

YOLO-based Detection of Drilled Blind Holes in Laminated Panels with Template Similarity Assessment

Lukas Zabulis, Rytis Augustauskas, Dominykas Lelešius, Arūnas Lipnickas
Department of Automation, Kaunas University of Technology,

Studentu 48-109, LT-51367 Kaunas, Lithuania, www.ktu.edu

lukas.zabulis@ktu.lt, rytisaugustauskas@yahoo.com, domlel@ktu.lt, arunas.lipnickas@ktu.lt

Abstract—This paper proposes an end-to-end approach for detecting drilled holes in high-resolution images of furniture panels and verifying their placement accuracy using a computationally efficient algorithm. The method utilizes YOLO object detection and a sliding-window inspection to analyze line-scanned images for any blind holes and corners in panels. The center point coordinates of each detection bounding box are used to create a vector mesh that connects all detected holes and corners. Euclidean distances of these vectors are then calculated, sorted out, and used as a board holes signature. The resulting signature is compared with template measurements to validate their consistency with expected values. The proposed approach achieves high accuracy in detecting drilled holes and corners in furniture panels, with a mean average precision (mAP50) of 99.4%. Additionally, the template similarity assessment algorithm, along with YOLO detection, can spot deviations of ± 10 px in hole positions compared to the template using our custom dataset. Higher-resolution images can further decrease the minimum deviation detection. The method offers the potential to streamline the furniture drilled hole inspection process and reduce defective production.

Keywords—YOLO; deep learning; object detection; sliding-window; template matching.

I. INTRODUCTION

The manufacturing of laminated MDF furniture panels is a multi-step process that involves chipboard production, lamination, cutting, milling, and drilling. The accuracy of the drilling process is critical as it determines the success of the final product assembly. However, deviations from the drilling template can occur due to failures or misalignments in the drilling machinery, wrong template set-up, and dull or broken drills. Therefore, it is crucial to have an efficient and accurate system for detecting drilled holes and validating their positions in furniture panels during the online production process. Manual inspection is a time-consuming process and demands knowledge about each part's template. The complexity of the problem increases even more due to different sizes of drillings, multiple holes, and a wide diversity of board surface textures. For this

reason, employing a computer vision system is an essential part of the fast-paced and error-prone manufacturing process.

In this paper, we present a data-driven approach for detecting drilled holes along with corners in furniture panels from high-resolution images. Our method employs the latest YOLOv8 architecture for object detection in patched regions of the image. We address the issue of overlapping detections by using a bounding box combining technique during post-processing. The algorithm of drilled holes template similarity assessment generates a vector mesh based on the center point coordinates of detections and calculates Euclidean distances of created vectors. These distances are sorted out and compared with the existing distances of the templates in a database. The presented method is resistant to varying panel orientation, different contexts, or the size of drillings and is capable to detect even minor deviations from the template.

This paper is structured as follows: Section 2 provides a comprehensive review of computer vision inspection applications for circular-shaped and other kinds of object detection problems. Section 3 explains the dataset used in this study, along with the techniques employed for preprocessing the data. Section 4 covers the architecture of the newest YOLOv8 object detection neural network, its training process, and the post-processing approach to obtain center point coordinates of holes along with corners. This section also involves the theory behind the template similarity assessment algorithm. The results obtained from the experiments are presented in Section 5, followed by conclusions drawn from the study and future directions for research in Section 6.

II. RELATED WORK

Numerous papers have put forth proposals for computer vision applications aimed at circular object detection. If the objects in the image are clearly distinguished or have a static context, then simple image processing techniques help to extract relevant information. The researchers make use of iterative statistical methods to

find the best-fitting circles with the Hough transform (HT). Study [1] proposes Angle-Aided Circle Detection (AACD) algorithm based on the randomized Hough transform to detect multiple circles in the image. Authors in [2] utilize an adaptive edge detector along with a rapid circle center search algorithm to detect holes in a PCB. However, in more sophisticated contexts, data-driven models are required for proper object (hole) detection. For example, researchers in their work [3], employ lightweight convolutional autoencoder architectures to perform drilling segmentation in textured wooden furniture boards to estimate drilling position from grayscale images and compare it with a template of furniture parts. In [4] researchers solve a similar problem of hole segmentation using Mask R-CNN and synthetic image dataset. Study [5] presents a more sophisticated solution for the inspection of drilled holes in aerospace Carbon Fiber Reinforced Polymers. The four-direction lighting is used to suppress the background of the image and the U-Net is trained to perform pixel-wise segmentation of different areas. Another machine learning approach is proposed in [6] where the authors design a CNN network that makes full use of multi-level features to extract the edge of the hole and propose a new circle-fitting method to obtain the hole. Researchers in [7] present a semi-supervised deep learning-based solution and demonstrate competitive detection accuracy of measuring drilled holes on composite parts with very limited and noisy training data. There also can be found papers where various object detection tasks are performed using YOLO-based [8] neural networks. Authors in their paper [9] present a fast and efficient YOLO-S architecture for small target detection in aerial imagery making use of a lightweight feature extractor. Another use case of YOLO is proposed in [10] where researchers use a modified YOLOv3 model for metal surface defect detection keeping attention to small defects with a help of a new scale feature layer.

However, very few papers address the issue of high-resolution image processing and training state-of-the-art object detection models with a limited amount of data. There is also a research gap in simple and computationally effective template similarity assessment algorithms. In this paper, we cover these topics and provide an end-to-end approach for hole detection and template validation process that could be applied to similar problems in the industry.

III. DATASET

A. Description

The dataset used for this study comprises a total of 192 images. Out of these, 168 images (87.5%) were used for training and validation, while the remaining 24 images

(12.5%) were reserved for end-to-end testing of the hole detection and template similarity assessment algorithm. Each image in the dataset contains a furniture panel, which comes in various dimensions, ranging in width from 1000 to 6144 pixels and height from 900 to 12,384 pixels. The furniture panel's exterior texture, colors, and hole sizes exhibit significant variations, making it a challenging task to detect holes using traditional image processing approaches. Figure 1 illustrates a few randomly cropped samples from the original dataset. The same dataset has been used by researchers in their work [3] for hole segmentation task.



Figure 1. Background texture and hole variations among different samples

B. Preprocessing

Training a YOLO object detection neural network requires a large amount of data, which is why preprocessing the existing database is necessary to extract a sufficient number of samples. However, it is also crucial to preserve the resolution of the images since the objects (i.e., holes) are relatively small compared to the context. Resizing the images to suit the input size of YOLO would result in a loss of spatial information, making it challenging to train the model effectively.

To extract samples from the images, we employed a sliding-window approach that covers the entire image area (as shown in Figure 2). In this study, we have utilized the modern YOLOv8 model that can handle arbitrary-sized images as long as both dimensions are a multiple of 32. This requirement is due to the backbone's maximum stride of 32, as the model operates as a fully convolutional network. To address the dataset's specific characteristics, particularly the lowest image dimension of 900 pixels, we have tailored our approach accordingly. We have set the window size to 544x544 pixels, taking into account an overlap of 272 pixels. This configuration enables us to effectively analyze the images and capture important features. To ensure a balanced dataset, the overlap is increased one and a half times in regions containing holes and corners. This dataset preprocessing technique allowed us to obtain a greater number of samples while preserving the resolution of the original images. Additionally, each image has a separate file with bounding box label information which is required for training the model. The bounding box of a hole region is the minimum area rectangle while the bounding box of a corner is a fixed-size (50x50 pixels) rectangle.

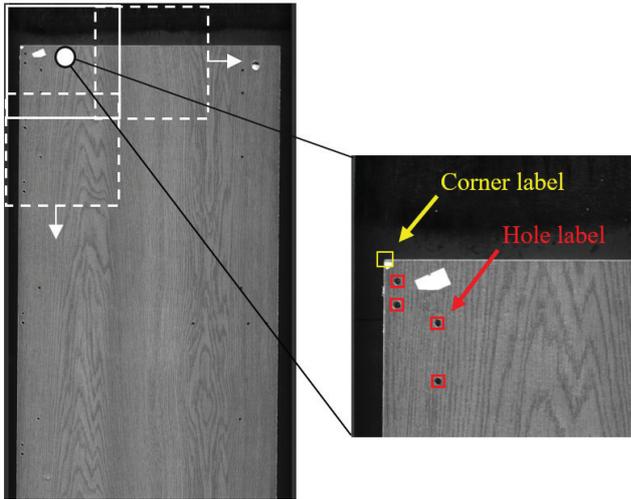


Figure 2. Image sample extraction methodology

A description of preprocessed sample database is provided in TABLE I.

TABLE I. PARAMETERS OF SAMPLE DATABASE

Parameter	Description
Number of high-resolution images	168
Sample size, px	544x544
Number of extracted samples	19318
Train-Validation split	0.8/0.2

C. Augmentations

It is important to ensure that the dataset is diverse and representative enough to train a high-performing model while at the same time reducing the risk of overfitting. For this purpose, a composed set of augmentation techniques with specified occurring probabilities is applied (TABLE II). Empirical evaluations and experiments revealed that higher HSV-Saturation and mosaic augmentation probabilities enhance model robustness.

TABLE II. AUGMENTATION TECHNIQUES

Augmentation	Probability
Scale	0.5
Flip left-right	0.5
Flip up-down	0.5
HSV-Saturation	0.7
Mosaic	0.8

In object detection tasks, mosaic augmentation [11] is one of the most effective augmentation techniques. It involves randomly selecting four images from the dataset and cropping them into four patches. These patches are then arranged in a new image to create a mosaic, with each patch occupying a specific quadrant. The resulting image

is utilized as a training sample (see Figure 3).

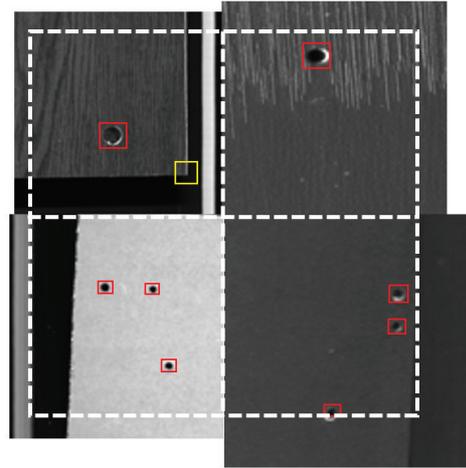


Figure 3. Mosaic augmented training sample

IV. METHODOLOGY

A. YOLOv8 architecture

Since the computer vision system is employed in a production environment and high-resolution images are analyzed the requirement of low-latency inference is quite important. This can be achieved with the latest architectures of single-shot object detection networks. For this research, we decided to employ YOLOv8n (nano) [12] architecture as a baseline model. This model achieves 37.3% mAP50-95 accuracy on a COCO val2017 dataset with as little as 3.2M parameters and 8.7B floating-point operations (FLOPs) per forward pass.

The YOLOv8 is a single-stage object detection neural network that represents the latest and most advanced version of the YOLO series. It has surpassed the previous versions in terms of accuracy and speed. YOLOv8 architecture consists of two main components: the backbone and the head (including the neck). The detection process begins by using the backbone to extract important features from an image. Then, these features are transmitted to the neck and the head. The neck builds feature pyramids by collecting feature maps extracted by the backbone and combining them. The head then predicts the bounding box and class based on these pyramids. The algorithm is guided by three loss functions for class, box, and object, as its guidelines.

Compared to the previous YOLOv5 model, this architecture replaced the C3 module with the C2f module in which all the outputs from the bottleneck (3x3 convolutions with residual connections) are concatenated. While in the C3 module, only the output of the last Bottleneck was used. The backbone convolution window size is 2 times smaller (3x3), it gets rid of the convolutional layer in the backbone. The bottleneck's first convolution layer size changed to 3x3 instead of 1x1. The architecture

also abandons decoupled head and objectness branches. YOLOv8 is an anchor-free model. This means it predicts directly the center of an object instead of the offset from a known anchor box. Anchor-free detection reduces the number of box predictions, which speeds up Non-Maximum Suppression (NMS), a complicated post-processing step that sifts through candidate detections after inference.

B. Training

The model was trained for 300 epochs through transfer learning with a batch size of 20. SGD (Stochastic Gradient Descent) was used as an optimizer with a learning rate of 0.01 and a momentum of 0.937. For the comparison, YOLOv5 was also trained with the same hyperparameters. The training results are given in TABLE III. The brief explanations of utilized metrics:

- Precision: Precision is the fraction of true positive detections (i.e., correctly detected objects) out of all detections made by the model. A high precision indicates that the model is good at avoiding false positive detections.
- Recall: Recall is the fraction of true positive detections out of all ground-truth objects in the dataset. A high recall indicates that the model is good at detecting all objects in the dataset, including those that may be difficult to detect.
- mAP₅₀: Mean Average Precision (mAP) is a common metric for evaluating object detection performance. It is calculated as the average precision across multiple object categories and IoU (Intersection over Union) thresholds, and mAP₅₀ specifically refers to the mAP score calculated at an IoU threshold of 0.5.
- mAP₅₀₋₉₅: This is similar to mAP₅₀, but calculated over a wider range of IoU thresholds, typically ranging from 0.5 to 0.95 with a step size of 0.05. This metric provides a more comprehensive measure of object detection performance, as it takes into account varying levels of object overlap between the ground-truth and predicted bounding boxes.

TABLE III. TRAINING RESULTS COMPARISON

Model	P (%)	R (%)	mAP ₅₀	mAP ₅₀₋₉₅
YOLOv5n	98.5	97.9	99.3	82.8
YOLOv8n	98.6	98.1	99.3	85.5

The evaluation metrics presented in the table suggest that YOLOv8 outperforms YOLOv5 in several metrics. Specifically, YOLOv5 achieves a higher (85.5%) mAP₅₀₋₉₀ score, indicating better overall object detection accuracy.

C. Post-processing

In the process of overlapped sliding-window inspection, it's common to encounter multiple bounding boxes that either intersect or touch each other. In such cases, merging bounding boxes that represent the same object is essential to prevent duplicate detections. A useful approach to accomplish this is to draw filled rectangles on an empty image that encompasses the overlapping bounding boxes. Edge detection can then be used to find contours for these filled rectangles. The contours can be utilized to fit a minimum area rectangle, which serves as the final bounding box of the object (see Figure 4).

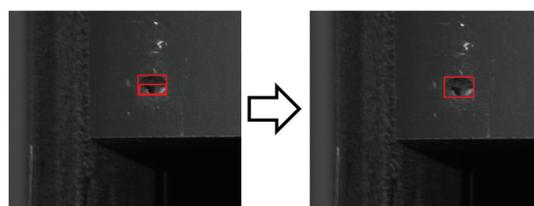


Figure 4. Combining multiple detections of the same object

D. Template similarity assessment

To verify the accuracy of the drilled hole positions in the panel, we propose a template similarity assessment algorithm capable of identifying hole position deviations from the template. Specifically, the algorithm generates a vector mesh from the center point coordinates of holes and corners. The distances of each vector are then calculated using the Euclidean distance formula. These measurements are sorted and subtracted from corresponding measurements of the template which is selected depending on the matching hole count and the nearest vector L2 sum (refer to Figure 5). If the resulting distances are consistent with the expected values in the template and do not exceed the predefined tolerance threshold (in pixels), the holes are considered to be accurately drilled. Otherwise, if one or more holes are misaligned and exceed the L2 difference threshold it may be necessary to remove the defective board or suspend the production line and check the drilling equipment.

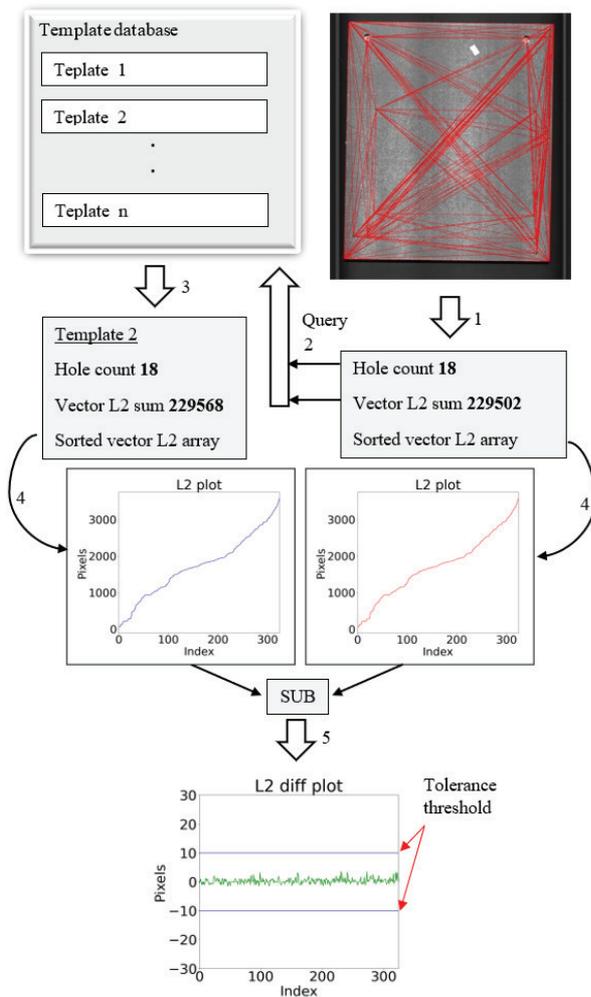


Figure 5. Template similarity assessment algorithm diagram

V. RESULTS

A. Object detection evaluation metrics

For evaluation purposes, test set images were pre-processed using the previously described technique. In total, 24 images resulted in 4096 patches with 1918 object instances (1680 holes and 238 corners). The evaluation metrics of the test set are given in Table IV, where the metrics are calculated for all classes and each specific class separately. The model reached a 99.5% mAP_{50} score for both classes demonstrating its effectiveness in accurately localizing objects. Due to the relatively low number of instances in the corner class during training, the resulting YOLO model exhibited a lower mAP_{50-95} (83.5%) score for corner detection compared to the hole detection class (85.2%).

TABLE IV. TEST SET EVALUATION METRICS

Class	Images	Inst. No.	P (%)	R (%)	mAP_{50}	mAP_{50-95}
all	4096	1918	97.6	99	99.4	84.4
hole	4096	1680	99.6	98.4	99.5	85.2
corner	4096	238	95.6	99.5	99.4	83.5

The inference time of a single 544x544px image is 4.7ms. The pre-processing stage takes 0.4ms while the post-processing takes 0.7ms. The tests were made utilizing a consumer-grade PC (Intel Core i5-2400 CPU @3.10GHz, SSD disk, 10GB RAM, and CUDA 11.7 enabled NVIDIA GeForce GTX 1060 6GB GPU using Python 3.9.16 interpreter and PyTorch v1.13.1 machine learning framework).

B. Template similarity assessment test

To ensure the accuracy of the detection system and establish the tolerance threshold, we utilized a template similarity assessment algorithm. For each of the test images, we generated templates and analyzed the pixel deviations from them. The range of minimum and maximum deviations is presented in Table V, which serves as an indicator of the algorithm's sensitivity. Any L2 differences outside this range are considered deviations from the template.

TABLE V. PIXEL DEVIATIONS FROM THE TEMPLATES

	Min	Max
Deviation, px	-6	4

Based on the determined sensitivity range, it is evident that a reliable threshold of $\pm 10px$ can be set, taking into account center point fluctuations caused by detection errors. To test this, one image was selected from the test set, and four images were artificially generated by rearranging the position of a single hole in four different directions with an offset of 11 pixels, as shown in Figure 6. Figure 7 shows L2 difference plots of the base image and images with hole position rearrangements. Each plot's title represents the abbreviations of various hole offsets (b – base, l – left, r – right, up – upper, lo – lower). The base (“b”) plot fluctuates around a 0-pixel value and does not exceed the threshold value. On the other hand, in every instance of the hole with an offset (“l,” “r,” “lo,” “up”), the threshold (red line indicating $\pm 10px$ offset) was reached or exceeded, confirming the effectiveness of the end-to-end algorithm. The test also demonstrates that hole misalignment detection is independent of the offset direction.

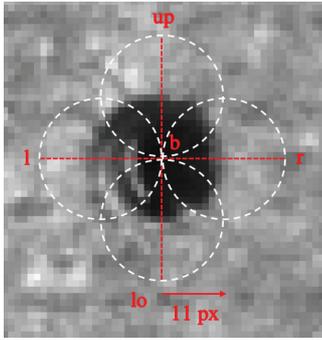


Figure 6. Hole position rearrangement. The abbreviations indicate different offsets (b – base, l – left, r – right, up – upper, lo – lower)

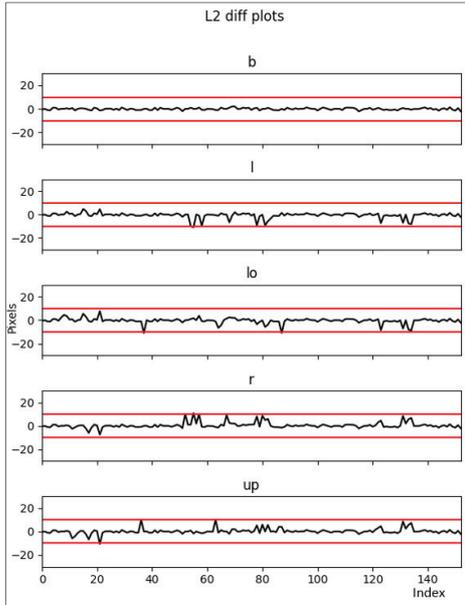


Figure 7. L2 difference plots comparison. Plot titles refer to Figure 6

VI. CONCLUSION

In this paper, we have demonstrated the successful application of the current state-of-the-art object detection model for the detection of holes and corners in laminated panel images. Despite using a limited amount of data for training, our model achieved 99.4% mean average precision (mAP₅₀). We also proposed high-resolution image inspection by a sliding-window approach which is an effective technique for relatively small object detection. The presented end-to-end algorithm does not rely on panel orientation, different contexts, or the size of drillings and is capable to detect ± 10 pixel hole position deviations from the template using our custom dataset. It is also evident that higher-resolution images can further decrease the minimum deviation detection.

For future research, one possible improvement could be made to the template similarity assessment algorithm by incorporating rotations of the vector mesh along with L2 distances. Currently, the algorithm lacks the ability to

identify holes that are out of their correct position, and addressing this limitation would enhance the effectiveness of the inspection. Another potential improvement could involve inspecting the holes themselves to ensure proper drilling, including assessing roundness and determining the extent of any delamination.

ACKNOWLEDGMENT

We would like to express our appreciation to the company MB Prorega (prorega.lt) for providing data. This work has been performed within Kaunas University of Technology master's degree level teaching course T125M157 "Application of Artificial Intelligence in Robotics and their Systems Control" under Assoc. Prof. Dr. A. Lipnickas supervision.

REFERENCES

- [1] Q. Liang *et al.*, "Angle aided circle detection based on randomized Hough transform and its application in welding spots detection," *Mathematical Biosciences and Engineering*, vol. 16, no. 3, pp. 1244–1257, 2019, doi: 10.3934/mbe.2019060.
- [2] W. Zhu, H. Gu, and W. Su, "A fast PCB hole detection method based on geometric features," *Meas Sci Technol*, vol. 31, no. 9, Sep. 2020, doi: 10.1088/1361-6501/ab8b21.
- [3] R. Augustauskas, A. Lipnickas, and T. Surgailis, "Segmentation of drilled holes in texture wooden furniture panels using deep neural network," *Sensors*, vol. 21, no. 11, Jun. 2021, doi: 10.3390/s21113633.
- [4] M. Al-Zubi and P. Plapper, "Segmentation of Drilled Holes in Textured Wood Panels using Deep Learning Framework," in *CIVEMSA 2022 - IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/CIVEMSA53371.2022.9853682.
- [5] A. Maghami, M. Salehi, and M. Khoshdarregi, "Automated vision-based inspection of drilled CFRP composites using multi-light imaging and deep learning," *CIRP J Manuf Sci Technol*, vol. 35, pp. 441–453, Nov. 2021, doi: 10.1016/j.cirpj.2021.07.015.
- [6] X. An, X. Xie, and X. Chen, "Hole Detection with Texture-Suppression on Wooden Plate Surfaces," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 393–404. doi: 10.1007/978-3-030-87355-4_33.
- [7] G. Li, S. Yang, S. Cao, W. Zhu, and Y. Ke, "A semi-supervised deep learning approach for circular hole detection on composite parts," *Visual Computer*, vol. 37, no. 3, pp. 433–445, Mar. 2021, doi: 10.1007/s00371-020-01812-w.
- [8] R. Ac and C. R. K, "Real Time Object Detection System with YOLO and CNN Models: A Review", doi: 10.37896/JXAT14.07/315415.
- [9] A. Betti and M. Tucci, "YOLO-S: A Lightweight and Accurate YOLO-like Network for Small Target Selection in Aerial Imagery," *Sensors (Basel)*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23041865.
- [10] Y. Xu, K. Zhang, and L. Wang, "Metal surface defect detection using modified yolo," *Algorithms*, vol. 14, no. 9, Sep. 2021, doi: 10.3390/A14090257.
- [11] W. Hao and S. Zhili, "Improved mosaic: Algorithms for more complex images," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Nov. 2020. doi: 10.1088/1742-6596/1684/1/012094.
- [12] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics." Mar. 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>