Project No. R923070

Investigation of UAS Platform and Personnel Development and Preparation through Lab and Site Testing

Final Report
October 1, 2023 – December 9, 2024

Report Submitted to
Research Bureau
New Mexico Department of Transportation
7500B Pan American Freeway NE
Albuquerque, NM 87199-4690

Prepared by

Pouya Almasi
Graduate Research Assistant
Department of Civil Engineering
New Mexico State University

Qianyun Zhang
Assistant Professor
Department of Civil Engineering
New Mexico State University

Su Zhang, Assistant Professor
Department of Geography and Environmental Studies
University of New Mexico

December 1, 2024

SUMMARY PAGE

<table>
<tr><td colspan="2">1. Report No. R923070</td><td colspan="2">2. Recipient’s Catalog No.</td></tr>
<tr><td colspan="2">3. Title and Subtitle
Investigation of UAS Platform and Personnel Development and Preparation through Lab and Site Testing</td><td colspan="2">4. Report Date
12/1/2024</td></tr>
<tr><td colspan="2">5. Author(s):
Pouya Almasi, Qianyun Zhang, Su Zhang</td><td colspan="2">6. Performing Organization Report No.</td></tr>
<tr><td colspan="2" rowspan="2">7. Performing Organization Name and Address
New Mexico State University
Civil Engineering Department
3035 S Espina St.
Las Cruces, NM 88003-8001</td><td colspan="2">8. Performing Organization Code</td></tr>
<tr><td colspan="2">9. Contract/Grant No.</td></tr>
<tr><td colspan="2" rowspan="2">10. Sponsoring Agency Name and Address

Research Bureau, NMDOT
7500B Pan American Freeway
Albuquerque, NM 87199-4690</td><td colspan="2">11. Type of Report and Period Covered

Final Report
October 1, 2023 – December 09, 2024</td></tr>
<tr><td colspan="2">12. Sponsoring Agency Code
NMDOT</td></tr>
<tr><td colspan="4">13. Supplementary Notes
None</td></tr>
<tr><td colspan="4">14. Abstract
we propose a pilot program to initialize the UAS-based bridge inspection program in component level to enhance the bridge inspection practice in New Mexico State. In the proposed program, UAS-based bridge inspection strategies, data interpretation frameworks, and implementation software for individual NBI classified bridge components will be developed. Advanced UAS platforms and high-resolution sensors will be utilized for automated data capturing. Novel image processing and machine learning algorithms will be investigated to interpret the collected data in a semi-auto/automated manner. Easy-to-use software based on the developed algorithms will be developed for implementation usage. In addition, workshops will be hosted for NMDOT personnel to demonstrate the research outcomes from the proposed project to extend the research impact. This project serves as an initialization stage of the proposed pilot program. In this project, an advanced UAS platform will be developed, pilots will be trained, laboratory flight and field tests has been performed to evaluate and optimize the UAS platform.</td></tr>
<tr><td colspan="2">15. Key Words
Bridge Inspection; UAS; Remote Sensing; Pilot Training.</td><td colspan="2">16. Distribution Statement

Available from NMDOT Research Bureau</td></tr>
<tr><td>17. Security Classification of this Report
None</td><td>18. Security Classification of this page
None</td><td>19. Number of Pages
58</td><td>20. Price
N/A</td></tr>
</table>

PREFACE

NOTICE

The United States government and the State of New Mexico do not endorse products or manufacturers. Trade or manufactures' names appear herein solely because they are considered essential to the object of this report. This information is available in alternative accessible formats. To obtain an alternative format, contact the NMDOT Research Bureau, 7500B Pan American Freeway NE, PO Box 94690, Albuquerque, NM 87199-4690, (505)-841-9145

DISCLAIMER

This report presents the results of research conducted by the authors and does not necessarily reflect the views of the New Mexico Department of Transportation. This report does not constitute a standard or specification.

TABLE OF CONTENTS

LIST OF FIGURES

TASK 1: Personnel Training and Certification

In this Section, a guideline is developed for either First-time Pilots or Existing Part 61 Certificate Holders. The first part of this guideline is for first-time pilots, and the second part of this guideline is for existing part 61 certificate holders.

1.1 First-time Pilots

1.1.1 Eligibility

1) Be at least 16 years old
2) Be able to read, speak, write, and understand English
3) Be in a physical and mental condition to safely fly a drone
4) Pass the initial aeronautical knowledge exam: "Unmanned Aircraft General –Small (UAG)"

1.1.2 Steps to Become a Drone Pilot

Step 1: Obtain an FAA Tracking Number (FTN) by creating an Integrated Airman Certification and Rating Application (IACRA) profile prior to registering for a knowledge test.10

Step 2: Schedule an appointment with an FAA-approved Knowledge Testing Center. Be sure to register for the ***"Unmanned Aircraft General –Small (UAG)"*** test and bring a government-issued photo ID to your test.

Step 3: Study UAG-related knowledge to pass the test. Knowledge test topic areas include:

1) Applicable regulations relating to small unmanned aircraft system rating privileges, limitations, and flight operation
2) Airspace classification and operating requirements, and flight restrictions affecting small unmanned aircraft operation
3) Aviation weather sources and effects of weather on small unmanned aircraft performance
4) Small unmanned aircraft loading and performance
5) Emergency procedures
6) Crew resource management
7) Radio communication procedures
8) Determining the performance of small unmanned aircraft
9) Physiological effects of drugs and alcohol
10) Aeronautical decision-making and judgment
11) Airport operations
12) Maintenance and preflight inspection procedures
13) Operation at night

Useful downloadable material to be reviewed:

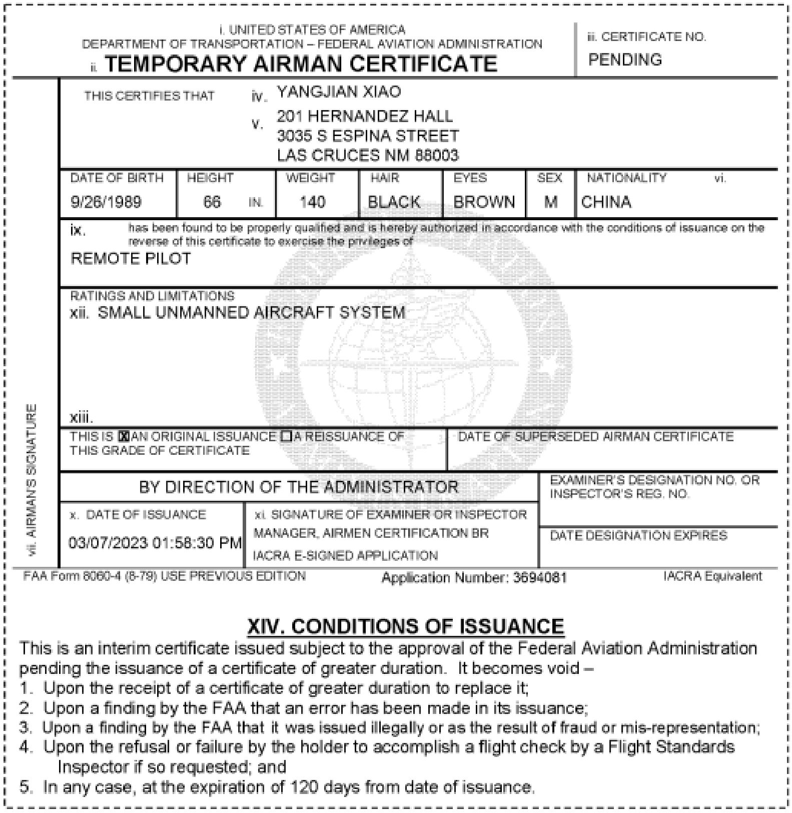
1) Unmanned Aircraft General Sample Questions (download here)
2) Remote Pilot –Small Unmanned Aircraft Systems Study Guide (download here)
3) Title 14 CFR Part 107 (download here)
4) Advisory Circular AC 107-2A (download here)
5) Airman Knowledge Testing Supplement for Sport Pilot, Recreational Pilot, Remote Pilot, and Private Pilot (FAA-CT-8080-2H) (download here)
6) Title 14 CFR Part 47 (download here)
7) Title 14 CRF Part 48 (download here)
8) Remote Pilot Airman Certification Standards (download here)

Step 4: Pass the initial aeronautical knowledge test: "Unmanned Aircraft General –Small (UAG)".

Step 5: Complete FAA Form 8710-13 for a remote pilot certificate (FAA Airman Certificate and/or Rating Application) using the electronic FAA Integrated Airman Certificate and/or Rating Application system (IACRA)*

1) Login with username and password
2) Click on "Start New Application": Application Type "Pilot", Certifications "Remote Pilot", Other Path Information, Start Application
3) Follow application prompts
4) When prompted, enter the 17-digit Knowledge Test Exam ID (Note: it may take up to 48 hours from the test date for the knowledge test to appear in IACRA)
5) Sign the application electronically and submit it for processing.

Step 6: A confirmation email will be sent when an applicant has completed theTSAsecurity background check. This email will provide instructions for printing a copy of the temporary remote pilot certificate from IACRA as shown in Figure 1.



i. UNITED STATES OF AMERICA
DEPARTMENT OF TRANSPORTATION – FEDERAL AVIATION ADMINISTRATION
ii. **TEMPORARY AIRMAN CERTIFICATE**

iii. CERTIFICATE NO.
PENDING

THIS CERTIFIES THAT iv. YANGJIAN XIAO
v. 201 HERNANDEZ HALL
3035 S ESPINA STREET
LAS CRUCES NM 88003

DATE OF BIRTH	HEIGHT	WEIGHT	HAIR	EYES	SEX	NATIONALITY vi.
9/26/1989	66 IN.	140	BLACK	BROWN	M	CHINA

ix. has been found to be properly qualified and is hereby authorized in accordance with the conditions of issuance on the reverse of this certificate to exercise the privileges of

REMOTE PILOT

RATINGS AND LIMITATIONS
xii. SMALL UNMANNED AIRCRAFT SYSTEM

xiii.

THIS IS ☒ AN ORIGINAL ISSUANCE □ A REISSUANCE OF THIS GRADE OF CERTIFICATE

DATE OF SUPERSEDED AIRMAN CERTIFICATE

BY DIRECTION OF THE ADMINISTRATOR

EXAMINER'S DESIGNATION NO. OR INSPECTOR'S REG. NO.

x. DATE OF ISSUANCE
03/07/2023 01:58:30 PM

xi. SIGNATURE OF EXAMINER OR INSPECTOR
MANAGER, AIRMEN CERTIFICATION BR
IACRA E-SIGNED APPLICATION

DATE DESIGNATION EXPIRES

vii. AIRMAN'S SIGNATURE

FAA Form 8060-4 (8-79) USE PREVIOUS EDITION Application Number: 3694081 IACRA Equivalent

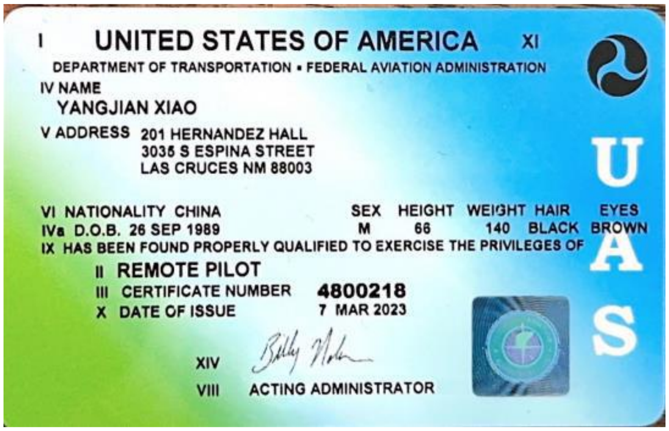
XIV. CONDITIONS OF ISSUANCE

This is an interim certificate issued subject to the approval of the Federal Aviation Administration pending the issuance of a certificate of greater duration. It becomes void –

1. Upon the receipt of a certificate of greater duration to replace it;
2. Upon a finding by the FAA that an error has been made in its issuance;
3. Upon a finding by the FAA that it was issued illegally or as the result of fraud or mis-representation;
4. Upon the refusal or failure by the holder to accomplish a flight check by a Flight Standards Inspector if so requested; and
5. In any case, at the expiration of 120 days from date of issuance.

Figure 1. Temporary remote pilot certificate

Step 7: A permanent remote pilot certificate (as shown in Figure 2) will be sent via mail once all other FAA-internal processing is complete.



I UNITED STATES OF AMERICA XI
DEPARTMENT OF TRANSPORTATION • FEDERAL AVIATION ADMINISTRATION
IV NAME
YANGJIAN XIAO
V ADDRESS 201 HERNANDEZ HALL
3035 S ESPINA STREET
LAS CRUCES NM 88003

VI NATIONALITY CHINA
IVa D.O.B. 26 SEP 1989

SEX	HEIGHT	WEIGHT	HAIR	EYES
M	66	140	BLACK	BROWN

IX HAS BEEN FOUND PROPERLY QUALIFIED TO EXERCISE THE PRIVILEGES OF
II REMOTE PILOT
III CERTIFICATE NUMBER 4800218
X DATE OF ISSUE 7 MAR 2023
XIV
VIII ACTING ADMINISTRATOR

Figure 2. Permanent remote pilot certificate

Step 8: Have your Remote Pilot Certificate available whenever you fly your UAS.

1.2 Existing Part 61 Certificate Holders

1.2.1 Eligibility

1) Must hold a pilot certificate issued under14CFRpart 61

2) Must have completed a flight review within the previous 24 months

1.2.2 Steps to Become a Drone Pilot

Step 1: Create an account, or log into your existing account, on the FAA Safety Team (FAASTeam) website.

Step 2: Complete the Part 107 Small UAS Initial (ALC-451) online training course. The course will cover these topic areas:

1) Applicable regulations relating to small unmanned aircraft system rating privileges, limitations, and flight operation
2) Effects of weather on small unmanned aircraft performance
3) Small unmanned aircraft loading and performance
4) Emergency procedures
5) Crew resource management
6) Determining the performance of small unmanned aircraft
7) Maintenance and preflight inspection procedures
8) Operation at night

Step 3: Create an account, or log into your existing account, in IACRA.

Step 4: Complete Form 8710-13 for a remote pilot certificate (FAA Airman Certificate and/or Rating Application) in IACRA.

1) Login with username and password
2) Click on "Start New Application" and 1. Application Type "Pilot", 2. Certifications "Remote Pilot", 3. Other Path Information, 4. Start Application
3) Follow application prompts
4) Sign the application electronically and submit for processing.

Step 5: Make an appointment with one of the following entities to validate your identity. Bring your completed Form 8710-13, proof of your current flight review, photo ID, and your online course completion certificate.

1) At an FAA Flight Standards District Office (FSDO)
2) With an FAA-designated pilot examiner (DPE)
3) An airman certification representative (ACR)
4) An FAA-certificated flight instructor (CFI)*

Please note: *CFIs cannot issue temporary certificates. They can process applications for applicants who do not want a temporary certificate.13

Step 6: The representative will sign your application and issue you a temporary airman certificate.

1) You'll receive your permanent certificate via U.S. mail within several weeks.

Step 7: Have your Remote Pilot Certificate available whenever you fly your UAS.

1.2.3 Requirements for Remote Pilot Certificate

1) Must be easily accessible by the remote pilot during all UAS operations
2) Certificate holders must complete an online recurrent training every 24 calendar months to maintain aeronautical knowledge recency

TASK 2: Hardware Selection

2.1 UAS Platform Selection

Before integrating UAVs into the airspace, a few aspects, including equipment features, pilot protocols, object qualities, surroundings, and safety rules, must be considered. It is crucial to have a thorough awareness of these aspects to allow safe and effective operations [1]. Some equipment-related aspects influence UAV performance including aerial system size and design, payload capacity and compatibility, battery capacity, control range distance, and duration for safe flight [2]. The typical UAV is composed of a frame, motors, control unit, onboard sensors, communication system, and power supply. Many UAVs display a dual tube substructure to make it easier to mount various payloads [3]. Fixed-wing, rotorcraft, multi-rotor drones, and hybrid vertical take-off and landing (VTOL) vehicles are the four common UAV designs [4]. Figure 3 shows examples of these different types of UAVs.

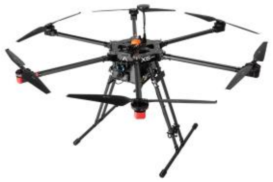
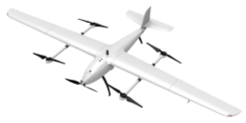
			
Fixed wing	Rotorcraft	Multi-rotor drones	Hybrid VTOL

Figure 3. Common UAV types

When delving into operational attributes and various airframes, fixed-wing drones mirror traditional aircraft, typically boasting larger sizes that efficiently cover extended distances. In comparison, rotorcraft UAVs employ rotating propellers integrated into their framework, akin to helicopters. Single-rotor drones exhibit enhanced efficiency over their multi-rotor counterparts. Functioning as compact helicopters powered by either gas or electricity, these single-rotor models represent a specialized niche. Among these, the multirotor UAV emerges, an evolution featuring multiple propellers extending from the core to amplify flight capabilities. This category finds common utility in applications such as aerial photography and surveillance. Variants include the Tricopter (with three rotors), Quadcopter (with four rotors), Hexacopter (with six rotors), and Octocopter (with eight rotors). Multirotor UAVs excel in intricate 3D mapping due to their agility and stability. Conversely, rotorcrafts demand more finesse in manual piloting compared to multirotors, thanks to their enhanced controllability, increased lift capacity, and built-in redundancy in case of motor malfunction. Measurement errors may arise from the UAV's instability or heightened maneuverability. Lastly, hybrid VTOL UAVs amalgamate fixed-wing and multirotor configurations, initially ascending vertically before transitioning into horizontal flight [2].

Several additional factors come into play when choosing the right platform for inspections. For close-range assessments, the chosen platform must possess the ability to maintain stability, even in the face of strong gusts. Typically, optimizing flight duration is paramount for various applications, often hinging on payload capacity and battery performance. A larger payload capacity allows for the accommodation of more sensors, but this can sometimes compromise overall flight time. Conversely, camera resolution and the desired quality of surveys can also influence flight duration. This is due to the correlation between higher-resolution cameras and the potential for shorter flight paths [2]. The choice of the appropriate UAV platform and sensors has proven to be a difficult problem due to UAV performance requirements related to flights close to the bridge structure (e.g., turbulent flow characteristics around the bridge) and terrain characteristics (e.g., surface roughness, temperature, and humidity) [5]. This includes positioning and maneuvering the UAV around or under bridges (operations prohibited by GPS) [6] and the stability of the platform in windy situations, where turbulence and other aerodynamic phenomena cause unpredictable wind effects [7]. The main difficulties in maximizing the UAV for bridge inspections are striking a balance between payload capacity, endurance, vehicle stability, and navigational capabilities. Figure 4 indicates the parameters for UAV platform selection. A vehicle with a stabilizing gimbal that can change the camera pointing angle to any vertical angle, a camera with optical zoom for capturing high-resolution imagery while at a safe standoff distance, a vertical takeoff and landing capability, and the capacity to hover in place during the flight are examples of parameters and vehicle characteristics appropriate for bridge inspections. In addition, Figure 5 shows some of the UAV platforms used in former studies. Figure 6 indicates the cost-endurance of these different UAV platforms.

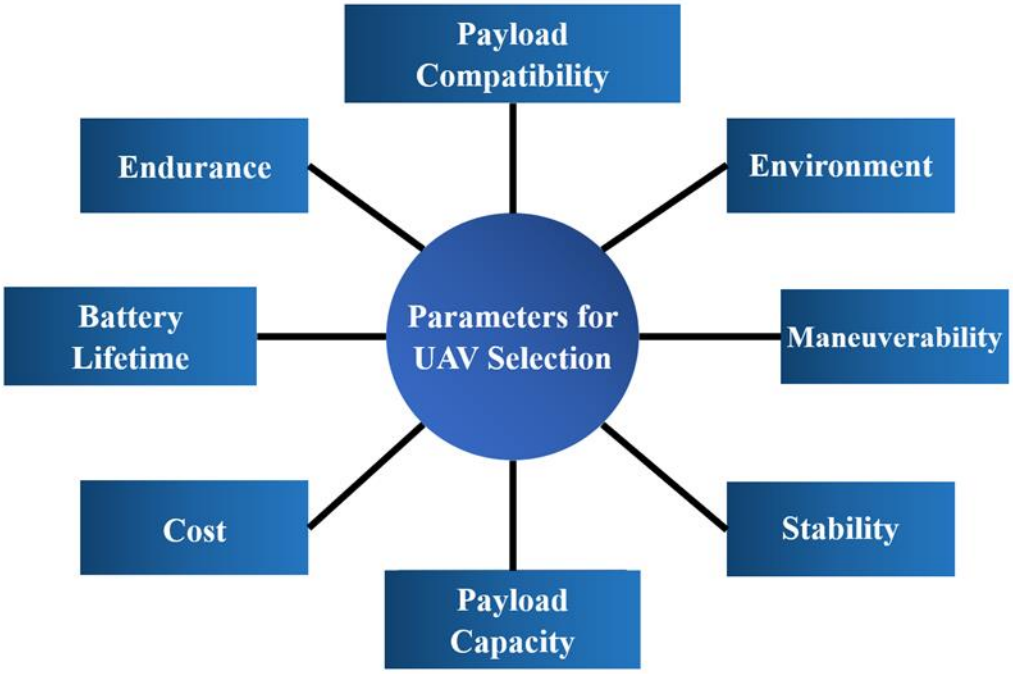


Figure 4. Considerable parameters for UAV platform selection

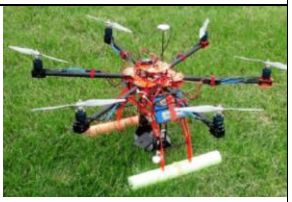
UAV	DJI Mavic	Aurelia X6 Standard LE	DJI Phantom 4	senseFly Albris
Price	$2700	$5700	$3000	$2000
Maximum Endurance	31 minutes	45 minutes	30 minutes	22 minutes
Payload Capacity	1 kg	5 kg	1 kg	N/A
				
UAV	3DR Solo	3DR Iris	DJI Inspire 1 Pro	Bergen hexacopter
Price	$1000	$750	$3900	$6000
Maximum Endurance	15 min	22 min	18 min	30 min
Payload Capacity	1.5 kg	0.4 kg	3.4 kg	5 kg

Figure 5. Examples of UAVs with some specs

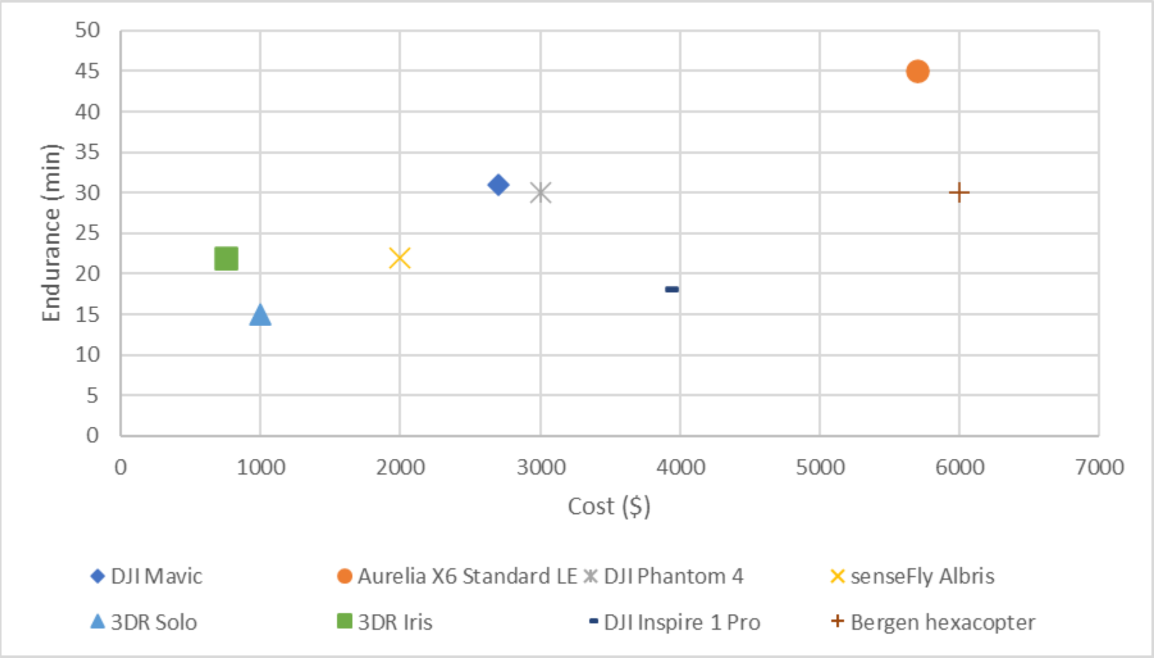


Figure 6. Cost-endurance relation of different UAV platforms

2.2 Payload Selection

The choice of the best sensors for bridge inspection depends on several factors, including cost, flight time, mission objectives, the UAV's payload capacity, and navigational needs [6]. These factors are shown in Figure 7.

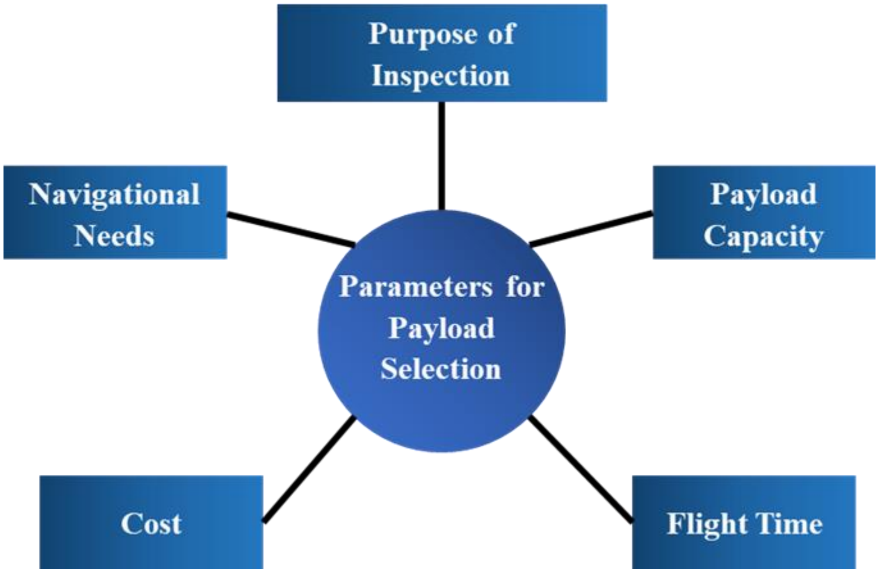


Figure 7. Considerable parameters for payload selection

The most important factor is the purpose of the inspection. Without considering the purpose of inspection, a suitable payload may not be chosen and consequently, it will lead to some problems in other steps such as low data quality or inappropriate data collection. Five common purposes are defined in this study: crack detection, delamination, fatigue, 3D modeling, and corrosion. These purposes of inspection are discussed briefly below with the proper sensors.

- Crack: Most of the studies mentioned crack detection as the basic and major application of bridge inspection using UAVs [3]. There are two primary steps in the image-based surface crack assessment method. Crack detection comes first, with the goal of removing noise and extracting cracked objects from the images. The extraction of crack edges and the calculation of crack characteristics, such as crack width and length, make up the second stage of crack assessment [2]. For this purpose, mostly RGB cameras are used to find cracks in the surface of bridges. Using optical cameras, the UAVs can take high-quality pictures from the bridge's difficult-to-reach regions. It is worth mentioning that due to the wide use of this purpose, the general framework of this study is generated for crack detection, but it can be applied for other purposes too with some adjustments.
- Delamination: Deck delamination, also known as horizontal debonding in the deck's subsurface, is frequently a sign that the deck reinforcement has deteriorated due to corrosion. The shape and depth of delamination, environmental factors like air temperature and solar intensity, which introduce feature variation of the same delamination, and surface textures like cracks, color differences, patching, and road painting, which add external noise, are the current challenges for the purpose of delamination profiling through thermography [8].
- Fatigue: Fatigue cracks can have lengths less than 7 mm and diameters as small as 0.1 mm, and they are exceedingly difficult to discern. Inaccessible areas such as huge cross frames, welded stiffeners, or other complex geometries are typically where fatigue cracks develop in the superstructure. Commonly, RGB and IRT cameras are utilized to identify fatigue cracks. The effectiveness of UAV-based fatigue crack detection is greatly influenced by the platform that is used, the environment, and the lighting [9].
- 3D Modeling: Bridge managers can view geometric data, such as damage location, and surface condition, such as damage kind and amount, by using 3D models of the structures, which provide a base from which damage information can be compared. To create 3D models, RGB cameras and LiDAR sensors can be used [10]. Photogrammetry creates 3D points from a set of 2D photos collected from various angles and positions all around the structure, as opposed to LiDAR, which often contains more 3D points. Photogrammetry has a higher processing cost and lower accuracy than LiDAR because it compares image attributes to build the 3D points. However, UAV-based LiDAR systems need expensive LiDAR sensors and GPS systems, which reduce battery life by adding more payload to the system, whereas photogrammetry merely needs an optical sensor.
- Corrosion: A positive charge is released during the electrochemical process of corrosion, which results in the formation of a stable compound. Despite some corrosion on the underlying metal components, such as the steel reinforcement used in bridge concrete, the surface of steel bridges experiences a great deal of corrosion. The most popular cameras for detecting corrosion are RGB and IRT cameras. Although infrared thermography is a promising technique for measuring, mapping, and detecting corrosion, more study is required before it can be used perfectly in the field [11].

Table 1 gives a summary of 3 commonly used payloads and some other information that is effective for the suitable payload selection. It is worth mentioning that if needed, using multi-sensors is possible if they are compatible with the UAV platform.

Payload Type	Weight Range (kg)	Mission	Limitations
Visual Camera	0.1 - 1	-Crack Detection -Fatigue -Delamination -Corrosion -3D Modeling	- Vibration and wind effect - Lightning condition - GPS deprived navigation
IRT Camera	0.2 – 1.5	-Fatigue -Delamination	- Low pixel resolution - Inspection time affects the results
LiDAR Sensor	1.3 – 2.8	-3D Modeling	- High weight - High price

Table 1. Specifics of different payloads and their limitations

2.3 Recommendations

UAV platform selection:

- Type of Drone: Select the drone based on the application. For bridge inspection, a multi-rotor UAV is recommended.

- Stability: Prioritize drones with advanced stabilization systems to handle varying wind conditions and ensure precise data collection.

- Payload Capacity and Compatibility: Choose a drone with sufficient payload capacity to accommodate necessary sensors and cameras without compromising flight time.

- Flight Duration: Balance payload capacity and battery performance to achieve a suitable flight duration that covers the inspection area effectively. An extra battery is recommended.

- Cost Efficiency: Evaluate the costs associated with the drone, additional equipment, and maintenance to align with your budget.

Payload Selection:

- Camera or Sensor Type: Select cameras or sensors that align with the inspection objectives. Consider RGB cameras for visual inspections, thermal cameras for detecting structural anomalies, or LiDAR for generating detailed 3D models.

- Resolution and Accuracy: Select cameras or sensors with appropriate resolution and accuracy levels to capture the required level of detail for the inspection.

- Integration: Ensure that the chosen payload can be seamlessly integrated with your drone's existing systems and can be easily attached and removed when needed.

- Cost-Effectiveness: Balance the capabilities of the payload with its cost to ensure it meets your inspection requirements within your budget.

TASK 3: Pre-flight Preparation for Data Collection

3.1 Flight Path Planning

A well-planned flight path is of paramount importance when using UAVs for inspection operations. To unlock the full potential of the UAV, the mission must be carefully designed to encompass all inspection targets [12], [13]. However, path planning for UAV-based bridge inspection presents challenges in finding the optimal or near-optimal path. The flight path comprises a set of camera positions from which images will be captured. These camera positions are determined by their horizontal and vertical distance from the object, as well as the angle of the camera. In the following two subparts of this section, we will delve into discussions on camera positions and flight path planning, exploring the crucial aspects of these elements in UAV-based bridge inspection [14].

In the process of selecting camera positions, one of the key considerations is the Ground Sampling Distance (GSD). The Ground Sampling Distance refers to the distance between two consecutive pixel centers measured on the ground. It plays a crucial role in determining the spatial resolution of the image and the level of visible details. A larger GSD value corresponds to lower spatial resolution, resulting in fewer visible details in the captured images. Figure 8 provides a visual representation of GSD and its associated parameters, illustrating its significance in the context of UAV-based bridge inspection.

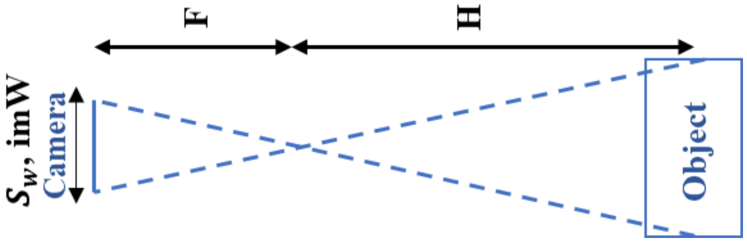


Figure 8. GSD example with related parameters

As shown in Figure 8, H is the flight height in meters, imW is the image width in pixels, F is the real focal length of the camera in millimeters, SW is the sensor width in millimeters, and GSD is the ground sampling distance in centimeters/pixels.

The equation for calculating the GSD is:

$$GSD = \frac{S_W \times H}{F_R \times imW} \times 100 \qquad (1)$$

It is important to decide on the GSD value before starting the image acquisition in order to adjust the flight height and the camera specifications to the project requirements. From equation (1), the required flight height can be calculated if the GSD is defined for an inspection as below:

$$H = \frac{GSD \times F_R \times imW}{S_w \times 100} \qquad (2)$$

Also, by knowing the image height width and height of the single image footprint on the ground (distance covered on the ground by one image in width and height directions) can be calculated as below:

$$D_w = GSD \times imW \qquad (3)$$

$$D_H = GSD \times imH \qquad (4)$$

where imH is the image height in pixels, and D_w, D_H are the width and height of a single image footprint on the ground in meters respectively.

The selection of GSD and camera positions depends on the specific purpose of the inspection and the characteristics of the payload. Debus et al. propose three distinct levels of interest for conducting the inspection, [12]. 2.0 mm/pixels, 1.0 mm/pixels, and 0.1 mm/pixels are defined as level 1(for rough geometry), level 2 (for detailed geometry), and level 3 (for crack detection) of interest respectively, which provides valuable guidance for tailoring the GSD and camera positions to effectively meet the inspection objectives. Also, according to the Specifications for the National Bridge Inventory 2022 (SNBI), the following quantitative standards are considered to categorize the cracks by their width:

- Insignificant - crack width less than 0.004 inches (prestressed) or 0.012 inches (reinforced), or medium width cracks that have been sealed.
- Medium - crack width ranging from 0.004 – 0.009 inches (prestressed) or 0.012 to 0.05 inches (reinforced).
- Wide - crack width wider than 0.009 inches (prestressed) or 0.05 inches (reinforced).

While the defined levels of interest are valuable as a starting point, it is crucial to recognize that they may need to be adjusted based on the specific requirements of each inspection task. Different cases may demand varying levels of interest to effectively address the inspection objectives. Additionally, even when flying at a constant height, the images captured during the project may exhibit variations in GSD. These discrepancies arise due to differences in terrain elevation and changes in the camera angle during image capture. To ensure comprehensive coverage and accurate data collection, it is generally recommended to have at least a 50% overlap of images between consecutive camera positions, as suggested by various studies [15]. This overlapping ensures that critical details are captured redundantly, minimizing the risk of missing essential information.

Once a set of points has been chosen, taking into account the previously mentioned parameters, the next crucial step is to determine the most optimum or near-optimum flight path. This task poses a challenging optimization problem, as it involves minimizing flight time (energy) or path while considering all relevant parameters.

3.2 Camera Calibration

Camera calibration is an essential step to extract metric data from 2D photos in 3D computer vision. Over the years, numerous studies have explored camera calibration, initially in the field of photogrammetry and more recently in the computer vision community [16], [17]. In aerial images, pre-calibration or on-the-job calibration is frequently used to handle camera parameters, such as intrinsic parameters and lens distortion coefficients. The goal of camera calibration is to establish the relationship between the 3D world coordinates of the object and their corresponding 2D image coordinates, forming the projection matrix.

We assume a point on the object such as X_w and the projection of this point in the captured image such as *U*. Coordinates of these points are as shown below:

$$X_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \qquad (5)$$

$$U = \begin{bmatrix} u \\ v \end{bmatrix} \qquad (6)$$

Where x_w, y_w, z_w are the coordinates of a known object in millimeters or inches, and *u*, *v* are the coordinates of the projection of that known point in the captured image in pixels. This can be done for every corresponding point of object and captured image. For each corresponding point *i* in the scene and image, we get a mapping from the point in 3D coordinates to the image coordinates in 2D using a projection matrix:

$$\begin{bmatrix} u^{(i)} \\ v^{(i)} \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{134} \end{bmatrix} \begin{bmatrix} x_w^{(i)} \\ y_w^{(i)} \\ z_w^{(i)} \\ 1 \end{bmatrix} \qquad (7)$$

As shown in equation (7), the only unknown matrix is the projection matrix which should be estimated.

With the fundamental steps for camera calibration, specifically for reference object-based calibration, various patterns and benchmarks can be utilized to perform the calibration. In this study, the commonly used and straightforward checkboard pattern is employed. Figure 9 displays the two checkboard patterns utilized for camera calibration in this study. Checkboard patterns are selected because of their simplicity and almost all the calibration tools are compatible with this type of benchmark. The control points for this pattern are the corners that lie inside the checkerboard. Because corners are extremely small, they are often invariant to perspective and lens distortion. The calibrator apps can also detect partial checkerboards, which can be useful when calibrating cameras with wide-angle lenses. A checkerboard should contain an even number of squares along one edge and an odd number of squares along the other edge, with two black corner squares along one side and two white corner squares on the opposite side. This enables the app to determine the orientation of the pattern and the origin. The calibrator assigns the longer side as the *x*-direction. A square checkerboard pattern can produce unexpected results for camera extrinsics.

In general, the checkboard size will not affect the camera calibration process or intrinsic and extrinsic camera parameters very much in mathematical representation. However, it is important practically but, if their size is within the recommended ranges for calibration tools (such as 15, 20, 30, 40, 50, or 60 millimeters) considering the distance from the target, it will not affect the crack detection. If the check squares are very small, probably the corners of the squares will not have appropriate quality and that will make the data "noisier". It is worth noting that more squares (more corners between squares) will give better results since there will be a more overdetermined system of equations to be solved [18].

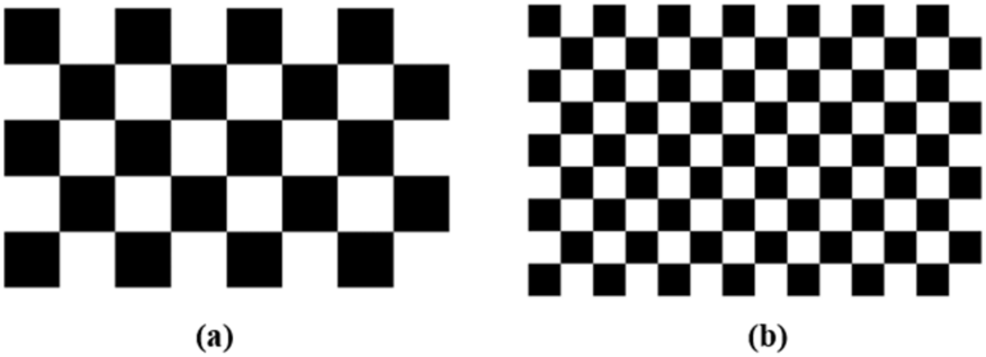


Figure 9. Checkboard patterns for camera calibration. (a) 297 mm×400 mm with check square size of 50mm (b) 400 mm×600 mm with check square size of 40mm

For the purpose of camera calibration in this study, the MATLAB camera calibration toolbox will be used which uses a robust feature detection algorithm based on Zhang's method [19], [20], and this approach will help to reduce potential errors related to feature point detection. By using this method and the checkboard benchmarks shown above, the internal and external parameters of the camera will be determined. The first important parameter is the reprojection error which is the distance between a pattern key point detected in a calibration image, and a corresponding world point projected into the same image. The acceptable and recommended mean reprojection error is less than 1 pixel but an error less than 0.5 pixel is better for a good alignment [21]. Another parameter to consider is the focal length in the x and y directions (f_x, f_y) which is in pixels, and the relation between this focal length and real focal length (F) is shown below:

$$f_x = F \times s_x \qquad (8)$$

$$f_y = F \times s_y \qquad (9)$$

Where F is the focal length in millimeters, and s_x, s_y are the number of pixels per millimeter in the x- and y-direction respectively.

In the methodology proposed in this study, there are two kinds of camera calibrations. In-lab camera calibration is a one-time calibration, and it is beneficial for flight path planning based on the intrinsic and extrinsic parameters of the drone's camera where this calibration gives a better understanding of the required flight plan considering different camera positions (different heights and angles) and their corresponding reprojection errors and distortion. On the other hand, the results of the on-site calibration will give the parameters considering the real-world conditions for the bridge inspection and due to the stability issues with the drones, these parameters from on-site calibration will result in calculating the exact GSD for the collected data which is crucial for the data processing and crack detection. While the required flight height was determined by using GSD from the previous section, considering the stability issue with UAVs, there would be a tolerance in the on-site flight. But, using the focal length in pixels from calibration and by knowing the real focal length of the camera in millimeters, the exact GSD for each image can be easily found by using equations 8 and 9 which would be beneficial for more accurate crack detection. Other than stability, another source of error could be lighting conditions which have been considered and explained in the experiments. Also, some other potential sources of error could be lens distortion, human error, and other environmental factors. To mitigate these errors, as will be presented in the experiments, multiple images have been collected.

Finally, the other important parameter is the radial distortion in the x (RD_x) and y (RD_y) directions. Radial distortion is the displacement of image points along radial lines extending from the principal point and it occurs when light rays bend more near the edges of a lens than they do at its optical center.

3.3 Case Studies and Results

3.3.1 Inspection Purpose and Bridges

To evaluate the feasibility of the proposed framework, two case studies have been carried out. The purpose of the inspection is generally to detect cracks on the top surface and side surface of the bridge decks.

Two bridges have been selected for the case study. The first bridge is located at 4.8 Mi N of Sierra C/L, New Mexico, United States (NBI bridge number: 01791). This bridge is in fair condition and consists of 5 simple spans at 39' each, 6 steel girders per span, full-height concrete abutments with concrete wingwalls, concrete pier caps on concrete pier walls, and a CIP concrete deck, as shown in Figure 10. The second bridge is in satisfactory condition according to the reports of the New Mexico Department of Transportation and is located at 1.9 Mi W of NM-28/NM-359, New Mexico, United States (NBI bridge number: 06255). The bridge consists of 8 spans, 2 units of 4 continuous spans at 54ft, 69ft-5in, 69ft-5in, and 54 ft, 5 rolled steel girders per span, concrete stub abutments, concrete pier caps on steel piles, and CIP concrete deck, as shown in Figure 11.

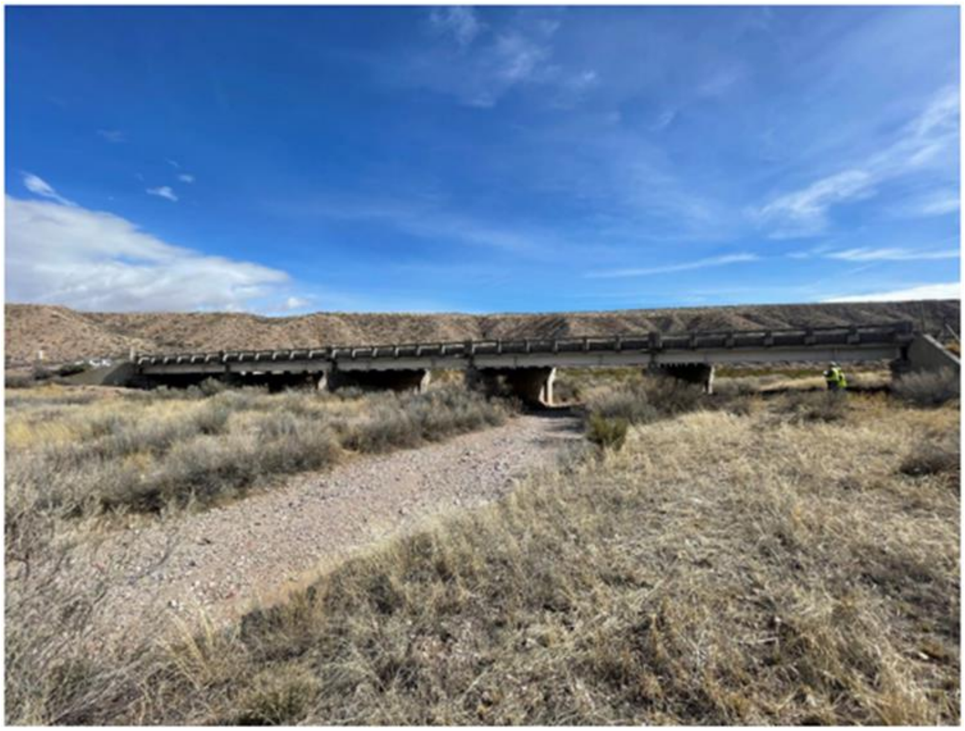


Figure 10. Profile view of the first bridge facing north (NBI number: 01791)

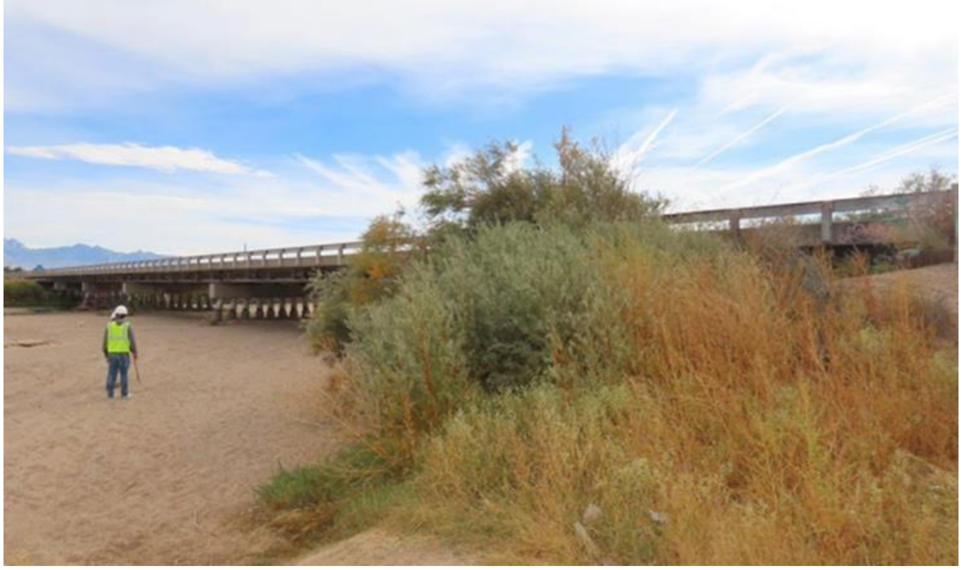


Figure 11. Profile view of the second bridge facing south (NBI number: 06255)

3.3.2 In-lab Calibration

In-lab calibration using various camera positions would be helpful to understand the camera parameters which would be beneficial for flight path planning. Considering the flight height range based on inspection purposes and GSD, various images are captured from different heights, angles, and directions. The results of the camera calibration for these sets of images will lead to better flight path planning and a better selection of camera positions for on-site bridge inspection.

The in-lab calibrations take place in a parking lot at New Mexico State University. During this calibration process, three distinct sets of images are captured from various heights, angles, and directions. Each set contains 36 photos with different heights and angles (shown in Table 2), resulting in a total of 108 different images used for camera calibration. The calibration results are then categorized into three parts, corresponding to the camera's direction, namely south, east, and north. In Figure 12, an example of captured images is depicted, along with their detected and projected points used for in-lab calibration. This approach of using different groups of images in three separate sets makes it more convenient for an inspector or pilot to create a flight plan, taking into account the calibration results from these different image groups. This division aids in tailoring the flight plan according to specific camera orientations and ensures accurate imaging during the inspection process.

Heights (m)	**Angles (°)**
0.5, 1, 1.5, 2, 2.5, 3	15, 30, 45, 60, 75, 90

Table 2. Heights and angles for in-lab calibration for each direction

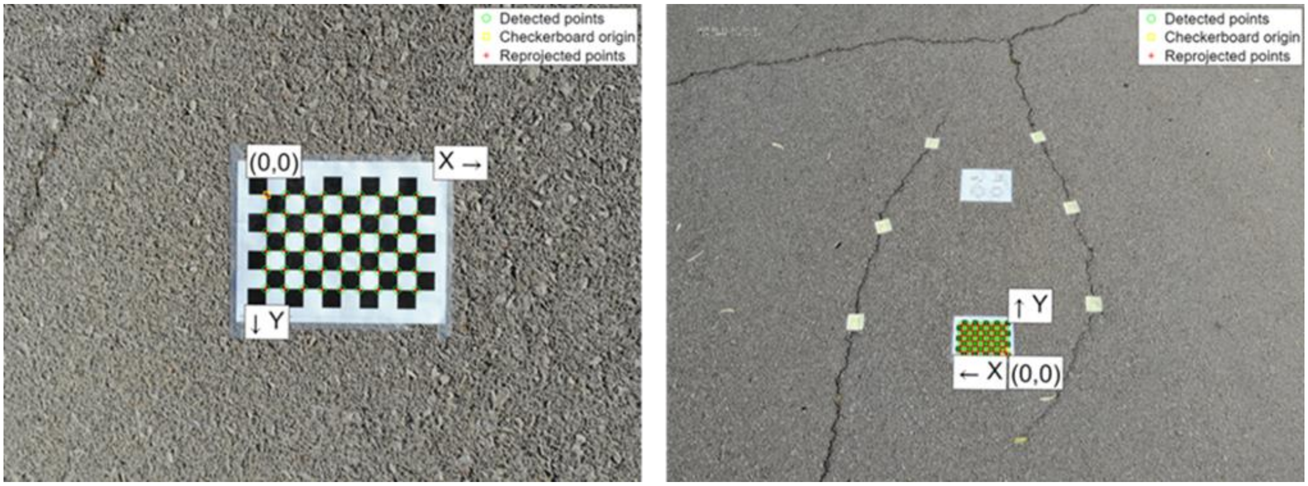


Figure 12. Different pictures from in-lab calibration with shown detected and reprojected points

Figures 13, 14, and 15 display the reprojection errors, overall mean error, and trendline obtained from data captured by the camera in the eastward, northward, and southward directions respectively. The overall mean reprojection error, as well as the focal length in the x and y directions, and radial distortion in the x and y directions are calculated based on equation 7, and by using MATLAB camera calibration, and presented in Table 3 for all three groups of images. When comparing the overall mean reprojection error with the individual error of each image within each group, it is observed that 41.7%,33.3%, and 36.1% of the images have a higher error than the mean error for eastward, northward, and southward groups respectively. Also, the images captured from the south direction have a noticeably lower overall mean error. It is worth mentioning that 2 images from the east side were rejected during calibration data processing due to the high reflection of sunlight on the benchmark. This analysis provides valuable insights into the accuracy and consistency of the calibration results for this set of images. By considering these results, the southward which is in the direction of the back to the sun has less mean reprojection error than the other sides and in this set of images, pictures with a height between 2m to 3m have fewer reprojection errors. Finally, the images which have been captured from angles 30 to 45 have less reprojection error. This information will be used in the next step for better flight path planning.

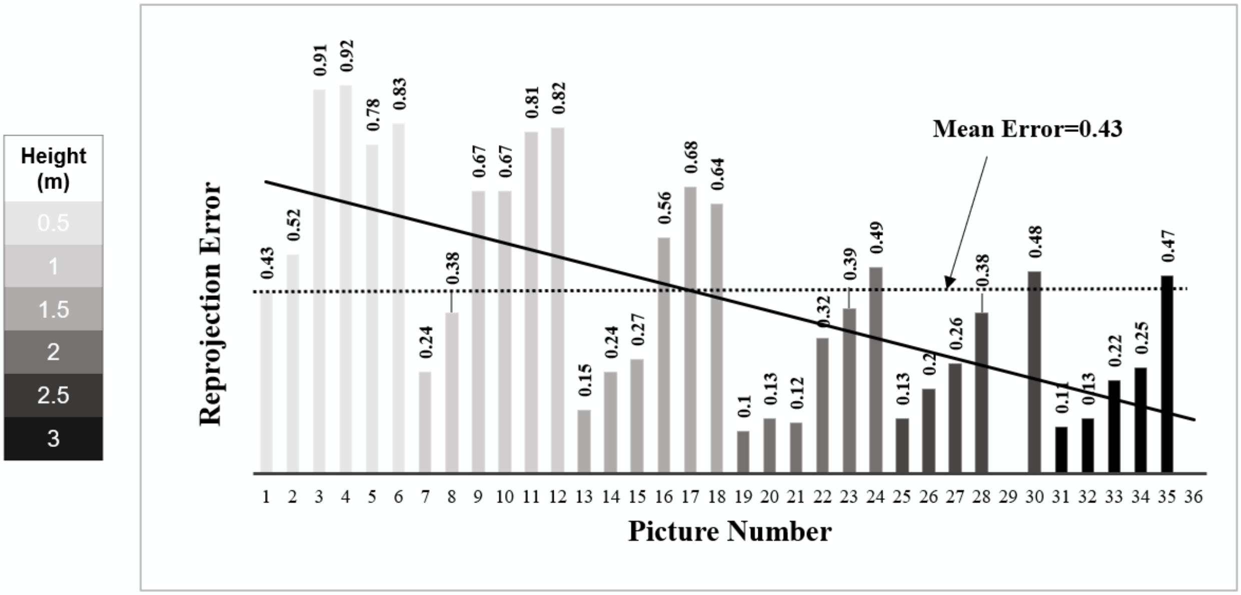


Figure 13. The reprojection errors of the pictures from the east side and the overall mean error

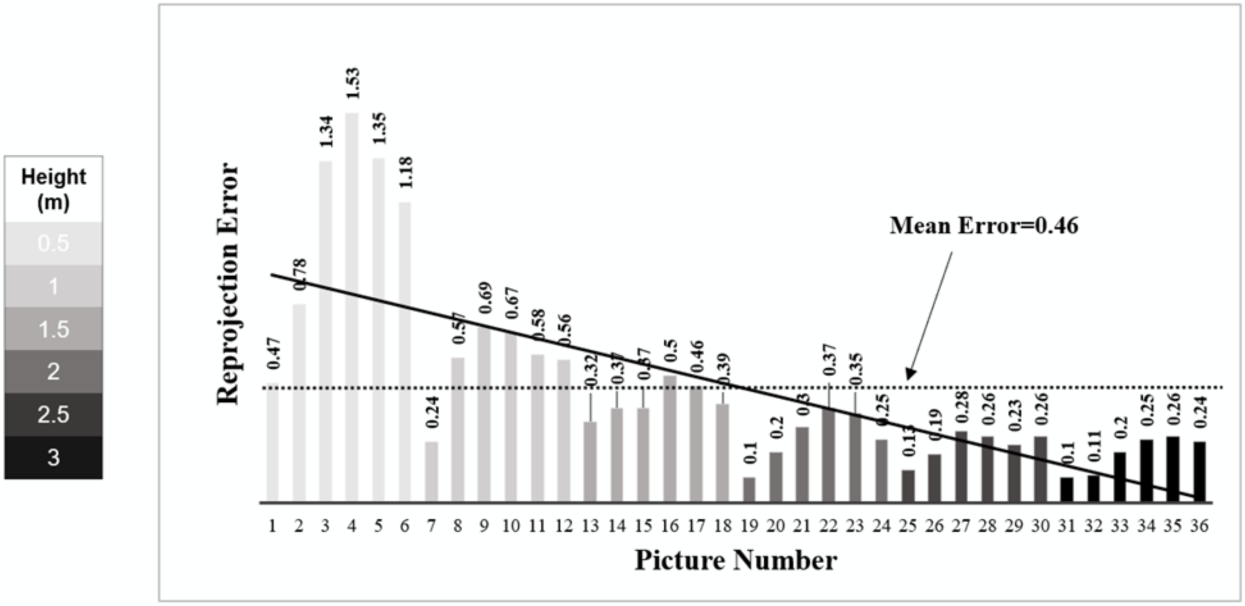


Figure 14. The reprojection errors of the pictures from the north side and the overall mean error

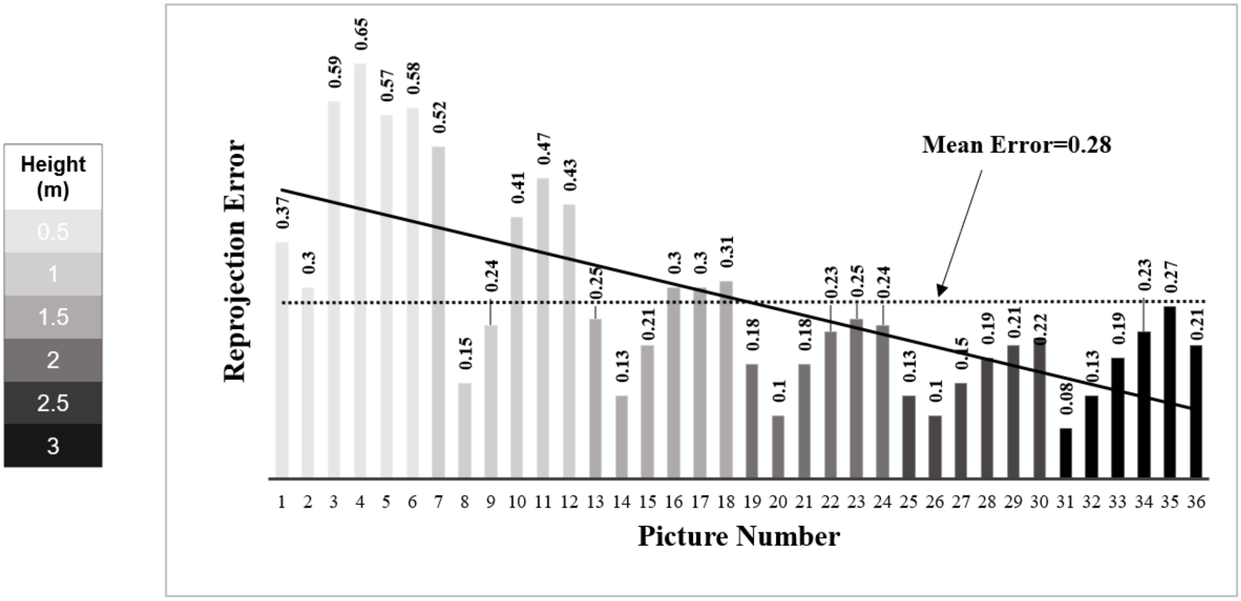


Figure 15. The reprojection errors of the pictures from the south side and the overall mean error

		East Direction		
Overall Mean Error (pixels)	x Focal Length (pixels)	y Focal Length (pixels)	x Radial Distortion (pixels)	y Radial Distortion (pixels)
0.43	$f_x = 3199$	$f_y = 3193$	$RD_x = 0.0548$	$RD_y = -0.2710$
		North Direction		
Overall Mean Error (pixels)	x Focal Length (pixels)	y Focal Length (pixels)	x Radial Distortion (pixels)	y Radial Distortion (pixels)
0.46	$f_x = 2993$	$f_y = 2995$	$RD_x = -0.0184$	$RD_y = -0.2410$
		South Direction		
Overall Mean Error (pixels)	x Focal Length (pixels)	y Focal Length (pixels)	x Radial Distortion (pixels)	y Radial Distortion (pixels)
0.28	$f_x = 3110$	$f_y = 3121$	$RD_x = -0.0265$	$RD_y = -0.0198$

Table 3. Results of the camera calibration for the images from the east side

3.3.3 Flight Path Planning

It's important to note that the real focal length of the visual camera provided by the manufacturer is 4.3 millimeters, which will be utilized for flight planning, particularly in determining camera positions and the image size is 4056×3040 pixels. In order to facilitate flight planning for this camera, various heights have been generated using the GSD equations and the camera's given specifications.

For the purpose of this study and experiment, which focuses on the top and side surface crack detection on bridge decks (level 3 interest), two different flight heights have been chosen for the inspection of the surface of the deck (2 and 3 meters) and three horizontal distances have been chosen for the inspection of the side of the deck (2, 3, and 4 meters) considering GSD calculations from flight path planning section and results of the in-lab camera calibration to achieve less image distortion and reprojection error in order to get more accurate results after data processing for crack detection.

Regarding the bridge inspection, the focus is on the deck of the bridge, and the FAA regulations require the flight to be conducted away from the traffic. Therefore, the flights should be performed off the road near the edge of the deck. Also, it is worth mentioning that the UAV is equipped with LiDAR sensors to avoid obstacles or unexpected objects.

The first bridge is oriented along the southeast to northwest direction, and for optimal lighting conditions, the flight will take place in the morning. It is preferred to capture photos from the east part of the deck to position the camera with the sunlight behind it. This arrangement will improve the image quality by minimizing unwanted reflections. Also, the second bridge is oriented along the east-to-west direction, and for the conditions mentioned above, the photos are captured from the south part of the bridge. Considering the length of the bridges, images are captured at intervals of 10 ft, ensuring a minimum overlap of 50% between consecutive images. A visual representation of the flight plan is depicted in Figure 16 for the first bridge. As shown in this figure, the flight paths for the top and side of the deck are straight lines parallel to the bridge orientation and with the transverse distances from the deck of the bridge as shown in Table 4 and Table 5. The positions of the UAV on the east part of the deck for photo capture are indicated by orange signs. Additionally, black rectangular signs mark the locations of benchmarks for camera calibration, located on the shoulder of the road, which will be discussed later. It is worth mentioning that the flight plan for the second bridge is similar to the first bridge and only the length of the bridge is different. Besides, for the second bridge only one unit of the bridge is inspected to have the same number of images for both bridges to ensure the comparison of the results will be more realistic.

For the data collection, 4 different flight plans are generated for the top surface of the deck for each bridge and 3 flight plans are generated for the side surface of the deck for each bridge. These flight plans maintain the same locations for the UAV positioned along a straight line, but they differ in terms of flight height, camera angles, and transverse distances from the edge of the deck. Table 4 shows the flight plans for the top surface of the bridges and Table 5 shows the flight plans for the side surface of the bridges.

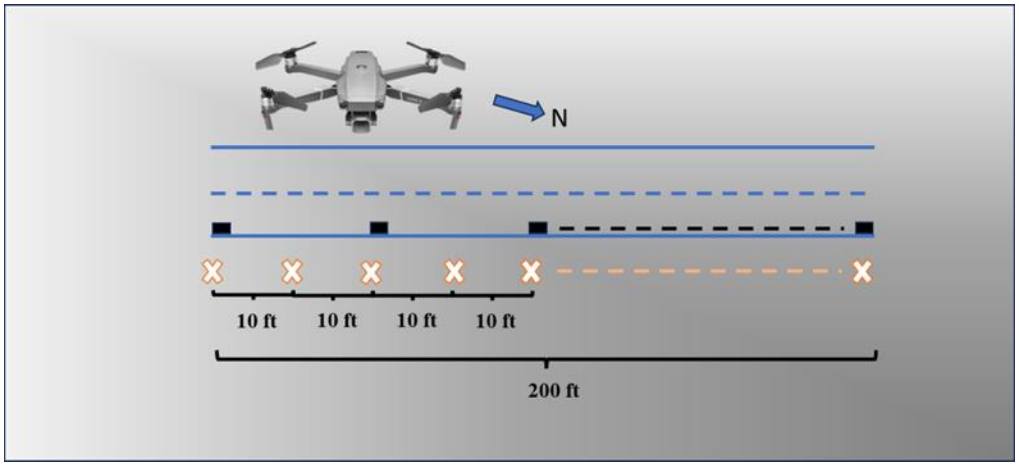


Figure 16. Schematic view of the flight plan for the first bridge (NBI number: 01791)

Flight Number	Flight Height (m)	Transverse Distance (m)	Camera Angles (°)
1	2	1	30, 35
2	2	2	30, 35
3	3	1	30, 35
4	3	2	30, 35

Table 4. Flight plans for the top surface of the bridges

Flight Number	Flight Height (m)	Transverse Distance (m)	Camera Angles (°)
1	0	2	0
2	0	3	0
3	0	4	0

Table 5. Flight plans for the side surface of the bridges

3.3.4 On-site Camera Calibration

On-site camera calibration offers the advantage of using the same camera parameters and distortion values for the collected bridge inspection data. To achieve on-site camera calibration, benchmarks are attached to specific parts of the bridge that will be covered in the inspection images. Then, the camera calibration can be performed directly during the bridge inspection simply by capturing the photos. Also, the study is not only limited to the top of the deck, and a calibration process is tested for the side of the deck for both bridges. Although the results of camera calibration for the top of the deck can be used for the side of the deck, a separate calibration for the side surface would give better and more accurate results to use for the inspection of the side of the deck. The reason for this difference in the calibration of the top and side of the deck is generated from the different photographic situations and parameters. As an example, the camera angle during the data collection for the top of the deck is different from the angle of the camera for the side of the deck which is zero. Another example for this reason is the lighting and reflection of the light which is different for the top of the deck and side of the deck.

In Figure 17, a collection of images is displayed from the on-site camera calibration process conducted on the first bridge for the top surface and side surface of the deck for this study's experiment. As indicated in the flight path schematic, a total of 13 benchmarks are strategically placed on the bridge deck, positioned near the edge of the bridge at regular intervals of approximately 20 feet.

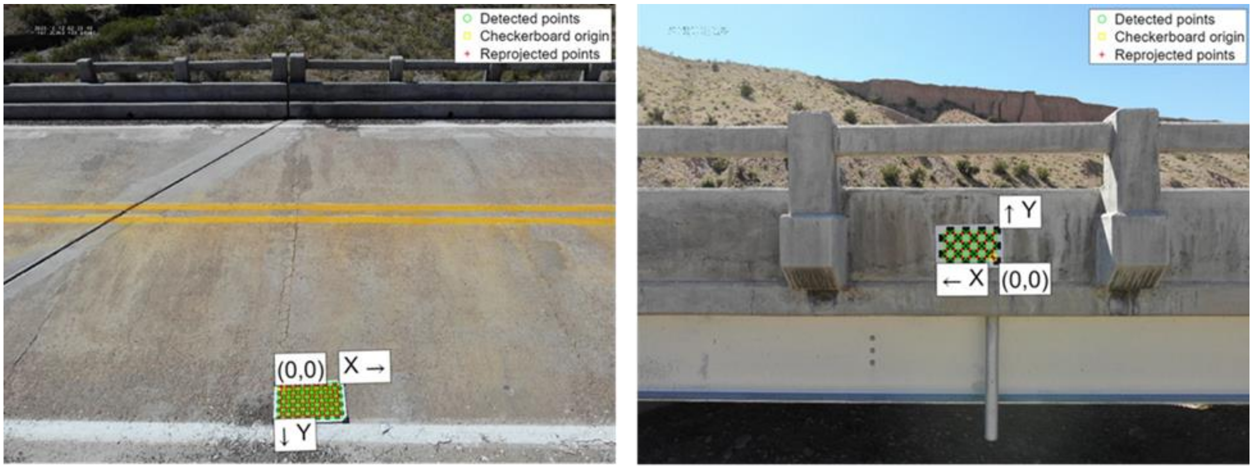


Figure 17. On-site camera calibration examples for the first bridge (NBI bridge number: 01791)

A total of 104 images from the bridge inspection dataset are utilized from each bridge's top surface for camera calibration and the estimation of camera parameters and reprojection errors. The specific details of these images, such as flight height, transverse distance from the edge of the deck, and camera angles have been presented in Table 4. For the side of the deck, 6 benchmarks have been used and a total of 18 images (from 3 flights) are captured for each bridge. The details of the flight are shown in Table 5.

In Table 6, the overall mean reprojection error, as well as the focal length in the x and y directions, and radial distortion in the x and y directions for the top surface of the first bridge are displayed for these 104 images undergoing on-site calibration. Also, in Table 7, the mentioned results are displayed for the side surface of the first bridge.

Overall Mean Error (pixels)	x Focal Length (pixels)	y Focal Length (pixels)	x Radial Distortion (pixels)	y Radial Distortion (pixels)
0.25	$f_x = 3141$	$f_y = 2965$	$RD_x = 0.0161$	$RD_y = -0.0298$

Table 6. Results of the camera calibration for the images from on-site calibration for the top surface of the first bridge (NBI bridge number: 01791)

Overall Mean Error (pixels)	x Focal Length (pixels)	y Focal Length (pixels)	x Radial Distortion (pixels)	y Radial Distortion (pixels)
0.20	$f_x = 4513$	$f_y = 4367$	$RD_x = 0.0746$	$RD_y = -1.3192$

Table 7. Results of the camera calibration for the images from on-site calibration for the side surface of the first bridge (NBI bridge number: 01791)

Figure 18 visually represents the histogram of the reprojection mean errors of the images for the top of the first bridge and Figure 19 represents the same histogram for the side surface of the bridge. The first bin in each histogram shows the number of images with a mean error lower than the overall mean error. This graphical representation aids in understanding the distribution and dispersion of the reprojection errors within the dataset. Comparing the mean error of each image with the overall mean error reveals that 55% of the images have a lower mean error than the overall mean error for the top surface of the first bridge and 56% of the images for the side surface have a lower mean error than the overall mean error. This analysis provides insights into the variations in calibration accuracy among the different images. Also, as mentioned before, these low reprojection errors are achieved by considering the results of the in-lab calibration, and the main advantage of this approach will be discussed in validation.

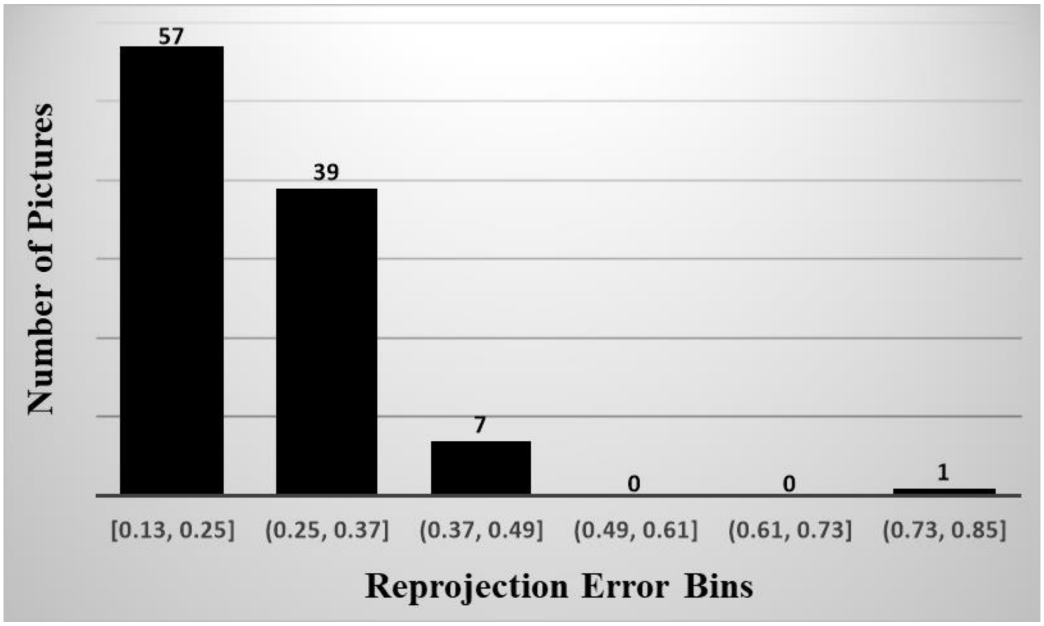


Figure 18. Histogram for the distribution of the images according to their mean reprojection errors for the top surface of the first bridge (NBI bridge number: 01791)

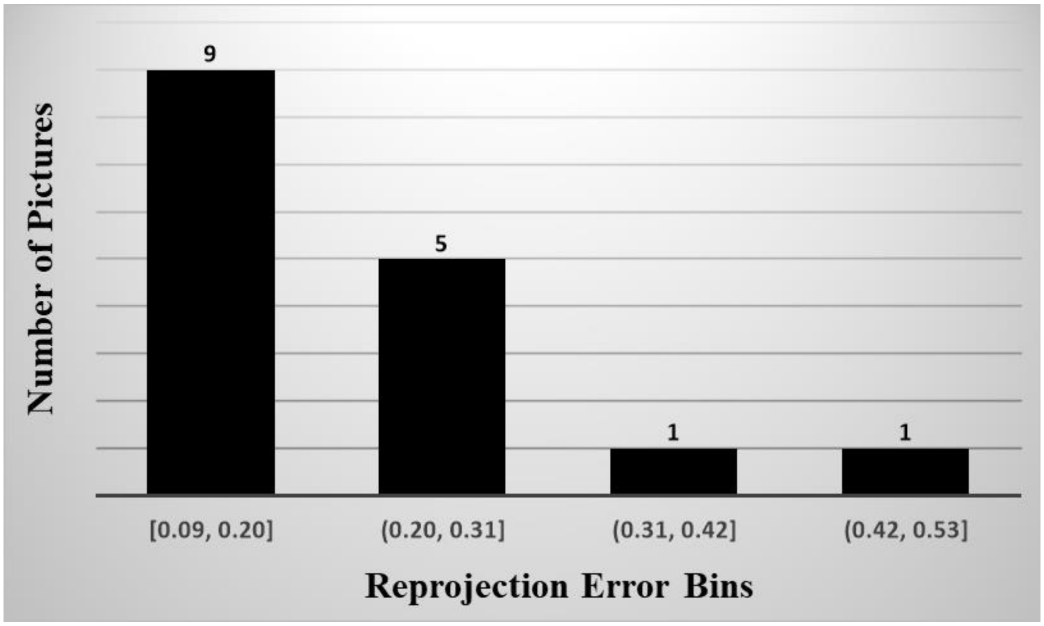


Figure 19. Histogram for the distribution of the images according to their mean reprojection errors for the side surface of the first bridge (NBI bridge number: 01791)

In Figure 20, a collection of images is displayed from the on-site camera calibration process conducted on the second bridge for the top and side surfaces of the deck for this study's experiment.

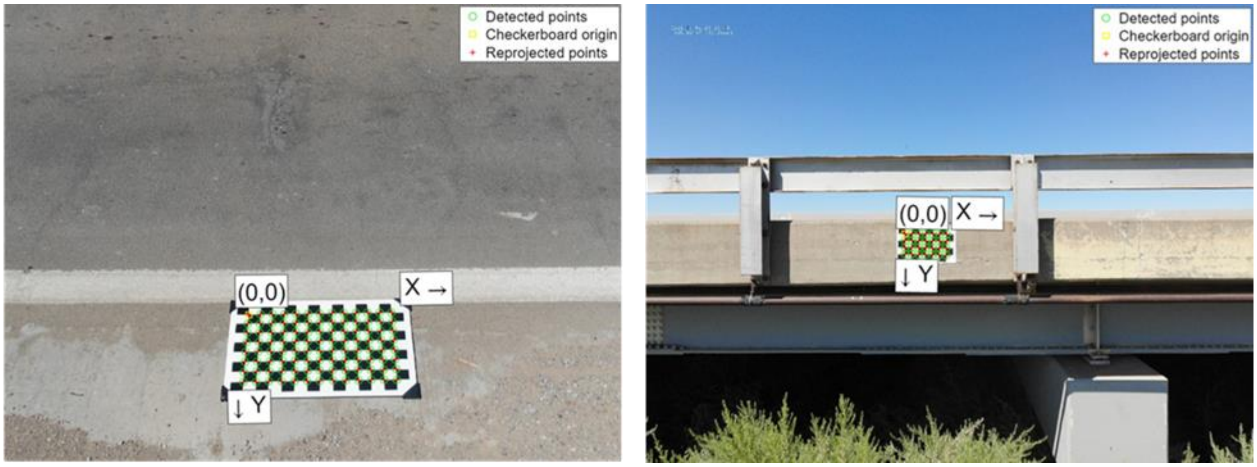


Figure 20. On-site camera calibration examples for the second bridge (NBI bridge number: 06255)

In Tables 8 and 9, the overall mean reprojection error, as well as the focal length in the x and y directions, and radial distortion in the x and y directions for the top and side surfaces of the second bridge are displayed for undergoing on-site calibration.

Overall Mean Error (pixels)	**x Focal Length (pixels)**	**y Focal Length (pixels)**	**x Radial Distortion (pixels)**	**y Radial Distortion (pixels)**
0.29	$f_x = 3165$	$f_y = 3002$	$RD_x = 0.0277$	$RD_y = -0.0519$

Table 8. Results of the camera calibration for the images from on-site calibration for the top surface of the second bridge (NBI bridge number: 06255)

Overall Mean Error (pixels)	**x Focal Length (pixels)**	**y Focal Length (pixels)**	**x Radial Distortion (pixels)**	**y Radial Distortion (pixels)**
0.14	$f_x = 6627$	$f_y = 6585$	$RD_x = 0.1661$	$RD_y = -6.9088$

Table 9. Results of the camera calibration for the images from on-site calibration for the side surface of the second bridge (NBI bridge number: 06255)

Figure 21 visually represents the histogram of the reprojection mean errors of the images for the top of the second bridge and Figure 22 represents the same histogram for the side surface of the bridge. The first bin in each histogram shows the number of images with a mean error lower than the overall mean error. This graphical representation aids in understanding the distribution and dispersion of the projection errors within the dataset. Comparing the mean error of each image with the overall mean error reveals that 68% of the images have a lower mean error than the overall mean error for the top surface of the bridge and 69% of the images for the side surface of the bridge have a lower mean error than the overall mean error. Comparing the results for the side and top of the deck for both bridges indicates that the percentage of images with lower mean error than the overall mean error is almost the same for the side and top of each bridge (only a 1% difference for the side and top of each bridge).

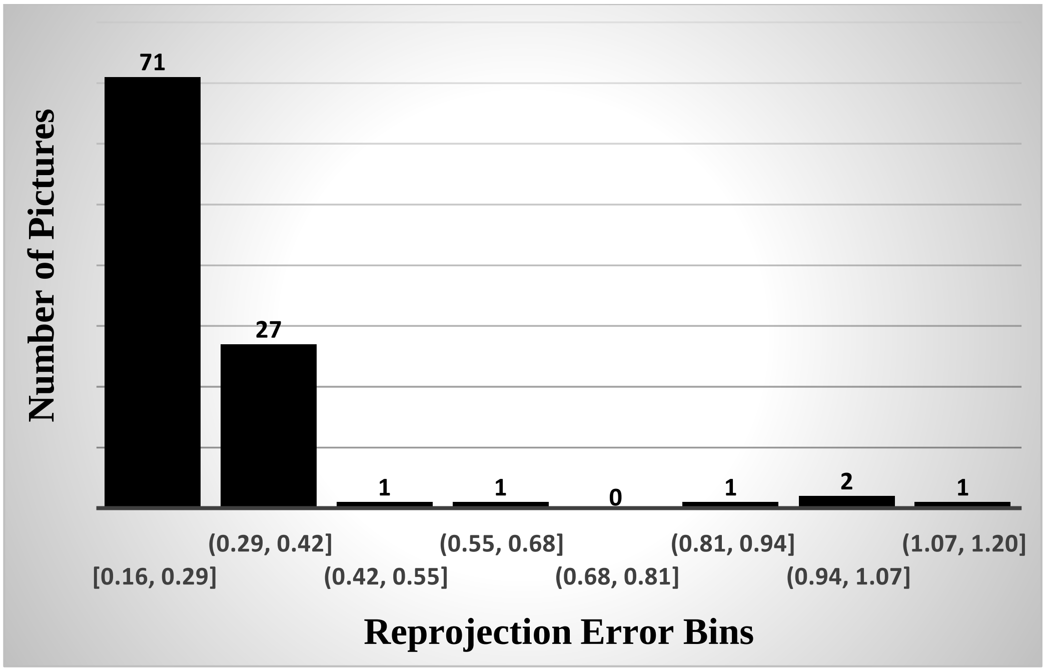


Figure 21. Histogram for the distribution of the images according to their mean reprojection errors for the top surface of the second bridge (NBI bridge number: 06255).

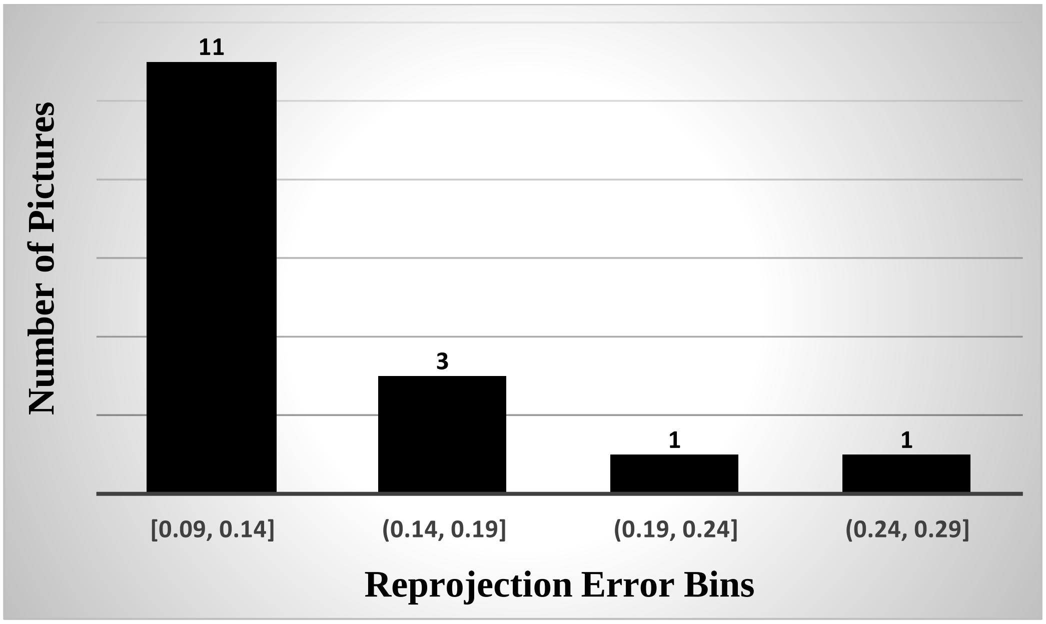


Figure 22. Histogram for the distribution of the images according to their mean reprojection errors for the side surface of the second bridge (NBI bridge number: 06255).

It is worth mentioning that among all the flight plans mentioned before, the minimum reprojection mean error for both bridges is achieved in the flight with 3m height, 1m transversal distance, and a camera angle of 30 for the top surface and transverse distance of 3m for the side surface, which aligns with the results of in-lab calibration. This result would be beneficial for the future inspection and flight plans.

3.3.5 Validation

To evaluate the feasibility of the proposed method, validations have been conducted for case studies. Two cracks from the bridge deck side surface were chosen for validation investigations. The widths of the cracks were measured from both the raw and calibrated images and subsequently compared with the ground truth measured by the inspector.

The width of the first crack is 0.51 mm, and the width of the second crack is 1.53 mm. The cracks are shown in Figure 23 and Figure 24 respectively. For crack detection in this study, the GSD is known from flight height and moreover, the accurate GSD is known from the calibration results which is another benefit of on-site camera calibration. By knowing the GSD, each pixel represents a known number of millimeters. The crack width is detected from raw data by image processing techniques, then the crack width is detected from the images after correction by using the calibration and reprojection results. For this purpose, first, the original image is converted to grayscale and the contrast of the grayscale image is enhanced. Then, adaptive thresholding is applied to create a binary image and finally, Gaussian smoothing is applied to the binary image to reduce noise and create smoother edges, facilitating more accurate crack detection.

The crack width detection results considering the flight plans for the side of the deck, are shown in Table 10 for the first crack and in Table 11 for the second crack. Finally, the percentage of the detection accuracy is compared between these two results. For both cracks, the results indicate that the result accuracy improvement is higher for the higher flight heights (more distance from the object). Also, the detected crack width is closer to the measured width in flight 2 which has a 3m distance from the object and it completely aligns with the results of camera calibration where the 3m flight had the minimum reprojection error. Moreover, it is worth mentioning that for small dimension cracks the human measurement has more error, and using this kind of accurate crack detection techniques leads to better assessments of the bridge condition.

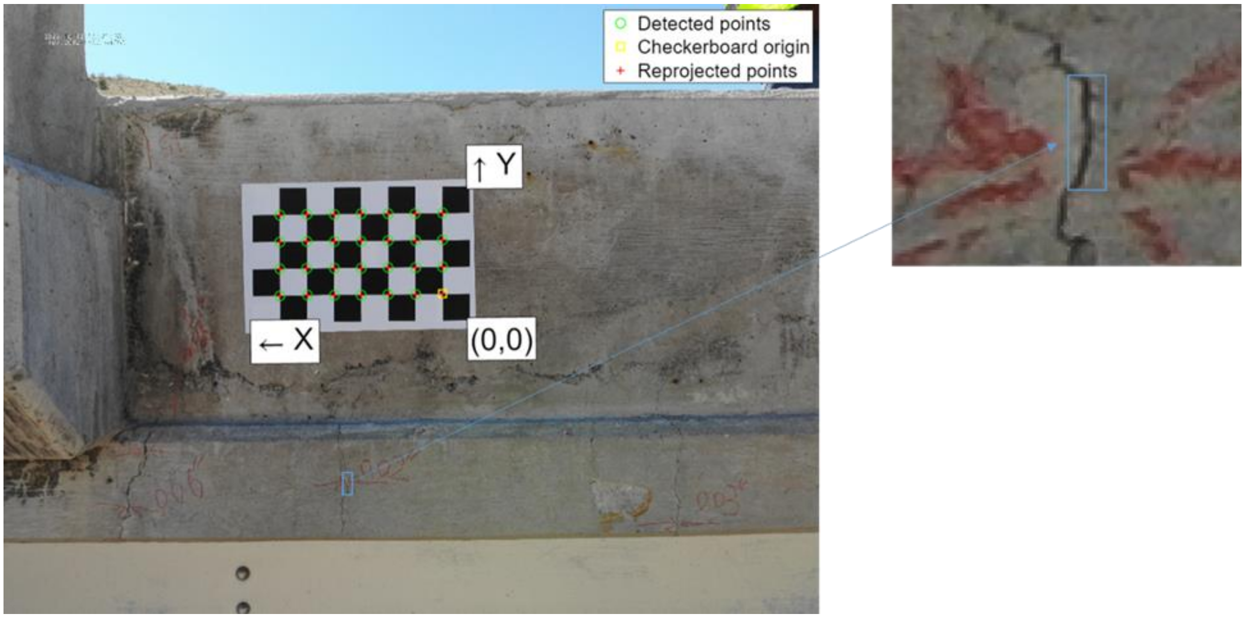


Figure 23. The first crack with a width of 0.51 mm on the side of the first bridge

Flight	Distance (m)	Measured Width (mm)	Detected Width from Raw Images (mm)	Detected Width After Correction (mm)	Result Accuracy Improvement (%)
1	2	0.51	0.71	0.66	9.80
2	3	0.51	0.71	0.62	17.65
3	4	0.51	1	0.9	19.61

Table 10. Results of the crack detection for the first crack from raw and corrected images

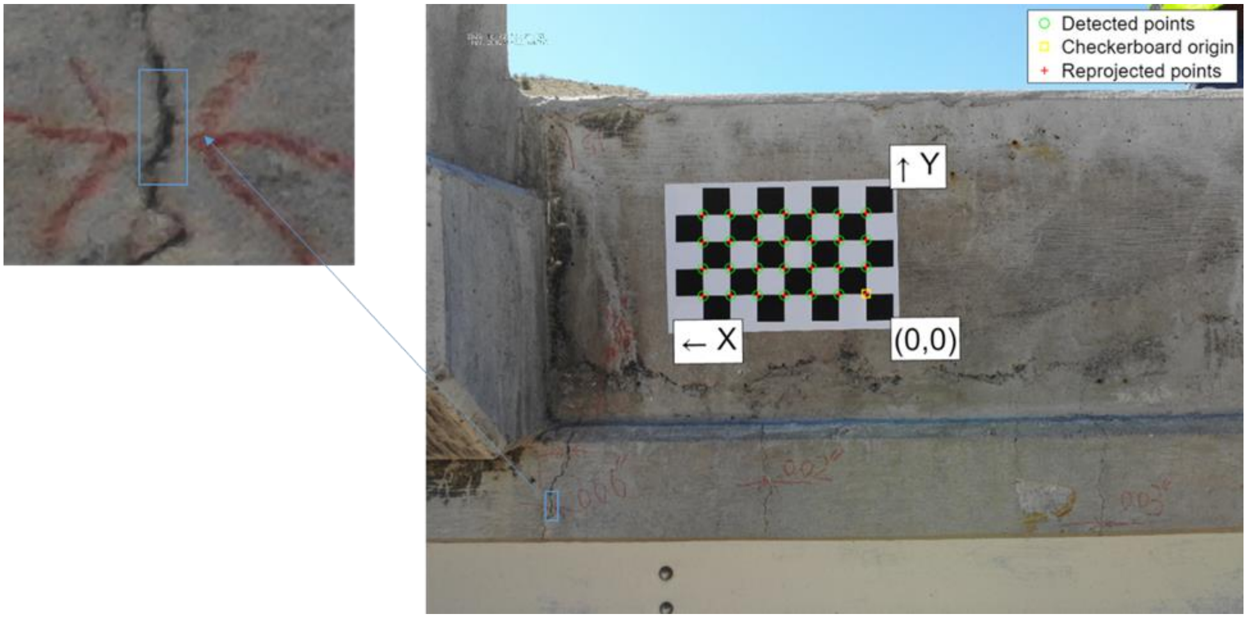


Figure 24. The second crack with a width of 1.53 mm on the side of the first bridge

Flight	Distance (m)	Measured Width (mm)	Detected Width from Raw Images (mm)	Detected Width After Correction (mm)	Result Accuracy Improvement (%)
1	2	1.53	1.78	1.67	7.19
2	3	1.53	1.77	1.55	14.38
3	4	1.53	2.13	1.80	21.57

Table 11. Results of the crack detection for the second crack from raw and corrected images

The validation for the case studies indicates an improvement in result accuracy from 7.19% to 21.57% and it shows that the improvement is higher in longer distances, which means that inspection can be done even in longer distances where more area is covered by each image and consequently less camera points are needed which leads to a shorter flight time.

TASK 4: Software Development for Flight Path Optimization

Among different UAV types, multi-rotor UAVs are flexible and can hover and vertically lift which results in higher maneuverability and makes them a suitable candidate to collect in-detailed data but on the other hand, they suffer from short endurance time [22]. To fully harness the time efficiency potential of the UAVs for inspection purposes, and to overcome the above-mentioned challenge of the short endurance time for multi-rotor UAVs, a well-planned flight path is crucial. UAV path planning involves addressing the optimal path planning for UAVs. The main goal of path optimization is to minimize energy consumption while ensuring the successful completion of the UAV's mission.

Recently, some studies have been conducted for UAV flight path optimization by using meta-heuristic algorithms. In contrast to typical optimization techniques, meta-heuristic algorithms' primary concept is based on simulating natural processes and does not require extensive mathematical computations during the optimization process [23], [24]. Over the past two decades, a multitude of meta-heuristic algorithms have been developed to aid in solving different optimization problems that were previously difficult or inconceivable to solve using mathematical programming algorithms. Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimizer (PSO), Charged System Search (CSS), Colliding Bodies Optimization (CBO), and Shuffled Shepherd Optimization Algorithm (SSOA) are some of the meta-heuristic algorithms [25], [26], [27], [28], [29], [30]. The majority of these algorithms are easy to implement and provide optimal/near-optimal solutions in acceptable computation times even in complex search spaces [31].

A heuristic evolutionary algorithm was presented by Fu et al. for the purpose of optimizing the flight path for a single path connecting two points. For 2D areas, they took into account the costs associated with path smoothness, length, and safety [32]. Yu et al. proposed a novel hybrid PSO algorithm for 3D complex environments [33]. In another 3D flight path planning in urban environments, Rienecker et al. used the extended A-Star-Algorithm for path optimization considering the wind field [34]. Phung et al. conducted research for flight path planning for surface inspection by using a discrete PSO optimization algorithm and tested the proposed method on an office building and a bridge [35]. For flight route generation between two points with obstacles between them, Zhang et al. proposed a new path-planning algorithm for UAVs based on a version of the White Shark Optimization (WSO) and defined the flight length, height, and smoothness as the constraints of the problem [36]. In another study for flight path optimization between two points, Bai et al. proposed an algorithm based on the enhanced Dynamic Window Algorithms (DWA) to achieve global path optimization by taking safety and motion constraints into consideration, and according to their findings, they achieved a better flight length even when the obstacles are unknown [37]. Also, in a recent study, Souto et al. conducted a method to lower the energy usage of drones by using the Q-Learning algorithm to analyze potential courses of action that the UAVs could follow in response to random urban obstacles distributed in the scenario and in response to wind speed [38]. The improvements, challenges, and future trends of UAV flight path planning and optimization have been studied and discussed in several studies [39], [40], [41].

One of the main challenges of coverage path planning is that the search space is infinite. Although this challenge was addressed in our previous study, the computation time was still high. Considering this challenge for software development, the optimization engine has been modified so that the area will be divided into a large set of points, and the optimum waypoints will be chosen from this set of points. For this purpose, several meta-heuristic algorithms are selected and used to find the optimal path on a continuous area, among these algorithms, PSO achieved better results and has been used for the software's optimization engine where the area is not considered continuous and is divided into a set of points. By finding the optimum flight path, the challenge of the optimum flight path planning for area coverage and the challenge of the short battery lifetime of the UAVs will be addressed simultaneously due to the energy-saving resulting from the optimal path.

Key features of the software are mentioned below and will be discussed in the next section:

1. Optimization Engine
2. Constraint Handling
3. Flight Path Conversion
4. User Interface

4.1 Optimization Engine

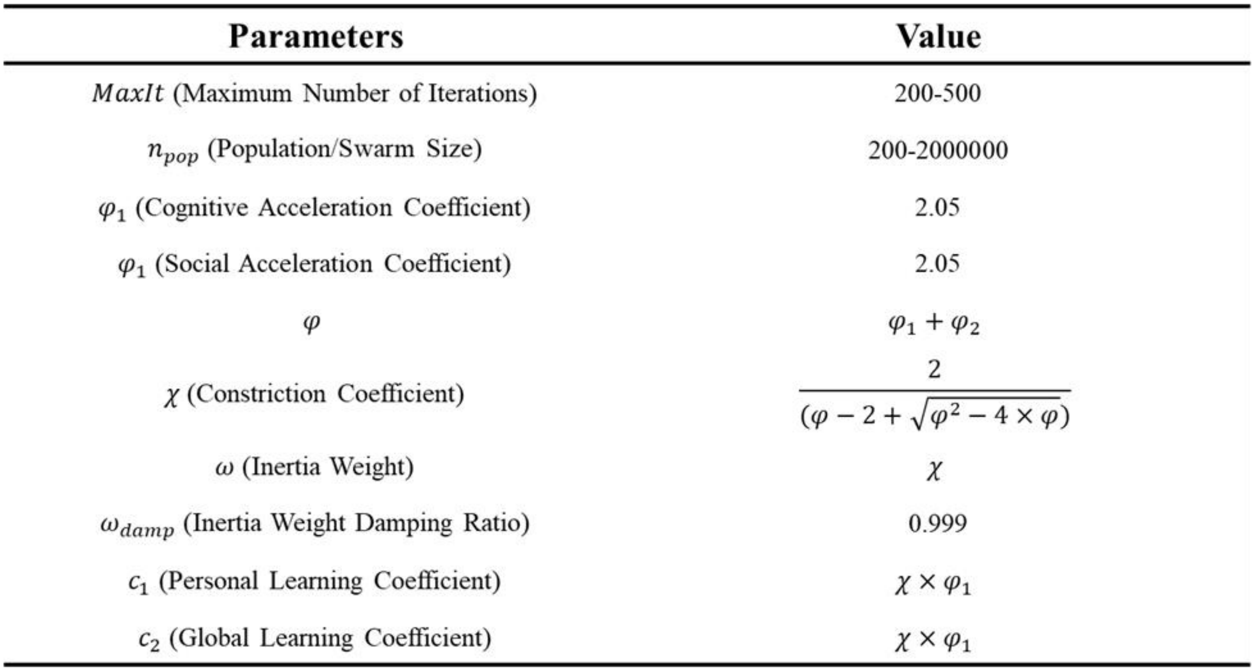
The flight path comprises a set of camera positions from which images will be captured. These camera positions are determined by their horizontal and vertical distance from the object, as well as the angle of the camera. In this section, the crucial aspects of the optimization engine for the software will be explored shortly.

The main component of the optimization engine is the meta-heuristic algorithm. The main idea of meta-heuristic algorithms is based on the simulation of natural phenomena and, unlike the traditional optimization methods, do not need to perform heavy mathematical calculations in the optimization process. Numerous meta-heuristic algorithms have been created within the last two decades to assist in fathoming optimization problems that were already troublesome or inconceivable to solve utilizing mathematical programming algorithms. Most of these algorithms are simple to implement and present (near) optimal solutions in acceptable computation times even in complex search spaces [15].

In this study, several meta-heuristic algorithms such as Particle Swarm Optimizer (PSO), Genetic Algorithm (GA), and Shuffled Shepherd Optimization Algorithm (SSOA) have been used to not only find the best flight path among these algorithms but also, to compare the results of the algorithms and their performance. After comparing the results for different missions with different constraints, PSO achieved better results compared to other algorithms. Therefore, for the optimization engine of the software, PSO will be used.

Particle Swarm Optimization (PSO) is an optimization technique inspired by the social behavior of birds flocking or fish schooling. It operates by having a population (or swarm) of candidate solutions (particles) that move through the solution space to find the optimum. Each particle adjusts its position based on its own experience and the experience of neighboring particles, converging towards the best solution over iterations. Key parameters of PSO include the number of particles, the self-learning coefficient (which influences how much a particle is influenced by its own best-known position), the global-learning coefficient (which influences how much a particle is influenced by the best-known positions of its neighbors), and the inertia weight (which controls the impact of the previous velocity on the current velocity) [27].

By tuning these above-mentioned parameters appropriately, this study aims to leverage the strengths of PSO to effectively solve flight path optimization problems. For this purpose, Clerc and Kennedy's constriction coefficient method [42] has been used in this study to tune the parameters to control the velocity update of particles for PSO. The selected coefficients and other parameters for PSO are shown in Figure 25.



Parameters	Value
$MaxIt$ (Maximum Number of Iterations)	200-500
n_{pop} (Population/Swarm Size)	200-2000000
φ_1 (Cognitive Acceleration Coefficient)	2.05
φ_1 (Social Acceleration Coefficient)	2.05
φ	$\varphi_1 + \varphi_2$
χ (Constriction Coefficient)	$\frac{2}{(\varphi - 2 + \sqrt{\varphi^2 - 4 \times \varphi})}$
ω (Inertia Weight)	χ
ω_{damp} (Inertia Weight Damping Ratio)	0.999
c_1 (Personal Learning Coefficient)	$\chi \times \varphi_1$
c_2 (Global Learning Coefficient)	$\chi \times \varphi_1$

Fig 25. Selected parameters for PSO

Similar to most of the classical path optimization problems, the energy consumption of UAV flying is represented by the flight path length/time. The length of the path refers to the path from the starting point to the end point of the flight. The flight time has a direct relation to the path's length at constant speed, which in turn influences the UAV platform's energy consumption. Considering the length of the path as the objective function to minimize, the cost function is defined as below:

$$Cost_l(X) = L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \qquad (10)$$

Where $Cost_l$ represents the length cost of the path in meters, L is the total length of the path in meters, and n is the number of points traversed by the path (waypoints) where the images are captured. x_i, y_i are the coordinates of the ith traversal point in meters.

By having the $Cost_l$, the consumed energy during a flight can be found by the below equation:

$$Cost_{El} = E_l{}' \times Cost_l \qquad (11)$$

Where $Cost_{El}$ is the consumed energy (J) based on the flight length and $E_l{}'$ is the energy consumption rate based on the flight length (J/m) and it can be found either from the drone's specifications or by experiments.

Under constant speed conditions, flight time is directly proportional to flight length, thus the consumed energy can be calculated by the following equation:

$$Cost_{Et} = E_t{}' \times t \qquad (12)$$

Where $Cost_{Et}$ is the consumed energy (J) based on the flight time, t is the flight time (s), V_{avg} is the average flight speed (m/s), and $E_t{}'$ is the energy consumption rate based on the time (J/s).

Search space for the variables will consist of points that are generated specifically for the mission area and the number of the points is based on the dimensions of the area. The required number of waypoints for the flight path will be chosen from the initial points of the search space.

4.2 Constraint Handling

For flight path optimization there are three constraints that should be checked: area coverage, minimum overlap, and obstacle avoidance. To ensure that the whole inspection area is covered by the images taken from the traversal points, the area will be meshed and divided into small triangle areas that contain a set of m points, and each point has x and y coordinates ($M = \{x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m\}$). To satisfy the coverage constraints each of these m points should be covered at least in one of the images. Image width and height (D_W, D_H) are known from equations 3 and 4, and by having this width and height, each point of the M will be checked to be covered at least in one of these image footprints. Figure 26 shows an example of meshing for an area of 10×10 $(100m^2)$.

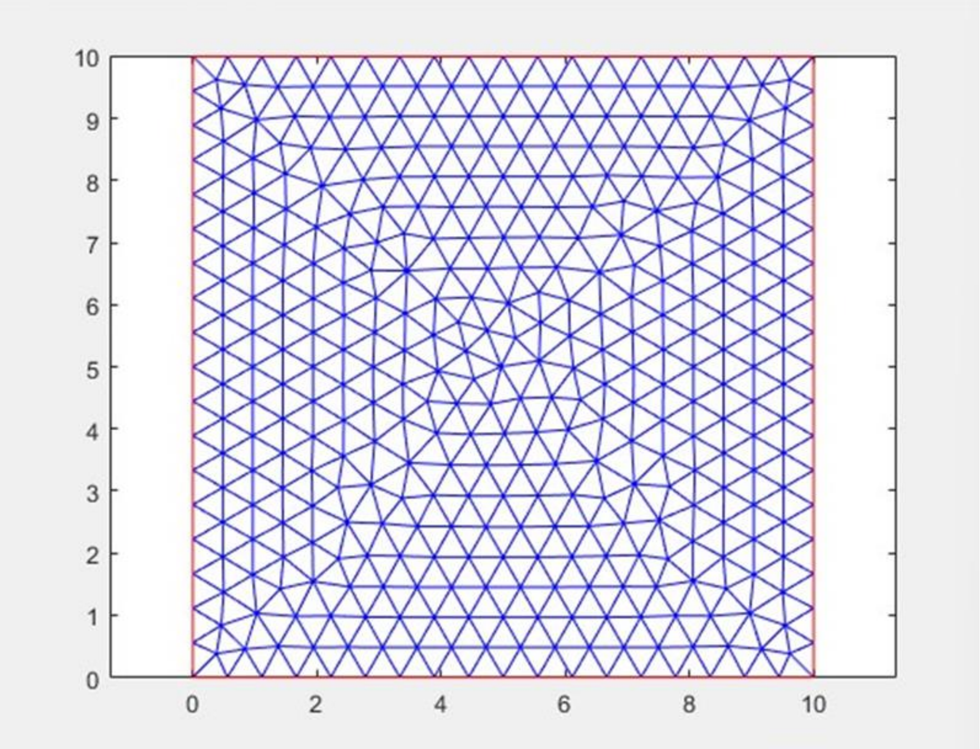


Fig 26. A meshing example for an area of $10 * 10$

To ensure at least 50% overlap between the consecutive images, an adjacency radius (r) is defined. This radius is the maximum allowable distance between two adjacent points which means that the allowable location for the traversal point $i + 1$ is inside or on a circle where the center of the circle is the point i and the radius of the circle is r. Figure 27 shows this overlap constraint between two consecutive points. As illustrated in the figure, the blue point is the ith point which has an image footprint equal to the blue rectangle and the possible area to select the $i + 1$th point is inside or on the circle.

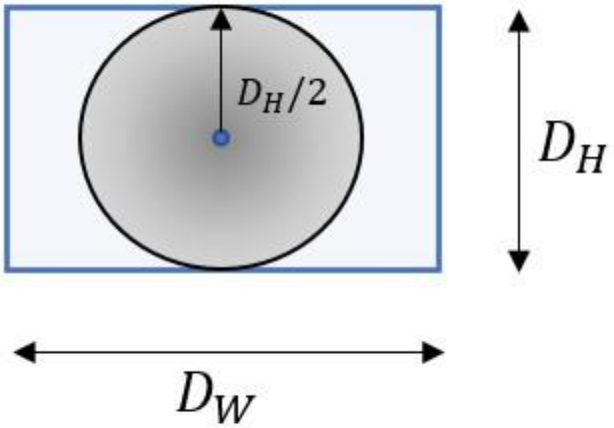


Figure 27. Overlap constraint between two traversal points

In this study, to satisfy the overlap constraint, the mentioned radius is set to $D_H/2$. This adjustment ensures at least 50% overlap between consecutive points. The minimum overlap occurs if the point $i+1$ is located exactly at a vertical distance of $D_H/2$ form point i as it is shown in Figure 28. Consequently, if the point $i+1$ is located in another position to the point i or the distance is lower than $D_H/2$, the overlap would be more than 50%. Some examples of this condition are shown in Figure 28. It is worth noting that this 50% overlap is the minimum consideration but in the software, users will be able to adjust the overlap between 50% to 90%.

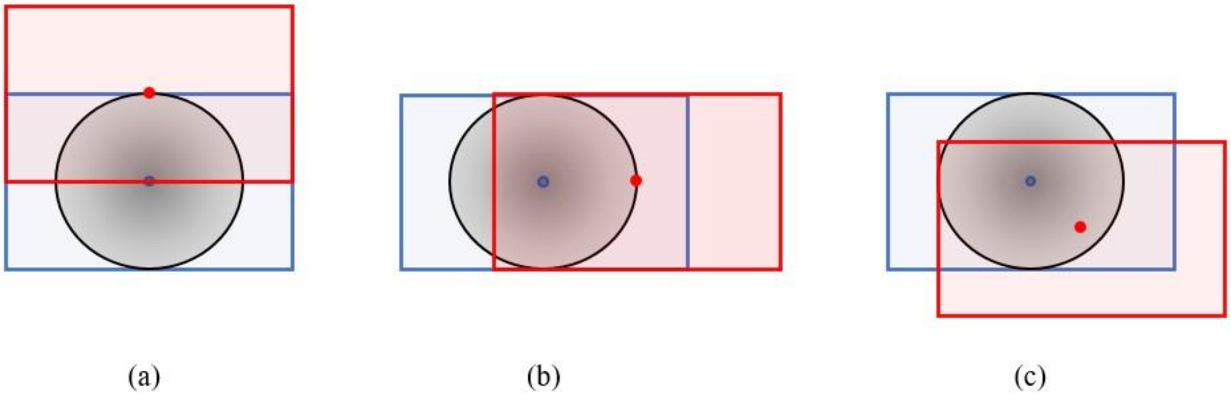


Figure 28. Overlap conditions: (a) overlap = 50% (b), (c) overlap > 50%

The last constraint is to make sure that obstacles have been avoided in the flight path to ensure the safety of the flight. For this purpose, obstacles are represented as rectangles and the number of obstacles is assumed to be *K*. By knowing the coordinates of the bottom-left corner of the obstacles, which are x_k, y_k respectively, and by knowing the length and width of the rectangles, which are l_k, w_k respectively, the intersection between each rectangle and the flight path between two waypoints could be checked. By finding the potential intersection points and their coordinates $(I_{1x}, I_{1y}, I_{2x}, I_{2y})$, the length of the flight path inside the obstacles can be determined using the following equation:

$$L_O = \sqrt{(I_{2x} - I_{1x})^2 + \left(I_{2y} - I_{1y}\right)^2} \quad (13)$$

Where L_O is the length of the flight path inside the obstacle in meters.

Finally, the optimum flight path problem is formulated by the following expression:

Find an integer design vector $\{X\} = \{x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n\}$, where x_i, y_i are the x and y coordinates of the ith traversal point of the flight path. Hence, the design problem can be expressed as:

Minimize Equation 10

Subjected to

$\boldsymbol{g_1}$: $For\ j = 1\ to\ m$:

$For\ i = 1\ to\ n$:

$$\left(x_i - x_j\right) - D_W/2 \leq 0 \quad \& \quad \left(y_i - y_j\right) - D_H/2 \leq 0 \tag{14}$$

$\boldsymbol{g_2}$: $For\ i = 1\ to\ n - 1$:

$$\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} - D_H/2 \leq 0 \tag{15}$$

$\boldsymbol{g_3}$: $For\ i = 1\ to\ n - 1$:

$For\ k = 1\ to\ K$:

$$L_O = 0 \tag{16}$$

Where m is the number of the generated nodes for meshing to check the coverage, n is the number of the nodes of the flight path (traversal points), x_i, y_i are the coordinates of the ith traversal point, and x_j, y_j are the coordinates of the jth generated points of the coverage meshing. Also, g_1, g_2, g_3 ae the constraints of the problem.

The steps and the flowchart of the process are shown in Figure 29. It is worth mentioning that termination criteria is the number of iterations for meta-heuristic algorithms.

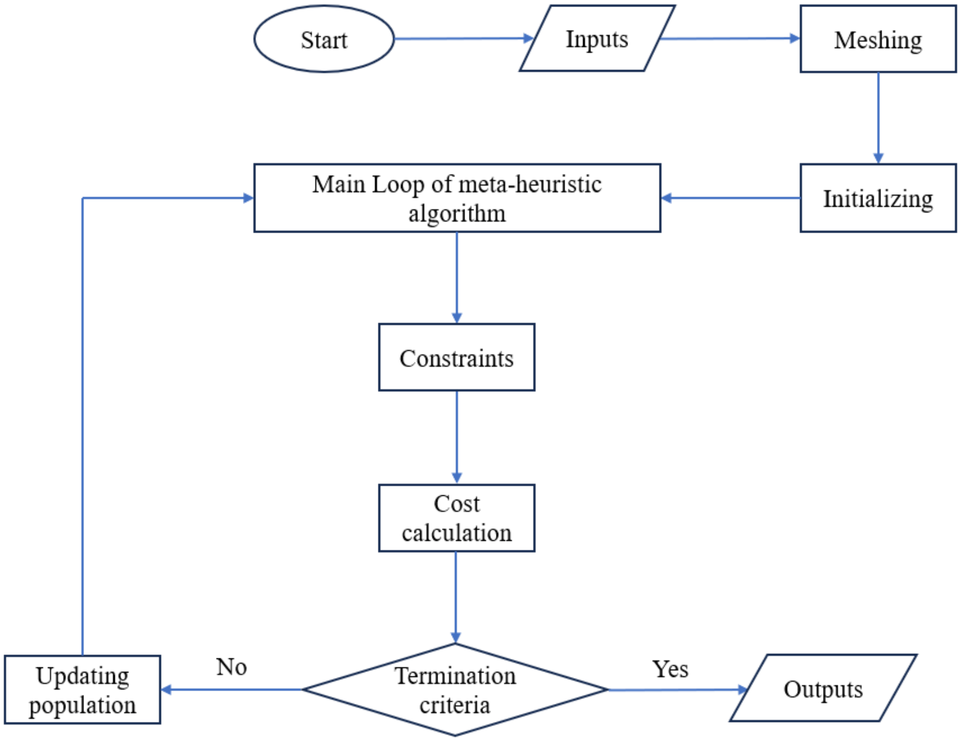


Figure 29. Flowchart of the flight path optimization process

4.3 Flight Path Conversion

With the adjustments that have been made to the code, users would be able to define any polygon shapes for the mission area. Once the optimum flight path is generated, it should be projected on the real mission area defined. For this purpose, the generated flight path, which is in decimal units (m) in the x and y directions, will be converted to the Geographic Coordinate System (GCS) so that the user can export the flight path of the mission area and import it into UAV.

4.4 Flight Path Planning Web Application

This section outlines the key features and progress of the flight planning software being developed in collaboration with NMSU researchers. The software program, which will be developed as a web application, aims to optimize drone flight paths by addressing key constraints and delivering a streamlined, intuitive, and user-friendly experience. It achieves this through the implementation of a particle swarm optimization algorithm. Additionally, the software program will be developed using Python, MATLAB, HTML, CSS, and JavaScript within the Flask framework.

Below is a comprehensive list of the key constraints that the web application will address, along with a detailed explanation of its features. This overview outlines how each constraint is managed and the corresponding functionalities that enhance the software's capabilities in optimizing drone flight paths.

Mission Area or Area of Interest (AOI): The web-based interface allows users to select an AOI. Based on this selection and the following parameters, the software automatically generates a flight path tailored to the AOI. Instead of a traditional grid system, which can be inefficient, the software employs a single, continuous flight line strategy. This method simplifies flight planning and optimizes both time and battery usage.

Battery Life: The web application considers battery life as a fundamental constraint in flight, aiming to maximize image collection within the battery's operational limits. The software calculates the maximum allowable flight time based on the drone's battery capacity, while reserving sufficient battery power to prevent depletion. It enhances the drone's practical usability and ensures reliable operation throughout a mission.

Flight Speed: It influences the stability of the drone, the quality of captured data, and the efficiency of the flight path. Properly planning and adjusting flight speed ensures smoother operation, optimal image or data collection, and helps in avoiding collisions or interference with other objects. Users will be able input the appropriate values for flight speed.

Camera Specifications: Users will be able to input the camera's focal length in millimeters, the sensor height and width in millimeters, and the image height and width in pixels.

Flight Height: This height information is crucial in drone flight path planning. It affects factors such as the quality of the captured images, the coverage area, the ground sampling distance (GSD), and the safety of the flight. Users will be able to input the appropriate values for flight height.

Overlap Considerations: To ensure comprehensive coverage and high-quality data collection, the software manages both forward and side overlap of collected aerial imagery. Forward overlap pertains to the extent of overlap between consecutive flight paths, while side overlap addresses the overlap between parallel flight lines or parallel imagery frames. By assigning values (50% to 90%) to the desired overlaps, the software ensures complete area coverage and enhances data reliability. This feature supports the creation of co-registered orthophotos and digital surface models (DSMs), allowing for the detection of finer-scale bridge deck distresses.

Waypoint Optimization: The software is designed to minimize the amount of waypoints required for a given flight mission. Advanced algorithms, including metaheuristic algorithms, are employed to generate efficient flight paths, reducing waypoint management complexity. This optimization simplifies user interaction and reduces the computational load on the drone's navigation system.

Ground Sampling Distance (GSD): The software ensures that the spatial resolution of captured data meets user-specified ground sampling distance (GSD) requirements by calculating and adjusting flight paths accordingly. This guarantees that the imaging mission fulfills end-user specifications and results in high-quality data collection. Additionally, this tailored approach not only fulfills user requirements but also enhances the accuracy and usefulness of the collected data for various applications.

Obstacle Avoidance: To enhance flight safety, the software integrates pre-defined obstacle avoidance algorithms within the user interface. This feature allows users to input and define specific obstacles, which the algorithm then uses to dynamically adjust flight paths and prevent collisions. By incorporating these user-defined obstacles, the software ensures safe and reliable operations throughout the mission.

For the outputs of the web application, users will have access to the following items:

- Visualized Flight Path: an example of the visualization is shown in Figure 30.

- Waypoints: displayed as individual points on a background image, including their coordinates in either a geographic or projected coordinate system.
- GSD: shown in mm/pixel, based on the flight height and overlap specified by the user.
- Flight Path Length: indicated in meters (m).
- Approximate flight time: provided in seconds (s).

Please note that as a web application, the software will be accessible only to authorized users. After the optimized flight path is generated, users will be notified and provided with a link to download the flight path file, which they can then upload to the drone.

The sprint planning for the software development is as follows:

- Sprint 1: Requirements gathering and initial design.
- Sprint 2: Basic optimization engine implementation.
- Sprint 3: UI design and integration with optimization engine.
- Sprint 4: Testing and validation.
- Sprint 5: Improvements.

The first two sprints are achieved, and the project team is working on the third sprint. More specifically, the UNM Team has begun software development in close collaboration with the NMSU team. This collaborative effort highlights the co-development approach, involving joint planning, design, and execution phases to ensure that the development aligns with our shared objectives and requirements. As of the reporting period, we have undertaken several key activities, including:

Algorithm Development: We have participated in the development of algorithms that support the constraints of battery life, overlap, GSD, collision avoidance, and waypoint optimization. This collaborative effort ensures that the software meets the specified requirements effectively. Since the project's inception, the UNM Team has conducted four meetings with the NMSU team to maintain active engagement in the algorithm development process. These meetings have been essential for coordinating our efforts, discussing progress, addressing challenges, and refining our approach.

Bounding Box System: In our approach to algorithm development of the AOI selection, we have decided to assign no value (null) to the cell centers of a minimum grid bounding box system that encloses the entire AOI. This decision aligns with our goal of simplifying the flight path planning process and avoiding unnecessary complexity. The bounding box system will also ensure that either a projected coordinate system or a geographic coordinate system is assigned to the waypoints within the AOI. Additionally, it ensures that the AOI properly overlaps with the bridge object on the ground.

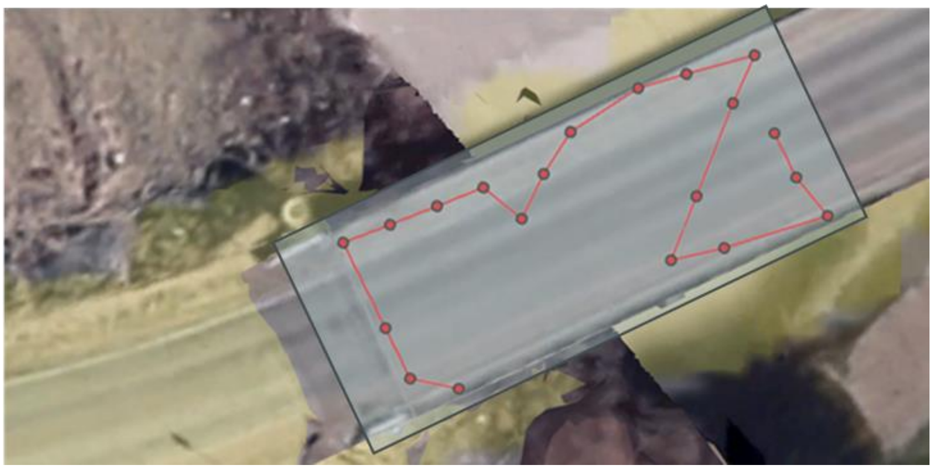


Figure 30. Flight path visualization example

4.5 Result Analysis

The PSO and GA algorithms have been tested on different mission areas, and the results have been compared to the results from DJI GS Pro (DJI GSP) mission planning app for the same areas. Figure 31 represents the energy consumption based on the flight time for each of the algorithms in each inspection flight path planning for different areas. As a result of this chart, PSO achieved a better result (less energy consumption and flight time) compared to GA in all the experiments. The energy consumption results of PSO are optimized for an average of 57% compared to the DJI GSP, whereas the results of GA indicate an average of 50% optimization compared to the DJI GSP. Since these results are based on the flight time, the optimization results for the flight time have the same percentages.

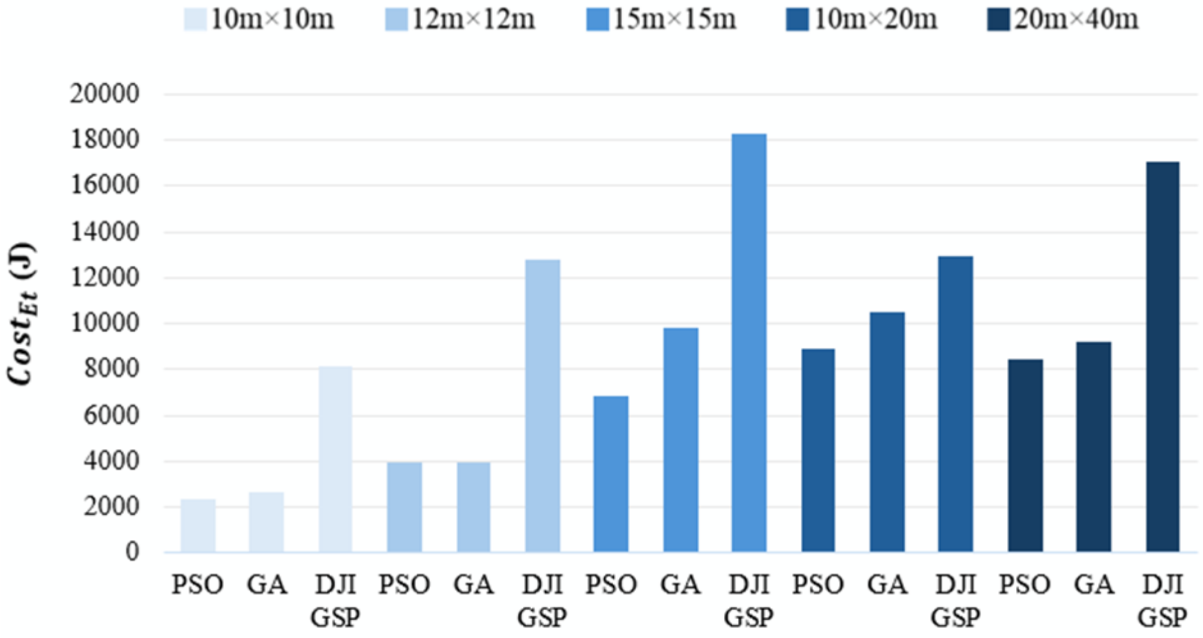


Figure 31. Energy consumption of algorithms based on the flight time for the experiments

TASK 5: Dataset Collection

The objective of this task was to collect comprehensive datasets from bridges across New Mexico, which will be utilized for image processing and model training for the purpose of damage detection and condition assessment. The dataset will be divided into three categories including the training dataset, the validation dataset, and the testing dataset. The collected data include high-resolution images and infrared thermography images capturing various aspects of bridge conditions, such as surface cracks, deformations, and material degradations. Also, different parts of the bridge such as piers, girders, and substructure are included in the captured images. These images are acquired and will be acquired using drones ensuring thorough coverage and detail.

The collected datasets play a crucial role in the field of image processing and model training. High-quality images from RGB cameras provide clear visual data on visible damages like cracks and surface wear, while IRT cameras capture thermal anomalies that indicate underlying structural issues. These datasets will undergo a rigorous preprocessing phase to enhance image quality, remove noise, correct distortions, and ensure consistency across all collected data. This preprocessing step is essential for creating a reliable dataset that can be used to train machine learning models effectively. Annotating these images with detailed information about the observed structural conditions will serve as ground truth for model training, ensuring that the models learn to accurately identify and classify different types of damages.

For this purpose and for now, datasets are collected from 11 bridges with these bridge numbers: 5825, 5629, 5734, 5801, 1791, 8900, 6255, 7264, 7265, 6803, and 5964. The location and NBI number of the bridges are shown in Figure 32.

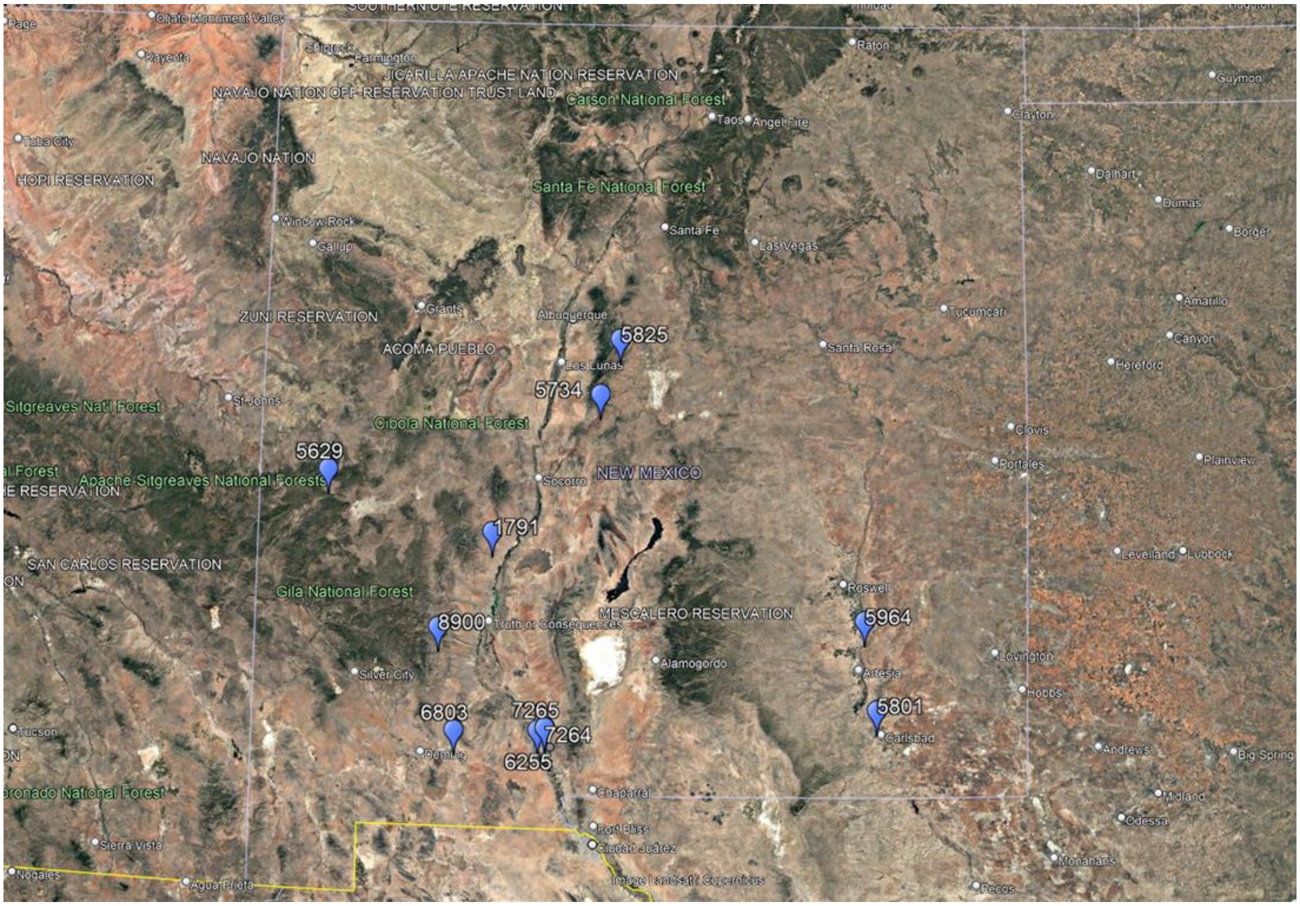


Figure 32. Bridges that are data have been collected from

The task is still ongoing, and data collection continues. The ultimate goal of this task is to develop robust image-processing algorithms and machine-learning models capable of autonomously detecting and classifying structural issues in bridges. Some of the collected RGB images are shown in Figure 33, and thermal image samples are shown in Figure 34:

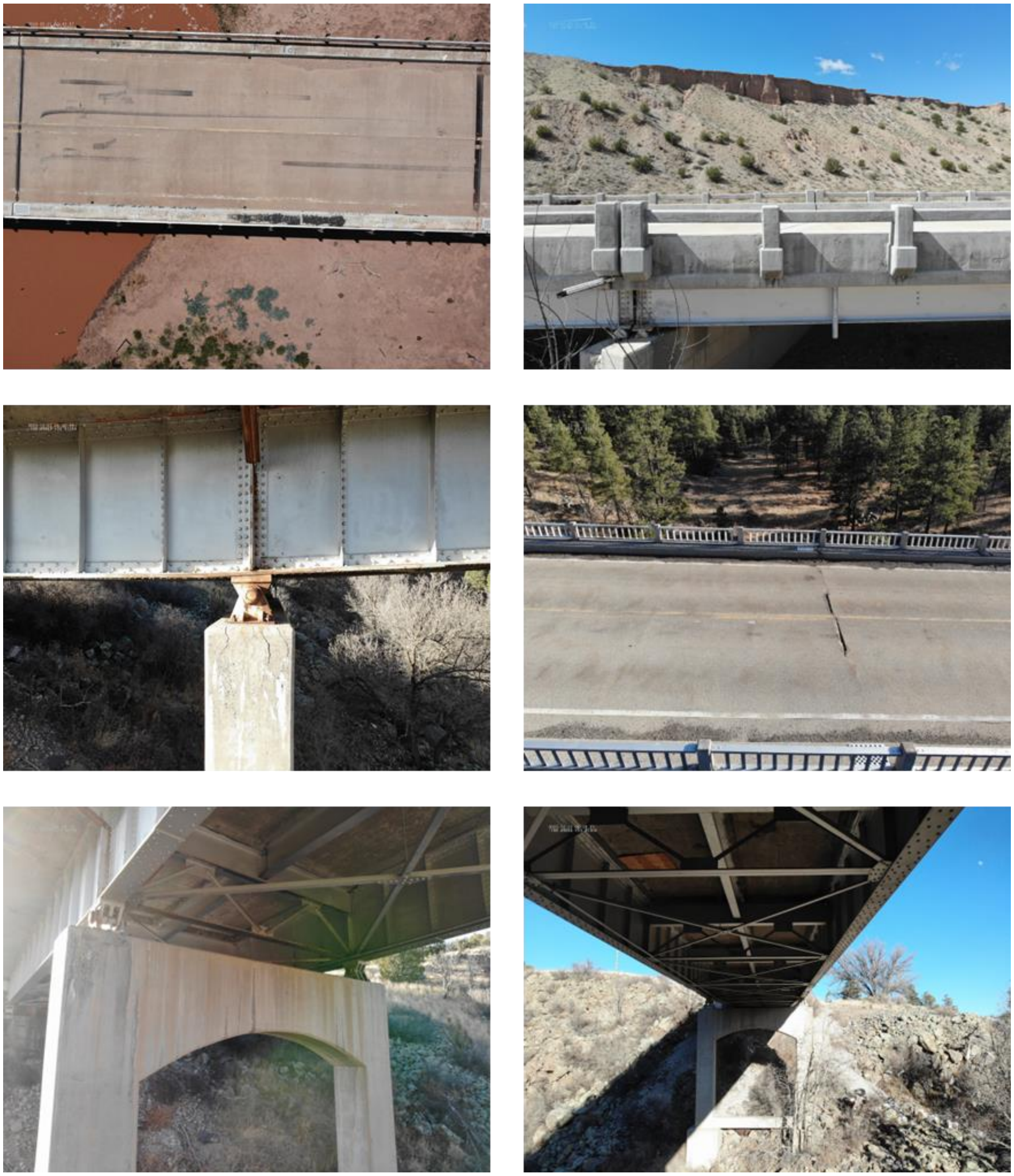


Figure 33. RGB samples from the collected datasets

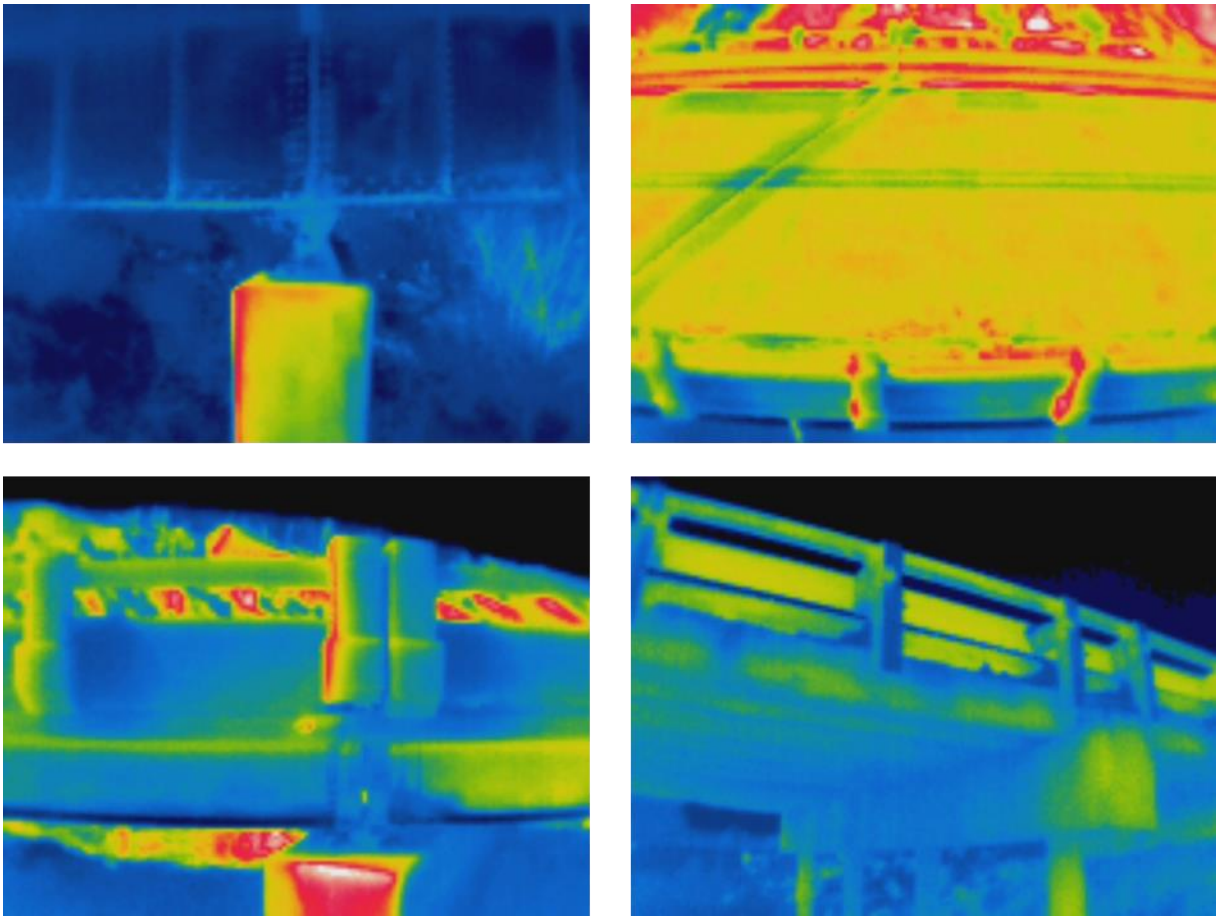


Figure 34. Thermal image samples from the collected datasets

TASK 7: Preliminary Data Processing

At the time of this report, 370 images have been collected from the bridges for the purpose of image processing for crack detection and quantification. Moreover, 40,000 images have been collected from the available datasets from previous crack detection studies which include 20,000 positive and 20,000 negative images.

The following steps have been defined for the data processing task:

1- Data Annotation: The required steps and considerations are defined below for data annotation:

- Two types of annotations are being applied to the data including bounding boxes which are useful for models like Faster R-CNN or YOLO, and pixel-wise segmentation which is essential for more detailed models like UNet or Mask R-CNN, where each pixel within a crack needs labeling.
- Multiple classes are defined to classify crack types (such as small, medium, and wide cracks).
- First, MATLAB is being used for the labeling, but later CVAT will be used as well for more complex segmentation tasks. MATLAB provides an Image Labeler app for labeling

objects within images using bounding boxes. For more detailed segmentation, MATLAB's Image Segmenter app allows us to label cracks pixel by pixel, producing segmentation masks.

- The quality of the annotation will be checked by regularly reviewing and correcting annotations to maintain high accuracy. Also, areas where cracks are hard to see due to lighting, shadows, or image blur, will be marked as "challenging" to be as precise as possible.
- By applying three types of data augmentation including rotation (4 different angles), flip (2 variations), and brightness adjustments (2 levels), the number of variations for each image would be 4*2*2 = 16.
- With 370 original images collected from the bridges, the total augmented images would be 370*16 = 5,920 images.
- The dataset will be divided into three sets including training, validation, and test sets in a 70:15:15 ratio to ensure fair evaluation.
- After evaluating different models, it has been decided to use R-CNN (Region-Based Convolutional Neural Network) architecture (specifically Faster R-CNN and Mask R-CNN) for training due to its advantages which will be discussed later.
- After training the model, evaluation metrics and cross-validation will be used to evaluate pixel-wise predictions and more reliable assessment.
- Post-processing will be applied to predictions by removing small, isolated detections that might be noise.
- The deep learning model will be set up to run directly on UAV onboard computers or will be integrated into a ground control station for real-time processing.

Figure 35 shows an example of the bounding box labeling for crack detection and Figure 36 shows the pixel-wise segmentation in the MATLAB environment.

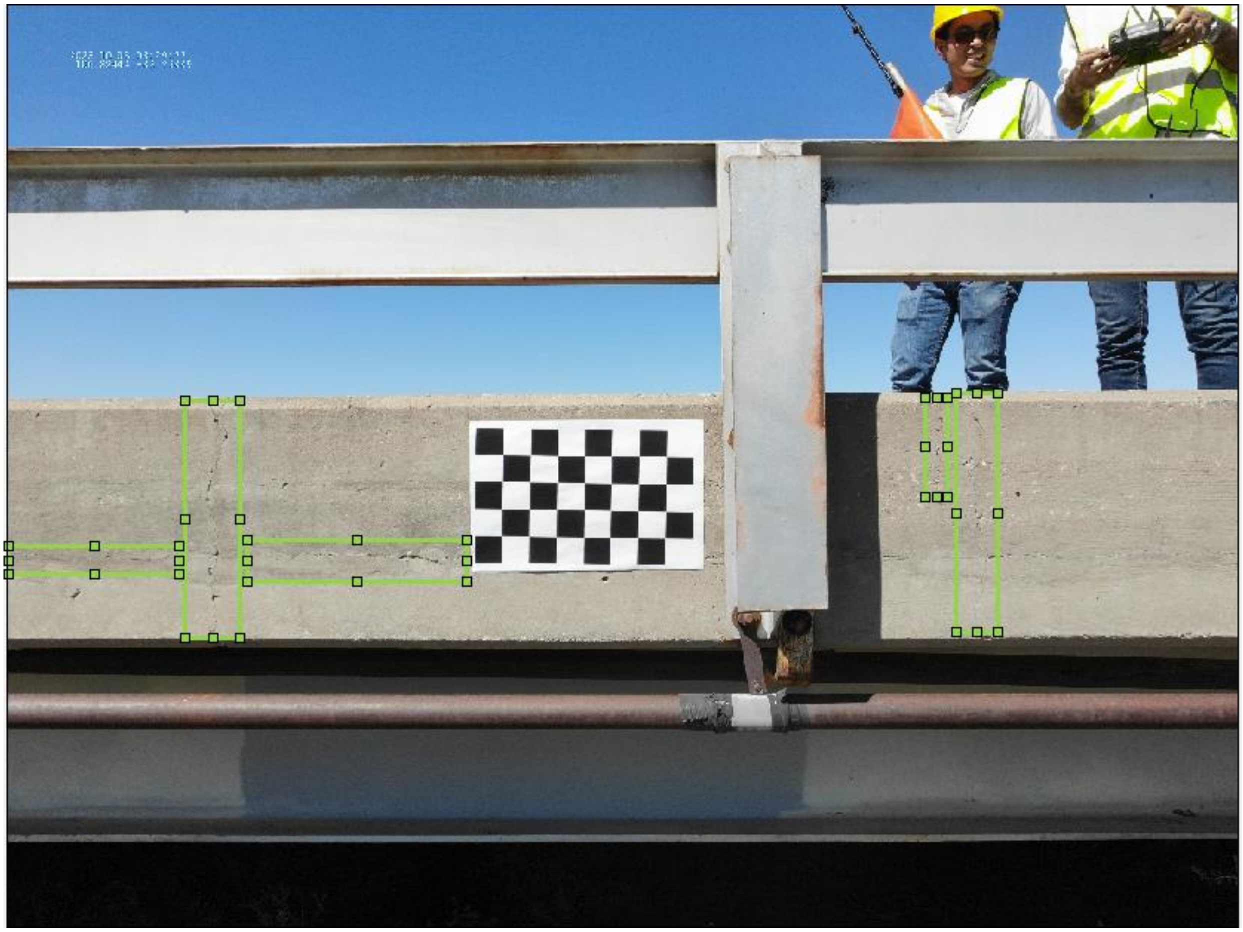


Figure 35. Boundary box labeling in MATLAB

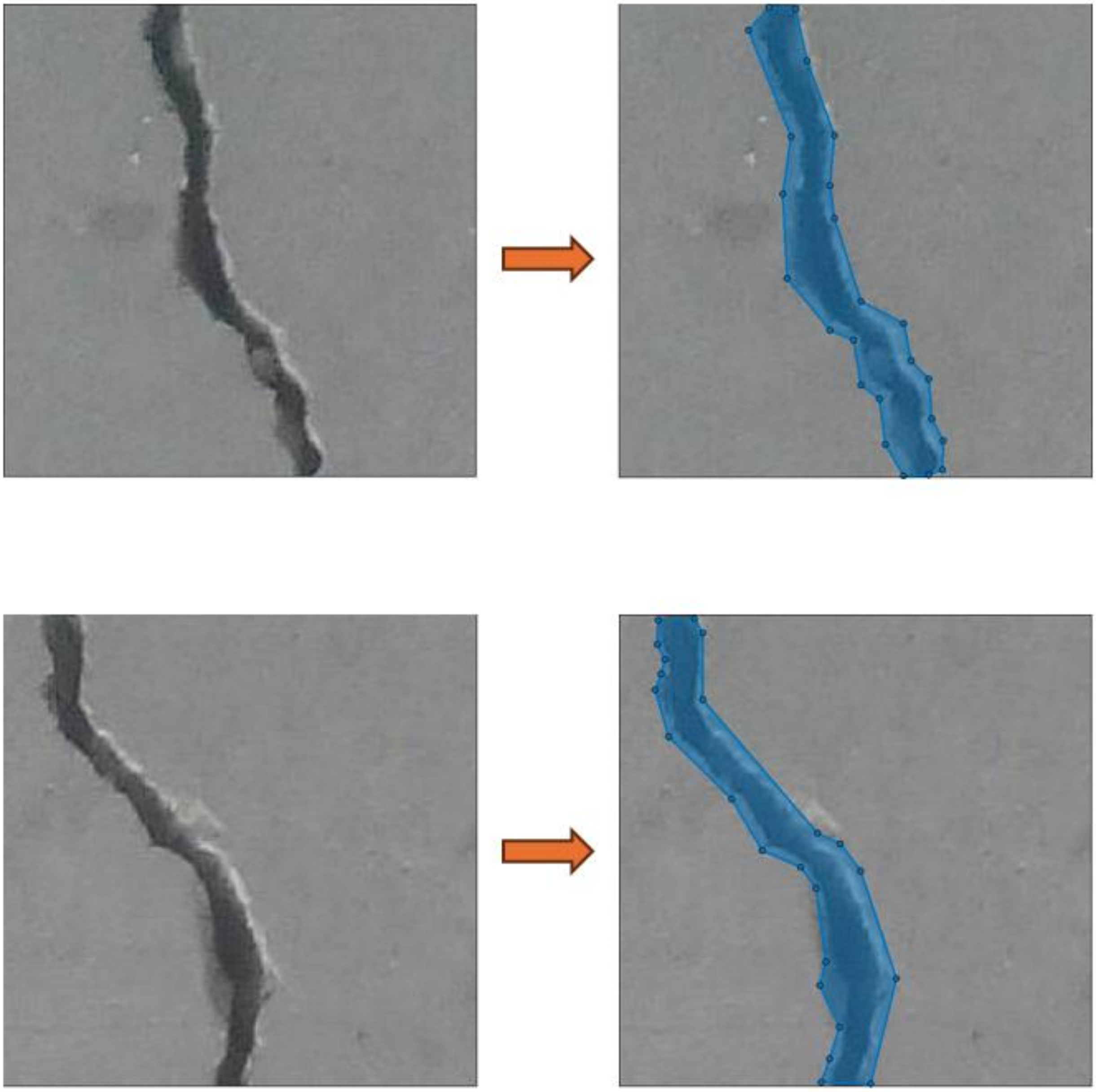


Figure 36. Pixel-wise segmentation in MATLAB

As mentioned above, R-CNN architecture will be used for object detection. R-CNN and YOLO (You Only Look Once) are two popular architectures for object detection, but they have different methodologies and strengths. A detailed comparison highlighting the advantages of R-CNN over YOLO v8 is provided below based on the previous studies and available data:

1. High Localization Accuracy:
    - R-CNN tends to have better localization accuracy for objects compared to YOLO. This is because R-CNN breaks down the process into two stages: region proposal and classification, which allows for more precise bounding box placement.

    - YOLO is designed for real-time detection and processes the entire image in a single pass, sometimes trading off accuracy for speed, leading to less precise localization for small or closely spaced objects.

2. Complex Object Detection:

    - R-CNN can handle the detection of complex objects and overlapping instances better than YOLO. The separate region proposal step allows for multiple detailed predictions, which is particularly beneficial for scenarios with complex scenes and overlapping objects.

    - YOLO can struggle with overlapping or small objects since it divides the image into a fixed grid, potentially missing detections or misclassifying them when multiple objects are close together.

3. Customizability:

    - R-CNN allows for more flexibility in architecture changes, as each stage (region proposal, feature extraction, and classification) can be modified or fine-tuned separately. This modular nature is advantageous for research purposes or when experimenting with custom approaches.

    - YOLO, on the other hand, is less modular because it relies on a unified network for end-to-end predictions.

4. Precision for High-Resolution Images:

    - R-CNN can be more effective for detecting small objects in high-resolution images, as it evaluates specific regions of interest at a finer scale. The dedicated region proposal network helps in focusing on parts of the image that are more likely to contain objects.

    - YOLO processes the entire image at once, which may miss smaller details or fail to detect tiny objects due to the spatial constraints imposed by its grid structure.

While R-CNN has the above advantages, it also comes with certain drawbacks:

- Speed: YOLO v8 is significantly faster than R-CNN, making it more suitable for real-time applications.

- Complexity and Training Time: R-CNN and its variants (e.g., Fast R-CNN, Faster R-CNN) involve multiple stages, making them more complex and time-consuming to train compared to the end-to-end approach of YOLO.

- Resource Usage: R-CNN generally requires more computational resources due to its two-step process, whereas YOLO is more lightweight and optimized for efficiency.

It is worth noting that with original images at 4056*3046 pixels, we have high-resolution data that can capture fine crack details, which is excellent for precision in crack detection. However, Common target sizes for image-based models are 224*224 or 512*512 pixels. Therefore, two model training will be done on the dataset. First training will be done by using the original images collected from the bridges by using UAV and the other training will be done by downscaling and tiling the original images into 224*224 or 512*512 pixels images.

CONCLUDING REMARKS

Based on this report, the following concluding remarks can be stated:

1) Different requirements, documents, and training to become a drone pilot by passing FAA Part 107 certification have been discussed.

2) The effective parameters in the UAV platform and camera/sensor selection have been analyzed and discussed as well as the different types of UAVs and cameras.

2) This study has outlined a comprehensive method for pre-flight preparation and data collection in bridge inspection using UAVs, which significantly impacts data quality and data processing. Diverse aspects of data collection, including flight objectives, hardware selection, flight path planning, and camera calibration, have been thoroughly examined in this study. Two case studies have been done for the evaluation of the proposed method and the results of in-lab and on-site camera calibration are discussed and compared. The results indicate a 7% to 22% improvement in crack detection accuracy.

3) By Optimizing the flight path for UAV-based bridge inspection, one of the main challenges of using UAVs which is short endurance can be solved. Flight path optimization is beneficial for time and energy saving to fully harness the potential of UAVs for bridge inspection. The PSO meta-heuristic algorithm has been used for the flight path optimization engine due to the better results of this algorithm (an average of 50% energy saving).

4) The UI design process for flight path planning software has been started, and inputs and outputs are defined.

5) RGB and thermal image datasets are collected from 11 bridges and the data collection will be continued. Also, 40,000 images have been collected from available datasets for the data processing phase.

6) The data processing phase has been started by labeling the images (bounding boxes and pixel-wise segmentation) in MATLAB and the next steps are defined.

References:

[1] S. Feroz and S. Abu Dabous, “UAV-Based Remote Sensing Applications for Bridge Condition Assessment,” *Remote Sensing*, vol. 13, no. 9, p. 1809, May 2021, doi: 10.3390/rs13091809.

[2] Y. Xu and Y. Turkan, “BrIM and UAS for bridge inspections and management,” *ECAM*, vol. 27, no. 3, pp. 785–807, Nov. 2019, doi: 10.1108/ECAM-12-2018-0556.

[3] S. Sreenath, H. Malik, N. Husnu, and K. Kalaichelavan, “Assessment and Use of Unmanned Aerial Vehicle for Civil Structural Health Monitoring,” *Procedia Computer Science*, vol. 170, pp. 656–663, 2020, doi: 10.1016/j.procs.2020.03.174.

[4] W. W. Greenwood, J. P. Lynch, and D. Zekkos, “Applications of UAVs in Civil Infrastructure,” *J. Infrastruct. Syst.*, vol. 25, no. 2, p. 04019002, Jun. 2019, doi: 10.1061/(ASCE)IS.1943-555X.0000464.

[5] S. HekmatiAthar, N. Goudarzi, A. Karimoddini, A. Homaifar, and D. Divakaran, “A systematic evaluation and selection of UAS-enabled solutions for bridge inspection practices,” in *2020 IEEE Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2020, pp. 1–11. doi: 10.1109/AERO47225.2020.9172795.

[6] Z. Ameli, Y. Aremanda, W. A. Friess, and E. N. Landis, “Impact of UAV Hardware Options on Bridge Inspection Mission Capabilities,” *Drones-Basel*, vol. 6, no. 3, p. 64, Mar. 2022, doi: 10.3390/drones6030064.

[7] B. Hubbard and S. Hubbard, “Unmanned Aircraft Systems (UAS) for Bridge Inspection Safety,” *Drones*, vol. 4, no. 3, p. 40, Aug. 2020, doi: 10.3390/drones4030040.

[8] C. Cheng, Z. Shang, and Z. Shen, “Automatic delamination segmentation for bridge deck based on encoder-decoder deep learning through UAV-based thermography,” *NDT & E International*, vol. 116, p. 102341, Dec. 2020, doi: 10.1016/j.ndteint.2020.102341.

[9] S. Dorafshan, L. E. Campbell, M. Maguire, and R. J. Connor, “Benchmarking Unmanned Aerial Systems-Assisted Inspection of Steel Bridges for Fatigue Cracks,” *Transportation Research Record*, vol. 2675, no. 9, pp. 154–166, Sep. 2021, doi: 10.1177/03611981211001073.

[10] S. Chen, D. F. Laefer, E. Mangina, S. M. I. Zolanvari, and J. Byrne, “UAV Bridge Inspection through Evaluated 3D Reconstructions,” *Journal of bridge engineering*, vol. 24, no. 4, 2019, doi: 10.1061/(ASCE)BE.1943-5592.0001343.

[11] Q. Chen, X. Wen, S. Lu, and D. Sun, “Corrosion Detection for Large Steel Structure base on UAV Integrated with Image Processing System,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 608, no. 1, p. 012020, Aug. 2019, doi: 10.1088/1757-899X/608/1/012020.

[12] P. Debus and V. Rodehorst, “Multi-scale Flight Path Planning for UAS Building Inspection,” in *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering*, vol. 98, E. Toledo Santos and S. Scheer, Eds., in Lecture Notes in Civil Engineering, vol. 98. , Cham: Springer International Publishing, 2021, pp. 1069–1085. doi: 10.1007/978-3-030-51295-8_74.

[13] P. Almasi, R. Premadasa, S. Rouhbakhsh, Y. Xiao, Z. Wan, and Q. Zhang, “A Review of Developments and Challenges of Preflight Preparation for Data Collection of UAV-based Infrastructure Inspection,” *CTCSE*, vol. 10, no. 2, Jan. 2024, doi: 10.33552/CTCSE.2024.10.000734.

[14] P. Almasi *et al.*, “A General Method for Pre-Flight Preparation in Data Collection for Unmanned Aerial Vehicle-Based Bridge Inspection,” *Drones*, vol. 8, no. 8, p. 21, Aug. 2024, doi: 10.3390/drones8080386.

[15] Y. Liu, X. Nie, J. Fan, and X. Liu, “Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 5, pp. 511–529, May 2020, doi: 10.1111/mice.12501.

[16] H. Yanagi and H. Chikatsu, “CAMERA CALIBRATION IN 3D MODELLING FOR UAV APPLICATION,” *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XL-4/W5, pp. 223–226, May 2015, doi: 10.5194/isprsarchives-XL-4-W5-223-2015.

[17] M. Cramer, H.-J. Przybilla, and A. Zurhorst, “UAV CAMERAS: OVERVIEW AND GEOMETRIC CALIBRATION BENCHMARK,” *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W6, pp. 85–92, Aug. 2017, doi: 10.5194/isprs-archives-XLII-2-W6-85-2017.

[18] S. Yu, R. Zhu, L. Yu, and W. Ai, “Effect of Checkerboard on the Accuracy of Camera Calibration,” in *Advances in Multimedia Information Processing – PCM 2018*, vol. 11166, R. Hong, W.-H. Cheng, T. Yamasaki, M. Wang, and C.-W. Ngo, Eds., in Lecture Notes in Computer Science, vol. 11166. , Cham: Springer International Publishing, 2018, pp. 619–629. doi: 10.1007/978-3-030-00764-5_57.

[19] *MATLAB*. ((R2023b)). The MathWorks, Inc. [Online]. Available: https://www.mathworks.com

[20] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000, doi: 10.1109/34.888718.

[21] “Reprojection error.” [Online]. Available: https://support.pix4d.com/hc/en-us/articles/202559369-Reprojection-error

[22] J. Yuan *et al.*, “Global Optimization of UAV Area Coverage Path Planning Based on Good Point Set and Genetic Algorithm,” *Aerospace*, vol. 9, no. 2, p. 86, Feb. 2022, doi: 10.3390/aerospace9020086.

[23] A. Kaveh, P. Almasi, and A. Khodagholi, “Optimum Design of Castellated Beams Using Four Recently Developed Meta-heuristic Algorithms,” *Iran J Sci Technol Trans Civ Eng*, vol. 47, no. 2, pp. 713–725, Apr. 2023, doi: 10.1007/s40996-022-00884-z.

[24] Ramin Babazadeh Dizaj and Nastaran Sabahi, “Optimizing LSM-LSF composite cathodes for enhanced solid oxide fuel cell performance: Material engineering and electrochemical insights,” *World J. Adv. Res. Rev.*, vol. 20, no. 1, pp. 1284–1291, Oct. 2023, doi: 10.30574/wjarr.2023.20.1.2183.

[25] D. E. Goldberg and J. H. Holland, “Genetic Algorithms and Machine Learning,” *Machine Learning*, vol. 3, no. 2/3, pp. 95–99, 1988, doi: 10.1023/A:1022602019183.

[26] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, no. 1, pp. 29–41, Feb. 1996, doi: 10.1109/3477.484436.

[27] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, Perth, WA, Australia: IEEE, 1995, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[28] A. Kaveh and S. Talatahari, “A novel heuristic optimization method: charged system search,” *Acta Mech*, vol. 213, no. 3–4, pp. 267–289, Sep. 2010, doi: 10.1007/s00707-009-0270-4.

[29] A. Kaveh and V. R. Mahdavi, “Colliding bodies optimization: A novel meta-heuristic method,” *Computers & Structures*, vol. 139, pp. 18–27, Jul. 2014, doi: 10.1016/j.compstruc.2014.04.005.

[30] A. Kaveh and A. Zaerreza, “Shuffled Shepherd Optimization Method: A New Meta-Heuristic Algorithm,” in *Structural Optimization Using Shuffled Shepherd Meta-Heuristic Algorithm*, vol. 463, in Studies in Systems, Decision and Control, vol. 463. , Cham: Springer Nature Switzerland, 2023, pp. 11–52. doi: 10.1007/978-3-031-25573-1_2.

[31] A. Kaveh and F. Shokohi, “APPLICATION OF GREY WOLF OPTIMIZER IN DESIGN OF CASTELLATED BEAMS”.

[32] Z. Fu, J. Yu, G. Xie, Y. Chen, and Y. Mao, “A Heuristic Evolutionary Algorithm of UAV Path Planning,” *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–11, Sep. 2018, doi: 10.1155/2018/2851964.

[33] Z. Yu, Z. Si, X. Li, D. Wang, and H. Song, “A Novel Hybrid Particle Swarm Optimization Algorithm for Path Planning of UAVs,” *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22547–22558, Nov. 2022, doi: 10.1109/JIOT.2022.3182798.

[34] H. Rienecker, V. Hildebrand, and H. Pfifer, “Energy optimal 3D flight path planning for unmanned aerial vehicle in urban environments,” *CEAS Aeronaut J*, vol. 14, no. 3, pp. 621–636, Jul. 2023, doi: 10.1007/s13272-023-00666-x.

[35] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, “Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection,” *Autom. Constr.*, vol. 81, pp. 25–33, Sep. 2017, doi: 10.1016/j.autcon.2017.04.013.

[36] R. Zhang, X. Li, H. Ren, Y. Ding, Y. Meng, and Q. Xia, “UAV Flight Path Planning Based on Multi-Strategy Improved White Sharks Optimization,” *IEEE Access*, vol. 11, pp. 88462–88475, 2023, doi: 10.1109/ACCESS.2023.3304708.

[37] X. Bai, H. Jiang, J. Cui, K. Lu, P. Chen, and M. Zhang, “UAV Path Planning Based on Improved A ∗ and DWA Algorithms,” *International Journal of Aerospace Engineering*, vol. 2021, pp. 1–12, Sep. 2021, doi: 10.1155/2021/4511252.

[38] A. Souto, R. Alfaia, E. Cardoso, J. Araújo, and C. Francês, “UAV Path Planning Optimization Strategy: Considerations of Urban Morphology, Microclimate, and Energy Efficiency Using Q-Learning Algorithm,” *Drones*, vol. 7, no. 2, p. 123, Feb. 2023, doi: 10.3390/drones7020123.

[39] A. Israr, Z. A. Ali, E. H. Alkhammash, and J. J. Jussila, “Optimization Methods Applied to Motion Planning of Unmanned Aerial Vehicles: A Review,” *Drones*, vol. 6, no. 5, p. 126, May 2022, doi: 10.3390/drones6050126.

[40] C. Zhang, Y. Zou, F. Wang, E. Del Rey Castillo, J. Dimyadi, and L. Chen, “Towards fully automated unmanned aerial vehicle-enabled bridge inspection: Where are we at?,” *Construction and Building Materials*, vol. 347, p. 128543, Sep. 2022, doi: 10.1016/j.conbuildmat.2022.128543.

[41] G. Gugan and A. Haque, “Path Planning for Autonomous Drones: Challenges and Future Directions,” *Drones*, vol. 7, no. 3, p. 169, Feb. 2023, doi: 10.3390/drones7030169.

[42] M. Clerc and J. Kennedy, “The particle swarm - explosion, stability, and convergence in a multidimensional complex space,” *IEEE Trans. Evol. Computat.*, vol. 6, no. 1, pp. 58–73, Feb. 2002, doi: 10.1109/4235.985692.