Computer Vision-based road surveillance system using autonomous drones and sensor fusion

1st Pablo Zubasti Recalde
Department of Computer Science and Engineering
Universidad Carlos III de Madrid
Madrid, Spain
pzubasti@pa.uc3m.es

2nd Mario Saiz Fernández
Department of Computer Science and Engineering
Universidad Carlos III de Madrid
Madrid, Spain
masaizf@pa.uc3m.es

3rd Jesús García Herrero
Department of Computer Science and Engineering
Universidad Carlos III de Madrid
Madrid, Spain
jgherrer@inf.uc3m.es

4th José Manuel Molina López
Department of Computer Science and Engineering
Universidad Carlos III de Madrid
Madrid, Spain
molina@ia.uc3m.es

***Abstract*—This paper shows an innovative approach to road monitoring by integrating autonomous drones and sensor fusion within a computer vision-based system. By employing different sets of algorithms, drones equipped with cameras, GPS, and ultrasonic distance sensors can efficiently detect and geolocate road damages, providing crucial data for maintenance and infrastructure management. The system's key components include automated planning, autonomous flight capabilities, object detection, and sensor fusion techniques, enhancing scalability and adaptability. The main objective is to use context information to compute a flying plan where the subsequent detection of defects allows us to expand and enhance GIS data. The implementation of such a system holds significant potential for improving road safety and optimizing maintenance costs, marking a notable advancement in the convergence of autonomous technologies, sensor fusion, and computer vision for effective road surveillance.**

***Index Terms*—Road surveillance, autonomous drones, sensor fusion, computer vision, GIS data, infrastructure management, automated planning, object detection.**

I. Introduction

Detecting defects in roads poses a significant challenge within the framework of road safety and the maintenance of public and private infrastructures. Roads represent one of the most important and relevant investments for the modern society of the 21st century [1]. Their deployment across the territory of any nation provides the structural basis upon which vehicles travel comfortably and supports all terrestrial transportation logistics. The scale of these structures becomes colossal, and their maintenance proves to be a complicated and often seemingly endless challenge due to the vast number of kilometers these structures accumulate. The human factor is crucial in the realm of maintenance, as it is responsible for supervising the state of roads to locate defects and establish compilations that capture the average state of the roads. The cost of road maintenance represents a significant investment for various government and management institutions since the deterioration of these structures is constant, and their maintenance is essential to ensure that roads continue to fulfill their primary objective effectively and safely. Deterioration is accentuated during the occurrence of violent weather events, such as heavy rains, ice, earthquakes and the continous effects of climate change [2]. The state of roads directly affects the wear and tear rate of the vehicles that traverse them, making their maintenance also a particular interest for users.

Recent works [3] related to road defect detection and classification have been developed in recent years, and the vast majority of them are entirely based on Machine Learning techniques such as Convolutional Neural Networks (CNN) [4] for detection, segmentation, and classification of the corresponding defects. Recent literature aims at using deep learning neural network architectures and, more precisely, from the YOLO family [5].

The proposed system augmentates context information of roads by detecting surface defects using UAV imagery, computer vision and sensor fusion, producing an additional GIS layer that could be used for further drone missions, road monitoring, maintenance, defect classification and road safety rankings based on the quality of the infrastructure.

II. Previous works

The detection of road defects through deep learning techniques and convolutional neural networks (CNNs) has undergone significant advancement in recent years. These methodologies have demonstrated their efficacy in automating and enhancing precision in identifying defects on road surfaces, thereby significantly contributing to maintenance and road safety.

Semantic segmentation stands as a fundamental technique in road defect detection. CNNs have proven effective in precisely segmenting road images [6]–[8], and identifying and categorizing various types of defects [9] such as cracks, potholes, and pavement deformations.

The availability of specific datasets for road defect detection has propelled the development of deep learning algorithms. Datasets like CFD (Crack Forest Dataset) and RDD (Road Damage Dataset) contain labeled images featuring diverse defect types, facilitating the training of detection models with high performance and generalization.

Architectures of convolutional neural networks specifically tailored for road defect detection have been proposed. These architectures typically incorporate specialized layers for defect identification and classification, alongside attention mechanisms to highlight relevant regions in images.

The utilization of transfer learning and fine-tuning has proven effective in road defect detection, particularly in scenarios with limited datasets. By leveraging pre-trained models on larger datasets, researchers can swiftly adapt models to new road defect detection tasks with reduced computational costs.

Using drones and UAVs for data collection enabling the application of detection algorithms has been extensively studied and developed in the past [10]. The low cost associated with the energy consumption of commercial drones and the ability to conduct remotely piloted flights have rendered drones a pivotal technological asset in acquiring high-quality aerial imagery. The applications of defect detection systems through drone-captured images extend beyond the realm of roadways, encompassing various infrastructural arrays [11] necessitating analogous monitoring and maintenance mechanisms. Leveraging the high computing capabilities embedded within onboard drone computers, researchers have proposed real-time detection systems [12] capable of conducting flight and detection processes online.

Autonomous drone control entails the utilization of flight controllers (software) embedded within lightweight microcontrollers (hardware), facilitating the execution of flight mission instructions defined by the user. The flight controller itself manages sensor data by filtering the inherent noise present in measurements, thereby enabling precise and finely-tuned flight quality. Recent studies [13] demonstrate how handling noise in these controllers is not a trivial task and its impact on flight precision and quality. While not the primary focus of this article, comprehending the presence of noise in sensor measurements crucially underpins the evaluation of accuracy in a data fusion system as proposed herein.

III. System Explanation

The proposed system aims to automate the process of detecting defects in roads, thereby adding an additional layer of contextual information to the original data. The primary objective of the system is to serve as a fully automated component within the context of road monitoring and maintenance, as defined by the iterative cycle illustrated in Fig. 1.

The execution pipeline of the system (and by extension, of an iteration) is delineated by five phases (Fig. 2):

1) Computation of suboptimal routes based on the original contextual information.
2) Execution of the mission and data collection.
3) Detection of road defects.

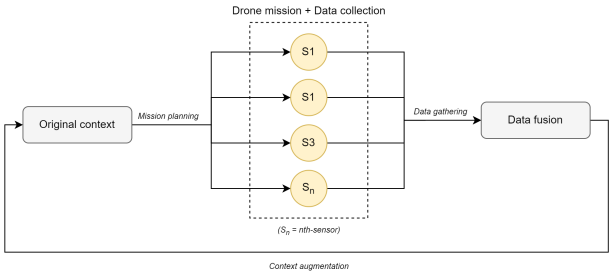

Fig. 1. Basic context augmentation loop (equivalent to one iteration).

4) Fusion of sensor information.
5) Build a GIS layer containing information about the state of the road infrastructure.

Fig. 2. System pipeline.

A. Base road context for automated planning

In the planning phase, our aim is to generate a graph defining the road network upon which suboptimal planning algorithms can be applied.

The original information on which the system relies is based on XODR (OpenDrive) files [14], which accumulate data regarding both the physical (dimensions, curvature, etc.) and logical (number of lanes, direction of traffic flow, etc.) characteristics of roads. XODR files provide a precise description of real roads, enabling the creation of a hyper-realistic 3D reconstruction. This reconstruction serves as both a simulation environment and a starting point for automated planning.

The road is described by sequences of vertices with local Cartesian coordinates. However, the flight plans constituting drone missions must be defined by global coordinates. Thus, these local coordinates are transformed into global geodetic coordinates using Eq. 1.

$$ENU \rightarrow ECEF$$

$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} = \begin{bmatrix} -\sin\lambda_r & -\sin\phi_r\cos\lambda_r & \cos\phi_r\cos\lambda_r \\ \cos\lambda_r & -\sin\phi_r\sin\lambda_r & \cos\phi_r\sin\lambda_r \\ 0 & \cos\phi_r & \sin\phi_r \end{bmatrix} \cdot \begin{bmatrix} E \\ N \\ U \end{bmatrix} + \begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} \tag{1}$$

Where $X_p,\ Y_p,\ Z_p$ are the ECEF coordinates of the flying object, $X_r,\ Y_r,\ Z_r$ are the ECEF coordinates of the observer and $E,\ N,\ U$ are the local tangent coordinates of the flying object.

$$
\begin{aligned}
&ECEF \;\rightarrow\; Geodetic \\
&\phi = \arctan\left[\frac{Z_p}{p}\left(1 - e^2\frac{N(\phi)}{N(\phi)+h}\right)^{-1}\right] \\
&\lambda = atan2(Y_p,\; X_p) \\
&h = \frac{p}{\cos\phi} - N(\phi)
\end{aligned}
$$

Where:

$$
\begin{aligned}
&N(\phi) = \frac{a}{\sqrt{1 - e^2\sin^2\phi}} \\
&e^2 = 1 - \frac{b^2}{a^2} \\
&p = \sqrt{X^2 + Y^2}
\end{aligned}
$$

Being a the equatorial radius and b the polar radius.

With geodetic coordinates, a set of scattered vertices describing the road is obtained, although the edges have not been defined (Fig. 3). At this stage, the problem is approached as a Traveling Salesman Problem (TSP) [15], where all previously mentioned vertices are connected to each other (except themselves), and the ultimate goal is to find a pseudo-optimal path traversing all vertices of the graph. The TSP problem is formally defined as presented in equation 2.

$$
\begin{aligned}
&\min z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \mid i \neq j \\
&\text{s.t.} \\
&\sum_{i=1, i\neq j}^{n} x_{ij} = 1 \mid j \in \{1, 2, 3, \ldots, n\} \\
&\sum_{j=1, i\neq j}^{n} x_{ij} = 1 \mid i \in \{1, 2, 3, \ldots, n\} \\
&u_i - u_j + nx_{ij} \leq n - 1 \mid 1 \leq i \neq j \leq n \\
&c_{ij} \in \mathbb{R}, x_{ij} \in \{0, 1\}
\end{aligned}
\tag{2}
$$

Where C_{ij} is the distance between vertices computed by the *haversine distance*. Analytical methods exist for solving the TSP problem and finding an optimal solution, such as brute force resolution, which has a time complexity of $\mathcal{O}(n!)$. Methods like the Held-Karp algorithm [16] employ a Dynamic Programming approach, achieving a time complexity of $\mathcal{O}(n^2 2^n)$. Other approaches like the Branch and Bound (B&B) algorithm are also quite popular for providing optimal solutions to the TSP problem.

These optimal algorithms can solve the TSP problem for small instances (up to 20, 40, or 200 vertices at most). Assuming that the resulting graphs from the XODR files will contain a much greater number of vertices than 200, the selection algorithm for planning resolution was the Nearest Neighbor (NN) algorithm [17]. This algorithm offers a suboptimal solution returning a path that is, on average, 25% more costly than the optimal one, and its operation falls within the group of algorithms based on constructive heuristics.

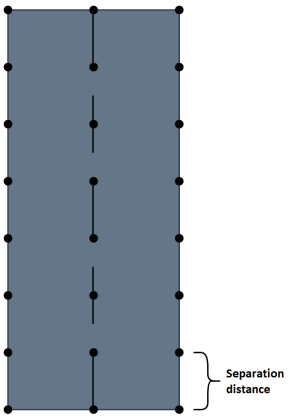

Fig. 3. Vertex-based road representation.

The NN algorithm operates on a simple principle, which involves moving towards the nearest vertex (taking into account edge weights) and repeating the process until all vertices are visited (see Figure 4). The time complexity of the NN algorithm is $\mathcal{O}(n^2)$, and while it remains a significant cost, the algorithm solves the problem (without guaranteeing optimal solutions) in polynomial time. There exists a parameter to be fixed in the generation of road vertices, known as the separation distance. This distance separates vertices within a road segment, such that if the distance is too short, the number of vertices generated from the XODR increases substantially, and vice versa. The effect of this separation distance on planning is significant since excessive separation between vertices of the same segment can result in routes taking on a "*Z-shape*" (see Figure 5). This effect represents a huge problem in terms of energy consumption since the drone needs to manoeuvre too many times, accelerating and decelerating persistently, which reduces the battery life and fligth stability.

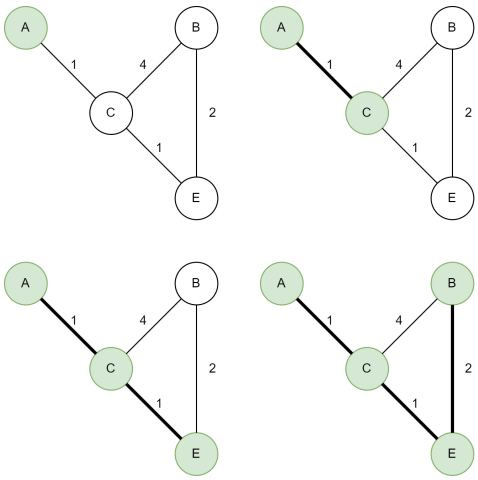

Fig. 4. Simple execution of the NN algorithm.

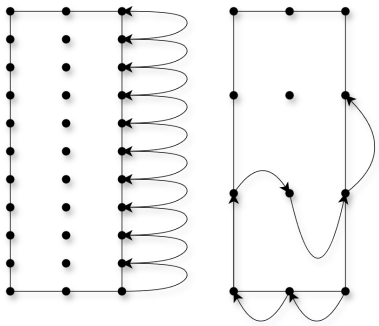

Fig. 5. "*Z-shape*" problem occurs when the separation distance between road segments is larger than lateral distance.

B. Data extraction

The data extraction process occurs during drone mission execution. The flight plan, previously generated using the NN algorithm, delineates the points (specified by their geodetic coordinates) to which the drone must navigate. During the transition between pairs of points, the drone captures images at a specified frequency, facilitating overlap to prevent gaps in road coverage and defect detection.

The drone-mounted camera is affixed to a gimbal articulation system, positioned at a 90-degree angle to the ground's horizontal axis (see Figure 6). The flight altitude can be statically set at the beginning of the mission or can be adapted and modified as necessary to ensure that the drone never flies too low or too high. With each image capture, a combination of an ultrasonic distance sensor and GPS records values for that moment, which are then stored as metadata within the image for subsequent processing. The metadata header structure for the images is illustrated in the table (see Table I).

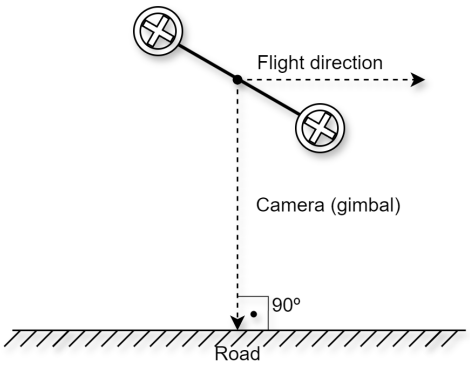

Fig. 6. Gimbal orientation during flight.

C. Processing the data with Computer Vision and fusion

The data processing unfolds across three distinct phases (Fig. 7): detection, fusion, and deployment. Detection is achieved by employing a YOLOv8 neural network [18], trained on road defect images, to yield TXT files containing corresponding bounding-box data of said defects, stored in YOLO format.

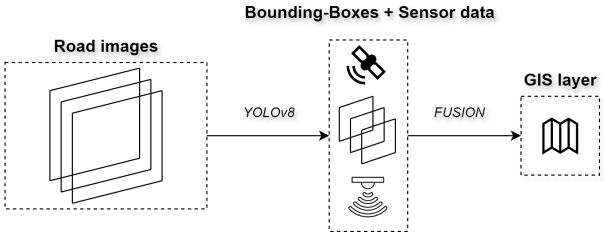

Fig. 7. Data processing phases.

The bounding box denotes a rectangular subset relative to the original image, necessitating additional measurements for its transformation into geodetic coordinates. This is where the data fusion phase comes into play, leveraging the distance sensor and GPS as supplementary sources of information. The GPS records the corresponding coordinates at the moment of image capture, precisely aligning with its center. In other words, the geometric center of the image corresponds to the drone's GPS coordinates at the time of capture. Determining the geodetic coordinates of the image corners necessitates the distance sensor observation, which provides an approximation (accounting for inherent sensor noise) of their locations (see Fig. 8). In Eq. 3 we can observe how the computation of absolute coordinates is done.

$$
\begin{aligned}
&F = 2H \cdot \tan\left(\frac{FOV}{2}\right)\\
&F = 2d \rightarrow d = \frac{F}{2}\\
&\overrightarrow{V_{UL}} = (-d, d)\\
&\overrightarrow{V_{LR}} = (d, -d)\\
&\overrightarrow{V_{UL}}^{(ECEF)} = \begin{cases} X_p = -d(\sin\lambda_r - \sin\phi_r\cos\lambda_r) + X_r \\ Y_p = -d(\cos\lambda_r - \sin\phi_r\sin\lambda_r) + Y_r \\ Z_p = d\cdot\cos\phi_r + Z_r \end{cases}\\
&\overrightarrow{V_{LR}}^{(ECEF)} = \begin{cases} X_p = d(\sin\lambda_r - \sin\phi_r\cos\lambda_r) + X_r \\ Y_p = d(\cos\lambda_r - \sin\phi_r\sin\lambda_r) + Y_r \\ Z_p = -d\cdot\cos\phi_r + Z_r \end{cases}\\
&\overrightarrow{V}^{(GEO)} = \begin{cases} \phi = \arctan\left[\frac{Z_p}{p}\left(1 - e^2\frac{N(\phi)}{N(\phi)+h}\right)^{-1}\right] \\ \lambda = atan2(Y_p,\ X_p) \\ h = \frac{p}{\cos\phi} - N(\phi) \end{cases}
\end{aligned}
\tag{3}
$$

Where F is the footprint (width of the image in meters), H is the relative distance (height) measured in meters by the distance sensor, FOV represents the *field of view* of the camera (in degrees), $\overrightarrow{V_{UL}}$ represents the coordinates of the *upper-left* corner and $\overrightarrow{V_{LR}}$ represents the coordinates of the *lower-right* corner.

TABLE I
METADATA STRUCTURE FOR IMAGES

Field	Description	Space in bytes
Relative distance	Measured by the ultrasonic distance sensor (in meters)	4 bytes (`float`)
GPS - Latitude	Latitude measured in degrees	8 bytes (`double`)
GPS - Latitude (REF)	Precision value. Default: 10^6	4 bytes (`int`)
GPS - Longitude	Longitude measured in degrees	8 bytes (`double`)
GPS - Longitude (REF)	Precision value. Default: 10^6	4 bytes (`int`)

Subsequently, the corners of defect bounding boxes (if any) are computed from their relative coordinates and the updated absolute corner information. The final deployment step involves converting defect information into a rectangular GIS layer, facilitating graphical visualization in geospatial data management software.

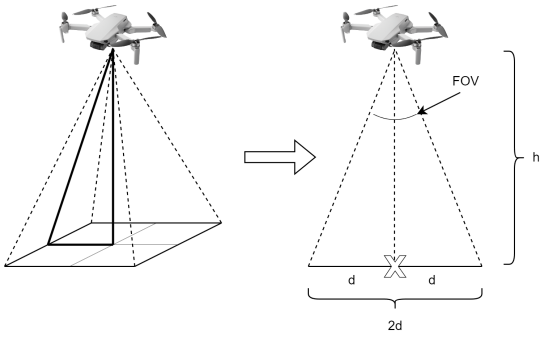

Fig. 8. Drone's geometry when taking images.

Ultimately, the processing integrates data from the camera (including YOLOv8 detection), distance sensor, and GPS to map and incorporate defects as supplementary contextual information.

IV. EXPERIMENTAL RESULTS

For experimentation, AirSim was employed as a realistic flight simulation software, coupled with PX4 as the flight controller. A network of streets interconnected by a main avenue in the vicinity of the Universidad de Alcalá (UAH) in Spain served as the testbed (see Fig. 9). These streets were meticulously replicated using Blender to construct a highly accurate digital twin road.

To establish an extensive test battery, fifty random points uniformly distributed along the roadway were selected to reproduce road defects. Defects were pseudo-randomly generated at these points, and their bounding boxes were stored in a JSON file, serving as ground truth. Mission planning was computed based on the same XODR file used for generating the 3D road model (see Fig. 10), yielding a total of 8514 images.

Image processing steps were executed offline, i.e., subsequent to mission completion, rather than concurrently. Following the detection and fusion phase, results indicate a 100% successful detection rate of defects, this means that every single defect was captured and detected by the system. A

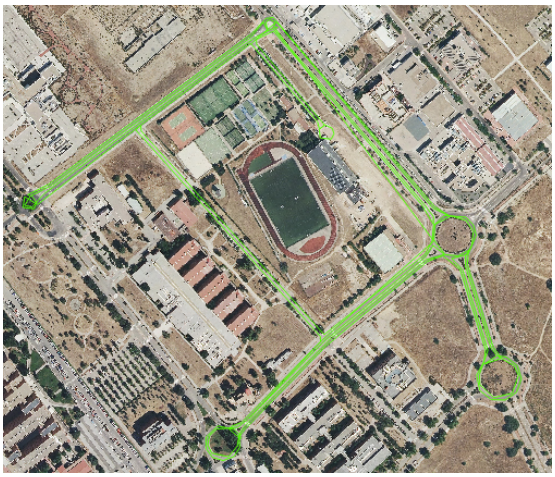

Fig. 9. Resulting mission plan (green) after applying NN algorithm.

Fig. 10. Example of a simulated road crack.

detected defect is considered when at least one predicted bounding box is closer to a particular ground truth defect than to the other defects. Some of those results are shown in Figs. 11, 12, 13 and 14 as illustrative examples. To evaluate the quality of the results, we define a metric named *mean centroid distance* (MCD), mathematically formulated as eq. 4 states:

$$MCD = \frac{1}{N_{pred}} \sum_{i=1}^{N_{pred}} \min_{H} \{H(C_i, C_j)\} \mid j \in N_{GT} \quad (4)$$

Fig. 11. First system output example (green is ground truth and blue are predicted bounding boxes).

Fig. 13. Third system output example (green is ground truth and blue are predicted bounding boxes).

Fig. 12. Second system output example (green is ground truth and blue are predicted bounding boxes).

Fig. 14. Fourth system output example (green is ground truth and blue are predicted bounding boxes).

$$K(C_i, C_j) = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$
$$H(C_i, C_j) = 2r \arcsin\left(\sqrt{K(C_i, C_j)}\right) \tag{5}$$

Where N_{pred} represents the number of predictions made (predicted bounding-boxes), C_i, C_j represent the *i-centroid* and the *j-centroid* of the predicted bounding-box and the ground truth bounding-box respectively, H represents the *haversine distance* computed as eq. 5 shows and N_{GT} represents the number of ground truth bounding-boxes. The metric of evaluation employed herein affords us the opportunity to scrutinize the system's quality, particularly the precision of the final outcomes resultant from fusion. Although distances between points could be calculated using the *euclidean distance* (and Earth's curvature wouldn't be significant), it is important to note that the *haversine distance* operates directly with points represented in global geodetic coordinates, so it is immediate to use it as the preferred distance metric.

The overall result states that the system predicts the position of defects with an average error distance of $\mu = 0.86m$ and a standard deviation of $\sigma = 0.43m$. Plotting these errors into a bar-plot and computing the corresponding normal distribution density function shown in Fig. 15, we can assert that the error is normally distributed which allows us to use $\mu = 0.86m$ as a trustworthy statistical value.

V. Conclusions

The proposed system successfully addresses the task of defect detection for which it was designed, with particular emphasis on the automatic planning and autonomous flight module, which entirely eliminates the need for human drone operators. The utilization of position and distance sensors for mapping and geolocating defects allows for the creation of GIS information layers, thereby enhancing contextual information about roadways. This enables *"second pass"* missions to corroborate or reconstruct defects, yielding more precise information, thereby opening avenues for defect classification, study of the optimal sequence for repairs based on damage severity and distance, among others. While the proposed system is tai-

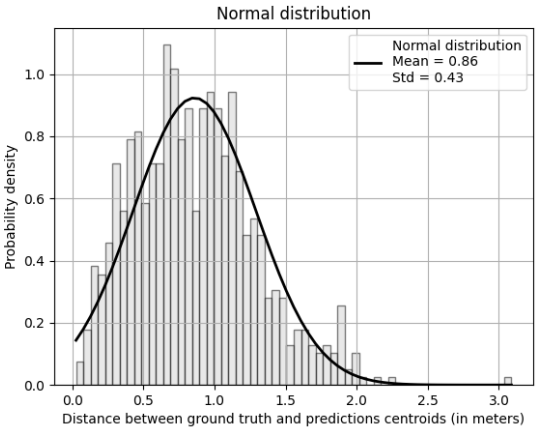

Fig. 15. Error distribution based on the distance that separates centroids between ground truth and prediction bounding-boxes.

lored for monitoring road conditions, its applicability extends to other infrastructures such as sidewalks, airport runways, warehouse rooftops, bridges, dams, and more. In conclusion, this article's proposal harnesses the use of autonomous flights, automatic planning, artificial vision algorithms, and data fusion as key elements in road defect detection.

VI. Future works

As future work, we will consider the modification of the system to enable real-time (online) detections, thereby expanding the data collection system to distributed drone swarms, facilitating quicker road scanning through imagery. The utilization of commercial drones is constrained by battery duration, typically around 30 minutes. An improvement approach would involve planning considering battery consumption, including visits to recharge stations if necessary. While the system's quality in defect detection and geolocation is more than reasonable, enhancements can still be made, particularly regarding noise filtering in GPS and distance sensors. Exploring the possibility of reconstructing defects using LiDAR stands out as a primary enhancement, adding additional layers of contextual information about the defects. Validating the system with real world data (real images of road defects) is also a major improvement that must be achieved in the future.

Acknowledgment

This work was funded by public research projects of the Spanish Ministry of Science and Innovation PID2020-118249RB-C22 and PDC2021-121567-C22 - AEI/10.13039/501100011033 and the project under the call PEICTI 2021-2023 with identifier TED2021-131520B-C22.

References

[1] E. Ivanová and J. Masárová, "IMPORTANCE OF ROAD INFRASTRUCTURE IN THE ECONOMIC DEVELOPMENT AND COMPETITIVENESS," *ECONOMICS AND MANAGEMENT*, vol. 18, no. 2, 2013.

[2] Y. Qiao, A. R. Dawson, T. Parry, and G. W. Flintsch, "Evaluating the effects of climate change on road maintenance intervention strategies and Life-Cycle Costs," *Transportation Research Part D: Transport and Environment*, vol. 41, 2015.

[3] H. T. Nguyen, L. T. Nguyen, A. D. Afanasiev, and L. T. Pham, "Classification of Road Pavement Defects Based on Convolution Neural Network in Keras," *Automatic Control and Computer Sciences*, vol. 56, no. 1, 2022.

[4] R. Ghosh and O. Smadi, "Automated detection and classification of pavement distresses using 3d pavement surface images and deep learning," in *Transportation Research Record*, 2021, vol. 2675, no. 9.

[5] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, "Pavement distress detection and classification based on YOLO network," *International Journal of Pavement Engineering*, vol. 22, no. 13, 2021.

[6] D. K. Dewangan and S. P. Sahu, "Road Detection Using Semantic Segmentation-Based Convolutional Neural Network for Intelligent Vehicle System," in *Lecture Notes on Data Engineering and Communications Technologies*, 2021, vol. 63.

[7] H. Ghandorh, W. Boulila, S. Masood, A. Koubaa, F. Ahmed, and J. Ahmad, "Semantic Segmentation and Edge Detection—Approach to Road Detection in Very High Resolution Satellite Images," *Remote Sensing*, vol. 14, no. 3, 2022.

[8] Z. Cao, X. Xu, B. Hu, and M. Zhou, "Rapid detection of blind roads and crosswalks by using a lightweight semantic segmentation network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, 2021.

[9] D. Nagaraj, M. Mutz, N. George, P. Bansal, and D. Werth, "A Semantic Segmentation Approach for Road Defect Detection and Quantification," in *ACM International Conference Proceeding Series*, 2022.

[10] K. Alkaabi and A. R. El Fawair, "Application of A Drone Camera in Detecting Road Surface Cracks: A UAE Testing Case Study," *Arab World Geographer*, vol. 24, no. 3, 2021.

[11] Y. Jiang, S. Han, and Y. Bai, "Building and Infrastructure Defect Detection and Visualization Using Drone and Deep Learning Technologies," *Journal of Performance of Constructed Facilities*, vol. 35, no. 6, 2021.

[12] I. Katsamenis, N. Bakalos, E. Protopapadakis, E. E. Karolou, G. Kopsiaftis, and A. Voulodimos, "Real time road defect monitoring from UAV visual data sources," in *ACM International Conference Proceeding Series*, 2023.

[13] J. P. Llerena, J. G. Herrero, and J. M. M. Lopez, "Error reduction in autonomous multirotor vision-based landing system with helipad context," in *2022 25th International Conference on Information Fusion, FUSION 2022*, 2022.

[14] M. Althoff, S. Urban, and M. Koschi, "Automatic Conversion of Road Networks from OpenDRIVE to Lanelets," in *Proceedings of the 2018 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI 2018*, 2018.

[15] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization." *The Journal of the Operational Research Society*, vol. 37, no. 5, 1986.

[16] M. Held and R. M. Karp, "A Dynamic Programming Approach to Sequencing Problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, no. 1, 1962.

[17] G. Gutin, A. Yeo, and A. Zverovich, "Traveling salesman should not be greedy: Domination analysis of greedy-type heuristics for the TSP," *Discrete Applied Mathematics*, vol. 117, no. 1-3, 2002.

[18] F. Wan, C. Sun, H. He, G. Lei, L. Xu, and T. Xiao, "YOLO-LRDD: a lightweight method for road damage detection based on improved YOLOv5s," *Eurasip Journal on Advances in Signal Processing*, vol. 2022, no. 1, 2022.