AI-based Road Inspection Framework Using Drones with GPS-less Navigation

Hovannes Kulhandjian[†], Jose Maciel Torres[†], Nicholas Amely[†], Cruz Nieves[†], Christian Reeves[†], and Michel Kulhandjian[‡]

[†]Department of Electrical and Computer Engineering, California State University, Fresno, Fresno, CA 93740, U.S.A.
E-mail: {hkulhandjian, mrmaciman101, namely, cruzer77, sirer12}@mail.fresnostate.edu
[‡]Department of Electrical and Computer Engineering, Rice University, Houston, TX, 77005, U.S.A.
E-mail: michel.kulhandjian@rice.edu

***Abstract*—The inspection of roads is an essential aspect of infrastructure maintenance in our country. Yet, conventional inspection methods often entail significant time and financial investments. Drones present an innovative and superior alternative for inspecting roads, offering swifter, safer, and more cost-efficient solutions. In this paper, we devise and deploy a low-cost framework for the inspection of roads using drones and machine learning. In our approach, we employ both an infrared (IR) camera in tandem with a high-resolution optical camera, as relying solely on optical cameras proves inadequate. While optical cameras excel in surface damage inspection of bridges and roads, IR cameras often yield valuable insights into the underlying structural issues. To enable autonomous drone navigation and the capture of images of the road structure when it identifies potential problems, our drone inspection system is outfitted with a minicomputer running sophisticated artificial intelligence (AI) algorithms. Leveraging these advanced AI algorithms, the drone autonomously performs inspection procedures without human intervention. The outcomes of these experiments demonstrated the system's capability to detect potholes with an average accuracy of 84.6% using the visible light camera and an impressive 95.1% using the IR camera.**

***Index Terms*—Artificial intelligence, machine learning, bridge and road inspection, drones.**

I. Introduction

A developed country's highway network spans thousands of centerline kilometers, comprising asphalt, concrete, or composite pavements that vary in age, condition, and performance. To effectively manage these diverse road systems, various maintenance programs have emerged. These programs aim to monitor ongoing performance, predict future conditions, aid investment planning, and identify necessary rehabilitation and maintenance measures. One such initiative is the Long-Term Pavement Performance Program (LTPP) established by the United States Department of Transportation. This active program focuses on comprehensive data collection, storage, analysis, and product development for both the United States and Canada [1]. Central to the LTPP is the assessment of pavement surface conditions, emphasizing the need for accurate measurements of distresses such as cracks, potholes, and other critical indicators [2].

Presently, pavement image and video data gathered from digital inspection vehicles undergo manual inspection by technicians on computer screens to detect and evaluate defects. However, this process is both time-consuming and expensive, and the outcomes can be subjective, influenced by the raters' experience and perspectives [3]. To overcome these limitations, several research efforts have focused on automating defect detection.

Related Work: Koch and Brilakis in [4], introduce an automated method for detecting potholes in asphalt pavement images. The process begins with segmenting the image into defect and non-defect areas through histogram shape-based thresholding. Morphological thinning and elliptic regression are used to approximate the potential shape of a pothole.

Zhang *et al.* in [5], propose a road crack detection using a deep convolutional neural network. The images are taken with a smartphone and trained on the neural network. This study does not utilize a drone or an infrared camera to automate the process and enhance fault detection.

Azhar *et al.* in [6] examine various images from asphalt pavement, focusing on potholes and non-pothole scenarios. Utilizing the appearance and shape characteristics of potholes, Histograms of Oriented Gradients (HOG) features are calculated from the input images. These features are trained and classified using a Naïve Bayes classifier to distinguish between pothole and non-pothole images. To pinpoint pothole locations within the detected images, a normalized graph cut segmentation method is applied.

Ouma and Hahn in [7] introduce a 2D vision-based method for spotting potholes on urban asphalt roads. It combines multiscale texture filtering with wavelet transform for texton representation, integrating it into superpixel clustering using the fuzzy c-means (FCM) algorithm. The process identifies pavement defects and non-defects. To pinpoint pothole boundaries, the method employs fine segmentation based on morphological reconstruction, refining and outlining the detected potholes.

Hassan *et al.* in [8], utilize deep neural networks to categorize roads according to their level of degradation. Again in this work, they do not employ a drone or an infrared (IR) camera to automate the process and improve fault detection.

In all the aforementioned studies, advanced machine learning algorithms were either not investigated, or only one type of camera, typically an optical camera, was employed, or the system was not automated using drones. Additionally, these studies that utilized drones often necessitated the presence of a specialist to operate the drone. Furthermore, none of the discussed studies offered a global positioning system (GPS) -less navigation for drones.

In contrast, our drone inspection system is equipped with a minicomputer that runs machine learning algorithms, en-

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

abling autonomous drone navigation and on-the-fly image capture of the road structure. Whenever it detects any damage, it can save the location information of that damage. Instead of relying on human operation, it can self-operate and carry out the inspection process independently, thanks to the advanced artificial intelligence (AI) algorithms we have developed.

Our proposed research, as detailed below, stands poised to significantly enhance future road inspection practices. In this study, we develop a cutting-edge hardware and software framework, rooted in AI, tailored for road inspections using drones equipped with multiple sensors. Recognizing the limitations of relying solely on cameras for inspections, we have incorporated an IR camera in conjunction with a high-resolution optical camera. Notably, the IR camera often provides more comprehensive insights into the interior structural condition of a road, complementing the capabilities of optical cameras primarily suited for surface damage assessment.

Therefore, we believe that the integration of an infrared camera along with a visible camera can greatly enhance the performance of fault detection on roads.

II. System Overview

The AI-based road inspection system seamlessly integrates advanced software, a machine learning neural network, and hardware subsystems, all operating in harmony to fulfill its designated task.

In the software subsystem, we leverage software development kit (SDK) libraries to establish connections with two distinct control setups. These libraries facilitate communication between the drone's flight controller and the companion computer, enabling efficient preflight checks, in-flight operations, and data transmission. Once the SDK libraries have successfully established communication between the companion computer and the control setups, the drone's movement is initiated. The software's initial step is to instruct the drone to ascend to predefined global GPS waypoints. While traversing these waypoints, the drone continuously captures images of the road terrain below. These real-time images are subjected to immediate analysis by the machine learning algorithms. If faults or anomalies are detected, the drone momentarily deviates from its waypoint path, descending in altitude to approach the identified fault area. Close-up images of the fault are then captured by both cameras and transmitted to the ground station for storage. Afterward, the drone resumes its designated flight path, continuing to capture images. Upon reaching the final waypoint, the ongoing mission is marked as complete, and the drone autonomously returns to its launch waypoint.

The machine learning subsystem encompasses several essential preflight processes. Initially, the model dataset is uploaded and consolidated to facilitate the training of the neural network. Once the model undergoes its initial training phase, it becomes operational for in-flight use, actively detecting faults (e.g., cracks, patrols) in real-time images obtained by the software subsystem.

III. Drone Framework Design

Our system is constructed around a Hexacopter 6-axle Aircraft Kit, featuring an HMF S550 Frame, PXI PX4 Flight Control, 920KV Motors, a GPS unit, and an AT9 Transmitter. The ultimate drone prototype, as displayed in Fig. 1, encompasses all peripherals, including six propellers, a GPS module equipped with a built-in compass, and a GPS antenna mount, along with the PXI PX4 flight controller.

Fig. 1: The final drone prototype.

Our S550 Hexacopter achieves seamless wireless communication with the ground station through telemetry transmitters and receivers. The Holybro 915 MHz radios were preferred over generic 5.8 GHz radios due to their longer communication range, reduced bandwidth usage, lower power consumption, and decreased susceptibility to interference from other devices, in contrast to the 5.8 GHz radios.

The chosen radios are directly connected to the drone's flight controller, the Pixhawk. The Pixhawk boasts an intuitive interface that empowers users to fine-tune settings, configure parameters, and establish specific flight directives for the Pixhawk to execute during flight operations.

Additionally, the drone is equipped with a companion computer, specifically the Raspberry Pi 4, which plays a central role in executing the project's core functions.

IV. Software Architecture

Our system is predominantly constructed upon the ROS2 (Robot Operating System) system architecture. ROS is a comprehensive collection of software libraries and tools designed for the creation of robotic applications. These libraries and algorithm implementations are at the forefront of technology and are widely adopted in the industry. ROS primarily employs a variation of the publisher-subscriber model referred to as 'nodes' and 'topics'. The ROS setup incorporated into the project is segmented into three distinct nodes running on the Raspberry Pi: the Imaging Node, Road Nav Node, and Defect Classification Node.

Imaging Node: This node interfaces physical cameras with the system, processing images via OpenCV and publishing image topics in "sensor msg" format from an OpenCV Mat type. It manages all image and video data from onboard drone sensors, publishing "thermal st" and "visible light" topics corresponding to thermal and visible light camera feeds, collected through driver scripts and SDK libraries within the Imaging Node.

Road Nav Node: This node handles motion planning and data collection, subscribing to image topics to perform edge detection for autonomous road navigation between predefined GPS start and endpoints. It also conducts periodic data collection, processing information from the drone's Inertial Measurement Unit (IMU) and location sensors. Subscribing to “imu pub” for IMU data, “global position” for GPS coordinates, and ‘flagged’ from the Defect Classification Node for crack detection, the Road Nav Node merges this data to control the drone’s movements.

Defect Classification Node: The Defect Classification Node uses machine learning models to spot image defects from subscribed image topics (“thermal st” and “visible light”) published by the Imaging Node. It processes these images through neural networks and scripts, publishing a ‘flagged’ topic containing all identified defect images.

The drone’s flight controller utilizes a square-root PID controller, ensuring swift and accurate stabilization. This configuration minimizes overcompensation and allows precise tuning of PIDs, reducing errors and mid-flight oscillations. ArduPilot software simplifies controller value adjustments, enabling easy optimization for stable flight performance.

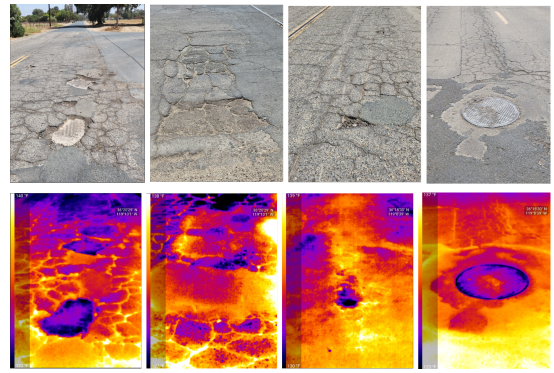

Fig. 2: Sample images of potholes and a manhole cover gathered with the RGB (top) and thermal (bottom) cameras.

V. Machine Learning

Two parallel machine learning models were implemented for road fault detection, each employing a distinct approach. The first method involves the use of an infrared camera in conjunction with a visible light camera to heighten the likelihood of fault detection. This combination is advantageous because the two cameras capture different types of information. While both methods rely on visual inspection, the infrared camera identifies variations in heat across the roadway’s surface. Faults typically exhibit different temperatures compared to the surrounding asphalt, making them stand out in the thermal images.

We have employed two distinct model types: a deep neural network followed by a region-based convolutional neural network (RCNN). The classification deep neural network operates onboard the drone and is responsible for rapidly analyzing images and categorizing them into predefined groups. These models can classify an entire image into specific categories but cannot localize and identify objects within the image. In contrast, the region-based convolutional neural network can perform tasks such as object localization, identification, classification, and bounding within the image itself. The classification deep neural networks excel in expeditiously completing image analysis, allowing them to conduct a preliminary assessment of photos and organize data for the region-based convolutional neural network. This prioritizes images classified as “faulty roadways” for in-depth analysis.

The machine learning component of this project was designed to operate both in conjunction with the drone and independently. The creation of smaller classification models served the initial purpose of organizing the photos captured by the drone. These classification deep neural networks were specifically designed to categorize the photos into two distinct groups: “faulty roadways” or “acceptable roadways”. The dataset comprising images captured through both IR and visible cameras included approximately 300 instances of “faulty roadways” and another 300 instances of “acceptable roadways”. The “faulty roadways” images encompassed various road issues such as potholes, manhole covers, and extensively deteriorated roads in need of repair. The images labeled as “acceptable roadways” depicted well-prepared roads with minimal asphalt cracks. To increase the dataset size fourfold, data augmentation techniques were employed, resulting in 1200 images for each category within both types of images. Additionally, a feature was integrated into the model to flag and save photos classified as “faulty roadways,” prioritizing them for subsequent analysis. Figure 2 displays sample images of potholes and a manhole cover obtained from the visible light camera and the thermal camera, which serve as the training data for the neural network.

The Faster R-CNN models were responsible for analyzing each captured photo, starting with those flagged as faulty. These models were developed using MATLAB, chosen for its extensive collection of toolboxes and comprehensive documentation for implementing deep learning models. Modifications to the classification of deep neural networks were carried out using the Deep Network Designer interface. The region-based convolutional neural network was constructed through MATLAB code, and it operates with images of uniform dimensions. For the creation of MATLAB data stores necessary for the region-based convolutional neural network, MATLAB’s Image Labeler application was utilized for annotation and dataset generation.

VI. GPS-less Navigation using Road Edge Detection

We investigated multiple algorithms aimed at guiding the drone through streets for road inspection without relying on GPS. These algorithms were intended to help the drone scan and inspect pathways for defects, especially in areas like street corners where GPS accuracy might be limited.

We utilized various computer vision algorithms in OpenCV to detect and locate road edge lines in the camera frame. Our approach centered on Canny edge detection, a multistage algorithm identifying changes in color gradient intensity. It uses non-maximum suppression to find local maxima in gradient contrast pixels, followed by hysteresis thresholding to determine edge pixels within a specified range.

Figure 3 provides a visual representation of Canny edge detection applied to a roadway. The image on the left represents the original frame, while the modified version is displayed on the right.

The algorithm’s hysteresis thresholding offers adaptability by setting thresholds based on each frame’s unique edge spread, effectively identifying gradient variations in asphalt

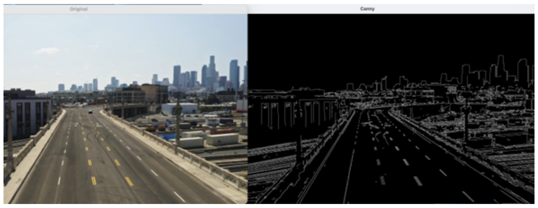

Fig. 3: Canny edge detection example on a roadway image original (left) and modified (right).

Fig. 4: Hough transform performed on a roadway image original and modified.

Fig. 5: Color masking and Hough transform performed on a roadway image.

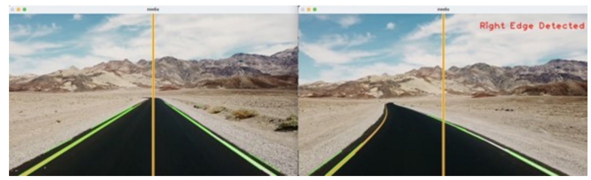

Fig. 6: Detecting road edge lines for altitude control.

and road edges. To fortify road detection, we utilized the Hough transform, detecting straight lines in images by representing them as sine curves. Intersection points between these curves define lines. Through user-defined thresholding, we isolated lines exceeding a predefined length, achieving accurate detection of painted road edges

Figure 4 provides a visual representation of the Hough transform in action on a roadway. The image on the left depicts the original frame, while the modified version is displayed on the right.

We used color masking as an alternative to traditional edge detection, particularly effective with blurry or light-colored roadways. This method isolates defined hue, saturation, and value (HSV)-based colors from the image frames, effectively distinguishing painted road edges. Subsequently, we applied the Hough transform to detect these isolated colors as lines, further refining line detection accuracy.

Figure 5 provides a visual representation of the color masking process along with the Hough transform in action on a roadway. The left-side image shows the original frame, the middle image displays the result after color masking and the Hough transform, and the right-side image incorporates the identified green lines for navigation.

The computer vision techniques we employed not only helped detect road edge lines in image frames but also provided critical x-axis and y-axis coordinates for the start and end points of each detected line, relative to the image frame's size in pixels.

To ascertain if the drone had crossed a detected road edge line during flight, we compared the x-coordinates of these lines to the x-coordinate value of the vertical asymptote positioned precisely in the middle of the frame. If the x-coordinates of the lines exceeded or fell below the middle x-coordinate, depending on the line's starting side, it signified a respective left or right edge. This edge information was then relayed to a MAVROS program, a ROS package facilitating drone communication.

The MAVROS program utilized this edge information to adjust the drone's attitude control. If an edge was detected by the computer vision system, the MAVROS program issued commands to the drone to halt its movement along the detected edge. This setup empowered the drone to achieve fully autonomous flight, navigating both straight and curved roadways without reliance on GPS. Figure 6 shows the process of detecting road edge lines, which is used for altitude control of the drone.

VII. Drone Prototype Experimentation

In the initial stages of development, the S550 Hexacopter faced significant challenges in achieving successful flights. These challenges were primarily attributed to the drone's tuning parameters and ESC, which caused desynchronization from the flight controller's initial configuration. To address these issues, the drone underwent structural reconfigurations, and the PID tuning was meticulously adjusted. These improvements were pivotal in enhancing the drone's stability and enabling it to execute consecutive successful flights. Figure 7 illustrates the drone experimentation conducted on a street in Fresno State.

Data collected during flights was securely transmitted to the ground station via a Secure Shell (SSH) protocol. This data served as the foundation for training and testing our anomaly detection models, allowing us to assess their effectiveness in identifying discrepancies within the scanned roadway structure.

The development of the imaging, road navigation, and defect classification node classes was facilitated through the ROS2 application programming interface (API). These classes were structured to inherit attributes from the parent ROS2 node template class, which introduced layers of abstraction, streamlining the development process.

To interface effectively with the physical sensors, the imaging node made use of two distinct libraries: OpenCV and libseek-thermal. OpenCV, an open-source image processing library, and libseek-thermal, an open-source device driver library, played key roles in sensor management. For both the visible light and infrared sensors, the approach involved opening these devices as OpenCV VideoCapture objects.

Fig. 7: Experiments conducted to detect faults in the road using the prototype quadcopter drone.

This VideoCapture object is an integral part of OpenCV, designed for capturing sequences of frames (video) from various sources. Each frame extracted from the VideoCapture object was represented as an OpenCV Mat type, essentially a matrix-based representation of pixel data.

The VideoCapture objects were kept continuously open throughout the node's lifecycle, avoiding repeated opening and closing for each frame. This optimization notably reduced computational load by eliminating the need to free buffers for the constrained 9 Hz infrared speed. Devices were opened in the class constructor and released in the destructor.

We achieved successful drone construction and calibration, seamlessly integrating essential components into the system. Both thermal and optical cameras were securely mounted on the drone, and we developed robust software to ensure their flawless operation.

Furthermore, the machine learning component of the project excelled in constructing various classification deep neural networks, as well as the necessary region-based convolutional neural networks. These models exhibit an impressive level of accuracy, as evidenced by the training dataset results outlined in the machine learning experimental section below.

All models developed for this project showcased notable levels of accuracy. 80% of the data set was used for training and the rest for validation. The optical image deep neural network achieved a commendable validation accuracy of 84.6%, while its thermal imaging counterpart surpassed expectations with a validation accuracy of 95.1%. The optical faster-RCNN model demonstrated exceptional performance with a mini-batch accuracy of 99.5%, and the thermal faster-RCNN model closely followed with a mini-batch accuracy of 98.9%. Figures 8 and 9 represent the training and validation results of the classification deep neural network for optical and thermal images, respectively.

VIII. Conclusion

In this work, we have introduced an AI-powered framework for inspecting roads utilizing drone technology. Our pioneering approach includes the development of GPS-independent navigation algorithms, specifically designed for road edge line detection. While our initial data collection focused on roads, employing both visible light and thermal infrared cameras, we successfully trained Machine Learning models to revolutionize road inspection by autonomously identifying areas in need of repair. Our experimental findings highlight a remarkable defect detection accuracy of over 95%. While this marks a significant achievement, our future endeavors will extend this innovative approach to bridge inspection, further advancing the field.

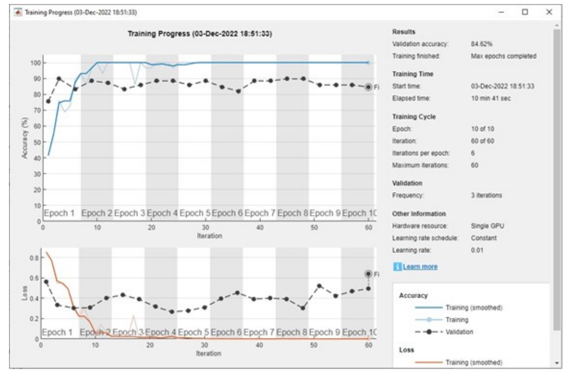

Fig. 8: Training and validation results of the classification deep neural network for optical images.

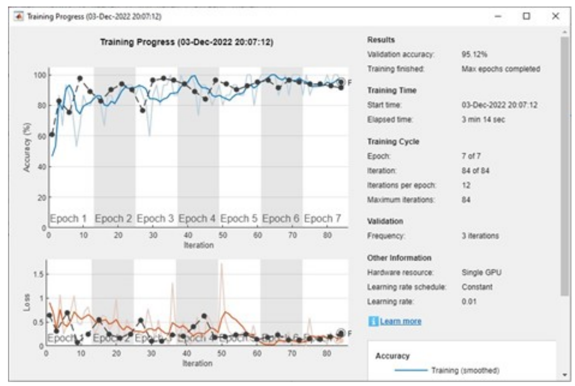

Fig. 9: Training and validation results of the classification deep neural network for thermal images.

Acknowledgment

The work was partially supported by the Fresno State Transportation Institute (FSTI).

References

[1] F. H. A. (FHwA), "Ltpp beyond fy 2009: What needs to be done?" *Technical Report No. FHWA-HRT-09-052*, 2009.

[2] J. Daleiden *et al.*, "Variability of pavement distress data from manual surveys," in *Pavement Evaluation Conference 2014*, 2014.

[3] A. Bianchini, P. Bandini, and D. W. Smith, "Interrater reliability of manual pavement distress evaluations," *Journal of Transportation Engineering*, vol. 136, no. 2, pp. 165–172, 2010.

[4] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Advanced engineering informatics*, vol. 25, no. 3, pp. 507–515, 2011.

[5] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3708–3712.

[6] K. Azhar, F. Murtaza, M. H. Yousaf, and H. A. Habib, "Computer vision based detection and localization of potholes in asphalt pavement images," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2016, pp. 1–5.

[7] Y. O. Ouma and M. Hahn, "Pothole detection on asphalt pavements from 2d-colour pothole images using fuzzy c-means clustering and morphological reconstruction," *Automation in Construction*, vol. 83, pp. 196–211, 2017.

[8] M. U. Hassan, O.-M. H. Steinnes, E. G. Gustafsson, S. Løken, and I. A. Hameed, "Predictive maintenance of norwegian road network using deep learning models," *Sensors*, vol. 23, no. 6, p. 2935, 2023.