# ABECIS: an Automated Building Exterior Crack Inspection System using UAVs, Open-Source Deep Learning and Photogrammetry

**P. Ko [a], S. A. Prieto [a], B. García de Soto [a]**

[a] S.M.A.R.T. Construction Research Group, Division of Engineering, New York University Abu Dhabi (NYUAD), Experimental Research Building, Saadiyat Island, P.O. Box 129188, Abu Dhabi, United Arab Emirates

E-mail: pk2269@nyu.edu, samuel.prieto@nyu.edu, garcia.de.soto@nyu.edu

**Abstract –**

**Inspecting the exteriors of buildings is a slow and risky task for workers, especially in high-rise buildings. Moreover, some areas are difficult to reach for large buildings, and in some cases, the inspections cannot be adequately done. In recent years, there has been an increase in open-source artificial intelligence (AI) technologies and commercially available Unmanned Aerial Vehicles (UAVs) with AI-assisted deep learning capabilities. They can provide a low-cost, open-source, and customizable methodology for building exterior inspections that are readily accessible for construction and facility managers.**

**This study presents a methodology to use UAVs and deep learning technology to conduct an automated inspection for cracks on high-rise buildings - improving the efficiency of the process and the workers' safety while reducing data-collection errors. The proposed methodology is divided into four components: 1) Developing a UAV system to capture the exterior wall images of the building in an autonomous way, 2) Collecting data, 3) Processing and analyzing the images captured for cracks using deep learning, and 4) Rendering the identified locations of the cracks on a 3D model of the building, constructed using photogrammetry, for clear visualization.**

**This study focuses on the virtual simulation of the methodology. The UAV used contains a built-in camera to capture the images of the building from different sides. Data Collection, Image-Analysis, and Photogrammetry are done using publicly available open-source deep learning and simulation technologies. The generated code for the UAV simulation and the crack detection algorithm with the pre-trained data model are released on GitHub.**

**Keywords – Unmanned aerial vehicles (UAVs); Building Inspection; Drone; Deep Learning; Artificial Intelligence; Crack Detection; Building Defects**

## 1 Introduction

Building inspection using only manual labor (i.e., human workers) is a time-consuming and, in some cases, dangerous process. Traditionally, the visual inspection part of the exterior of a building requires an inspector to abseil down over different sides of a building [1]. Not only is this a risky process, but also subjective and prone to errors. Moreover, the inspection of buildings for quality and identification of defects is a repeating process with intervals usually between 5 to 10 years, depending on the maintenance plan [2]. In general, manual inspections do not leave consistent computerized (i.e., digital) data that can be later used to compare the results of successive inspections over time.

One of the most common indicators determined in visual inspection to assess the damage or deterioration of a building is the cracks on the walls. In recent years, there has been an increase in research that utilizes image processing techniques and Artificial Intelligence (AI) technologies to detect and classify cracks with varying levels of success [3].

The emergence and widespread use of commercial Unmanned Aerial Vehicles (UAVs) in this decade have allowed an easier method to collect image and video data. Therefore, the ability to capture building image data using UAVs and analyze them for defects using AI and image-processing techniques has become a reality. The use of UAVs for the data collection process is already being used in some aspects of the industry, but the automation of the processing of said data is still something that has not been fully implemented. To address that, this approach helps with the automation of the whole inspection process and can improve the precision compared to traditional (i.e., manual) building inspection methods.

This study proposes one readily accessible and cost-effective methodology for the inspection of building exteriors. The rest of this article is organized as follows. First, a literature review discusses the recent state-of-the-art research, including a discussion of UAVs, flight path planning, crack detection using deep learning, and the photogrammetry visualization process of UAV captured images. Then, the research methodology section discusses the process of developing an Automated Building Exterior Crack Inspection System (ABECIS), along with assumptions made and limitations of the study. After that, an example illustrates the methodology with a proof-of-concept demonstration in a virtual simulation environment. Finally, the rest summarizes the results, discussion, and outlook.

## 2 Literature Review

For this study, the literature review focused on publications in the proceedings of the International Symposium on Automation and Robotics in Construction (ISARC), as well as *ICACT* [4], *Automation in Construction* [5], and *Drones* [6].

Through keyword-based search, this study identified the state-of-the-art research trends in construction automation, using the following keywords: “Deep Learning”, “photogrammetry”, “UAVs”, “crack detection”, “building inspection”, and “computer simulation”. The sources are selected based on the clarity and novelty in the methods they present.

From the literature review, it was found that many ISARC papers deal with topics relating to building crack analysis and UAV path planning. Similar studies on crack detection using UAVs include the work by Phung et al. [7], who carried out crack detection experiments on wooden walls. They used a UAV to take images of the suspected surfaces, which then were stitched and processed based on histogram analysis. They developed a peak detection algorithm for image clustering and a locally adjustable thresholding technique for crack detection. Results from their experiments showed that out of two crack candidates, only one was detected. The authors planned to extend their work to improve their algorithms to account for other crack properties such as length, width, and orientation.

In literature outside ISARC, attempts to make use of UAVs for building inspection have been investigated. For example, Choi and Kim [4] used an open-source hardware-based Hexa-copter (UAV with six propellers) to acquire building images and videos. Afterward, the images and videos were converted to grayscale, and the Canny edge detection algorithm was used to detect the edges and cracks. The authors proposed follow-up studies by installing various sensors and thermal infrared cameras on the UAV.

The study by Rakha and Gorodetsky [5] provides a comprehensive literature review on the use of UAVs to visualize the heat transfer in buildings with infrared imaging and create digital models using 3D photogrammetry. They presented a procedure to generate a 3D model of a Syracuse University building using a DJI Inspire 1 drone. Their procedure can be summarized as follows. First, images of the building were captured by the drone with autonomous path planning. Then, the Pix4D program was used to generate a 3D point cloud of the building from the 2D images captured. However, the software used in their study was not open-source.

A summary of literature utilizing UAV and image-processing techniques for crack detection is given in Table 1.

Table 1. Comparison of UAV and image processing-based methodologies for crack detection

| Author | Level of automation* | Open-source software used? | Consumer drones used? |
|---|---|---|---|
| Choi and Kim[4] | N/A | No | No |
| Rakha and Gorodetsky [5] | FA | No | Yes |
| Phung et al. [7] | FA | No | Yes |
| Kim et al. [8] | N/A | No | Yes |

**Fully autonomous (FA), Not mentioned (N/A)*

Within the scope of this specific literature review, there have not been studies that emphasize the possibility of utilizing publicly available open-source software and off-the-shelf consumer drones to carry out building crack detection processes. Therefore, to make it more accessible, the proposed method only considers open-source software to provide a cost-effective and easy-to-replicate approach compared to existing ones since expensive or specialized hardware and software are not necessary.

### 2.1 UAV for Building Exterior Inspection

#### 2.1.1 Commercially Available UAVs

UAVs, also known as drones, had origins in military research. However, small and inexpensive commercially available UAVs, built from easily available components, are becoming increasingly common as an emerging technology. Some of the leading consumer UAVs include DJI, 3DRobotics, and Parrot [9].

For this research, a DJI Mavic 2 drone (Figure 1) is used in a simulated environment. The drone used is a lightweight UAV with a high-resolution camera and uses GPS and GLONASS technologies for navigation. Once fully charged, it has an autonomy of 31 minutes under normal conditions and usage [8].


Figure 1. DJI Mavic 2 [10]

#### 2.1.2 Advantages and Limitations of UAVs

The benefits of using UAVs for building inspection are numerous. Autonomous navigation enables a high level of automation, and the flight ability allows the UAV to reach points on structures or roofs, which are otherwise difficult or dangerous to access. Nevertheless, the current autonomous features of the UAVs are quite limited. For instance, there are inevitable mismatches between the planned flight paths and real paths followed by the drone due to localization errors caused by built-in GPS [7], and under most regulations, drones are not allowed to fly without an operator and need to stay within the operator's visual line of sight [11].

#### 2.1.3 Flight Path Planning Methods

Multiple flight patterns exist for UAVs to explore areas (Figure 2) [6]. Nevertheless, for building exterior inspections, varying external conditions such as different viewpoints, view angles, and daylight must be considered for a successful exploration [12]. The process is simpler when the building is divided into rectangular walls for the UAV to explore in multiple phases. For the drone to inspect the greatest area while remaining autonomous, the back-and-forth path in Figures 2 (a) and (b) prove sufficient and effective.

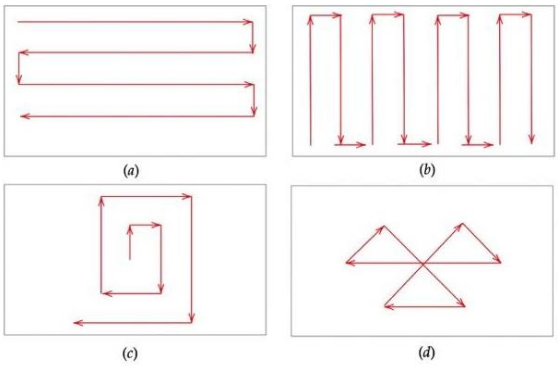


Figure 2. Simple UAV flight patterns in rectangular areas (a) Parallel; (b) Creeping Line; (c) Square; (d) Sector Search (adapted from [6])

### 2.2 Building Cracks

#### 2.2.1 Crack Classification

In visual building inspection for cracks, three factors, namely the distribution, width, and the depth of the cracks, are identified, with the distribution and the width being more important factors than depth [2].

Moreover, the building cracks can be classified based on their pattern [13] into one of the following categories: (a) alligator cracking; (b) block cracking; (c) longitudinal cracking; (d) hair cracking; (e) diagonal cracking; (f) multi cracking; and (g) transverse cracking. For this research, the cracks considered are those discussed above, except for the alligator cracking and hair cracking due to data limitations to training the algorithm.

### 2.3 Deep Learning and Crack Detection

#### 2.3.1 Open-Source Deep Learning Frameworks

Deep Learning is a branch of artificial neural networks, an AI technique widely used to classify images [14]. Deep Learning image classification methods began gaining popularity in 2012 [15] and outperform existing classification methods (such as Local Binary Patterns (LBP) and shape-based algorithms [14]). In recent years, there has been an increase in publicly available open-source software libraries which allow users to build artificial neural networks for Deep Learning very easily, with Keras [16] being the most popular open-source library. This research utilizes Keras to build a network to classify the segments of the collected images of the building as "containing crack" or "no crack."

#### 2.3.2 Crack Detection Algorithm

In order to classify images using Deep Learning, the algorithm must be trained first. The training requires thousands of pre-classified images. The dataset can be obtained by oneself or using publicly available data sets.

Regarding the crack detection on walls, researchers, such as Özgenel, have shared their dataset containing pre-classified images of wall cracks publicly online [3]. This study uses Özgenel's dataset to train the Deep Learning algorithm.

### 2.4 Photogrammetry Visualization

In order to visualize the data collected from UAVs, two methods, namely LiDAR and photogrammetry, can be used. However, LiDAR requires specialized, expensive, and heavy equipment (around 14 kg [17]), which is not ideal to be mounted on a commercial off-the-shelf UAV, which typically has a limited payload (0.83 kg in the case of the DJI Mavic 2 [18]).

Therefore, in this research, photogrammetry is used for a 3D reconstruction from the data collected from a UAV. Photogrammetry is the technique of building a 3D model from numerous images captured from multiple viewpoints around an object (in this case, a building). Normally, specialized professional drone mapping software such as Pix4D [19] is used in the industry. However, such software can be quite costly. An alternative is to look at open-source photogrammetry software developed by the community. For example, Meshroom [20] by AliceVision is an open-source software that provides results comparable to most industry-grade software. In addition, Meshroom can perform photogrammetry autonomously and has a very easy learning curve, making it ideal for this application.

# 3 Research Methodology

### 3.1.1 Real-world constraints and Assumptions

In the real world, numerous constraints can interfere with the use of UAVs for building inspections. Some of these practical challenges include the following:

- According to US Federal Aviation Administration regulations, UAVs need to be operated by a registered operator, and they must fly within the visual line of sight of the operator [11].
- The image data collection will only yield optimal results if favorable lighting and weather conditions are fulfilled.
- In urban areas, the existence of adjacent buildings can interfere with the autonomous path planning of the UAV. Moreover, flying over pedestrians, traffic, etc., is also a big constraint.
- Photogrammetry technology has limitations. Depending on the camera angle, shadows in images, and blind spots, the constructed 3D visualization and image-processing results could diverge from real-world data [21].
- Moreover, the UAV, especially the off-the-shelf commercial ones, have limited battery life, and the survey might need to be done in multiple stages.

For this study, these constraints are relaxed. An ideal situation with only one building of interest, isolated in the middle of an open and flat area under favorable weather conditions and lighting, was studied via computer simulation.

### 3.1.2 System Architecture

The overall system architecture for crack detection using UAVs and deep learning is proposed in Figure 3. It includes the UAV system, 3D model photogrammetry, and deep learning image analysis. The code for the UAV simulation and the crack detection algorithm with the pre-trained data model [22] have been released on GitHub for other researchers to use.

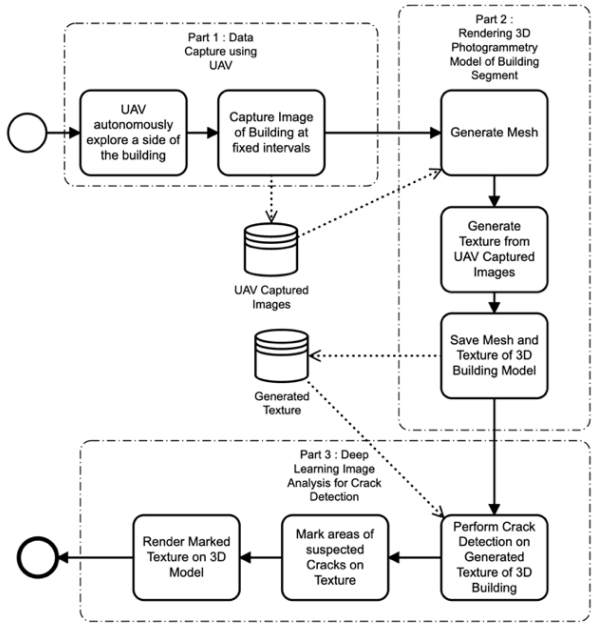


Figure 3. System architecture of the Automated Building Exterior Crack Inspection System (ABECIS)

### 3.1.3 Part 1: Data Capture using UAV

An algorithm was developed in C by implementing a Proportional-Integral-Derivative (PID) controller [23] to control the position of the drone hovering in mid-air. The algorithm uses the Creeping Line flight pattern, shown in Figure 2 (b), to autonomously explore one side of the rectangular wall of the building. Images from the drone camera are captured when the drone is near the wall at fixed intervals. The entire process is autonomous, and a human operator is only required to activate the drone and place it at the bottom of the wall to be analyzed. The operator could also assist (i.e., take control of the UAV) in case of emergency.

The algorithm and the autonomous drone image capture process are simulated using Webots, a free open-source mobile robot simulation software [24]. The captured images are then stored for further analysis.

### 3.1.4 Part 2: 3D Photogrammetry Model

Using the images captured by the drone, a 3D model of the building segment is generated using Meshroom photogrammetry software for clear visualization. Meshroom creates a 3D mesh and a texture automatically from the images. The generated texture is to be analyzed for cracks using deep learning and rendered later on the 3D model of the building.

### 3.1.5 Part 3: Deep Learning Image Analysis

A deep learning image classification algorithm was developed and trained based on Google's Xception convolutional neural network [25]. Özgenel's dataset

[3], composed of 20,001 non-crack images and 20,001 crack images with 227×227 pixels resolution, was used to train the image classification algorithm. Figure 4 shows an example of the sample training data from the dataset used.

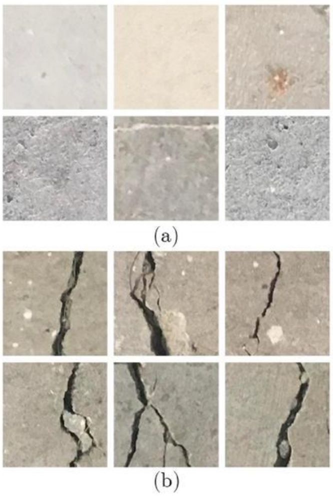


Figure 4. (a) Sample non-crack images from original dataset (b) Sample crack images from the original dataset (reproduced from [3])

The image classification algorithm was developed using Keras, an open-source Python deep learning library. The algorithm operates in the following steps: 1) load pre-trained model 2) take the input image (the texture generated by 3D photogrammetry), 3) split each image into an array of rectangular segments, 4) perform analysis on each segment and classify it as “crack” or “no-crack”, 5) mark the segments which are classified as “crack” for visualization later. The process is visualized in Figure 5.

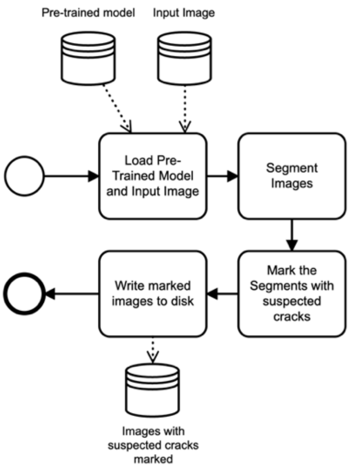


Figure 5. Deep Learning Image Analysis Algorithm

## 4 Implementation

A small example implementation in an ideal simulated environment is then carried out to illustrate the research methodology discussed. The following objects are investigated for cracks in this study.

1. A tall building with cracks on the top floor and
2. A rectangular wall with cracks

The rectangular wall is painted plain white, contains a single long crack, and is used to study whether the algorithm can identify the location of the cracks. The tall building contains cracks present on the colored tiles and is used to study whether the algorithm can identify the cracks superimposed on other textures.

In the following examples, many assumptions are being made. Firstly, the drone inspection is carried out on a single building or a wall without any neighboring buildings. Secondly, the environment is free from obstacles that might interfere with the drone in the real world, such as trees, overhead cables, pedestrians, and vehicles.

For *Part 1: Data Capture* using UAV, a simulated environment was constructed (Figures 6 and 7). The UAV autonomously captures a wall of the building following a Creeping Line flight pattern. The autonomous flight algorithm takes the starting coordinate, length of the wall, distance from the wall, and minimum/maximum heights as input parameters. The building images are taken every second by the drone and are transferred to the computer for analysis.

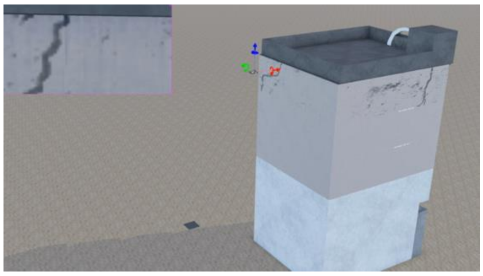

Figure 6. Autonomously Capturing Images of a tall building with UAV (Webots Simulation)

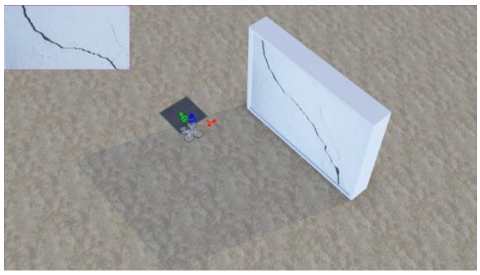

Figure 7. Autonomously Capturing Images of a wall with UAV (Webots Simulation)

Once the drone captured images are ready, *Part 2: 3D Photogrammetry* model generation is carried out using Meshroom photogrammetry software for both the building and the wall. Meshroom analyzes the different angles that the images are taken from and constructs the

3D mesh of the back wall of the building (Figure 8). A texture is also generated with the model (Figure 9 a and b).

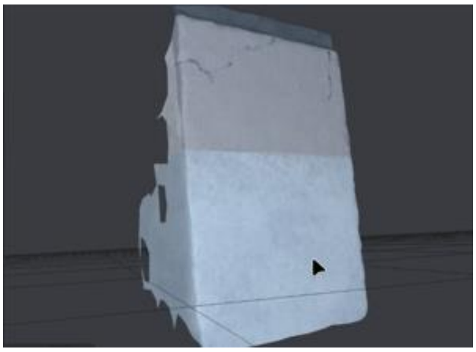

Figure 8. Rendering 3D Photogrammetry Model of Building Segment from captured images in Meshroom

(a) (b)

Figure 9. Textures of (a) the building and (b) the wall, generated from captured images by Meshroom

Once the texture and 3D mesh are generated by Meshroom, *Part 3: Deep Learning Image Analysis* is carried out using the generated building texture as the input image for the deep learning algorithm. The deep learning algorithm used has the following adjustable parameters:

1. **Tile Resolution (TR):** The dimensions of the smallest square images that the algorithm will split the building textures. Then, the algorithm performs analysis for cracks on each tile, and
2. **Confidence Interval (CI):** The cut-off confidence threshold in % to decide whether each tile contains a crack or not.

The deep learning algorithm segments the texture into multiple smaller square images (tiles) and analyses each image to see if cracks are present. Images with a "crack" confidence interval greater than 85% are marked with red crosses. The 85% CI was chosen based on previous experiments since this value gives the lowest number of false positives. From experiments, 75% CI gives two false positives, 85% gives no false positives, and 100% does not detect cracks. Finally, all smaller images are stitched together to form the original building texture, but with the cracks marked. This analyzed texture is then rendered onto the 3D model of the building generated in Part 2 to generate a 3D report as in Figure 10 and Figure 11.

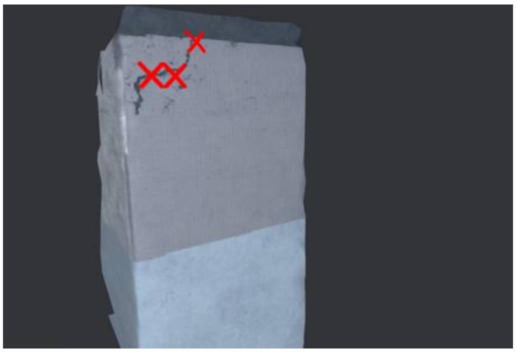

Figure 10. Example of a 3D report for building, with the suspected areas of cracks on the 3D model of the wall marked with an 'X'

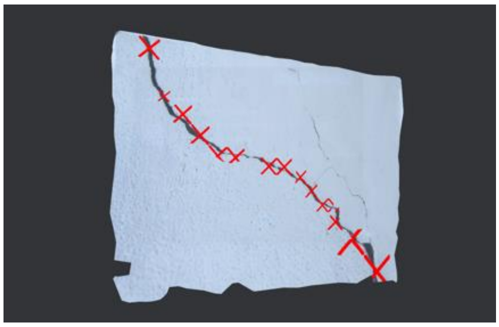

Figure 11. Example of a 3D report for wall, with the suspected areas of cracks on the 3D model of the wall marked with an 'X'

## 5 Results and Discussion

On the 3D report for the building (Figure 10), the proposed method correctly identified the location of most cracks. In the case of the wall (Figure 11), it missed a very thin hairline crack on the right-hand side.

It is suspected that two parameters (TR and CI) contribute to the failure of detection of cracks in our implementation example. Firstly, the cracks may be too small to be seen by a camera mounted on the drone, perhaps due to bad lighting conditions. Secondly, the resolution of textures used for cracks may be altered by the Webots simulation environment, making the crack textures not as realistic as they can be.

This study examines a general overview of the crack detection process (i.e., is a part of one crack detected?) rather than how much detail of one crack is detected. To provide a quantitative measure for the success of our approach, we used a measure called Success Percentage. To calculate Success Percentage, the total number of cracks on generated building texture, NT, and the number of cracks correctly detected (partly or fully), NC are counted. Success Percentage is then defined as

$$Success\ Percentage\ = \frac{N_C}{N_T} \times 100$$

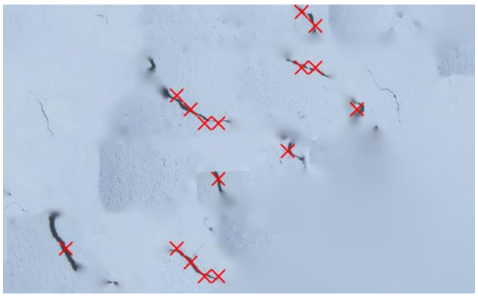

Figure 12. Texture of the wall with cracks identified

The calculated Success Percentages are given in Table 2 below.

Table 2. Success Percentage Results

| Object | Total Number of cracks | Number of cracks detected (partly or fully) | Success Percentage |
|---|---|---|---|
| Building | 2 | 2 | 100% |
| Wall | 10 | 8 | 80% |

In this particular example, the Success Percentage for the wall crack detections is quite high. Except for hairline cracks on the wall, most cracks are detected.

| | | Tile Resolution (TR) | |
|---|---|---|---|
| | | High | Low |
| Confidence Interval (CI) | High | Some cracks are not detected | Cracks larger than the TR not detected |
| | Low | Some False Positives Detected | Many False Positives Detected |

Table 3. Possible negative outcomes of different input parameters

However, the Success Percentage can be increased or decreased by varying the tile resolution and confidence interval input parameters of the algorithm. Possible undesirable outcomes for using too high or too low input parameters for the algorithm are summarized in Table 3. From experiments, CI of 98% and above does not detect cracks. The input parameter should be tuned to suit the inspection requirements.

Nevertheless, the proposed method can accurately pinpoint the estimated locations of most cracks on the wall's surface at a high level (i.e., whether a crack exists or not). Once the estimated locations are identified, one can repeat the process on the suspected areas and move the drone closer to the wall for a clear view of the cracks for further investigation.

## 6 Conclusion and Outlook

This study proposes a low-cost and accessible methodology for identifying cracks on walls using off-the-shelf consumer drones, free and open-source photogrammetry software, and Deep Learning libraries. Although the process failed to detect very thin cracks in the simulated environment, it was able to identify the estimated locations of most cracks correctly. Interested readers can easily replicate our method by using the software mentioned above and our algorithm that has been released on GitHub [22].

Ongoing work by the authors includes extending this study to use an actual drone to conduct the data capturing process in a real environment. We also expect to improve the quality of crack images taken by flying the drones close enough to the building. Moreover, future work includes performing a detailed analysis of each of the cracks and providing more information, such as crack lengths, type of cracks, and the estimated scope and cost of repair to building owners and facility managers.


## Acknowledgment

Part of this research was funded through the undergraduate research assistantship program of New York University Abu Dhabi (NYUAD).


## References


[1] A. M. Paterson, G. R. Dowling, and D. A. Chamberlain, "Building inspection: can computer vision help?," *Automation in Construction*, vol. 7, no. 1, pp. 13–20, 1997, doi: https://doi.org/10.1016/S0926-5805(97)00031-9.

[2] M. P. Ismail, "Selection of suitable NDT methods for building inspection," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 271, p. 012085, Nov. 2017, doi: 10.1088/1757-899X/271/1/012085.

[3] Ç. F. Özgenel and A. G. Sorguç, "Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings," in *Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC)*, Taipei, Taiwan, Jul. 2018, pp. 693–700. doi: 10.22260/ISARC2018/0094.

[4] S. Choi and E. Kim, "Building crack inspection using small UAV," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, Jul. 2015, pp. 235–238. doi: 10.1109/ICACT.2015.7224792.

[5] T. Rakha and A. Gorodetsky, "Review of Unmanned Aerial System (UAS) applications in the built environment: Towards automated building inspection procedures using drones," *Automation in Construction*, vol. 93, pp. 252–264, 2018, doi: https://doi.org/10.1016/j.autcon.2018.05.002.

[6] T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr., "Survey on Coverage Path Planning with Unmanned Aerial Vehicles," *Drones*, vol. 3, no. 1, 2019, doi: 10.3390/drones3010004.

[7] M. D. Phung, T. H. Dinh, V. T. Hoang, and Q. Ha, "Automatic Crack Detection in Built Infrastructure Using Unmanned Aerial Vehicles," *Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC)*, Jul. 2017, doi: 10.22260/isarc2017/0115.

[8] J. Kim, S.-B. Kim, J.-C. Park, and J.-W. Nam, "Development of Crack Detection System with Unmanned Aerial Vehicles and Digital Image Processing," presented at the Advances in Structural Engineering and Mechanics, 2015.

[9] B. Rao, A. G. Gopi, and R. Maione, "The societal impact of commercial drones," *Technology in Society*, vol. 45, pp. 83–90, 2016, doi: https://doi.org/10.1016/j.techsoc.2016.02.009.

[10] "Mavic 2 - Product Information - DJI," *DJI Official*. https://www.dji.com/nl/mavic-2/info (accessed Apr. 17, 2021).

[11] "Federal Aviation Administration - Unmanned Aircraft Systems (UAS)." https://www.faa.gov/uas/recreational_fliers/ (accessed Apr. 17, 2021).

[12] Z. Zheng, M. Pan, and W. Pan, "Virtual Prototyping-Based Path Planning of Unmanned Aerial Vehicles for Building Exterior Inspection," presented at the 37th International Symposium on Automation and Robotics in Construction, Kitakyushu, Japan, Oct. 2020. doi: 10.22260/ISARC2020/0004.

[13] F. M. Nejad and H. Zakeri, "An expert system based on wavelet transform and radon neural network for pavement distress classification," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7088–7101, 2011, doi: https://doi.org/10.1016/j.eswa.2010.12.060.

[14] L. Pauly, H. Peel, S. Luo, D. Hogg, and R. Fuentes, "Deeper Networks for Pavement Crack Detection," in *Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC)*, Taipei, Taiwan, Jul. 2017, pp. 479–485. doi: 10.22260/ISARC2017/0066.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[16] "Keras: the Python deep learning API." https://keras.io/ (accessed Apr. 17, 2021).

[17] J. R. Kellner *et al.*, "New Opportunities for Forest Remote Sensing Through Ultra-High-Density Drone Lidar," *Surveys in Geophysics*, vol. 40, no. 4, pp. 959–977, Jul. 2019, doi: 10.1007/s10712-019-09529-9.

[18] S. Captain, "DJI Mavic Air 2 can lift a lot more than its body weight," *DroneDJ*, Jun. 01, 2020. https://dronedj.com/2020/06/01/dji-mavic-air-2-can-lift-a-lot-more-than-its-body-weight/ (accessed Apr. 24, 2021).

[19] "Professional photogrammetry and drone mapping software," *Pix4D*. https://www.pix4d.com/ (accessed Apr. 17, 2021).

[20] "AliceVision | Meshroom - 3D Reconstruction Software." https://alicevision.org/#meshroom (accessed Apr. 17, 2021).

[21] S. Gupta and S. Nair, "Challenges in Capturing and Processing UAV based Photographic Data From Construction Sites," in *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*, Kitakyushu, Japan, Oct. 2020, pp. 911–918. doi: 10.22260/ISARC2020/0126.

[22] *SMART-NYUAD/Concrete-Wall-Crack-Detection*. S.M.A.R.T. Construction Research Group, 2021. Accessed: Jun. 02, 2021. [Online]. Available: https://github.com/SMART-NYUAD/Concrete-Wall-Crack-Detection

[23] A. R. Al Tahtawi and M. Yusuf, "Low-cost quadrotor hardware design with PID control system as flight controller," *TELKOMNIKA*, vol. 17, no. 4, p. 1923, Aug. 2019, doi: 10.12928/telkomnika.v17i4.9529.

[24] "Webots: Open-source Mobile Robot Simulation Software." https://cyberbotics.com/ (accessed Apr. 24, 2021).

[25] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.