The Journal of the Society for Art and Science, Vol. 23, No. 2, pp. 3:1 – 3:13 (2024)

Matching up Stone Tools and Storage Bags via Deep Learning of Stable Posture Images

Mengbo You[1)] **Fumito Chiba**[2)] **Kouichi Konno**[1)]

1) Iwate University 2) LANG CO., LTD.

Abstract

Ancient stone tools excavated from ruins are important materials for archaeological research. The manufacturing process of stone tools is revealed by frequently assembling and disassembling the joining materials. Each stone tool is stored in a bag with its identification number and photo. However, storage and management of stone tools may occur human errors, such as mis-recognition and mistaking of the storage bag. This study proposes an image-based identification method to match each stone tool with its corresponding storage bag. Two commonly used stable postures were defined for each stone tool. The collected image dataset is used for deep learning of an untrained CNN model and seven pre-trained models. The experimental results showed better accuracy and processing speed than those of previous research. Finally, three of the models with high classification performances are selected to construct the detector with the YOLO framework for practical scenario.

1 Introduction

Stone tools and their fractured objects play an important role in understanding the relationship between stone tools and human organizational strategies and in finding evidence for the origin and spread of stone-tool technology [1, 2]. For instance, the stone tools were made by crushing large mother rocks by hitting them with stones or bones, and arranging them into shapes suitable for their intended use. The collection of excavated stone tools that are combined to reconstruct the original mother rock is called joining material. To investigate the joining material in the prehistory period, stone tools excavated from ruins are cleaned, numbered, and categorized. To reveal the process of making stone tools, archaeologists use these stone tools to assemble and disassemble the joining material through trial and error.

To avoid trial and error, it is crucial to obtain the optimal spatial arrangement and document the assembly orders for guiding the assembly and disassembly operations. Takahashi et.al [3] proposed a partial matching method to reconstruct the spatial arrangement of the stone tool point clouds used in the joining material. Yang et.al [4] visualized the spatial arrangement by a hierarchical tree in which each stone tool is represented by the 3D shape data in each node. Although the visualization of the spatial arrangement is easy to understand, it still takes time to find the actual stone tools according to the identification number while reassembling the joinery materials. Because the correlation between the actual stone tools and the data needs to be established. To establish such a correlation, general archaeological centers and similar institutions typically opt for directly inscribing data onto the surface of an artifact. However, as shown in Fig. 1, LANG CO., LTD. opts to store each stone tool in a bag and use a piece of paper with the identification number and the PEAKIT image (typically a photograph) affixed to the bag [5]. This approach ensures that the surface of a stone tool remains unobstructed by text or markings. The assembly order can be represented by recording the order of the identification numbers. Tanaka et.al [6] proposed a video recording method to assist the reassembly of components by recording and reproducing the order of disassembly. However, in the case of stone tools, once mixed with other stone tools during disassembly, it is also difficult to recognize each stone tool according to the storage bag. Manual management of a number of stone tools may also lead to mistakes, such as returning stone tools to incorrect bags, losing identification paper pieces, or other manual managements along with stone tool investigation. To avoid such mistakes during manual managements, identifying stones automatically will bring great convenience.

Figure 1: The current stone tool management system.

This paper proposes a novel stone identification method based on stable posture images using the convolutional neural network (CNN) framework. The proposed method needs to capture the image of the stone tools using a camera above the working desk. The posture in which a stone can be placed on a desk is referred to as a stable posture. When observing a stone in such a stable posture from directly above, the in-plane rotation caused by different viewing angles result in different image rotation angles in the captured images, hereinafter referred to as different angles. Since different postures of a stone may have different appearances, all the appearances of stable postures should be considered. In this study, only thin stone tools with two stable postures are considered. A number of images are collected for each stable posture, and these images are assigned with a unique class label, representing its identification number and its stable posture index. We firstly make a dataset with the images only containing a single stone tool by cropping out the desired region. Then, a customized CNN model was built and trained on this dataset. In comparison, seven pre-trained models are transferred to verify the classification ability to distinguish one stable posture of a stone from another posture or other stones. Several evaluation metrics were considered comprehensively to select the optimal models. Finally, three promising models are selected to

construct the detector using the YOLO framework. The trained detector can output the predicted label for each stone. Using the predicted label information, users can easily recognize the stone and put the stone back into its corresponding bag.

2 Related work

Considering the stones have complex 3D shape features and show different appearances from different sides, the point cloud will be the first choice to describe the shape features. To compare the point cloud of a stone with that of all stones in the database, a common solution is to perform local registration and evaluate the difference with each category. Local registration algorithms assumes point cloud pairs to start in close alignment and then refine their alignment. The most popular local approach is Iterative Closest Points (ICP), both point-to-point [7] and point-to-plane [8] along with their many variants [9]. These methods guarantee convergence only when the scanned pairs are roughly aligned to start with. When point cloud pairs start in arbitrary initial poses, registration requires solving a global problem to find the optimal alignment through a rigid transform across the 6 degrees of freedom (6DOF) space, encompassing translations and rotations. A popular strategy is to invoke RANSAC to find the aligning triplets of point pairs [10]. Various alternatives have been proposed to improve the complexity using special four point basis instead of triplets as basis in RANSAC, such as the 4PCS algorithm [11], and SUPER 4PCS algorithm [12]. Although the point cloud registration can be used to accurately compute the difference with each category, it is time-consuming to register the unknown stone to all categories and find the desired category with smallest registration distance. Instead of using all points in the point cloud in the matching process, a shape-based matching scheme was proposed to match the unknown object mask in videos with known 3D model silhouettes [13]. The matching score was calculated for each 3D model in the video, and the highest score was used as the prediction result. This method requires massive matching pairs between various object poses in the video and all 3D models, resulting in a low processing speed. Recently, a region proposal network was proposed to detect semantic parts inside multiple views for 3D shape classification [14]. This method embeds part- and view-level attention mechanisms into the deep learning framework to highlight the discriminative parts from the 3D global features and discriminative views from multiple viewpoints. However, in contrast to the objects of cars, chairs, and aircrafts used in their validation dataset, the attention mechanism is not applicable to stone tools because of the lack of protrusions in the semantic parts. Many deep-learning-based 3D point cloud classification methods have been validated on public datasets of large objects, such as cars, chairs, and aircraft. These objects have clearly discriminative shape features when compared to each other. However, stone tools have few protrusions and textures that distinguish them from other stone tools. Applying deep learning-based methods to stone tools is still a challenging problem, and there is few literature reported on the subject.

On the other hand, as a crucial topic in the field of computer vision, image-based object recognition has proven to be effective and rapid over the past decade, especially in the case of facial recognition. However, most faces are depicted within rectangular regions in 2D images, and the shared facial features lead to a fixed arrangement of characteristics, such as the eyes, nose, and mouth typically appearing at specific positions within the rectangular region. In contrast, the size and shape of stone tools are uncertain, and obsidian may present a textureless appearance under light illumination, making feature extraction challenging due to extensive reflections. Therefore, image features have been used as an unreliable feature for auxiliary screening. For example, a previous study [15] used the contours extracted from stone tool images to narrow down the search range and then used ICP to find a few candidates for the best match. The image contour can indeed filter out a few candidates to assist with subsequent point cloud matching. However, once the correct solution does not appear in the candidate list, the opportunity to find the correct solution is also missed in the subsequent point cloud matching. To alleviate this issue, the solution is to increase the number of candidates filtered out. However, as the number of candidates increases, the time taken to match the reference to all candidates increases exponentially.

In this paper, we focus on applying image ob-

ject recognition techniques to stone tools. A well-established image classification framework based on CNN is used to evaluate the classification performance of stone tools. However, to successfully train the deep learning model, three main challenges need to be addressed. Firstly, deep learning typically requires a large dataset for training. Capturing images of individual stone tools one by one would be labor-intensive. To build the detector for recognition from a scene image, annotating the rectangular region for each stone tool in all images is also indispensable. Secondly, stone tools are usually placed on the desktop in random orientations each time they are presented, making it difficult to define predefined angles for their positioning. This necessitates a recognition system capable of handling all possible angles. Moreover, thinner stone tools often have two distinct sides, each displaying different appearances. This demands the recognition system to account for at least two sides. Finally, it is essential to establish the concept of ground truth. This involves defining the appropriate categorization of stone tools and ensuring that each category contains a sufficient amount of data to effectively discern distinguishing image features from other categories.

3 Proposed Method

This paper proposes a stone identification method based on stable posture images using the CNN. Object recognition based on images has been developed for decades[16, 17, 18]. Similar with the surface point cloud, the stone image captured by monocular camera also contains the upper side and no occluded parts. To solve this problem, we build the database according to different stable postures. Each unknown stone is compared with all angles of both stable postures of all stones in the database, and the posture with the largest similarity is predicted as its category information, ie. identification number and posture index. To extract image features, a simple custom untrained CNN model was constructed and trained. Furthermore, to select the optimal feature extraction CNN models, the pretrained CNN models of AlexNet [19], GoogLeNet [20], SqueezeNet [21], ResNet-18 [22], ResNet-50 [22], Inception-v3 [23], and ResNet-101 [22], were transferred to extract distinguishing features for stones. For selecting feature extraction layers from pretrained models, transfer learning has showed good performance compared with untrained CNN models [24, 25]. All the pretrained networks require the replacement of the final fully connected layer to fit the number of classes in our task. After feature extraction, an end-to-end convolutional network is used for stone tool detection inspired by YOLO-v2[26], which shows good performance without a complex processing pipeline.

3.1 Image Acquisition

The images of stone tools are acquired by the smartphone camera of iPhone SE (2nd generation) from 45cm straight above the desk, as shown in Fig. 2. To keep the appropriate gap between stones and distance to the edge of the captured image, 4 pieces of stones are captured to an image together. Theoretically, by setting the camera higher or changing to use cameras with a wider photographing range, more stones can be captured simultaneously. After acquiring an image of the 4 stones, the stones are rotated in the plane parallel to the desk and prepared to capture another image. The image with in-plane-rotation is different by lighting directions and position of shadows. However, both the image before and after rotation belong to the same stable posture of the same stone. To acquire the other stable posture of the stone, we manually flip the stone to make it front-side down and capture images using the aforementioned method. After taking pictures of all the stable postures of all stones, the database can be built by labeling these images.

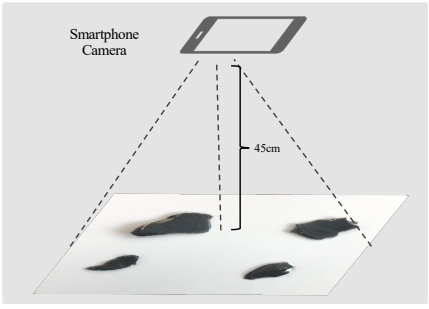

Figure 2: Image acquisition of stone tools.

Since we have 12 pieces of stones in the lab and each stone has 2 stable postures, 24 classes are defined and each class label is combined with the stone

identification number and stable posture index. To organize the database effectively, we make 24 folders named with the class label and each folder contains the images belonging to its class. However, since each image contains 4 pieces of stones and large area of background, we crop the boundary box region of each stone out to be saved as an independent image file. When users place an unknown stone on the desk, controlling the in-place rotation angle of the stone becomes challenging. Thus, the images in the database should contain all possible in-plane rotation angles for each stable posture. To achieve this goal, in addition to manually adjusting the stone tool to different angles, an auto-rotation process is applied to all images in the dataset. As shown in Fig. 3, the captured images are rotated counterclockwise to a series of successive angles. Subsequently, the rotated bounding box of each stone is cropped out and stored in the database.

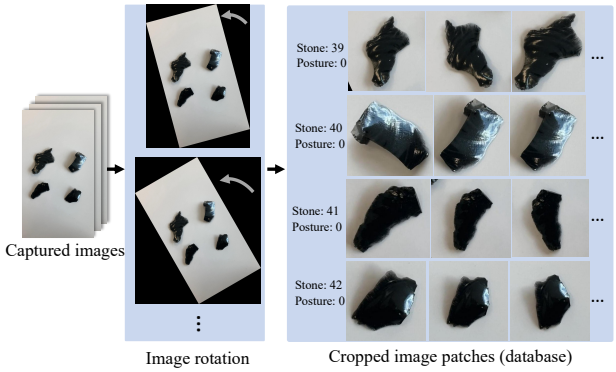

Figure 3: The database consists of stone patches cropped from both captured images and their rotations.

3.2 Feature Extraction

Stone tool detection needs to handle all possible rotations and postures, which is different from general object detection problems, such as face detection, pedestrian detection, and vehicle detection. The general object detection method always employs a bounding box to denote the position and size of the target object and extracts useful features from the rectangular area. However, the bounding boxes for stones have different size and aspect ratio. Although all bounding boxes are saved according to its original size and aspect ratio, they will be resized to fit the input layer of CNN models for classification. However, the YOLO framework learns to make proposals by anchors of pre-defined sizes and its input layer accepts the whole image, which does not have the resizing problem.

Previous image-based stone methods use the edge characteristic to distinguish a stone from another. The contour extraction algorithm can be used to extract the stone edge information. However, the parameters must be set manually according to different scenarios, and the shadow may affect the extracted contour shape slightly. Hence, the edge characteristic is less reliable than color images because it ignores all the color information inside the stone area.

In this study, color images are used to extract distinguishable features by the CNN architecture. To validate distinguishing ability learned by the CNN architecture, classification experiments are designed to evaluate the performance on all 24 classes. If the CNN architecture can classify these classes correctly after training, the CNN architecture can be used to identify any stones stored in the database. Firstly, a custom CNN model with 15 layers was constructed. The input layer was designed to accept an image size of $100 \times 100 \times 3$.

The four-layer module of a convolution layer, batch normalization layer, rectified linear unit (ReLU) layer, and max-pooling layer are repeated twice. The convolution layer extracts a feature map from the input image. The batch normalization layer is set between the convolutional and ReLU layers to accelerate the training and reduce the sensitivity to network initialization. The final module uses a fully connected layer, a softmax layer, and an output layer to replace the max-pooling module in previous modules. The output layer contains 24 units to show the predicted confidence of the input image. The unit with highest confidence is used as the predicted class. This customized network was designed to validate the classification ability with a simple structure.

In addition, seven pretrained CNN models learned from ImageNet were fine-tuned: AlexNet, SqueezeNet, ResNet-18, GoogLeNet, ResNet-50, Inception-v3, and ResNet-101. In the case of AlexNet, the last three layers, i.e., a fully connected layer, softmax layer, and classification layer, are replaced by new layers that fit the curb recognition task. The other pretrained models require similar modifications in the last few layers before training on curb images. In other words, the early layers are reused

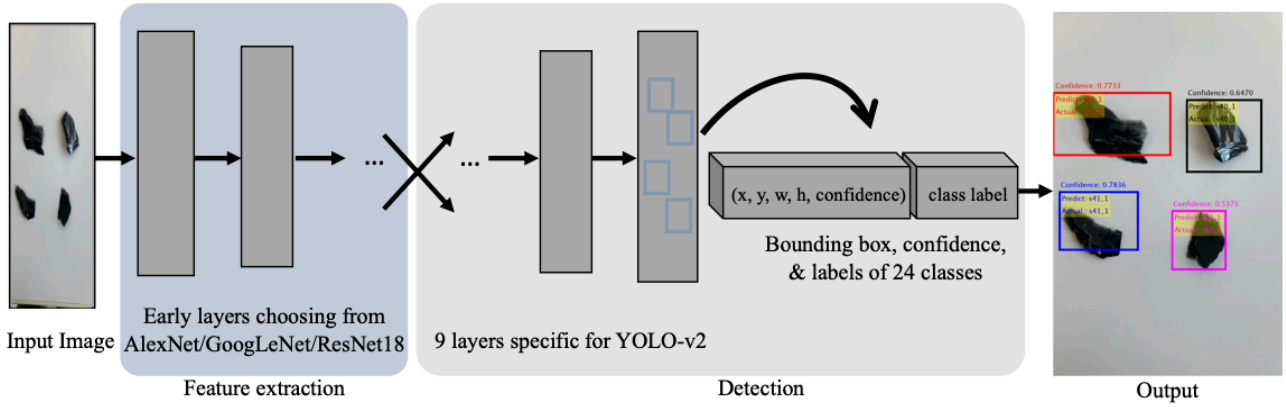

Figure 4: The architecture of YOLO-v2 detection network for stone identification from the entire image.

for image feature extraction. By freezing the parameters in the early feature extraction layers, the training process of transfer learning is much faster and easier than training a network with randomly initialized parameters. The pretrained models are initially trained to classify the images into 1000 classes of objects. In our case, they are transferred to classify images into 24 classes of stones. However, the input images must be resized to satisfy the requirements of the pretrained model. For instance, AlexNet requires input images with a size of $227 \times 227 \times 3$, Inception-v3 requires input images with a size of $299 \times 299 \times 3$, and other models require a size of $224 \times 224 \times 3$. The training results are reported in Section 4.

3.3 Detection

The aforementioned CNN architectures can be used to predict the class of a small image patch. This patch should only contains one stone normalized to the center and no large area of background. However, they cannot directly localize the stone region from the entire image. To address this issue, the YOLO-v2 detection network was connected to the CNN feature extraction architecture, as shown in Fig. 4. The reason of choosing YOLO-v2 is that it is currently the latest version to support C++ code generation and we plan to combine this work with our previous work implemented in C++ code. According to the classification performance, three promising CNN feature extraction architectures were selected to build the detector: AlexNet, GoogLeNet, and ResNet-18. For instance, to combine ResNet-18 with YOLO-v2, the first 140 layers were preserved for feature extraction. The following layers were replaced by nine layers specific to YOLO-v2, which contains convolutional layers, transform layers, and the final output layer. The transform layers extract the activations of the last convolutional layer and constrains the location predictions to fit the locations of the ground truth. YOLO-v2 uses anchor boxes to predict bounding boxes and to predict the class label with a confidence score.

4 Experimental Results

The classification experiment was conducted to verify whether CNN architectures can extract distinguishable features. Then, three promising CNN models were embedded into the detection framework as feature extractors to detect stones automatically from entire images.

4.1 Classification

A total of 10,418 stone images are acquired to store in the database for classification experiments. Among them, 1359 images are individually captured and rotated for testing. 9059 images are used to train the CNN models, in which 15% are randomly split out for validation, and 85% for training. As shown in the database part of Fig. 3, the stone images have different sizes, such as 319×473×3, 529×583×3, etc. The custom CNN model and seven other pretrained models were evaluated on this database. To transfer weights from the pretrained network learned from massive images from ImageNet, only a few layers of the pretrained CNN were replaced with new ones, and the remaining weight parameters in the original model were

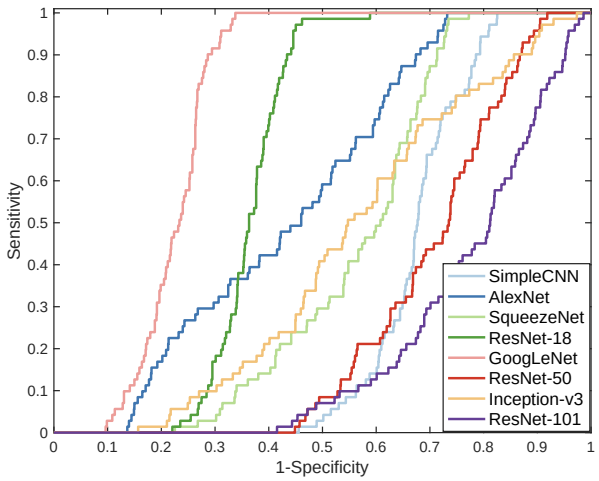

Figure 5: ROC curves of all classification models.

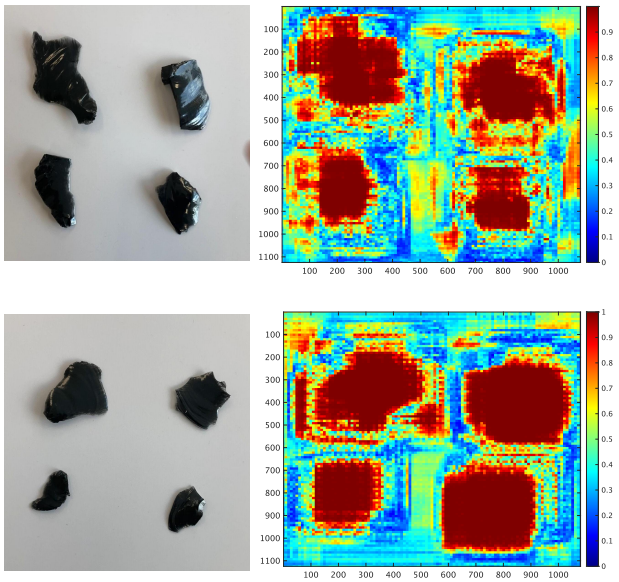

Figure 6: The original images (left) and their corresponding confidence maps (right).

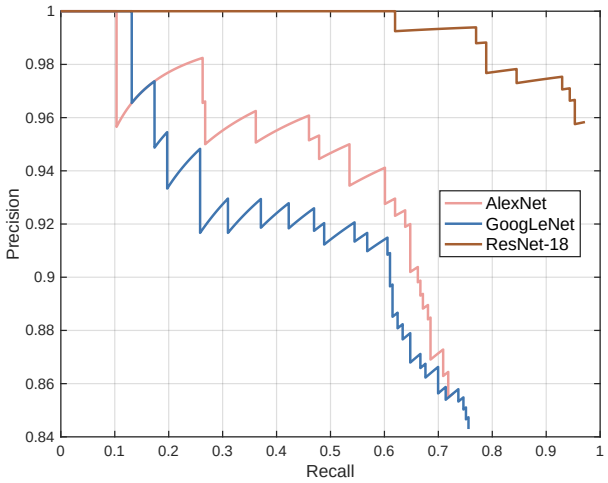

Figure 7: The PR curves of 3 detectors.

preserved. For example, in GoogLeNet, a fully connected layer and a classification layer were replaced to perform the new classification task, and the other weights were preserved. For comparison, the same initial parameter configuration was used. The initial learning rate was set to 1×10^{-3}. The stochastic gradient descent with momentum optimizer was used with a minibatch size of 64, weight decay factor of 1×10^{-4}, and momentum of 0.95. The maximum epoch number was set to 6. The class label links the stone identification number and stable posture index together. For instance, $s39\_0$ means the stone with identification number 39 and stable posture 0. For each class label, the test images were considered as unknown images to test whether the predicted label matches the ground-truth label. The matched images are considered as positive, and the dismatched images are considered as negative. The ROC curves of different models is shown in Fig. 5, where GoogLeNet has the largest value of AUC, ResNet-18 achieves the second place, and AlexNet is the third place. This result means that these three models are robust to varying prediction threholds.

To evaluate the classification performance of different models, accuracy, sensitivity, specificity, F1-score, precision, AUC, and prediction time were used. The calculation methods for these evaluation metrics can be found in the appendix. Table 1 compares the evaluation results of different CNN models. The best results of each metric are highlighted in bold. The prediction time is the time taken to predict all the 190 test images. The optimal model can be chosen according to different metrics. Inception-v3 had the best performance in terms of accuracy, sensitivity, specificity, and F1-score, but this performance comes with the long prediction time. GoogLeNet achieved the highest specificity and AUC value. The custom CNN required the least prediction time, mainly because of its simple structure and the small size of the input image. However, the balance between performance and time cost is important when choosing a model. Especially for a practical stone identification system, the classification must be accurate and fast. Because the most accurate model of Inception-v3 did not achieve a large gap than that of the second and third places, we choose GoogLeNet, ResNet-18, and AlexNet as

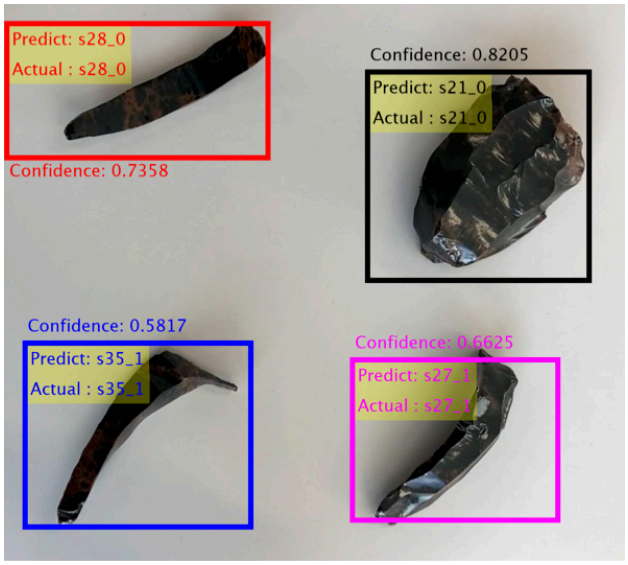

Figure 8: Examples of successfully detected curbs.

Figure 9: Examples of detection failures.

our optimal feature extractors for building detectors. Then, the sliding-window method was used to classify every possible position with a size of 350×350×3 on a large image of size $1080 \times 1920 \times 3$ to calculate the confidence map, as shown in Fig. 6. The high-confidence positions have a high probability to be a stone. By comparing the original image with its corresponding confidence map, we find that not only the actual stone positions but also their neighboring positions achieve high confidence values. This means it is feasible for detector to extract bounding boxes containing stones. In the next section, the three optimal models of GoogLeNet, ResNet-18, and AlexNetTo are embedded into the detection framework.

4.2 Detection

Three feature extraction networks of the first 16 layers of AlexNet, the first 111 layers of GoogLeNet, and the first 66 layers of ResNet-18 were used as feature extractors and embedded in the YOLO-v2 framework for detection. 2354 images are collected for training and 428 images for testing. Each image has 3 or 4 stones. The testing images are also captured individually to avoid duplication with the training images. The image size varies from $1080 \times 1920 \times 3$ to $2121 \times 2121 \times 3$ because of different rotation angles. The ground-truth data contains the annotations of the bounding boxes and their corresponding class label. For comparison, the three detectors were configured with the same initial parameters. The maximum epoch number was set to 10. The precision recall curve is used to evaluate the performance of the 3 detectors, as shown in Fig. 7. To numerically compare the curves, the evaluation metric of average precision (AP) is used to evaluate both the ability of the detector to find all stones and the ability to predict correctly, which is also the average precision over the curve. The calculation method of AP can be found in the appendix. The AP values and average detection speed are shown in Table 2. YOLO-v2 with ResNet-18 is the best model in terms of AP. However, YOLO-v2 with AlexNet achieves the fastest detection speed with a large compromise on AP. The detection speed is calculated on the PC of MacbookPro (14 inch, 2021), with CPU of Apple M1 Pro and memory of 16GB. The examples of successfully detected examples are shown in Fig. 8. Both square bounding boxes and narrow ones are detected accurately. However, there are also some cases that the detector failed to detect, as shown in Fig. 9. There are 4 stones and only 2 of them are detected. The main reason for the failure is that the maximum confidence values in all categories are still far below 0.5. For instance, two successfully detected instances have confidence scores of 0.4121 and 0.3623, while the confidence scores for two undetected instances are even lower. Certainly, it is possible to detect all instances by adjusting the confidence threshold, but this approach would also introduce false alarms and lead to less accurate categorization. To address this issue, we plan to further perform a double-checking process using point cloud matching for regions with insufficiently

Table 1: Evaluation of transfer learning with different pre-trained CNN models. (Best results are highlighted in bold)

Model	#Layers	Input Size	Accuracy	Sensitivity	Specificity	F1−score	Precision	AUC	Prediction Time
CustomCNN	15	101×101×3	0.979	0.680	0.989	0.568	0.599	0.411	**5.569**
AlexNet	25	227×227×3	0.991	0.825	0.995	0.798	0.809	0.558	10.579
SqueezeNet	68	227×227×3	0.988	0.882	0.994	0.684	0.692	0.565	10.871
ResNet-18	72	224×224×3	0.990	0.895	0.995	0.750	0.804	0.540	16.482
GoogLeNet	144	224×224×3	0.995	0.955	**0.998**	0.856	0.869	**0.568**	17.889
ResNet-50	177	224×224×3	0.994	0.946	0.997	0.825	0.828	0.360	35.703
Inception-v3	316	299×299×3	**0.996**	**0.961**	**0.998**	**0.887**	**0.885**	0.414	58.623
ResNet-101	347	224×224×3	0.995	0.950	0.997	0.839	0.838	0.374	73.446

high confidence scores. This approach aims to enhance accuracy, albeit at the expense of speed. However, in practical scenarios, it is worth exploring if a slight decrease in speed can result in improved accuracy.

Table 2: The *AP* and detection time of 3 detectors. (Detection time is calculated on the testing image size of $1080 \times 1920 \times 3$. Best results are highlighted in bold)

Model	#Layers	*AP*	Detection Time
YOLO-v2 (AlexNet)	25	69.40%	**206.84ms**
YOLO-v2 (GoogLeNet)	120	70.25%	477.57ms
YOLO-v2 (ResNet-18)	75	**96.57**%	396.50ms

4.3 Discussion

Compared with the previously proposed method [15], the proposed method (using a custom CNN as an example) has the following advantages:

Firstly, less time is required to achieve similar accuracy. In terms of accuracy, [15] selected the top five best matches on the same dataset of 12 stone tools. Three of them are not listed in the candidate table, which means that they fail to be selected. Hence, the accuracy was 75% (calculated using 9 correct samples / 12 test samples). Our custom CNN method achieved an accuracy of 97.9%. In terms of time cost, [15] needs five times the point cloud matching for each stone tool, and the time for each matching pair was 2 s. The time required to check each reference stone tool was 10 s. However, our method requires 0.008 s (calculated as 5.569 s of total time cost / 1359 test images × 2 postures for each stone tool) to complete the identification for each stone tool. The difference in time may also be due to differences in computer specifications: an i7-9700 CPU was used in [15], and Apple M1 Pro was used in our work. On the other hand, when there were no concerns about time consumption, in [15], 30 candidates were retained for each stone tool, and all 12 test samples were correctly listed in the table, but the time cost reached 330 s.

Secondly, more convenient way to acquire data. [15] used the 2D image contours to screen candidates, and 3D point cloud registration to estimate the matching score between each reference and each candidate. Both 2D images and 3D point clouds are required to obtain reference data, which requires a depth camera. To build the database, a laser scanner is necessary to obtain an accurate point cloud without occlusion. The proposed method requires only a monocular camera.

Thirdly, potential to handle various working environments. The result of [15] is based on the successful extraction of contours for each stone tool. Contour extraction is easily affected by lighting conditions and image background. Moreover, the surface point cloud captured by the depth camera may contain noise, which may also affect the final precision. The CNN-based image identification method used in our study has the potential to handle various lighting conditions and backgrounds.

5 Conclusion

A deep learning framework was proposed to identify stone tools automatically from the images captured by a monocular camera. A customized CNN model and seven pre-trained models were trained for classification. By comparing all the models with evaluation metrics, such as, accuracy, and prediction time. Three promissing models were employed as feature extraction networks and embedded into the YOLO framework to construct stone detectors. The detection performance was evaluated by the average precision on the stone image dataset. Through extensive experiments and analysis of the results, the effectiveness of the proposed method was verified, providing clear

guidance on how to choose pre-trained networks for stone detection. Future plans include improving the accuracy and reducing the unnecessary processes of the proposed method for practical use and estimating the performance for more stones. To ensure the detection result to be correct, we also plan to combine the image based recognition and point cloud matching for double-checking.

6 Acknowledgements

The basic concept of stone tool classification part in our method has already been presented at the Nicograph International 2023 [27], and we extended the concept to the stone tool detection from a scene using promising classification models in this paper. A part of this work was supported by JSPS KAKENHI Grant Number JP22K00998.

References

[1] William Andrefsky. The analysis of stone tool procurement, production, and maintenance. *Journal of archaeological research*, 17(1):65–103, 2009.

[2] Yue Hu, Ben Marwick, Jia-Fu Zhang, Xue Rui, Ya-Mei Hou, Jian-Ping Yue, Wen-Rong Chen, Wei-Wen Huang, and Bo Li. Late middle pleistocene levallois stone-tool technology in southwest china. *Nature*, 565(7737):82–85, 2019.

[3] Tsukasa Takahashi, Mengbo You, and Kouichi Konno. A study on partial shape matching between flake surface and surface of joining material using measured point cloud. *The journal of the Society for Art and Science*, 22(1):1:1–1:10, 2023.

[4] Xi Yang, Kouichi Konno, Fumito Chiba, and Shin Yokoyama. Visualization of flake knapping sequence with analyzing assembled chipped stone tools. *The Journal of Art and Science*, 18(1):40–50, 2019.

[5] Fumito Chiba, S Yokokoyama, Akihiro Kaneda, and Kouichi Konno. Development of network-type archaeological investigation system. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5):99, 2015.

[6] Ryuuta Tanaka, Mengbo You, Kouichi Konno, and Takamitsu Tanaka. A study on component reassembly assist method using simple marker by recording and replaying disassembly order. *The journal of the Society for Art and Science*, 22(2):3:1–3:11, 2023.

[7] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[8] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[9] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001.

[10] Chu-Song Chen, Yi-Ping Hung, and Jen-Bo Cheng. Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1229–1234, 1999.

[11] Dror Aiger, Niloy J Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008.

[12] Nicolas Mellado, Dror Aiger, and Niloy J Mitra. Super 4pcs fast global pointcloud registration via smart indexing. *Computer graphics forum*, 33(5):205–215, 2014.

[13] Alexander Toshev, Ameesh Makadia, and Kostas Daniilidis. Shape-based object recognition in videos using 3d synthetic object models. In *2009 IEEE conference on computer vision and pattern recognition*, pages 288–295. IEEE, 2009.

[14] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Fine-grained 3d shape classification with hierarchical part-view attention. *IEEE Transactions on Image Processing*, 30:1744–1758, 2021.

[15] Yoshiki Sawada, Tsutomu Kinoshita, Amartuvshin Renchin-Ochir, Fumito Chiba, and Kouichi Konno. Stone tool identification method based on measured points by rgb-d camera and points of stone tool database. *The journal of the Society for Art and Science*, 21(4):213–224, 2022.

[16] Kasthurirangan Gopalakrishnan, Siddhartha K Khaitan, Alok Choudhary, and Ankit Agrawal. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 157:322–330, 2017.

[17] Min Zou, Mengbo You, and Takuya Akashi. Application of facial symmetrical characteristic to transfer learning. *IEEJ Transactions on Electrical and Electronic Engineering*, 16(1):108–116, 2021.

[18] Min Zou, Mengbo You, and Takuya Akashi. Reconstruction of partially occluded facial image for classification. *IEEJ Transactions on Electrical and Electronic Engineering*, 16(4):600–608, 2021.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[21] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[24] Qijie Wei, Xirong Li, Hao Wang, Dayong Ding, Weihong Yu, and Youxin Chen. Laser scar detection in fundus images using convolutional neural networks. In *Asian Conference on Computer Vision*, pages 191–206. Springer, 2018.

[25] M Waqar Akram, Guiqiang Li, Yi Jin, Xiao Chen, Changan Zhu, and Ashfaq Ahmad. Automatic detection of photovoltaic module defects in infrared images with isolated and develop-model transfer deep learning. *Solar Energy*, 198:175–186, 2020.

[26] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[27] Mengbo You and Kouichi Konno. Matching up stone tools and storage bag using image identification with cnn. In *NICOGRAPH International 2023*, pages 001–004, 2023.

7 Appendix

7.1 Evaluation Metrics for Classification

Accuracy denotes the number of correctly labeled samples. In this study, it denotes the proportion of the correctly identified image numbers to the number of test images, which can be written as

$$\text{Accuracy} = (n_{\text{tp}} + n_{\text{tn}})/(n_{\text{tp}} + n_{\text{fp}} + n_{\text{tn}} + n_{\text{fn}}). \quad (1)$$

where n_{tp} denotes the number of true-positive samples, n_{tn} denotes the number of true-negative samples, n_{fn} denotes the number of false-negative samples. Sensitivity is the number of correctly identified positives divided by the number of true positives. In this case, it denotes the number of correctly identified curb images divided by all curb images, which can be written as

$$\text{Sensitivity} = n_{\mathrm{tp}}/(n_{\mathrm{fn}} + n_{\mathrm{tp}}). \tag{2}$$

Specificity is the the number of correctly identified negatives divided by the number of true negatives. In this case, it denotes the number of correctly identified non-curb images divided by all non-curb images, which can be written as

$$\text{Specificity} = n_{\mathrm{tn}}/(n_{\mathrm{fp}} + n_{\mathrm{tn}}). \tag{3}$$

Precision is the number of correctly identified images divided by the number of images identified as positives. In this case, it denotes the number of correctly identified curb images divided by the total images identified as curb images, which can be written as

$$\text{Precision} = n_{\mathrm{tp}}/(n_{\mathrm{tp}} + n_{\mathrm{fp}}). \tag{4}$$

F1-score is the harmonic mean of the precision and sensitivity, which can be written as:

$$\text{F1-score} = \frac{2 \times (\text{Sensitivity} \times \text{Precision})}{(\text{Sensitivity} + \text{Precision})} \tag{5}$$

AUC denotes the area under the corresponding receiver operating characteristic (ROC) curves.

7.2 Evaluation Metrics for Detection

Precision is calculated following Eq. 4. Recall is calculated as

$$\text{Recall} = n_{\mathrm{tp}}/(n_{\mathrm{tp}} + n_{\mathrm{fn}}). \tag{6}$$

The average precision (AP) is defined by:

$$\text{AP} = \int_0^1 f(r)dr, \tag{7}$$

where r denotes the variable of recall, and $f(r)$ denotes the PR curve.

Mengbo You

is an assistant professor in the Faculty of Science & Engineering, Iwate University. He received the B.S. degree in 2012 from the Computer Science Department, Northwest A&F University, the M.S. degree in 2015 and the Dr.Eng. in 2018, both from Iwate University, Japan. His research interests include machine learning, image processing, object detection and deep learning. He is a member of the Society for Art and Science, and the Institute of Image Electronics Engineers of Japan.

Fumito Chiba

received the Dr.Eng. degree from the Department of Electronic Engineering and Computer Science, Graduate School of Engineering, Iwate University. He worked as a research associate in the Department of Computer Science, Faculty of Engineering, Iwate University, and now he is the Managing Director of LANG CO., LTD. He is engaged in research and development of 3D shape measurement and processing of archaeological artifacts. He is a member of Japan Association for Archaeoinformatics.

Kouichi Konno

is a professor of Faculty of Science and Engineering at Iwate University. He received a BS in Information Science in 1985 from the University of Tsukuba. He earned his Dr.Eng. in precision machinery engineering from the University of Tokyo in 1996. He joined the solid modeling project at RICOH from 1985 to 1999, and the XVL project at Lattice Technology in 2000. He worked on an associate professor of Faculty of Engineering at Iwate University from 2001 to 2009. He has written a book *Introduction to 3D shape processing*. His research interests include 3D modeling, 3D surface data compression, archaeological relics restoration. He is a member of The Society for Art and Science, The Institute of Image Information and Television Engineers , Japan Association for Archaeoinformatics, Information Processing Society of Japan, and EuroGraphics.