

# Abnormal Behavior Recognition in Classroom Pose Estimation of College Students Based on Spatiotemporal Representation Learning



Yunfang Xie<sup>1</sup>, Su Zhang<sup>1</sup>, Yingdi Liu<sup>2\*</sup>

<sup>1</sup> College of Mechatronics & Electrical Engineering, Hebei Agricultural University, Baoding 071001, China

<sup>2</sup> College of Physics & Mechanical and Electrical Engineering, Shijiazhuang University, Shijiazhuang 050035, China

Corresponding Author Email: [bluesky20182018@163.com](mailto:bluesky20182018@163.com)

<https://doi.org/10.18280/ts.380109>

## ABSTRACT

**Received:** 5 November 2020

**Accepted:** 28 January 2021

### Keywords:

*artificial intelligence, college students, pose estimation, spatiotemporal representation learning, k-means clustering (KMC), convolutional neural network (CNN)*

Artificial intelligence and fifth generation (5G) technology are widely adopted to evaluate the classroom poses of college students, with the help of campus video surveillance equipment. To ensure the effective learning in class, it is important to detect and intervene in abnormal behaviors like sleeping and using cellphones in time. Based on spatiotemporal representation learning, this paper presents a deep learning algorithm to evaluate classroom poses of college students. Firstly, feature engineering was adopted to mine the moving trajectories of college students, which were used to determine student distribution and establish a classroom prewarning system. Then, k-means clustering (KMC) was employed for cluster analysis on different student groups, and identify the features of each group. For a specific student group, the classroom surveillance video was decomposed into several frames; the edge of each frame was extracted by edge detection algorithm, and imported to the proposed convolutional neural network (CNN). Experimental results show that our algorithm is 5% more accurate than the benchmark three-dimensional CNN (C3D), making it an effective tool to recognize abnormal behaviors of college students in class.

## 1. INTRODUCTION

In recent years, computer vision has developed rapidly, under the driving of data, computing power and algorithms. As the main objects of computer vision, image recognition and video recognition are similar in many aspects. However, far more scholars have explored videos than images. The video research not only involves detection and recognition [1], style conversion, and retrieval, but also covers target detection and tracking, personal behavior recognition, action prediction, and determining the nature of behaviors. Besides saving manpower, detecting student poses in surveillance videos [2] can alert students to abnormal poses, help them to cultivate a good learning habit, and prevent hunchback and myopia caused by improper poses, heralding the realization of unattended classroom.

Many conventional methods can identify student behaviors in class. The most common methods of behavior recognition include feature extraction, feature processing, and feature classification [3]. In the HMDB51 dataset, the improved dense trajectory (IDT) algorithm [4] has the highest recognition rate, reaching 57.2%. Xiao et al. [5] and Lu [6] were the first to apply deep learning extensively in video recognition [7]. Comparing the influence of four single-flow neural network structures on behavior recognition, it is found that a single frame of video contains lots of behavior features. Simonyan and Zisserman [8] divided the monitoring mode into a time flow and a space flow, regarded optical flow as the input of time flow, and conducted behavior recognition with dual-flow

convolutional neural network (CNN), revealing that the time flow has better input performance than the original frame input. Many deep learning studies on student behavior recognition [9, 10] are based on single-flow or dual-flow neural network. In traditional empirical research, the necessary pre-test hypotheses ignore some potentially useful functions. Fortunately, the emerging technology of feature engineering helps to discover some subtle influencing factors. The intrinsic development logic of these factors can associate student trajectories outside the classroom, and greatly promote the monitoring of student behaviors.

In video teaching, both time and space information should be considered to recognize the classroom behaviors of students. How to balance the relationship between the two dimensions is the key to identifying good behaviors in time and space. Our work is unfolded based on the key information needed to recognize human behaviors in videos, that is, the edge trajectories of human body. Firstly, feature engineering was implemented to generate a series of features to describe the lifestyles, income patterns, and Internet use patterns of students. Then, the features of different student groups were identified through K-means clustering (KMC) [11]. On classroom performance, the edge features of video frames were taken as the basis for the spatial dimension of behavior recognition, and the three-dimensional CNN (C3D) was adopted to fuse the spatiotemporal information of continuous frames. On this basis, repeated experiments were conducted, and a robust abnormal behavior recognition algorithm was designed to timely detect and prewarn abnormal student poses.

## 2. LITERATURE REVIEW

So far, various schemes have been adopted to analyze or monitor student behaviors in class. For example, Buniyamin et al. [12] classified and predicted student scores based on the data of campus information system. Wu et al. [13] intuitively demonstrated the spatiotemporal features of student scores with the data on campus activities and consumption. Zhou and Xiao [14] predicted the information that interests college students according to the inspection data on students. Li et al. [15] explored the scholarships and grants of colleges against the data on campus information system. Drawing on some ideas and methods in these studies, this paper discusses the factors affecting the scores of college students. The relevant trajectory data record the body information of students, and also reflect their habits and opinions to a certain extent.

Some scholars evaluated the similarities between participants through the mining of global positioning system (GPS) trajectories [16]. Considering the difficulty in acquiring the GPS information of college students, this paper creates an activity sequence for each student from the campus behavior data. The two focuses of our research include: identifying the factors affecting student scores based on behavior data, and cluster analysis, which has been applied in many other fields.

The distribution of feature weights is very important in cluster analysis, because only a few features are truly valuable. Chen et al. [17] proposed the W-k-means algorithm, without sacrificing the efficiency of the K-means algorithm, under the inspiration of the fuzzy C-means algorithm [18]. Hung et al. [19] suggested improving the performance of the K-means algorithm by initializing the weights as per the coefficient of variation. In addition, retroactive learning is a good framework of machine learning and logical reasoning. It can effectively explain the observed facts and hypotheses based on the known background knowledge. On this basis, this paper decides to determine the feature weights of the clustering algorithm from both objective and subjective aspects.

## 3. SPATIOTEMPORAL REPRESENTATION LEARNING FRAMEWORK

### 3.1 Problem definition

When a college student makes a payment with his/her campus card, his/her personal information is recorded. If he/she uses his/her identity to access the Internet or rent a house, his/her personal information will be added to some data records. These records will continue to increase with the elapse of time. The recorded data can accurately reflect the daily behaviors of the college student.

The scores of college students are closely correlated with self-discipline and time management. Good students generally have different behavior patterns from average students [20]. Therefore, the scores of college students can be reflected in daily life activities, such as smart card usage, Internet access, and cellphone/computer interactions.

### 3.2 Preparatory work

The functional areas are distributed in different places on campus. For example, the dormitories might scatter across the college, and have different names. Let  $S_\gamma$  be the campus of the target college;  $F_\eta$  be the functional areas on campus, including

dormitories, canteens, classrooms, libraries, yards, bathrooms, etc.;  $(p, lo, p, fu)$  be the  $\lambda$ -th position  $p(\gamma, \lambda)$  on campus, where  $p, lo$  is the position of physical position  $p$ ,  $p, fu$  is the functional area of  $p$ ;  $c(\cdot)$  be the function to transform  $p, lo$  into functional area  $p, fu$ . The time and position information are primarily extracted from consumption data, network service data, and access control data.

The activity of a student A can be labeled by his/her time tag  $t_A$ , functional area  $p \in F_\eta$ , activity area  $b \in B_\phi$ , as well as other relevant attributes. The activity sequence of A can be vectorized as  $b$ . In addition, the consumption behavior consists of such attributes as consumption amount, and card balance. A charging behavior at least covers information like "maximum amount". After being extracted from the consumption data, network service data, and access control data, the time and position of each student in every activity can be sorted in that period, without changing the spatiotemporal sequence of students.

The activity sequence of student A contains both spatial order and time order. Here, the activity sequence is defined as  $Aseq = \{(t_1, q_1), \dots, (t_k, q_k)\}$ , where  $t_i < t_j (i < j)$  is time,  $q_i \in q_\phi$  is the  $i$ -th item in the sequence, and  $q_i$  could be equal to  $q_j$ .

For the given activity sequence  $Aseq = \{(t_1, p_1), \dots, (t_k, p_k)\}$  and parameter  $\xi$ , if  $(t_i, P_i)$  and  $(t_{i+1}, P_{i+1})$  satisfy  $P_i = P_{i+1}$  and if  $|t_i - t_{i+1}| < \xi$ , then  $(t_i, p_i)$  and  $(t_{i+1}, p_{i+1})$  belong to the same position.

The trajectory  $Tra$  of student A is a subset of  $Aseq$ :  $Tra \in Aseq$ .

Using the node information in behavioral data, the original data were converted to semantic trajectories. Following the time-based trajectory segmentation method, this paper correctly extracts the trajectory from the activities of each student on a day [21-23]. Note that the day refers to the "day" for the student to complete a set of activities, rather than a natural day (from 00:00 to 00:00). For example, the "daily" trajectories of a student could cover  $\{(10pm, November 3, classroom)-(11pm, November 3, bedroom) \dots (0am, November 4, bedroom)\}$ . Although the trajectories cover two days, all the activities are clearly completed within the same day. Thus, the daily journey of each student can be defined as follows.

Here, the behavior trajectory is defined as  $Tra = \{(t_1, q_1), \dots, (t_k, q_k)\}$ , and the daily behavior trajectories as  $DTra = \{(t_1, q_1), \dots, (t_s, q_s)\}$ . The daily trajectories are organized into a trajectory set ( $DTra \in Tra$ ). In the  $Tra$ , there must exist a unique  $(t_j, q_j)$  such that  $(t_1, q_1) = (t_j, q_j)$ , and  $\forall n \in \{1, 2, \dots, S-1\}, (t_{1+n}, q_{1+n}) = (t_{j+n}, q_{j+n})$ .

The same activity recorded by the campus information system may contain multiple data that needs to be checked. The trajectory of such an activity can be expressed as  $TraP = q_1 \xrightarrow{\Delta t_1'} q_2 \xrightarrow{\Delta t_2'} \dots \xrightarrow{\Delta t_{u-1}'} q_u'$ , where  $q_i' \in \{q_j\}$ . If  $(t_j, q_j)$  corresponds to  $(t_i, q_i)$ , and  $q_{i+1}'$  corresponds to  $(t_j, q_j)$ , then  $\Delta t_i' = t_j - t_i$ .

### 3.3 Cluster analysis

For the given  $\partial = \{\partial_1, \partial_2, \dots, \partial_N\}$  and  $\beta = \{\beta_1, \beta_2, \dots, \beta_K\}$ , if  $\partial$  is the  $n$ -th instance of  $\partial_n$ , then  $\beta_k$  is the  $k$ -th cluster in  $\beta$ . Then,  $\partial_n = \{\partial_{n1}, \partial_{n2}, \dots, \partial_{nM}\}$ , where  $\partial_{nm}$  is the  $m$ -th eigenvalue. Moreover,  $\beta_1 \cup \beta_2 \cup \dots \cup \beta_K = \partial$ , and  $\beta_1 \cap \beta_2 \cap \dots \cap \beta_K = \phi$ .

For the given  $\partial = \{\partial_1, \partial_2, \dots, \partial_{IV}\}$  and  $\beta = \{\beta_1, \beta_2, \dots, \beta_K\}$ , if  $S_\partial$  is the  $n$ -th  $\partial$ , then  $\beta_k$  is the  $k$ -th cluster in  $\beta$ .  $\beta_k = \{\beta_{k1}, \beta_{k2}, \dots, \beta_{kM}\}$ ,  $k=1, 2, \dots, K$ .

The K-means algorithm aims to maximize the intra-cluster similarity and minimize inter-cluster similarity. Therefore, the

constraint function is the sum of the distance between the intra-cluster instances and the cluster head:

$$P(\mathbf{U}, \beta) = \sum_{k=1}^K \sum_{n=1}^N u_{nk} \sum_{m=1}^M d(x_{nm}, \beta_{km}) \quad (1)$$

where,  $U$  is an  $N \times K$  matrix;  $u_{nk} \in \{0, 1\}$  (If  $u_{nk}=1$ , then the  $n$ -th instance belongs to the  $k$ -th cluster);  $\sum_{k=1}^K u_{nk} = 1, n = 1, 2, \dots, N$ ;  $\beta$  consists of  $K$  clusters, i.e.,  $\beta = \{\beta_1, \beta_2, \dots, \beta_k\}$ ;  $d(x_{nm}, \beta_{km})$  is the distance between an intra-cluster instance and cluster head. The constraint function can be rewritten by Euclidean distance:

$$P(\mathbf{U}, \beta) = \sum_{k=1}^K \sum_{n=1}^N u_{nk} \sum_{m=1}^M (x_{nm} - \beta_{km})^2 \quad (2)$$

Obviously, each function is equal in formula (2). In fact, only some special functions are useful. Thus, functional selection was adopted to remove redundant, noisy functions [24]. In addition, feature weighting is a generalization technology, which assigns a weight to each feature within the range of  $[0, 1]$  rather than delete that feature [25]. Therefore, some scholars held that, before applying the K-means algorithm, each feature should be given a weighted tag, and the weights of  $M$  features should be defined as  $W=(\omega_1, \omega_2, \dots, \omega_M)$ . Then, formula (2) can be rewritten as:

$$P(\mathbf{U}, \beta) = \sum_{k=1}^K \sum_{n=1}^N u_{nk} \sum_{m=1}^M w_m^\beta (x_{nm} - \beta_{km})^2 \quad (3)$$

where,  $W \in [0, 1]$ ;  $\sum_{m=1}^M \omega_m = 1$ ;  $\beta$  is a user defined parameter. Under the constraint that the sum of all weights equals 1, the feasible approach is to express the minimum value of formula (3) as:

$$w_m = \frac{1}{\sum_{i \in F} [D_m / D_i]^{1/(\beta-1)}} \quad (4)$$

where,  $D_m = \sum_{k=1}^K \sum_{n=1}^N u_{nk} (x_{nm} - c_{km})^2$  is the sum of variances of each cluster.

By modifying the basic K-means algorithm, this paper solves the weight difference between features. Drawing on subjective weight  $V$  and objective weight  $W$ , an integrated weight can be defined as  $\gamma=(\gamma_1, \gamma_2, \dots, \gamma_M)$  with  $\gamma_m = \frac{w_m v_m}{\sum_{m=1}^M w_m v_m}$ . Whereas  $A=(a_1, a_2, \dots, a_n)$  and  $B=(b_1, b_2, \dots, b_m), n \leq m$ , if there exists a sequence satisfying  $1 \leq j_1 \leq j_2 \leq \dots \leq j_n \leq m, a_1 \in b_{j_1}, a_2 \in b_{j_2}$  then,  $A$  is the subsequence of  $B$ , and  $B$  is the hyper sequence of  $A$  [26]. The subsequences satisfy the following assumptions: (1) the items in every subsequence exist in the hyper sequence; (2) the items in every subsequence follow the same order as they are in the hyper sequence. For example,  $S=\langle a(abc)(ac)d(cf) \rangle, \langle a(abc) \rangle$  is a subsequence of  $S$ . For sequences  $A=(a_1, a_2, \dots, a_n)$  and  $B=(b_1, b_2, \dots, b_m), n \leq m$ , if  $a_1=b_1, a_2=b_2, \dots, a_{n-1}=b_{n-1}$  and  $a_n \subseteq b_n$ , then  $A$  is the prefix of  $B$ .

## 4. 3D CNN

### 4.1 Edge extraction

Boundary extraction aims to detect the contour of objects in

the target image, which usually exists at places with high grayscale changes. Edge detection algorithms are often adopted to preserve the contour, while hiding the pixels. This paper chooses the Canny edge detection algorithm, in view of its good effect on edge extraction [27]. The specific steps of this algorithm are as follows:

First, a two-dimensional (2D) Gaussian function [28] was adopted to filter and denoise the image pixel by pixel:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (5)$$

where,  $x$  and  $y$  are the abscissa and ordinate, respectively;  $\sigma$  is the standard deviation.

In fact, Gaussian convolution was adopted to verify the convolutional effect of the image. As shown in Figure 2, the Gaussian convolution kernel is a regularized kernel with the size of  $3 \times 3$  and the variance of  $\sigma=1.5$ :

$$G = \begin{bmatrix} 0.0947 & 0.1183 & 0.0947 \\ 0.1183 & 0.1477 & 0.1183 \\ 0.0947 & 0.1183 & 0.0947 \end{bmatrix} \quad (6)$$

The Gaussian convolution kernel was adopted to convolute the image to remove noise points and smooth the image.

The Sobel operator in formulas (3) and (4) was used to calculate the image gradient during the convolution. It can be divided into a horizontal operator  $S_x$  for calculating horizontal gradient and a vertical operator  $S_y$  for calculating vertical gradient:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (7)$$

$$S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (8)$$

The image gradient can be expressed as:

$$d_x = f(x, y) \cdot S_x \quad (9)$$

$$d_y = f(x, y) \cdot S_y \quad (10)$$

Then, the amplitude  $M$  and direction  $\theta_M$  of image gradient can be respectively computed by:

$$M = \sqrt{d_x^2 + d_y^2} \quad (11)$$

$$\theta_M = \arctan(d_y / d_x) \quad (12)$$

Through the above calculation, two images were obtained, including a gradient map and an angle map. Then, the gradient image was binarized by non-maximum suppression (NMS) [29, 30], which finds the local extreme edges and labels edges/non-edges with 1/0. The binarization could reveal the edge of a single pixel. The non-maximum points in the

gradient direction were filtered to refine the edges.

## 4.2 Model structure

2D CNN has been successfully applied to the detection of abnormal behaviors in videos [31], and the classification of images. However, 2D CNN cannot be directly implemented to classify video frames. Because the classroom surveillance video contains time information, every action by the students has its own trajectory. Therefore, this paper expands 2D kernel with 3D CNN into a 3D kernel: the kernel size was changed from wavelength  $\times$  W into wavelength  $\times$  D, with D being the number of continuous frames in the 3D kernel. The 3D kernel

can effectively capture the time information in video.

Figure 1 compares the 2D convolution with 3D convolution. The left subgraph illustrates how 2D convolution processes an image; the right subgraph explains the 3D convolution of video frames, which generates an additional dimension to save the time information in video.

The C3D [32, 33] is an early 3D CNN that has been successfully applied in behavior recognition. This original model consists of seven parts. The first five parts mainly includes convolutional layers and pooling layers; the sixth part contains two fully-connected layers; the seventh part is a Softmax layer. The original C3D structure is shown in the upper subgraph of Figure 2.

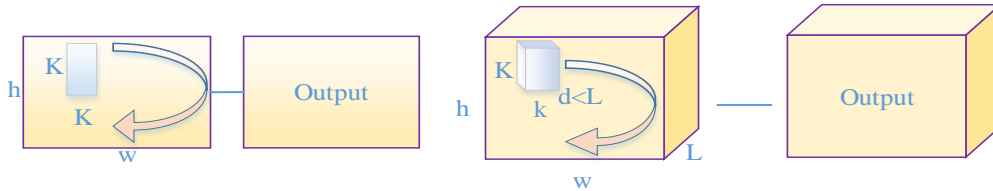


Figure 1. Comparison between 2D and 3D convolutions

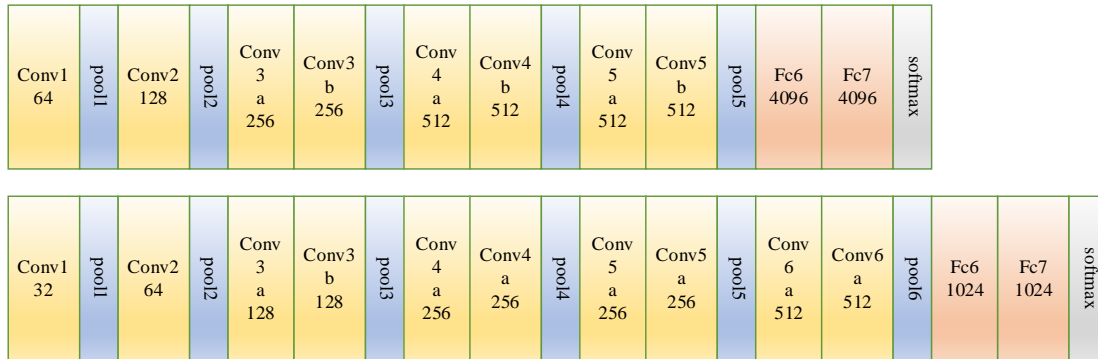


Figure 2. Structure of 3D CNNs

The C3D contains lots of parameters, as it was designed to classify a massive number of videos. The parameters mainly exist in the form of 3D kernels, with many channels in each layer. As shown in Figure 2, 512 channels can be outputted by convolutional layers 4 and 5. Because abnormal behaviors do not need to be recognized in convolutions, the number of convolution kernels in each layer can be reduced by half. Experimental results show that the C3D could shorten more than half of the training time and the reasoning time. The reduction of kernel number did not lower the precision, but slightly improves the accuracy. A possible reason is that the original model is so large as to cause overfitting. To further improve the recognition accuracy, this paper adds two similar convolutional layers and a pooling layer to the C3D. The improved C3D for abnormal pose detection is presented in the lower subgraph of Figure 2.

## 5. EXPERIMENTAL VERIFICATION

### 5.1 Experimental environment

The experimental environment is as follows: the central processing unit (CPU) is Intel hexacore i7-8750h; the graphics card is Gigabyte GeForce RTX 3080 EAGLE OC 10G; the memory is 16G; the operating system is Ubuntu 18.04; the

programming language is Python; the deep learning framework is TensorFlow 2.0 (GPU version).

### 5.2 Dataset

Our dataset comes from the opensource code dataset of Sun et al. [34]. It covers the student behaviors identified, detected, and labeled in class. In total, there are videos on 11 subjects taught in 128 classes. The dataset was split into a detection dataset, two recognition datasets, and a title dataset. The detection dataset contains 4,542 samples for time detection and 3,343 samples for action detection; the two recognition datasets respectively contain 4,276 samples and 4,296 samples. The datasets are representative and authentic, because all the actions in them are spontaneous actions by students in actual classrooms. The datasets were uploaded to <https://github.com/BNU-Wu/Student-Class-Behavior-Dataset>.

### 5.3 Training results analysis

#### (1) Video segmentation

Each video was divided into several frames. Following the edge extraction method in Section 2, the single-pixel, 3-pixel, and 5-pixel edges were extracted from the video frames. Then, the data were split into a training set and a test set by the ratio of 7:3. Our model was trained on the training set and evaluated

on the test set.

(2) Model training

In each iteration, 8 samples were randomly selected, and 16 continuous frames were extracted by random from each sample. Then, the spatiotemporal data acquired by KMC from the students were fused by the scheme of Lu [6], Wu et al. [13]. To evaluate the recognition effect, each model was trained and evaluated by the training and test sets. The improved C3D model was compared with the original C3D in terms of accuracy. Figure 3 compares the two models with 3-pixel edges as the input. It can be seen that the improved C3D is much more accurate than the original model; through the improvement, the number of parameters was reduced by half, and the training speed and reasoning speed were almost doubled.

(3) Model comparison with different inputs

Next, the generalization ability of our model was compared with four different inputs: original video frames (Figure 4(a)), single-pixel edges (Figure 4(b)), 3-pixel edges (Figure 4(c)), and 5-pixel edges (Figure 4(d)). It is generally believed that the model generalization ability is positively correlated with the precision of the training set and the size of the test set.

As shown in Figure 4, our model reached an accuracy as high as 95% on the training set, with the inputs of original video frames and single-pixel edges, but its accuracy on the test set with these inputs was 10% lower. Therefore, the two types of inputs are not good for the generalization ability of our model. When 3-pixel edges were inputted, the accuracy on test set gradually approximated that on training set, with the growing training steps, after 2,400 iterations, the accuracy on test set rose to around 93%.

Finally, the model accuracies on four different inputs were compared on the test set. It can be seen from Figure 5 that, after 800 training steps, the accuracy with the input of 3-pixel edges was much higher than that with another input; after 2,400 training steps, the accuracy with the input of 3-pixel edges increased to 93%, way higher than that (85%) with other types of inputs. Hence, 3-pixl edges are the best input features for video behavior recognition. In Figure 5, the four curves had basically the same trend. The behavior recognition rates with edges as inputs were comparable and even better than the rate with original video frames as inputs. As a result, our assumption is verified that the key information of human behaviors in videos is contained on the body contour.

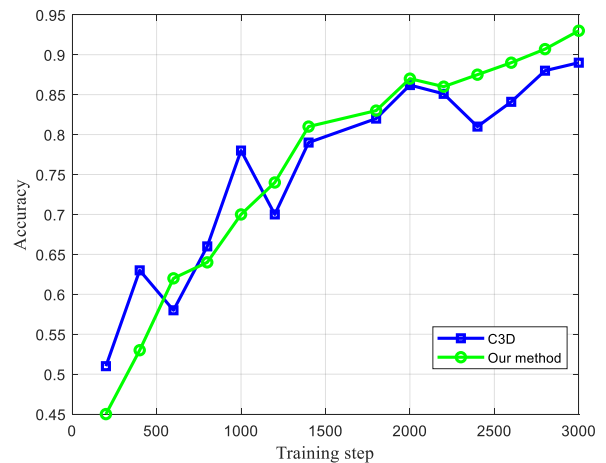


Figure 3. Accuracies of original and improved C3Ds

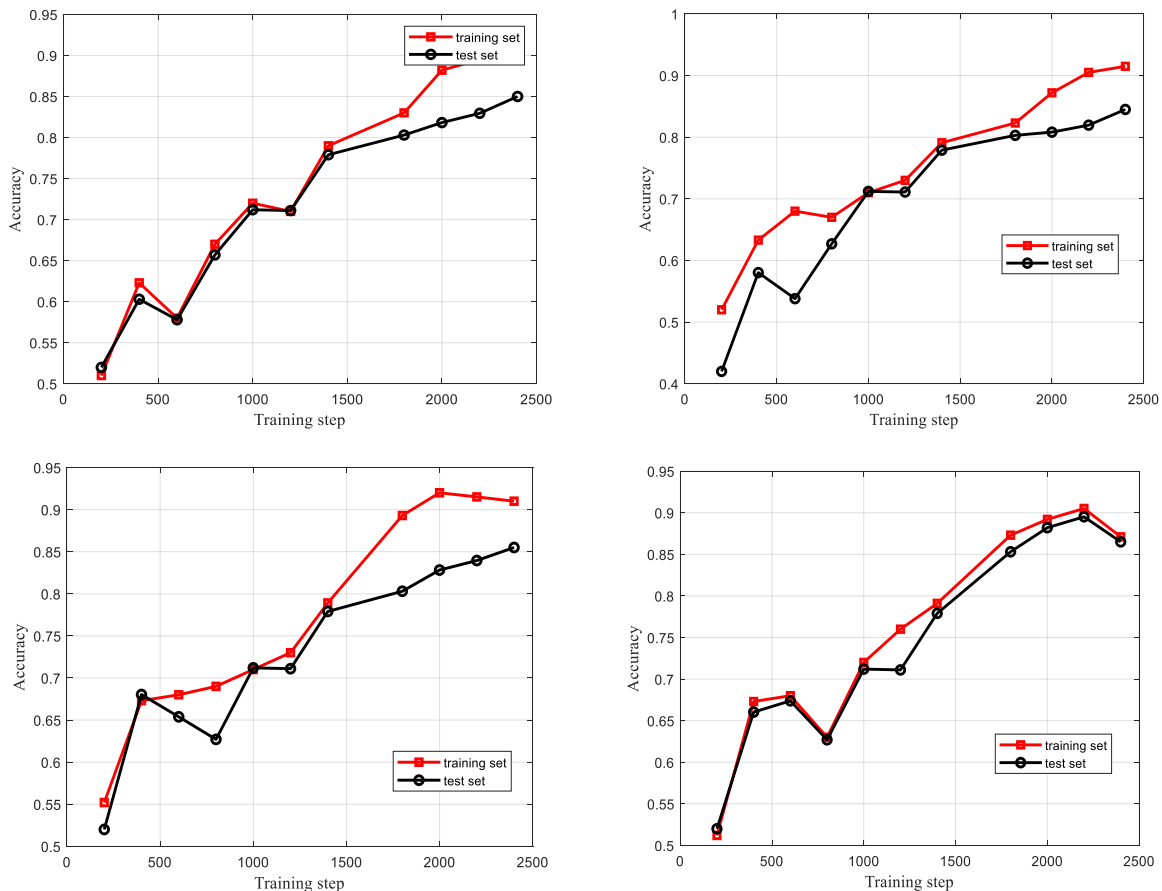


Figure 4. Comparison between accuracies on training set and test set





- [12] Buniyamin, N., Bin Mat, U., Arshad, P.M. (2015). Educational data mining for prediction and classification of engineering students achievement. In 2015 IEEE 7th International Conference on Engineering Education (ICEED), pp. 49-53. <https://doi.org/10.1109/ICEED.2015.7451491>
- [13] Wu, Y., Gong, R., Cao, Y., Wen, C., Teng, Z., Pu, J. (2016). EduCircle: Visualizing spatial temporal features of student performance from campus activity and consumption data. In International Conference on Cooperative Design, Visualization and Engineering, pp. 313-321. [https://doi.org/10.1007/978-3-319-46771-9\\_41](https://doi.org/10.1007/978-3-319-46771-9_41)
- [14] Zhou, Q., Xiao, Y. (2018). Analysis and Precaution of College Students' Academic Record based on Data Mining Technology. China Educational Technology & Equipment, 06.
- [15] Li, F.L., Xiang, H., Liu, X. (2015). Effects of Scholarships and Grants on College Students' Academic Achievement: Taking Tsinghua University as an Example. Tsinghua Journal of Education.
- [16] Zhao, J., Wang, H., Tang, B., Yang, Z. (2016). Research of GPS track generation and location based on mobile robot. Industry and Mine Automation, 01.
- [17] Chen, G.P., Wang, W.P., Huang, J. (2012). Improved initial clustering center selection method for K-means algorithm. Journal of Chinese Computer Systems, 33(6): 1320-1323.
- [18] Bouyer, A., Hatamlou, A., Masdari, M. (2015). A new approach for decreasing energy in wireless sensor networks with hybrid LEACH protocol and fuzzy C-means algorithm. International Journal of Communication Networks and Distributed Systems, 14(4): 400-412. <https://doi.org/10.1504/IJCND.2015.069675>
- [19] Hung, W.L., Chang, Y.C., Lee, E.S. (2011). Weight selection in WK-means algorithm with an application in color image segmentation. Computers & Mathematics with Applications, 62(2): 668-676. <https://doi.org/10.1016/j.camwa.2011.05.048>
- [20] Kickert, R., Meeuwisse, M., Stegers-Jager, K.M., Koppenol-Gonzalez, G.V., Arends, L.R., Prinzie, P. (2019). Assessment policies and academic performance within a single course: the role of motivation and self-regulation. Assessment & Evaluation in Higher Education, 44(8): 1177-1190. <https://doi.org/10.1080/02602938.2019.1580674>
- [21] Ham, S., Lee, G. (2017). Time-based joining method for generating phylogenetic trees of architectural plans. Journal of Computing in Civil Engineering, 31(2): 04016055. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000626](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000626)
- [22] Sasubilli, S.M., Kumar, A., Dutt, V. (2020). Machine learning implementation on medical domain to identify disease insights using TMS. In 2020 International Conference on Advances in Computing and Communication Engineering (ICACCE), pp. 1-4. <https://doi.org/10.1109/ICACCE49060.2020.9154960>
- [23] Ji, H.B, Wang, J.H. (2016). Research on Improved PrefixSpan Algorithm and its Application in Web User Behavior Patterns Mining. Computer Science.
- [24] Wang, Z. (2015). Removal of noise and radial lens distortion during calibration of computer vision systems. Optics Express, 23(9): 11341-11356. <https://doi.org/10.1364/OE.23.011341>
- [25] Alshomrani, S., Bawakid, A., Shim, S.O., Fernández, A., Herrera, F. (2015). A proposal for evolutionary fuzzy systems using feature weighting: dealing with overlapping in imbalanced datasets. Knowledge-Based Systems, 73: 1-17. <https://doi.org/10.1016/j.knosys.2014.09.002>
- [26] Yu, X. (2015). Hyper-sequence-graph mining. The University of Texas at San Antonio.
- [27] Nikolic, M., Tuba, E., Tuba, M. (2016). Edge detection in medical ultrasound images using adjusted Canny edge detection algorithm. In 2016 24th Telecommunications Forum (TELFOR), 1-4. <https://doi.org/10.1109/TELFOR.2016.7818878>
- [28] Chen, Z., Zhou, Z., Adnan, S. (2021). Joint low-rank prior and difference of Gaussian filter for magnetic resonance image denoising. Medical & Biological Engineering & Computing, 1-14. <https://doi.org/10.1007/s11517-020-02312-8>
- [29] Hou, Z.Q., Liu, X.Y., Yu, W.S., Ma, S.G. (2019). Improved algorithm of Faster R-CNN based on double threshold-non-maximum suppression. Opto-Electronic Engineering, 2019.
- [30] Forcado, M., Estrada, J.E. (2019). Model development of marble quality identification using thresholding, Sobel edge detection and Gabor Filter in a mobile platform. IEEE, 2019. <https://doi.org/10.1109/HNICEM.2018.8666359>
- [31] Nicolaou, M.A., Zafeiriou, S., Kotsia, I., Zhao, G. (2019). Editorial of Special Issue on Human Behaviour Analysis "In-the-Wild". IEEE Transactions on Affective Computing, 10(1): 4-6. <https://doi.org/10.1109/TAFFC.2019.2895141>
- [32] Li, Y., Miao, Q.G., Tian, K., Fan, Y.Y., Xu, X., Ma, Z.X., Song, J.F. (2017). Large-scale Gesture Recognition with a Fusion of RGB-D Data Based on Saliency Theory and C3D model. IEEE Transactions on Circuits and Systems for Video Technology, 119: 187-194. <https://doi.org/10.1016/j.patrec.2017.12.003>
- [33] Cecen, A, Dai, H.J., Yabansu, Y.C., Kalidindi, S.R., Song, L. (2017). Material structure-property linkages using three-dimensional convolutional neural networks. Acta Materialia, 146: 76-84. <https://doi.org/10.1016/j.actamat.2017.11.053>
- [34] Sun, B., Wu, Y., Zhao, K.J., Yu, L.J., Yan, H.Q., Luo, A. (2021). Student Class Behavior Dataset: a video dataset for recognizing, detecting, and captioning students' behaviors in classroom scenes. Neural Computing and Applications, 1-20. <https://doi.org/10.1007/s00521-020-05587-y>