Deep Learning-Based Method for Detecting Cheating Behaviors in Remote Examination Settings

Rongcong Chen, Yang Chen, Zuozheng Lian*, Guangda Zhang
College of Computer and Control Engineering, Qiqihar University, Qiqihar, Heilongjiang, China
**Corresponding Author*

Abstract: Examinations serve as a crucial means of assessing students' abilities and knowledge levels, necessitating supervision during the process to ensure fairness. However, most existing examinations rely on multiple invigilators stationed in the examination hall, supplemented by video surveillance and computer equipment, which demands substantial human resources and time costs while exhibiting low invigilation efficiency. To address this issue, this paper proposes a method for detecting cheating behaviors in examination halls, which primarily consists of two modules: face-recognition and Alpha Pose. The former module implements facial recognition to prevent impersonation, while the latter captures key point data and employs the ST-GCNs model to detect examinees' actions, thereby preventing cheating behaviors. This method can be deployed in examination halls equipped with video surveillance and computer devices, demonstrating high recognition rates for common cheating behaviors and effectively reducing the human resources and time costs associated with examinations.

Keywords: Image Processing; Facial Recognition and Detection; Behavior Recognition; Cheating Action Detection

1. Introduction

To ensure fairness of examinations, invigilators must implement supervision measures. Currently, most examination venues employ a monitoring system that combines video-surveillance equipment with computers. However, such systems typically only capture examination room footage and store invigilation videos, requiring manual screening and identification of examinee behavior, which consumes substantial human resources and time and results in low monitoring efficiency. Against this backdrop, this paper presents an examination cheating detection system based on face recognition and AlphaPose [1], which automatically identifies cheating behaviors during exams, thereby improving monitoring efficiency.

2. Related Research

Related research has been proposed for detecting cheating behaviors in examination halls. For instance, Liu et al. [2] introduced a cheating detection method based on gaze estimation that predicts the examinee's gaze direction while integrating face recognition and object detection technologies to calculate the gaze angle and fixation distance for cheating detection. Li [3] proposed a method for detecting abnormal behaviors in examination halls based on head motion analysis. This method employs a 2D motion estimator and a 3D head motion estimator, utilizing algorithms to automatically recognize faces and locate facial features for head pose estimation, followed by an abnormal behavior analysis based on pose parameters. However, these two methods are limited to examinees' head movements and may fail to achieve the desired results when examinees employ other cheating methods. Lin et al. [4] presented a cheating detection method based on background subtraction, which uses an iterative thresholding approach to determine dynamic thresholds, segment the different image, and then update the background based on the segmented image for cheating detection using background subtraction algorithms. Nevertheless, this method struggles to achieve satisfactory detection performance under conditions such as sudden weather changes, variations in lighting equipment causing fluctuations in light sources and brightness, or disturbances in the background owing to wind or camera vibrations. This study utilizes video surveillance equipment to capture examination room footage and employs the multi-person pose estimation system Alpha Pose to identify the upper-body postures of examinees above their desks. The detected key point data were transmitted to a model constructed using the ST-GCN network [5] to analyze examinees' postures during normal examination conditions and potential cheating behaviors, thereby determining whether cheating occurred. Additionally, this study incorporated facial recognition functionality developed using Python's face-recognition library to verify examine identities and prevent impersonation cheating.

3. Design of Examination Cheating Detection Methods

Design of Examination Cheating Detection Methods The method proposed in this paper primarily achieves two functions: a visualization interface to provide information for users and cheating detection, which includes examinee face recognition to determine impersonation and abnormal behavior recognition to identify cheating during exams. Based on this method, a monitoring system was designed using the design framework illustrated in Figure 1. The system consists of two modules: a user interface and cheating detection. The former is implemented using PyQt and primarily manages system users, including functions such as creating, deleting, and modifying data for regular users and administrators. The latter comprises two functionalities: face recognition to prevent impersonation, and abnormal behavior recognition to deter misconduct. These functionalities are realized through impersonation detection and cheating-action inspection methods. Design of Examination Cheating Detection Methods The method proposed in this paper primarily achieves two functions: a visualization interface to provide information for users and cheating detection, which includes examinee face recognition to determine impersonation and abnormal behavior recognition to identify cheating during exams. Based on this method, a monitoring system was designed using the design framework illustrated in Figure 1. The system consists of two modules: a user interface and cheating detection. The former is implemented using PyQt and primarily manages system users, including functions such as creating, deleting, and modifying data for regular users and administrators. The latter comprises two functionalities: face recognition to prevent impersonation, and abnormal behavior recognition to deter misconduct. These functionalities are realized through impersonation detection and cheating-action inspection methods.

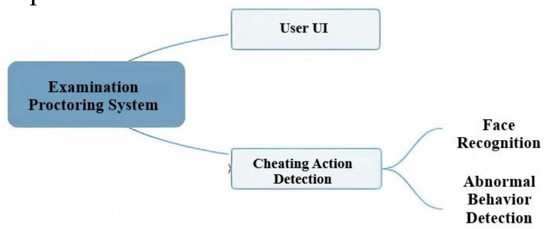

Figure 1. Schematic Diagram of the System Architecture

4. Design of the Test-Taking Detection Method

The Python face recognition library is an open-source tool designed for facial recognition and feature extraction. This library employs deep learning algorithms to efficiently perform functions, such as face detection, facial landmark detection, face alignment, and facial recognition. It supports multiple algorithms for face detection, including HOG and CNN, and primarily utilizes the Histogram of Oriented Gradient (HOG) model [6]. HOG is widely applied in computer vision and image processing to characterize the features for object detection [7]. The impersonation detection method proposed in this study consists of four main steps, as illustrated in Figure 2.

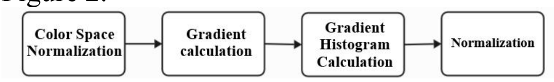

Figure 2. Flowchart of impersonation detection method

4.1 Color Space Normalization

Since images can be affected by factors such as equipment and environmental conditions, preprocessing is required. Preprocessing includes grayscale conversion and Gamma normalization. Image grayscale conversion reduces the color space of the image, thereby improving processing efficiency. Gamma normalization is employed to address uneven brightness in images. After processing, the grayscale value G and output value μ of the pixels can be obtained, as shown in Equations (1)-(2).

$$G = 0.3R + 0.59G + 0.11B \quad (1)$$

$$V_{out} = AV_{in}^{\gamma} \quad (2)$$

In grayscale processing, R, G, and B represent the luminance of the respective color channels for the pixel. In Gamma correction, is a constant, denotes the input value, and stands for the Gamma value.

4.2 Gradient Calculation

The gradient is calculated according to Equations (3)-(6).

$$G_x(x, y) = I(x+1, y) - I(x-1, y) \quad (3)$$

$$G_y(x, y) = I(x, y+1) - I(x, y-1) \quad (4)$$

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (5)$$

$$\theta(x, y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \quad (6)$$

Where and are the horizontal and vertical gradient values of the image at pixel, respectively. represents the gradient magnitude at pixel, and denotes the orientation of pixel undefined.

4.3 Gradient Histogram Computation Based on Cell Units

After gradient computation, the image is divided into multiple equally sized cell units (as shown in Figure 3), with each cell measuring 8*8 pixels. The gradient orientation information of pixels within each cell unit is binned into 9 corresponding orientation intervals and accumulated into the histogram.

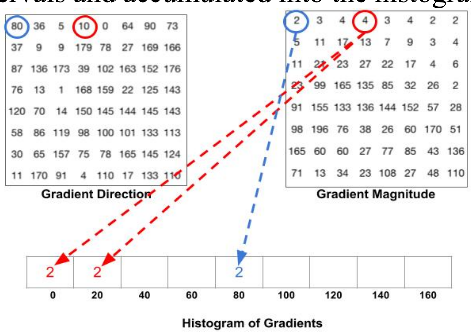

Figure 3. Interval Histogram Calculation

4.4 Normalization

After computing the gradient magnitude and orientation, the histogram is statistically calculated for the detection window using the Cell as the smallest unit. Multiple Cells are then combined into a Block to form a feature vector (typically composed of 2×2 Cells). The feature vector is normalized using either the L1 or L2 norm to obtain the HOG features. The normalization formulas for the L1 and L2 norms are shown in Equation (7).

$$\widehat{v_1} = \frac{v}{\sqrt{||v||_1 + \varepsilon}}, \quad \widehat{v}_2 = \frac{v}{\sqrt{||v||_2^{\,2} + \varepsilon}} \quad (7)$$

5. Design of the Cheating Action Detection Method

The method first captures video frames of the examination room, then utilizes Alpha Pose to obtain key point data of examinees in the current scene. Finally, the data is transmitted to a convolutional neural network model for processing to determine whether examinees exhibit cheating behaviors.

5.1 Acquisition of Candidate Pose Key Point Data Using the Alpha Pose Algorithm

Alpha Pose is a multi-person pose estimator based on the deep learning library PyTorch. It is the first open-source system to achieve over 70 mAP (75 mAP) on the COCO dataset [8] and surpasses 80 mAP (82.1 mAP) on the MPII dataset [9]. AlphaPose supports multiple output formats, and this system primarily utilizes its key point data output after human pose estimation [10].

5.2 Cheating Action Recognition

This study employs the Spatial Temporal Graph Convolutional Networks (ST-GCNs) model to obtain the probabilities of examinees exhibiting normal behavior, passing items, looking around, or peeking with their heads down. The max function is utilized to determine the examinee's current state. The processing pipeline of this model is illustrated in Figure 4.

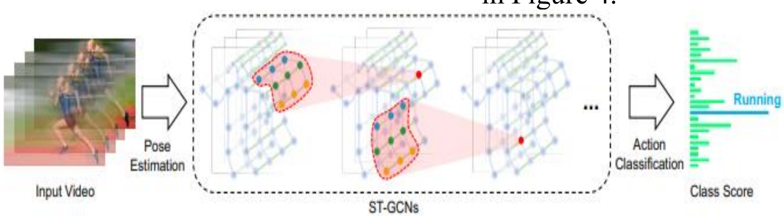

Figure. 4 Flowchart of ST-GCNs

The ST-GCNs model was trained using the NTU RGB+D 60 dataset. The NTU RGB+D 60 dataset consists of 56,880 action samples, encompassing RGB videos, depth map sequences, 3D skeleton data, and infrared videos for each sample. The dataset was simultaneously captured by three Microsoft Kinect v.2 cameras, with RGB videos at a resolution of 1920 × 1080, depth maps and infrared videos at 512 × 424, and 3D skeleton data containing the 3D positions of 25 body joints per frame. The dataset includes 60 different action categories, covering daily activities, interactive actions, and health-related movements. The training results of ST-GCNs are shown in Figure 5.

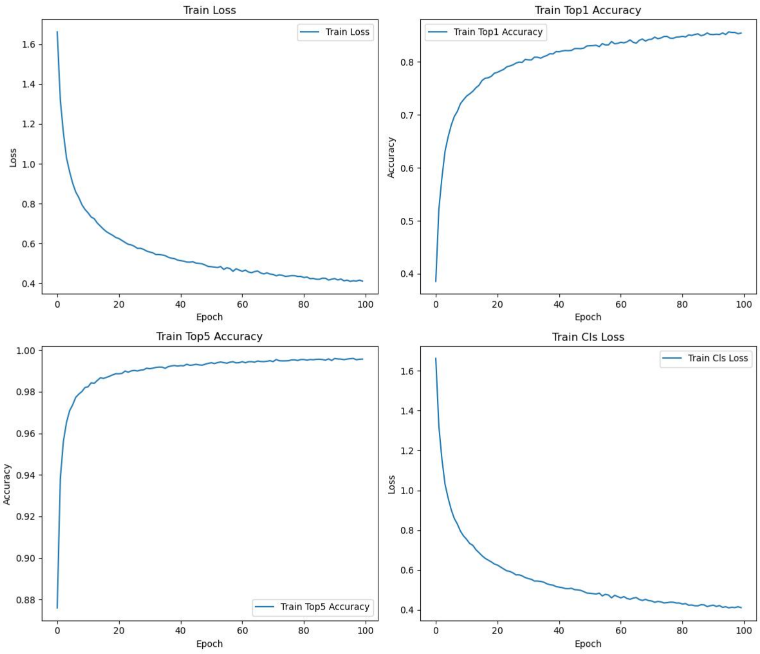

Figure 5. Training results of ST-GCNs

As shown in Figure 6, the loss value gradually decreases with increasing iteration numbers, indicating an improvement in the model's predictive performance. The training loss can be observed to decline from approximately 1.6 to nearly 0.4 within 100 iterations. Both Top1 and Top5 accuracy rates progressively increase with iterations, demonstrating enhanced classification accuracy of the model. The training classification loss (Train Cls Loss) approaches around 0.4 after approximately 100 iterations.

6. Experimental Testing

This section evaluates the feasibility and effectiveness of the testing system, with the main contents including Examination environment setup, describing the physical configuration of the experimental environment; Implementation of the visual interface, detailing the user interaction interface developed based on the PyQt framework, which demonstrates system login, cheating detection, and facial information entry functions. Facial recognition testing: By registering and recognizing examinees' faces, the impersonation detection capability of the face-recognition library is verified. Action recognition testing (4.4): The Alpha Pose algorithm is employed to obtain examinees' pose key point data, combined with ST-GCN model analysis, demonstrating the system's recognition performance for four typical cheating behaviors (normal behavior, passing items, looking around, and peeking down).

6.1 Examination Room Environment Setup

When utilizing this system for examination hall surveillance, a monitoring device should be positioned at the front of the examination hall. The device can be placed on the front desk, with its monitoring range required to cover all examinees, as illustrated in Figure 6.

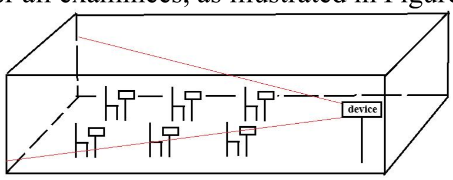

Figure 6. Schematic Diagram of the Examination Room Environment

6.2 Visualization Interface Implementation Using PyQt

The system provides an interactive interface for user engagement, which dynamically displays different widget controls based on user interactions. Figure 7 illustrates the runtime interface of the system, where the top-left section shows the login interface, the top-right section presents the cheating detection interface, the bottom-left section displays the facial information input interface, and the bottom-right section features the facial recognition verification interface.

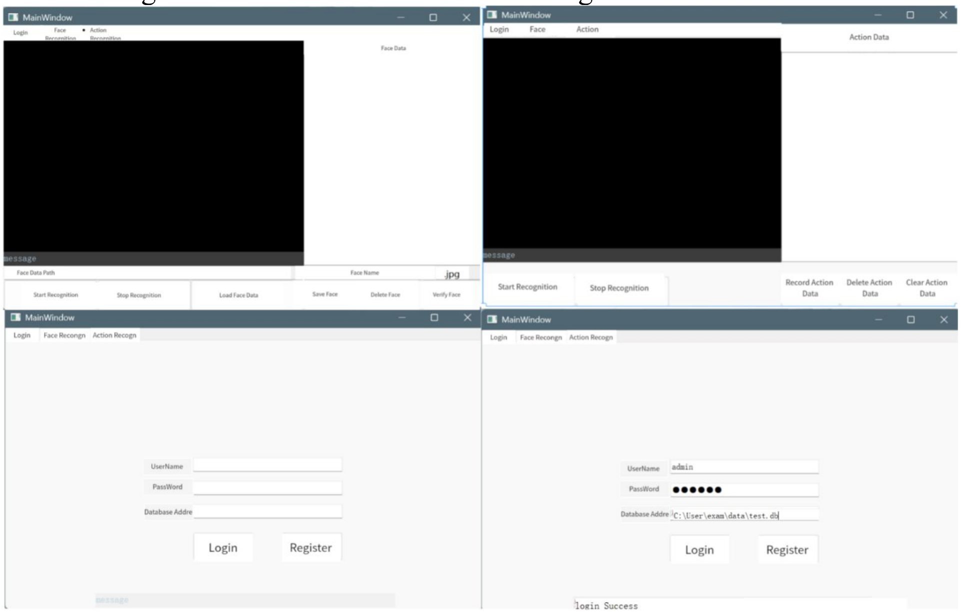

Figure 7. System Operation Interface

7. Conclusion

This paper presents a proctoring system implemented using deep learning techniques. The system primarily features a visual interface and cheating detection capabilities, which consist of facial recognition and abnormal behavior identification to detect impersonation and common cheating behaviors in examination settings. By leveraging hardware resources, this system can assist in examination supervision, reduce human and time costs associated with proctoring, and enhance monitoring efficiency.

Acknowledgments

This paper is supported by National College Students' Innovation and Entrepreneurship Training Project Foundation of China (2022 10232102, 202410232047), Heilongjiang Provincial College Students' Innovation and Entrepreneurship Training Project Foundation of China (S202410232073).

References

[1] Huang et al. Fall detection model based on AlphaPose combined with LSTM and Lightgbm. Proceedings of SPIE, 2024, 13086(000):8. DOI:10.1117/12.2692742.

[2] Liu et al. A method for detecting cheating in examination rooms based on line of sight estimation. Journal of Taiyuan Normal University (Natural Science Edition), 2022, 21 (03): 47-52+96

[3] Li. Abnormal behavior detection in examination rooms based on head motion analysis. Changchun University of Science and Technology, 2018.

[4] Lin et al. A method for detecting cheating behavior in examination rooms based on background subtraction. Journal of Jinan University (Natural Science Edition), 2015, 29 (06): 406-409. DOI: 0.13349 /j.cnki. jdxbn. 2015.06.002.

[5] Yan et al. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. AAAI Conference on Artifi cial Intelligence (2018).

[6] N. Dalal & B. Triggs. Histograms of oriented gradients for human detection //2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1, DOI: 10.1109/CVPR.2005.177.

[7] Li et al. A review of human joint motion

detection and tracking. Mechanical, 2020,47 (08): 18-23.
[8] Singh S et al. Benchmarking Object Detectors with COCO: A New Path Forward // European Conference on Computer Vision. Springer, Cham, 2025.DOI:10.1007 /978-3-031-72784-16.
[9] Wei & Gao. Research on Video Surveillance Moving Target Detec tion Algorithm. Electronic Testing, 2016, (07): 53-54.
[10] Sheng & Wang. Research on Human Pose Recognition Based on Computer Vision. Modern Information Technology, 2022,6 (16): 87-91+95. DOI: 10.19850/j.cnki. 2096-4706.2022.16.023.

