SHANGHAI JIAO TONG UNIVERSITY

Visual Lego© Counter

Course Project of *Fundamentals of Digital Image Processing*

School:	School of Electronic Information And Electrical engineering
Class:	F2103203
ID:	521021910571
Name:	Jinghui Wang

Jan. 14, 2024

Abstract

Lego©, renowned globally, comprises variously shaped and colored pieces. Manual counting by retailers becomes challenging when these pieces accumulate. Retailers must ensure each piece is accounted for to prevent loss or misappropriation. This paper introduces the Visual Lego© Counter (VLC) as a solution. The VLC operates by utilizing shelf-mounted monitors to capture images of the Lego© pieces. An algorithm is proposed and evaluated to count different types of Lego© pieces. Initially, the Segment Anything Model (SAM) [1] is employed for image segmentation, producing cropped images of individual Lego© pieces. Subsequently, ResNet50 [2] was trained using a Lego© pieces dataset [3], enabling it to categorize 22 types of Lego© pieces in test images. The algorithm's effectiveness is then evaluated using test images, culminating in a conclusion.

Contents

I. Introduction

Lego©, with its iconic status in the world of toys and education, has fascinated generations with its simple yet ingenious design. Each piece, distinct in shape and color, offers unlimited possibilities for creativity and innovation. However, this variety also presents a significant challenge in inventory management, particularly for retailers. Accurate counting and categorization of these pieces are essential, not only for inventory control but also for ensuring customer satisfaction.

Figure 1. Various Lego© piece (Source: www.lego.com)

In recent years, the advancement of Digital Image Processing (DIP) technologies has opened new avenues for addressing such challenges. This paper introduces the Visual Lego© Counter (VLC) system, a prime example of integrating DIP advancements with practical retail applications.

Utilizing advanced image processing and recognition techniques, such as the Segment Anything Model (SAM) and ResNet neural networks, the VLC system effectively identifies and quantifies Lego© pieces. Notably, the system ensures high accuracy in identification and counting of Lego© pieces while maintaining a swift processing time. It will free Lego retailers from cumbersome and repetitive counting job, and enable them to keep more accurate and efficient track of every single piece.

The following sections of this paper will explore the intricacies of the Visual Lego© Counter system in depth. Initially, we delve into the related theories that form the foundation of our approach, specifically discussing the principles of the Segment Anything Model (SAM) and the architecture of ResNet neural networks. This theoretical grounding sets the stage for understanding our methodological approach.

In the methodology section, we detail the processes of image segmentation and Lego© piece prediction. Here, we explain how SAM aids in segmenting images of Lego© pieces and how ResNet is utilized to predict the type and count of these pieces. Our experimental section then presents the application of these methods in a real-world scenario, providing insights into the practical effectiveness and limitations of the VLC system.

Finally, the paper concludes with a comprehensive review of our findings, discussing the implications of the VLC system for retailers and the potential for future applications and improvements.

II. Related Theories

II.1. SAM

The Segment Anything Model (SAM) is a revolutionary approach in the field of digital image processing, particularly in the domain of image segmentation. SAM is designed to segment objects in images with high precision, regardless of the object's type or the complexity of the scene. It stands out for its adaptability and effectiveness in diverse applications, from medical imaging to retail inventory management. Figure 2 presents an overview of how SAM works.

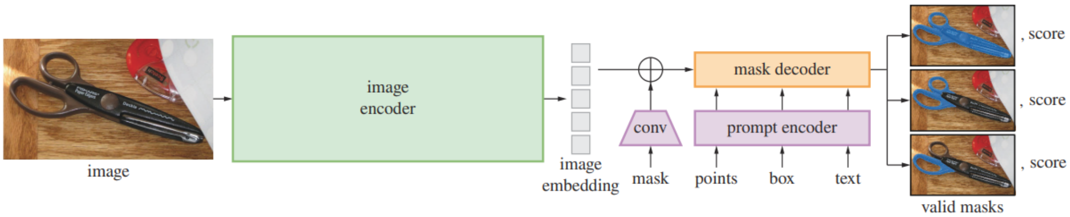


Figure 2. Segment Anything Model (SAM) overview. [1]

Initially, an image is passed through the image encoder, which effectively distills the visual information into a comprehensive set of features—referred to as the image embedding. This embedding captures the essential characteristics of the image, such as shapes, edges, and textures, which are crucial for differentiating various objects within the image.

Following the feature extraction, the process transitions to the mask decoder. Here, the image embedding is combined with other inputs including points, boxes, and text. These inputs provide additional context that guides the segmentation. The points could represent specific locations in the image where objects are present, boxes might delineate the boundary of objects, and text can offer descriptive cues about the objects to be segmented.

The mask decoder then utilizes convolutional operations, denoted as 'conv' in the figure, to interpret the amalgamated data. Through a series of convolutional layers, the decoder progressively refines the segmentation masks. These masks correspond to the different objects identified in the image, with each mask containing a score that indicates the confidence level of the segmentation's accuracy.

The output of SAM consists of these segmentation masks alongside their associated scores, which are indicative of the validity of each identified object. In essence, the masks with higher scores represent segments that the model has identified with greater certainty. This scoring system is particularly useful when processing complex images with multiple objects, as it allows for a prioritization of results based on confidence levels.

In conclusion, the SAM employs a sophisticated architecture that integrates context-aware components with advanced convolutional techniques to deliver precise segmentation results. This system is instrumental in applications where high fidelity in object identification is paramount, such as the Visual Lego© Counter system, where it ensures that each Lego piece is accurately detected and segmented from the rest.

II.2. ResNet

Residual Networks, or ResNets, are a series of deep neural networks that have been instrumental in advancing the field of deep learning. The ResNet architecture is groundbreaking due to its use of identity shortcut connections, which allow for the training of networks that are substantially deeper than those previously used.

The principle behind ResNet is to address the degradation problem — the phenomenon where the network accuracy starts to saturate and then rapidly deteriorates as the network depth increases. This is counterintuitive as deeper networks are expected to capture more complex features and perform better. ResNet solves this problem through the introduction of identity shortcut connections that skip one or more layers. These connections perform identity mapping, and their outputs are added to the outputs of the stacked layers. Unlike traditional networks, where layers are expected to learn the desired underlying mapping directly, ResNet layers are designed to learn residual mappings. Since the learning objective becomes simpler, it is easier to train deeper networks effectively.

ResNets have profoundly impacted image recognition tasks, setting new records in accuracy. Their performance was notably demonstrated in the ImageNet competition, a prestigious image classification challenge. In this competition, ResNet models have outperformed other architectures by a significant margin, leading to theirs widespread adoption. ResNet's ability to learn from a massive number of parameters without overfitting, thanks to its deep structure and residual learning approach, has made it a preferred choice for many image recognition tasks.

The ResNet family includes various models, such as ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152, each differing in the number of layers. The VLC system employs ResNet50, which includes 1 input layer, 48 convolutional layers, and 1 fully connected output layer. The choice of ResNet50 for the VLC is strategic, as it offers a compelling balance between computational efficiency and the capacity to learn complex features. ResNet50 is large enough to model the intricate patterns needed for accurate piece categorization yet compact enough to maintain a reasonable computational load.

III. Method

The VLC consists of two main modules, *image segmentation* and *Lego© piece prediction*. As Figure 3 shows, The VLC operates through a streamlined process beginning with an original image of scattered Lego© pieces, which is segmented by the SAM to isolate individual pieces. These segmented images are then analyzed by the ResNet50 neural network, which has been trained to recognize and classify 22 types of Lego© pieces. The network assigns a prediction for each piece, and the aggregated data is then used to annotate the original image with identified piece counts and categories, providing a clear visual representation of the inventory. This method effectively automates the counting and classification of Lego© pieces, greatly aiding in inventory management.

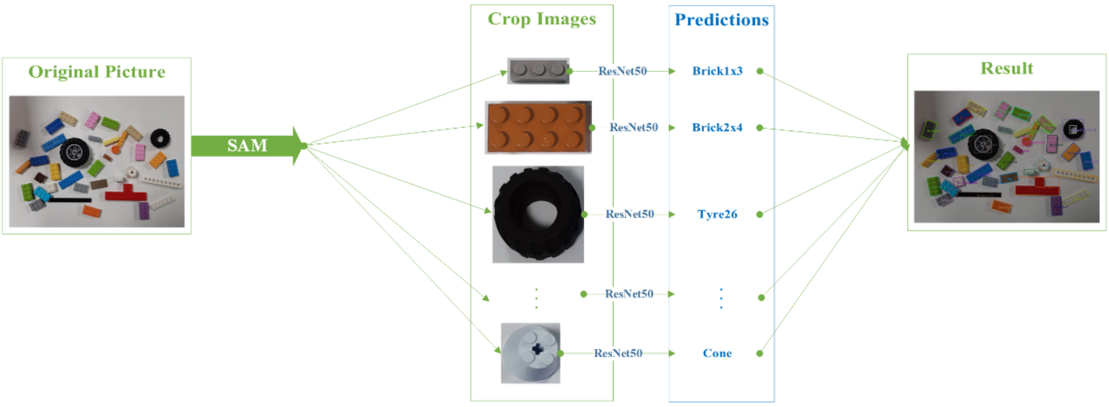


Figure 3. Visual Lego© Counter (VLC) overview.

III.1. Image Segmentation

The image segmentation module of the Visual Lego© Counter (VLC) system is a crucial component that ensures the precise identification of individual Lego© pieces from a complex image. Utilizing the Segment Anything Model (SAM), this module processes the raw images captured by the shelf-mounted monitors, segmenting each Lego© piece for further analysis. The effectiveness of this segmentation stage is paramount, as it directly impacts the accuracy of the subsequent classification and counting processes.

The quality of adaptive image segmentation using SAM depends on the model setup parameters. After testing with all test images, we optimized the parameters, detailed in Table 1 in Appendix. Inadequate parameters can lead to several issues: (1) interlocking Lego© pieces may be incorrectly segmented as a whole; (2) failure to separate two neighboring blocks; (3) segmentation mistakenly focusing solely on the small circular bumps on the blocks; (4) rough split edges that do not match the shape of the Lego© pieces. To resolve issues (1) and (2), we fine-tuned the non-maximum suppression parameters to enhance bounding box precision. To mitigate issue (3), we calibrated the number of points sampled along one side of the image, taking into account the size of the test images. To correct issue (4), we fine-tuned the times of mask prediction on image crops to ensure seamless segmentation.

Even with optimal parameters, the segmentation results produced by SAM often require further refinement for practical application. We first eliminate the segmentation with the largest area, as it typically corresponds to the background; however, this step could inadvertently eliminate the largest Lego© pieces if SAM fails to segment the background. Additionally, we discard any segmentation with an area below a specified threshold to omit the ubiquitous circular bumps on the Lego© pieces. Lastly, we monitor the area occupied by previously extracted crop images to prevent redundancy in segmentation. If a new segmentation significantly overlaps with these areas, it is discarded. This postprocessing guarantees that the majority of invalid segments do not advance to the classification module, thus ensuring the accuracy of the inventory count.

III.2.Lego© Piece Prediction

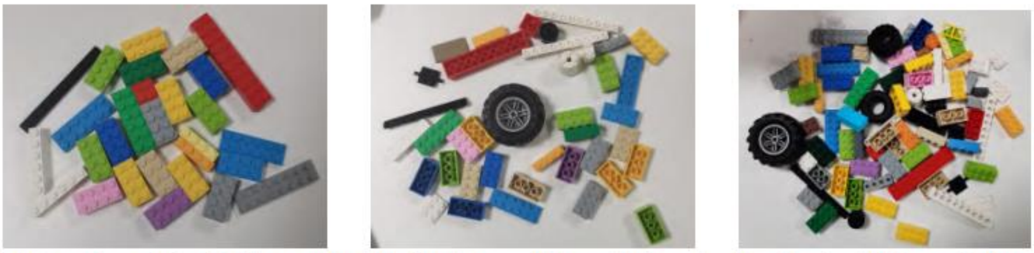
This section of the methodology focuses on the Lego© Piece Prediction module, which utilizes a trained ResNet50 neural network to classify individual Lego© pieces. After successful segmentation, the prediction module's task is to accurately classify each Lego© piece. This is achieved by training the ResNet50 model on a robust dataset[3] that captures the diverse characteristics of Lego© pieces, enabling highly precise recognition and categorization. The dataset includes a variety of Lego© pieces, with types enumerated in Table 2 of the Appendix. Additionally, it is important to note that the model was not trained with every type of Lego© piece found in the test images. For pieces that appear only once or twice in the test images, the model may not accurately predict their type.

To construct a comprehensive dataset, we combined rendered images with photographs. The use of renders allowed us to generate a large volume of training data with varied lighting, angles, and resolution, ensuring a wide representation of possible scenarios. Additionally, we supplemented the dataset with photographs of actual Lego© pieces to capture real-world imperfections and inconsistencies. The dataset[3] does not encompass all types of Lego© pieces, specifically lacking WheelRim14x18, Tyre18, Tyre26. Consequently, we had to create a supplementary dataset. However, time constraints limited our ability to develop a high-quality dataset. As a result, the prediction module encountered difficulties learning these three types of Lego© pieces

For training the ResNet50 model, we utilized the PyTorch framework, taking advantage of its pre-trained ResNet50 model as a starting point. Considering we do not need to differentiate Lego© pieces by color, we trained with grayscale images, reducing computational demands while preserving vital shape and texture information for precise classification. Despite the model's pre-trained status, we chose to train all the layers instead of freezing any. This strategy enables the model to adjust all weights and biases according to our specific dataset, rather than relying solely on features learned from ImageNet (although it significantly increased training time). This comprehensive training process ensures that the model is finely tuned to the nuances of Lego© piece classification, resulting in a more robust and accurate prediction system.

IV. Experiments

We conducted tests on the set of ten test images, which can be broadly classified into three levels of complexity as depicted in Figure 4. Each complexity level comprises 3 to 4 images, totaling 10 images. All these images were captured from a top view, providing a consistent perspective for analysis.



(a) Simple: Only Bricks and Plates (1∼4)

(b) Mediate: More kinds of pieces and more status (5∼7)

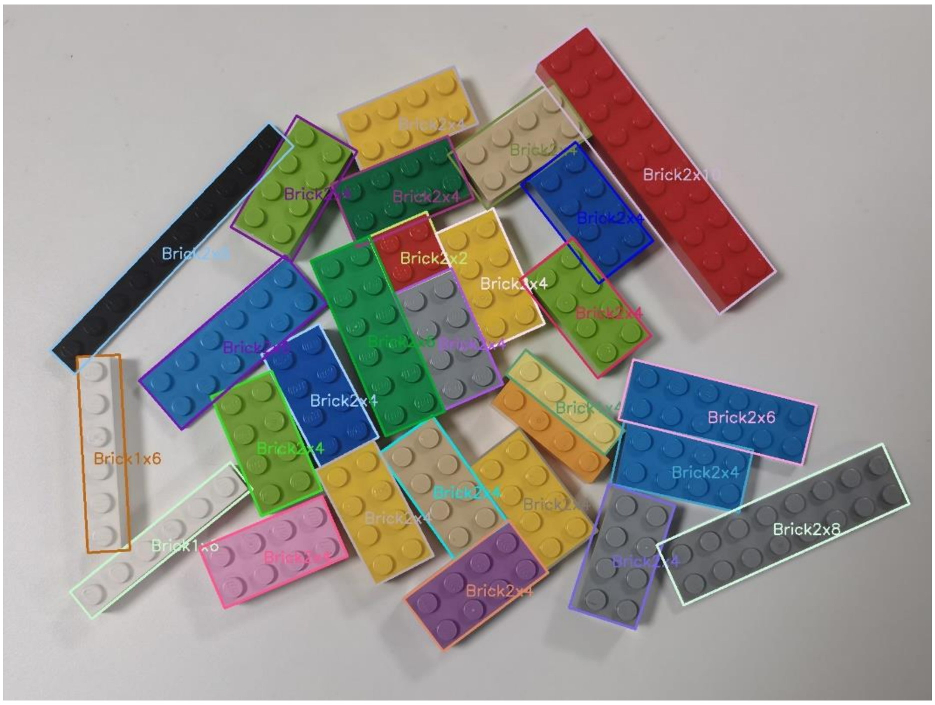
(c) Hard: With occlusions (8∼10)

Figure 4: Sample images from all levels of complexity

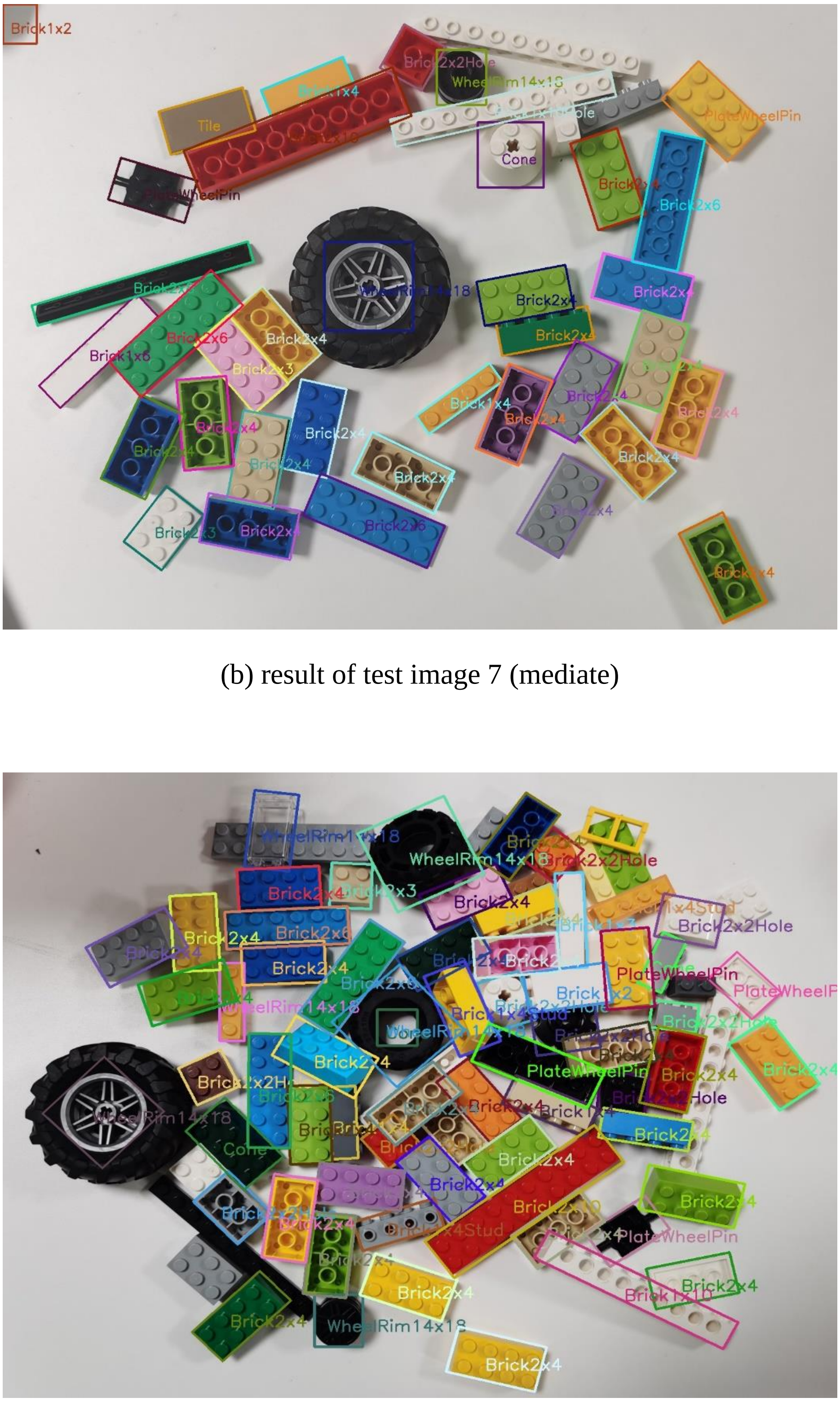
Our model demonstrated a high degree of accuracy in processing images across all three levels of complexity. To illustrate the model's performance, we selected one representative image from each complexity level, which is shown in Figure 5. The detailed processing results and the count of Lego© pieces for all images are provided in the supplementary materials accompanying this paper.

In the 'simple' complexity category, as depicted in Figure 5(a), the model successfully identified the majority of Lego© piece types, with only a few adjacent pieces remaining unidentified. For the 'mediate' complexity images, depicted in Figure 5(b), the model maintained high accuracy despite the various orientations of the Lego© pieces. However, it notably failed to detect Tyre18, as discussed in Section III.1. Additionally, the sensitivity of SAM to shadows led to the erroneous segmentation of a shadow in the top-left corner. In the 'hard' complexity category, characterized by overlapping Lego© pieces and shown in Figure 5(c), the model accurately identified top-layer pieces but struggled with those underneath, leading to less satisfactory outcomes. Furthermore, as detailed in Section III.2, Tyre18 and Tyre26 were not correctly identified.

Concluding the results, our model shows promising capabilities in recognizing and counting Lego© pieces in images of varying complexities. While it excels in simpler scenarios, challenges arise in more complex images, especially with adjacent or overlapping pieces. These findings underscore the potential of the VLC system, as well as areas for future enhancement.



(a) result of test image 3 (simple)



(b) result of test image 7 (mediate)

(c) result of test image 10 (hard)

Figure 5: Result of sample images from all levels of complexity

V. Limitations and Future Work

1. **Shadow and Reflection Removal**: Due to time constraints, the Visual Lego© Counter (VLC) system did not explore complex algorithms to stably remove shadows and reflections from scenes. This limitation has resulted in SAM occasionally producing erroneous segmentations. Future iterations of the system could benefit from incorporating advanced image preprocessing techniques that effectively mitigate these environmental factors, thereby enhancing segmentation accuracy.

2. **High-Resolution Image Processing**: SAM's segmentation of the high-resolution images used in our tests requires significant memory consumption, estimated to be over 20GB. Given hardware limitations, we were compelled to compress the images before segmentation with SAM, leading to a loss of information and subsequent impact on prediction accuracy. Future work could explore more memory-efficient segmentation methods or utilize more powerful hardware to process images at their original resolution.

3. **Segmentation of Overlapping Pieces**： One of the notable challenges with SAM is its inability to effectively segment overlapping Lego© pieces. In our testing, VLC could only recognize pieces that were on top, with reduced accuracy for pieces underneath. Improving the ability to segment and identify overlapped pieces remains a critical area for future development.

4. **Dataset Selection and Cleaning**: The dataset used in this study was not subjected to meticulous data cleaning and processing due to time constraints. For the training of a more accurate predictive model, the careful selection of the dataset based on effective strategies is necessary. Additionally, the dataset lacked representation for some less common Lego© pieces, which were not included. Future research should focus on expanding the dataset to include a wider variety of pieces, especially those that are less common, to enhance the model's overall accuracy and robustness.

5. **Distinguishing Similar Pieces**: the model faces challenges in accurately predicting certain types of Lego© pieces that are extremely similar in appearance, differing only in subtle characteristics that are difficult to discern from images. For instance, the model struggles to distinguish between a brick2x4 and a plate2x4 when viewed from the top, or between a sideways brick2x4 and a brick1x4. Similarly, differentiating between a Brick1x4Stud and a Brick1x4 remains a challenge. These issues highlight the need for advanced feature extraction techniques in future models that can capture and utilize these subtle differences more effectively.

VI. Conclusion

This study has successfully demonstrated the capability and effectiveness of the Visual Lego© Counter (VLC) system in automating the process of counting and classifying Lego© pieces. The system, leveraging advanced digital image processing techniques such as the Segment Anything Model (SAM) and the ResNet50 neural network, addresses a significant challenge in inventory management for Lego© retailers. Our experiments, covering a range of image complexities, have shown that the VLC can accurately identify and count Lego© pieces in most scenarios, with its performance being particularly noteworthy in simpler setups.

Despite its successes, the study also highlights certain limitations of the current VLC system. The system's performance in complex scenarios, especially where pieces are adjacent or overlapping, indicates the need for further refinement. Additionally, the challenges posed by shadows and reflections in images point to the necessity for more advanced image preprocessing techniques. These limitations provide a clear direction for future work and potential improvements, emphasizing the need for more comprehensive datasets and enhanced algorithmic approaches to handle a wider variety of real-world conditions.

References

[1] Kirillov A, Mintun E, Ravi N, et al. Segment anything[J]. arXiv preprint arXiv:2304.02643, 2023.

[2] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[3] Boiński T M. Photos and rendered images of LEGO bricks[J]. Scientific Data, 2023, 10(1): 811.

Appendix

Table 1. Fine-tuned parameters of SAM

Parameter	Value	Meaning
points_per_side	32	The number of points to be sampled along one side of the image.
points_per_batch	64	Sets the number of points run simultaneously by the model.
box_nms_thresh	0.4	The box IoU cutoff used by non-maximal suppression to filter duplicate masks.
stability_score_thresh	0.97	The stability of the mask under changes to the cutoff.
pred_iou_thresh	0.97	The model's predicted mask quality.
crop_n_layers	1	The times of mask prediction on crops of the image.

Table 2. Types of Lego© pieces in the dataset used to train the classifier

Index	ID	Name	Abbreviation
0	17485	Brick 2x2 round with pin hole	Brick2x2Hole
1	2456	Brick2x6	Brick2x6
2	2730	Technic Brick 1x10 with holes	Brick1x10Hole
3	3001	Brick2x4	Brick2x4
4	3002	Brick2x3	Brick2x3
5	3003	Brick2x2	Brick2x2
6	3004	Brick1x2	Brick1x2
7	3006	Brick2x10	Brick2x10
8	3007	Brick2x8	Brick2x8
9	3009	Brick1x6	Brick1x6
10	3010	Brick1x4	Brick1x4
11	30414	Brick 1x4 with Studs on Side	Brick1x4Stud
12	3622	Brick1x3	Brick1x3
13	45176	Cone 3x3x2	Cone
14	4600	Plate 2x2 with 2 Wheel Pins	PlateWheelPin
15	55981	Wheel Rim 14x18 with Holes on Both Sides	WheelRim14x18
16	56145	Wheel Rim 20x30 with 6 Dual Spokes and External Ribs	WheelRim20x30
17	56891	Tyre 18/ 56x17 Off-Road with Offset Centre	Tyre18
18	57520	Technic Sprocket Wheel 25.4	Wheel25.4
19	6111	Brick1x10	Brick1x10
20	70695	Tyre 26/ 49x30 Tract	Tyre26
21	87079	Tile 2x4 With Groove	Tile