An Effective Method for the Detection of Wall Brick Defects using Machine Vision

Ngoc-Tien Tran
School of Mechanical and Automotive Engineering, Hanoi University of Industry, Vietnam
tientn@haui.edu.vn (corresponding author)

Ngoc-Duy Le
School of Mechanical and Automotive Engineering, Hanoi University of Industry, Vietnam
lengocduy@haui.edu.vn

Van-Nghia Le
School of Mechanical and Automotive Engineering, Hanoi University of Industry, Vietnam
lenvannghia@haui.edu.vn

Received: 15 April 2024 | Revised: 28 April 2024 | Accepted: 3 May 2024

ABSTRACT

The production lines for wall bricks have achieved a high level of automation. Most brick production lines in developing countries have automated the steps up to placing the bricks in the kiln. However, the manual loading and unloading of bricks after firing still remains. This manual process reduces labor productivity and increases the cost of the final product. To address this issue, this study aims to utilize machine vision algorithms to detect cracks in bricks, thereby facilitating the automation of the brick loading and unloading process. A comprehensive image processing method is developed, which combines square detection and moment algorithms to analyze image properties. This integrated approach enables the accurate detection of cracks and the determination of their respective areas, ensuring precise and reliable results. By detecting defects in the bricks, we can replace faulty ones and employ robots to automatically handle rows of bricks. The study's results demonstrate the proposed method's ability to accurately identify brick defects. These findings are significant as they contribute to the automation of brick loading and unloading, which can be implemented in large-scale brick factories, leading to a safer and more efficient working environment.

Keywords-machine vision; wall brick; brick crack; product defects

I. INTRODUCTION

In today's era of the 4.0 industrial revolution, robots are becoming increasingly popular [1-3] and manipulator robots, in particular, have found wide applicability across various industries [4-6], including the wall brick production lines. Traditionally, taking bricks has been a labor-intensive task requiring manual effort and precision. However, the introduction of robotic arms has revolutionized this process. These robotic arms possess the capability to perform repetitive tasks with high precision, speed, and consistency, significantly reducing the reliance on human labor and enhancing overall productivity. Robot arms employed for stacking bricks are equipped with multiple joints and exhibit a wide range of motion, enabling them to reach different positions and angles. They are also equipped with grippers or end-effectors that can securely hold and manipulate bricks (Figure 1). These grippers are designed to accommodate various sizes and shapes of tiles, ensuring versatility in handling different types of bricks. However, in current production lines, robots are only utilized in the pre-kiln loading and unloading stages. Human labor is still required for the post-firing loading and unloading of bricks. During the brick firing process, cracks, defects, or even partial breakage can occur on the brick surface. Consequently, using robots for loading and unloading, similar to the pre-kiln stage, becomes impractical. It becomes impossible to pick up rows of bricks without identifying and removing defective ones. To automate this process, it is essential to employ brick defect detection techniques that identify and remove flawed bricks, allowing the robot to handle rows of bricks seamlessly. Several studies have been conducted to develop and improve error detection methods for various products. For instance, in [7], the authors developed an accurate image processing algorithm using two CCD cameras, a lighting system, and a personal computer to determine the volume and surface area of an orange. The algorithm effectively segments the background and divides the image into frustums, enabling the calculation of their volume and surface area. The results demonstrate a negligible difference between the calculated and measured volumes, as well as a highly correlated regression formula.

Authors in [8] focused on estimating the surface area and volume of a Williams pear quarter, with specific emphasis on the heat energy required for drying. Two different methods were explored in this research. The first involved multiplying the pears by the average surface area, while the second calculated the pear length using correlation and integral calculus techniques. Both methods aimed to provide accurate estimations of the pear's surface area and volume.

The detection of object contours is also crucial in various applications. This includes not only object recognition in computer vision tasks, like pedestrian detection and face recognition, but also brick boundary recognition. Brick edge detection employs a multi-threshold algorithm to address edge detection challenges such as brick image contrast, pixel selection, and error handling. Qualitative and quantitative experiments have been conducted to improve both grayscale and color images. In [9], the authors compare two Edge Detection Algorithms: the Bacterial Foraging Algorithm (BFA) and the Canny Algorithm. In [10], the authors propose a method for image classification using transfer learning to extract image features, detect edges, and classify images using deep learning algorithms. This approach preserves the fundamental structure of the input image and employs a multilayer scattered vector representation for classification. The study utilizes 10,000 images of various leaves, applying fuzzy logic for edge detection and transfer learning for leaf classification. This approach enhances the accuracy of image identification. In [11], the authors present a novel fuzzy edge detector based on fuzzy divergence and fuzzy entropy minimization for grayscale images. The technique utilizes eddy currents, thermal infrared, and electrospinning images for testing the procedure. The fuzzy content of each image is quantified using new specific indices formulated in terms of fuzzy divergence. The results are evaluated using appropriate assessment metrics and show promising outcomes when compared to well-known I- and II-order edge detectors.

Crack detection in objects is another area of interest for many researchers. Image-based techniques and machine learning have been employed for brick segmentation and crack detection in masonry walls [12]. Deep learning networks were evaluated using a large dataset of hand-labelled images to develop quality and geometric models. In [13], researchers proposed an autonomous method for identifying fractures in masonry constructions by integrating deep CNNs and SVMs. The system utilized visual aids and drones to train and validate the model, achieving an accuracy of 86%. Similarly, in [14], deep learning networks were employed for crack detection in masonry facades using transfer learning with limited annotated data. A pre-trained deep convolutional neural network model was utilized as a feature extractor, resulting in an accuracy and F1-score of up to 100% after end-to-end training. Crack detection is also addressed in [15], where cracks are identified during data capture, and a stitched image is generated for analysis before returning to service. An automated image processing approach is proposed to detect cracks in AGR fuel channel images and trepanned holes. Convolutional networks are utilized in [16-18] for automatic detection and segmentation of bridge surface cracks using computer vision deep learning models. The method incorporates a dataset, an improved You Only Look Once (YOLO) algorithm, and a PSPNet algorithm to identify and segment cracks. In [19], the authors introduced the deep super-resolution crack network (SrcNet) for enhancing automated crack detection in large-scale civil infrastructures. SrcNet improves crack detectability by enhancing pixel resolution through deep learning. It consists of two phases: super resolution for image generation and automated crack detection. Experimental validation on South Korean concrete bridges demonstrates that SrcNet achieves 24% better crack detectability compared to raw digital images. Furthermore, crack detection in bridge deck surfaces is also explored in [20].

In reality, the quality of construction work is a crucial aspect governed by quality standards. The level of completion of wall bricks significantly impacts the overall project quality. Therefore, detecting defects in wall brick products holds great importance. This paper presents algorithms and formulas used for color space conversion, noise filtering, thresholding, contour detection, and area calculation to identify brick defects. Any errors on the bricks after firing will be detected and notified, allowing for their replacement and enabling the automation process with a brick-picking robot.

II. MATERIALS AND METHODS

To obtain a grayscale image in the RGB color space, each pixel must have equal values for the red (R), green (G), and blue (B) color channels. These values are denoted as $R(x,y)$, $G(x,y)$, and $B(x,y)$, where (x,y) represent the pixel coordinates. Figure 1 visualizes the color values associated with a pixel in the RGB color space.

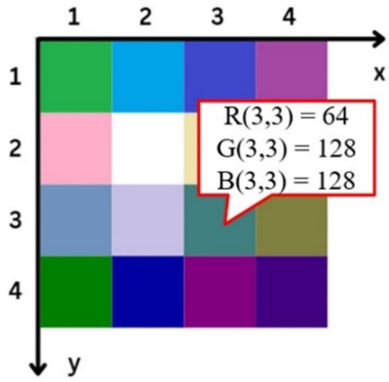


Fig. 1. RGB color system.

Color system conversions are commonly accomplished through matrix transformations. The weighted average conversion formula is:

$$I(x,y) = 0.299 * R(x,y) + 0.587 * \mathrm{G}(x,y) + 0.114 * B(x,y) \quad (1)$$

The weights used in this conversion are obtained by calculating a weighted average of the value of each color channel. Specifically, the weights assigned to the red, green, and blue channels are 0.299, 0.587, and 0.114, respectively. The application of these weights results in the generation of a new image, which is depicted in Figure 2.

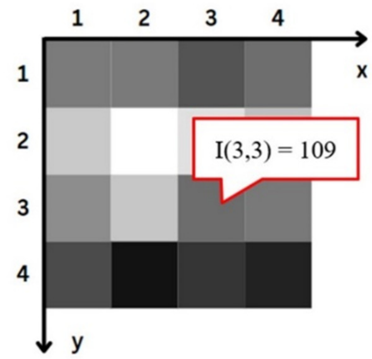


Fig. 2. RGB color system to grayscale.

Once the image has been converted to grayscale, we can enhance its contrast level by applying a Gaussian filter. This is achieved by convolving the input image with a Gaussian kernel matrix. To determine the kernel matrix, we utilize the following formula:

$$H(x,y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

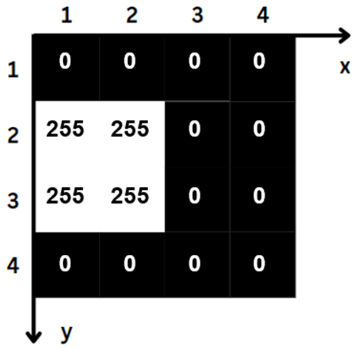
To enhance the contrast level of the grayscale image, we perform a convolutional multiplication of the original matrix I (gray image) with the kernel matrix H.

$$U = I \otimes H \quad (3)$$

The resulting matrix U represents the new image after applying the Gaussian blur filter to reduce noise. To simplify the process of identifying defects in the object, the image is then thresholded using (4). This thresholding operation converts the color range from 0 to 255 into a binary image, where pixel values are represented by either 0 (black) or 255 (white). This binary image is shown in Figure 3.

$$\begin{cases} INP(x,y) = 0 \text{ if } U(x,y) < \theta \\ INP(x,y) = 1 \text{ if } U(x,y) \ > \ \theta \end{cases} \quad (4)$$

In this study, the conversion threshold θ used for thresholding is set to the average value of 128.



	1	2	3	4
1	0	0	0	0
2	255	255	0	0
3	255	255	0	0
4	0	0	0	0

Fig. 3. Binary image after thresholding.

In the process of detecting object borders from binary images, the square detection algorithm is utilized. The image grid is considered to be composed of a group of black pixels on a white background. The algorithm begins at the bottom-left corner of the grid and scans each column of pixels from bottom to top, starting with the leftmost column and moving towards the right. The algorithm continues scanning until it encounters a black pixel. At this point, the algorithm visualizes the current position as a "ladybird" standing on the starting pixel, as depicted in Figure 4. To extract the border of the pattern, the algorithm follows these steps: whenever the ladybird is positioned on a black pixel, the next step is to turn left. Conversely, whenever the ladybird is positioned on a white pixel, the next step is to turn right. This process continues until the ladybird returns to the starting pixel. The black pixels that the ladybird traverses during this process represent the border of the sample image.

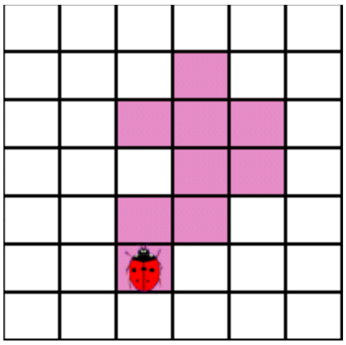


Fig. 4. The algorithm's process of finding pixels.

It is important to note that the concept of left and right turns in the algorithm is based on the direction of the ladybird while scanning the grid. The algorithm for retrieving the pixels is described below.

SQUARE DETECTION ALGORITHM

```
Input: $T, P$
Output: B
B← ∅
s ← $T_1$
for $i = 2$ to $N$ do
  if Scancell($s$) then
    $B$ ← s
    Turnleft(s)
  else
    Turnright(s)
  end if
end for
return $B$
```

In the algorithm, the input consists of the initial image T, where P represents the black cells. The output is the set B, which contains the boundary pixels. The algorithm starts by scanning the cells of T from bottom to top and left to right until a black pixel s belonging to P is encountered. This black pixel s is then inserted into the set B as a boundary pixel. The next step is to set the current pixel P as the starting pixel s and perform a left turn, which means visiting the left adjacent pixel of P. After this, the algorithm proceeds to calculate the area of the border and compares it to draw conclusions. To find the area of the contour, the moment algorithm is utilized:

$$M_{p,q} = \int_{a_1}^{a_2}\int_{b_1}^{b_2} x^p y^q f(x,y)dxdy \qquad (6)$$

In the context of the moment algorithm, $f(x,y)$ represents the pixel value at coordinates (x,y) and $\mathrm{p,q}$ are non-negative integer values ($\mathrm{p,q} = 0,1,2,\ldots$). The moment $M_{0,0}$ is referred to as moment 0, and it represents the border area of the object being analyzed. $M_{0,1}$ or $M_{1,0}$, known as moment 1, contain information about the center of gravity of the object. The area of the contour can be calculated using the following formula:

$$M_{0,0} = \sum_{x=0}^{w}\sum_{y=0}^{h} f(x,y) \qquad (7)$$

where w and h represent the width and height of the image. This formula can be used to calculate the area of the contour. To detect cracks or broken tiles, we follow the algorithm flow chart illustrated in Figure 5, where ε and η are the upper and lower limits of the defect area, respectively.

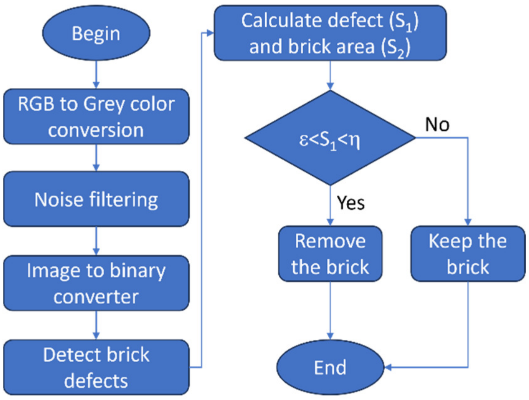

Begin
RGB to Grey color conversion
Noise filtering
Image to binary converter
Detect brick defects
Calculate defect (S1) and brick area (S2)
ε<S1<η
No
Yes
Remove the brick
Keep the brick
End

Fig. 5. Diagram of brick defect detection.

III. RESULTS AND DISCUSSION

In this study, an experimental model was constructed, consisting of a camera and sample bricks arranged in two rows with three bricks in each row. The process involved various steps, as illustrated in Figure 5. The results obtained after converting the color space to grayscale images and applying noise filtering to the bricks can be observed in Figure 6. The results after thresholding and detecting defects are depicted in Figure 7.

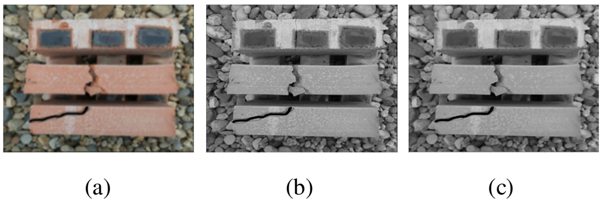


Fig. 6. Bricks after color space conversion and noise filtering: (a) Original image, (b) grayscale image, (c) image after noise filtering.

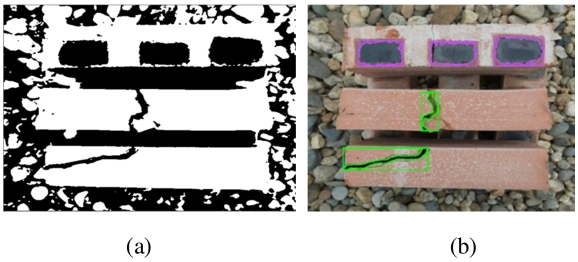


Fig. 7. (a) Thresholded image, (b) defect detection results.

Based on observations from Figures 6 and 7, there are several recognizable signs to identify cracks on the tile surface. These signs include:

- Bisecting Crack: A clear crack that runs across the brick, dividing it into two parts.
- Diagonal Crack: A crack that follows a diagonal path across the brick's surface.
- Small Side Crack: A small crack located near the side of the tile.
- Dark Gray Stains: Dark gray stains on the surface of the brick, indicating cracks that occurred during the firing process.

The crack area of each brick is represented by S_1, while the rectangular area surrounding the cracks is denoted by S_2. The estimated survey area is determined using $\varepsilon = 1500$, and $\eta = 0.5 \times S_2$. If S_1 exceeds 50% of S_2, it is considered to be outside the estimated range. In such cases, it is not classified as a crack, but rather as a burnt gray stain resulting from the firing process. These bricks still meet user requirements and are not considered product defects.

To the best of our knowledge, this study represents the first endeavor to employ machine vision techniques, including threshold filtering, square detection algorithm, and crack area calculation, for pixel-wise crack detection. This method has demonstrated its ability in identifying cracks at various angles and shapes without being limited by training data. This capability enhances the practical applicability of the model and paves the way for its future utilization in real-world scenarios. The proposed method achieves an accuracy rate of approximately 99.2%. However, it is important to note that this implementation was carried out in a well-lit environment. We also conducted tests in a low-light environment, which revealed a decrease in accuracy assessment performance by around 15% and this limitation is acknowledged. Nevertheless, it is worth highlighting that our research presents a unique approach by not relying on training data set selection principles or a uniform sampling method with nearest neighbors. Despite this, the methodology demonstrates diverse assessment performance across different crack shapes. However, it is important to note that there is a limitation regarding the size of the bricks. In future applications, we plan to address this issue by adjusting the model parameters to improve the accuracy of identifying defects in wall bricks. By considering the brick size, we aim to enhance the overall performance of our model and ensure more accurate results in detecting cracks.

IV. CONCLUSION

In this study, an effective machine vision method for detecting defects in wall bricks is proposed. The proposed method includes a comprehensive process to remove noise from the original image, involving color space conversion and the application of a Gaussian filter. Thresholding and the square detection algorithm are utilized to accurately determine the boundaries of bricks and identify the boundary areas of defects. To assess the extent of defects, an estimation range is proposed in the study, allowing for the calculation of the defect area. This estimation range helps determine whether the brick has cracks that need to be addressed or if the observed marks are simply burn marks from the firing process. The tests demonstrate a defect identification accuracy of over 99.2%. The study's notable advantage lies in its ability to identify defects in various fracture directions without the reliance on model training data, unlike most classical methods. Nevertheless, it is important to acknowledge a limitation of the study, which lies in its dependence on sufficiently bright lighting conditions. In cases where the lighting conditions are inadequate, the accuracy of the assessment may decrease by approximately 15%. It is important to note that the scope of this research is limited to processing photos of bricks. For practical applications in large-scale brick factories, future work would involve integrating this method with robotics to automate the defect detection process.

ACKNOWLEDGEMENTS

This research was funded by the Hanoi University of Industry (HAUI) (grant number 05-2022-RD/HĐ-ĐHCN).

REFERENCES

[1] H. Bilal, B. Yin, M. S. Aslam, Z. Anjum, A. Rohra, and Y. Wang, "A practical study of active disturbance rejection control for rotary flexible joint robot manipulator," *Soft Computing*, vol. 27, no. 8, pp. 4987–5001, Apr. 2023, https://doi.org/10.1007/s00500-023-08026-x.

[2] T.-L. Bui and N.-T. Tran, "Navigation Strategy for Mobile Robot Based on Computer Vision and Yolov5 Network in the Unknown Environment," *Applied Computer Science*, vol. 19, no. 2, pp. 82–95, Jun. 2023, https://doi.org/10.35784/acs-2023-16.

[3] N.-T. Tran, T.-D. Ngo, D.-K. Nguyen, P. X. Son, and N. H. Thai, "Mapping and Path Planning for the Differential Drive Wheeled Mobile Robot in Unknown Indoor Environments Using the Rapidly Exploring Random Tree Method," in *The AUN/SEED-Net Joint Regional Conference in Transportation, Energy, and Mechanical Manufacturing Engineering*, 2022, pp. 516–527, https://doi.org/10.1007/978-981-19-1968-8_43.

[4] M. B. Ayed, L. Zouari, and M. Abid, "Software In the Loop Simulation for Robot Manipulators," *Engineering, Technology & Applied Science Research*, vol. 7, no. 5, pp. 2017–2021, Oct. 2017, https://doi.org/10.48084/etasr.1285.

[5] H. Medjoubi, A. Yassine, and H. Abdelouahab, "Design and Study of an Adaptive Fuzzy Logic-Based Controller for Wheeled Mobile Robots Implemented in the Leader-Follower Formation Approach," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 6935–6942, Apr. 2021, https://doi.org/10.48084/etasr.3950.

[6] J. Iqbal, "Modern Control Laws for an Articulated Robotic Arm: Modeling and Simulation," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 4057–4061, Apr. 2019, https://doi.org/10.48084/etasr.2598.

[7] M. Khojastehnazhand, M. Omid, and A. Tabatabaeefar, "Determination of orange volume and surface area using image processing technique," *International Agrophysics*, vol. 23, no. 3, pp. 237–242.

[8] L. Babic, S. Matic-Kekic, N. Dedovic, M. Babic, and I. Pavkov, "Surface Area and Volume Modeling of the Williams Pear (Pyrus Communis)," *International Journal of Food Properties*, vol. 15, no. 4, pp. 880–890, Jul. 2012, https://doi.org/10.1080/10942912.2010.506020.

[9] A. Agarwal and K. Goel, "Comparative Analysis of Digital Image for Edge Detection by Using Bacterial Foraging & Canny Edge Detector," in *2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)*, Ghaziabad, India, Oct. 2016, pp. 125–129, https://doi.org/10.1109/CICT.2016.33.

[10] J. V. N. Lakshmi and R. Kamalraj, "Extracting Pixel Edges on Leaves to Detect Type Using Fuzzy Logic," in *Computational Intelligence in Image and Video Processing*, Chapman and Hall/CRC, 2023, pp. 33–54.

[11] M. Versaci and F. C. Morabito, "Image Edge Detection: A New Approach Based on Fuzzy Entropy and Fuzzy Divergence," *International Journal of Fuzzy Systems*, vol. 23, no. 4, pp. 918–936, Jun. 2021, https://doi.org/10.1007/s40815-020-01030-5.

[12] D. Loverdos and V. Sarhosis, "Automatic image-based brick segmentation and crack detection of masonry walls using machine learning," *Automation in Construction*, vol. 140, Aug. 2022, Art. no. 104389, https://doi.org/10.1016/j.autcon.2022.104389.

[13] M. Ravichand, R. Kumar, B. Hazela, and T. Suthar, "Crack on Brick Wall Detection by Computer Vision using Machine Learning," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, Coimbatore, India, Sep. 2022, pp. 1017–1020, https://doi.org/10.1109/ICECA55336.2022.10009343.

[14] S. Katsigiannis, S. Seyedzadeh, A. Agapiou, and N. Ramzan, "Deep learning for crack detection on masonry façades using limited data and transfer learning," *Journal of Building Engineering*, vol. 76, Oct. 2023, Art. no. 107105, https://doi.org/10.1016/j.jobe.2023.107105.

[15] M. G. Devereux, P. Murray, and G. M. West, "A new approach for crack detection and sizing in nuclear reactor cores," *Nuclear Engineering and Design*, vol. 359, Apr. 2020, Art. no. 110464, https://doi.org/10.1016/j.nucengdes.2019.110464.

[16] Z. Y. Wu, R. Kalfarisi, F. Kouyoumdjian, and C. Taelman, "Applying deep convolutional neural network with 3D reality mesh model for water tank crack detection and evaluation," *Urban Water Journal*, vol. 17, no. 8, pp. 682–695, Sep. 2020, https://doi.org/10.1080/1573062X.2020.1758166.

[17] M. H. Talukder, S. Ota, M. Takanokura, and N. Ishii, "Crack Detection on Brick Walls by Convolutional Neural Networks Using the Methods of Sub-dataset Generation and Matching," in *Deep Learning Theory and Applications*, 2023, pp. 134–150, https://doi.org/10.1007/978-3-031-37320-6_7.

[18] J. Zhang, S. Qian, and C. Tan, "Automated bridge surface crack detection and segmentation using computer vision-based deep learning model," *Engineering Applications of Artificial Intelligence*, vol. 115, Oct. 2022, Art. no. 105225, https://doi.org/10.1016/j.engappai.2022.105225.

[19] H. Bae, K. Jang, and Y.-K. An, "Deep super resolution crack network (SrcNet) for improving computer vision–based automated crack detectability in in situ bridges," *Structural Health Monitoring*, vol. 20, no. 4, pp. 1428–1442, Jul. 2021, https://doi.org/10.1177/1475921720917227.

[20] J. Cao, "Research on crack detection of bridge deck based on computer vision," *IOP Conference Series: Earth and Environmental Science*, vol. 768, no. 1, Feb. 2021, Art. no. 012161, https://doi.org/10.1088/1755-1315/768/1/012161.

[21] Y. Zhang, R. Hou, H. Wang, X. Chen, and P. Yang, "Numerical Simulation of Weak Parts of Main Components of Heavy-Duty Precision Brick Palletizing Robot," *IOP Conference Series: Earth and Environmental Science*, vol. 252, no. 2, Dec. 2019, Art. no. 022109, https://doi.org/10.1088/1755-1315/252/2/022109.