Three-dimensional Quantitative Analysis on Granular Particle Shape using Convolutional Neural Network

Pin ZHANG[1], Zhen-Yu YIN[2], Yin-Fu JIN[3]

Abstract: To identify all desired shape parameters of granular particles with less computational cost, this study proposes a three-dimensional convolutional neural network (3D-CNN) based model. Datasets are made of 100 ballast and 100 Fujian sand particles, and the shape parameters (i.e. aspect ratio, roundness, sphericity and convexity) obtained by conventional methods are used to label all particles. For the model training, by feeding the slice images of particles into the model, the contour of particles is automatically extracted, thereby the shape parameters representing particle shape can be learned by the model. Thereafter, the model is applied to predict shape parameters of new particles as model testing. All results indicate the model trained based on slice images cut from three orthogonal planes presents the highest prediction accuracy with an error of less than 10%. Meanwhile, the accuracy for concave and angular particles can be guaranteed. The rotation-equivariant of the model is confirmed, in which the predicted values of shape parameters are roughly independent of changeable rotations of the particle when cutting slice images. Superior to conventional methods, all desirable shape parameters can be obtained by one unified 3D-CNN model and its prediction is independent of particle complexity and the number of triangular facets, thus saving computation cost.

Keywords: Microscopy; gravels; particle-scale behaviour; grain shape; X-ray computed tomography

[1] Ph. D candidate, Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China, ORCID number: https://orcid.org/0000-0002-9004-647X

[2] Associate Professor, Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China, ORCID number: https://orcid.org/0000-0003-4154-7304

Corresponding author, E-mail: zhenyu.yin@polyu.edu.hk; Tel: +852 3400 8470; Fax: +852 2334 6389

[3] Postdoctoral fellow, Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China, ORCID number: https://orcid.org/0000-0003-0522-1702

1. Introduction

Numerous experiments have been conducted to investigate the effects of particle shape on the mechanical behaviour of granular material, such as shear strength,[1] shear stiffness,[2] and compactness.[3] Besides, numerical modelling methods, particularly for the discrete element method (DEM),[4] have also been extensively employed to explore microscopic and macroscopic behaviours of particles with different shapes.[5-8] No matter for experimental or numerical research works, quantitative analysis of particle shape is a necessity for understanding the mechanical responses to their shape.

A series of indices have been proposed for quantitatively describing 2D or 3D particle shapes.[9-12] 2D particle shape tends to be analysed based on its projections, thereby the calculated shape parameters rely heavily on the orientations of particles and tend to be inaccurate.[13,14] 3D shape parameters are more representative and accurate descriptors. The currently used methods for reconstructing and calculating 3D shape parameters can be categorized into three groups.[15] The first one is achieved by voxelization of a particle, in which the entire body of a particle is assembled by voxels, thereafter particle size and morphology are calculated based on the total number and spatial distribution of voxels,[16] respectively. The second way is developed based on the reconstruction of particle surface using triangular mesh. The Marching Cube algorithm is a commonly used method to detect boundary voxels and build surface inside these boundary voxels.[17] Spherical harmonic (SH) analysis is the third method to theoretically reconstruct the particle surface.[18] However, the boundary of particles cannot be clearly distinguished using the first method, thereby it generally integrates with error analysis

to mitigate such deficiency.[16] The reconstruction of 3D shape is a necessary step in the second and third methods, but such a process also involves certain approximation calculations, leading to the overlap of error generated during the scanning of particles. The calculation of each shape parameter relies on an independent formulation among these methods. Meanwhile, the calculation process also requires domain-based empirical knowledge to set parameters and the computational cost depends on the number of triangular facets, which is not concise and user-friendly.

Recently, a deep learning algorithm convolutional neural network (CNN) provides a state-of-the-art method to extract the features hidden in images such as micro-structural characteristical of the studied object[19,20] and the identified features can be further leveraged for the analysis of macro-scale responses.[21,22] In 3D-CNN, a series of 2D slices images of the studied object can be extracted by X-ray computed tomography or photography, etc.,[23-25] and then they are aligned along the depth dimension as the input of 3D-CNN. Inspired by the characteristics of 3D-CNN, 2D slice images cut from the particle can be directly fed into 3D-CNN, and the outputs can be set as the 3D particle shape parameters. All desired shape parameters can be simultaneously obtained with one unified model and the computation process is independent of the number of triangular facets. Meanwhile, myriad problems are still open for the application of 3D-CNN to identify shape parameters, such as the selection of slice images, the effect of particle orientation on the prediction (rotation equivariant) and the dependency of prediction accuracy on the particle shape.

Hence, this study aims to employ 3D-CNN to identify slice images of real granular

particles and to predict the 3D shape parameters of particles. Slice images of 100 ballast and 100 Fujian sand particles are extracted by using a 3D laser scanner and their 3D shape parameters including aspect ratio, roundness, sphericity and convexity are pre-calculated for labelling each particle. Slice images collected from various particle orientations are extracted to train different models and their performance are comprehensively compared for devising model development strategy and model selection. The relation between the predicted output and the particle orientation is presented for revealing the robustness of the model. The prediction accuracy against the particle shape such as angularity and concavity is investigated for investigating the applicability of the model. Finally, the performance of the developed model on randomly synthetic particles is discussed for understanding its generalization ability.

2. Applied methods for model development

2.1 Extraction of particle shape parameters

3D X-ray micro-computed tomography and 3D laser scanner are two commonly used methods to obtain the 3D shape of particles. Thereafter image processing methods are performed to reconstruct particle surfaces such as using numerous triangular meshes, in which the number of facets, vertices, coordinate of vertices, unit normal vectors of facets and the pair relationships between facets and vertices are recorded. Particle shape parameters can be calculated based on such information. Herein, four representative parameters: aspect ratio (AR), roundness (R), sphericity (S) and convexity (C_X) for describing a particle are studied.

The form of a particle is generally represented by the longest (l), intermediate (i) and

shortest (s) axes.[26] Principle component analysis is a commonly used method to identify the orientations of three axes.[27] Thereafter the particle is rotated to guarantee its principal axes of inertia parallel to the Cartesian axes, and the values of l, i and s can be calculated based on voxel coordinates ($l \geq i \geq s$, **Fig. 1a**). Then the AR can be calculated using:

$$AR = \left(i/l + s/i\right)/2 \tag{1}$$

For the particle roundness, Dong and Wang[28] proposed a method to determine the maximum (k_1) and minimum (k_2) normal curvature of a surface. Herein, the surface is formed by neighbouring triangles around one or several points (**Fig. 1b**). After the maximum and minimum normal curvatures of a surface being determined, the roundness of a particle can be calculated using the method proposed by Wadell[29], as follows:

$$R = \sum_{i=1}^{N} \left(r_i / R_{is}\right) \Big/ N \tag{2}$$

where r_i and R_{is} are the radius of curvature of a corner and the maximum inscribed sphere; N is the total number of corners. Herein, r_i can be represented by k_1 or k_2, and k_1 is used in this study. Meanwhile, the local curvature value larger than its inscribed sphere is treated as a part of a "corner".[15]

Sphericity is another important index of particle shape to quantify the compactness of granular materials. The sphericity index proposed by Wadell[29] is used in this study, which is defined as the ratio of the surface area of a sphere, which has the same volume as the studied particle, to the surface area of the studied particle (**Fig. 1b**):

$$S = \sqrt[3]{36\pi V^2} \Big/ SA \tag{3}$$

where V and SA are the volume and surface area of the studied particle, respectively.

The convexity index is defined as the ratio of the convex hull of the particle to its volume, in which the convex hull is the minimal convex surface enclosing the particle.

$$C_X = V/V_{CH} \tag{4}$$

where V_{CH} is the volume of the convex hull.

2.2 Convolutional neural network

CNN generally consists of convolutional, pooling and fully connected layers, as shown in **Fig. 2**. The kernel is a key component of CNN, which is represented by a three-dimensional matrix for 3D-CNN and its function is similar to a filter to extract information. Given a raw image with the size of h^r (height)×w^r (width)×d^r (depth)×c^r (channel, 1 and 3 for greyscale and color images, respectively), the kernel is first employed to directly compute the dot product between the entries of the kernel and the matrix of the raw image. Such a process is generally known as feature extraction, i.e., aiming to relate the pixel distribution to the desired outputs such as shape parameters in this study. The corresponding output is known as the feature map. The training of CNN is to adjust the kernel matrix for optimizing feature extraction until the predictions converging to measured values. Given that each kernel moves with a fixed stride s_t, the size of the feature map (i.e. height, width and depth) is rescaled to h^c×w^c×d^c×c^c,[30] which can be obtained by:

$$S_j^c = \left\lceil \frac{S_j^r - f + 1}{s_t} \right\rceil, j = h, w, d \tag{5}$$

where S_j^r and S_j^c denote the size along the ith dimension of images in the input and

convolutional layers, respectively; f is the size of the kernel; $\lceil\ \rceil$ is the ceiling function, which maps x to the least integer great than or equal to x.

The pooling layer aims to merge similar features in the feature map. The size of the feature map can be further reduced to $h^p \times w^p \times d^p \times c^p$ using Equation (5). The max-pooling layer is used in this study, in which the maximum value in a region of the feature map represented by a kernel is retrained. Before entering the fully connected layer, all feature maps are required to be flattened and connected in turn. The rear section of CNN is a multi-layer perceptron. The introduction of CNN and the illustration of the computation process can be found in Zhang et al[31].

3. Modelling process using 3D-CNN

3.1 Framework of modelling strategy

A primary characteristic of the supervised learning algorithm is that the outputs are required to be known so that the relationships between images of particles and shape parameters can be learned. Therefore, a database with the true shape of particles (obtained by laser scanner in this study) and corresponding shape parameters is first built. After the model is developed, the images of arbitrary particles can be fed to the model and shape parameters can be directly predicted.

Fig. 3 presents the schematic view of developing a 3D-CNN based model to identify 3D shape parameters of particles. Based on the real 3D particle, an algorithm is developed to automatically extract random cross-sections of a 3D particle, i.e., the slice image. First, the

coordinates of the real particle are rescaled by its longest axis. The rescaled particle is imported and put in the same cubic box with the size of 1 The centroid of the particle (o) is located at the centre of the box. Herein, the cubic box is used so that the slice images of particles retain the same square, which is useful for further image processing. Another noteworthy point is that the dimensionless particle shape parameters are not affected by the size of the box. Thereafter, the cutting plane is located by setting coordinates, based on which the slice image can be obtained accordingly. Each slice image is assigned 3 RGB channels for clearly exhibiting details of particles by colour, thus further improving the potential of using 3D-CNN to identify particle shape. Slice images are extracted along three orthogonal orientations. From the boundary, 40 equally-spaced slices are extracted along each orientation. There is no general method to judge how many slices should be extracted. After examining with different numbers, 40 slices are found sufficiently dense, thereby ensures the entire shape contour of particles to be retained. Therefore, a total of 24000 slice images along three dimensions are ultimately extracted from all particles.

These slice images form the input $\mathbf{x}$ of the 3D-CNN. This $\mathbf{x}$ has five dimensions, i.e., the number of particles, width, height, depth and channels of slice images. After passing through the convolutional (**C**, with a total of l), pooling (**P**, with a total of m) and fully-connected (**F**, with a total of n) layers, outputs $\boldsymbol{y}$, i.e. AR, R, S and C_X, can be predicted simultaneously. Such a process involves a step-by-step computation of matrix $\mathbf{x}\rightarrow\mathbf{C}^1\rightarrow\ldots\rightarrow\mathbf{C}^l\rightarrow\mathbf{P}^1\ldots\rightarrow\mathbf{P}^m\rightarrow\mathbf{F}^1\rightarrow\ldots\rightarrow\mathbf{F}^n\rightarrow\boldsymbol{y}$. The sequence and number of convolutional, pooling and fully-connected layers are related to the ultimate framework of the 3D-CNN based

model. The mathematic expression of such a process is shown as follows:

(1) Computation from the input x to the first convolutional C^1 layers:

$$\mathbf{C}_j^1 = \mathrm{ELU}\left(\sum_{i=1}^{f^{C^1}} \mathbf{W}^{C_j^1}\mathbf{x}_i + b_j\right) \tag{6}$$

where $\mathbf{C}_j^1$ denotes the outputs convolved by the jth kernel in the C^1 layer; $\mathbf{W}^{C_j^1}$ is the weight of the jth kernel in the C^1 layer; f^{C^1} is the size of the kernel in the C^1 layer; b_j is a bias term at the jth kernel for shifting the transfer function curve along the input axis, allowing more flexible input-output mapping;[32] ELU represents the activation function exponential linear unit (ELU).

(2) Computation from the first C^1 to the second convolutional C^2 layers:

$$\mathbf{C}_j^2 = \mathrm{ELU}\left(\sum_{i=1}^{f^{C^2}} \mathbf{W}^{C_j^2}\mathbf{C}_i^1 + b_j\right) \tag{7}$$

(3) Computation from the lth convolutional C^l to the first pooling P^1 layers:

$$\mathbf{P}_{i,j}^1 = \max\left(\mathbf{C}_{i,j}^l, \mathrm{K}, \mathbf{C}_{i,j+f^{P^1}-1}^l, \ldots, \mathbf{C}_{i+f^{P^1}-1,j}^l, \mathrm{K}, \mathbf{C}_{i+f^{P^1}-1,j+f^{P^1}-1}^l\right) \tag{8}$$

where max function indicates the output in the P^1 layer is the maximum value of a region covered by a kernel with the size of f^{P^1}.

(4) Computation from the first P^1 to the second pooling P^2 layers:

$$\mathbf{P}_{i,j}^2 = \max\left(\mathbf{P}_{i,j}^1, \mathrm{K}, \mathbf{P}_{i,j+f^{P^2}-1}^1, \ldots, \mathbf{P}_{i+f^{P^2}-1,j}^1, \mathrm{K}, \mathbf{P}_{i+f^{P^2}-1,j+f^{P^2}-1}^1\right) \tag{9}$$

(5) Computation from the mth pooling P^m to the first fully-connected F^1 layers:

$$\mathbf{F}^1 = \left(\mathbf{P}_1^m, \mathrm{K}, \mathbf{P}_2^m, \ldots, \mathbf{P}_{n^{P^m}}^m\right) \tag{10}$$

where n^{P^m} denotes a total of elements in the P^m layer. This procedure does not involve mathematical computation. It aims to extract all elements in the feature maps and align them with a two-dimensional matrix so that they can enter the fully-connected layer.

(6) Computation from the first F^1 to the second fully-connected F^2 layers:

$$\mathbf{F}^2 = \mathrm{ELU}\left(\mathbf{W}^{F^2}\mathbf{F}^1 + \boldsymbol{b}^{F^2}\right) \tag{11}$$

where $\mathbf{W}^{F^2}$ and $\boldsymbol{b}^{F^2}$ are the weights and biases connected with F^1 and F^2 layers, respectively.

(7) Computation from the *n*th fully-connected F^n to the output *y* layers:

$$\boldsymbol{y} = \mathbf{W}^{y}\mathbf{F}^{n} + \boldsymbol{b}^{y} \tag{12}$$

where $\mathbf{W}^{y}$ and $\boldsymbol{b}^{y}$ are the weights and biases connected with F^n and *y* layers, respectively.

A noteworthy point is that slice images vary with the cutting orientations, but the performance of the CNN based model is dependent on the input images. To investigate the influence of slice cutting-plane direction on the model performance, a total of 7 models are thus trained based on slice images cut from *x*, *y*, *z*, *xy*, *xz*, *yz* and *xyz* directions, respectively. Herein, *x* denotes all images are collected from cut planes perpendicular to the *x*-axis, *xy* denotes images of cutting-plane perpendicular to *x* and *y* axes, *xyz* denotes images of cutting-plane perpendicular to *x*, *y* and *z* axes. The meaning of other notations is the same. Hereafter such 7 models are labelled with CNN-*x*, CNN-*y*, CNN-*z*, CNN-*xy*, CNN-*xz*, CNN-*yz* and CNN-*xyz* for convenient demonstration. The number of slice images for each cutting-plane direction is the same, thus the depth of input for the first to third models is half of the fourth to sixth models, and one-third of the last model. Such an investigation can understand the influence of slice orientations on the performance of the 3D-CNN based model, and the optimum number of slice orientations can also be determined.

3.2 Data source

The data source is collected from 100 ballast and 100 Fujian particles due to their rich

morphologies. Their 3D shape is obtained by using a laser scanner (Handy Scan 700 TM). The principle of the laser scanner is measuring the distance from the laser light to the studied object in different directions. The point cloud of a particle is first obtained during scanning, thereafter adjacent points are connected to form a triangular vertex. The final surface of the particle is reconstituted by numerous triangular facets (**Fig. 1b**). Previous research works confirm that a reconstituted particle with over 1500 triangular facets is roughly identical to its real morphology,[15,33] thereby all ballast and Fujian sand particles used in this study are reconstituted using more than 1500 triangular facets. Nie et al[33] have presented the details of the used laser scanner and the process of reconstructing particle surface. The shape parameters of these particles are computed in advance using the conventional methods as mentioned before for labelling each particle. **Fig. 4** illustrates the distribution of all data points and the statistical values are summarized in **Table 1**. It can be seen that the shape characteristics of ballast and Fujian sand can be distinguished clearly. Fujian sand is globally rounder than ballast but is more angular. Overall, particle shape is abundant in the database to develop a general model.

3.3 Data pre-processing

CNN-based model tends to suffer from erratic fluctuations in predictions when the input images rotate.[34] To enhance rotation equivariance and enlarge the database, the flip data augmentation method is adopted,[31,35] through which each image is flipped horizontally and vertically, thus the database size triples. Finally, a database with 72000 slice images is used in this study. Gholamy et al[36] have verified that training and testing sets with 80% and 20% of the data, respectively, are the best division to ensure the desirable prediction accuracy and avoid over-

estimation simultaneously. Thus, 80% of particles are randomly selected to train the 3D-CNN based model and the remaining 20% are used to test the model.

The raw slice image are 512×512 pixels. Considering a larger amount of slices images are used in this study, using raw slice images would impose a large burden on computational cost, but the too-small size of images may omit the local characteristics of particles. To achieve the trade-off between computational cost and accuracy, the size of slice images are reduced to 64×64 pixels by trial and error. This is a commonly used technique in the CNN domain as long as the contour of the studied object in the compressed images is retained.[25] Besides, RGB values of a raw slice image range from 0 to 255. Large RGB values are sensitive to the variation of weights and biases, leading to the difficulty in the convergence of a training process and stability of the model. Therefore, RGB values are rescaled to the range (0, 1) by multiplying 1/255 factor before the training of the model.[37]

4. Analysis of 3D-CNN based model

4.1 Training of 3D-CNN based model

The primary objective of training the 3D-CNN based model is to determine its topology. In addition, other configurations such as initial weights, biases, optimizer, learning rate, activation function, batch size and epochs are required to be finely tuned. Moreover, some configurations such as the activation function also has their intrinsic parameters that need to be finely tuned. Such factors indicate the determination of all parameters of a 3D-CNN based model is a tremendous task. An automatic search of optimum configurations of 3D-CNN such as using

meta-heuristic algorithms would cause a computational disaster, thereby the most practical and commonly used method is trial and error combined with knowledge of domain experts. To this end, the configurations of the 3D-CNN based model in this study are determined manually. Weights and biases of the model are updated using a stochastic gradient descent algorithm with the learning rate of $1e^{-4}$, the decay of $1e^{-6}$ and the momentum of 0.9. The configurations of seven 3D-CNN based models maintain consistency. The final configurations are summarized in **Table 2**. Inspired by the VGG model,[38] the kernel sizes used in the convolutional and pooling layers are 3×3 and 2×2, respectively. Such a combination can improve the extraction of image features. The detailed process for determining such configurations are not presented for brevity, and each configuration has been explained by Chollet[39].

The prediction of particle shape parameters is a regression issue, thereby mean square error (MSE) is defined as the loss function. Meanwhile, the *k*-fold cross-validation method is employed. Such a method further divides the original training set into 10 subsets, in which 9 random subsets form a new sub-training set to be used to train the model, and the remaining one as the validation set is used to test the model. Each subset has the same probability to be selected as the sub-training and validation sets. Therefore, 10 sub-models are developed based on 9 random subsets at each training epoch, which can detect the overfitting issue and reduce the effects of the split of training and testing sets on the model performance.[40] To avoid the overfitting issue, an L_2 regularization term is added to constrain the weights $\mathbf{W}$ and biases $\boldsymbol{b}$ values of the 3D-CNN based model. The loss function ($\mathcal{L}$) is ultimately formulated by:

$$\mathcal{L}(\mathbf{W},\boldsymbol{b}) = \frac{1}{k}\frac{1}{n_{sub}}\sum_{1}^{k}\sum_{i=1}^{n_{sub}}\left(y_i^p - y_i^a\right)^2 + \frac{\lambda}{2}\boldsymbol{w}^T\,\boldsymbol{w} \tag{13}$$

where n_{sub} is the number of datasets in a sub validation set; k is the number of folds, which is usually set as 10; y_i^p and y_i^a are predicted and actual shape parameters, respectively. $\boldsymbol{w}$ is vector consists of all elements in the weights and biases.

Fig. 5 presents the evolution of loss values on both sub-training and validation sets generated by 7 models individually during the training process. MAE values on the sub-training set rapidly reduce at the early 1000 epochs, then roughly converges at 0.001. Compared with the loss values generated by CNN-*x*, CNN-*y*, CNN-*z*, CNN-*xy*, CNN-*xz* and CNN-*yz*, there is no large difference of MAE values generated by the models trained based on slices images collected from one or two directions. From the perspective of CNN-*xyz*, the model trained by slices images collected from three directions present an obvious decrease of MAE values with a more rapid convergence rate. Regarding the validation set, MAE values generated by all models on the validation set roughly maintains unchanged as the epoch exceeds 1000. The MAE values do not show increase trend with the increasing epoch, which indicate these 7 models are well trained without overfitting issue. The difference in the validation set among 7 models can be distinguished. It seems that slices images collected from *y*-direction are less important than the remaining two directions, because the error of models trained with images of this direction are larger than other same type models, e.g., error of CNN-*y* is larger than CNN-*x* and CNN-*z*, errors of CNN-*xy* and CNN-*yz* are larger than CNN-*xz*. The side effect of the *y*-direction is mitigated by images from *x* and *z* directions in the model CNN-*xyz*. CNN-*xyz* outperforms the remaining models on both sub-training and validation in terms of prediction accuracy and convergence rate.

Revealing image transformation inside the model is a necessity of understanding the identification mechanism of particle shape in the 3D-CNN based model. The first feature map in each layer is illustrated, and only the results generated by model CNN-*xyz* are presented as a representative example, as shown in **Fig. 6**. Feature maps in the low layers (Con3D_1) and (Con3D_2) are close to the shape of the raw particle. The deeper layer (MaxPooling3D) aims to fuse features, thereby the illustration is abstract. It can be seen from **Fig. 6** that the boundary of the particle is accurately identified and marked with different colours, which indicates 3D-CNN is capable of capturing particle shape information.

4.2 Comparison and selection of the model

After the optimum configuration of the CNN based model is determined, the training and testing sets are fed to the model to evaluate its prediction performance. Quantitative evaluation indicators mean absolute error (MAE) and mean absolute percentage error (MAPE), are calculated.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^p - y_i^a\right| \tag{14}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i^p - y_i^a}{y_i^a}\right| \times 100\% \tag{15}$$

where n is the total number of datasets.

Fig. 7 presents MAE and MAPE values on the training and testing sets generated by 7 models and the detailed values are summarized in **Table 3** and **Table 4**. The prediction error on the testing set is about 6-7 times the error on the training set. The MAE values are related to the magnitude of the shape parameter values, thus MAE values for predicting AR and C_X are

larger than S and R. The magnitude of R is the smallest, thus MAPE values on both training and testing sets are the largest. This indicates that the prediction of R using 3D-CNN is more difficult than the remaining three shape parameters, which complies with the real condition, i.e., the calculation of R is much more complex than the others.

To comprehensively compare the performance of 7 individual models, MAE and MAPE values for all particles in the testing set generated by all models are illustrated in **Fig. 8**. Prediction errors of CNN-*x*, CNN-*y* and CNN-*z* show a large variation, indicating that the performance of the 3D-CNN based model is affected by slice orientations as expected. The performance of the model stabilizes and improves with the increasing slice orientations particularly when the slice images from three orientations are used to train the model. Such a phenomenon is consistent with the convergence values generated during the training process (**Fig. 5**). Note that the mean MAE and MAPE values show a slight increase when the training slice images are expanded from one orientation to two orientations. Such a difference can be neglected in the context of small prediction errors and it should be highlighted that the stability of the model performance is enhanced with a smaller variation of MAE and MAPE values. Hereafter, the model CNN-*xyz* trained using the slice images from three orientations is used for further analysis considering its optimum performance.

4.3 Robustness of 3D-CNN based model

4.3.1 Generalization ability

The predicted values of AR, S, R and C_X on both training and testing sets using the optimum model CNN-*xyz* are presented in **Fig. 9**, compared with the measured values. All data points

are close to the line with the slope of 1. In **Table 4**, it can be seen that the individual prediction errors of AR, S, R and C_X on the testing set are well controlled within 10%, 4%, 10% and 7%, respectively. Such factors indicate one unified 3D-CNN based model is capable of directly predicting all desirable particle shape parameters and the prediction accuracy can be guaranteed. Note that all shape parameters can be obtained within several seconds once the model is trained. Such a process is completely independent of the number of triangular facets because the model makes a prediction solely based on the slice images as inputs. From this perspective, 3D-CNN based model outperforms conventional methods, in which the computational cost for obtaining shape parameters relies heavily on the number of triangular facets.

Considering R and C_X can represent the complexity of a particle shape, the prediction errors on these two indices are thus presented in detail for further analysing the model performance. Herein, the values of shape parameters are categorized into several groups with an interval of 0.1 and the prediction errors on the testing set in each group are also summarized. As shown in **Fig. 10**, the prediction errors increase with the decreasing values of R and C_X. It reveals that the 3D-CNN based model presents the same characteristic as the conventional methods, i.e., the difficulties for capturing particle shape increase with the increasing angularity and concavity. Meanwhile, prediction errors seem to be more sensitive to the C_X value, which may be contributed to the paucity of data with the small C_X value (see Fig. 11) causing poor model performance at that range.

4.3.1 Rotation equivariant

Particle shape parameters are independent of their orientations. The slice images are changed

as the particle rotates, which may vary the predicted shape parameters using the 3D-CNN based model. Therefore, the predicted shape parameters are required to hold steadily with the changeable orientations of a given particle. This is a significant characteristic to confirm the applicability of a 3D-CNN based model, whereby a rotation equivariant (R_E) index is proposed to quantify its robustness, which is defined as:

$$R_E = \frac{y^p_{j,\max} - y^p_{j,\min}}{\sum_{k=1}^{m} y^p_{j,k}} \times 100\%,\ j = AR, S, R, C_X \tag{16}$$

where m denotes the total number of orientations; $y^p_{j,k}$ denotes the predicted jth shape parameter at the kth orientation; $y^p_{j,\max}$ and $y^p_{j,\min}$ corresponds to the maximum and minimum values. The small value of R_E indicates a 3D-CNN based model exhibits excellent robustness.

The orientation of a particle is represented by azimuth θ, which ranges from 0 to 2π, as shown in **Fig. 11**. To evaluate the stability of the optimum model CNN-*xyz*, oritentations of particles in the testing set are changed. Considering the slice images of a particle with the azimuth of θ and θ+π are same, the orientation of particles thus varies between 0 to π and four azimuths, i.e., π/4, π/2π, 3π/4 and π are selected as the representative cases. Slice images cut from particles with such four orientations are fed to the model CNN-*xyz*. Two particles corresponding to maximum and minimum values R_E are labelled with A and B, respectively. The predicted shape parameters of particles *A* and *B* with the changing azimuth are presented in **Fig. 12**, and the detailed values are summarized in **Table 5**. It can be seen that the predicted values of shape parameters show a slight difference at different azimuths as expected, and there is no evidence indicates the predicted values are larger or smaller at a certain azimuth. The

maximum discrepancy with the R_E of 10% is observed for R and the corresponding mean value of R_E is also the largest (9.092%). Thus, R is more sensitive to the orientation of the particle in comparison with the remaining three shape parameters with the R_E around 5%. This result is reasonable because the computation of R is more complex involving the identification of local features. Overall, the performance of the model is roughly independent to the orientation of particles.

5. Discussion

To further examine the generalization ability of the proposed model, 15 elliptical particles with various s, i and l are created (**Fig. 13**) and their shape parameters are directly predicted by the 3D-CNN based model. The predicted AR, R, S and C_X, are plotted together against the exact values. The mean MAE and MAPE values for four shape parameters are also computed. The predictions agree well with the actual values, thus further indicating the excellent generalization ability of the model.

The real shape of particles in the database is known and they are labelled with their shape parameters. For instance, numerous research works investigated the effects of particle shape on mechanical behaviour, in which particles with target shape parameters were first devised. Thus, the particle in the database with shape parameters closest to the target values can be selected to represent the real particle. Thus, this database is also an alternative method to roughly provide particles with expected shape parameters. The difference between the target particle and the selected particle in the database is quantified by the average MAE values of

AR, *S*, *R* and C_X. The particle with the minimum difference is finally selected to represent the target particle. The mathematical expression of such a process can be formulated by:

$$\arg\min_{n_p} \ \frac{1}{4}\sum_{j}\left|y_j^t - y_{j,n_p}^p\right|,\ j = AR, S, R, C_X \tag{17}$$

where n_p denotes a selected particle in the database; y_j^t and y_{j,n_p}^p are the value of the *j*th shape parameter for the target and selected particles, respectively.

This formulation aims to search the particles in the training set whose difference to the given values of shape parameters is the lowest. Two representative particles A and B (**Fig. 12**) are used to verify the feasibility of such a method. Using shape parameters of A and B, two particles in the training set are identified. For comparison, the real shape of particles A and B are reconstructed with a 3D laser scanner. **Fig. 14** indicates the identified 3D shape of particles is close to the real shape, thereby this rough estimation method of 3D particle shape is feasible to a certain extent. Note that this study provides an effective approach, based on which more parameters of particles can be easily introduced and more accurate identification can thus be achieved.

In essence, a data-driven model developed based on 3D-CNN primarily relies on the database. The number of particle samples and types used in this study is limited, and the ranges of shape parameters cannot involve all types of granular particles, which also limits application scopes of the developed model. Note that with this effective approach it is easy to feed more numbers and types of particles, based on which the model is feasible to predict 3D shape parameters of arbitrary particles given. The configurations of the 3D-CNN based model are flexible and it can be adjusted with simple operation to be adapted with more images of

particles.

6. Conclusions

Using a series of 2D cross-sectional images of the particle, its 3D shape parameters can be obtained by the deep learning algorithm 3D-CNN. This study has achieved such a process, and developed a 3D-CNN based model applied to real particles. The datasets for model training and testing have been created with a total of 100 ballast and 100 Fujian sand particles labelled with their representative shape parameters, i.e. aspect ratio, roundness, sphericity and convexity.

By feeding the slice images of these particles to the model, the 3D-CNN based model extracted the contour of particles, thereby the relationships between particle shape and shape parameters were learned for model training. In this study, a total of 7 models were trained using 7 combinations of slice images cut from *x*, *y*, *z*, *xy*, *xz*, *yz* and *xyz* directions, respectively. Through model testing it has been examined that, the 3D-CNN based model trained based on slice images from three orthogonal planes (i.e. *xyz*) can well predict all four shape parameters simultaneously. The prediction process can be completed within a second and is independent of the triangular facets. The prediction accuracy is high an error of less than 10% and the accuracy can also be ensured for the complex particles with concavity and angularity. The model has also been examined for different rotational degrees of particles, which shows the predicted results roughly maintain consistency whatever the rotational degree is. Therefore, the robustness and applicability of the proposed model are guaranteed.

Moreover, the developed model is generic, i.e., the outputs in the model can be represented by a random number of desirable shape parameters, the database can be further expanded to improve the application scopes of the model, and for given shape parameters the model can also find possible particles with closest values of parameters from the database. 3D-CNN is a novel promising and effective method of predicting shape parameters, reconstructing the surface and implementing discrete-element modelling.

Acknowledgements

This research was financially supported by the Research Grants Council (RGC) of Hong Kong Special Administrative Region Government (HKSARG) of China (Grant No.: 15220221). Authors sincerely thank Dr. Gong for providing images of ballast particles.

References

1. Sezer A, Altun S, Göktepe BA. Relationships between shape characteristics and shear strength of sands. *Soils Found.* 2011;51(5):857–871.
2. Cho GC, Dodds J, Santamarina JC. Particle shape effects on packing density, stiffness, and strength: natural and crushed sands. *J Geotech Geoenviron Eng.* 2006;132(5):591-602.
3. Altuhafi FN, Coop MR, Georgiannou VN. Effect of particle shape on the mechanical behavior of natural sands. *J Geotech Geoenviron Eng.* 2016;142(12):04016071.
4. Cundall PA, Strack ODL. A discrete numerical model for granular assemblies. *Géotechnique.* 1979;29(1):47-65.
5. Ueda T, Matsushima T, Yamada Y. DEM simulation on the one-dimensional compression behavior of various shaped crushable granular materials. *Granul Matter.* 2013;15(5):675-684.
6. Nie Z, Fang C, Gong J, Liang Z. DEM study on the effect of roundness on the shear behaviour of granular materials. *Comput Geotech.* 2020;121:103457.
7. Zhao S, Evans TM, Zhou X. Effects of curvature-related DEM contact model on the macro- and micro-mechanical behaviours of granular soils. *Géotechnique.* 2018;68(12):1085-1098.
8. Lim K-W, Andrade JE. Granular element method for three-dimensional discrete element calculations. *Int J Numer Anal Methods Geomech.* 2014;38(2):167-188.

9. Zheng J, Hryciw RD. Traditional soil particle sphericity, roundness and surface roughness by computational geometry. *Géotechnique* 2015;65(6):494-506.
10. Su D, Yan WM. Prediction of 3D size and shape descriptors of irregular granular particles from projected 2D images. *Acta Geotech.* 2019;15(6):1533-1555.
11. Altuhafi F, O'Sullivan C, Cavarretta I. Analysis of an image-based method to quantify the size and shape of sand particles. *J Geotech Geoenviron Eng.* 2013;139(8):1290-1307.
12. Zheng J, Hryciw RD. Roundness and sphericity of soil Particles in assemblies by computational geometry. *J Comput Civil Eng.* 2016;30(6):04016021.
13. Hryciw RD, Zheng J, Shetler K. Particle roundness and sphericity from images of assemblies by chart estimates and computer methods. *J Geotech Geoenviron Eng.* 2016;142(9):04016038.
14. Zhou B, Wang J, Wang H. Three-dimensional sphericity, roundness and fractal dimension of sand particles. *Géotechnique.* 2018;68(1):18-30.
15. Zhao B, Wang J. 3D quantitative shape analysis on form, roundness, and compactness with μCT. *Powder Technol.* 2016;291:262-275.
16. Alshibli KA, Druckrey AM, Al-Raoush RI, Weiskittel T, Lavrik NV. Quantifying morphology of sands using 3D imaging. *J Mater Civil Eng.* 2015;27(10):04014275.
17. Lorensen W, Cline H. Marching cubes: a high resolution 3D surface construction algorithm. *Comput Graph.* 1987;21(4):163-169.
18. Garboczi EJ. Three-dimensional mathematical analysis of particle shape using X-ray tomography and spherical harmonics: Application to aggregates used in concrete. *Cem Concr Res.* 2002;32:1621-1638.
19. Xu M, Papageorgiou DP, Abidi SZ, Dao M, Zhao H, Karniadakis GE. A deep convolutional neural network for classification of red blood cells in sickle cell anemia. *PLoS Comput Bio.* 2017;13(10):e1005746.
20. Singh VK, Rashwan HA, Romani S, et al. Breast tumor segmentation and shape classification in mammograms using generative adversarial and convolutional neural network. *Expert Syst Appl.* 2020;139:112855.
21. Zhang P, Yin Z-Y. A novel deep learning-based modelling strategy from image of particles to mechanical properties for granular materials with CNN and BiLSTM. *Comput Meth Appl Mech Eng.* 2021;382:113858.
22. Rao C, Liu Y. Three-dimensional convolutional neural network (3D-CNN) for heterogeneous material homogenization. *Comput Mater Sci.* 2020;184.
23. Milletari F, Ahmadi S-A, Kroll C, et al. Hough-CNN: Deep learning for segmentation of deep brain regions in MRI and ultrasound. *Comput Vis Image Und.* 2017;164:92-102.
24. Li L, Zhao X, Lu W, Tan S. Deep learning for variational multimodality tumor segmentation in PET/CT. *Neurocomputing.* 2020;392:277-295.
25. Ouyang D, He B, Ghorbani A, et al. Video-based AI for beat-to-beat assessment of cardiac function. *Nature.* 2020;580(7802):252-256.
26. Zingg T. *Beitrag zur Schotteranalyse* [PhD]. Zürich, Switzerland, ETH Zürich; 1935.
27. Fonseca J, O'Sullivan C, Coop MR, Lee PD. Non-invasive characterization of particle morphology of natural sands. *Soils Found.* 2012;52(4):712-722.

28. Dong CS, Wang GZ. Curvatures estimation on triangular mesh. *J Zhejiang Univ-Sc A.* 2005;6(S1):128-136.
29. Wadell H. Volume, shape, and roundness of rock particles. *J Paleontol.* 1932;40:443-451.
30. Zhong Z, Carr TR, Wu XM, Wang G. Application of a convolutional neural network in permeability prediction: A case study in the Jacksonburg-Stringtown oil field, West Virginia, USA. *Geophysics.* 2019;84(6):363-373.
31. Zhang P, Yin Z-Y, Chen W-B, Jin Y-F. CNN-based intelligent method for identifying GSD of granular soils. *Int J Geomech.* 2021;21(12):04021229.
32. Géron A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems.* 2nd Edition ed: O'Reilly Media, Inc.; 2019.
33. Nie Z, Wang X, Liang Z, Gong J. Quantitative analysis of the three-dimensional roundness of granular particles. *Powder Technol.* 2018;336:584-593.
34. Veeling BS, Linmans J, Winkens J, Cohen T, Welling M. Rotation equivariant CNNs for digital pathology. *arXiv:180603962v1* 2018.
35. Kang E, Min J, Ye JC. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Med Phys.* 2017;44(10):e360-e375.
36. Gholamy A, Kreinovich V, Kosheleva O. Why 70/30 or 80/20 relation between training and testing sets: a pedagogical explanation. *Int J Intell Technol Appl Stat.* 2018;11(2):105-111.
37. Joshi S, Saxena S, Khanna N. First steps toward CNN based source classification of document images shared over messaging app. *Signal Process Image Commun.* 2019;78:32-41.
38. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv:14091556v6* 2015.
39. Chollet F. Keras. https://github.com/keras-team/keras. 2015.
40. Zhang P, Yin ZY, Jin YF, Ye GL. An AI-based model for describing cyclic characteristics of granular materials. *Int J Numer Anal Methods Geomech.* 2020;44(9):1315-1335.

Table

Table 1 Summary of shape parameters

Parameter	Ballast				Fujian sand			
	Mean	SD	Min.	Max.	Mean	SD	Min.	Max.
AR	0.733	0.074	0.541	0.924	0.808	0.060	0.683	0.948
S	0.785	0.043	0.640	0.868	0.806	0.103	0.356	0.897
R	0.402	0.044	0.232	0.515	0.196	0.058	0.040	0.336
C_X	0.884	0.032	0.789	0.941	0.867	0.145	0.265	1.0

Note: SD = standard deviation

Table 2 Configurations of the 3D-CNN based model

Layer	Num. of nodes	Num. of kernels	Kernel initializer	Activation function		Num. of parameters
Convolutional	/	8 (3,3,3)	random_normal	ELU	valid	656
Max-pooling	/	/ (2,2,2)	/	/	valid	/
Convolutional	/	16 (3,3,3)	random_normal	ELU	valid	3472
Max-pooling	/	/ (2,2,2)	/	/	valid	/
Convolutional	/	32 (3,3,3)	random_normal	ELU	valid	13856
Max-pooling	/	/ (2,2,2)	/	/	valid	/
Fully-connected	32	/	glorot_uniform	ELU	/	479264
Fully-connected	4	/	glorot_uniform	Linear	/	128

Table 3 Prediction errors of 7 3D-CNN based models on the training set

Model	MAE				MAPE (%)			
	AR	*S*	*R*	C_X	*AR*	*S*	*R*	C_X
CNN-*x*	0.011	0.003	0.003	0.006	1.489	0.378	1.336	0.895
CNN-*y*	0.012	0.005	0.003	0.007	1.711	0.592	1.097	0.696
CNN-*z*	0.008	0.003	0.003	0.006	1.192	0.395	1.114	0.797
CNN-*xy*	0.013	0.005	0.003	0.006	1.796	0.620	1.238	0.947
CNN-*xz*	0.010	0.005	0.003	0.006	1.416	0.584	1.020	0.745
CNN-*yz*	0.010	0.004	0.003	0.005	1.475	0.478	1.011	0.695
CNN-*xyz*	0.007	0.003	0.003	0.005	1.009	0.374	0.994	0.559

Table 4 Prediction errors of 7 3D-CNN based models on the testing set

Model	MAE				MAPE (%)			
	AR	*S*	*R*	C_X	*AR*	*S*	*R*	C_X
CNN-*x*	0.078	0.020	0.022	0.045	9.563	2.636	8.789	6.597
CNN-*y*	0.072	0.029	0.026	0.052	8.755	3.697	10.390	6.981
CNN-*z*	0.064	0.021	0.026	0.046	8.154	2.711	10.090	6.128
CNN-*xy*	0.071	0.028	0.024	0.040	8.660	3.577	9.269	5.172
CNN-*xz*	0.060	0.024	0.027	0.034	7.569	3.031	10.721	5.456
CNN-*yz*	0.065	0.024	0.023	0.035	8.277	3.131	8.959	4.782
CNN-*xyz*	0.055	0.022	0.022	0.031	6.902	2.880	8.459	4.090

Table 5 Variation of predicted values of shape parameters

Particle	A	B	Mean
AR	5.767%	2.749%	4.258%
S	5.668%	4.071%	4.870%
R	10.568%	7.616%	9.092%
C_X	7.457%	3.958%	5.708%

Figure Caption

Fig. 1 (a) three principal axes of a particle; (b) factors for calculating roundness and sphericity

Fig. 2 Framework of 3D-CNN

Fig. 3 Framework of calculating 3D shape parameters of a particle using 3D-CNN

Fig. 4 Distribution of particle shape in the database

Fig. 5 Evolution of loss value during training process: (a) sub-training set; (b) validation set

Fig. 6 Feature maps generated at different layers

Fig. 7 Prediction performance of the 3D-CNN based model on (a) training set; (b) testing set

Fig. 8 Performance of 3D-CNN model developed based on different slice images (a) MAE; (b) MAPE

Fig. 9 Comparison between measured and predicted particle shape parameters for (a) AR, (b) S, (c) R and (d) C_X

Fig. 10 Relationships between prediction error and shape parameters on the testing set (a) MAE, (b) MAPE

Fig. 11 Definition of azimuth

Fig. 12 Influence of particle orientations on the predicted shape parameters of model CNN-*xyz* for (a) AR, (b) S, (c) R and (d) C_X

Fig. 13 Model performance on synthetic particles (a) illustration of synthetic particles, (b) prediction of shape parameters

Fig. 14 Identified 3D shape of particles A and B

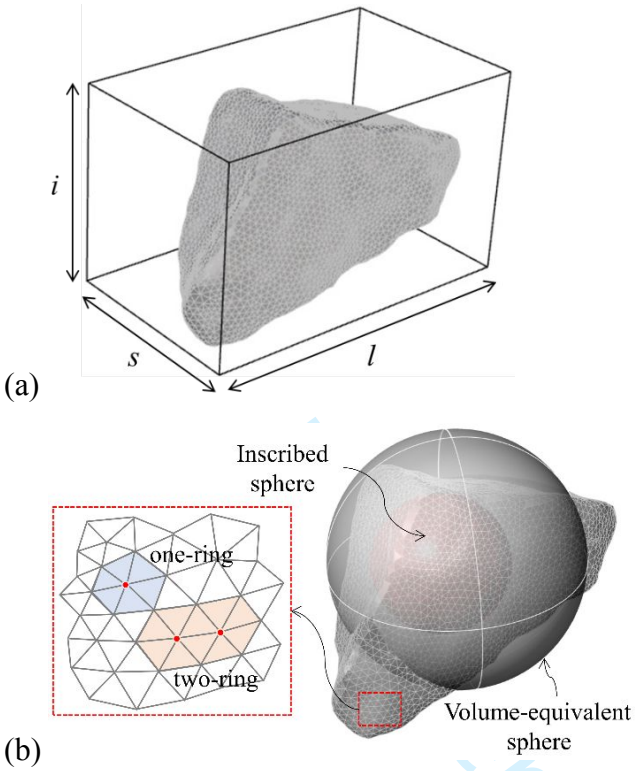


Fig. 1 (a) three principal axes of a particle; (b) factors for calculating roundness and sphericity

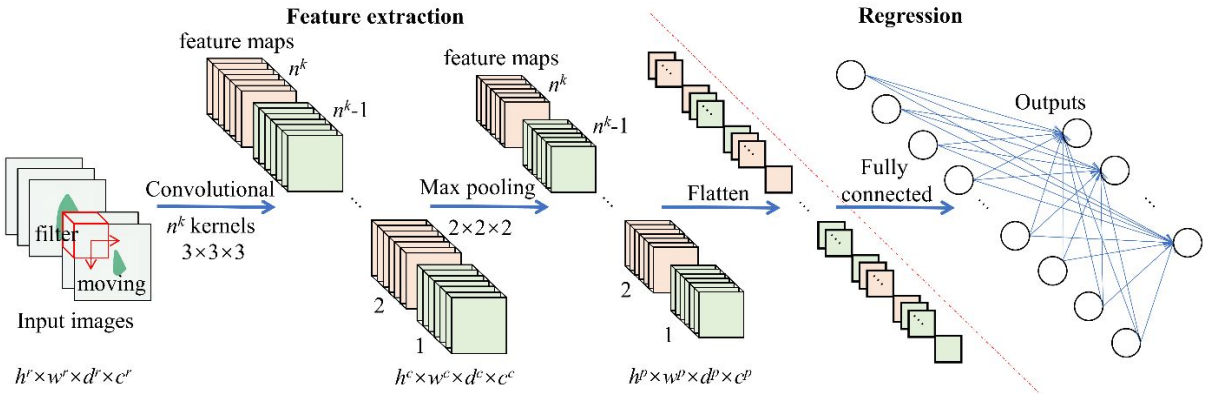


Fig. 2 Framework of 3D-CNN

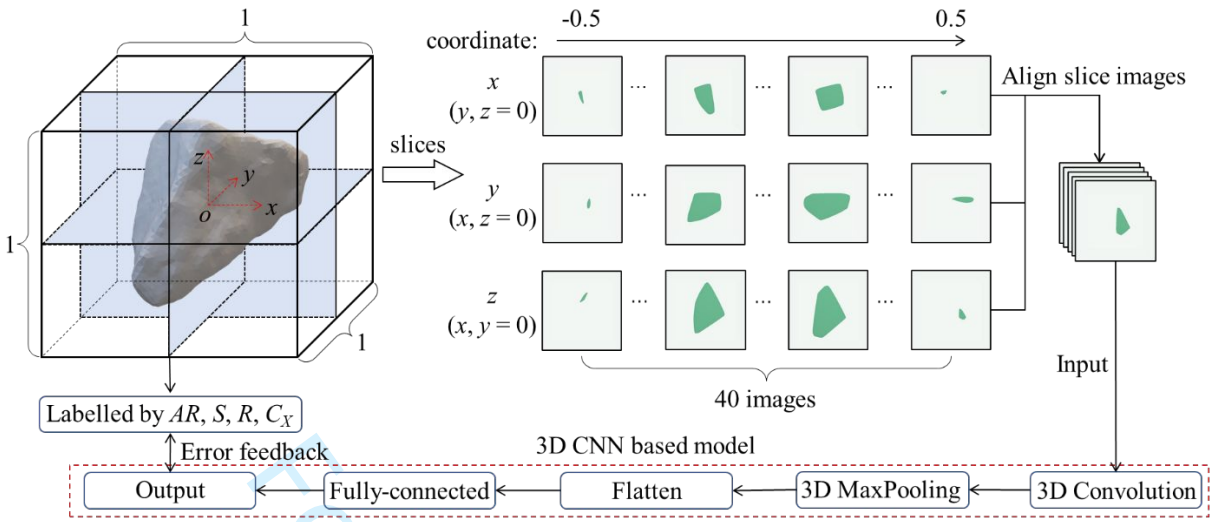


Fig. 3 Framework of calculating 3D shape parameters of a particle using 3D-CNN

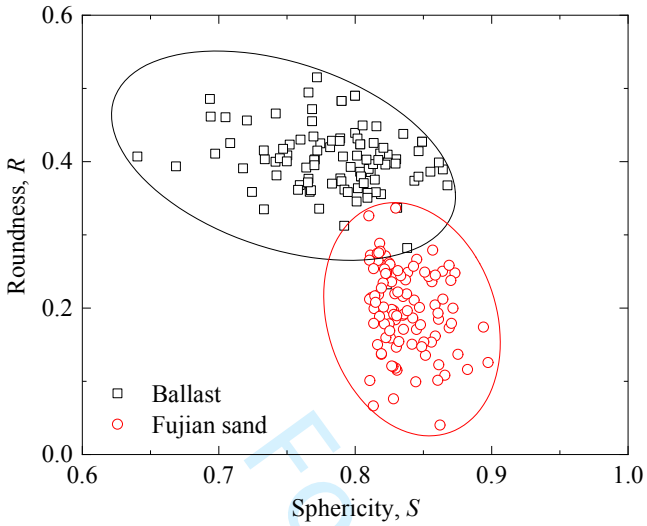


Fig. 4 Distribution of particle shape in the database

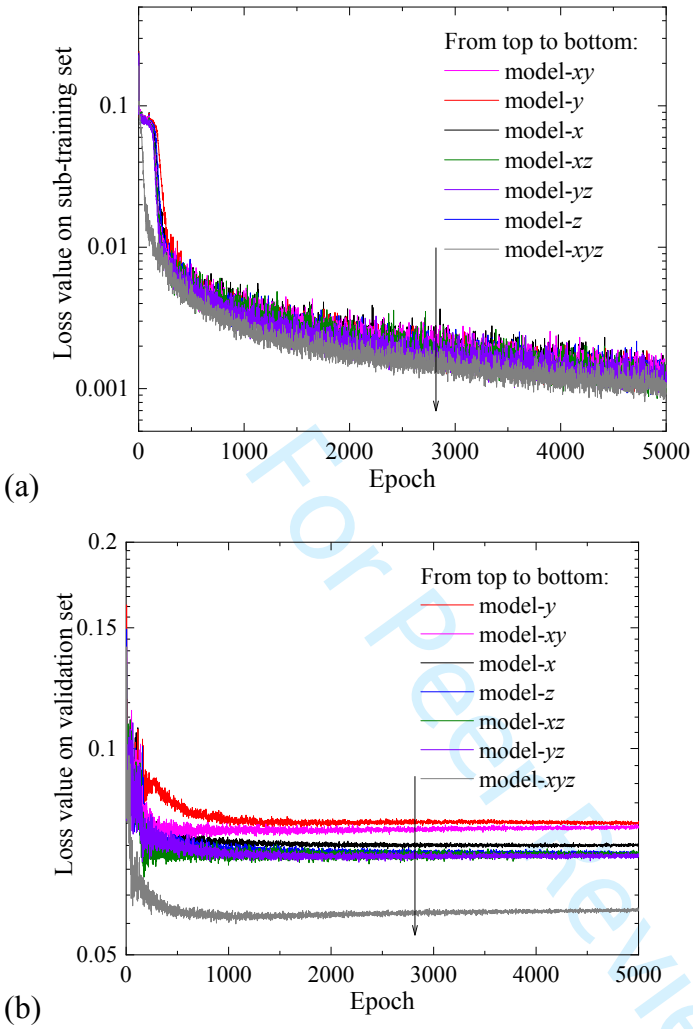


Fig. 5 Evolution of loss value during training process: (a) sub-training set; (b) validation set

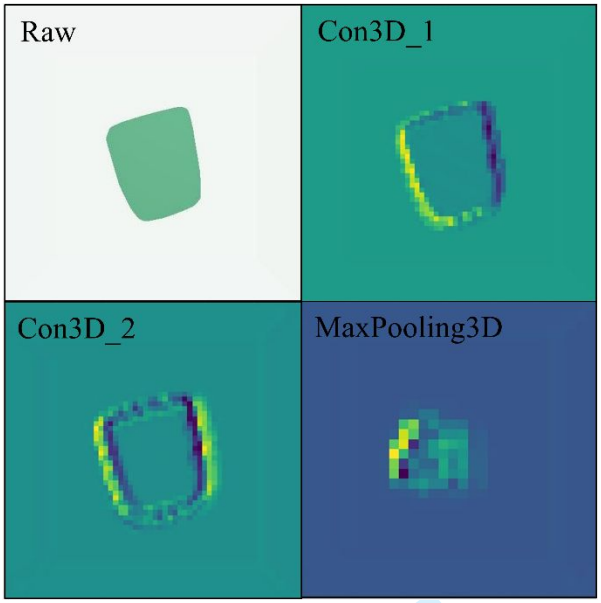


Fig. 6 Feature maps generated at different layers

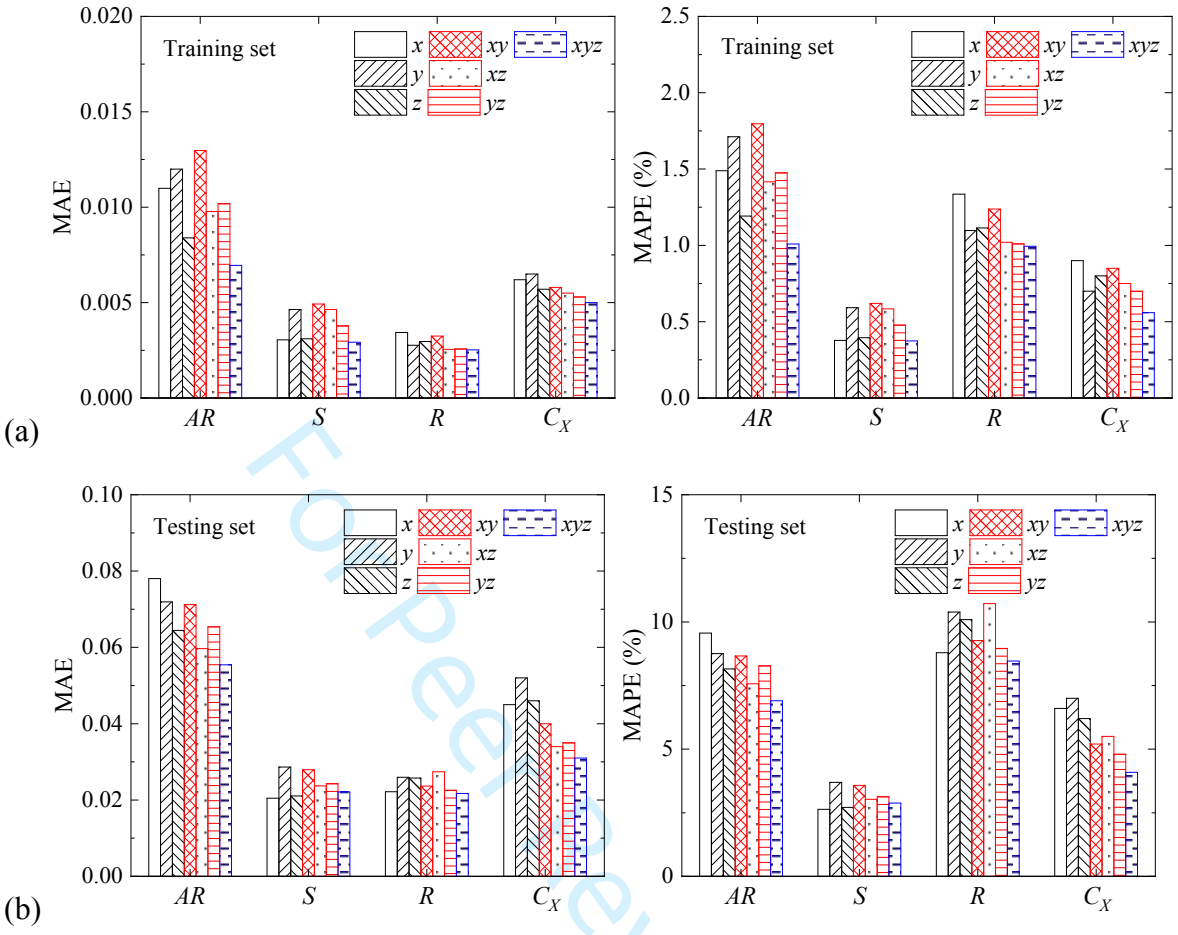


Fig. 7 Prediction performance of the 3D-CNN based model on (a) training set; (b) testing set

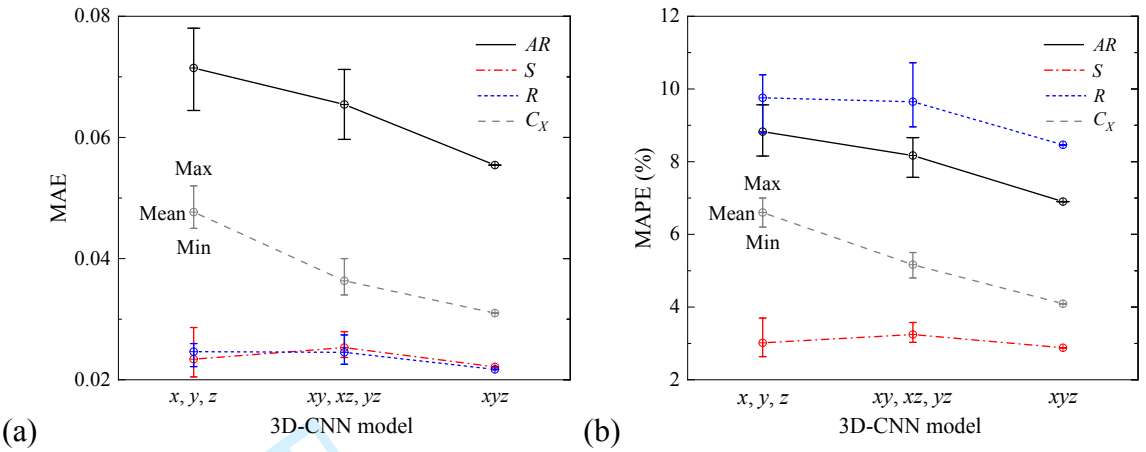


Fig. 8 Performance of 3D-CNN model developed based on different slice images (a) MAE; (b) MAPE

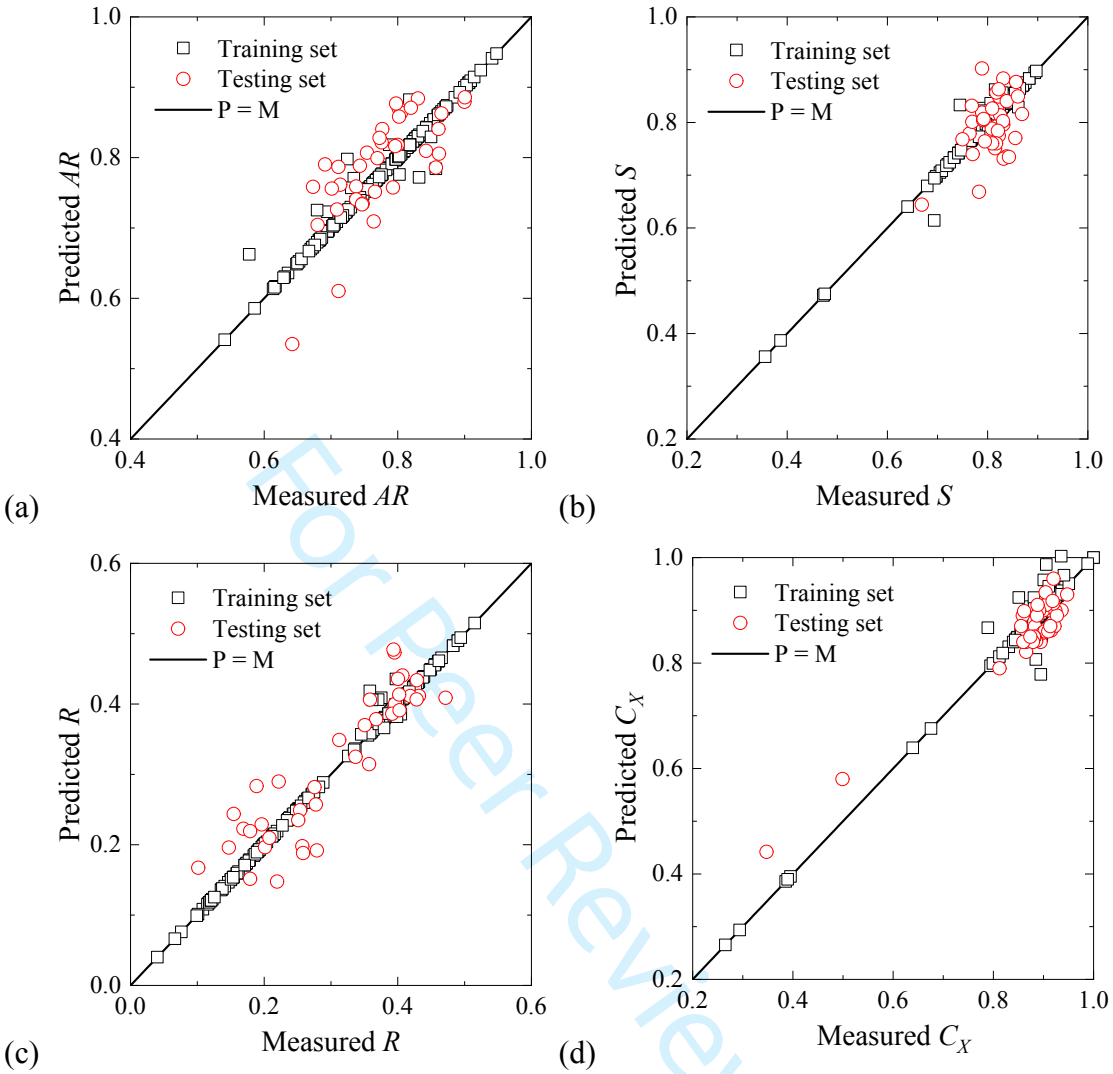


Fig. 9 Comparison between measured and predicted particle shape parameters for (a) *AR*, (b) *S*, (c) *R* and (d) C_X

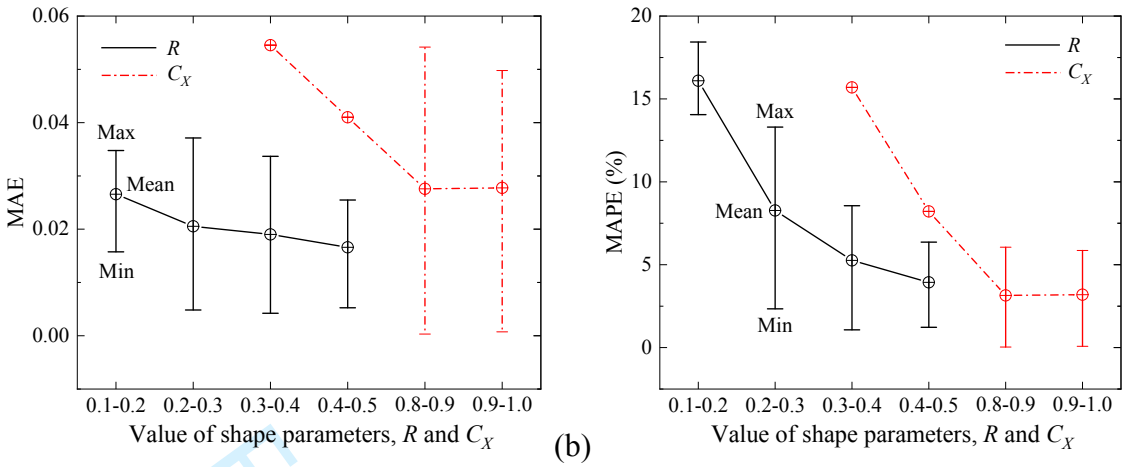


Fig. 10 Relationships between prediction error and shape parameters on the testing set (a) MAE, (b) MAPE

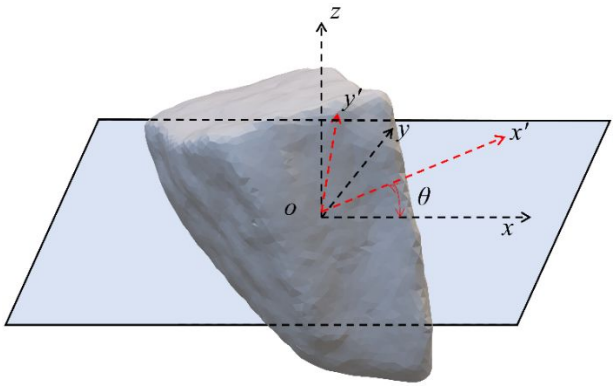


Fig. 11 Definition of azimuth

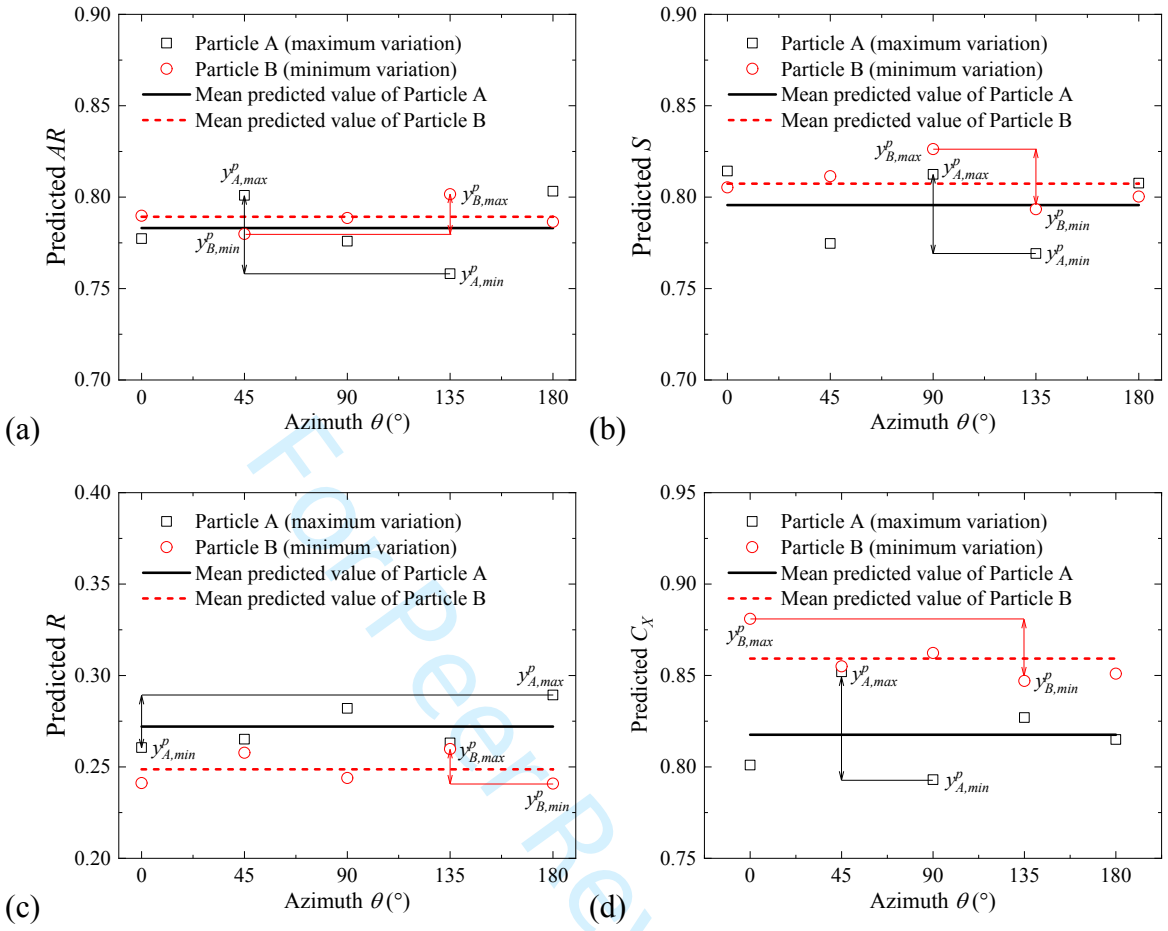
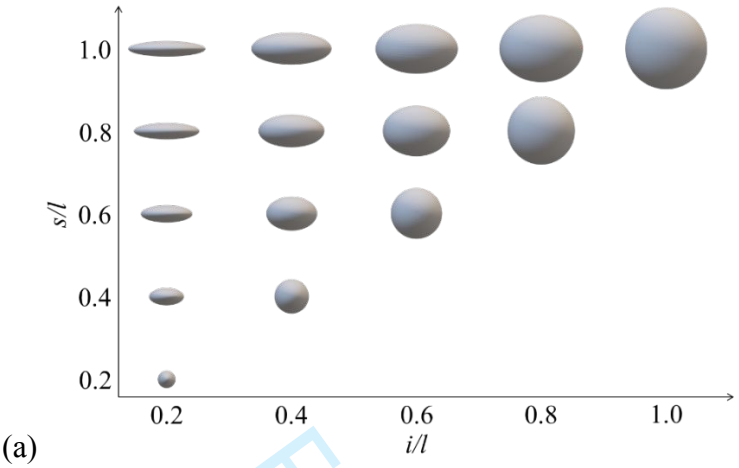
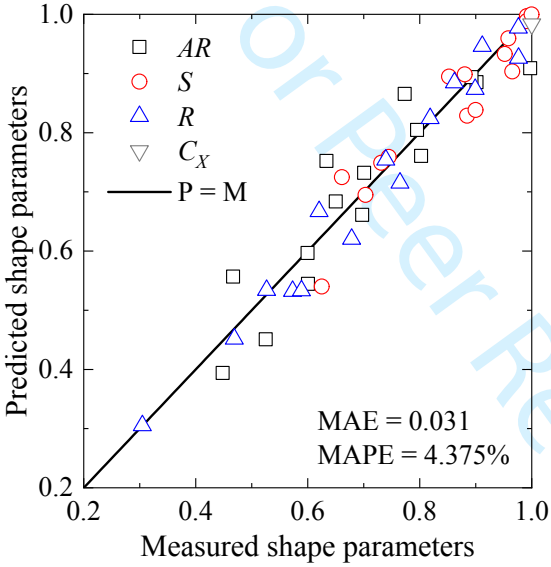


Fig. 12 Influence of particle orientations on the predicted shape parameters of model CNN-*xyz* for (a) *AR*, (b) *S*, (c) *R* and (d) C_X



(a)



(b)

Fig. 13 Model performance on synthetic particles (a) illustration of synthetic particles, (b) prediction of shape parameters

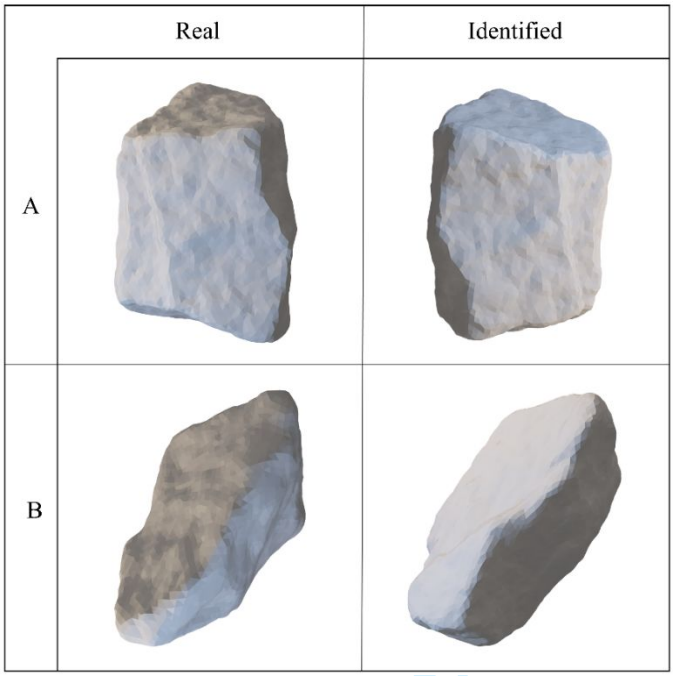


Fig. 14 Identified 3D shape of particles A and B