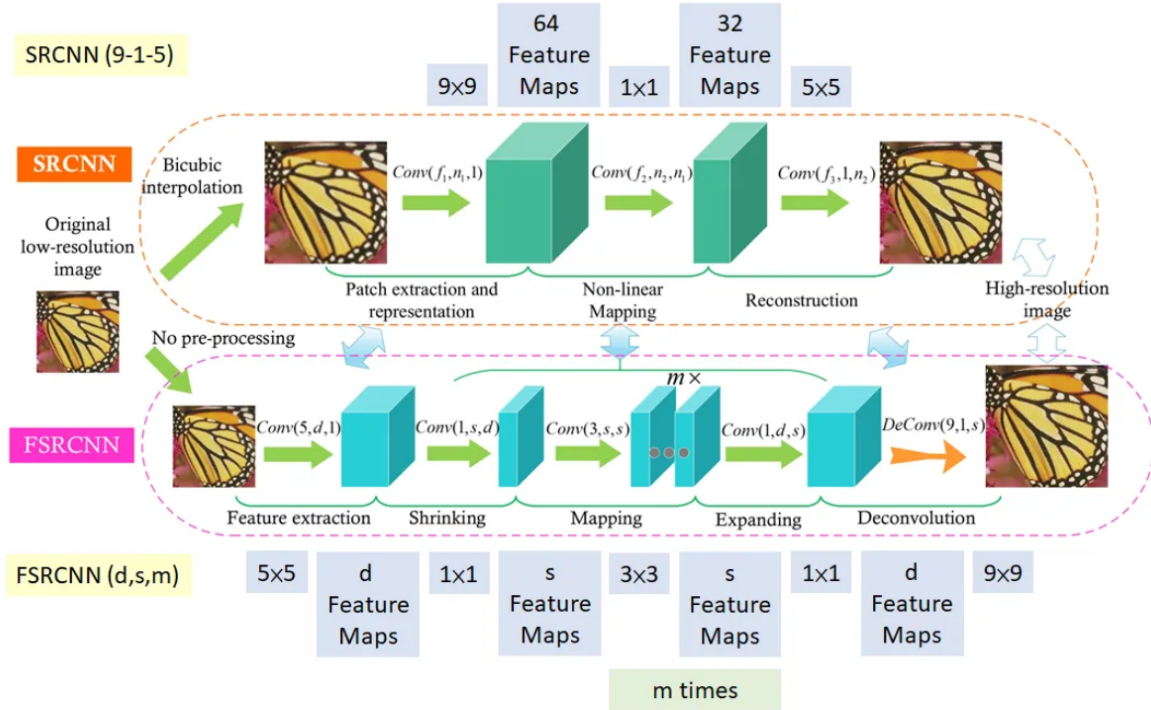


Figure 6.2 FSRCNN network architecture

multiple layers are applied 3x3; (4) expanding where the feature map is increased by 1x1 convolution and deconvolution where the HR image is reconstructed using 9x9 filter.

The key strengths of FSRCNN lie in its ability to produce enhanced image quality and its efficiency in terms of runtime. These attributes make it an appealing option for tasks that require high-quality image upscaling. By utilizing the advantages of deep learning techniques, FSRCNN contributes to advancements in the field of image resolution enhancement.

6.5.3 LapSRN

Laplacian Pyramid Super-Resolution Network [61] (LapSRN) is a deep learning model specifically designed for single-image super-resolution. The network consists of two main components: the analysis network and the synthesis network. The analysis network takes the low-resolution input image and generates a set of residual images at different levels of the Laplacian pyramid. These residuals capture the high-frequency details missing in the low-resolution image. The synthesis network then takes these residuals and progressively combines them with upsampled versions of the low-resolution image to reconstruct the high-resolution image at each level. LapSRN progressively reconstructs the sub-band residuals of HR images at multiple pyramid levels, specifically $\log_2(S)$ levels where S is the scaling

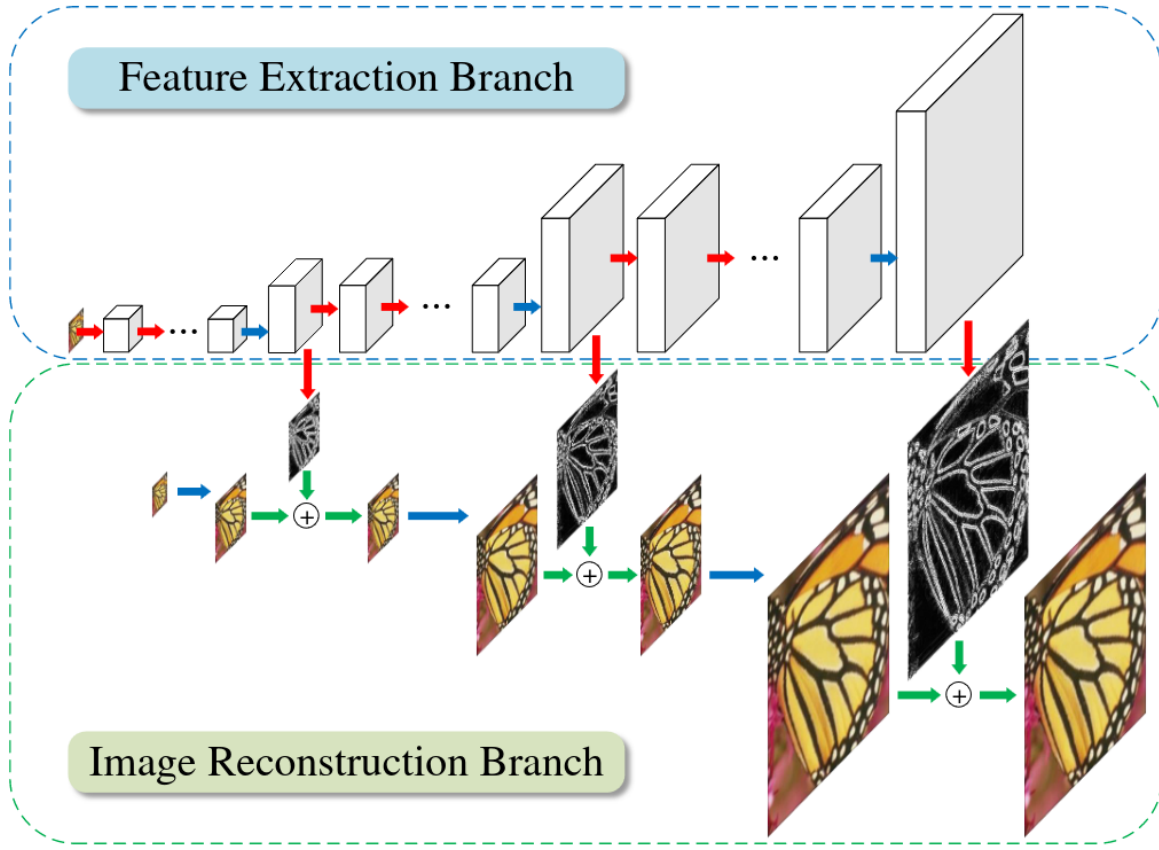


Figure 6.3 LapSRN network architecture

factor (i.e. 2, 4, 8). There is low computational load as it directly extracts features from the LR input image. Laplacian pyramid has been used which has one residual image outputted at each level of feature extraction branch. There are two branches which are Feature Extraction and Image Reconstruction.

One of the advantages of LapSRN is its ability to handle images of arbitrary sizes since it operates on the Laplacian pyramid representation. It can effectively enhance the resolution of both small and large images. Additionally, LapSRN can achieve state-of-the-art performance in terms of reconstruction accuracy and visual quality.

6.5.4 EDSR

Enhanced Deep Super-Resolution [62] (EDSR) is a deep learning model specifically designed for single-image super-resolution tasks. The architecture of EDSR is based on the Residual Network (ResNet) design, which consists of multiple residual blocks. Residual blocks allow the network to learn residual mappings that capture the high-frequency details necessary for super-resolution. This helps

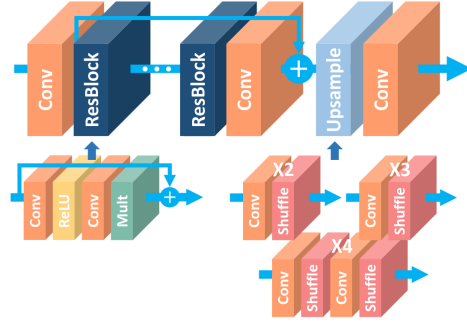


Figure 6.4 EDSR network architecture

alleviate the challenge of effectively learning the mapping between low-resolution and high-resolution images.

EDSR is used for single image SR. EDSR is a conv-net based SR which up-samples the LR in the network. The batch normalization is removed from the residual blocks which improves the performance without impacting the training speed of the network. To avoid the numerical instability, the full single-scale model uses residual scaling of 0.1. Pre-training of 3x and 4x networks with 2x for faster convergence using L1 loss. 8 possible flip/rotation combinations are applied on the output before averaging.

6.6 Evaluation Metrics

In the context of evaluating the quality of the resultant output in various machine learning and deep learning tasks, it's common to employ specific evaluation metrics to quantitatively measure and assess how well a model is performing. There are two evaluation metrics used to test the quality of the resultant output, which are:

6.6.1 Peak Signal-to-Noise Ratio

Peak Signal-to-Noise Ratio (PSNR) is a metric used to evaluate the quality of digital images. It provides a numerical measurement of how much noise or distortion is present in an image compared to the original, noise-free version. The PSNR is calculated by comparing the pixel values of the original image (referred to as the "signal") with the pixel values of the distorted image (which includes noise or other forms of distortion). The difference between corresponding pixels in the two images is squared, averaged, and then transformed into a logarithmic scale.

In the context of superresolution tasks, Peak Signal-to-Noise Ratio (PSNR) is used as a quantitative measure to assess the quality of a superresolved image compared to its original low-resolution counter-

part. It helps to evaluate how well the superresolution algorithm has reconstructed the high-resolution details and reduced the noise in the final output. Here's how PSNR is applied to superresolution tasks:

1. *Low-resolution image*: Start with a low-resolution image, which serves as the input to the super-resolution algorithm. This image typically has fewer pixels and lower spatial detail compared to the desired high-resolution image.
2. *High-resolution image*: Have access to the ground truth or reference high-resolution image, which represents the ideal target or the image you want to achieve through superresolution. This image is used for comparison purposes.
3. *Superresolution algorithm*: Apply a superresolution algorithm to enhance the low-resolution image and generate a superresolved output. The algorithm employs various techniques like interpolation, upsampling, or deep learning-based approaches to estimate the missing high-frequency details.
4. *Compute the PSNR*: Calculate the PSNR between the superresolved image and the high-resolution reference image. This is done by comparing the pixel values of corresponding pixels in both images and applying the same formula used for general image PSNR. In this case, the MSE (mean squared error) represents the average squared differences between pixel values of the superresolved image and the high-resolution reference image.
5. *Interpretation*: A higher PSNR value for the superresolution task indicates that the superresolved image has less noise and is more similar to the high-resolution reference image. It suggests that the superresolution algorithm has successfully reconstructed the missing details and improved the overall image quality.

In lossy transformations such as image compression and image inpainting, Peak signal-to-noise ratio (PSNR) is one of the most popular quality measurement. Given two images I and \hat{I} both with N pixels, the Mean Squared Error (MSE) and PSNR are defined as:

$$MSE = \frac{1}{N} \|I - \hat{I}\|_F^2 \quad (6.1)$$

$$PSNR = 10 \log_{10} \left(\frac{L^2}{MSE} \right) \quad (6.2)$$

However, it's important to note that PSNR alone might not provide a complete assessment of the visual quality of the superresolved image. Superresolution techniques can sometimes introduce artifacts or over-smoothing, which might not be adequately captured by PSNR. Hence, it's often recommended to complement PSNR with other metrics like Structural Similarity Index (SSIM) or perceptual evaluation to ensure a comprehensive evaluation of the superresolution results. Besides its application in superresolution tasks, Peak Signal-to-Noise Ratio (PSNR) is widely used in various other domains for quality assessment and comparison purposes. Other common applications where PSNR is utilized are

image and video compression, image denoising, image restoration, image watermarking, image quality enhancement.

6.6.2 Structural Similarity

The Structural Similarity Index (SSIM) is a metric used to measure the perceived similarity between two images. Unlike Peak Signal-to-Noise Ratio (PSNR), which focuses on pixel-wise differences, SSIM takes into account both structural information and pixel values to assess the visual quality of an image. The Structural Similarity Index (SSIM) is proposed for measuring the structural similarity between images which is based on independent comparisons in terms of contrast, luminance and structures.

SSIM considers three key components of image perception:

1. *Luminance comparison*: SSIM evaluates the similarity in terms of the overall brightness or luminance of the images. It measures the mean luminance values of corresponding image patches and computes the similarity based on their similarity in luminance.
2. *Contrast comparison*: SSIM takes into account the contrast or local variations in pixel values. It evaluates how well the local structures and details are preserved in the images. By comparing the standard deviations of the corresponding image patches, SSIM measures the similarity in terms of contrast.
3. *Structure comparison*: SSIM also considers the correlation between image patches. It captures the structural information by assessing how well the spatial dependencies and relationships between pixels are maintained in the images. It measures the similarity of the covariance between corresponding image patches.

Based on these three components, SSIM calculates a similarity index value between 0 and 1, where 1 indicates a perfect match or similarity between the images. The measurement is highly adapted to extract image structures inspired by the human visual system. Given two images I and \hat{I} both with N pixels, SSIM is defined as:

$$SSIM(I, \hat{I}) = \frac{2\mu_I\mu_{\hat{I}} + k_1}{\mu_I^2 + \mu_{\hat{I}}^2 + k_1} \cdot \frac{\sigma_{I\hat{I}} + k_2}{\sigma_I^2 + \sigma_{\hat{I}}^2 + k_2} \quad (6.3)$$

where μ_I and σ_I^2 are mean and variance of I , $\sigma_{I\hat{I}}$ is the covariance between I and \hat{I} , and k_1 and k_2 are constant relaxation terms.

The application of SSIM to superresolution involves the following steps:

1. *Low-resolution image*: Begin with a low-resolution image, which serves as the input to the super-resolution algorithm.
2. *High-resolution image*: Have access to the ground truth or reference high-resolution image, representing the desired target or the image you want to achieve through superresolution.

3. *Superresolution algorithm*: Apply a superresolution algorithm to enhance the low-resolution image and generate a superresolved output.
4. *Compute the SSIM*: Calculate the SSIM between the superresolved image and the high-resolution reference image. SSIM compares the structural information, luminance, and contrast of corresponding image patches in both images. By considering these factors, SSIM quantifies the perceived similarity.

The SSIM value ranges between 0 and 1, where 1 indicates a perfect match or high similarity between the superresolved image and the high-resolution reference image. Applying SSIM to superresolution tasks helps to assess the ability of the superresolution algorithm to preserve the structural details, textures, and overall visual appearance of the high-resolution image. Higher SSIM values indicate that the superresolved image maintains a higher degree of structural similarity with the reference image, indicating a better quality superresolution result. However, it's important to note that SSIM is just one of several metrics that can be used to evaluate the quality of superresolved images. It is often combined with other metrics such as Peak Signal-to-Noise Ratio (PSNR) and perceptual evaluation to obtain a comprehensive assessment of the superresolution output.

SSIM is commonly used in image and video processing applications, such as image compression, denoising, superresolution, and image quality assessment. It provides a more comprehensive evaluation of image quality, taking into account perceptual factors beyond pixel-wise differences. SSIM is particularly useful in cases where human visual perception is essential, as it aligns better with human judgment compared to metrics like PSNR.

It's important to note that while PSNR is a widely used metric, it does have limitations. It focuses solely on pixel-wise differences and may not align with human perception. In certain applications, perceptual metrics like Structural Similarity Index (SSIM) or subjective evaluation by human observers are used in conjunction with PSNR for a more comprehensive quality assessment.

6.7 Experiments and Results

Superresolution is essential in building inspection tasks due to its ability to enhance image quality and detail. By increasing the resolution of images, it enables inspectors to identify and analyze fine-scale features, defects, or anomalies that may not be clearly visible in standard-resolution images. This technology improves the accuracy and effectiveness of building inspections, aiding in the detection of critical issues and supporting informed decision-making for maintenance and repairs.

6.7.1 Implementation Details

The implementation details of deep learning model training and end-to-end inference pipeline to perform superresolution are as follows:

Model	Version
Tensorflow	1.15.2
Tensorflow GPU	1.15.2
Python	3.6.9
Ubuntu	18.04.5
GCC	8.4.0

Table 6.2 Final model setup dependencies.

6.7.1.1 Model Training

We executed the training task using three types of datasets. Firstly, the models were trained using RGB images (DIV2K dataset). Secondly, the models were fine-tuned using a random thermal dataset. This was done to let models understand the thermal images and to lower the haziness/blurriness. Thirdly, further fine tuning was done using the contrast-variant and wide view/angle thermal dataset for making the contrast better in the resultant images. The target images used in the dataset are the thermal images captured by our thermal camera and the input images are their respective bicubic interpolation down-sampled images (as per required scaling factor) as per standardized convention of SR dataset.

6.7.1.2 Inference Pipeline

Here, the inference pipeline for super-resolution task has the objective of decent up-sampling super-resolution, low latency and high fps on the NVIDIA Jetson Nano board. Initially, each frame of the real-time video was cropped with respect to the center to obtain the frame of $(640/x, 480/x)$ units dimension where x is the scaling factor. This fed low dimensional input image to the model which further reduced the processing time of image and maintained the image dimensions of the output as $(640, 480)$ units. After that, each cropped frame of 16-bit integer value and grayscale format is converted into 32-bit float value in the YCrCb color space to conserve the pixel information as much as possible. For further optimization to increase fps and decrease latency, parallel processing of I/O tasks were done. In this, an entirely different thread was prepared to read frames in parallel and stack it. This was done so that the model does not have to wait for the complete pipeline script to finish. Figure (6.5) shows the schematic diagram of the complete inference pipeline.

All of these implementations speeded up the process to a great extent. Earlier, only the deep learning model module in the pipeline script was utilizing the GPU and was giving a peak of 25% in GPU utilization which further increased to 90% peak value when most of the modules in the script used CUDA backend. This activation of CUDA backend in most of the inference pipeline modules led to 10 times

Model	2x	3x	4x	8x
ESPCN -T1	30.088dB / 0.8414	30.086dB / 0.7616	29.659dB / 0.6735	-
-T2	32.838dB / 0.8856	31.475dB / 0.7985	30.637dB / 0.7278	
-T3	32.931dB / 0.8895	31.865dB / 0.8015	30.948dB / 0.7289	
FSRCNN -T1	32.706dB / 0.8841	31.340dB / 0.7995	30.527dB / 0.7242	-
-T2	32.716dB / 0.8846	31.447dB / 0.8002	30.603dB / 0.7246	
-T3	32.880dB / 0.8884	31.399dB / 0.8008	30.573dB / 0.7259	
LapSRN -T1	32.799dB / 0.8857	-	30.689dB / 0.7305	29.468dB / 0.5185
-T2	32.8656dB / 0.8859		30.690dB / 0.7346	29.382dB / 0.5042
-T3	32.8656dB / 0.8859		30.914dB / 0.7532	29.704dB / 0.5711
EDSR -T1	32.58dB/0.9165	31.23dB/0.8942	30.05dB/0.8930	-
-T2	33.56dB/0.9248	31.85dB/0.9054	30.49dB/0.8927	
-T3	34.43dB/0.9362	32.06dB/0.9132	31.08dB/0.9064	

Table 6.3 Accuracy (PSNR/SSIM) of various DL models on multiple scaling factors (2x, 3x, 4x, 8x) after two times fine-tuning of models. T1, T2 and T3 denote first, second and third training respectively.

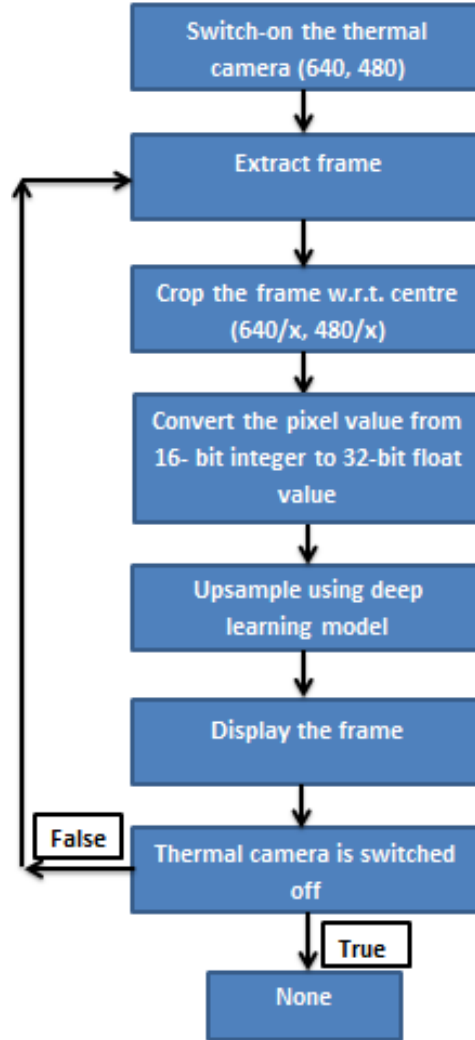


Figure 6.5 Inference Pipeline.

decrease in the latency and 10 times increase in FPS. Table (6.2) shows the final model dependencies with their respective implementation versions.

6.7.2 Results

Table (6.4) shows the latency and frames per second (FPS) of the experimented models on NVIDIA Jetson Nano board. There has been 10 times improvement when the inference whole pipeline including the model use GPU. Table (6.3) displays the PSNR and SSIM values of all the 4 models on multiple scaling factors (2x, 3x, 4x, 8x) during two times fine-tuning process and Table (6.5) displays their respective final results. Among the selected DL models, ESPCN performed the best after studying various

LATENCY(ms)/ FPS(GPU enabled)	x2	x3/x8	x4
ESPCN	34.6	27.3	22.1
	28.9	36.63	45.24
FSRCNN	209.9	98.2	60.5
	4.78	10.18	16.52
LapSRN	419.5	562	519.4
	2.38	1.77	1.92
Is only CPU enabled: 10 times FPS			

Table 6.4 Latency and Frames per second (FPS) of various models.

evaluation metrics. On the other hand, EDSR being a very heavy model is unable to perform in real time.

In Table (6.3), ESPCN shows the best result in comparison to the others after final training of the models. It's CPU and GPU utilizations are also less as shown in Table (6.5). FSRCNN has also shown some good results, it's PSNR and SSIM values are good but the latency for processing each frame is highest in FSRCNN as compared to all the other models. RAM consumption of all the models is approximately the same but the GPU utilization of models other than ESPCN touches its maximum threshold, which will make the system throttle.

In conclusion, the ESPCN model in the experiment has demonstrated its success and achieves the real-time superresolution result with good accuracy on NVIDIA Jetson Nano. Also, it consumes less power to perform the inference on the device. Figure 6.6 shows sample target image (HR) and predicted output of various deep learning models using various scaling factors 2x, 3x, 4x, 8x on test set.

PARAMETERS		ESPCN				FSRCNN			LapSRN		
		x2	x3	x4	x2	x3	x4	x2	x4	x8	
CPU utilization (%)	CPU1	88	64	64	47	50	76	54	35	61	
	CPU2	64	68	77	58	32	60	47	42	22	
	CPU3	54	64	57	54	36	58	50	76	60	
	CPU4	73	76	59	42	44	52	86	50	32	
GPU utilization (%)		54-99	44-87	23-73	80-99	71-99	35-99	80-99	70-99	69-99	
Power (mW)		8055	5868	5088	7523	7603	8253	9003	9225	8622	
Memory (1.7/4.1GB)	RAM (GB)	2.5/4.1	2.5/4.1	2.5/4.1	2.5/4.1	2.5/4.1	2.5/4.1	2.6/4.1	2.6/4.1	2.6/4.1	

Table 6.5 Comparison of models on various parameters.

6.7.3 Contribution

This paper proposes a highly optimized inference pipeline for performing real-time super-resolution task on thermal imaging. This pipeline has very low latency and high frames per second. Fewer resources have been used to perform the task along with less thermal dataset. The two times fine-tuning process of the models have significantly improved the results. This high performance is achieved in real-time on NVIDIA Jetson Nano board. The super-resolution task on thermal images is highly required in many surveillance systems to get the clarity of object at long distance. We have also applied the model on our thermal dataset.

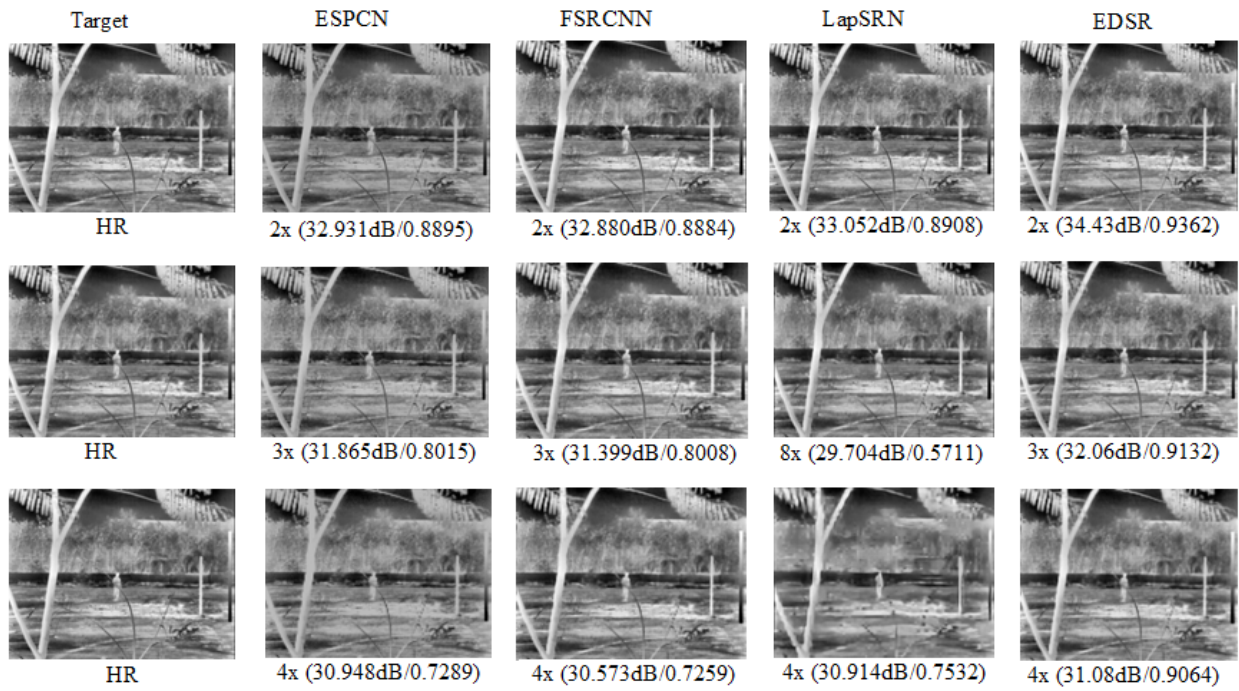


Figure 6.6 Sample target image (HR) and predicted output of various deep learning models using various scaling factors (2x, 3x, 4x, 8x) on test set.

Chapter 7

Conclusion and Future Work

In conclusion, our thesis presents a novel autonomous inspection pipeline for building crack detection, achieving high precision, recall, and mAP scores. The pipeline consists of autonomous drone navigation, facade detection, and model construction modules, demonstrating their effectiveness in the results. This automated pipeline represents a significant contribution to the structural engineering domain and serves as a foundation for future work in other domains, including the automobile industry, agriculture, and underwater environments.

Additionally, we proposed a highly optimized end-to-end real-time video up-scaling super-resolution pipeline using a thermal camera. Among the experimented DL super-resolution models deployed on the NVIDIA Jetson Nano board, the ESPCN model demonstrated superior performance, achieving high FPS and low latency, as well as impressive PSNR and SSIM scores for various scaling factors. Through empirical analysis, we discovered that small modifications to deep learning models can yield significant improvements in performance optimization. Future research can focus on adapting these models to work with variable scaling factors and evaluate their performance on different hardware platforms. This can be extended to the building inspection.

Overall, our research contributes to automated inspection techniques of building using depth vision. This can be extended using the advancements in video super-resolution of thermal camera, opening avenues for further exploration and applications in various industries.

Related Publications

- A Real-time Super-Resolution for Surveillance Thermal Cameras using optimized pipeline on Embedded Edge Device

Prayushi Mathur, Syed Azeemuddin, *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 16-19 November 2021, Washington, DC, USA

- Autonomous Inspection of High-rise Buildings for Facçade Detection and 3D Modeling using UAVs

Prayushi Mathur, Charu Sharma, Syed Azeemuddin, *IEEE Access*

- System for Monitoring Building Damage by Mapping 2D Image of Damage onto 3D Building Model.

Patent Application Number: **202341072812**

Prayushi Mathur, Charu Sharma, Syed Azeemuddin, *Indian Patent Office*

Bibliography

- [1] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, Zeng Yifu, Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultra-lytics/yolov5: v7.0 - yolov5 sota realtime instance segmentation, 2022.
- [2] Tarek Rakha and Alice Gorodetsky. Review of unmanned aerial system (uas) applications in the built environment: Towards automated building inspection procedures using drones. *Automation in Construction*, 93, 2018.
- [3] Shidrokh Goudarzi, Nazri Kama, Mohammad Hossein Anisi, Sherali Zeadally, and Shahid Mumtaz. Data collection using unmanned aerial vehicles for internet of things platforms. *Computers & Electrical Engineering*, 75:1–15, 2019.
- [4] Carlo Dal Mutto, Pietro Zanuttigh, and Guido M Cortelazzo. *Time-of-flight cameras and Microsoft KinectTM*. Springer Science & Business Media, 2012.
- [5] Hao Bai, Xiangyu Hu, Fei Chen, Zhiyong Liao, Kai Li, Guangjiong Ran, Fengni Wei, et al. A depth camera-based intelligent method for identifying and quantifying pavement diseases. *Advances in Civil Engineering*, 2022, 2022.
- [6] Steffen Herbort and Christian Wöhler. An introduction to image-based 3d surface reconstruction and a survey of photometric stereo methods. *3D Research*, 2(3):1–17, 2011.
- [7] Jiang Bian, Xiaolong Hui, Xiaoguang Zhao, and Min Tan. A novel monocular-based navigation approach for uav autonomous transmission-line inspection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [8] Nicolai Iversen, Oscar Bowen Schofield, Linda Cousin, Naeem Ayoub, Gerd Vom Bögel, and Emad Ebeid. Design, integration and implementation of an intelligent and self-recharging drone system for autonomous power line inspection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

- [9] Giacomo Picardi, Rossana Lovecchio, and Marcello Calisti. Towards autonomous area inspection with a bio-inspired underwater legged robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [10] Kelly Steich, Mina Kamel, Paul Beardsley, Martin K Obrist, Roland Siegwart, and Thibault Lachat. Tree cavity inspection using aerial robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [11] Matteo Fumagalli, Roberto Naldi, Alessandro Macchelli, Raffaella Carloni, Stefano Stramigioli, and Lorenzo Marconi. Modeling and control of a flying robot for contact inspection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.
- [12] AE Jimenez-Cano, J Braga, Guillermo Heredia, and Aníbal Ollero. Aerial manipulator for structure inspection by contact from the underside. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2015.
- [13] Narcis Palomeras, Marc Carreras, Pere Ridao, and Emili Hernandez. Mission control system for dam inspection with an auv. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.
- [14] Soohwan Song and Sungho Jo. Online inspection path planning for autonomous 3d modeling using a micro-aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [15] GT Ferraz, J De Brito, VP De Freitas, and JD Silvestre. State-of-the-art review of building inspection systems. *Journal of performance of constructed facilities*, 30, 2016.
- [16] Alastair M Paterson, Geoff R Dowling, and Denis A Chamberlain. Building inspection: can computer vision help? *Automation in Construction*, 7, 1997.
- [17] Hafiz Suliman Munawar, Ahmed WA Hammad, Assed Haddad, Carlos Alberto Pereira Soares, and S Travis Waller. Image-based crack detection methods: A review. *Infrastructures*, 6, 2021.
- [18] Christian Eschmann. Unmanned aircraft systems for remote building inspection and monitoring. 2012.
- [19] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338, 2019.
- [20] Georgios Darivianakis, Kostas Alexis, Michael Burri, and Roland Siegwart. Hybrid predictive control for aerial robotic physical interaction towards inspection operations. In *IEEE international conference on robotics and automation (ICRA)*, 2014.

- [21] Facundo José López, Pedro M Leronés, José Llamas, Jaime Gómez-García-Bermejo, and Eduardo Zalama. A review of heritage building information modeling (h-bim). *Multimodal Technologies and Interaction*, 2, 2018.
- [22] Fabrizio Flacco, Torsten Kröger, Alessandro De Luca, and Oussama Khatib. A depth space approach to human-robot collision avoidance. In *IEEE international conference on robotics and automation*, 2012.
- [23] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE conference on computer vision and pattern recognition*, 2016.
- [24] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*. Springer, 2016.
- [25] Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE transactions on visualization and computer graphics*, 18, 2012.
- [26] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. In *ACM siggraph*. 2007.
- [27] Gaël Guennebaud, Marcel Germann, and Markus Gross. Dynamic sampling and rendering of algebraic point set surfaces. In *Computer Graphics Forum*, volume 27. Wiley Online Library, 2008.
- [28] Jingjing Guo, Qian Wang, Yiting Li, and Pengkun Liu. Façade defects classification from imbalanced dataset using meta learning-based convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, 35(12):1403–1418, 2020.
- [29] Michael YL Chew. Façade inspection for falling objects from tall buildings in singapore. *International Journal of Building Pathology and Adaptation*, (ahead-of-print), 2021.
- [30] Yiqing Liu, Justin KW Yeoh, and David KH Chua. Deep learning-based enhancement of motion blurred uav concrete crack images. *Journal of computing in civil engineering*, 34(5):04020028, 2020.
- [31] Hyo Seon Park, HM Lee, Hojjat Adeli, and I Lee. A new approach for health monitoring of structures: terrestrial laser scanning. *Computer-Aided Civil and Infrastructure Engineering*, 22(1):19–30, 2007.
- [32] Min-Koo Kim, Hoon Sohn, and Chih-Chen Chang. Localization and quantification of concrete spalling defects using terrestrial laser scanning. *Journal of Computing in Civil Engineering*, 29(6):04014086, 2015.

- [33] Xingu Zhong, Xiong Peng, Shengkun Yan, Mingyan Shen, and Yinyin Zhai. Assessment of the feasibility of detecting concrete cracks in images acquired by unmanned aerial vehicles. *Automation in Construction*, 89:49–57, 2018.
- [34] Yit Lin Michael Chew. *Building facades: a guide to common defects in tropical climates*. World Scientific, 1998.
- [35] Jingjing Guo and Qian Wang. Human-related uncertainty analysis for automation-enabled façade visual inspection: A delphi study. *Journal of Management in Engineering*, 38(2):04021088, 2022.
- [36] Biyanka Ekanayake, Johnny Kwok-Wai Wong, Alireza Ahmadian Fard Fini, and Peter Smith. Computer vision-based interior construction progress monitoring: A literature review and future research directions. *Automation in construction*, 127:103705, 2021.
- [37] D Mader, R Blaskow, P Westfeld, and C Weller. Potential of uav-based laser scanner and multi-spectral camera data in building inspection. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41:1135, 2016.
- [38] Wallace Mukupa, Gethin W Roberts, Craig M Hancock, and Khalil Al-Manasir. A review of the use of terrestrial laser scanning application for change detection and deformation monitoring of structures. *Survey review*, 49(353):99–116, 2017.
- [39] Ko Tomita and Michael Yit Lin Chew. A review of infrared thermography for delamination detection on infrastructures and buildings. *Sensors*, 22(2):423, 2022.
- [40] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.
- [41] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. Acado toolkit—an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32, 2011.
- [42] Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 1981.
- [43] Claude E Duchon. Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology and Climatology*, 18(8):1016–1022, 1979.
- [44] Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991.
- [45] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence*, 32(6):1127–1133, 2010.

- [46] Zhiwei Xiong, Xiaoyan Sun, and Feng Wu. Robust web image/video super-resolution. *IEEE transactions on image processing*, 19(8):2017–2028, 2010.
- [47] Gilad Freedman and Raanan Fattal. Image and video upscaling from local self-examples. *ACM Transactions on Graphics (TOG)*, 30(2):1–11, 2011.
- [48] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [49] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.
- [50] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *2009 IEEE 12th international conference on computer vision*, pages 349–356. IEEE, 2009.
- [51] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.
- [52] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.
- [53] Olivier Rukundo and Hanqiang Cao. Nearest neighbor value interpolation. *arXiv preprint arXiv:1211.1768*, 2012.
- [54] Petr Hurtik and Nicolas Madrid. Bilinear interpolation over fuzzified images: enlargement. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2015.
- [55] Zhou Dengwen. An edge-directed bicubic interpolation algorithm. In *2010 3rd international congress on image and signal processing*, volume 3, pages 1186–1189. IEEE, 2010.
- [56] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017.
- [57] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.
- [58] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016.

- [59] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014.
- [60] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.
- [61] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.
- [62] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.