Received 16 August 2023, accepted 3 September 2023, date of publication 11 September 2023,
date of current version 20 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3314337

RESEARCH ARTICLE

A Hand-Drawn Language for Human–Robot Collaboration in Wood Stereotomy

CRISTHIAN A. AGUILERA-CARRASCO[1], (Member, IEEE), **LUIS FELIPE GONZÁLEZ-BÖHME**[2], **FRANCISCO VALDES**[3], **FRANCISCO JAVIER QUITRAL-ZAPATA**[2], AND **BOGDAN RADUCANU**[4]

[1]Facultad de Ingeniería, Arquitectura y Diseño, Universidad San Sebastián, Puerto Montt 5501842, Chile
[2]Department of Architecture, Universidad Técnica Federico Santa María, Valparaíso 2390123, Chile
[3]CIPHER Laboratory, Georgia Tech Research Institute, Atlanta, GA 30309, USA
[4]Computer Vision Center (CVC), 08193 Barcelona, Spain

Corresponding author: Cristhian A. Aguilera-Carrasco (cristhian.aguilera@uss.cl)

This work was supported in part by the National Research and Development Agency through the FONDEF Project under Grant ID20I10262, in part by Patagón Supercomputer of Universidad Austral de Chile under Grant FONDEQUIP EQM180042, and in part by Vicerrectoría de Investigación y Doctorados from Universidad San Sebastián under Grant FONDO VRID_APC23/15.

ABSTRACT This study introduces a novel, hand-drawn language designed to foster human-robot collaboration in wood stereotomy, central to carpentry and joinery professions. Based on skilled carpenters' line and symbol etchings on timber, this language signifies the location, geometry of woodworking joints, and timber placement within a framework. A proof-of-concept prototype has been developed, integrating object detectors, keypoint regression, and traditional computer vision techniques to interpret this language and enable an extensive repertoire of actions. Empirical data attests to the language's efficacy, with the successful identification of a specific set of symbols on various wood species' sawn surfaces, achieving a mean average precision (mAP) exceeding 90%. Concurrently, the system can accurately pinpoint critical positions that facilitate robotic comprehension of carpenter-indicated woodworking joint geometry. The positioning error, approximately 3 pixels, meets industry standards.

INDEX TERMS Computer vision, cooperative systems, hand-drawn language, human–robot interaction, robot learning, timber-joinery layout, wood stereotomy.

I. INTRODUCTION

Wood stereotomy, the process of producing wooden structures for assemblablity requires great skill and is a key aspect of carpentry and joinery. Among others, this process involves two critical tasks: layout and cutting. Layout is a highly skilled process that involves locating and marking woodworking joints, such as mortise-and-tenon joints, that connect timbers in a timber frame [1]. Cutting involves using saws, drills, and chisels to shape the wood according to the marked layout. Traditionally, due to the expertise involved, senior carpenters predominantly handle the layout, while cutting the joints becomes a collective effort, with everyone contributing [2].

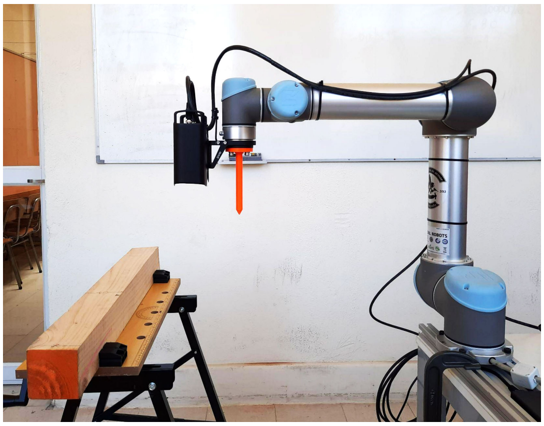

FIGURE 1. A proof-of-concept prototype that reads hand-drawn lines and symbols on a workpiece to locate and classify cutting operations.

Motivated by the need to reposition wood stereotomy as the most sustainable wood construction method and overcoming the challenges posed by automation in traditional trades, there has been growing interest in exploring how industrial

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Liu.

robots can work collaboratively with skilled carpenters and joiners. By combining robot precision and speed with human creativity and experience-based decision making, there is room for significant improvements in the wood stereotomy process that can potentially expand the possibilities of both the trade and the construction industry. Thus, effective communication between skilled workers and robots is crucial for successful collaboration in wood stereotomy.

Traditionally, carpenters and joiners have used drawings to convey instructions to their colleagues regarding where to make cuts and how to organize their tasks. However, recent advancements in computer vision and machine learning have opened up new possibilities for utilizing visual cues to enable robots to understand the intentions of human carpenters and joiners, making this relationship more efficient and productive.

Recent cutting-edge research by Pedersen and Reinhardt [3], [4] has delved into the application of computer vision and machine learning for the autonomous interpretation of instructions from experts in hand-drawn digital fabrication tasks. These investigations have made significant strides by highlighting the critical role of incorporating additional semantic content into visual cues and stressing the need to create visual languages that foster efficient communication between humans and robots. Despite the undeniable value of these studies, there is still a necessity to develop task-specific languages tailored to unique applications, such as wood-stereotomy. Moreover, it is essential to devise more quantifiable approaches to assess the effectiveness of these languages under real-world challenges, including varying materials, lighting conditions, and other factors.

Our work proposes a novel approach to human-robot collaboration in wood stereotomy by presenting a hand-drawn language inspired by traditional techniques from various cultures worldwide. This language includes a richer vocabulary that allows for a broader range of operations and provides solutions for handling mistakes made by skilled workers, which are difficult to erase. We generate a large dataset with more than 600 images, including various materials and drawers, which serve as a benchmark for future work in this field. To evaluate the effectiveness of our proposal, we also propose and develop a computer vision solution with its corresponding metrics to assess its suitability. We posit that collaborative human-robot wood stereotomy has the potential to integrate time-tested, sustainable building techniques into Construction 4.0. Our proof-of-concept prototype is depicted in Figure 1.

This study presents several contributions to the field of wood stereotomy and human-robot collaboration, including:

- Generation of a novel dataset of hand-drawn timber-joinery layouts, drawn by six different individuals on six distinct wood species' sawn surfaces.
- Development of a hand-drawn language specifically designed for human-robot collaboration in wood stereotomy, capable of being interpreted by both carpenters and robots utilizing computer vision techniques.
- Empirical assessment of the proposed hand-drawn language and computer vision solution, demonstrating the validity of the approach and highlighting potential future challenges.
- Provision of all necessary code for the replication of the study's results, including the novel dataset.[1]

The rest of the manuscript is organized as follows: Section II provides an overview of the state of the art and related work, Section III describes the characteristics of timber-joinery layout, Section IV describes the proposed hand-drawn language, Section V describes the computer vision approach, Section VI describes the construction of the dataset used in this work, Section VII presents the experimental results, and Section VIII describes our conclusions and future work.

II. LITERATURE REVIEW

A. HUMAN-ROBOT COLLABORATION IN WOOD STEREOTOMY

An increasing number of small and medium-sized manufacturers recognize the potential productivity benefits of integrating humans' problem-solving skills and creativity with the strength and repeatability of industrial robots to tackle ill-structured problems. However, to achieve this goal, effective communication channels must be established between humans and robots that are easily understandable by both parties. Failure to do so can lead to frustration and inefficiencies in human-robot collaboration (HRC) [5].

According to Sziebig [6], mastering the communication challenges involved in working with these new types of *workers* will enable us to create a new type of *colleagues*. Several methods are available to support HRC communication, including gestures (e.g. [7]), speech (e.g., [8]), and computer vision techniques (e.g.l [4]), such as the method employed in this paper, which uses draws and cameras to facilitate communication between humans and robots. However, selecting the appropriate communication method depends on the specific HRC application, and safety remains the primary concern at all times.

Research in HRC for carpentry tasks has primarily focused on the use of robots for the assembly of timber structures. Numerous studies have explored this area, such as Kramberger et al. [9], Zhang et al. [10], Kunic et al. [11], Wang and Wang [12], Kyjanek et al. [13], Devadass et al. [14], and Stumm et al. [15]. These studies typically involve the on-site manipulation of wooden building components, with communication between humans and robots facilitated by offline programming and teach programming through Augmented and Virtual Reality media. On the other hand, Solvang et al. [16] and Sziebig et al. [6] have taken a different approach by exploring HRC in manufacturing processes like machining, grinding, and deburring. They use hand-drawn sketches on the workpiece to communicate the tool path to an industrial robot.

[1] https://github.com/ngunsu/hand-drawn-article/tree/main

Pedersen et al. [4] have proposed a novel method for providing visual feedback to robotic fabrication by detecting hand-drawn markings on objects, which uses a camera to recognize closed/open curves or lines representing cut lines that a robot system can execute. In a more recent work [3], the same team emphasizes the importance of using a visual language for communication through drawings, rather than isolated symbols. However, despite its innovation, their approach lacks certain crucial elements required to determine its suitability for use in wood stereotomy. For example, it would benefit from a more diverse set of samples, different materials, a quantifiable analysis of results, and a more comprehensive language that aligns with carpenters' existing knowledge and allows the robot to automatically analyze the entire workpiece, regardless of its orientation and position.

Our work introduces a comprehensive language that draws from traditional timber-joinery layout lines and symbols, which would be familiar to carpenters. By adopting such a rich language, we can expand the range of operations that a vision-based collaborative industrial robot system can perform in wood stereotomy.

B. COMPUTER VISION AND MACHINE LEARNING

It is crucial to apply computer vision and machine learning techniques to achieve precise comprehension and interpretation of hand-drawn languages. By leveraging these techniques, robots can accurately comprehend and integrate the cutting instructions that human partners draw on the workpiece by hand. Thus, it is necessary to thoroughly review techniques such as object detection, image segmentation, and keypoint regression to comprehend the proposed solution that we will present in section V.

Object detection involves identifying and categorizing one or more objects in an image, which can be visually distinct from their background, such as a car, an apple, or in our case, a particular type of sketch. Typically, an object detector scans a list of pre-trained objects and returns each detected object's label and bounding box coordinates, usually defined by four coordinates, $x1$, $y1$, $x2$, and $y2$, that form a rectangular shape around the object.

There are two main types of learning-based object detection methods: one-stage and two-stage. One-stage methods, including YOLO [17], and others like SSD [18], tend to be faster but less accurate than two-stage methods such as Mask R-CNN [19] and Cascade R-CNN [20], as mentioned in [21]. YOLO, a popular one-stage method, involves training a single convolutional neural network to predict a fixed number of bounding boxes across an input image. After predicting the bounding boxes, the result is filtered to avoid redundant detections. YOLO achieves a balance between speed and accuracy, particularly in newer versions such as [22], [23], and [24].

Keypoint regression serves as an essential method for identifying specific feature-related coordinate pairs (x, y) of an object. For instance, in facial recognition domains [25], this technique aids in pinpointing facial landmarks. In our context, it can help determine drilling locations. The application of keypoint regression needs solving a regression problem, with the quantity of keypoints per object being a configurable parameter. A straightforward implementation can be realized via a classifier such as [26] and [27], with the output calibrated to yield a specified number of coordinates for regression analysis. Nonetheless, certain methodologies may incorporate supplementary steps to enhance the precision of keypoint locations. For example, in [28], the authors introduced a pose estimation strategy utilizing pose refinement to generate more accurate 2D image poses.

Lastly, it is important to delve into image segmentation techniques, which assign unique labels to each pixel in an image, facilitating differentiation between hand-drawn sketches and wood textures. A fundamental strategy is color segmentation, focusing on filtering a specific color range. Despite being simple and effective, this method lacks robustness, often necessitating recalibration in response to varying environmental conditions [4]. Alternatively, more sophisticated solutions use deep learning to distinguish between categories [29]. Although these methods exhibit higher resilience to lighting variations, they necessitate extensive pixel labeling, which may not be always practical. In this article, we put forth an alternative grounded in traditional techniques, demonstrating robustness to varying illumination changes and requiring less frequent manual calibration

III. TIMBER JOINERY LAYOUT

According to Beemer [30], layout is the handcrafted method used by carpenters for locating and marking each woodworking joint in each member of a wooden structure. Carpenters often use timbers that vary from nominal dimensions and may be out of square. Timbers may be unseasoned and change shape over time. In Western culture, there are mainly two distinct layout approaches known as the *scribe rule* and *square rule* to work through these irregularities. Timber layout is also a communication system between colleagues. The graphic language of the carpenters consists of lines and symbols drawn or marked on the faces of the workpieces. Some symbols indicate the location and orientation of each timber in the frame so that any colleague on site will know.

Carpenters use a unique set of symbols (similar to an alphabet) and lines to draw the layout, as seen in Figure 2. While these symbols may vary slightly from country to country, their purpose remains consistent, which is to identify the position and orientation of each part within the final frame assembly while the workpiece is on the workbench (sawhorses). Hand-drawn marks on the workpiece's faces indicate where waste needs to be cut and where holes should be drilled. Additionally, carpenters use numerals, letters, and special symbols to communicate a visual graphic language to other carpenters and apprentices about the local coordinate system. Timber layout allows carpenters to adapt the joinery to each workpiece's individual imperfections, deformations,

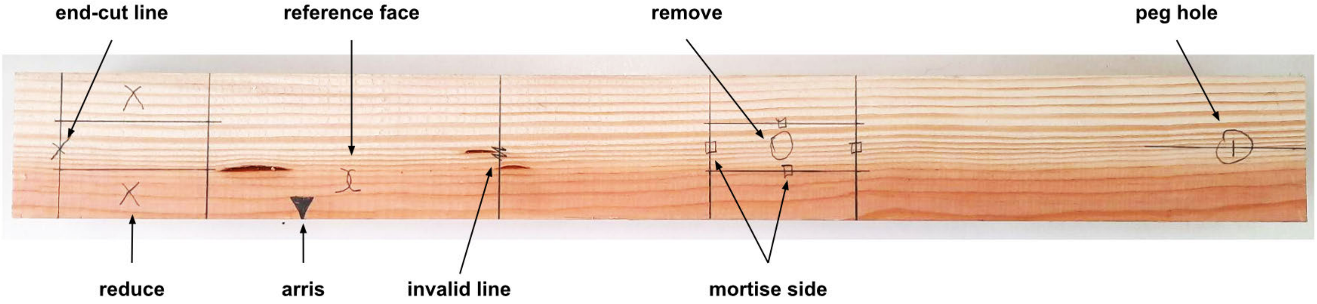

FIGURE 2. Example of a timber joinery layout using our proposed hand-drawn language.

and dimensional variations. Therefore, carpenters may use different measurement systems, such as the *square rule* or *scribe rule*, depending on how regular or irregular the workpiece is.

In the next section, we describe a hand-draw language for human-robot collaboration on wood stereotomy, using traditional carpenters' alphabets as the base.

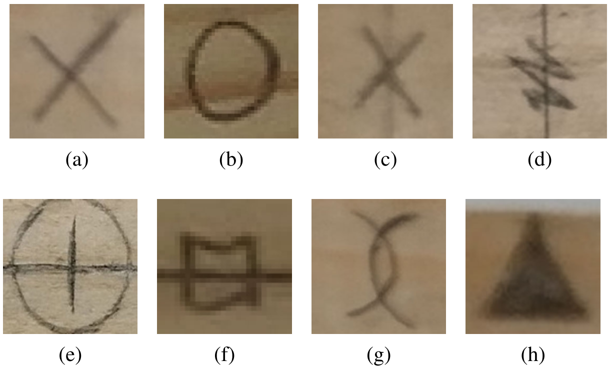

FIGURE 3. Layout symbols (a) ***reduce***, (b) ***remove***, (c) ***end-cut line***, (d) ***invalid line***, (e) ***peg hole***, (f) ***mortise side***, (g) ***reference face***, and (h) ***arris***.

IV. PROPOSED HAND-DRAWN LANGUAGE

Our proposed visual language is composed of eight symbols and lines, as illustrated in Figure 3. The symbols help identify the coordinate system of the workpiece, and each line or segment's purpose. The combination of these lines, symbols, and strokes defines the cutting operations required on the workpiece. The symbols selected comprise the smallest possible set needed for wood stereotomy and are currently used by carpenters. This choice was based on evidence from both literature and practice [1]. The eight symbols in the language are:

- *Reduce* (see Figure 3a): The *X* symbol within an area bounded by lines or the edge of the workpiece indicates where to reduce a cross-sectional area of the workpiece. Figure 2 shows the instruction *cut the tenon cheeks (broadsides) here*
- *Remove* (see Figure 3b): The *O* symbol within an area enclosed by lines indicates where to completely remove a cross-sectional area of the workpiece. Figure 2 shows the instruction *cut a through mortise here*.
- *End-cut line* (see Figure 3c): The *X* symbol overlaid on a line indicates where to cut off the end of the workpiece. Figure 2 shows the instruction *cut off the workpiece here*.
- *Invalid line* (see Figure 3d): The zigzag symbol overlaid on a line indicates that this line is useless. Figure 2 shows the instruction *ignore this line*.
- *Peg hole* (see Figure 3e): The symbol of a cross inside a circular stroke indicates where to drill a hole for a peg. Figure 2 shows the instruction *drill a hole here*. The center of the cross inside the circular stroke indicates the precise location for drilling.
- *Mortise side* (see Figure 3f): The symbol of a little rectangle overlaid on a line segment indicates that the line segment represents one side of a mortise. Figure 2 shows the instruction *cut a through mortise here*. This symbol is not traditional and was created by the authors to make it easier for the robot to recognize the sides of a rectangle.
- *Reference face* (see Figure 3g): The Coco Chanel-like symbol indicates the main surface (usually receiving the floor or wall and roof sheathing) on a workpiece from which measurements are taken for the layout. In general, each workpiece has two reference faces that are adjacent and square to each other. Figure 2 shows the instruction *this side up or out*.
- *Arris* (Figure 3h): The arrowhead symbol indicates the edge along which two adjacent reference faces of the workpiece meet. Figure 2 shows the instruction *this arris up and out*

In the language we propose, symbols, lines, and segments can be placed on any side of the workpiece. However, the side showcasing the reference face is the richest in terms of information and serves as the starting point for extracting the 3D details of each task. Once the reference face is processed, it is critical to collate additional information from the remaining sides to ascertain the depth, inclination, and shape of the task. To achieve this, we use the same symbols, lines, and segments on the other sides, but as a 2D projection of the whole task.

For clarity, let us consider the example illustrated in Figure 4. In this scenario, two peg holes can be employed

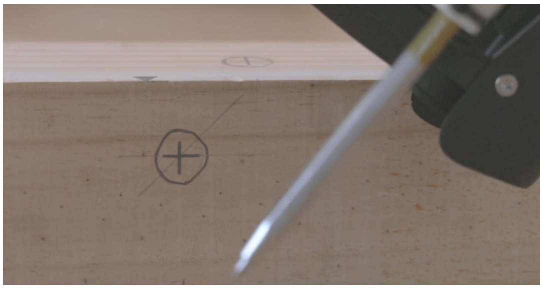

FIGURE 4. This image illustrates how two peg holes can be utilized to determine the angle of attack for drilling a hole in the workpiece.

to set the angle and distance the robot-tool must adopt when drilling a hole. The top peg hole indicates the starting point of the drilling, while the lateral peg hole represents the projection of the distance and angle that the drilling tool must adhere to. This example showcases how we apply the same approach to other symbols, utilizing the projection technique to obtain the necessary information.

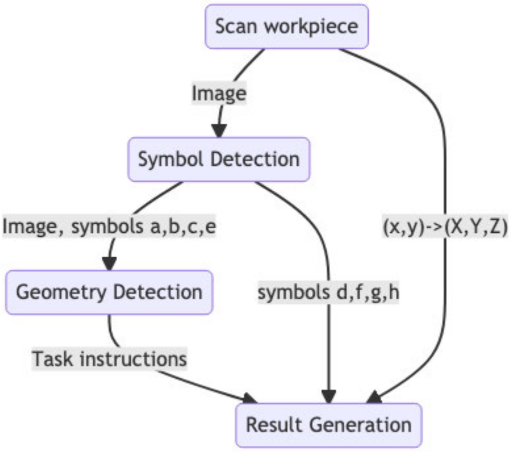

FIGURE 5. Computer vision pipeline.

V. COMPUTER VISION APPROACH

Our proposed solution leverages computer vision to automatically detect and recognize symbols, lines, and segments present on a workpiece. These elements encapsulate crucial information pertinent to the task at hand. The solution unfolds in a four-step process, ultimately resulting in the information being fed into the robot's software to perform the corresponding tasks. Figure 5 provides a visual representation of this pipeline.

Step 1: In the first step of our proposed process, the entire workpiece is scanned using a depth camera. The camera generates a point cloud representation, which maps the 2D pixel coordinates (x, y) of the image I to their corresponding 3D positions (X, Y, Z) with respect to the robot's manipulator arm. If the workpiece is larger than the camera's field of view, the robot must take multiple shots from different positions to obtain a complete representation. Additionally, this step allows us to segment the workpiece from the workbench by filtering by distance, as we assume that the position and size of the workbench are known. The output of this step is a fully mapped workpiece, with 2D to 3D mapping $(x, y) \rightarrow (X, Y, Z)$. In our proposal, the only face of the workpiece that is not mapped is the one facing the workbench. However, this face is not essential to our solution.

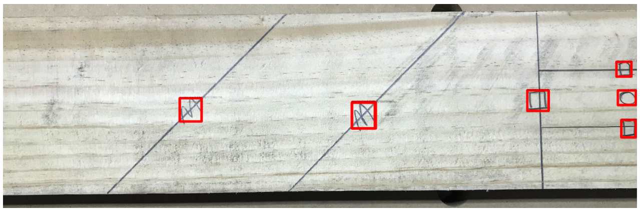

FIGURE 6. Example of symbol detection on a workpiece.

Step 2: In the second step of our proposed process, we aim to detect symbols as illustrated in Figure 6. These symbols can either represent instructions, as shown by symbols (a, b, c, e) in Figure 3, or serve as complementary information, as depicted by symbols (d, f, g, h) in the same figure. To achieve this, we employ a CNN-based object detector. This detector yields a list of 2D bounding boxes (pixel coordinates) accompanied by their respective labels and confidence scores, which assist in filtering out improbable symbols. In Section VII, we train and evaluate three distinct object detector solutions, including their variations, to assess the effectiveness of our proposal. It is important to remember that when selecting an object detector for robotic applications, a balance between speed and accuracy is essential; two crucial criteria for real-time performance.

Step 3: During the third step, we pinpoint the reference face and all its associated symbols. Out of the detected symbols, we are primarily interested in the symbols that indicate a task instruction, such as cutting, reducing, or similar. Following this, we extract the task-specific details for each such symbol. This step encompasses two potential scenarios.

- **If it is a peg hole symbol**: Initially, we locate the corresponding peg hole on one of the lateral sides. Using a CNN-based regression model, we pinpoint the central point of the cross within the peg holes. The regression model is tailored to ascertain the central point's normalized 2D coordinates (x, y), with values ranging between 0 and 1. We transform a generic CNN-based image classification model to create this regression model, altering its output layer to accommodate two features. These features indicate the coordinates x and y of the peg hole. For training, the CNN regression model uses cropped images of the peg holes from the ground truth. During inference, the cropped images are from the bounding boxes detected in the preceding step. In our exploration of model options, our focus remains on efficiency, emphasizing compact and fast models in both training and inference. A detailed evaluation of different models is presented in Section VII.

- **If it is a reduce, remove, or end-cut line symbol**: In this scenario, the system is required to distinguish between the lines drawn by a human operator and the inherent texture of the wood. While traditional color segmentation techniques based on color range have been used for this task (e.g., [4]), they often require manual calibration due to variations in perceived color resulting from different locations and times of the day. In contrast, we propose a similar yet more illumination-invariant approach requiring less frequent recalibration. Our methodology involves *k*-means color segmentation [31], effectively differentiating lines from the wood texture.

  We use the bounding box information from step 2 to automatically differentiate the colors of the lines and the background using color clustering. Given that the bounding boxes of the symbols contain more background than lines or strokes, we use two clusters to separate the lines from the background. The cluster with fewer pixels is identified as stroke, with the other cluster representing the background. We segment the entire workpiece by utilizing these established clusters, assigning each pixel to its nearest cluster. For instance, in Figure 7, we use a peg hole symbol to automatically identify the two color clusters. The image on the left displays the original bounding box, while the one on the right shows the segmented image, revealing two clusters: yellow and purple pixels. Since the yellow cluster has fewer pixels, it is identified as the drawn lines.

  Upon completion of *k*-means clustering, the workpiece is fully segmented. We then leverage computer vision tools like contour detection to determine the details of each task, similar to previous aproaches [4]. For instance, we identify the contour enclosing the symbol for tasks requiring reduction or removal. We detect the line passing through the symbol for the end-cut line, a task accomplished using Hough Line Transform.

  While this method is suitable, it is robust under specific conditions. Its effectiveness heavily relies on a high contrast between the drawings and the wood texture, which can be achieved using stroke colors that starkly contrast the workpiece. This process is applied to all faces involved in the task.

Step 4: The final stage of the process entails using all the information collected in the previous steps to generate the tasks that the robot must execute. The deep camera's output plays a crucial role in this step, as it supplies a normals data point cloud that allows us to determine each task's details with high accuracy. Since our primary focus is on wood stereotomy, the majority of tasks can be predefined offline, necessitating only the adjustment of parameters based on the locations identified on the workpiece.

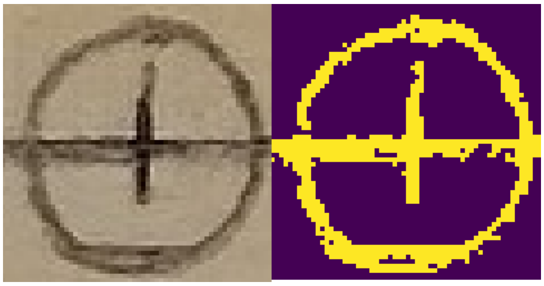

FIGURE 7. Example of *k*-means color clustering: On the left is the original image, while on the right is the segmented image, achieved through the use of two clusters.

VI. DATASET GENERATION

The proposed visual language was employed to produce timber layouts across more than eight different wood species, such as pinus radiata and lenga. A team of six individuals was tasked with drawing symbols, lines, and segments on each workpiece. Our strategy involved combining layouts from different team members and wood species to generate a varied dataset that can assist in evaluating the effectiveness and robustness of the suggested computer vision solution and future ones.

Our process for generating each sample in the dataset was methodical. Each workpiece was within the length range of 30-90 cm and had a maximum width of 60 cm. To ensure a well-balanced dataset, we maintained a consistent number of symbols per sample. Among the symbols, only the borderline symbol had a higher occurrence, appearing four times more frequently than the other symbols.

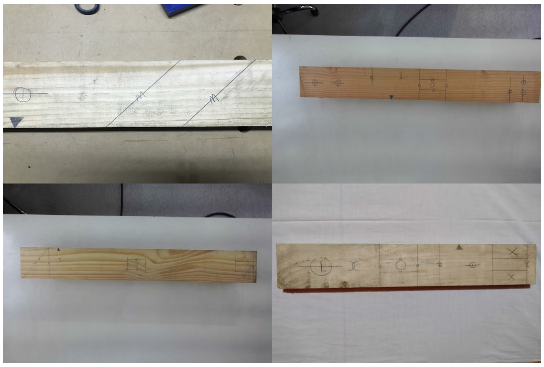

FIGURE 8. Image samples from the generated wood stereotomy dataset.

We set up a collaborative robot, UR5, which had a ZED2 stereo camera mounted on it, to capture visual images of each workpiece. Starting from a position 40 cm above the workpiece, the robot began to capture image sequences by making small displacements along the long axis of the workpiece. To ensure enough images for the object detector's training process, we obtained ten visual images of 1280 × 720 with depth information for each workpiece. However, due to practical limitations, only 70% of the samples were captured using this procedure, while the remaining 30% was

TABLE 1. Dataset specifics: Each group in the dataset corresponds to a different individual who drew a timber layout on the workpieces. The abbreviations FQ, EV, CQ, JN, LG, and LC correspond to initials where the first letter represents the first name and the second letter represents the last name of each expert.

Group	End-cut line	Invalid line	Reference face	Arris	Mortise side	Remove	Peg-hole	Reduce	Images	Total symbols
FQ	110	131	71	961	1124	282	658	123	149	3460
EV	97	144	121	124	409	102	130	134	100	1261
CQ	54	38	40	48	162	41	87	52	100	522
JN	25	47	52	52	185	50	21	75	100	507
LG	27	61	50	50	228	58	85	70	100	629
LC	58	71	79	79	336	91	161	130	100	1005
Total	371	492	413	1314	2444	624	1142	584	649	7384

obtained using standard cameras in an attempt to manually emulate the robot procedure.

The details of the symbols dataset are presented in Table 1, where each group's name represents the first two letters of the human collaborator who drew the layout. Figure 8 shows image samples from the dataset.

We ensure that the symbols and their corresponding bounding boxes are provided for each image, accurately outlining the objects in the image. To further aid in object detection and recognition, we provide an image mask that delimits the boundaries of each workpiece in the image. We also give a set of points of interest, as discussed in the previous section, which serves as additional cues to define the robot tasks.

VII. EXPERIMENTS

A. SYSTEM SETUP

We use C++ and Python for the implementation of each one of our experiments. In particular, we use Pytorch [33] to train the different models evaluated in this work. For the evaluation, we use an 8-Core Intel Core i7-1070 desktop computer with 32 GB in RAM and an NVIDIA GeForce RTX 3090.

B. SYMBOL DETECTION

Our experiments evaluated three object detection solutions for identifying hand-drawn symbols on workpieces. Specifically, we trained and assessed seven models with varying numbers of parameters. We aimed to measure their precision and their runtime performance, determining the solution's suitability for real-time applications. The models we trained and evaluated include YOLO5nu [23], YOLO5mu [23], YOLO5m6u [23], YOLO8n [24], YOLO8m [24], YOLO8x [24], and RT-DETR [32]. Among these, YOLO5nu is the smallest and fastest, while YOLO8x is the largest and slowest. We determined the optimal parameters for each model through a grid search, considering batch sizes and learning rates. During training, we utilized multiple image augmentations such as angle, saturation, exposure, and hue value adjustments. After identifying the best parameters, we retrained the network on the entire training dataset and conducted the final evaluation of the testing set.

Almost all models were trained using batch size 16 and image size 640 × 640x3 except for YOLO5m6u, which used images of 1280 × 1280x3 and batch size 8 due to memory requirements. Other parameters were left as default. The learning rate of all models was the same at 0.01.

Table 2 displays the results of our symbol detection experiments. The mean average precision for each model exceeded 85%, indicating robust generalization across diverse collaborators and wood types. Among the symbols, the invalid line exhibited the weakest performance. The invalid line symbol poses a unique challenge as it is drawn over other symbols or lines, resulting in greater appearance variability. This intricacy hinders the detector's ability to generalize, requiring more training samples than other symbols. Additional challenges arose with similar X symbols, such as the reduced operation and the end-cut line. Notably, the X symbols were often mistakenly identified between the reduce and the end-cut-line, indicating a need for further refinement in distinguishing between these two symbols.

In Figure 9, it is evident that most models can operate in real-time (exceeding 30 FPS) on desktop devices and GPU-embedded devices like the Jetson AGX Orin. The latter is particularly well-suited for robotics applications, owing to its compact size and reduced energy consumption. The importance of using embedded devices in robotics stems from their ability to offer real-time processing, reliability, customizability, seamless integration, and cost-effectiveness. These factors contribute to developing more efficient, responsive, and affordable robotic systems that perform tasks in various environments and conditions.

C. PEG HOLE DETECTION

We evaluated six models to determine the drilling locations of peg holes. On the one hand, we assessed faster and smaller models such as ShuffleNet [26] and ResNet-18 [34]. On the other hand, we looked into larger and relatively slower models like ConvNeXt Tiny [35] and ConvNeXt Large [35]. We utilized pretrained models from Torchvision, modifying only the final layers to perform regression instead of classification. The Optuna [36] hyperparameter optimization framework was employed to identify the best parameters for each model. The hyperparameter optimization was done using a 4-fold cross-validation strategy and conducting 100 trials per model. The parameter search covered a range of batch sizes, learning rates, optimizers, learning decay rates, and scheduler

TABLE 2. Dataset symbols evaluation using different object detectors. Higher is better. The results are an average of 5 runs.

Model	AVG	Mortise side	Peg hole	Invalid line	Reduce	Arris	Reference face	Remove	End-cut line
YOLO5nu [23]	0.8767	0.8234	0.9936	0.5226	0.9652	0.9768	0.9818	0.9026	0.8656
YOLO5mu [23]	0.9194	0.8930	0.9808	0.6988	0.9386	0.9868	0.9934	0.9282	0.9354
YOLO5m6u [23]	0.9110	0.8940	0.9718	0.7426	0.9150	0.9536	0.9706	0.8950	0.9458
YOLO8n [24]	0.8846	0.8626	0.9872	0.5496	0.9540	0.9616	0.9772	0.9226	0.861
YOLO8m [24]	0.9178	0.9150	0.9770	0.6648	0.9374	0.9880	0.9938	0.9322	0.9366
YOLO8x [24]	0.9428	0.9340	0.9836	0.7996	0.9412	0.9892	0.9948	0.9612	0.9414
RT-DETR-l [32]	0.8548	0.8498	0.9604	0.5154	0.9064	0.9878	0.9900	0.8460	0.7840

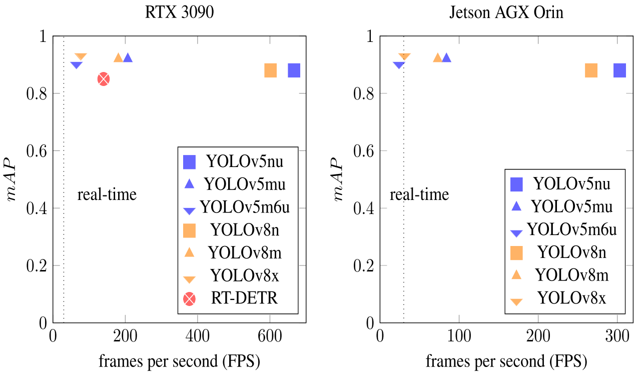

FIGURE 9. The runtime performance of the symbol object detector is evaluated on both a desktop RTX 3090 GPU and a Jetson AGX Orin embedded device. For all experiments, we utilize TensorRT. In the case of the Jetson AGX Orin, evaluating RT-DETR runtime was impossible due to software limitations.

strategies. Table 3 shows the final hyperparameters chosen for each model.

TABLE 3. Hyperparameters for regression models, optimized using Optuna. Abbreviations: lr (learning rate), bs (batch size), g (gamma or factor), gs (gamma step), opt (optimization method), and sch (optimization scheduler). Training duration ranged from 100 to 200 epochs, with early stopping implemented after 30 epochs without improvement. All models were trained in 16-bit mixed precision.

Model	lr	bs	g	gs	opt	sch
Shufflenet	0.001	32	0.35	-	adamw	plateau
Resnet-18	0.001	32	0.45	45	adam	steplr
Efficientnet	0.005	64	0.2	-	adamw	plateau
ConvNeXt Tiny	0.0005	32	0.15	-	adamw	plateau
ConvNeXt Large	0.0005	32	0.35	40	adam	steplr
ResNeXt	0.001	32	0.35	-	adam	plateau

To ensure robustness, we augmented the input data with rotations, vertical and horizontal flips, and color jitter. We chose the Smooth L1 Loss as our loss function. The input image size for each peg hole was resized to 128 × 128 × 3.

The outcomes of our experiments are presented in Table 5. As depicted in the table, the average L2 error is under 4 pixels, demonstrating the viability of our proposed method. The JN subset posed a greater challenge for the network due to the texture of the material, as JN was the sole collaborator using non-brushed wood.

All regression models demonstrated efficiency in runtime, completing tasks in under 12 ms on desktop and embedded GPUs. Notably, ShuffleNet and ResNeXt exhibited a favorable balance among performance, training duration, and runtime. Generally, the regression phase barely added to the total processing time, ensuring the real-time performance of our computer vision solution. Figure 10 provides a glimpse of the regression outcomes.

TABLE 4. This table displays the quantity of peg holes allocated for training and testing purposes. Specifically, the JN and LC subsets were employed for testing and the others subsets for training, consistent with previous experiments.

Group	CQ	EV	FQ	JN	LC	LG
Holes	86	129	655	21	161	85

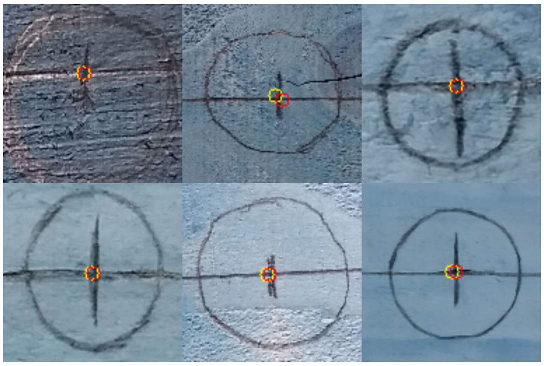

FIGURE 10. Regression outcomes from utilizing ShuffleNet on the testing subset, where red represents ground truth and yellow indicates predicted values.

D. UTILIZING COLOR SEGMENTATION FOR TASK GEOMETRY IDENTIFICATION

Geometry detection has been addressed using traditional computer vision techniques, such as contour and line detection [4]. However, for these techniques to function effectively, they require an accurate color segmentation technique that discriminates between pixels from strokes and those from the background, precisely, the wood texture.

Conventionally, color segmentation is accomplished using a range of color values that need to be pre-selected and calibrated. This process must often be repeated multiple times, as the color range is influenced by varying illumination conditions that change throughout the day.

In this article, particularly in Section V, we present our innovative solution to this challenge. By leveraging *k*-means, an unsupervised clustering technique, we are not

TABLE 5. Regression results for the evaluated models. Runtimes measured using TensorRT.

Model	Parameters	L2 mean	RTX 3090 (ms)	Jetson Orin (ms)
Shufflenet [26]	1.26 M	2.56 ± 0.36	0.34	0.83
ResNet-18 [34]	11.18 M	3.46 ± 0.96	0.63	0.60
EfficientNet [37]	4.01 M	3.49 ± 0.91	0.59	1.31
ResNeXt [38]	22.98 M	2.52 ± 0.39	0.61	1.74
ConvNeXt Tiny [35]	27.82 M	2.74 ± 0.34	1.04	3.41
ConvNeXt Large [35]	196.23 M	2.66 ± 0.23	5.16	11.26

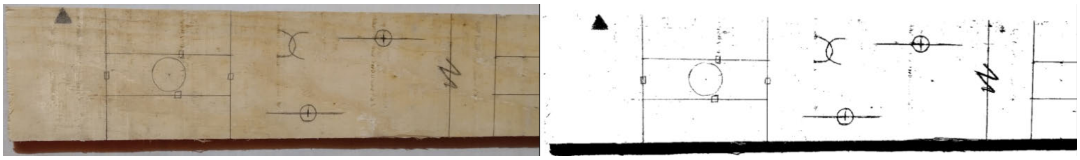

FIGURE 11. Effective use of the *k*-means algorithm for color-based segmentation of hand-drawn carpentry drawings. The original image is on the left, and the segmented one is on the right.

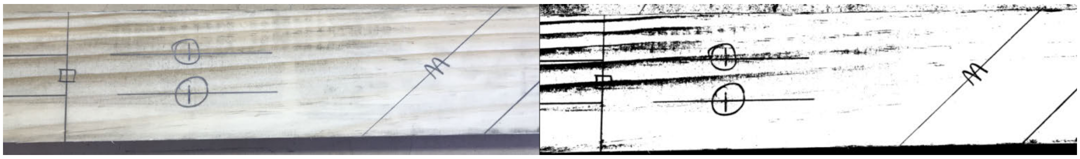

FIGURE 12. Ineffective use of the *k*-means algorithm for color-based segmentation of hand-drawn carpentry drawings. The pencil color is too similar to wood imperfections. The original image is on the left, and the segmented one is on the right.

only addressing the issue but doing so with a significantly reduced need for manual interventions.

Given the absence of a segmentation ground truth, we believe a qualitative evaluation provides a comprehensive and meaningful assessment of the segmentation results. By sharing instances where our technique excels and instances where it falls short, we offer a balanced view of its performance. Furthermore, we delve into the causes of any failures to provide a holistic understanding. This approach ensures that our evaluation is not just about success or failure but a deep exploration into understanding the strengths and limitations of our method, paving the way for further improvement.

Figure 11 effectively illustrates the successful application of our technique in segmenting strokes from the background. Despite the strokes being drawn with a conventional carpenter's pencil, the contrast with the background is substantial enough to allow the technique to accurately distinguish between pixels corresponding to strokes and those belonging to the background. Moreover, the use of custom pencils, which provide even greater contrast with the background, could further enhance the segmentation, particularly in scenarios where the wood's texture closely resembles that of the strokes.

Conversely, Figure 12 illustrates a situation where our proposed technique may not be as effective due to insufficient contrast. This can be partially attributed to the wood's texture, which features several natural lines similar in color to the carpenter's pencil. Additionally, imperfections in the wood tend to produce comparable effects on the images. In these cases, opting for a different pencil color could readily address the issue. Importantly, the system does not need to be adjusted to this new color; it can autonomously differentiate between strokes and the background when there is a high contrast between the pencil and the texture.

VIII. CONCLUSION

In conclusion, our study offers promising insights for enhancing human-robot collaboration in carpentry and joinery. We introduced a hand-drawn language designed for human-robot collaboration in wood stereotomy tasks. Furthermore, we presented a method for its automatic interpretation. Our results indicate that the proposed language and computer vision solutions are feasible on embedded devices, achieving mAP values that exceed 90% while maintaining real-time speed. Concurrently, regression models can precisely pinpoint specific locations under 3 pixels of error, taking less than 1 ms on embedded systems. However, a limitation of our study is its focus on only six types of wood. While our findings generalize well, different wood types might necessitate further training. More than just additional data, our results emphasize the importance of ensuring sufficient contrast between drawings and the wood surface.

REFERENCES

[1] W. Beemer, *Learn to Timber Frame: Craftsmanship, Simplicity, Timeless Beauty*. North Adams, MA, USA: Storey Publishing, 2016.

[2] R. Newman, *Oak Framed Buildings*. Lewes, U.K.: Guild of Master Craftsman, 2005.

[3] J. Pedersen and D. Reinhardt, "Robotic drawing communication protocol: A framework for building a semantic drawn language for robotic fabrication," *Construct. Robot.*, vol. 6, pp. 239–249, Jan. 2023.

[4] J. Pedersen, A. Søndergaard, and D. Reinhardt, ‘‘Hand-drawn digital fabrication: Calibrating a visual communication method for robotic on-site fabrication,’’ *Construct. Robot.*, vol. 5, pp. 159–173, Apr. 2021.

[5] D. Strazdas, J. Hintz, A.-M. Felßberg, and A. Al-Hamadi, ‘‘Robots and wizards: An investigation into natural human–robot interaction,’’ *IEEE Access*, vol. 8, pp. 207635–207642, 2020.

[6] G. Sziebig, ‘‘Man-machine and intermachine interaction in flexible manufacturing systems,’’ Ph.D. thesis, Fakultet for Ingeniørvitenskap og Teknologi, Institutt for Produksjons-og Kvalitetsteknikk, Norges Teknisk-Naturvitenskapelige Universitet, Trondheim, Norway, 2013.

[7] B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. Alcazar, ‘‘Gestures for industry intuitive human–robot communication from human observation,’’ in *Proc. 8th ACM/IEEE Int. Conf. Human–Robot Interact. (HRI)*, Mar. 2013, pp. 349–356.

[8] C. Deuerlein, M. Langer, J. Seßner, P. Heß, and J. Franke, ‘‘Human–robot-interaction using cloud-based speech recognition systems,’’ *Proc. CIRP*, vol. 97, pp. 130–135, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827120314359

[9] A. Kramberger, A. Kunic, I. Iturrate, C. Sloth, R. Naboni, and C. Schlette, ‘‘Robotic assembly of timber structures in a human–robot collaboration setup,’’ *Frontiers Robot. AI*, vol. 8, Jan. 2022, Art. no. 768038.

[10] R. Zhang, Q. Lv, J. Li, J. Bao, T. Liu, and S. Liu, ‘‘A reinforcement learning method for human–robot collaboration in assembly tasks,’’ *Robot. Comput.-Integr. Manuf.*, vol. 73, Feb. 2022, Art. no. 102227. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584521001095

[11] A. Kunic, R. Naboni, A. Kramberger, and C. Schlette, ‘‘Design and assembly automation of the robotic reversible timber beam,’’ *Autom. Construct.*, vol. 123, Mar. 2021, Art. no. 103531.

[12] X. V. Wang and L. Wang, ‘‘Augmented reality enabled human–robot collaboration,’’ in *Advanced Human–Robot Collaboration in Manufacturing*. Berlin, Germany: Springer, 2021, pp. 395–411.

[13] O. Kyjanek, B. Al Bahar, L. Vasey, B. Wannemacher, and A. Menges, ‘‘Implementation of an augmented reality AR workflow for human robot collaboration in timber prefabrication,’’ in *Proc. Int. Symp. Autom. Robot. Construction (IAARC)*, May 2019, pp. 1223–1230.

[14] P. Devadass, S. Stumm, and S. Brell-Cokcan, ‘‘Adaptive haptically informed assembly with mobile robots in unstructured environments,’’ in *Proc. Int. Symp. Autom. Robot. Construction (IAARC)*, May 2019, pp. 469–476.

[15] S. Stumm, J. Braumann, M. Hilchen, and S. Brell-Cokcan, ‘‘On-site robotic construction assistance for assembly using a-priori knowledge and human–robot collaboration,’’ in *Proc. Int. Conf. Robot. Alpe-Adria Danube Region*, vol. 540, 2017, pp. 583–592.

[16] B. Solvang, G. Sziebig, and P. Korondi, ‘‘Robot programming in machining operations,’’ in *Robot Manipulators*, M. Ceccarelli, Ed. Rijeka, Croatia: IntechOpen, 2008, ch. 26, doi: 10.5772/6221.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ‘‘You only look once: Unified, real-time object detection,’’ 2015, *arXiv:1506.02640*.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ‘‘SSD: Single shot MultiBox detector,’’ 2015, *arXiv:1512.02325*.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ‘‘Mask R-CNN,’’ 2017, *arXiv:1703.06870*.

[20] Z. Cai and N. Vasconcelos, ‘‘Cascade R-CNN: Delving into high quality object detection,’’ 2017, *arXiv:1712.00726*.

[21] P. Soviany and R. Tudor Ionescu, ‘‘Optimizing the trade-off between single-stage and two-stage object detectors using image difficulty prediction,’’ 2018, *arXiv:1803.08707*.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, ‘‘YOLOv4: Optimal speed and accuracy of object detection,’’ 2020, *arXiv:2004.10934*.

[23] G. Jocher et al., ‘‘ultralytics/yolov5: v3.1—Bug fixes and performance improvements,’’ Zenodo, v3.1, Oct. 2020, doi: 10.5281/zenodo.4154370.

[24] G. Jocher, A. Chaurasia, and J. Qiu. (2023). *Ultralytics YOLOv8*. [Online]. Available: https://github.com/ultralytics/ultralytics

[25] J. Zhang, H. Hu, and S. Feng, ‘‘Robust facial landmark detection via heatmap-offset regression,’’ *IEEE Trans. Image Process.*, vol. 29, pp. 5050–5064, 2020.

[26] X. Zhang, X. Zhou, M. Lin, and J. Sun, ‘‘ShuffleNet: An extremely efficient convolutional neural network for mobile devices,’’ 2017, *arXiv:1707.01083*.

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ‘‘MobileNets: Efficient convolutional neural networks for mobile vision applications,’’ 2017, *arXiv:1704.04861*.

[28] G. Moon, J. Y. Chang, and K. M. Lee, ‘‘PoseFix: Model-agnostic general human pose refinement network,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7765–7773.

[29] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, ‘‘A review on deep learning techniques applied to semantic segmentation,’’ 2017, *arXiv:1704.06857*.

[30] W. Beemer, ‘‘Timber framing for beginners: III. Introduction to layout,’’ *Timber Framing*, vol. 63, pp. 12–17, Mar. 2002.

[31] A. Z. Chitade and S. Katiyar, ‘‘Colour based image segmentation using K-means clustering,’’ *Int. J. Eng. Sci. Technol.*, vol. 2, no. 10, pp. 5319–5325, 2010.

[32] W. Lv, S. Xu, Y. Zhao, G. Wang, J. Wei, C. Cui, Y. Du, Q. Dang, and Y. Liu, ‘‘DETRs beat YOLOs on real-time object detection,’’ 2023, *arXiv:2304.08069*.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, ‘‘PyTorch: An imperative style, high-performance deep learning library,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32. Vancouver, BC, Canada: Curran Associates, 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[34] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[35] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, ‘‘A ConvNet for the 2020s,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11976–11986.

[36] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, ‘‘Optuna: A next-generation hyperparameter optimization framework,’’ in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2623–2631.

[37] M. Tan and Q. Le, ‘‘EfficientNet: Rethinking model scaling for convolutional neural networks,’’ in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[38] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, ‘‘Aggregated residual transformations for deep neural networks,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1492–1500.

CRISTHIAN A. AGUILERA-CARRASCO (Member, IEEE) received the Ph.D. degree in computer science from the Autonomous University of Barcelona, in 2017. Throughout his academic journey, he has dedicated his research efforts to the fields of cross-spectral imaging, machine learning, and robotics. He has utilized GPU computing to expedite his research tasks. His current research interests include exploring and implementing computer vision in real-time robotics and industrial scenarios while harnessing the potential of GPUs.

LUIS FELIPE GONZÁLEZ-BÖHME received the B.Arch. degree from the Pontifical Catholic University of Chile (PUC), Santiago, Chile, in 1999. From 2001 to 2006, he was a half-time Research Scientist and a Teaching Assistant with the Chair of Computer Science in Architecture, Bauhaus-University Weimar (BUW), Germany. Since 2007, he has been a full-time Professor with the Department of Architecture, Technical University Federico Santa María (UTFSM), Valparaíso, Chile. He is currently the Head of the Department of Architecture. His current research interests include timber joinery robotics, constraint-based design, and affordable housing.

FRANCISCO VALDES received the Ph.D. degree in design computing and fabrication from the Georgia Institute of Technology (GaTech). He is currently a Senior Research Engineer with the Cybersecurity, Information Protection, and Hardware Evaluation Research Laboratory (CIPHER), Georgia Tech Research Institute, and a Teaching Fellow with GaTech. He is also a Fulbright Scholar and a former Professor with Technical University Federico Santa María (UTFSM) and a Researcher with the System Architectures Group, NASA's Jet Propulsion Laboratory. His current research interests include applied engineering, manufacturing and product lifecycle management (PLM) in renewable energy for Earth and Martian applications, model-based system engineering (MBSE), and building information modeling (BIM).

FRANCISCO JAVIER QUITRAL-ZAPATA received the B.Arch. degree from Technical University Federico Santa María, Valparaíso, Chile, in 2015. He is currently pursuing the Ph.D. degree with the University of Bío-Bío, Concepción, Chile. Since 2015, he has been a partial Professor with the Department of Architecture, Technical University Federico Santa María. His current research interests include robotic timber construction, computational design, digital fabrication, and robotic construction.

BOGDAN RADUCANU received the bachelor's degree in computer science from the Polytechnic University of Bucharest, Romania, in 1995, and the Ph.D. degree (cum laude) in computer science from the University of the Basque Country, in 2001. From 2002 to 2004, he was a Postdoctoral Researcher with the Technical University of Eindhoven, The Netherlands. In 2004, he joined the Computer Vision Center (CVC) with a "Ramon y Cajal" Fellowship and has been a Senior Researcher and the Project Director, since 2009. His current research interests include computer vision and deep learning with applications to lifelong learning, generative models, and robotics.