Mathematical Biosciences and Engineering

MBE, 20(5): 9062–9079.
DOI: 10.3934/mbe.2023398
Received: 10 December 2022
Revised: 11 January 2023
Accepted: 19 January 2023
Published: 13 March 2023

http://www.aimspress.com/journal/MBE

Research article

Combine EfficientNet and CNN for 3D model classification

Xue-Yao Gao, Bo-Yu Yang and Chun-Xiang Zhang*

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

* **Correspondence:** Email: z6c6x666@163.com.

Abstract: With the development of multimedia technology, the number of 3D models on the web or in databases is becoming increasingly larger and larger. It becomes more and more important to classify and retrieve 3D models. 3D model classification plays important roles in the mechanical design field, education field, medicine field and so on. Due to the 3D model's complexity and irregularity, it is difficult to classify 3D model correctly. Many methods of 3D model classification pay attention to local features from 2D views and neglect the 3D model's contour information, which cannot express it better. So, accuracy the of 3D model classification is poor. In order to improve the accuracy of 3D model classification, this paper proposes a method based on EfficientNet and Convolutional Neural Network (CNN) to classify 3D models, in which view feature and shape feature are used. The 3D model is projected into 2D views from different angles. EfficientNet is used to extract view feature from 2D views. Shape descriptors D1, D2, D3, Zernike moment and Fourier descriptors of 2D views are adopted to describe the 3D model and CNN is applied to extract shape feature. The view feature and shape feature are combined as discriminative features. Then, the softmax function is used to determine the 3D model's category. Experiments are conducted on ModelNet 10 dataset. Experimental results show that the proposed method achieves better than other methods.

Keywords: 3D model; EfficientNet; Convolutional Neural Network; shape feature; 2D view; discriminative feature

1. Introduction

Now, 3D models have been widely applied in construction industry, mechanical design, medical

treatment, education, computer vision, molecular biology, entertainment, e-commerce and so on. 3D model classification is an important task in computer graphics and computer vision [1]. It becomes increasingly important for retrieving, organizing, managing and storing 3D models. Therefore, research on 3D model classification is of great significance. In the beginning, features are extracted manually for 3D model classification. With the development of deep learning technology, artificial neural networks are applied for classifying 3D models. Now, scholars at home and abroad are focusing on it. It is complex to represent a 3D model and there are mainly 3 kinds of solutions to 3D model classification including the view-based method, voxel-based one, and point cloud-based one.

In the view-based classification method, a set of 2D views are used to describe 3D model and deep learning algorithms are adopted to extract discriminative features from 2D views. It is relatively simple to extract features from 2D views, which are a simple expression form of the 3D model [2]. In the voxel-based one, 3D model is expressed as a voxel matrix. 3D voxel CNN is adopted to extract deep features from the voxel matrix. 3D voxel CNN effectively enhances the capability of extracting features from a 3D model, which significantly improves the accuracy of 3D model classification [3]. In the point cloud-based one, the point cloud is preprocessed to solve its sparseness and disorder. Then, neural networks are directly used to extract features for classifying 3D models [4].

EfficientNet is a group of CNNs including 8 models between B0 and B7, in which the Swish activation function is used [5].

Feature descriptors are often used to express a 3D model's shape and structure. According to feature descriptors, 3D model classification is divided into view-based classification methods, voxel-based ones, and point cloud-based ones.

In the view-based classification method, the 3D model is projected into many 2D views. Nie proposes a multi-channel-attention CNN to represent a 3D model including view extraction, transform function learning, and descriptor generation [6]. Liu designs a fusion network to extract contextual information from continuous view sequences and get semantic information from individual views [7]. Chen uses multimodal SVM to combine 3 modalities of image features including Sift descriptor, Fourier descriptor and Zernike moments to classify objects [8]. Huang gives a view-based weight network for 3D object recognition, in which view-based weights are assigned to different projections [9]. Zhang presents an inductive multi-hypergraph learning algorithm to obtain optimal projection from 3D objects' representations [10]. Sfikas gives a method of 3D model classification and retrieval, in which 2D panoramic views are input into an ensemble of CNNs to extract features [11]. Ma uses deformable CNN to learn details and features related to geometric transformation [12]. A view-pooling layer is adopted to combine descriptors from multiple views as 3D shapes' representations. Alotaibi introduces a novel computational intelligence-based harmony search algorithm for real-time object detection and tracking (CIHSA-RTODT) technique on video surveillance systems [13]. Lin proposes a supervised classification method based on the sparse learning joint and the weighted elastic loss to extract important views for view classification task and decrease computational complexity [14].

In the voxel-based classification method, a 3D voxel matrix is used to describe 3D model. Yang applies voxelization technology to transform 3D polygon mesh into a voxel matrix. Deep voxel CNN is adopted to extract features from the voxel matrix [15]. Wang combines voxels and point clouds to design data structure Layer-Ring [16]. VoxPoint annular network on Layer-Ring is proposed to extract features and predict an object's category. Liu develops a 3D object classification system in which a learning network with a feature extractor is used [17]. Wang presents a voxel-based CNN to

extract features from normal vectors of object surfaces for 3D vision tasks [18]. Muzahid designs a multi-orientation volumetric deep neural network for 3D object classification, which limits octree partition to a certain depth for reserving all leaf octants with sparse features [19]. Kang proposes a voxel-based method to classify 3D objects by analyzing spatial characteristics [20]. Wang gives an octree-based CNN for 3D shape analysis, which takes normal vectors of 3D models sampled from the finest leaf octants as the input [21]. 3D convolution is performed on octants occupied by a 3D shape surface.

In the point cloud-based classification method, the neural network is used to extract features from a point cloud. Guo designs a network of feature fusion to classify and segment point cloud, including global feature extractor, local feature extractor and adaptive feature fusion [22]. Gao presents a 3D model classification method based on multi-head self-attention which consumes a sparse point cloud and learns its representation [23]. Ma designs a multiview-based network for 3D shape recognition and retrieval, which combines CNN with Long Short-Term Memory (LSTM) network to exploit correlative information from multiple views [24]. Maligo gives a two-layer classification model for 3D Lidar data [25]. The first layer consists of a Gaussian mixture model. This model is determined in an unsupervised manner and the second layer contains a group of categories. Zhang develops a graph CNN to classify 3D point clouds, which combines local graph convolutions with two graph down-sampling operations [26]. Ng presents a deep neural network, which uses radial basis function to exploit the local structure of point cloud and incorporates it into CNN [27]. Wang gives a neural network to classify and segment point cloud, which acts on graphs dynamically [28].

Zhang presents a transfer learning method based on pre-trained EfficientNet with a fine-tuning strategy to classify remote sensing images [29]. Alhichri designs a deep attention CNN to classify remote sensing scenes [30]. Successive convolutional layers are adopted to learn feature maps from larger regions. The attention mechanism is applied to compute the weighted average of original feature maps. Tan proposes a base network and scales it up to obtain a family of EfficientNets, which achieves better accuracy and efficiency than ConvNets [31]. Kamble uses a modified U-Net++ architecture and EfficientNet-B4 to detect OD with a cup and fovea [32]. Features from EfficientNet are utilized through skip connections in U-Net++ for precise segmentation. EfficientNet is used to extract view features from 2D views in this paper. CNN is applied to extract shape features from shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor of 2D views. Softmax function is adopted to determine the 3D model's category based on view features and shape features.

This paper proposes to combine EfficientNet and CNN to classify 3D models. 3D model is projected into a series of 2D views. 2D views, and their shape descriptors, Zernike moments and Fourier descriptors are fused to express 3D models. The purpose is to enhance the ability of describing 3D models. EfficientNet and CNN are respectively applied to extract the view features and shape features. EfficientNet is used to extract local features from 2D views of the 3D models in which the contour boundary is not emphasized. The extracted feature is local. Points are randomly sampled from the contour boundary of the 2D view and statistical methods are used to compute shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor. The contour boundary is emphasized and it makes up for the defects of view feature. CNN is adopted to extract global shape feature from shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor. The extracted feature is global. Global and local features are fused as a discriminative feature. EfficientNet is used to extract view feature and CNN is adopted to extract contour features. View feature and contour feature are fused to describe 3D models better, which can improve the performance of 3D model

classification. Then, softmax function is used to determine the 3D model's category. Its main contributions are shown as follows:

1) Combine 2D view, shape descriptors, Zernike moment and Fourier descriptor to express 3D model.

2) EfficientNet is used to extract view feature from 2D views and CNN is adopted to extract shape feature from shape descriptor D1, D2, D3, Zernike moment and Fourier descriptor.

3) Softmax function is applied to classify 3D models based on view feature and shape feature.

Now, 3D models are often projected into 2D views which are applied to determine its category in many research work. But, 3D model's contour information is not emphasized. Our work is to combine view feature extracted by EfficientNet from 2D views and shape feature extracted by CNN from shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor for classifying 3D models. We fuse view information and contour detail to express the 3D model fully for improving the performance of the 3D model classification.

The rest of the paper is arranged as follows. Expression of the 3D model and the network architecture used in this article is introduced in Section 2. Experiments are provided based on ModelNet 10 dataset and analyzed in Section 3. The work of this paper is summarized in the last section.

2. Methods

2.1. *The expression of 3D model*

2.1.1. *2D views*

3D models are very complex, and it is difficult to express and deal with them. The 3D model is often projected into a series of 2D views, which can describe its shape and structure. The fixed projection algorithm is used here. The 3D model is fixed to the center of the virtual sphere, where a virtual camera is placed above it. 3D model is rotated one circle at 60° every step to render six 2D views. They are respectively V1, V2, V3, V4, V5, V6, and input into CNN for extracting features. The fixed projection algorithm has the advantages of comprehensive and easy sampling. For 3D model 'bed', the fixed projection algorithm is adopted to extract 2D views as shown in Figure 1.

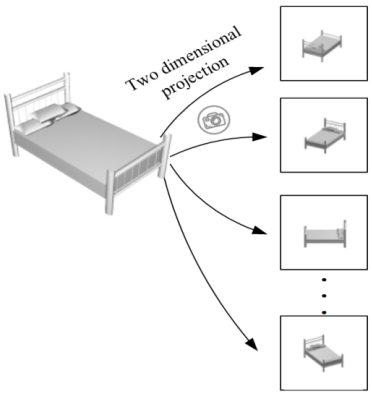

Figure 1. The projection of 3D model 'bed'.

Edge detection algorithm is applied to extract contours from 2D views.

2.1.2. *Shape descriptior D1, D2, D3*

Osada uses shape distribution to represent the 3D model. Shape function is used as descriptor to calculate statistical distribution of distance and area of the 3D model. The main idea is to use shape function to represent the relationship between point pairs on the 3D model's surface. The 2D shape distribution is principal component of the 3D shape distribution. Shape descriptor D1 is adopted to describe the distance between the centroid and a sampling point on the contour of 2D view. Shape descriptor D2 is used to describe the distance between two sampling points on the contour of the 2D view. Shape descriptor D3 is applied to describe the square root of area formed by 3 sampling points on the contour of 2D view. Figure 2 shows shape descriptors of the 2D view respectively.

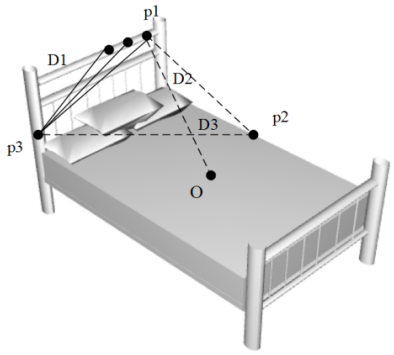

Figure 2. Shape descriptors of 2D view.

The process of computing shape distribution of the 2D view is shown as follows:

Points on the contour of 2D view are randomly sampled equidistantly, which are collected into PS = $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$.

Extract N points from PS and collect them into *PD1*. Shape distribution D1 is {*D1_1*, …, *D1_i*, …, *D1_Bins*}. Here, *D1_i* is statistics in interval (*BinsSize**(*i*-1), *BinsSize***i*), *Bins* represents the number of intervals, and *BinsSize* is the interval length. D1_i is computed as shown in formula (1).

$$D1\_i = \left|\left\{P \middle| dist(P,O) \in (BinSize*(i-1), BinSize*i), P \in PD1\right\}\right| \tag{1}$$

Where, $BinsSize = max(\{dist(P, O)|P \in PD1\})/N$, *dist*() is Euclidean distance between two points, max() is the function of taking maximum value. O is the centroid of 2D view.

Extract N point pairs from PS and collect them into *PD2*. A point pair is constructed by 2 points from PS. Shape distribution *D2* is {*D2_1*, …, *D2_i*, …, *D2_Bins*}. Here, *D2_i* represents statistics in interval (*BinSize**(*i*-1), *BinsSize***i*). *D2_i* is calculated as shown in formula (2).

$$D2\_i = \left|\left\{P \middle| dist(P) \in (BinSize*(i-1), BinSize*i), P \in PD2\right\}\right| \tag{2}$$

Where, $BinsSize = max(\{dist(P)| P \in PD2\})/N$.

Extract *N* point triples from PS and collect them into *PD3*. A point triple is constructed by 3 points from *PS*. Shape distribution *D3* is {*D3_1*, …, *D3_i*, …, *D3_Bins*}. Here, *D3_i* represents statistics in interval (*BinSize**(*i*-1), *BinsSize**i*). *D3_i* is computed as shown in formula (3).

$$D3\_i = \left| \left\{ P \middle| herson(P) \in (BinSize*(i-1), BinSize*i), P \in PD3 \right\} \right| \tag{3}$$

Where, $BinSize = \max(\{\sqrt{herson(P)} \mid P \in PD3\})/N$.

Here, herson() represents Helen formula, which is used to compute the area of triangle constructed by *P* = (*P1*, *P2*, *P3*) as shown in formulas (4) and (5).

$$herson(P) = herson(P1, P2, P3) = \sqrt{s(s-a)(s-b)(s-c)} \tag{4}$$

$$s = \frac{1}{2} * (a+b+c) \tag{5}$$

Where, *a* = dist (P1, P2), *b* = dist (P1, P3), *c* = dist (P2, P3).

D1_i, *D2_i* and *D3_i* (*i* = 1, 2, …, *Bins*) are concatenated to form shape distribution *DV* = (*D1_1*, …, *D1_i*, …, *D1_Bins*, *D2_1*, …, *D2_i*, …, *D2_Bins*, *D3_1*, …, *D3_i*, …, *D3_Bins*). Shape distribution feature has the scale invariance.

2.1.3. *Zernike moment*

In order to describe shape difference better, Zernike moment is used to express features of the 3D model. Zernike moment can express global feature of 2D view. Zernike moment of 2D view is extracted and normalized to the interval [0, 1]. Zernike moments are defined as Zernike polynomials inside unit circle. Formula (6) defines a set of complex functions $V_{nm}(x,y)$ on unit circle.

$$V_{nm}(x,y) = V_{nm}(\rho,\theta) = R_{nm}(\rho)\exp(jm\theta) \tag{6}$$

Where, ρ is the vector length from the origin to point (x, y). ϑ is the counterclockwise angle between the vector and x-axis. *j* is plural unit. *n* and *m* are the dimensions of Zernike moments.

Zernike moment Z_{nm} of 2D view $V(x, y)$ is shown in formula (7).

$$\begin{aligned} Z_{nm} &= \frac{n+1}{\pi}\sum_x\sum_y V(x,y)V_{nm}(\rho,\theta) \\ &= \frac{n+1}{\pi}\sum_x\sum_y f(\frac{x}{m_{00}} - \frac{m_{01}}{m_{00}}, \frac{y}{m_{00}} - \frac{m_{10}}{m_{00}})V_{nm}(\rho,\theta) \end{aligned} \tag{7}$$

Where, m_{01} represents the sum of the abscissa of all black pixels in 2D view. m_{10} denotes the sum of vertical coordinates of all black pixels in the 2D view. m_{00} is the sum of white pixels in 2D view.

A lot of experiments show that the larger value of Zernike moments are suitable to represent global view features. Here, the front 25 larger moments are selected to form feature vector.

Zernike moment Z_{nm} of 2D view $V(x, y)$ has the translation, scale and rotation invariance.

2.1.4. *Fourier descriptor*

Fourier descriptor is used to express 2D view. The contour of 2D view is obtained. Its 1D Fourier operator is extracted and normalized to the interval (0, 1). The centroid distance is used, which is the distance between the point in the contour and the centroid.

Suppose that PV represents the set of all pixels in the contour and N denotes the number of boundary pixels. The centroid of the object (x_c, y_c) is denoted as:

$$x_c = \frac{1}{N+1}\sum_{i=0}^{N} x(i), y_c = \frac{1}{N+1}\sum_{i=0}^{N} y(i) \tag{8}$$

Where, (x(i), y(i)) denotes the location of the ith pixel in PV. Here, r(i) is the distance between (x(i), y(i)) to the centroid, as shown in formula (9).

$$r(i) = \left(\left[x(i) - x_c\right]^2 + \left[y(i) - y_c\right]^2\right)^{1/2} \tag{9}$$

Fourier transform is applied to r(i), and Fourier coefficient R(n) is shown in formula (10):

$$R(n) = \frac{1}{N}\sum_{i=1}^{N} r(i)\exp\left(\frac{-j2\pi ni}{N}\right), n = 1,\ldots,N \tag{10}$$

Where, j is complex unit. Fourier descriptor is shown in formula (11):

$$fourier = \left[\frac{|R(1)|}{|R(0)|}, \frac{|R(2)|}{|R(0)|}, \ldots, \frac{|R(n)|}{|R(0)|}\right] \tag{11}$$

The smaller value of Fourier descriptor is suitable for representing global features. Here, the front 18 smaller values of Fourier descriptor are selected. Fourier descriptor has also the translation, scale and rotation invariance.

2.2. *3D model classification based on EfficientNet and CNN*

The 3D model is projected into a set of 2D views for expressing its shape. The advantage is that the dimension is decreased from 3 to 2 and neural network can be applied to 2D view for extracting view feature. But, contour detail is not emphasized. Here, points are randomly sampled from contour boundary of 2D view and shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor are computed. Contour feature is extracted. Then, contour feature is used to augment the description ability of view feature.

EfficientNet and CNN are combined to extract discriminative features from the 3D model. The framework of the proposed network is designed as shown in Figure 3. It includes data-processing, EfficientNet, CNN, flatten layer, merge layer, fully connected layer and softmax function. The 3D model is rotated one circle at 60° every step and six 2D views including *V1*, *V2*, *V3*, *V4*, *V5*, *V6* are gotten. They are input into EfficientNet for extracting view feature. Shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor of 2D views are computed based on *V1*, *V2*, *V3*, *V4*, *V5*, *V6*.

Then, they are input into CNN for extracting shape feature. View feature and shape feature are processed in flatten layer. They are concatenated in merge layer. The fused feature is input into fully connected layer and softmax function is adopted to classify 3D models.

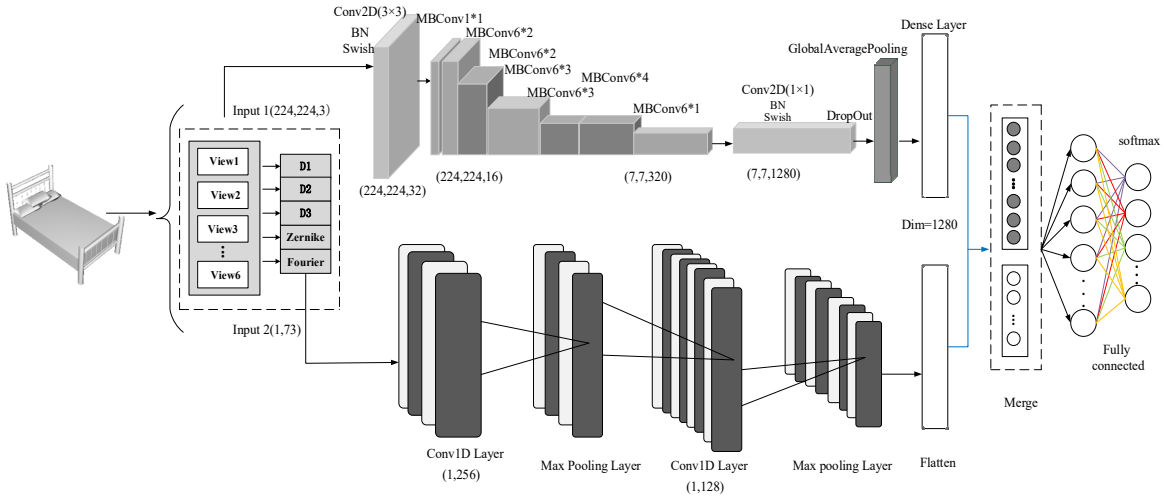

Figure 3. The framework of the proposed network.

Efficientnet-B0 is adopted in the proposed network. It consists of 16 mobile inverted bottleneck convolutions including 1 MBConv1 and 15 MBConv6, 2 convolution layers including Conv2D (1 × 1) and Conv2D (3 × 3), 1 global average pooling layer (GAP). Its parameters are shown in Table 1.

Table 1. Efficientnet-B0 network parameter.

Stage	Operator	Resolution	Channels	Layers
1	Conv3 × 3	224 × 224	32	1
2	MBConv1, k3 × 3	122 × 122	16	1
3	MBConv6, k3 × 3	122 × 122	24	2
4	MBConv6, k5 × 5	56 × 56	40	2
5	MBConv6, k3 × 3	28 × 28	80	3
6	MBConv6, k5 × 5	28 × 28	112	3
7	MBConv6, k5 × 5	14 × 14	192	4
8	MBConv6, k3 × 3	7 × 7	320	1
9	Conv1 × 1 & Pooling & FC	7 × 7	1280	1

The expansion ratio of MBConv1 and MBConv6 is respectively 1 and 6. Here, k3 × 3 and k5 × 5 represents the size of convolution kernels respectively. The size of the input 1 is 224*224*3.

2D view with 224*224*3 is input into Conv2D (3 × 3) for extracting preliminary feature, in

which BN and Swish activation function are used. The feature is processed in stacked way by 16 MBconvs whose expansion ratios are respectively 1 and 6. Conv2D (1 × 1), dropout layer, global average pooling layer and dense layer are used in sequence to get feature vector whose dimension is 1280.

CNN is a feedforward neural network and is often used to process sequence data. In order to get discriminative information, multilayer CNN is adopted to compress shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor for extracting efficient features. Two convolutional layers and two max-pooling layers are adopted. These two convolutional layers are respectively Conv1D (1, 256) and Conv1D (1, 128). Relu activation functions are used. In order to avoid the overfitting problem, dropout is added. Shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor are normalized. The size of input 2 is 1*73 and it is input into multilayer CNN. After it is processed by Conv1D (1, 256), Conv1D (1, 128) and two max-pooling layers, the size of the output is 1*128.

EfficientNet is used to extract view feature X_E from 2D view. CNN is applied to extract shape feature X_S from shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor of 2D view. X_E and X_S are processed by flatten layer. They are fused in merge layer as shown in formula (12).

$$X = Merge(X_E, X_S) \tag{12}$$

The fused feature is input into softmax layer for classifying 3D models. Softmax function has wide applications in machine learning and deep learning. When multi-class problems are processed, softmax function is used to convert the output into relative probabilities. Here, softmax function is adopted to determine category of 3D model M as shown in formula (13).

$$Y = soft\max(WX + B) \tag{13}$$

Output is Y = (y_1, y_2, ..., y_m). For model M, there are m categories. W and B are parameters in softmax layer. $P(s_i|M)$ indicates the probability that the category of model M is s_i, as shown in formula (14).

$$P(s_i \mid M) = e^{y_i} \Big/ \sum_{l=1}^{m} e^{y_l} \tag{14}$$

Category of model M is determined as shown in formula (15).

$$s = \arg\max_{i=1,2,...,m} P(s_i \mid M) \tag{15}$$

The process of classifying model M is shown as follows:

Input: 2D views, shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor.

Output: Category of model M.

Step 1. Use EfficientNet to extract view feature X_E from 2D views.

Step 2. Use CNN to extract shape feature X_S from shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor.

Step 3. Merge X_E and X_S to get discriminative feature X according to formula (10).

Step 4. Category of model M is determined according to formulas (12)–(14).

3. Experiments

ModelNet10 dataset is used to evaluate the proposed method in this paper. It is a model library published by researchers in Princeton university for evaluating 3D model classification. It includes 4899 CAD models and 10 categories. Training set and test set are officially specified, including 3991 training models and 908 test models respectively. We extract view features of training models and test models respectively. EfficientNet-B0 is chosen in experiments. The server includes CPU Intel (R) Core (TM) i5-6300HQ at 2.30 GHz and 12 GB RAM. GPU is NVDIA_Geforce GTX 950M. Keras 2.1.5 deep learning framework and python 3.6 programming language are used.

Four groups of experiments are conducted. The first group of experiments are conducted to compare the performance of CNN-based method with D1 + D2 + D3 + Zernike + Fourier, Efficientnet-based method with view feature, the proposed one. The second group of experiments are performed to testify the influence of shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor on the proposed network. The third group of experiments are conducted to investigate the influence of block number on the proposed network. The fourth group of experiments are performed to testify the influence of iteration time on the proposed network.

The first group of experiments include Experiments 1–4, which investigate the influence of shape feature on the 3D model classification. In Experiment 1, Efficientnet is used to extract discriminative features from 2D views and softmax function is adopted to classify 3D models. Training set is used to optimize Efficientnet. Then, test set is adopted to testify the performance of Efficientnet. In Experiment 2, CNN is used to extract discriminative features from shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor. Then, softmax function is adopted to classify the 3D models. Training set is used to optimize CNN. Then, test set is adopted to testify the performance of CNN. In Experiment 3, training set is used to optimize the proposed network. Then, test set is adopted to testify the performance of the proposed network. In Experiment 4, Mobilenet is used to extract discriminative feature from 2D views and softmax function is adopted to classify the 3D models. Training set is used to optimize Mobilenet. Then, test set is adopted to testify the performance of Mobilenet. The accuracies of 4 experiments are shown in Table 2.

Table 2. Accuracies of the 3D model classification under Efficientnet, CNN, Efficientnet + CNN and Mobilenet.

	Exp 1	Exp 2	Exp 3	Exp 4
bathtub	0.970	0.660	0.990	0.980
bed	0.960	0.480	0.970	0.880
chair	0.940	0.560	0.960	0.940
desk	0.730	0.620	0.730	0.760
dresser	0.740	0.560	0.820	0.730
monitor	0.960	0.780	0.980	0.940
night	0.830	0.580	0.800	0.800
sofa	0.930	0.680	0.970	0.910
table	0.820	0.490	0.870	0.810
toliet	0.980	0.430	0.980	0.99
Average accuracy	0.886	0.542	0.906	0.872

From Table 2, it can be seen that average accuracy of Experiment 1 is better than that of Experiment 2. Average accuracies of Experiments 1 and 2 are respectively 0.886 and 0.542. This shows that view feature has better description ability for the 3D models than D1 + D2 + D3 + Zernike + Fourier. Efficientnet has better discriminative ability than CNN. The reason may be that 2D view gives more comprehensive information about the 3D model's surface. Shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor are computed based on points sampled from 2D view's contour. The number of the sampled points is limited, and there is potential loss of information when the 3D model is described. So, average accuracy of Experiment 2 is bad. The performance of Experiment 3 is the best. This is because that when 2D views, shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor are combined, they can describe the 3D model completely from different perspectives. Shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor are fused to make up the deficiency of view feature and provide more contour information. The proposed network combines advantages of Efficientnet and CNN, and its discriminative ability is the best. View information and contour detail are all considered in the proposed network. Its accuracy is higher than that under view feature or contour feature. So, average accuracy of Experiment 3 is the best. Experiment 4 achieves better than Experiment 2 at average accuracy. This shows that the 2D views provide more discriminative information than shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor. Average accuracy of Experiment 1 is better than that of Experiment 4. This shows that the discriminative ability of Efficientnet is better than that of Mobilenet. Average accuracy of Experiment 3 is higher than that of Experiment 4. The reason is that Mobilenet extracts discriminative features from the 2D views. The proposed network uses Efficientnet and CNN to extract discriminative features respectively from the 2D views and shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor, which can express the 3D model adequately.

In order to verify the influence of shape feature on 3D model classification, 6 comparative experiments are conducted based on view feature. The second group of experiments include 6 experiments. View feature is input into Efficientnet of the proposed network. In Experiment 5, D1 is input into CNN of the proposed network. In Experiment 6, D2 is input into CNN of the proposed network. In Experiment 7, D3 is input into CNN of the proposed network. In Experiment 8, Zernike moment is input into CNN of the proposed network. In Experiment 9, Fourier descriptor is input into CNN of the proposed network. In Experiment 10, D1 + D2 + D3 + Zernike + Fourier is input into CNN of the proposed network. Training set is used to optimize the proposed network and test set is adopted to testify it as shown in Table 3.

From Table 3, it can be seen that view + D1, view + D2 and view + D3 achieve better than view + Zernike and view + Fourier. Average accuracies of view + D1, view + D2 and view + D3 are respectively 0.894, 0.891 and 0.895, and they are almost equal. Shape descriptors D1, D2 and D3 have the same ability of describing the 3D model. Average accuracies of view + Zernike, view + Fourier are respectively 0.882 and 0.875. The description ability of Zernike moment is better than that of Fourier descriptor. This shows that shape descriptors D1, D2 and D3 can better describe the 3D model than Zernike moment and Fourier descriptor. This is because that D1, D2 and D3 features have better ability of description. The performance of view + D1 + D2 + D3 + Zernike + Fourier is the best. This is because that the 3D model can be described completely from different perspectives, when the 2D views, shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor are combined.

Table 3. The influence of shape feature on the proposed network.

	Exp 5	Exp 6	Exp 7	Exp 8	Exp 9	Exp 10
bathtub	0.980	0.990	0.960	0.980	0.990	0.990
bed	0.960	0.940	0.930	0.970	0.970	0.970
chair	0.930	0.940	0.960	0.940	0.950	0.960
desk	0.690	0.740	0.760	0.740	0.680	0.730
dresser	0.790	0.780	0.750	0.690	0.720	0.820
monitor	0.960	0.970	0.970	0.960	0.990	0.980
night	0.790	0.820	0.840	0.880	0.850	0.800
sofa	0.950	0.900	0.950	0.890	0.870	0.970
table	0.900	0.840	0.840	0.860	0.840	0.870
toliet	0.990	0.990	0.990	0.980	0.950	0.980
Average accuracy	0.894	0.891	0.895	0.882	0.875	0.906

The block number of Efficientnet affects the performance of the proposed network. All blocks all contain 1*1 convolutional kernel. Each block contains 3*3 or 5*5 convolutional kernels. When there are more blocks in Efficientnet, accuracy of the proposed network increases. But, time and space costs will grow greatly. The third group of experiments are performed to investigate the influence of the block number on the 3D model classification. 1–7 blocks are respectively used in Efficientnet of the proposed network. The training set is respectively used to optimize these 7 networks and test set is adopted to testify them as shown in Table 4.

Table 4. Accuracies of 3D model classification under different block number.

	1	2	3	4	5	6	7
bathtub	0.93	0.94	0.98	0.93	0.97	0.98	0.99
bed	0.88	0.87	0.96	0.95	0.96	0.94	0.97
chair	0.89	0.88	0.94	0.96	0.95	0.94	0.96
desk	0.69	0.76	0.79	0.75	0.70	0.75	0.73
dresser	0.64	0.66	0.69	0.74	0.74	0.81	0.82
monitor	0.79	0.83	0.94	0.93	0.95	0.98	0.98
night	0.77	0.73	0.75	0.75	0.82	0.78	0.80
sofa	0.63	0.76	0.85	0.91	0.91	0.94	0.97
table	0.72	0.74	0.83	0.83	0.85	0.82	0.87
toliet	0.90	0.92	0.95	0.98	0.98	0.98	0.98
Average accuracy	0.788	0.806	0.865	0.877	0.884	0.893	0.906

From Table 4, it can be seen that the larger is the block number, the higher is average accuracy of the proposed network. Average accuracy of the proposed network with 1 block is the lowest and its value

is 0.788. Average accuracies of the proposed network with 2 blocks and 3 blocks are respectively 0.806 and 0.865. The proposed network increases greatly on average accuracy from 2 blocks to 3 blocks and there is an increase of nearly 7.3%. There is a fast growth from 3 blocks to 4 blocks. But, there is a slow growth from 4 blocks to 5 blocks. The growth rate from 5 blocks to 6 blocks becomes slower. There is only a smaller increase from 6 blocks to 7 blocks. But, time and space complexity is increasing greatly. At the same time, the growth speed of average accuracy is decreasing.

Iteration number affects the performance of the proposed network. The fourth group of experiments are performed to testify the influence of iteration number on the 3D model classification. Training set is used to optimize the proposed network respectively with 20, 40, 60, 100 and 200 times. Then, test set is adopted to testify these 5 networks as shown in Table 5.

Table 5. Accuracies of the proposed network under different iteration number.

	20	40	60	100	200
bathtub	0.980	0.960	0.960	0.990	0.990
bed	0.890	0.970	0.930	0.950	0.970
chair	0.870	0.920	0.950	0.950	0.960
desk	0.670	0.810	0.750	0.740	0.730
dresser	0.710	0.710	0.710	0.790	0.820
monitor	0.860	0.960	0.980	0.980	0.980
night	0.810	0.820	0.830	0.810	0.800
sofa	0.870	0.920	0.950	0.910	0.970
table	0.840	0.790	0.800	0.830	0.870
toliet	0.970	0.970	0.990	0.990	0.980
Average accuracy	0.843	0.882	0.884	0.895	0.906

From Table 5, it can be seen that as iteration time increases, average accuracy of the proposed network becomes the larger. When the proposed network is iterated 20 times and 40 times, its average accuracy is respectively 0.843 and 0.882. There is an increase of nearly 4.6% and the growth speed is very fast. When the proposed network is iterated 60 times, its average accuracy is 0.884. Average accuracy of the proposed network increases less from 40 epochs to 60 epochs. There is only an increase of 1.2% from 60 epochs to 100 epochs. But, iteration number increases greatly. There is only an increase of 1.2% from 100 epochs to 200 epochs. But, iteration number doubles. When iteration time is set to 200, the proposed network achieves the best. But, time and space complexity is increasing greatly. At the same time, the growth speed of average accuracy is decreasing. So, iteration time is set to 200 in experiments. For category 'bathtub', accuracy of the proposed network at 20 iterations is 0.98. At 40 and 60 iterations, accuracy of the proposed network is 0.96 and its accuracy decreases. This is because that there is the overfitting problem in the process of training the proposed network. But, with the increase of iteration number, the generalization ability of the proposed network is improved to some extent and it can extract effective features. So, accuracy of the proposed network is improved and its accuracy is 0.99 at 200 iterations.

In order to investigate the convergence of the proposed network, 90% of training set are selected as training data and the rest is used as validation data. In order to extract effective features from

training data, initial learning rate is set to 1E-4 and batchsize is set to 16. Iteration number is set to 200. At each epoch, training data is used to optimize the proposed network. The optimized network is adopted to classify the 3D models in training data and accuracy is computed. At the same time, the optimized network is used to classify the 3D models in validation data and accuracy is calculated as shown in Figure 4.

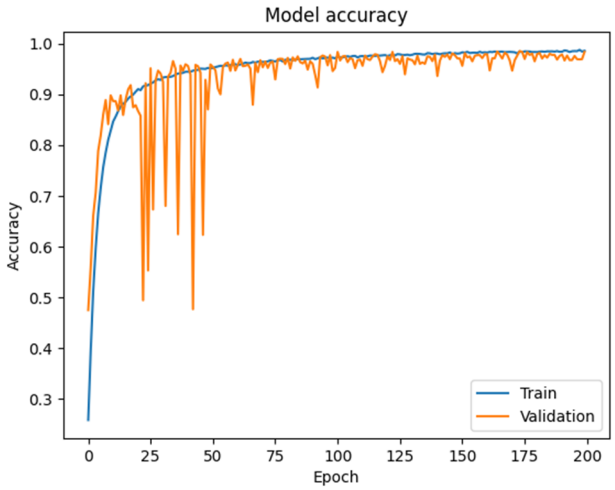

Figure 4. Accuracies of training data and validation data.

From Figure 4, it can be seen that the proposed network has rapid convergence capability. With the increase of iteration number, accuracy of training data increases. But, accuracy of validation data fluctuates wildly. Sometime, accuracy of validation data is higher than that of training data. But, it is lower than that of training data for the rest time. The fluctuation range decreases with the increase of epochs. After 50 epochs, accuracies of training data and validation data are all over 90%. Training curve also reaches steady at 100 epochs. These two curves show that the proposed network is trained well. At the same time, the proposed network achieves the best at 200 epochs. At 200 epochs, accuracy of validation data reaches steady. Iteration number is set to 200 in experiment, and the proposed network is optimized. Then, the optimized network is applied to classify test data and accuracy achieves 90.6%.

Loss of the proposed network for training data and validation data in training process is shown in Figure 5.

From Figure 5, it can be seen that the loss steadily decreases and convergence speed is fast at learning rate 1E-4 on training data and validation data. With the increase of iteration number, the loss of training data decreases. But, the loss of validation data fluctuates wildly. Sometime, the loss of validation data is lower than that of training data. But, it is higher than that of training data for the rest time. The fluctuation range decreases with the increase of epochs. After 50 epochs, the loss of the proposed network is lower than 0.3 for training data and validation data. Training curve also reaches steady at 100 epochs. But validation curve reaches steady at 175 epochs. These two curves show that the proposed network is trained well at 175 epochs. At the same time, the proposed network achieves the best at 200 epochs. At 200 epochs, the loss of validation data reaches steady. Iteration number is set to 200 in experiment, and the proposed network is optimized. On training data and validation data, the

loss tends to decrease and gradually achieves a balance. This shows that the proposed network has faster convergence rate and better convergence ability.

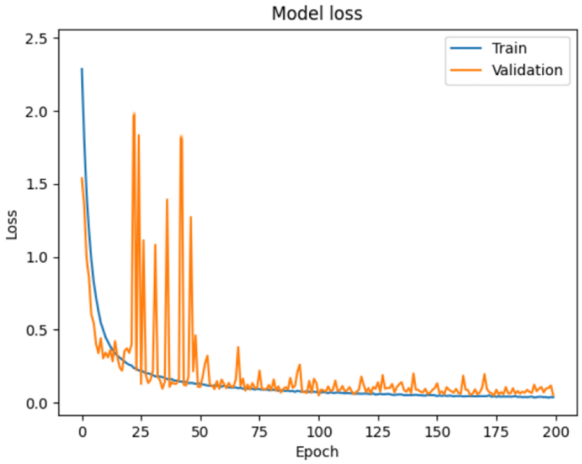

Figure 5. Loss of the proposed network.

4. Conclusions

This paper combines Efficientnet and CNN to design a new network for 3D model classification. It includes Efficientnet, CNN, flatten layer, merge layer and softmax function. The 3D model is fixed to the center of the virtual sphere and virtual camera is placed above it. The 3D model is rotated one circle at 60° every step and six 2D views are gotten. View feature is extracted from 2D views by EfficientNet, which consists of 16 mobile inverted bottleneck convolutions including 1 MBConv1 and 15 MBConv6, 2 convolution layers including Conv2D (1 × 1) and Conv2D (3 × 3), 1 global average pooling layer. An edge detection algorithm is applied to extract contours from 2D views. Shape descriptors D1, D2, D3, Zernike moment, Fourier descriptor of 2D views are adopted to describe the 3D model and the shape feature is extracted by CNN, which consists of 2 convolutional layers and 2 maximum pooling layers. Softmax function is adopted to classify 3D models based on the view feature and shape feature. Experiments are conducted on the ModelNet10 dataset and the proposed network is iterated with 200 epochs. Experiments show that the performance of the proposed network is better than those of CNN with D1 + D2 + D3 + Zernike + Fourier and Efficientnet with view feature. The accuracy of the proposed method achieves 90.6%. The reason is that EfficientNet is used to extract view information and CNN is adopted to extract contour detail. View information and contour detail are combined to describe a 3D model, which can discriminate between two 3D models better.

The novelty of this paper is that local features extracted by Efficientnet and global features extracted by CNN are fused for classifying 3D models. Efficientnet extracts local features from 2D views, which does not pay more attention to contour boundary. CNN extracts global features from shape descriptors D1, D2, D3, Zernike moment and Fourier descriptor, which only consider points in the contour boundary of 2D views. Global features and local features are fused to make up for their defects. Previous methods only use view information or only contour detail. But, they are not

considered together to describe the 3D model. In this paper, we combine local feature extracted by Efficientnet and global feature extracted by CNN for the 3D model classification, which complements one another perfectly.

It is key for 3D model classification to express 3D model's shape and structure comprehensively. So, more descriptors will be introduced to describe 3D models in the future. At the same time, more neural networks will be tried to find effective discriminative features for classifying 3D models.

Acknowledgments

This study was supported by Heilongjiang Provincial Natural Science Foundation of China (No. LH2022F030).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. W. Tangelder, R. C. Veltkamp, A survey of content based 3D shape retrieval methods, *Multimedia Tools Appl.*, **39** (2008), 441–471. https://doi.org/10.1007/s11042-007-0181-0
2. H. Y. Zhou, A. A. Liu, W. Z. Nie, J. Nie, Multi-view saliency guided deep neural network for 3-D object retrieval and classification, *IEEE Trans. Multimedia*, **22** (2020), 1496–1506. https://doi.org/10.1109/TMM.2019.2943740
3. C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, L. Guibas, Volumetric and multi-view CNNs for object classification on 3D data, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 5648–5656. https://doi.org/10.1109/CVPR.2016.609
4. X. A. Li, L. Y. Wang, J. Lu, Multiscale receptive fields graph attention network for point cloud classification, *Complexity*, **2021** (2021), 1076–2787. https://doi.org/10.1155/2021/8832081
5. Y. L. Zhang, J. T. Sun, M. K. Chen, Q. Wang, Y. Yuan, R. Ma, Multi-weather classification using evolutionary algorithm on EfficientNet, in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events*, (2021), 546–551. https://doi.org/10.1109/PerComWorkshops51409.2021.9430939
6. W. Nie, K. Wang, Q. Liang, R. He, Panorama based on multi-channel-attention CNN for 3D model recognition, *Multimedia Syst.*, **25** (2019), 655–662. https://doi.org/10.1007/s00530-018-0600-2
7. A. A. Liu, F. B. Guo, H. Y. Zhou, W. Li, D. Song, Semantic and context information fusion network for view-based 3D model classification and retrieval, *IEEE Access*, **8** (2020), 155939–155950. https://doi.org/10.1109/ACCESS.2020.3018875
8. F. Chen, R. Ji, L. Cao, Multimodal learning for view-based 3D object classification, *Neurocomputing*, **195** (2016), 23–29. https://doi.org/10.1016/j.neucom.2015.09.120
9. Q. Huang, Y. Wang, Z. Yin, View-based weight network for 3D object recognition, *Image Vision Comput.*, **93** (2020). https://doi.org/10.1016/j.imavis.2019.11.006

10. Z. Zhang, H. Lin, X. Zhao, R. Ji, Y. Gao, Inductive multi-hypergraph learning and its application on view-based 3D object classification, *IEEE Trans. Image Process.*, **27** (2018), 5957–5968. https://doi.org/10.1109/TIP.2018.2862625
11. K. Sfikas, I. Pratikakis, T. Theoharis, Ensemble of panorama-based convolutional neural networks for 3D model classification and retrieval, *Comput. Graphics*, **71** (2018), 208–218. https://doi.org/10.1016/j.cag.2017.12.001
12. P. Ma, J. Ma, X. Wang, L. Yang, N. Wang, Deformable convolutional networks for multi-view 3D shape classification, *Electron. Lett.*, **54** (2018), 1373–1375. https://doi.org/10.1049/el.2018.6851
13. M. F. Alotaibi, M. Omri, S. Abdel-Khalek, E. Khalil, R. Mansour, Computational intelligence-based harmony search algorithm for real-time object detection and tracking in video surveillance systems, *Mathematics*, **10** (2022), 1–16. https://doi.org/10.3390/math10050733
14. Q. Lin, Z. Wang, Y. Y. Chen, P. Zhong, Supervised multi-view classification via the sparse learning joint the weighted elastic loss, *Signal Process.*, **191** (2022). https://doi.org/10.1016/j.sigpro.2021.108362
15. J. Yang, S. Wang, P. Zhou, Recognition and classification for three-dimensional model based on deep voxel convolution neural network, *Acta Optica Sinica*, **39** (2019), 1–11. http://dx.doi.org/10.3788/AOS201939.0415007
16. T. Wang, W. Tao, C. M. Own, X. Lou, Y. Zhao, The layerizing voxpoint annular convolutional network for 3D shape classification, *Comput. Graphics Forum*, **39** (2020), 291–300. https://doi.org/10.1111/cgf.14145
17. Z. Liu, S. Wei, Y. Tian, S. Ji, Y. Sung, L. Wen, VB-Net: voxel-based broad learning network for 3D object classification, *Appl. Sci.*, **10** (2020). https://doi.org/10.3390/app10196735
18. C. Wang, M. Cheng, F. Sohel, M. Bennamoun, J. Li, NormalNet: a voxel-based CNN for 3D object classification and retrieval, *Neurocomputing*, **323** (2019), 139–147. https://doi.org/10.1016/j.neucom.2018.09.075
19. A. Muzahid, W. Wan, F. Sohel, N. Ullah Khan, O. Villagómez, H. Ullah, 3D object classification using a volumetric deep neural network: an efficient octree guided auxiliary learning approach, *IEEE Access*, **8** (2020), 23802–23816. https://doi.org/10.1109/ACCESS.2020.2968506
20. Z. Kang, J. Yang, R. Zhong, Y. Wu, Z. Shi, R. Lindenbergh, Voxel-based extraction and classification of 3D pole-like objects from mobile LiDAR point cloud data, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **11** (2018), 4287–4298. https://doi.org/10.1109/JSTARS.2018.2869801
21. P. S. Wang, Y. Liu, Y. X. Guo, C. Sun, X. Tong, O-CNN: octree-based convolutional neural networks for 3D shape analysis, *ACM Trans. Graph*, **36** (2017), 1–11. https://doi.org/10.1145/3072959.3073608
22. R. Guo, Y. Zhou, J. Zhao, Y. Man, M. Liu, R. Yao, et al., Point cloud classification by dynamic graph CNN with adaptive feature fusion, *IET Comput. Vision*, **15** (2021), 235–244. https://doi.org/10.1049/cvi2.12039
23. X. Y. Gao, Y. Z. Wang, C. X. Zhang, J. Lu, Multi-head self-attention for 3D point cloud classification, *IEEE Access*, **9** (2021), 18137–18147. https://doi.org/10.1109/ACCESS.2021.3050488

24. C. Ma, Y. Guo, J. Yang, W. An, Learning multi-view representation with LSTM for 3-D shape recognition and retrieval, *IEEE Trans. Multimedia*, **21** (2019), 1169–1182. https://doi.org/10.1109/TMM.2018.2875512
25. A. Maligo, S. Lacroix, Classification of outdoor 3D lidar data based on unsupervised gaussian mixture models, *IEEE Trans. Autom. Sci. Eng.*, **14** (2017), 5–16. https://doi.org/10.1109/TASE.2016.2614923
26. Y. Zhang, M. Rabbat, A graph-CNN for 3D point cloud classification, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, (2018), 6279–6283. https://doi.org/10.1109/ICASSP.2018.8462291
27. Y. T. Ng, C. M. Huang, Q. T. Li, J. Tian, RadialNet: a point cloud classification approach using local structure representation with radial basis function, *Signal*, *Image Video Process.*, **14** (2020), 747–752. https://doi.org/10.1007/s11760-019-01607-0
28. Y. Wang, Y. Sun, Z. Liu, S. Sarma, M. Bronstein, J. Solomon, Dynamic graph CNN for learning on point clouds, *ACM Trans. Graphics*, **38** (2019), 1–12. https://doi.org/10.1145/3326362
29. D. Zhang, Z. Liu, X. Shi, Transfer learning on EfficientNet for remote sensing image clas sification, in *2020 5th International Conference on Mechanical, Control and Computer Eng ineering (ICMCCE)*, (2020), 2255–2258. https://doi.org/10.1109/ICMCCE51767.2020.00489
30. H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, N. Alajlan, Classification of remote sensing images using EfficientNet-B3 CNN model with attention, *IEEE Access*, **9** (2021), 14078–14094. https://doi.org/10.1109/ACCESS.2021.3051085
31. M. Tan, Q. Le, Efficientnet: rethinking model scaling for convolutional neural networks, *arXiv preprint*, (2019), arXiv:1905.11946. https://doi.org/10.48550/arXiv.1905.11946
32. R. Kamble, P. Samanta, N. Singhal, Optic disc, cup and fovea detection from retinal images using U-Net++ with EfficientNet encoder, in *Lecture Notes in Computer Science*, (2020), 93–103. https://doi.org/10.1007/978-3-030-63419-3_10

AIMS Press