Engineering Applications of Artificial Intelligence 122 (2023) 106143

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

AI-powered health monitoring of anode baking furnace pits in aluminum production using autonomous drones

Tasnim Basmaji [a], Maha Yaghi [a], Marah Alhalabi [a], Abdallah Rashed [a], Huma Zia [a], Mohamed Mahmoud [b], Pragasan Palavar [b], Sara Alkhadhar [b], Halima Alhmoudi [b], Mohammad Alkhedher [c], Ayman Elbaz [d], Mohammed Ghazal [a,*]

[a] Electrical, Computer, and Biomedical Engineering Department, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates
[b] Emirates Global Aluminum, Technology Development and Transfer Midstream, United Arab Emirates
[c] Mechanical Engineering Department, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates
[d] Bioengineering Department, University of Louisville, Louisville, USA

ARTICLE INFO

Keywords:
Drones
Artificial intelligence applications
Automated industrial monitoring
Predictive maintenance
Semantic segmentation
Object detection
3D modeling
Multi-modal mosaicking

ABSTRACT

Industrial health monitoring in factories is essential for quality assurance, energy and cost reduction, and health and safety. In aluminum factories, anode furnace pits' flue walls deform over time due to cyclic heating and cooling. They are inspected and classified using manually acquired measurements in a process that takes several hours and is done under high temperatures using specialized equipment. We propose an end-to-end AI-powered system for automated inspection using drones. We fly a drone carrying color and depth cameras to film and navigate a 50-pit furnace floor autonomously in a GPS-denied environment. We then mosaic the recorded videos to produce color and depth mosaics using frame-to-frame motion parameters estimated using the color videos and applied to both. We finish the mosaics using depth-to-color mosaic registration based on maximizing mutual information on gradients. We extract pit images from the mosaics using a YOLOv5 object detection initially trained using a physical floor model with a proposed data augmentation scheme and then fine-tuned for the on-site environment. We achieve a mean average precision of 94.7%. Once pit images are detected and initially classified, we propose a dual-stage segmentation algorithm using the Hough transform and a semantic segmentation network trained using probabilistic feature images with a precision of 96.67%. Segmented pits with depth information allow us to produce 3D models of pits to aid in temporal monitoring and diagnosis confirmation. Our system is cost-effective and reduces inspection downtime by 87%, eliminating the need for human intervention.

1. Introduction

Industrial systems and infrastructure are monitored and maintained to meet production and operational efficiency goals safely. Process engineers and technical workers oversee critical processes while facing potential threats. They detect faults in the structure or equipment and report the need for maintenance before manufacturing is affected (Tkac et al., 2016).

In aluminum factories, anodes are produced in anode-baking furnaces composed of continuously monitored preheating, firing, and cooling sections for quality assurance. Flue walls are where flue-gas flows and combustion occurs. They deform over time due to cyclic heating and cooling. The deformation results in non-homogeneous baking of the carbon anodes, deterioration of anode quality, and over-consumption of energy (Zaidani et al., 2018). Monitoring the health of flue walls for maintenance is critical to prolonging their service life until a replacement is procured (Tews et al., 2013). The current process is manual, time-consuming, and poses health and safety concerns. It involves taking tape measurements under high temperatures while lowering surveyors into the pits using specialized equipment. Limited research has been conducted on automating the health monitoring of anode baking furnaces by assessing the deformation in flue walls. Advances in Unmanned Aerial Vehicles (UAVs), Computer Vision (CV), and Artificial Intelligence (AI) offer a gateway to achieving this goal.

The market for drone technology has grown significantly from 0.85 billion USD in 2015 (goldsteinresearch, 2023) to 8.15 billion USD in 2022 (fortunebusinessinsights, 2023). UAVs and AI are increasingly proposed in engineering applications, industrial inspection, quality monitoring, and labor safety (Yao et al., 2019). Surface inspection

* Corresponding author.
E-mail address: mohammed.ghazal@adu.ac.ae (M. Ghazal).

https://doi.org/10.1016/j.engappai.2023.106143
Received 17 November 2022; Received in revised form 21 January 2023; Accepted 9 March 2023
Available online 14 March 2023

using drones has been the focus of much research lately. Early studies primarily focused on assessing simple, linear infrastructure such as highways, roads, and canals (Rathinam et al., 2008; Metni and Hamel, 2007). The focus at the time was low-level control rules for quasi-stationary flights above planar surfaces, using data from the onboard visual systems (Hoang et al., 2020). Recently, drone designers improved control, navigation, and maneuverability, allowing drones to operate autonomously without a clear line of sight. Drone-based monitoring applications have been proposed, ranging from public area surveillance to infrastructure inspection, intelligent farming, and aerial mapping (Koppány and Lucian, 2015). Assessing building façades and other surfaces to evaluate their integrity is one of the most frequent application domains. The primary goal of such applications is to design flight plans that allow for proper data acquisition. The authors in Baiocchi et al. (2013) use quadrotors in a post-seismic environment for inspecting historic buildings for cracks and other damages. They developed a GPS-based path planner to optimize flight paths and reduce redundant acquisition. They also proposed a processing software for the 3D reconstruction of building façades from pairs of images. The work in Eschmann et al. (2012) presents a similar application but with manual flight control. In Wang and Zhang (2017), Haar-like features and a cascading classifier were used to locate cracks on wind turbine blade surfaces using aerial, drone-based photos. In Morgenthal and Hallermann (2014), the authors evaluated the quality and feasibility of drone-based defect detection photos and discussed methods to improve image quality. One of the challenges of working in factory settings is drone localization and navigation. GPS is widely used to localize and plan a mission and is thus required in most of the proposed inspection approaches. Nevertheless, there are some situations where GPS is limited or unavailable, such as in urban and natural canyons or inside indoor industrial facilities. The authors in Nikolic et al. (2013) proposed a drone-based power plant boiler inspection system through an automated trajectory-following system and a personalized sensor suite for visual navigation. The system is intended to inspect the internal walls of boilers, extending the quadrotor's navigation capabilities with visual navigation to work in areas where GPS is unavailable. Not only must the drone localize itself and navigate as planned, but it must also avoid obstacles and withstand high temperatures and magnetic interference. Furthermore, the type of sensors and algorithms used for localization and path planning is constrained by the environment in which the drone operates (Hoang et al., 2020). Authors in Alireza et al. (2015) propose a simulated quad-copter drone system with inertial and sonar sensors that can fly autonomously in a GPS-denied indoor environment with obstacles.

Aerial drone inspections over a large area or while regulating altitude generate many images, raising the need for mosaicking techniques to produce a complete depiction of the area (Botterill et al., 2010). Considerable research effort has been invested in increasing the robustness of mosaicking to illumination variation (Xu and Mulligan, 2010; Yang et al., 2009), exposure differences (Doutre and Nasiopoulos, 2009), lens distortion (Jin, 2008; Ju and Kang, 2010), improvement of the computational efficiency (Civera et al., 2009; Lovegrove and Davison, 2010; Pulli et al., 2010), and other challenges in static light images. Researchers also attempted depth-light or (depth-color) image mosaicking to overcome the difficulties of parallax and object motion in dynamic light images (Zhi and Cooperstock, 2012; Shen et al., 2018; Zhi and Cooperstock, 2008). However, depth mosaicking is still an open research area (Ding et al., 2019). Cylindrical or spherical projection is usually chosen to generate a depth mosaic with little distortion and a large field of view. Each input image is warped to a cylindrical or spherical plane according to an estimated 3×3 camera matrix or homography (Szeliski, 2006). This problem is well addressed by the authors in Brown and Lowe (2007), in which the camera matrix was estimated and refined based on matched Scale-Invariant Feature Transform (SIFT) features between input light images. Since depth images lack SIFT features, this estimation method is less applicable to depth image registration. Authors in Li et al. (2017) align depth and light images and use the registration matrix of color images to register depth images but require the intrinsic camera parameters of both cameras to work.

After mosaics of large inspection areas are created, objects of interest must be identified and localized (or detected). Object detection algorithms used in images collected during industrial inspections are mainly designed based on the requirements of the application task. The work in Li et al. (2023) proposed a deep learning-based novel automatic defect detection solution, You Only Look Once (YOLO)-attention, based on YOLOv4. Their solution was to achieve a fully automated, fast, and accurate defect detection system for Wire and Arc Additive Manufacturing (WAAM) production. Another YOLO-based real-time method was proposed in Hayat and Morgado-Dias (2022) for safety helmet detection at a construction site. The system showed excellent results in detecting safety helmets, achieving a mean Average Precision (mAP) of 92.44%, even in low-light conditions. Another example is the work in Saeed et al. (2021), where the authors proposed a robust approach for industrial small-object detection using an improved faster Regional Convolutional Neural Network (Fast RCNN) to detect faults in the product images.

Once monitored objects of interest are detected, further analysis and processing (e.g., classification or measurement) can be performed to achieve the monitoring goals. The analysis typically begins by segmenting and modeling the objects. Segmentation and 3D modeling have multiple applications in industrial inspections since they provide information relevant to the height or depth that are not accessible in 2D images (Hartung et al., 2022). The authors in Jayawardena et al. (2012) proposed a model-assisted segmentation method that registers a 3D model of the object over a given image by optimizing a novel gradient-based loss function, obtaining the full 3D pose from an image of the object. Another approach is presented in Cheng et al. (2020), where the authors proposed a transfer learning approach to learn boundary-aware material segmentation, with only a limited number of fine pixel-wise annotations required. Their approach is integrated with a Simultaneous Localization And Mapping (SLAM) system to fuse the per-frame understanding delicately into a 3D global semantic map. Fully autonomous industrial indoor inspection needs integrated solutions. Currently available sensor-based industrial health monitoring approaches are high-costly to cover large industrial areas and must deal with conditions that affect the electronics and communication capabilities of the sensing and computing units (Ghazal et al., 2020).

In this research, we propose an autonomous AI-powered indoor quad-copter drone system for flue wall deformation monitoring, assessment, and reporting in carbon baking furnaces. Our inspection system involves real-time cooperation of modular subsystems to inspect the furnace and upload recorded videos and location information to a cloud server for 3D modeling, fine localization, and further analysis of the flue walls, including pit classification and deformation assessment. The proposed drone records low-altitude visual and depth aerial videos that are then processed to create mosaics of the baking furnace. We use AI-based approaches to isolate the pits from the background and segment the top and bottom layers to measure the deformation rate. The measurements are checked against predefined criteria to classify each of the pits into one of three classes, i.e., normal, narrow, or wide, following the criteria set by the production factory within a short response time. The collected data is reported to a centralized repository to facilitate the access of system operators on and off-site. The large-scale installation of custom-designed communication solutions is not feasible, and the drone is required to fly in a magnetic interference area.

Timely monitoring flue walls in carbon baking furnaces is critical for establishing a reliable maintenance schedule. Regular maintenance prevents the pits from deforming so much that production quality deteriorates. It ensures the production of high-quality baked anodes, keeps energy consumption low, reduces environmental emissions, and reduces production costs. Currently, pits health is monitored through

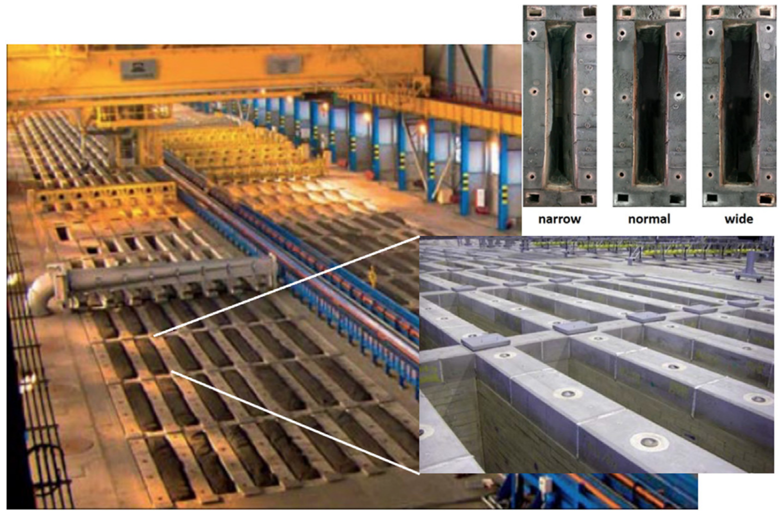

Fig. 1. Industrial environment characterized by an ambient temperature up to 80 °C, dust, low-light, high-magnetic interference, and 4.6 m deep pits.

surveys conducted by human operators using measuring devices at an ambient temperature that could reach as high as 80 °C (Arkhipov et al., 2016). The measurements are checked against predefined criteria for pit classification into one of three classes, i.e., normal, narrow, or wide. Pits with a width of 760–840 mm are normal, <730 mm are narrow, and > 900 mm are wide. Our major global aluminum production factory's industrial partner designed and uses these criteria. Additionally, manual assessment requires operator training, specialized equipment, and operator endurance in these tough settings (Zhou et al., 2014). While safety measures are in place to protect operators, it remains a significant concern. Another concern is the slow inspection process delaying production. Regular inspection is typically carried out once every month and causes downtimes. The authors were approached to develop automated systems for assessing and tracking the temporal development for each pit to help process engineers respond timely. Fig. 1 shows the operating environment and the desired classification output. It is also important to mention that flagging abnormal pits is only an initial output. 3D pit models that allow for digital measurements are then needed to confirm the diagnosis and support decision-making.

While recent technological advancement can help automate and speed up the process, the harsh nature of the industrial environment imposes a new array of challenges. These challenges include high temperature's impact on materials and electronics, magnetic interference effect on drone control and communications, poor illumination effects on cameras, and the absence of GPS signals effect on navigation. Researchers aimed to address these challenges by novel materials, indoor navigation systems, computer vision algorithms, and machine learning techniques Tews et al. (2013), Kroll et al. (2014), Kubota and Yamamoto (2019).

The main contribution of our work is an end-to-end system for the autonomous monitoring and assessment of critical equipment health in industrial settings, reducing surveying time from 4 h, reported by our industrial collaborator, to 30 min per section while providing a classification accuracy of 95%. To achieve this goal, we also propose: (1) a data augmentation scheme to increase the training data size for initial pit classification; (2) a vision-based top-layer segmentation algorithm and a deep learning-based bottom-layer semantic segmentation algorithm; (3) a multi-modal light-depth mosaicking technique for floor map generation; (4) integrated algorithms to produce 3D pit models that end-users can use to take measurements and confirm the initial system classifications; and (5) a visualization and temporal assessment user-friendly interface, to eliminate or reduce the need for human intervention.

The remainder of the paper is organized as follows: Section 2 introduces the materials and methods. Section 3 discusses the testing and validation results of our proposed system and subsystems. Finally, conclusions are drawn in Section 4.

2. Materials and methods

2.1. Overall system architecture

We propose an AI-powered system consisting of an autonomous indoor drone video-surveying the area using dual cameras, a visual, and a depth camera. We use a multi-modal mosaicking algorithm to turn the videos into aerial views of the entire floor with all pits. We then isolate the pits from the background using a YOLO object detector (Jocher, 2022). We need many images to train the object detector, while there are only three floors with hardly sufficient variation in the data for deep learning. For this purpose, we propose a data-augmentation technique that introduces realistic, industry-verified, deformed, and normal (healthy) pit models along with their ground truth locations and labels. We match our industrial partner criteria for classifying pits. Once we detect pits, we extract their depth and visual images. The top and bottom of the pits are segmented using two proposed segmentation pipelines. The pit-top segmentation technique detects peepholes and uses them to identify wall pixels. Wall pixels are then used to estimate the parameters of a Gaussian Mixture Model (GMM) (Stauffer and Grimson, 2007). Edge detection and the Hough transform are used to find the model of the top layer. The bottom layer uses a semantic segmentation approach which trains a DeepLabV3+ architecture (Chen et al., 2018) using probability maps. Depth data from the depth camera helps create a gradient on the remaining visible part of the wall. The occluded portion of the wall, which our model does not capture, does not affect the classification of the pits conditions. The 3D model of the pits and the classification label from the object detector are presented to end users, allowing them to take estimated digital measurements to confirm the diagnosis and order maintenance. Taking digital distance measurements requires calibrating the drone's camera to find its extrinsic and intrinsic parameters to transform image coordinates in pixels to real-world coordinates in meters. The overall system diagram is shown in Fig. 2 with subsystems numbered chronologically. We discuss each of the main sub-systems in the subsequent sections.

The central unit of the overall system is a multi-rotor drone programmed to navigate indoor GPS-denied areas autonomously. The drone integrates all sensors needed for navigation, positioning, and inspection. To make creating, maintaining, and updating the system easier, we designed it using a layered architecture shown in Fig. 3. Communication between all sub-systems uses a publish–subscribe pattern. The autonomous indoor flight control system utilizes a computer

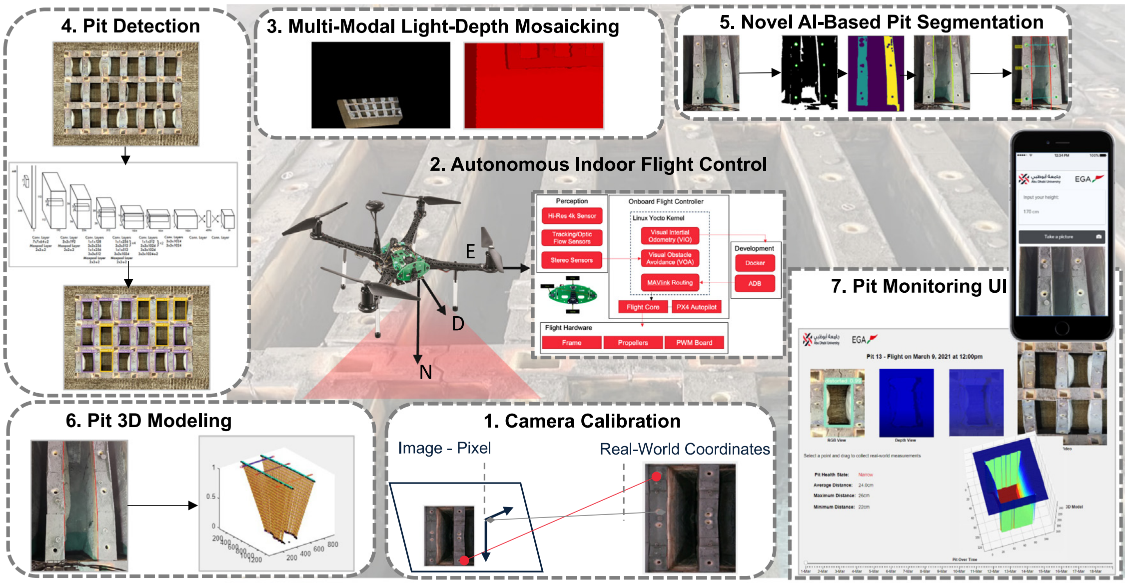

Fig. 2. Overall System Overview Diagram.

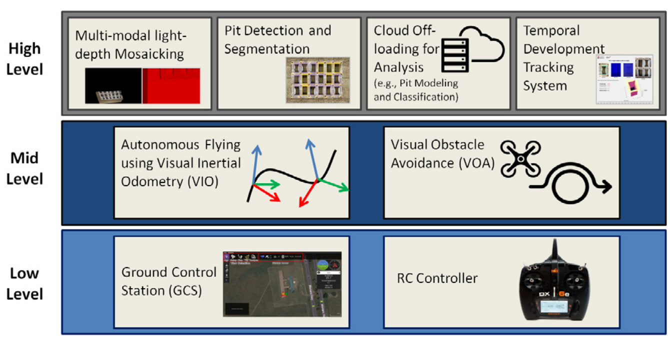

Fig. 3. Pit health surveyor drone operating system architecture with modular sub-systems. The architecture can be adapted to different use cases through the high-level layer (Applications Layer).

vision module on a companion computer attached to the drone frame, along with a set of drivers and peripherals to estimate local position using Visual Inertial Odometry (VIO) and establish Communication with the operator. The subscribed Ground Control Station (GCS) receives real-time telemetry data and decides whether to let the drone interact with the environment or regain manual control using the Remote Control (RC) controller. The GCS also publishes the drone's velocity and altitude feedback to subscribed sub-systems (e.g., for localization and image processing). For every flight, the drone posts captured videos of the environment from the depth and RGB cameras. The high-level layer receives those videos for processing and reporting. High-level sub-systems are application layer sub-systems such as our multi-modal color-depth mosaicking sub-system, AI-based pit detection, segmentation, and 3D modeling. We present the output in a user interface for pit monitoring for long-term temporal development tracking of the aluminum furnace pits that receive the final results offloaded to the cloud. Other sub-systems can be tailed to the application needs.

2.2. Drone construction and autonomous indoor flight control

We use a lightweight quad-copter drone with a glass-fiber reinforced nylon frame with 22.86 cm plastic propellers, 2216 KV8800 brushless motors, and a 433 MHz telemetry radio. The frame material is insensitive to temperature variation. The carried hardware components have an absolute maximum temperature that varies between 85 °C to 120 °C. If the temperature increases above these levels and the main Central Processing Unit (CPU) gets hotter, the drone will throttle down speeds automatically to maintain temperature limits. The linear temperature expansion coefficient of glass-fiber reinforced nylon is 23 measured at 25 °C. The onboard optical sensors are designed to withstand harsh conditions in industrial environments. They all operate at temperature ranges of 0 °C–85 °C. The drone's length, width, and height dimensions are 38.3 cm × 38.5 cm × 10.0 cm, with a total payload capacity of 1 kg. The drone uses the PX4 flight controller. We equip it with IMU and hi-res stereo optical sensors for localization, tracking, obstacle avoidance, and navigation in a GPS-denied setting. We use hardware triggers to ensure maximum precision in data acquisition. The total cost of the drone and sensors is 4k USD, making it a cost-effective solution, especially when compared to operational costs of manual labor, health and safety risks taken by workers, and the longer production downtimes.

We collect frames using the optical sensors and process them for feature extraction, matching, and optical flow calculation to extract the drone pose. We fuse the data to estimate the VIO from the orientation, velocity, position, and pose information collected. The VIO block diagram is shown in Fig. 4.

Communication between the optical sensors and the PX4 flight controller happens using a library that runs on the VOXL® flight companion computer. Collected data from the optical sensors about the environment is then processed by the flight companion computer and transmitted to the flight controller over the high-speed UART interface using the MAVROS package and MAVLink protocol. MAVROS is a Robot Operating System (ROS) package that allows drones to be controlled via the MAVLink protocol. MAVLink is a communication protocol that connects flight controllers with ground control stations and the companion computer (ROS, 2023). The flight controller then uses this data to adjust the drone's flight path and avoid obstacles. Low-level flight control algorithms are processed using the flight core, the controller's CPU. The flight companion computer sends high-level control commands and receives low-level sensor data from the flight core.

The flight control and mission planning software running on the GCS is connected to the VOXL® flight companion computer through Wi-Fi allowing the GCS to send flying command signals to the flight

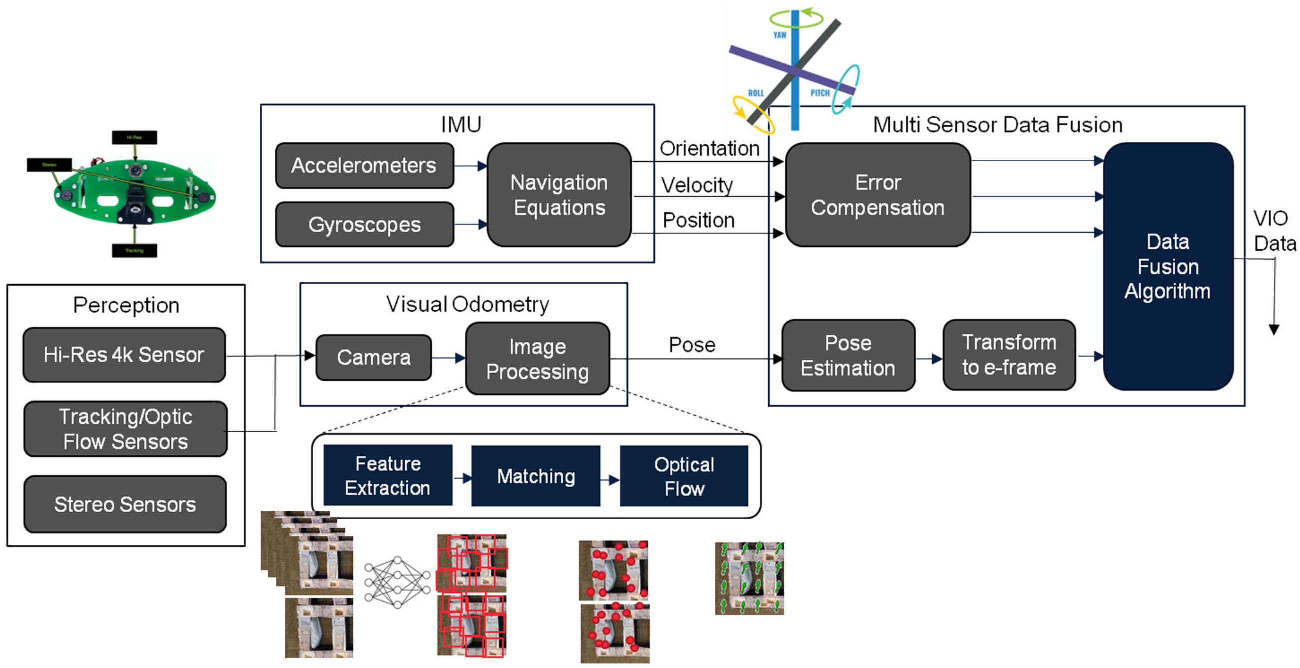

Fig. 4. VIO block diagram for estimating local position in GPS-denied environments.

controller, such as takeoff, landing, and navigation commands. The flight controller also receives signals from the RC, typically used for manual flight control, and the Visual Obstacle Avoidance (VOA) system, with conflicts resolved through priority setting. The commands we send set the drone's position in the XYZ plane (real-world coordinates) using the body coordinate system (NED) and guide the drone to the destination position from takeoff until landing. RC controller and VOA commands override the GCS commands in emergencies. The drone mission is designed to travel over the entire floor in a zigzagged pattern with sufficient frame overlap for registration and mosaic generation.

2.3. Proposed Multi-modal Light-depth mosaicking

We use drone flight videos to synthesize two mosaics of the entire floor in preparation for detecting and identifying pits (one color mosaic in RGB and one depth mosaic). Pit identification is necessary for tracking its health over time. Once pits are detected, we use their fixed relative pixel positions to identify them. Pit detection isolates pit images from the background in preparation for segmentation and 3D modeling. We segment the most important layers from the visual image (i.e., top and bottom layers) and blend the depth frame for the rest to add wall visualization (i.e., all other layers) using a color map to help process engineers perceive how deep the thickening is. Because visual cameras offer a significantly higher number of usable feature points for matching and, eventually, transformation matrix estimation, we stitch both mosaics using the color motion vectors. We finally apply a final transformation that warps the depth frame to the color frame coordinate system to correct the minor distance and frame resolution differences.

We use an 8-parameters projective geometric transformation model for the camera motion between two consecutive video frames. Let $\mathbf{a}_{n\leftarrow n-1} = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7]$ be the projective model parameters between color frames I_n and I_{n-1}, where n is the frame number. The motion vector $\mathbf{d}_{n\leftarrow n-1} = [dx_{n\leftarrow n-1}, dy_{n\leftarrow n-1}]^T$ necessary to warp or transform the pixel $I_{n-1}(x, y)$ to the I_n frame coordinate system is thus given by

$$dx_{n\leftarrow n-1} = a_0 + a_1 x + a_2 y + a_3 xy \tag{1}$$

$$dy_{n\leftarrow n-1} = a_4 + a_5 x + a_6 y + a_y xy \tag{2}$$

or in matrix form

$$\mathbf{d}_{n\leftarrow n-1} = \begin{bmatrix} dx_{n\leftarrow n-1} \\ dy_{n\leftarrow n-1} \end{bmatrix} = \begin{bmatrix} 1 & x & y & xy & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x & y & xy \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \mathbf{Xa}. \tag{3}$$

Rearranging $\mathbf{a}_{n\leftarrow n-1}$ in matrix form gives $\mathbf{A}_{n\leftarrow n-1}$ or

$$\mathbf{A}_{n\leftarrow n-1} = \begin{bmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ a_6 & a_7 & 1 \end{bmatrix}. \tag{4}$$

Using the transformation between consecutive frames, we estimate the transformation between distant frames I_n and I_1 using

$$\hat{\mathbf{A}}_{n\leftarrow 1} = \mathbf{A}_{n\leftarrow n-1} \times \mathbf{A}_{n-1\leftarrow n-2} \times \cdots \times \mathbf{A}_{3\leftarrow 2} \times \mathbf{A}_{2\leftarrow 1} \tag{5}$$

and refine the estimation by using it as an initial condition to a final round of robust optimization using the frames I_n and I_1. We produce the color mosaic image M_C and the depth mosaic M_D by image warping using the motion vectors $\mathbf{d}_{n\leftarrow 1}$ estimated from color images

$$M_C(x, y) = I_n(x + dx_{n\leftarrow 1}, y + dy_{n\leftarrow 1}) \tag{6}$$

$$M_D(x, y) = D_n(x + dx_{n\leftarrow 1} + dx_{C\leftarrow D}, y + dy_{n\leftarrow 1} + dy_{C\leftarrow D}), \tag{7}$$

where $[dx_{C\leftarrow D}, dy_{C\leftarrow D}]$ is the fixed motion field between the color and depth images estimated using multi-modal calibration. Our multi-modal calibration uses mutual information between the derivatives to find the color-depth transformation. We blend the two images in the wall region of interest using alpha-blending or

$$M_B = \alpha M_C + (1 - \alpha) M_D \tag{8}$$

with α as a parameter available for the user to modify in the user interface as needed.

To estimate $\mathbf{A}_{n\leftarrow n-1}$, we first detect the Speeded-Up Robust Features (SURF) (Bay et al., 2008) for the current and the previous gray-scale frames as candidate points of interest. We then perform feature matching between the points as shown in Fig. 5. The red circles represent

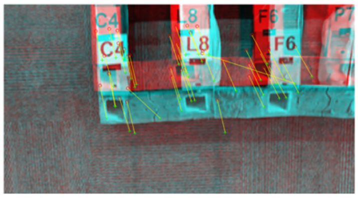

Fig. 5. Feature matching.

the points in the previous frame, while the green + signs represent the corresponding points in the current frame. A minimum of four matches are needed to estimate $\mathbf{A}_{n\leftarrow n-1}$.

The full pipeline for the mosaicking process with intermediate outputs is illustrated in Fig. 6.

2.4. Proposed data augmentation and pit detection algorithm

We propose a data-augmentation technique to increase the training data size and balance the number of normal and deformed pits of all three classes (i.e., normal, wide, and narrow). Our data augmentation is depicted in Fig. 7.

We first extract many natural deformations from images taken on-site. We then subject those deformations to image transforms to synthesize new naturally-looking ones. We follow by taking a pit model of certain class and blend natural and synthetic deformations in to produce deformed pit models. Aligning the deformed pit models allows us to build a synthetic floor mosaic. We apply standard data augmentation techniques (e.g., scale, rotation, illumination changes) to the full mosaics to create new ones. We also record ground truth data to pass it to the object detector during the training stage. To further increase our training data size and be able to tune and test our system in the lab before taking it on-site, we built a physical scaled model of the factory

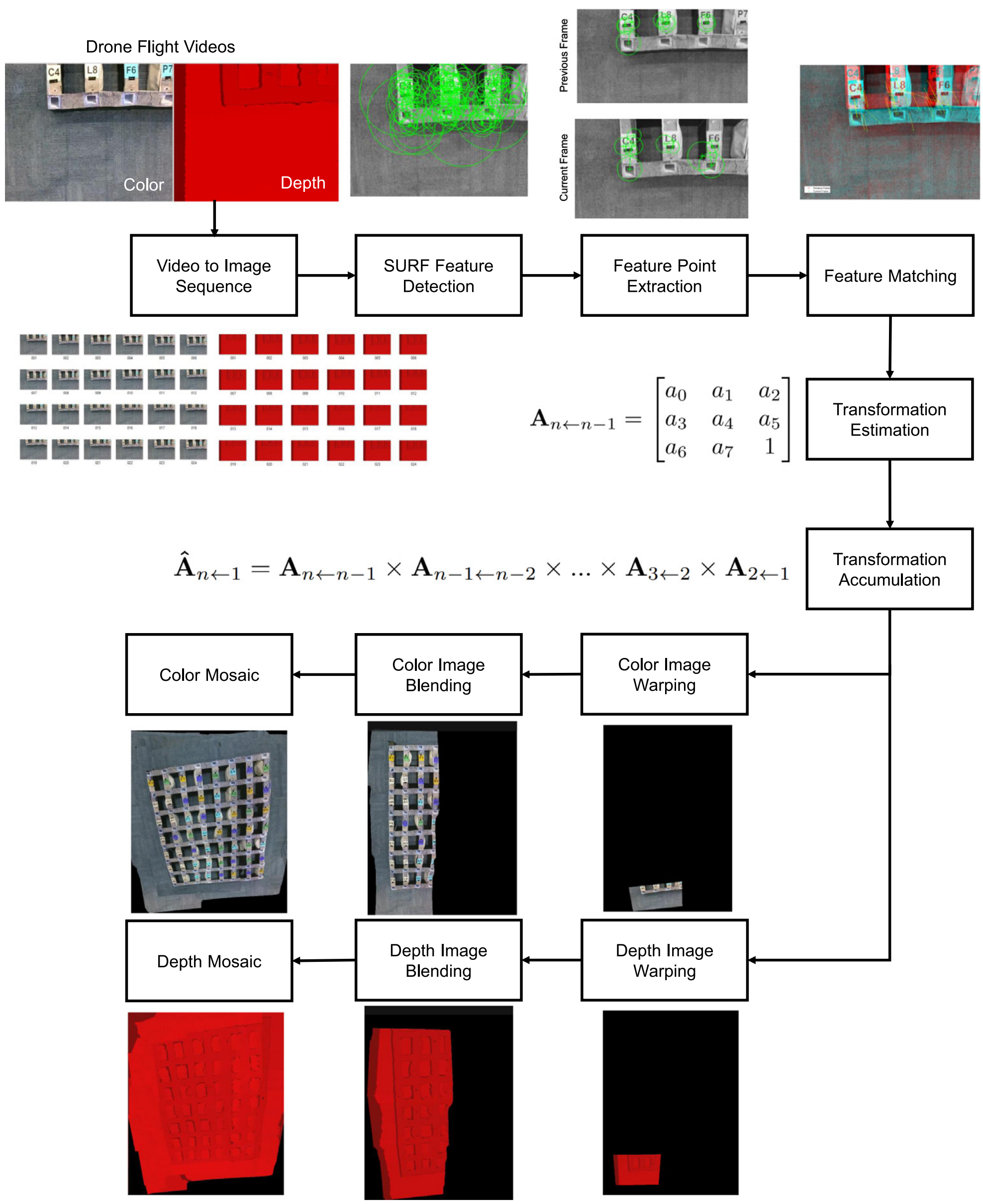

Fig. 6. Full pipeline for the mosaicking process using color and depth flight videos.

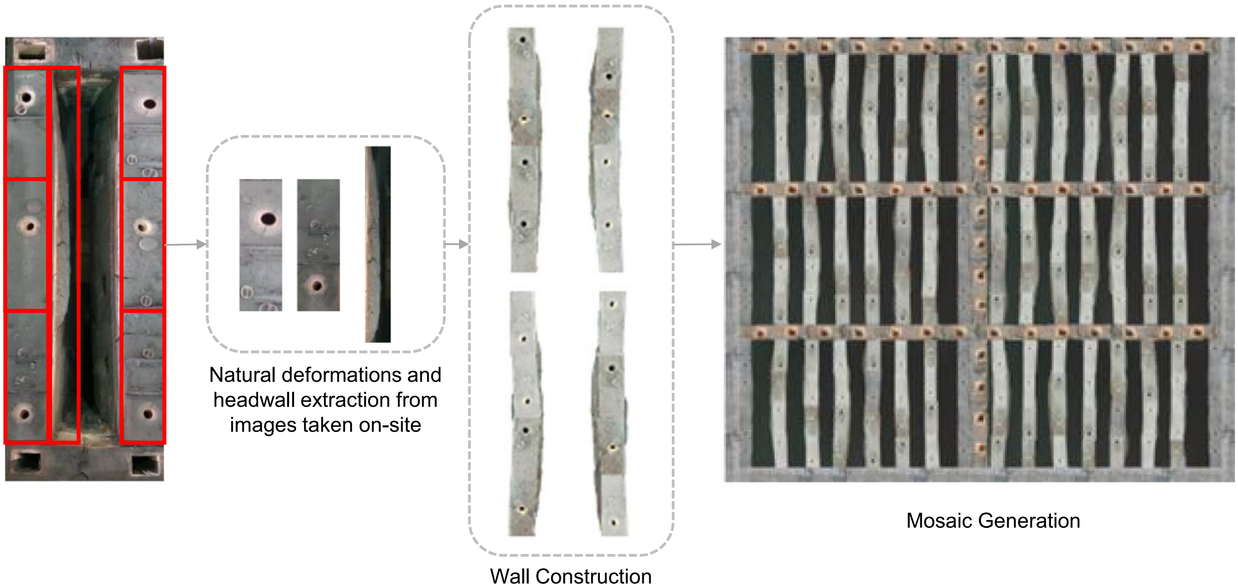

Fig. 7. Color mosaic data augmentation for object detection.

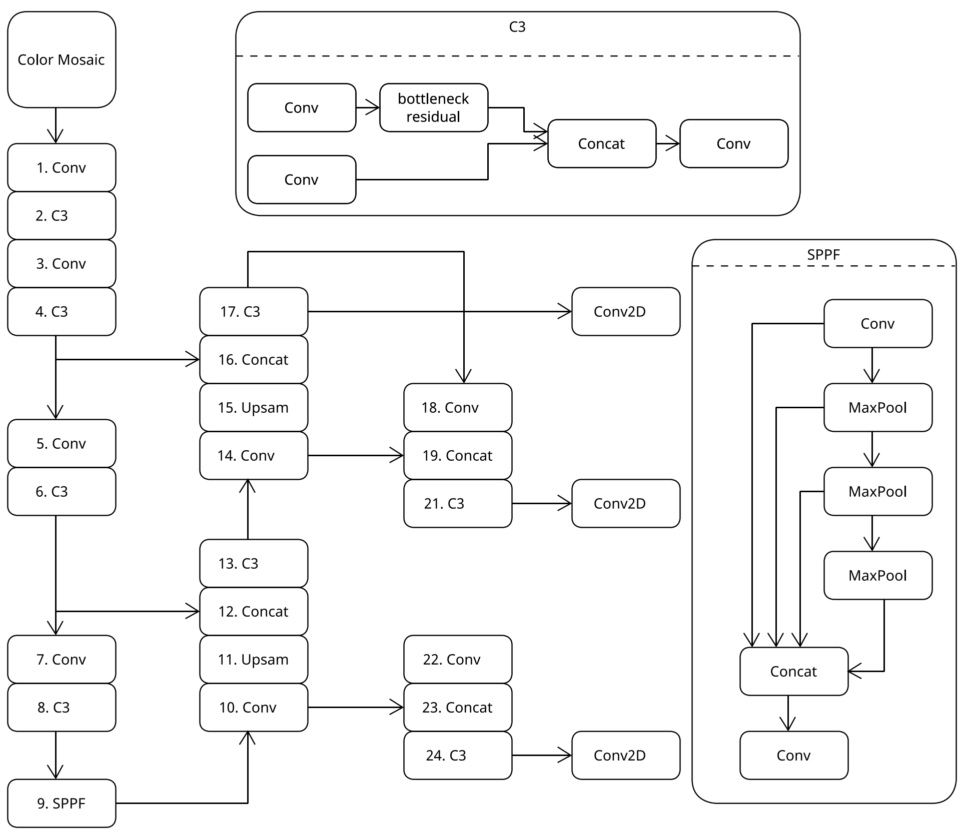

Fig. 8. YOLOv5 Network Architecture (Jocher, 2022).

floor. To create visually similar color and depth videos to ones collected on site, we reduced the drone flight altitude during in-lab indoor flights. We used the physical model for additional data augmentation. The physical model size is 2 m×4 m with 49 pits. Model pit walls were thinned or thickened using paste.

We use the YOLOv5 object detector (Jocher, 2022) whose architecture is depicted in Fig. 8 to localize and initially-classify all pits on a floor.

Pit images are then extracted and passed on to the pit segmentation and 3D modeling subsystem. We downscale the mosaics M_C and M_D

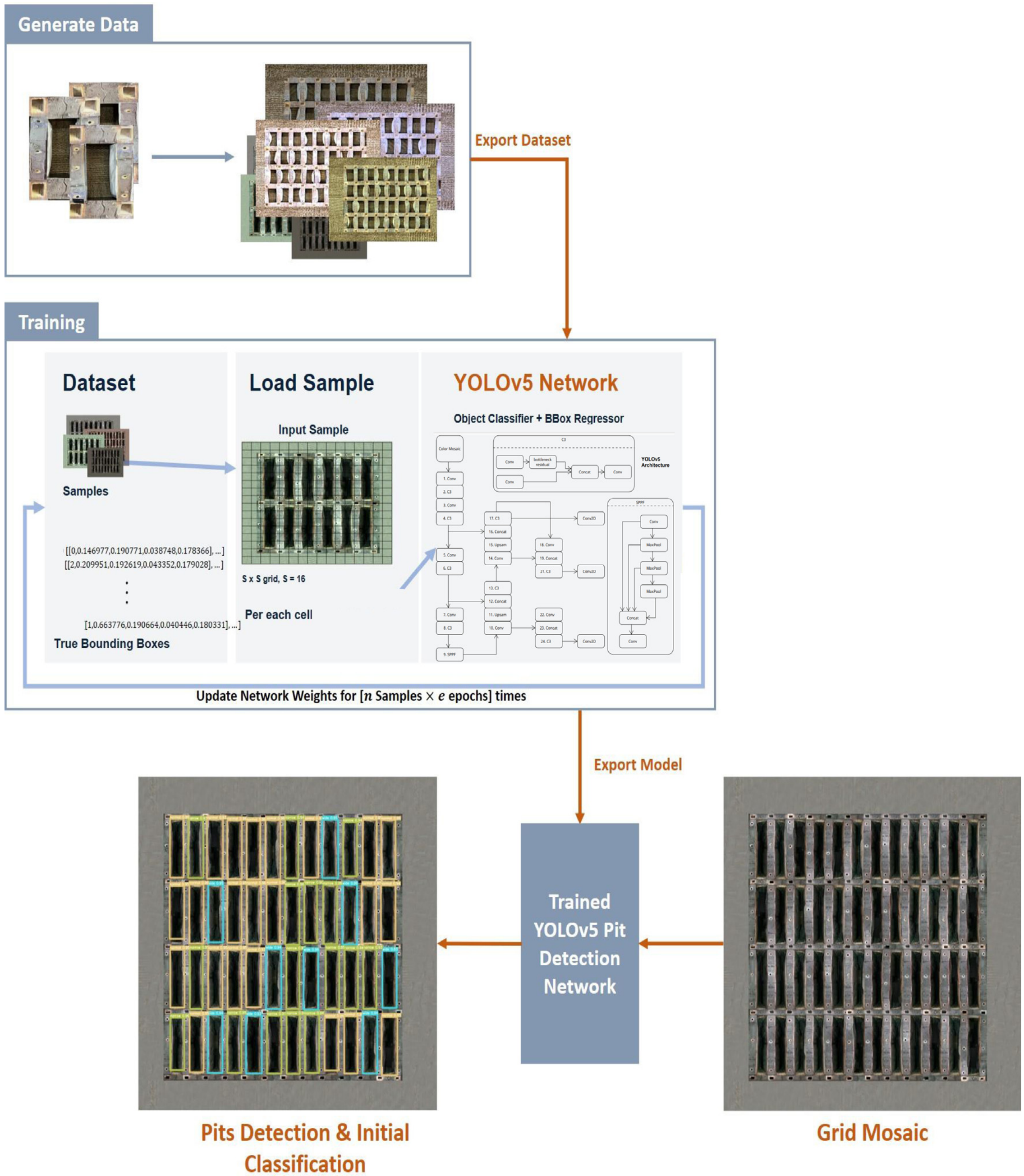

Fig. 9. Pit detection system overview.

to a smaller resolution 512 × 736 to reduce training time. Once pit locations are found, they are scaled back up. We do not require the detection to be highly accurate since the segmentation subsystem refines the localization in the next stage. We illustrate the pit detection subsystem in Fig. 9.

YOLOv5 divides the color mosaic image M_C in (6) into 32×32 cells. The detector predicts the confidence of a pit ρ being in a each cell i, or ρ_c^i, the location of the pit using four bounding box parameters (i.e., pit top left corner coordinates (ρ_x, ρ_y), pit width ρ_w and pit height ρ_h), and the probability of an pit belonging to one of our classes $P(\text{normal}|\rho)$, $P(\text{narrow}|\rho)$, and $P(\text{wide}|\rho)$. Example cells and probabilities are shown in Fig. 10.

The model is trained by optimizing the loss function in

$$
\begin{aligned}
L = {} & \lambda_{coord} \sum_{i=0}^{1024} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(\rho_x^i - \hat{\rho}_x^i)^2 + (\rho_y^i - \hat{\rho}_y^i)^2] \\
& + \lambda_{coord} \sum_{i=0}^{1024} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(\sqrt{\rho_w^i} - \sqrt{\hat{\rho}_w^i})^2 + (\sqrt{\rho_h^i} - \sqrt{\hat{\rho}_h^i})^2] \\
& + \sum_{i=0}^{1024} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(\rho_c^i - \hat{\rho}_c^i)^2] \\
& + \lambda_{noobj} \sum_{i=0}^{1024} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} [(\rho_c^i - \hat{\rho}_c^i)^2] \\
& \sum_{i=0}^{1024} \mathbb{1}_{i}^{obj} [P(\text{normal}|\rho^i) - \hat{P}(\text{normal}|\rho^i)]^2 \\
& + [P(\text{narrow}|\rho^i) - \hat{P}(\text{narrow}|\rho^i)]^2 \\
& + [P(\text{wide}|\rho^i) - \hat{P}(\text{wide}|\rho^i)]^2,
\end{aligned} \tag{9}
$$

where j is the bounding box index, λ_{coord} and λ_{noobj} are constants to emphasize certain sections of the loss function, B is the bounding box

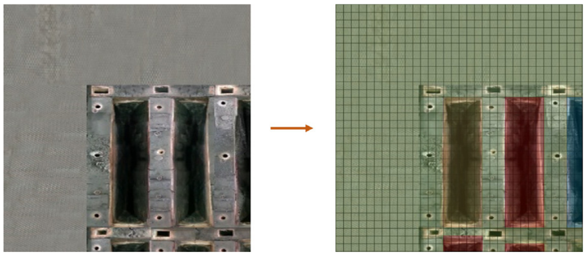

Fig. 10. Prediction class probability map. Yellow is for normal, red is for narrow, and blue is for wide.

prediction for each grid cell, $\mathbb{1}_i^{obj}$ is an indicator function giving 1 if there is an object in cell i and 0 otherwise, $\mathbb{1}_{ij}^{obj}$ is 1 if there is an object in cell i and the confidence of the jth predictors of this cell is the highest among all the predictors of this cell, and $\mathbb{1}_{ij}^{noobj}$ is similarly defined for to trigger when there are no objects in cell i.

2.5. Proposed pit segmentation and 3D modeling

Once the pits are detected and isolated from the background, we apply two segmentation pipelines to segment the top and bottom layers of the pits. We use the pit depth data to visualize the walls. We then create a 3D model of the pit using the segmented layers and the blended depth frames for all other layers. Segmenting the bottom layer of the pit is harder than segmenting the top layer due to its similar-color to wall pixels and general homogeneity. Hence the need to segment both using different methods.

2.5.1. Proposed pit-top segmentation

We propose a pit-top segmentation technique that first detects peepholes and uses them to identify the flue wall pixels. We first convert the RGB image of the detected pits from the pit detection subsystem into HSV due to the low illumination conditions. We use the Saturation (S) channel since the peepholes have much lower saturation than their surroundings (i.e., flue walls). We apply OTSU binarization on the S channel to obtain an optimum threshold. Given the binarized image, we apply morphological operations (i.e., opening and closing) to restore the shape of the holes. We then apply Hough transform to detect the circle blobs representing the peepholes. For an accurate segmentation of flue walls, we need to identify at least one peephole in each flue wall, preferably two and, at best, three. This will then be used to generate a Region of Interest (ROI) around each peephole detected to acquire an initial segmentation of the flue walls. A mask is applied on the value channel of the image. Due to some pixels being misclassified as a wall, we train a GMM with k = 2 to estimate a Gaussian model of the wall pixels and differentiate between the wall pixels and non-wall pixels, as illustrated in Fig. 11. Since the color values of the flue walls are more distinctive than the non-flue wall pixels, the Gaussian model of the flue walls is easily extracted to produce a refined segmented region. We perform another round of morphological operations (i.e., blurring and closing) to have a fully connected region of the walls. We follow this operation with OTSU binarization. The output mask is applied to the S channel of the HSV domain of the detected pit image, which is followed by area filtering using connected component analysis. This filters out unwanted connected sets (i.e., unwanted small regions) and is then used to differentiate between the left and right flue walls. Canny edge detection is applied, followed by horizontal dilation to detect the outer and inner edges of the left and right flue walls. Finally, the outer edges are discarded, and we overlay the inner edges of the flue walls on the detected pit images. We use the resulting inner edges and circles for distance measurement and pit modeling. The full pit-top segmentation pipeline with intermediate outputs can be seen in Fig. 12.

2.5.2. Proposed pit-bottom segmentation

We use manually segmented Ground-Truth (GT) images of the wall and the bottom-layer of the pits to build two atlases. The atlases are used to transform the image from the color to a probability feature space. Our feature image is later used to train a semantic segmentation network. Let (u, v) be a different coordinate system from the mosaic's (x, y), $\Phi(u, v)$ be the pit sub-image extracted after pit detection from the floor color mosaic $M_C(x, y)$. To produce $\Phi(u, v)$, we populate it from $M_C(x, y)$ using the detection results bounding boxes, i.e., the top left coordinates, width, and height, or $(\rho_x, \rho_y, \rho_w, \rho_h)$. We do so using

$$\Phi(v, w) = \{M_C(x, y) : \quad \forall \ \rho_x \le x \le \rho_x + \rho_w \\ \rho_y \le y \le \rho_y + \rho_h \\ 0 \le v \le \rho_w \\ 0 \le w \le \rho_h\}. \tag{10}$$

Furthermore, let $W_{GT}^h(u, v)$ be the hth training binary image where wall pixels are the foreground pixels, $B_{GT}^h(u, v)$ is another binary image where bottom-layer pixels are foreground, and $\mathrm{ROI}_{GT}(u, v)$ is a region of interest binary image where both wall or bottom-layer pixels are foreground. We apply a pixel-wise ORing operation on $W_{GT}^h(u, v)$ and $B_{GT}^h(u, v)$ to produce $\mathrm{ROI}_{GT}(u, v)$. We detect the edges in $\mathrm{ROI}_{GT}(u, v)$ to produces a single-pixel wide closed outer contour $b_1 = ((u_1, v_1), (u_2, v_2), (u_3, v_3), \ldots (u_1, v_1))$. We reset (to zero) all pixels in b_1 and reapply binary edge detection on the updated $\mathrm{ROI}_{GT}(u, v)$ to find b_2, i.e., the 2nd outermost contour and so on. We stop when we reach the innermost contour. We propose b_k, the list of all contours, as a distance map. All pixels belonging to the kth contour are logically grouped in this distance map.

We use b_k to build two atlases, namely the *wall* atlas ζ_k^W containing contour point coordinates (u, v) and their corresponding RGB colors $(\Phi^R(u, v), \Phi^G(u, v), \Phi^B(u, v))$, and the similarly structured *bottom-layer* atlas. To create the atlases, we use

$$\zeta_k^W = \{(u_i, v_j, \Phi^R(u_i, v_j), \Phi^G(u_i, v_j), \Phi^B(u_i, v_j)) : (u_i, v_j) \in b_k \text{ and} \\ W_{GT}^h(u_i, v_j) = 1 \ \forall \ i, j, \text{and } h\}, \tag{11}$$

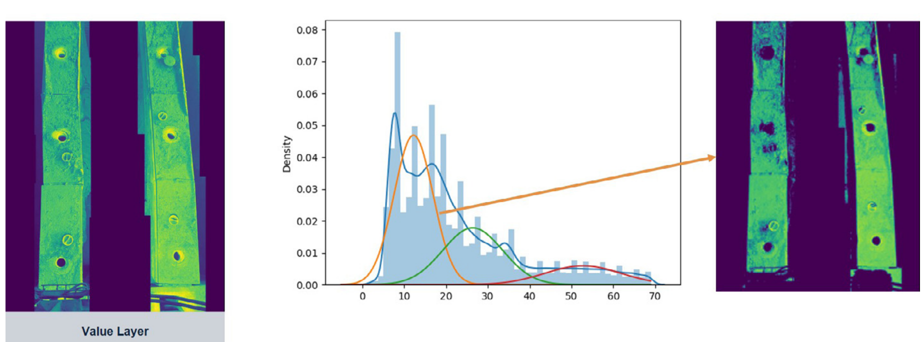

Fig. 11. Finding the walls' Gaussian model.

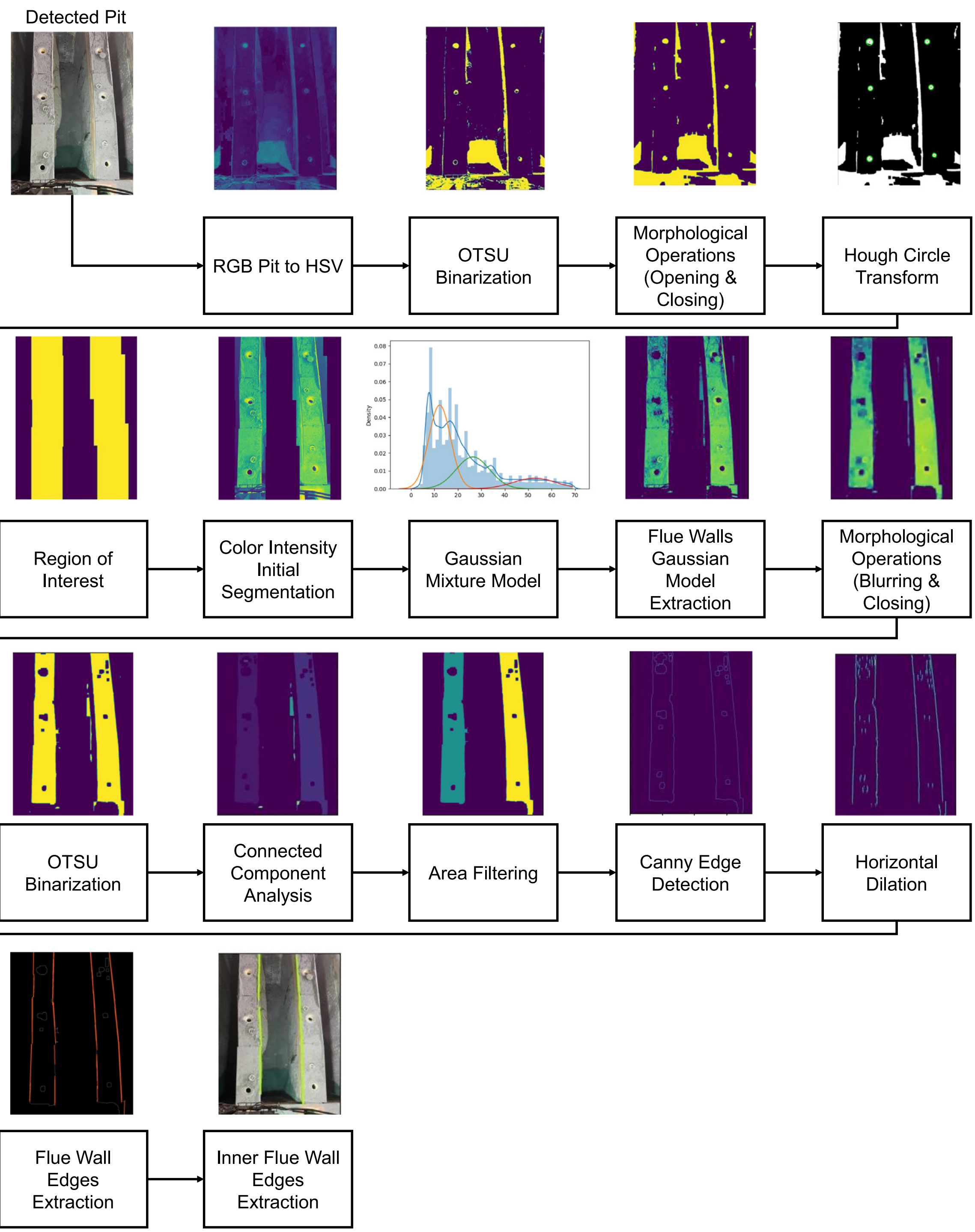

Fig. 12. Full pipeline for the pit-top segmentation.

$$\zeta_k^B = \{(u_i, v_j, \Phi^R(u_i, v_j), \Phi^G(u_i, v_j), \Phi^B(u_i, v_j)) : (u_i, v_j) \in b_k \text{ and } B^h_{GT}(u_i, v_j) = 1 \ \forall\ i, j, \text{and } h\}. \quad (12)$$

Algorithm 1 summarizes the atlases generation algorithm.

In Fig. 13 we show our distance map image to the left and the final top layer and bottom layer contours detected.

We do not feed $\Phi(u, v)$ directly for the deep learning module due to the high similarity between the wall and bottom-layer appearance features and the lack of sufficient textures in the original image. Instead, we transform $\Phi(u, v)$ into a probability-based feature domain image $\Theta(u, v)$ using the two atlases ζ_k^W and ζ_k^B. When segmenting a new image, we use the ROI(u, v) image generated using the top layer segmentation and build its distance map b_k. For all (u, v) pixels in ROI(u, v) that belong to contour k, we calculate the first channel of the $\Theta(u, v)$ image using the gray-scale value calculated from the RGB values using

$$\Theta^R(u, v) = 0.2989\ \Phi^R(u, v) + 0.5870\ \Phi^G(u, v) + 0.1140\ \Phi^B(u, v). \quad (13)$$

We then compare the pixel color to the pixel colors of the same contour k in both the wall and the bottom-layer atlases. We accomplish this by finding the Euclidean distance using the color values, sorting the distances, and finding the lowest values (nearest neighbors in color). We do it for all contours and pixels of both atlases. We then compare

Algorithm 1: Atlas Database Generation

> **Result:** Atlas Database ζ_k^W and ζ_k^B
> **for** *each training image h* **do**
>   Detect the boundary level b_k from the ground truth ROI image $\mathrm{ROI}_{GT}(u, v)$;
>   **for** *each pixel in* b_k **do**
> $(u_i, v_j, \Phi^R(u_i, v_j), \Phi^G(u_i, v_j), \Phi^B(u_i, v_j)) \leftarrow$ Form a tuple from color and position
>   **if** *bottom-layer pixel* **then**
>   Add $(u_i, v_j, \Phi^R(u_i, v_j), \Phi^G(u_i, v_j), \Phi^B(u_i, v_j))$ to floor atlas ζ_k^B;
>   **else**
>   Add $(u_i, v_j, \Phi^R(u_i, v_j), \Phi^G(u_i, v_j), \Phi^B(u_i, v_j))$ to wall atlas ζ_k^W;
>   **end**
>   **end**
> **end**

the nearest neighbors to build a probability (or potential) measure and use it to set the green channel in the feature space image $\Theta^G(u, v)$ using

$$T^3(u,v) = \sum_{r=1}^{3} \min_{ij}^{r} \sqrt{\Big(\left[\Phi^R(u,v) - \Phi^R_{\zeta_k^B}(u_i,v_j)\right]^2 + \left[\Phi^G(u,v) - \Phi^G_{\zeta_k^B}(u_i,v_j)\right]^2 + \left[\Phi^B(u,v) - \Phi^B_{\zeta_k^B}(u_i,v_j)\right]^2 \Big)} < \min_{lo}^{r} \sqrt{\Big(\left[\Phi^R(u,v) - \Phi^R_{\zeta_k^W}(u_l,v_o)\right]^2 + \left[\Phi^G(u,v) - \Phi^G_{\zeta_k^W}(u_l,v_o)\right]^2 + \left[\Phi^B(u,v) - \Phi^B_{\zeta_k^W}(u_l,v_o)\right]^2 \Big)} \tag{14}$$

$$\Theta^G(u,v) = \frac{T^3(u,v)}{3} \tag{15}$$

where $\min^r$ is the rth smallest value. We use i, j to index in one of the atlases and l, o to index in the other atlas to emphasize the fact that the number of pixels belonging to a certain class in particular contour number k is typically different. Note that in (17) the $\sum$ operator sums binary decisions (the output of the rational statement) to count the number of times the bottom-layer minimums were lower than the wall minimums. In other words, $T^3(u, v)$ counts the number of times the color contenders from the bottom-layer atlas were more similar in color to the contenders from the wall atlas. While $\Theta^G(u, v)$ has the probability in case of using three neighbors. We populate $\Theta^B(u, v)$ by using five instead of three. $\Theta^B(u, v)$ is populated using

$$T^5(u,v) = \sum_{r=1}^{5} \min_{ij}^{r} \sqrt{\Big(\left[\Phi^R(u,v) - \Phi^R_{\zeta_k^B}(u_i,v_i)\right]^2 + \left[\Phi^G(u,v) - \Phi^G_{\zeta_k^B}(u_i,v_j)\right]^2 + \left[\Phi^B(u,v) - \Phi^B_{\zeta_k^B}(u_i,v_j)\right]^2 \Big)} < \min_{lo}^{r} \sqrt{\Big(\left[\Phi^R(u,v) - \Phi^R_{\zeta_k^W}(u_l,v_o)\right]^2 + \left[\Phi^G(u,v) - \Phi^G_{\zeta_k^W}(u_l,v_o)\right]^2 + \left[\Phi^B(u,v) - \Phi^B_{\zeta_k^W}(u_l,v_o)\right]^2 \Big)} \tag{16}$$

$$\Theta^B(u,v) = \frac{T^5(u,v)}{5} \tag{17}$$

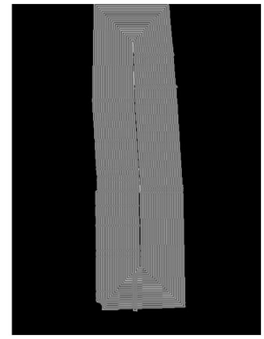

Fig. 13. Distance map as shape priors. We use contours to achieve a level of rotational invariance.

To balance the gray-level, we turn the probabilities again to colors by scaling them $0 < \frac{T^5(u,v)}{5} < 1$ by 255. If the color distance is equal for both the bottom-layer and wall pixels, we use position as a tie breaker. We first find all pixels in the atlas which share the same color similarity (distance) and compare all bottom-layer pixels' locations to the wall pixel locations using the Euclidean distance to populate $T(u, v)$ using

$$T^{3,5}(u,v) = \sum_{r=1}^{3,5} \min_{ij\ \textbf{color-similar}}^{r} \sqrt{\Big(\left[u - u_{i\zeta_k^B}\right]^2 + \left[v - v_{j\zeta_k^B}\right]^2 \Big)} < \min_{ij\ \textbf{color-similar}}^{r} \sqrt{\Big(\left[u - u_{l\zeta_k^W}\right]^2 + \left[v - v_{o\zeta_k^W}\right]^2 \Big)}. \tag{18}$$

With all three channels in $\Theta(u, v)$ populated, a false color image is generated. We use this feature images to train a semantic segmentation network using the DeepLabV3+ architecture (Chen et al., 2018) as shown in Fig. 14. Probability map generation is summarized in Algorithm 2.

3. Validation and results

In this section, we present the testing results for each of the drone's sub-systems. We also validate our proposed architectures by discussing the end-to-end results of deploying our system on-site at an aluminum factory in the United Arab Emirates.

3.1. Experimental configuration

To test flight control and pit detection, we used on-site collected images, images from the physical model built in the lab, and data augmentation images. The lab and data augmentation environment allowed us to optimize controls and weights before further training and testing on-site.

On-site testing requires production pausing and pits clearing. Therefore, we built an experimental setup for initial lab testing, shown in Fig. 15(a). The dimensions of the experimental physical model are 2 m × 4 m with 49 pits mimicking a furnace floor on-site seen in Fig. 15(b). The dimensions of each section in the baking furnace on-site are 55 m × 50 m. We model the deformations in the pit walls using paste.

The drone model used to test our AI-powered autonomous indoor drone video-surveying system is a VOXL® m500 drone (ModalAI, 2023). The drone uses a ModalAI Flight Core® PX4 flight controller. We equip our drone with a DJI Osmo Pocket (DJI, 2023) for stabilized visual video recording and an Intel® RealSense™ Depth Camera D435 (Intel, 2023) for depth and RGB video recording with up to 90 FPS.

We performed more than 20 drone flights in the lab and on-site. Each drone flight took approximately 3–5 min of surveying. The on-site factory has three furnace floors, each with 48 pits. We use a total

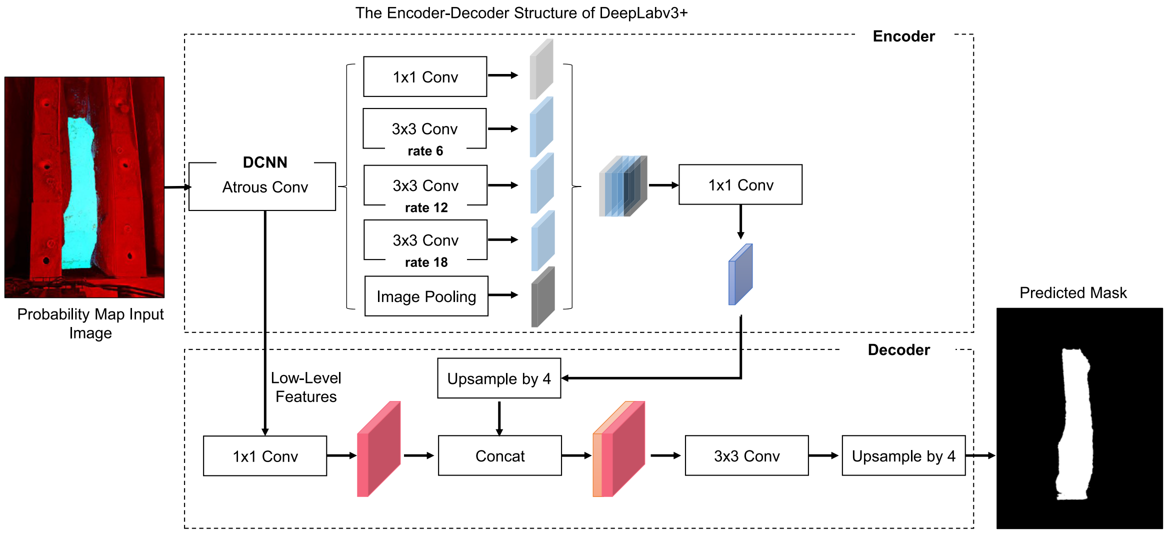

Fig. 14. Proposed architecture for pit-bottom segmentation using DeepLabV3+ (Chen et al., 2018).

Algorithm 2: Probability Maps Generation

Result: Probability Map $\Theta(u, v)$

for *Each image in to segment* **do**

Detect boundaries b_k for ROI(u, v). ;

for *For each test pixel* ROI(u, v) **do**

(i) Get k, boundary level of ROI(u, v)

(ii) Extract color and position features for (u, v) $(u, v, \Phi^R(u, v), \Phi^G(u, v), \Phi^B(u_i, v_j))$ in b_k

(iii) Compute Euclidean distance between $\Phi(u, v)$ colors and colors in ζ_k^W and ζ_k^B.

(iv) Sort the color distance lists ascendingly

(v) Compare the nearest color neighbours from both atlases

(vi) Count the number of times bottom-layer color neighbors were closer than wall color neighbours. Normalize to turn into probability.

(vii) If bottom-layer and wall tie, use candidate pixels with the same position as tie breakers

(viii) Compute $T^3(u, v)$

(ix) Repeat step (viii) for five neighbours instead of three neighbours to get $T^5(u, v)$

end

Generate a probability map image with false colors using the three layers: gray-level, $255\frac{T^3(u,v)}{3}$, and $255\frac{T^5(u,v)}{5}$ to replace the RGB layers. A sample of the generated probability map is shown as the input of the proposed architecture for bottom segmentation in Fig. 14. Train

end

---

of 18,000 images of both real and synthetic pits collected from the lab environment and on-site to train our AI-powered system. We train the model for 12 h with Nvidia RTX™ 3070 GPU Intel Core i7 1 TB SSD computing device. We apply the test-set cross-validation technique for the real and synthesized images of the pits generated from our experimental setup in the lab. We opted to use the combination of 50:20:30 for the training to validation to testing ratio (i.e., 9000, 3600, and 5400 images, respectively).

We evaluate the performance of the YOLO-based pit detection model in terms of the mAP, a popular metric for such a detector (Zou et al., 2019). We use an Intersection Over Union (IoU) threshold of 0.5 to calculate average precision. Using Eq. (19), we calculate mAP, where N represents the number of classes (i.e., three classes being narrow, wide, and normal pits).

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{19}$$

For pits segmentation, we used only on-site images since segmentation on the physical model did not add value. We use transfer learning and apply Leave One Out Cross Validation to train our AI system on the on-site collected images.

We evaluate the performance of the pit segmentation model in terms of accuracy, precision, sensitivity, specificity, Dice-Similarity Coefficient (DSC), and overlap. Using the parameters in the confusion matrix, we obtain the performance metrics of the pit segmentation algorithm. Accuracy is the percentage of the correctly classified pit and non-pit pixels among all pixels, which is computed as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \tag{20}$$

where TP, TN, FP, and FN represent True Positive, True Negative, False Positive (Type 1 Error), and False Negative (Type 2 Error), respectively. Precision is another metric we report on, which measures the proportion of true positive pixels among all pixels classified as

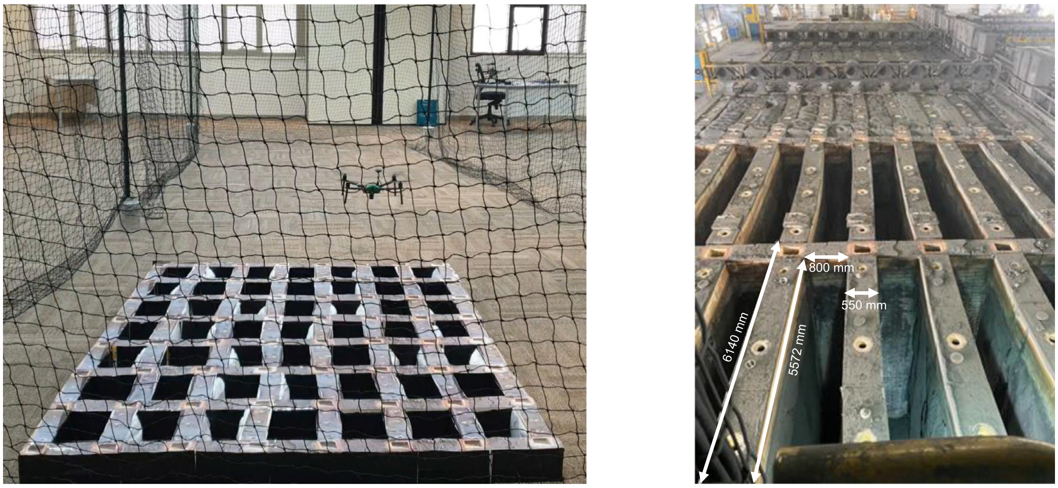

Fig. 15. (a) The full experimental setup of our scaled model of the factory floor built in the lab and (b) the furnace floor on-site with its dimensions in mm.

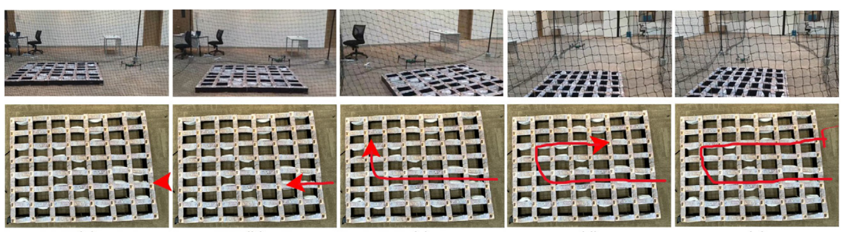

Fig. 16. [Lab test flight 1] Autonomous navigation in the lab environment results — U pattern: In (a), the drone identifies the starting position as the set point and then takes off to an altitude of 1.5 m. To simulate the U-pattern, it flies forward in (b) towards the North, covering the length of the model. In (c), it turns east to cover the width, then flies south back to its starting position, in (d), where it lands eventually as depicted in (e). The drone mission was completed in 27 s.

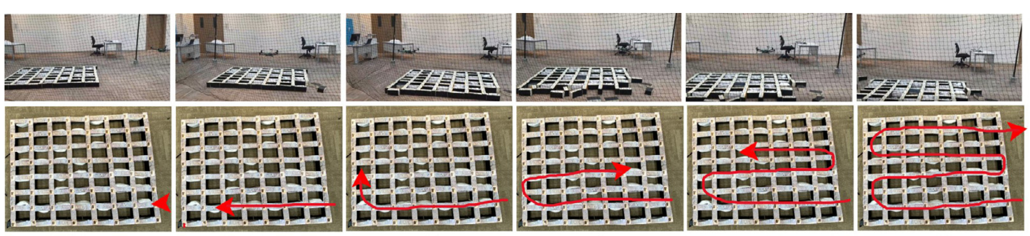

Fig. 17. [Lab test flight 2] Autonomous navigation in the lab environment results — Z pattern: In (a), the drone identifies the starting position as the set point and then takes off to an altitude of 1.5 m. To simulate the Z pattern, it flies forward in (b) towards the North, covering the length of the model. In (c), it turns east to cover one-third of the width, then flies south back to its starting position in (d). Then, it moves another one-third east and flies north, covering the length again in (e). To cover the last one-third of the model, the drone moves east and then flies south, where it lands eventually as depicted in (f). The drone mission was completed in 69 s.

positive (i.e., pit pixels). It can be calculated using Eq. (21).

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

Sensitivity, or Recall, is the ratio of the correctly segmented pit pixels (TP) to the actual pit pixels calculated using Eq. (22). Sensitivity is used to evaluate our model's performance in segmenting the pit region.

$$Sensitivity = \frac{TP}{TP + FN} \tag{22}$$

Specificity measures the proportion of the correctly classified non-pit pixels among all actual non-pit pixels using Eq. (23).

$$Specificity = \frac{TN}{TN + FP} \tag{23}$$

We also report on the overlap and DSC for our pit segmentation model. Overlap measures the ratio of the area of intersection between the predicted and ground-truth pit segmentations to the area of union between the two, while DSC measures the similarity between the predicted and ground-truth pit segmentations.

3.2. Autonomous indoor flight control results

We tested our drone surveyor in both the lab and site environments, and it was able to follow a predetermined path reaching the way points set without collisions or deviations. We test 3 patterns proposed in the literature (U-pattern, zig-zag pattern (Z-pattern), and spiral pattern (S-pattern)) (Xia et al., 2020; Araujo et al., 2020; Bezas et al., 2022) to evaluate the effect of different flight path choices on the performance of the system. We set the drone to fly at a 1.5 m altitude in all patterns to ensure the top and bottom features of the model pits are visible. Figs. 16, 17, and 18 depict samples captured during the test flights conducted in the lab environment. In each of the test flights, the drone takes off to the set altitude, follows the way points simulating the

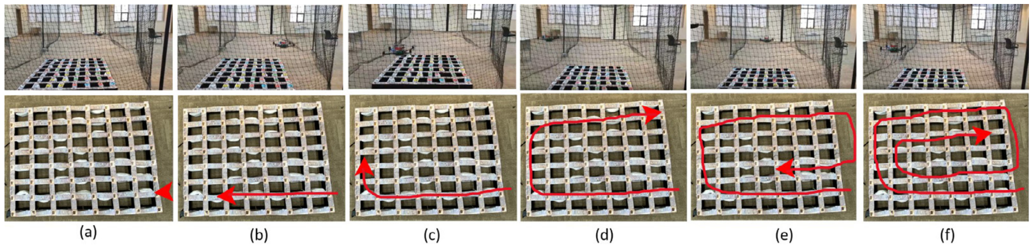

Fig. 18. [Lab test flight 3] Autonomous navigation in the lab environment results — S pattern: In (a), the drone identifies the starting position as the set point and then takes off to an altitude of 1.5 m. To simulate the S pattern, it flies forward in (b) towards the North, covering the length of the model. In (c), it turns east to cover two-thirds of the width, then flies south back to its starting position in (d). Then, it moves one-third east and flies north, covering the length again in (e). To cover the middle one-third of the model, the drone moves east and then flies south, where it lands eventually as depicted in (f). The drone mission was completed in 77 s.

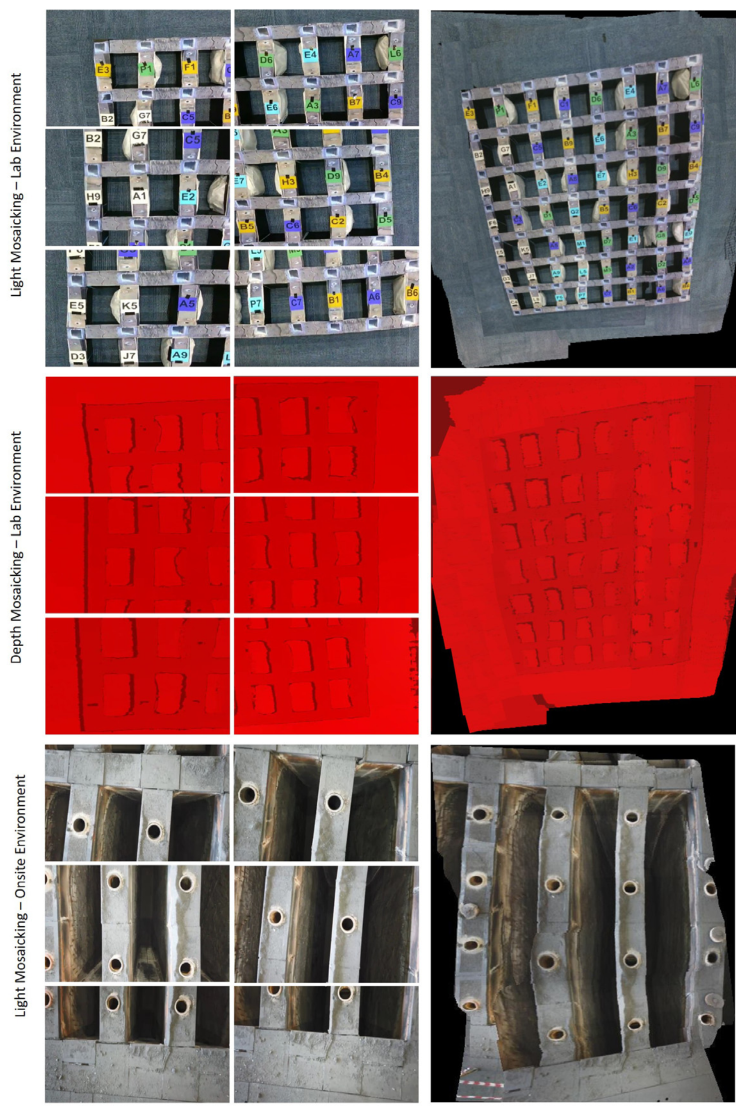

Fig. 19. Visual results of the proposed multi-modal light-depth mosaicking algorithm in the lab environment and on-site: (a) shows the drone's light and depth views and (b) shows the final mosaic outputs.

selected flight pattern while recording aerial images, then lands once the mission is completed. We report the flight duration for each of the patterns to estimate the average flight time required to cover the inspected area.

We compare the data collected in each of the patterns visually to identify an optimized flight path that yields sufficient image overlap while maintaining battery life. The results depict that the U-pattern does not provide sufficient overlap for the mosaicking algorithm as opposed to the Z-pattern and the S-pattern. The reported flight duration shows that the Z-pattern is more time-efficient with an average time reduction of 10.4% for the same area size compared to the S-pattern, which matches the findings in Bezas et al. (2022).

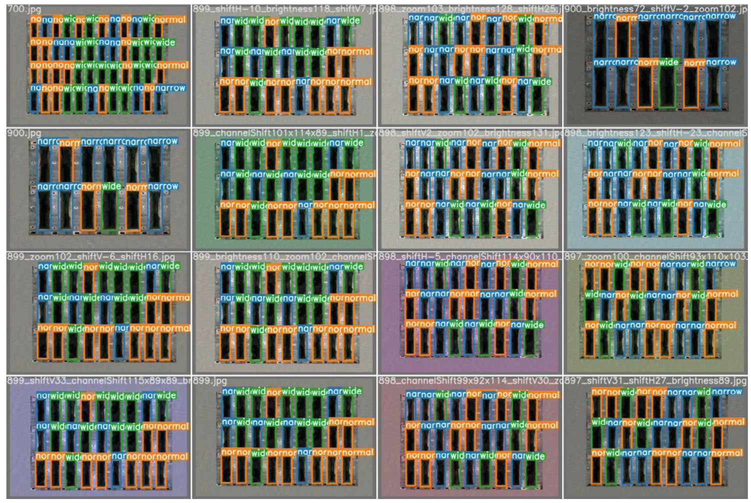

Fig. 20. Samples of the detected pits within the testing dataset.

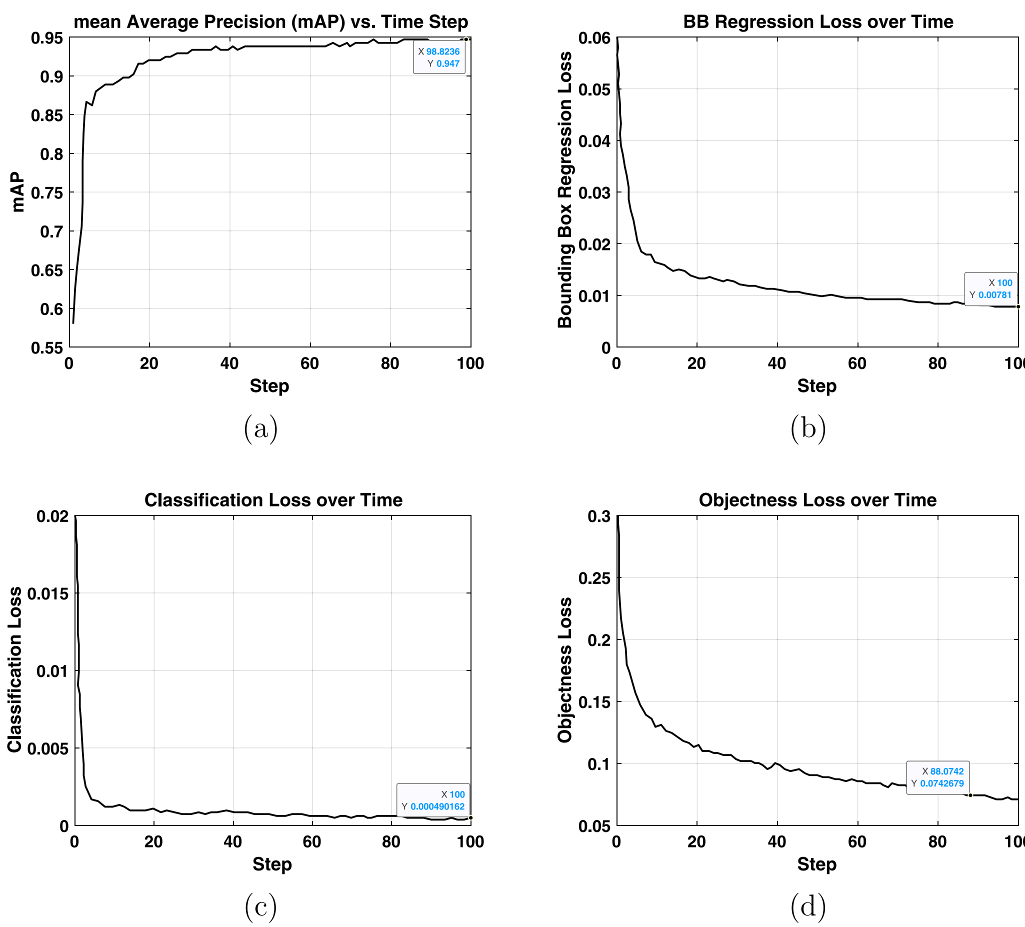

Fig. 21. Performance evaluation results of pit detection algorithm: (a) depicts Mean Average Precision = 94.67%. (b) depicts the Bounding Boxes regression loss = 0.781%, (c) shows the Classification loss score = 0.049%, and (d) shows Objectiveness loss score = 7.102%.

3.3. Multi-modal light-depth mosaicking results

We tested our multi-modal mosaicking algorithm using videos collected throughout the flights in both the lab environment and on-site. Testing results are depicted in Fig. 19. As can be seen in the footage frames, our proposed algorithm successfully generated the mosaic outputs in both settings. Due to the characteristics of the working environment, the system and collected data are inaccessible to readers.

3.4. AI-based pit detection algorithm results

We collected a total of 18,000 images by generating 10,800 grid samples using our physical model and 7200 samples using the synthetic pits to train the pit detection model. We trained the model for 100 epochs with a batch size of 4 samples using Adam optimizer. Fig. 20 shows some samples for the detected pits within the testing dataset. As can be seen in Fig. 21, the mean average precision for the prediction

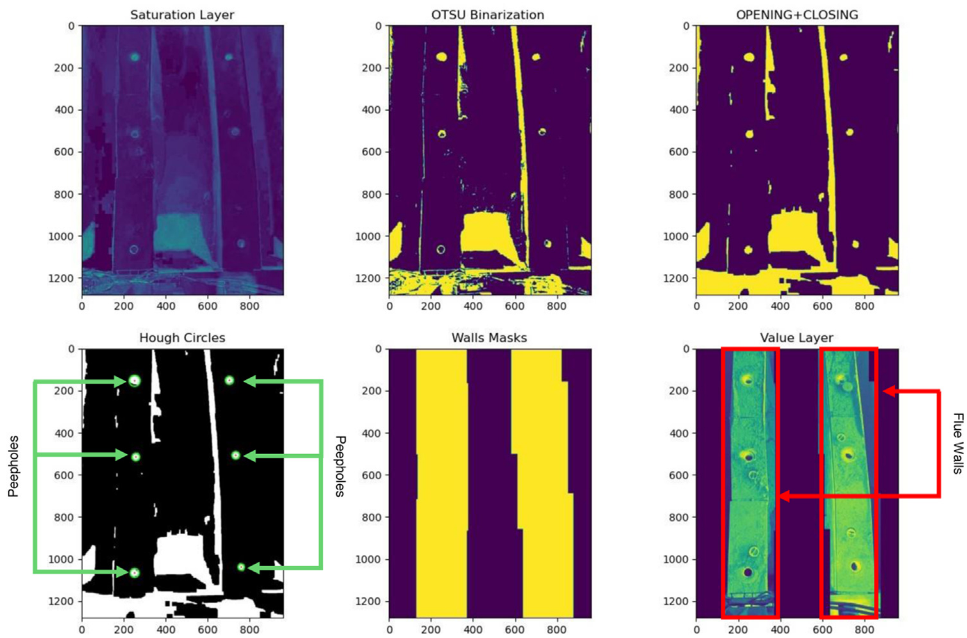

Fig. 22. Finding the peepholes in a detected pit image. The flue walls are indicated using the red boxes, while the peepholes are indicated using the green arrows.

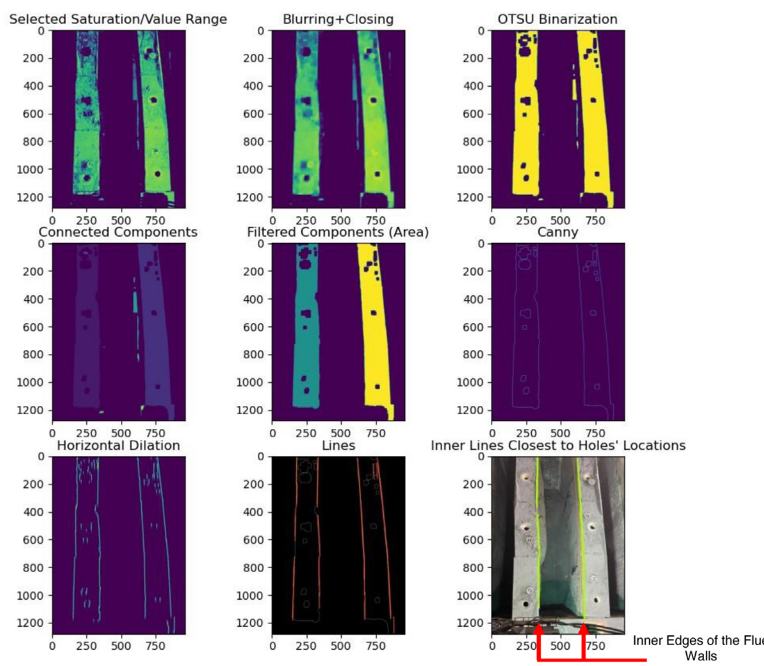

Fig. 23. Finding the inner edges of the flue walls. The red arrows point towards them in the final stage.

score reached 94.67%, and the prediction loss score reached low levels during training.

3.5. Pit segmentation and 3D modeling results

To test our proposed pit segmentation subsystem, we first start by extracting the top pit layer that detects peepholes and uses them to identify the flue wall pixels. Images taken on-site were input to the algorithm, processed using a computer vision pipeline that extracts the peepholes and provides a refined segmentation of the fluewalls. Fig. 22 shows the results obtained when finding the peepholes in a detected pit image. Morphological operations, OTSU binarization, area filtering, and canny edge detection were then used to detect the outer and inner edges of the left and right flue walls. The algorithm extracted the inner edges of the flue walls when applied to on-site images, as depicted in Fig. 23. We then extract the bottom pit layer using the bottom-pit segmentation algorithm detailed in Section 2.5.2. Fig. 24 shows the probability maps and visual comparison of our bottom-pit segmentation results with the ground truth. The overlap, the intersection area between the

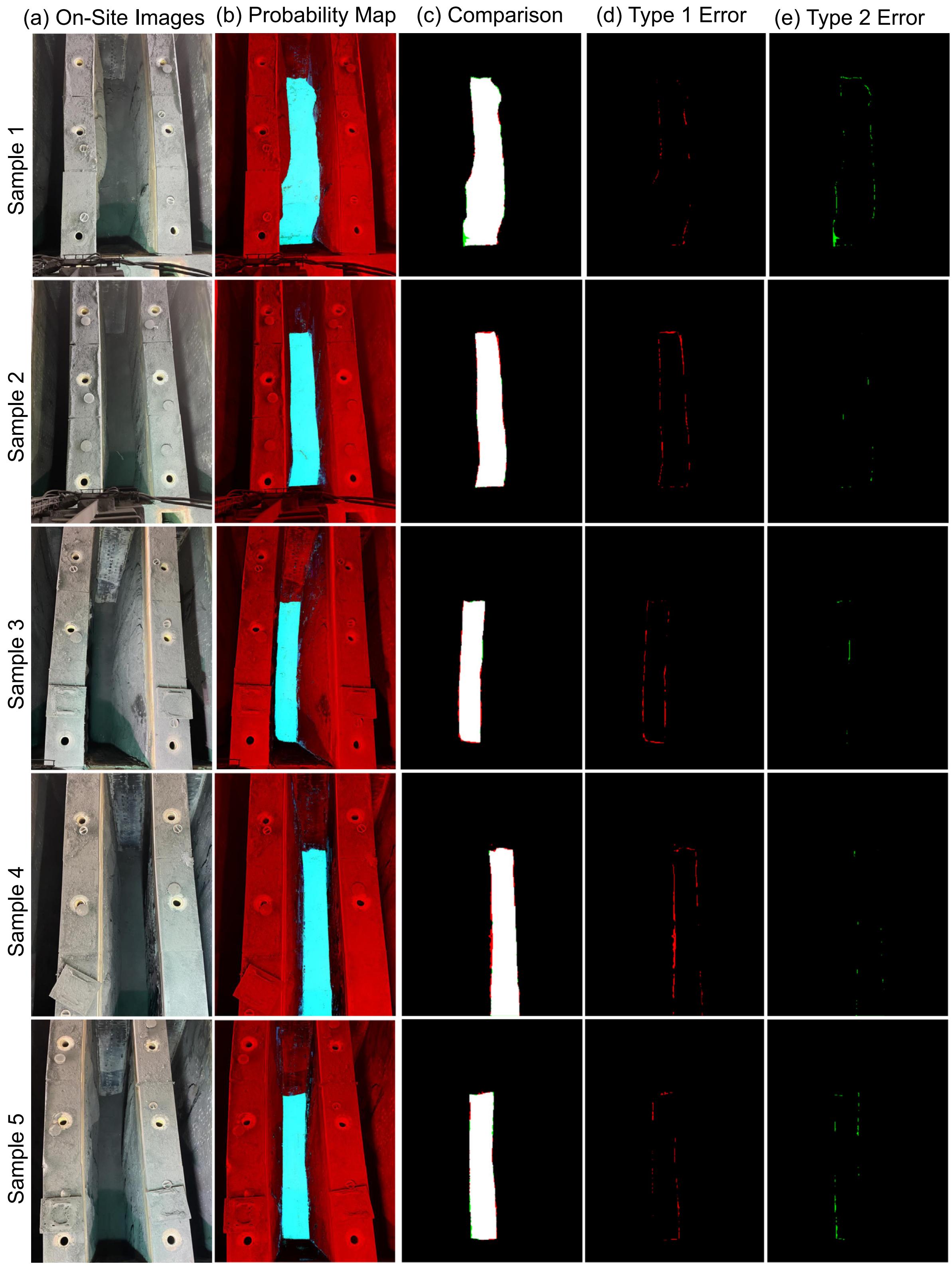

Fig. 24. Pit-bottom segmentation results. For each sample, (a) shows the image captured on-site, (b) illustrates the probability map, (c) shows a visual comparison between the ground truth and bottom segmentation outputs, Type 1 and Type 2 errors are illustrated in (d) and (e), respectively.

predicted and ground truth, is shown in white, indicating correctly classified bottom pixels. For visualization, we then overlay the actual and predicted segmentation bottom masks, as shown in Fig. 25.

To evaluate the performance of our segmentation algorithm, we computed several evaluation metrics. These metrics include accuracy, precision, sensitivity, specificity, overlaps, and DSC. The accuracy was

Table 1
Segmentation performance evaluation metrics.

Evaluation metric	Accuracy	Precision	Sensitivity	Specificity	Overlaps	DSC
Reported value	99.59%	96.57%	98.82%	99.67%	95.46%	97.67%

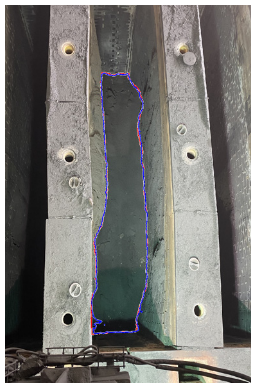

Fig. 25. An example for the actual mask vs. the predicted mask.

found to be 99.59%, indicating that our algorithm correctly classified 99.59% of the pixels. The performance metrics of the pit segmentation algorithm are detailed in Table 1.

Overall, the performance evaluation metrics obtained for the pit segmentation indicate that our algorithm is able to achieve high levels of accuracy, precision, sensitivity, specificity, overlaps, and DSC, which further validates the effectiveness of our proposed method.

Using the segmented layers, we then created a 3D model of the pit and measure the inner distance between the flue walls at each third of the pit's height. As shown in Fig. 26, the modeling algorithm was tested on data collected from the physical model built in the lab, synthetic pit images, and on-site images. As can be seen from the result, our algorithm was able to model the pits recorded during the on-site flight without prior training.

3.6. End-to-end integration testing and validation

We performed end-to-end integration testing in our lab environment and on-site. After our drone surveyor completed its monitoring flight, recorded videos and location data were uploaded to the cloud server for storage, processing, and analysis. Pits are detected, segmented, and a 3D model is generated for each pit. A report is then uploaded to the cloud and presented to the end users. The user interface integrated with our system is designed to facilitate monitoring the pits' condition by allowing operators to trace the actual temporal development of the pits and monitor deformations over time. In all testing scenarios, the floor map of the inspected area is first displayed, giving the user the liberty to investigate the condition of each pit independently and track it over time. Then, as the slider's value changes, our cloud server delivers a detailed report, including a light image with the classification class (normal, narrow, or wide), the depth image, and the hybrid view produced by overlaying the light and depth videos. This helps end-users analyze the pit health dispatched from the cloud server to the interface. It also allows the user to view the full inspection run video captured by the drone when needed and explore the 3D model of the pit in various directions and views. Fig. 27 shows our Graphical User Interface (GUI) results for an on-site flight.

The interface also allows users to take digital measurements to confirm the pit diagnosis and order maintenance after calibrating the drone camera and extracting the extrinsic and intrinsic parameters. Once the camera has been calibrated, the parameters transform image coordinates in pixels to real-world coordinates in meters, enabling accurate digital distance measurements. This calibration process must be done every time the camera or its environment is changed to ensure precise measurements. A sample shown in Fig. 28 shows that the pit width measured by the proposed system is 800.6 mm compared to the actual pit width of 800 mm, shown in Fig. 15.

4. Conclusions

In this paper, we proposed an AI-powered drone for automated health monitoring of anode baking furnaces that reduces inspection time to 12.5% and eliminates concerns over the safety of the previously manual process. Our drone uses a fusion of sensors for navigation and obstacle avoidance in a GPS-denied environment. After an autonomous inspection flight, the collected color and depth video frames are registered to synthesize multi-modal furnace mosaics. We then applied object detectors to localize pits and initially classify them into narrow, normal, or wide with a mean average precision of 94.7%. We overcame data scarcity in object detection using custom-designed data augmentation. Localizing pits allowed us to extract pit images, segment them, and blend them with depth images to produce 3D pit models used to collect digital measurements, visually localize and identify deformations, confirm the diagnosis, and perform predictive maintenance. Depth images are imprecise near the top and bottom layers. To precisely and accurately segment the pits, we proposed a two-stage segmentation technique for the top layer, followed by the bottom layer, and used them as anchors. For the initial top layer segmentation, we used the circle Hough transform to detect peel holes on the walls and used them to detect the inner wall boundaries using the line Hough transform. We then used position, color, and shape cues to build dual atlases for the wall and the floor of the pit using training images. We finally used the atlases to transform the color images into feature images to feed them to a deep semantic segmentation

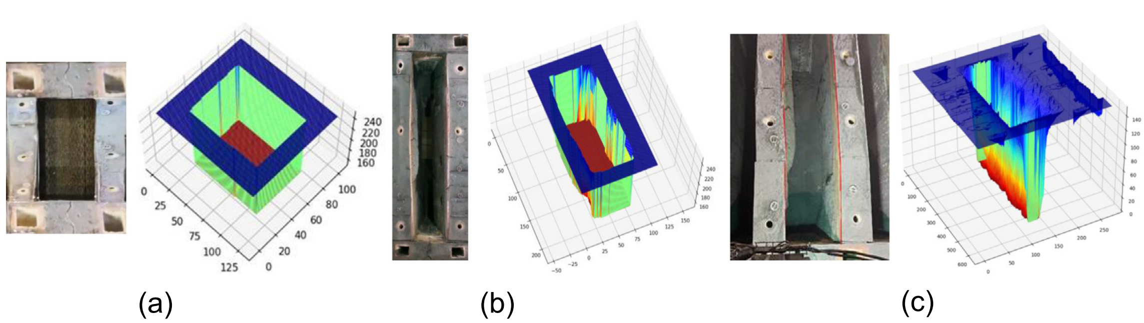

Fig. 26. Pit modeling results. (a) shows the 3D model generated for the physical model built in the lab, (b) illustrates the model generated for the synthetic pit, and (c) shows the model generated for the on-site pit image.

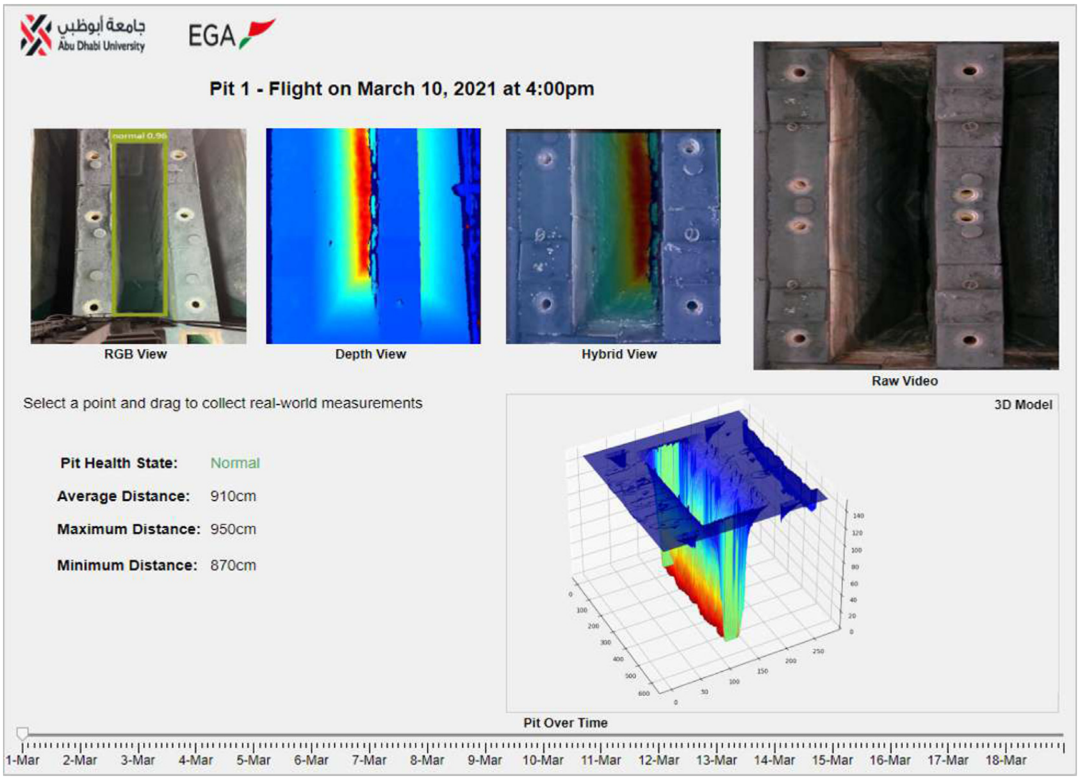

Fig. 27. User interface displaying a detailed report on the inspected area, the pit view, where the selected pit measurements, classification, visual view, inspection run video and the 3D model of the pit are displayed.

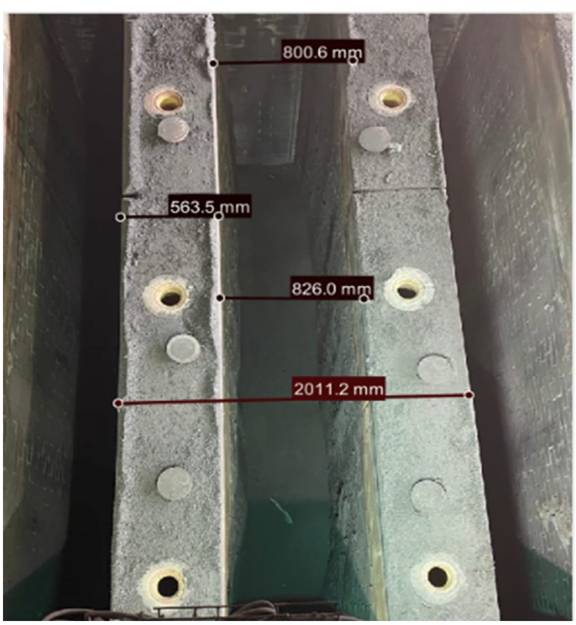

Fig. 28. Pit measurements taken by the user.

network for precise segmentation through pixel-level classification. Our segmentation has a precision of 96.67%. We overlay depth information in the remaining wall areas and synthesize a novel visual model of the pits accessible to the end user over the cloud using web and mobile front-ends. In the future, we plan to lower the drone into the pit to capture occluded areas.

CRediT authorship contribution statement

Tasnim Basmaji: Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Maha Yaghi:** Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Marah Alhalabi:** Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Abdallah Rashed:** Investigation, Methodology, Software, Validation, Writing – original draft. **Huma Zia:** Investigation, Supervision. **Mohamed Mahmoud:** Funding acquisition, Validation, Writing – review & editing. **Pragasan Palavar:** Funding acquisition, Validation, Writing – review & editing. **Sara Alkhadhar:** Funding acquisition, Validation, Writing – review & editing. **Halima Alhmoudi:** Funding acquisition, Validation, Writing – review & editing. **Mohammad Alkhedher:** Funding acquisition, Investigation, Methodology, Project administration, Software, Supervision. **Ayman Elbaz:** Investigation, Methodology, Software, Supervision. **Mohammed Ghazal:** Funding acquisition, Investigation, Methodology, Project administration, Software, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

Alireza, N., Sarim, M., Hashemi, M., Schnipke, E., 2015. Autonomous navigation of UAV through GPS-denied indoor environment with obstacles. In: AIAA Infotech at Aerospace. http://dx.doi.org/10.2514/6.2015-0715.

Araujo, J.O., Valente, J., Kooistra, L., Munniks, S., Peters, R.J.B., 2020. Experimental flight patterns evaluation for a UAV-based air pollutant sensor. Micromachines 11 (8).

Arkhipov, A., Zarouni, A., Baggash, M.J.I., Akhmetov, S., Reverdy, M., Potocnik, V., 2016. Cell electrical preheating practices at dubal. In: Hyland, M. (Ed.), Light Metals 2015. Springer, Cham, Switzerland, pp. 445–449. http://dx.doi.org/10.1007/978-3-319-48144-9_75.

Baiocchi, V., Dominici, D., Mormile, M., 2013. UAV application in post-seismic environment. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 2, 21–25.

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (SURF). Comput. Vis. Image Underst. 110 (3), 346–359.

Bezas, K., Tsoumanis, G., Angelis, C.T., Oikonomou, K., 2022. Coverage path planning and point-of-interest detection using autonomous drone swarms. Sensors 22 (19).

Botterill, T., Mills, S., Green, R., 2010. Real-time aerial image mosaicing. In: International Conference Image and Vision Computing New Zealand. http://dx.doi.org/10.1109/IVCNZ.2010.6148850.

Brown, M., Lowe, D.G., 2007. Automatic panoramic image stitching using invariant features. Int. J. Comput. Vis. 59–73.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder–decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 801–818.

Cheng, Z., Li, S., Rustam, S., 2020. Simultaneous material segmentation and 3D reconstruction in industrial scenarios. Front. Robot. AI 7, http://dx.doi.org/10.3389/frobt.2020.00052.

Civera, J., Davison, A.J., Magalln, J.A., Montiel, J.M., 2009. Drift-free real-time sequential mosaicing. Int. J. Comput. Vis. 81 (2), 128–137.

Ding, C., Liu, H., Li, H., 2019. Stitching of depth and color images from multiple RGB-D sensors for extended field of view. Int. J. Adv. Robot. Syst. 16 (3), http://dx.doi.org/10.1177/1729881419851665.

2023. Osmo Pocket - DJI. DJI Official, [Online]. Available: https://www.dji.com/ae/osmo-pocket.

Doutre, C., Nasiopoulos, P., 2009. Fast vignetting correction and color matching for panoramic image stitching. In: Proc. IEEE Int. Conf. Image Process.. pp. 709–712.

Eschmann, C., Kuo, C., Boiler, C., 2012. Unmanned aircraft systems for remote building inspection and monitoring. In: Proceedings of the Sixth European Workshop on Structural Health Monitoring, Dresden, Germany. pp. 3–6.

2023. Commercial drone market size, share & covid-19 impact analysis, by weight ((no date) commercial drone market size, share | forecast report [2029]. Available at: https://www.fortunebusinessinsights.com/commercial-drone-market-102171.

Ghazal, M., Basmaji, T., Yaghi, M., Alkhedher, M., Mahmoud, M., El-Baz, A.S., 2020. Cloud-based monitoring of thermal anomalies in industrial environments using AI and the internet of robotic things. Sensors 20 (21).

2023. Drone services market report 2016-2024: industry demand, scope, trends, growth driver, opportunities & ; future prospects, revenue data (no date) goldstein market intelligence. Available at: https://www.goldsteinresearch.com/report/drone-services-market-outlook-2024-global-opportunity-and-demand-analysis-market-forecast-2016-2024.

Hartung, J., Dold, P.M., Jahn, A., Heizmann, M., 2022. Analysis of AI-based single-view 3D reconstruction methods for an industrial application. Sensors 22 (6425), http://dx.doi.org/10.3390/s22176425.

Hayat, A., Morgado-Dias, F., 2022. Deep learning-based automatic safety helmet detection system for construction safety. Appl. Sci. 12 (16), http://dx.doi.org/10.3390/app12168268.

Hoang, V.T., Phung, M.D., Dinh, T.H., Ha, Q.P., 2020. System architecture for real-time surface inspection using multiple UAVs. IEEE Syst. J. 14 (2), 2925–2936. http://dx.doi.org/10.1109/JSYST.2019.2922290.

2023. Depth camera D435. In: Intel® RealSense™ Depth and Tracking Cameras. [Online]. Available: https://www.intelrealsense.com/depth-camera-d435/.

Jayawardena, S., Yang, D., Hutter, M., 2012. 3D model assisted image segmentation. In: Proceedings - 2011 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2011. http://dx.doi.org/10.1109/DICTA.2011.17.

Jin, H.L., 2008. A three-point minimal solution for panoramic stitching with lens distortion. In: Proc. Comput. Vis. Pattern Recog.. pp. 1–8.

Jocher, Glenn, 2022. Ultralytics/Yolov5: V7.0 - YOLOv5 SOTA Realtime Instance Segmentation. Zenodo, http://dx.doi.org/10.5281/zenodo.7347926.

Ju, M.H., Kang, H.B., 2010. A new simple method to stitch images with lens distortion. In: Proc. Adv. Vis. Comput.. pp. 273–282.

Koppány, M., Lucian, B., 2015. Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection. Sensors 15 (7), 14887–14916. http://dx.doi.org/10.3390/s150714887.

Kroll, B., Schaffranek, D., Schriegel, S., Niggemann, O., 2014. System modeling based on machine learning for anomaly detection and predictive maintenance in industrial plants. In: Proceedings of the IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, Spain. pp. 1–7. http://dx.doi.org/10.1109/ETFA.2014.7005202.

Kubota, T., Yamamoto, W., 2019. Anomaly detection from online monitoring of system operations using recurrent neural network. Procedia Manuf. 30, 83–89. http://dx.doi.org/10.1016/j.promfg.2019.02.013.

Li, W., Zhang, H., Wang, G., Xiong, G., Zhao, M., Li, G., Li, R., 2023. Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing. Robot. Comput.-Integr. Manuf. 80, http://dx.doi.org/10.1016/j.rcim.2022.102470.

Li, H., et al., 2017. Real-time RGB-D image stitching using multiple kinects for improved field of view. Int. J. Adv. Robot. Syst. 2 (14), http://dx.doi.org/10.1177/1729881417695560.

Lovegrove, S., Davison, A.J., 2010. Real-time spherical mosaicing using whole image alignment. In: Proc. Eur. Conf. Comput. Vis.. pp. 73–86.

Metni, N., Hamel, T., 2007. A uav for bridge inspection: Visual servoing control law with orientation limits. Autom. Constr. 17 (1), 3–10.

2023. Voxl m500 - Development Drone for PX4 GPS-Denied Navigation and Obstacle Avoidanceusb/debug Add-On / Spektrum SPM9X45 Receiver. ModalAI, Inc., [Online]. Available: https://www.modalai.com/products/voxl-m500?variant=41019913764915.

Morgenthal, G., Hallermann, N., 2014. Quality assessment of unmanned aerial vehicle (uav) based visual inspection of structures. Adv. Struct. Eng. 17 (3), 289–302.

Nikolic, J., Burri, M., Rehder, J., Leutenegger, S., Huerzeler, C., Siegwart, R., 2013. A UAV system for inspection of industrial facilities. In: Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, Vol. 2. pp. 1–8, (9).

Pulli, K., Tico, M., Xiong, Y., 2010. Mobile panoramic imaging system. In: Proc. ECVW. pp. 108–115.

Rathinam, S., Kim, Z., Sengupta, R., 2008. Vision-based monitoring of locally linear structures using an unmanned aerial vehicle. J. Infrastr. Syst. 14 (1), 52–63.

2023. Wiki. ros.org [Online]. Available: http://wiki.ros.org/mavros.

Saeed, F., Ahmed, M.J., Gul, M.J., et al., 2021. A robust approach for industrial small-object detection using an improved faster regional convolutional neural network. Sci. Rep. 11 (23390), http://dx.doi.org/10.1038/s41598-021-02805-y.

Shen, Y., Zhu, J., Liu, H., Sun, L., 2018. Plant image mosaic based on depth and color dual information feature source from kinect. Trans. Chin. Soc. Agric. Eng. 34, 176–182.

Stauffer, C., Grimson, W., 2007. Adaptive background mixture models for real-time tracking. In: Proceedings of IEEE Conf. Computer Vision Patt. Recog., Vol. 2.

Szeliski, R., 2006. Image Alignment and Stitching: A Tutorial. Now Publishers, Hanover, MA.

Tews, A., Bosse, M., Zlot, R., Flick, P., Noonan, M., 2013. Laser mapping of carbon bake furnaces. In: Sadler, B.A. (Ed.), Light Metals. In: The Minerals, Metals & Materials Series, Springer, Cham, http://dx.doi.org/10.1007/978-3-319-65136-1_198.

Tkac, M., Ruud, A., Holden, I., Linga, H., 2016. Hydro aluminium's historical evolution of closed type anode baking furnace technology. Light Metals 2013, 1133–1138.

Wang, L., Zhang, Z., 2017. Automatic detection of wind turbine blade surface cracks based on uav-taken images. IEEE Trans. Ind. Electron. 64 (9), 7293–7303.

Xia, Y., Chen, C., Shi, J., Liu, Y., Li, G., 2020. Two-layer path planning for multi-area coverage by a cooperated ground vehicle and drone system. http://dx.doi.org/10.36227/techrxiv.13158443.

Xu, W., Mulligan, J., 2010. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In: Proc. Comput. Vis. Pattern Recog.. pp. 263–270.

Yang, W., Zheng, J., Cai, J., Rahardja, S., Chen, C.W., 2009. Natural and seamless image composition with color control. IEEE Trans. Image Process. 18 (11), 2584–2592.

Yao, H., Qin, R., Chen, X., 2019. Unmanned aerial vehicle for remote sensing applications—A review. Remote Sens. 11 (12).

Zaidani, M., Qureshi, A., Shamim, Z., Abu Al-Rub, T., Rashid, K., 2018. Investigating the flue-wall deformation effects on performance characteristics of an open-top aluminum anode baking furnace. Appl. Energy 231, 1033–1049, Elsevier.

Zhi, Q., Cooperstock, J.R., 2008. Depth-based image mosaicing for both static and dynamic scenes. In: 19th International Conference on Pattern Recognition, Tampa, FL, USA. pp. 1–4. http://dx.doi.org/10.1109/ICPR.2008.4761469.

Zhi, Q., Cooperstock, J.R., 2012. Toward dynamic image mosaic generation with robustness to parallax. IEEE Trans. Image Process. 21 (1), 366–378. http://dx.doi.org/10.1109/TIP.2011.2162743.

Zhou, D., Wei, M., Si, X., 2014. A survey on anomaly detection, life prediction and maintenance decision for industrial processes. Acta Autom. Sin. 39, 711–722. http://dx.doi.org/10.3724/SP.J.1004.2013.00711.

Zou, Z., Shi, Z., Guo, Y., Ye, J., 2019. Object detection in 20 years: A survey. arXiv preprint arXiv:1905.05055.