Depth Camera-based Human Action Recognition with Object Detection and Graph Convolutional Network

Qingqi ZHANG

The Graduate School of East Asian Studies

Yamaguchi University

A thesis submitted for the degree of

Doctor of Philosophy

October 2025

Abstract

Title of Thesis: Depth Camera-based Human Action Recognition with
Object Detection and Graph Convolutional Network

Name of Degree Candidate: Qingqi ZHANG
Asian Education Course
The Graduate School of East Asian Studies
Yamaguchi University, Japan

Degree and Year: Doctor of Philosophy, 2025

Dissertation directed by: Dr. Qi-Wei GE
Professor
The Graduate School of East Asian Studies
Yamaguchi University, Japan

Human Action Recognition (HAR) is a challenging and engaging research area in computer vision with diverse applications, including smart surveillance, robotics, industrial automation, healthcare, and education. Traditional methods relying on RGB video data often struggle to handle environmental noise, such as variations in lighting, viewpoint, background colors, and clothing, limiting their effectiveness in real-world scenarios. The rapid development of depth camera technology has opened up new opportunities for action recognition by enabling the use of three-dimensional human joint coordinates, which are highly effective for accurately modeling human actions. Additionally, depth cameras provide robustness in complex environments and offer privacy-preserving capabilities, making them particularly suited for sensitive applications.

This thesis proposes a depth camera-based framework that integrates object detection and Graph Convolutional Network (GCN) to achieve

accurate and efficient HAR. First, the proposed framework employs an object detection algorithm to localize individuals in the scene and extract precise Regions of Interest (RoI). Next, depth camera data is utilized to extract skeleton data of individuals within the RoIs, providing robust input for spatiotemporal modeling. Finally, a GCN specifically designed for action classification is applied to analyze skeletal sequences and classify actions. By effectively combining multi-modal data sources, including RGB and skeletal data, the framework significantly improves recognition accuracy and robustness in complex scenarios.

As the first critical step in the proposed framework, object detection faces significant challenges, particularly the diversity of object scales and the overlapping of objects within complex scenes. To address the issue of scale diversity, this thesis proposes the Re-BiFPN feature fusion method, which incorporates a Coordinate Attention Atrous Spatial Pyramid Pooling (CA-ASPP) module and a recursive connection. The CA-ASPP module extracts direction-aware and position-aware information from feature maps, enhancing the representation of multi-scale objects. The recursive connection further refines the feature maps by feeding the processed multi-scale features back to the backbone network for additional feature extraction. Additionally, to improve localization accuracy and robustness in cluttered scenes, a Rep-CIoU loss function is designed to mitigate the impact of object overlap. To evaluate the effectiveness of the proposed methods, experiments were conducted on X-ray security inspection images, a highly challenging real-world application characterized by varying object scales and frequent object overlaps. The results demonstrate that the proposed methods significantly improve detection accuracy and robustness. Furthermore, to assess the generalizability of the methods, extensive experiments were also performed on the VOC and COCO datasets, which include human detection tasks. The experimental results show that the proposed methods achieve competitive performance, highlighting their broad applicability across diverse domains.

As another key step in the proposed framework, action classification is a challenging task, particularly when dealing with subtle motion patterns characterized by small amplitudes and prolonged durations. Two pivotal issues warrant further exploration: the development of enhanced temporal feature representations and the expansion of convolutional models'

capacity to capture long-range temporal dependencies. This thesis proposes a novel Skeleton Temporal Fusion Graph Convolutional Network (STF-GCN) for action classification, which effectively models advanced temporal feature representations. Specifically, the STF-GCN employs three encoding strategies to integrate two types of temporal feature representations. These strategies are designed to capture the intricacies of motion dynamics and the subtleties in action variations, enabling more accurate and robust recognition of complex actions. Furthermore, a Skeleton Temporal Fusion (STF) module is proposed to highlight temporal feature representations, employing a structure that alternates between large and small kernel convolutions to achieve diverse effective receptive fields. The integration of large kernel convolutions allows our model to perceive an expanded temporal context, significantly enhancing its ability to understand action dynamics in depth. To evaluate the effectiveness of the proposed methods, extensive experiments were conducted on both elderly action and general action datasets. The results demonstrate that the proposed methods achieve superior performance, not only in classifying elderly actions with subtle and prolonged motion patterns but also in general action classification tasks, highlighting its robustness and broad applicability.

Finally, this thesis explores the application of the proposed framework in two key areas: elderly care and intelligent education. In elderly care, fall detection serves as a case study. Fall detection is critical for enhancing caregiving safety and supporting the independence of elderly individuals. Experimental results demonstrate that the proposed framework achieves an impressive average accuracy of 96.3% in real-world scenarios, while maintaining low false positive and false negative rates. In intelligent education, hand-raising recognition serves as a case study. Recognizing hand-raising actions is essential for comprehensively evaluating student engagement and adapting teaching strategies accordingly. Experimental results show that the proposed framework achieves an average accuracy of 89.7% in real-world scenarios, also maintaining low false positive and false negative rates. These results demonstrate that the proposed depth camera-based action recognition framework is both accurate and practical for real-time applications in critical domains, and has strong potential for broader deployment in real-world scenarios.

学位論文内容要旨

学位論文題目：物体検出とグラフ畳み込みネットワークを用いた深度カメラベースの人間行動認識

指導教員：葛　崎偉　教授

申請者名：張　慶琪 (チョウ　ケイキ)
令和 4 年度入学
山口大学大学院東アジア研究科 (博士後期課程)
アジア教育開発コース

本論文は、深度カメラを用いた高精度かつ高効率な人物行動認識（HAR）フレームワークを提案する。このフレームワークは、まず物体検出アルゴリズムでシーン内の人物を特定し、正確な関心領域（RoI）を抽出する。次に、RoI 内の人物の骨格データを深度カメラデータから抽出し、これを時空間モデリングのためのロバストな入力として利用する。最後に、行動分類に特化して設計されたグラフ畳み込みネットワーク（GCN）を適用し、骨格シーケンスを分析して行動を分類する。RGB データと骨格データを含む複数のモダリティを効果的に組み合わせることで、本フレームワークは複雑なシナリオにおいても認識精度とロバスト性を大幅に向上させる。

提案フレームワークの最初の重要なステップである物体検出では、特に物体のスケール多様性や複雑なシーンにおける物体の重なりという課題に直面する。スケール多様性の問題に対処するため、本論文では、座標注意型 ASPP（CA-ASPP）モジュールと再帰接続を組み込んだ Re-BiFPN 特徴融合手法を提案する。CA-ASPP モジュールは、特徴マップから方向認識および位置認識情報を抽出し、マルチスケール物体の表現を強化する。再帰接続は、処理されたマルチスケール特徴をバックボーンネットワークにフィードバックしてさらなる特徴抽

出を行うことで、特徴マップをより洗練する。さらに、散らかったシーンでの局在化精度とロバスト性を向上させるため、物体の重なりの影響を軽減する Rep-CIoU 損失関数を新たに設計した。

提案手法の有効性を評価するため、物体のスケール変動や頻繁な重なりが顕著な X 線手荷物検査画像を用いた実験を行った。その結果、提案手法が検出精度とロバスト性を大幅に向上させることが示された。さらに、手法の汎化能力を評価するため、人物検出タスクを含む VOC および COCO データセットでも広範な実験を実施し、競争力のある性能を達成した。これにより、多様なドメインにおける広範な適用可能性が裏付けられる。

提案フレームワークのもう一つの重要なステップである行動分類は、特に振幅が小さく、持続時間が長い微妙な動作パターンを扱う場合に困難を伴う。これに対処するため、高度な時系列特徴表現の開発と、長距離の時系列依存性を捉える畳み込みモデルの能力の拡張という 2 つの重要な問題について、さらなる探求が必要となる。本論文は、行動分類のための新しい骨格時系列融合グラフ畳み込みネットワーク（STF-GCN）を提案し、高度な時系列特徴表現を効果的にモデリングする。STF-GCN は、3 つの符号化戦略を用いて 2 種類の時系列特徴表現を統合する。これらの戦略は、動作ダイナミクスの複雑さや行動のバリエーションにおける微妙な違いを捉えるように設計されており、複雑な行動をより正確かつロバスト的に認識することを可能にする。さらに、時系列特徴表現を強調するための骨格時系列融合（STF）モジュールを提案した。このモジュールは、大小のカーネル畳み込みを交互に用いる構造を採用することで多様な有効受容野を実現し、大規模カーネル畳み込みの統合により、モデルは拡張された時系列コンテキストを認識し、行動ダイナミクスを深く理解する能力が大幅に向上する。

提案手法の有効性を評価するため、高齢者の行動データセットと一般的な行動データセットの両方で広範な実験を行った。その結果、提案手法は、微妙で持続的な動作パターンを持つ高齢者の行動分類だけでなく、一般的な行動分類タスクにおいても優れた性能を達成し、そのロバスト性と幅広い適用可能性が強調された。

最後に、本論文は提案フレームワークの応用例として、高齢者介護とスマート教育の 2 つの主要な領域を探求する。高齢者介護では、転倒検出をケーススタディとして取り上げ、介護の安全性を高め、高齢者の自立を支援する上で極めて重要であることを示す。実験結果は、提案フレームワークが実世界シナリオにおいて平均精度 96.3% という目覚ましい成果を達成し、低い偽陽性率と偽陰性率を維持することを示している。スマート教育では、挙手認識をケーススタディとして取り上げ、学生の参加度を包括的に評価し、指導戦略を適応させる上で不可欠であることを示す。実験結果は、提案フレームワークが実世界シナリオにおいて平均精度 89.7% を達成し、同様に低い偽陽性率と偽陰性率を維持することを示している。これらの結果は、提案する深度カメラベースの行動認識フレームワークが、重要な領域におけるリアルタイムアプリケーションにおいて高精度かつ実用的であり、実世界シナリオでのより広範な展開に強い可能性を秘めていることを示している。

Depth Camera-based Human Action Recognition with Object Detection and Graph Convolutional Network

by

QINGQI ZHANG

Dissertation submitted to the Graduate School of East Asian Studies of Yamaguchi University,

in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Advisory Committee:

Professor Qi-Wei GE, Chairman/Advisor
Professor Mitsuru NAKATA
Professor Chisato KITAZAWA
Professor Shingo YAMAGUCHI

Acknowledgements

Looking back on the past three years of my doctoral journey in Japan, I am deeply grateful for the unique experiences that have enriched my life. Living in Japan has allowed me to immerse myself in its rich culture, from its traditions and values to its way of life. These experiences have broadened my horizons and left me with unforgettable memories. I am truly thankful for this opportunity, which has become an invaluable chapter in my life.

I would like to express my heartfelt gratitude to my supervisor, Qi-Wei GE, for his continuous guidance, encouragement, and wisdom throughout my research journey. His expertise, constructive feedback, and unwavering support have been instrumental in shaping this dissertation. Over the past three years, his rigorous work ethic and tireless dedication have deeply influenced me, inspiring me to adopt his exemplary approach to research and learning. His commitment to excellence has set a standard that I strive to follow.

I am very grateful to the other members of the examination committee for this dissertation: Professor Mitsuru NAKATA, Professor Chisato KITAZAWA, and Shingo YAMAGUCHI. Their careful review and valuable suggestions have greatly contributed to improving the quality of the final work.

I am also grateful to my laboratory colleagues, including Mr. Zhenyu AN, Mr. Hang YANG and Ms. Menglan GUO, whose support and friendship have made my time in Japan both productive and enjoyable.

Finally, I would like to express my deepest gratitude to my family for their unconditional love, unwavering support, and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Background

In the field of computer vision, Human Action Recognition (HAR) is a longstanding challenge and an actively researched topic. The primary focus of HAR is the automatic analysis and understanding of full-body motion sequences, followed by the automated annotation of these sequences [1]. HAR plays a crucial role in various downstream tasks, including healthcare [2], intelligent education [3], remote monitoring [4], entertainment, video storage and retrieval [5], and human-computer interaction [6, 7]. Despite significant progress, HAR still faces challenges such as background clutter, occlusions, viewpoint variations, and real-time constraints. Addressing these challenges requires an effective integration of detection, skeleton extraction, and classification techniques.

Many HAR systems [8, 9, 10] do not incorporate object detection but instead operate directly on raw RGB videos or skeleton sequences. These approaches often suffer from challenges such as background interference, which limits their generalization ability. Object detection [11] serves as a crucial preprocessing step for locating individuals in complex environments, thereby reducing background noise and improving recognition accuracy. Particularly in multi-person scenarios, object detection helps accurately extract Regions of Interest (RoIs) for each individual, ensuring robustness against occlusions and environmental variations. In this thesis, the proposed framework first employs an object detection algorithm to localize individuals in the scene, followed by skeleton data extraction and action classification.

For skeleton data extraction, some methods use 2D Human Pose Estimation (HPE) [12, 13] algorithms to estimate joints in each frame. However, since these 2D HPE methods cannot capture depth information, they struggle to accurately represent the spatial relationships between joints. Additionally, these methods are highly sensitive to occlusions, camera angles, and environmental variations, which can lead to unstable or inaccurate skeleton representations [14]. These challenges hinder the reliability of action recognition systems. To address these issues, some methods attempt to reconstruct 3D skeleton data from 2D using lifting techniques [15, 16]. However, these approaches rely heavily on prior knowledge and assumptions, which can introduce significant errors when dealing with unseen poses or complex movements. Moreover, the accuracy of current 2D-to-3D lifting methods still has considerable room for improvement, limiting their suitability for real-world deployment [14]. Given these limitations, an alternative approach is to use depth cameras, which eliminate the need for indirect estimation by directly capturing 3D skeletal data. Unlike 2D-to-3D lifting methods, depth cameras, such as the Azure Kinect DK [17], utilize Time-of-Flight (ToF) technology to obtain precise 3D skeletal information, enhancing the understanding of complex environments and interactions. Additionally, depth cameras are robust to lighting conditions and background variations, ensuring stable and reliable 3D skeleton data for action classification.

For action classification, numerous methods have been proposed in recent years, each leveraging different techniques to improve accuracy and robustness. These methods can be categorized by the modality of the data used into RGB-based methods [18, 19, 20, 21], skeleton-based methods [22, 23, 24, 25], and multimodal methods [26, 27]. RGB-based methods excel at leveraging rich visual information, such as color and texture. However, they often fail to achieve satisfactory results in practical applications due to their inability to robustly handle environmental noise, such as variations in viewpoints, lighting conditions, background colors, and clothing [28]. Skeleton-based methods rely on the abstract representation of human poses. These methods analyze motions by utilizing the spatial and temporal relationships of human joints. They are robust against changes in color, appearance, lighting, and background [25]. Additionally, since skeleton data represents an abstraction of human poses, these methods inherently offer strong privacy-preserving capabilities. Multimodal methods leverage both RGB and skeleton data simultaneously, aiming to combine the strengths of each modality. However, these approaches face challenges such as complex data fusion, high computational costs, modality imbalance, as well as limited generalization capabilities and poor real-time performance [1]. Based on these, the method proposed for aciton classification in this thesis is a skeleton-based method, which achieves high performance while ensuring real-time capability during deployment.

Based on their technical approaches, HAR methods can be categorized into Convolutional Neural Network (CNN)-based methods [29, 30, 31], Recurrent Neural Network (RNN)-based methods [32, 33], and Graph Convolutional Network (GCN)-based methods [22, 24, 34, 35]. CNNs have been a popular choice for tackling HAR problems, particularly when applied to RGB video data. These methods leverage spatial information within individual frames and temporal relationships between frames to classify actions. However, CNN-based approaches often struggle with handling long-term dependencies or subtle motion variations [30]. RNNs have also been extensively studied in the context of HAR. These methods are particularly effective at capturing temporal dependencies in motion sequences [33]. However, RNNs face significant challenges in computational efficiency and in modeling complex spatial relationships. More recently, GCNs have emerged as a powerful approach for HAR. By modeling human joints as graph nodes and their connections as edges, GCNs enable effective representation of the spatial and temporal dynamics of human motion [22]. This makes GCNs especially suitable for tasks requiring a detailed understanding of human poses and movements. Building upon these advancements, this thesis proposes a GCN-based method to tackle several key challenges in action classification for HAR, including inter-class similarity caused by subtle motion differences, viewpoint variations, incomplete or noisy skeleton data, and the need for effectively modeling spatial-temporal dependencies in human poses.

In this thesis, the proposed framework first employs an object detection algorithm to localize individuals in the scene and extract precise RoIs. Then, depth camera is used to extract skeleton data within the RoIs, providing a robust input for spatiotemporal modeling. Finally, a GCN specifically designed for action classification is applied to analyze the skeleton sequences and classify human actions.

1.2 Research Motivation

The primary motivation of this thesis, as described in Section 1.1, stems from the fact that HAR is a fundamental yet challenging task in computer vision with a wide range of applications. Despite significant advancements, the deployment of HAR systems in real-world scenarios remains hindered by several key limitations. Addressing these challenges requires a more effective integration of object detection, skeleton extraction, and action classification techniques.

Many HAR systems process RGB videos or skeletal sequences directly without first detecting and isolating individuals. This can lead to background interference, reducing accuracy and robustness in complex environments. Integrating an object detection module can help precisely locate individuals in a scene, thereby improving both recognition accuracy and computational efficiency. Therefore, incorporating object detection into HAR systems is another motivation of this thesis.

Current HAR systems often rely on HPE algorithms to extract skeletal data. However, 2D HPE methods cannot capture depth information, making it difficult to accurately model human motion in three-dimensional space. Additionally, 2D-to-3D lifting methods heavily depend on prior knowledge and assumptions, leading to significant estimation errors for unseen poses or complex movements. In contrast, depth cameras provide a direct and accurate 3D skeleton data with improved robustness. Therefore, leveraging depth cameras in HAR systems is another motivation of this thesis.

The performance of HAR systems largely depends on the effectiveness of the action classification model. CNN-based methods primarily focus on spatial features but lack the ability to effectively capture temporal information. RNN-based methods suffer from high computational costs and the vanishing gradient problem. GCNs have demonstrated great potential in modeling both spatial and temporal dependencies in skeleton data. Therefore, employing GCN-based methods for action classification in HAR systems is the final motivation of this thesis.

1.3 Thesis Organization

The rest of this thesis is organized as follows:

Chapter 2 provides a comprehensive literature review, covering key research areas related to this work, including object detection, skeleton extraction, and action classification. This chapter discusses existing methods and highlights their strengths and limitations, forming the foundation for the proposed framework.

Chapter 3 presents an object detection algorithm based on a cascade network, detailing its overall architecture, Re-BiFPN feature fusion, and the Rep-CIoU loss function. The effectiveness of the method is evaluated through experiments on prohibited item detection and human detection, demonstrating its robustness in various scenarios.

Chapter 4 introduces the Skeleton Temporal Fusion Graph Convolutional Network (STF-GCN) for action classification. It describes temporal dimension encoding, including temporal motion and angle features, and discusses the architectural design of the proposed method. This chapter also evaluates the model's performance through experiments on elderly action recognition and general action recognition tasks.

Chapter 5 focuses on the depth camera-based human action recognition system, detailing the system architecture, data preprocessing, and experimental setup. Furthermore, it explores two real-world applications: fall detection in elderly care and hand-raising detection in classroom environments, demonstrating the system's practical significance.

Chapter 6 concludes the dissertation by summarizing the key findings and contributions. It also discusses future research directions to further improve the proposed methods and extend their applications.

Chapter 2

Literature Review

This chapter provides a comprehensive review of existing methods related to Human Action Recognition (HAR), focusing on object detection, skeleton extraction methods, and action classification methods. Section 2.1 discusses object detection, first introducing traditional detection methods, followed by an overview of deep learning-based object detection methods. Section 2.2 presents skeleton extraction methods from three perspectives: 2D pose estimation methods, 3D pose estimation methods, and depth camera. Section 2.3 reviews action classification methods, categorizing them into three main types: CNN-based, RNN-based, and GCN-based methods. Section 2.4 highlights human action recognition applications, with a particular emphasis on elderly care and smart education.

2.1 Object Detection

Object detection is a fundamental task in computer vision that aims to identify and localize specific categories of objects (such as humans and prohibited items) in RGB images. Its goal is to equip computers with basic visual understanding, enabling them to answer the questions: What objects are present? and Where are they located?

2.1.1 Traditional Detection Methods

Early object detection algorithms were primarily based on handcrafted features. Viola and Jones [36, 37] proposed the VJ detector, which achieved real-time face detection for the first time without any restrictions. This detector employed a straightforward sliding window approach, scanning all possible positions and scales in an image to determine whether a window contained a face. Dalal and Triggs introduced the Histogram of Oriented Gradients (HOG) feature descriptor [38], as shown in Figure 2.1, which was considered a significant improvement over the scale-invariant feature transform [39] and shape context [40] methods at the time. Although HOG could be applied to various object categories, its primary motivation was pedestrian detection. As an extension of the HOG detector, Felzenszwalb et al. [41] proposed the Deformable Part Model (DPM) detector. This approach formulated training as learning the optimal way to decompose an object and inference as aggregating detections of different object parts. Later, Girshick further extended DPM into a "mixture model" to handle objects with greater variability in real-world scenarios, along with several other improvements [42, 43].

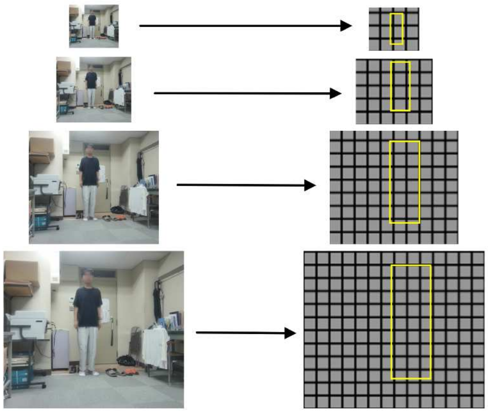


Figure 2.1: HOG feature pyramid for human detection.

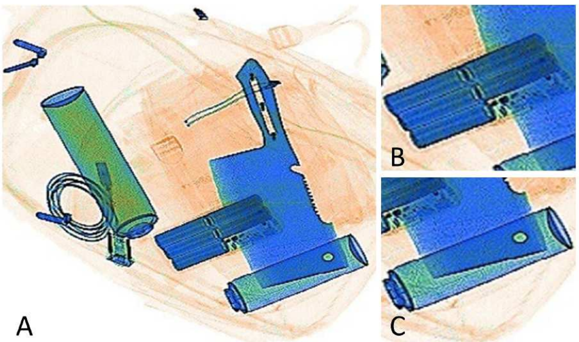


Figure 2.2: Example of luggage image (A) with corresponding regions for a camera (B) and a firearm (C).

In addition to human detection, object detection methods have been increasingly applied in security screening to identify prohibited items, as shown in Figure 2.2. The goal of this task is to automatically locate and recognize potential contraband in images generated by security scanning devices. Baştan et al.[44] suggested using the Bag of Visual Word (BoVW) framework combined with the SVMalgorithm to detect prohibited items. Mery et al. [45] proposed a method based on multiple X-ray views to detect regular prohibited items. The method consists of two steps: "structure estimation", to obtain a geometric model of the multiple views from the object to be inspected (baggage); and "parts detection", to detect the parts of interest (prohibited items). Inspired by [44] and the advantages of neural networks, Akçay et al. [46] employed a transfer learning paradigm combined with an SVM such that a pre-trained CNN can be optimized explicitly as a later secondary process that targets this specific application domain. Roomi et al. [47] trained fuzzy KNN classifiers were trained with contextual descriptors and Zernike polynomials to study pistol detection, but only fifteen image examples were evaluated.

2.1.2 Deep Learning-based Detection Methods

With the limitations of traditional hand-crafted feature-based methods becoming apparent and the rapid advancements in deep learning, object detection research has entered a new era of growth and innovation. Girshick et al. [48] pioneered the Region-based Convolutional Neural Network (R-CNN), marking a breakthrough in object detection. Following this milestone, the field of object detection progressed at an unprecedented pace. He et al. [49] introduced the Spatial Pyramid Pooling Network (SPPNet), whose key contribution was the introduction of the Spatial Pyramid Pooling (SPP) layer. This layer enabled CNNs to generate fixed-length representations without the need to resize images or regions of interest. Shortly after, Girshick [50] proposed the Fast R-CNN detector, which further improved upon R-CNN [48] and SPPNet [49]. Subsequently, Ren et al. [51] introduced the Faster R-CNN detector, building upon the advancements of Fast R-CNN. Since most detectors performed detection only on the feature maps of the top network layers, Lin et al. [52] proposed the Feature Pyramid Network (FPN). As shown in Figure 2.3, FPN introduced a top-down architecture with lateral connections, facilitating the construction of high-level semantic features across all scales.

In addition to tradition algorithms [46], Akcay et al. also studied deep learning strategies to improve the performance of cluttered datasets further [53]. Wang et al. [54] collected a dataset named PIDray and proposed a selective dense attention

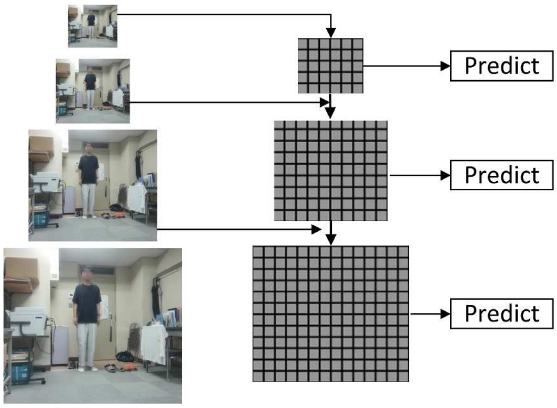


Figure 2.3: Feature pyramid netword.

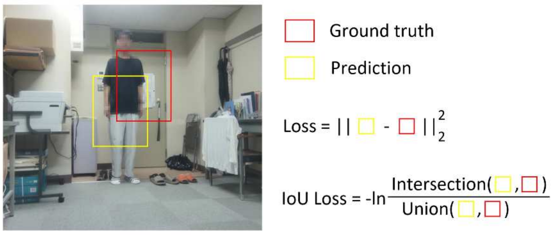


Figure 2.4: Illustration of *IoU* loss and Smooth *l*2 loss for bounding box prediction.

network consisting of a dense attention module and a dependency refinement module. Miao et al. [55] collected a dataset named SIX-ray and presented a CHR model, which achieves class balance through a class-balanced loss function. To learn the different scales of prohibited items, Zhang et al. [56] proposed a novel asymmetrical convolution multi-view neural network (ACMNet). However, from their experimental results, the detection accuracy of some targets was not significantly improved. The fundamental reason is that there is a significant semantic gap between each feature layer. Moreover, there is a lack of handling prohibited item coordinate information across different scale feature maps. Feature pyramids are mainly used to improve the semantic gap in object detection [57, 58, 59]. The loss function for the object detection task consists of two parts, classification loss and bounding box regression loss. The bounding box regression loss for the object detection task has undergone the evolution of Smooth L2 loss [60], IoU loss [61] (as shown in Figure 2.4), repulsion loss [62], GIoU loss [63], DIoU loss [64] and CIoU loss [65] in recent years.

2.2 Skeleton Extraction Methods

Human Pose Estimation (HPE) aims to identify key body parts and construct a skeletal representation of the human body based on sensor-captured data such as images and videos. By providing detailed geometric and motion information, HPE has been widely applied across various fields, including human-computer interaction, motion analysis, Augmented Reality (AR), Virtual Reality (VR), and healthcare.

2.2.1 2D Human Pose Estimation

HPE aims to predict the 2D locations of human keypoints from images or videos. Traditional 2D HPE methods rely on handcrafted feature extraction techniques for identifying body parts. Toshev and Szegedy [66] introduced DeepPose, a cascaded deep neural network regressor designed to learn keypoints directly from images. This marked a shift in HPE research from classical approaches to deep learning-based methods. Deep learning-based HPE methods can be broadly categorized into top-down and bottom-up approaches [14].

Top-down methods first use a person detector to obtain bounding boxes for individuals in the input image [14], as shown in Figure 2.5. A single-person pose estimator is then applied to each detected bounding box to generate multi-person poses. In top-down methods, the number of people in the input image directly affects the computational time. Xiao et al. [67] added deconvolution layers to ResNet, creating a simple yet effective structure for generating high-resolution heatmaps, from which skeletal positions are obtained. To improve the accuracy of keypoint localization, Wang et al. [68] proposed a method called Graph-PCNN, a graph-based and model-agnostic two-stage framework. Subsequently, Cai et al. [69] introduced a multi-stage network comprising a Residual Steps Network (RSN) module and a Pose Refinement Machine (PRM) module. The RSN module learns fine-grained local representations through an efficient intra-stage feature fusion strategy, while the PRM module balances local and global feature representations. With the rapid development of Transformer-based methods, Yang et al. [70] proposed a Transformer-based 2D HPE model named TransPose. To enhance memory and computational efficiency, Yuan et al. [71] replaced the blocks in HRNet [72] with Transformer modules, introducing HRFormer, a high-resolution Transformer model.

In contrast, bottom-up methods estimate the body joints of all individuals in an image simultaneously (as shown in Figure 2.6) and then group them into separate subjects based on geometric relationships. Since bottom-up methods do not require

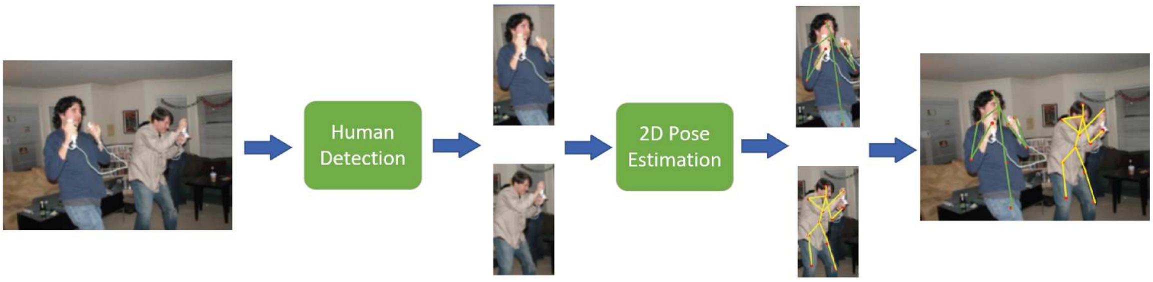


Figure 2.5: Top-down methods. Top-down methods have two sub-tasks: human detection and 2D pose estimation in the region of a single human

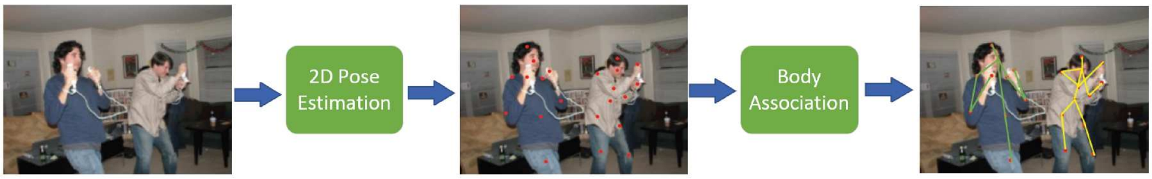


Figure 2.6: Bottom-up methods. Bottom-up methods also have two sub-tasks: detect all keypoints of body and associate body in different human bodies

detecting each person individually, they generally achieve faster computation speeds compared to top-down methods [14]. Cao et al. [73] proposed a detector named OpenPose, which utilizes Convolutional Pose Machines [74] to predict keypoint coordinates through heatmaps. It then employs a vector representation encoding limb positions and orientations to associate keypoints with individuals. Although OpenPose achieves remarkable results on high-resolution images, its performance significantly degrades on low-resolution images and under occlusions. To address this issue, Kreiss et al. [75] introduced the PifPaf method, which consists of a Part Intensity Field (PIF) to predict body part locations and a Part Association Field (PAF) to encode joint connections. Later, Sun et al. proposed HRNet [76], followed by Cheng et al. [72], who introduced the Higher Resolution Network. This network enhances the high-resolution feature maps generated by HRNet through deconvolution, aiming to resolve the multi-scale challenges in bottom-up HPE methods. Bottom-up HPE approaches have also adopted multi-task architectures. Kocabas et al. [77] proposed MultiPoseNet, a multi-task learning model incorporating a Pose Residual Network. MultiPoseNet is capable of performing keypoint prediction, human detection, and semantic segmentation simultaneously.

2.2.2 3D Human Pose Estimation

3D HPE methods can be categorized into direct estimation methods and 2D-to-3D lifting methods. As shown in Figure 2.7, direct estimation methods infer 3D human poses directly from 2D images without the need for intermediate 2D pose representations. Li et al. [78] employed a shallow network that simultaneously trains a body part detector with sliding windows and a pose coordinate regressor for 3D pose estimation. Sun et al. [79] proposed a structure-aware regression method that utilizes a skeleton-based representation for greater stability. By leveraging the 3D skeletal structure and encoding interactions between joints, they defined a compositional loss to enhance pose accuracy. Pavlakos et al. [80] introduced a volumetric representation approach, transforming the highly non-linear problem of 3D coordinate regression into a more manageable form within a discrete space.

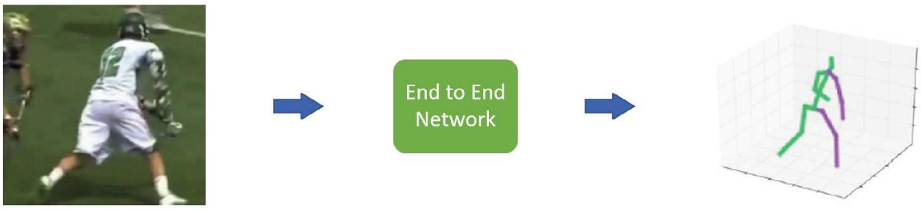


Figure 2.7: Direct estimation methods.

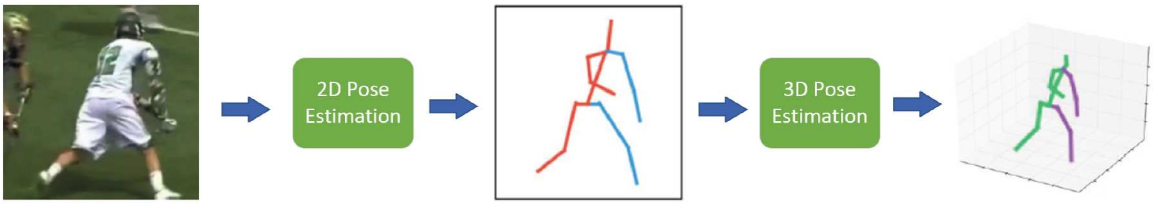


Figure 2.8: 2D-to-3D Lifting methods.

As shown in Figure 2.8, driven by the success of 2D HPE, the 2D-to-3D lifting approach has become a popular solution for 3D HPE. Martinez et al. [81] proposed a method that regresses 3D joint positions directly from 2D joint locations. While this approach achieved impressive results, its primary limitation lies in its heavy reliance on 2D pose detectors, which can lead to ambiguities in 3D reconstruction. To address this issue, Wang et al. [82] introduced a pairwise ranking CNN. They employed a coarse-to-fine pose estimator that regresses 3D poses based on both 2D joints and a depth ranking matrix. Since human poses can be represented as a graph, where joints serve as nodes and bones as edges, GCNs have been increasingly applied to 2D-to-3D lifting methods. Ci et al. [83] proposed a Locally Connected Network

(LCN) that leverages both fully connected networks and GCNs to model relationships between local joint neighborhoods. Zhou et al. [84] further introduced a modulated GCN, incorporating weight modulation and affinity modulation to enhance 3D pose estimation performance.

2.2.3 Depth Camera

Depth cameras, such as the Azure Kinect DK [85], RealSense [86] and Orbbec, not only provide traditional RGB images but also offer rich depth information, revolutionizing human pose estimation. Unlike conventional cameras that capture only color and intensity, depth cameras measure the distance between objects and the sensor, creating a 3D representation of the scene. As shown in Figure 2.9, depth cameras utilize technologies such as Time-of-Flight (ToF), structured light, and infrared sensing to capture depth information, enhancing the understanding of complex environments and interactions [87]. They typically generate depth and amplitude images, where each pixel represents the distance between the target object and the image sensor.

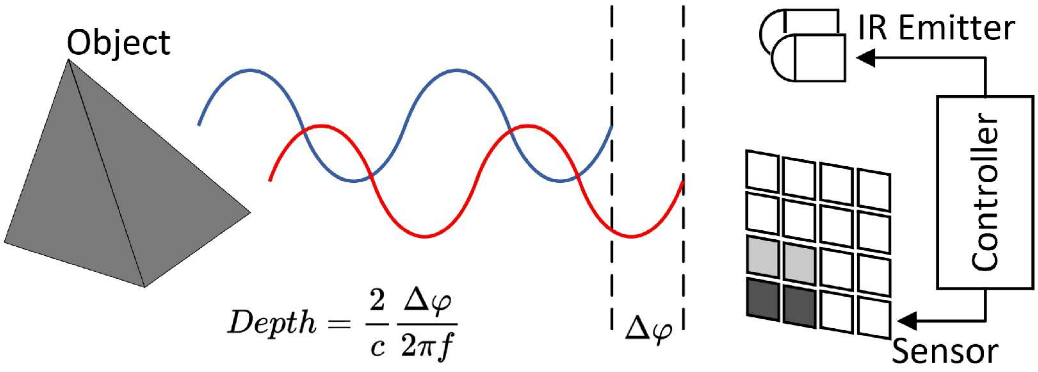


Figure 2.9: The principle of depth camera [88].

One of the most critical applications of depth cameras is skeleton extraction, which involves estimating human keypoints from depth data. Many modern depth cameras come with built-in SDKs, such as the Azure Kinect Body Tracking SDK, which leverage depth information to infer 3D joint positions with high accuracy. Andriyanov et al. [89] proposed a method that combines YOLOv3, a widely used object detection model, with Intel RealSense depth camera data to accurately determine the real-world coordinates of apples in robotic harvesting tasks. By integrating depth and brightness channel information, this approach calculates relative distances and transitions to a symmetric coordinate system, enabling high-precision localization. Gai et al. [90] developed a vision-based under-canopy navigation system using a ToF camera. The system detects parallel crop rows from depth images captured beneath the crop canopy. Based on the detection results, it accomplishes two key tasks in navigation: robot localization and path planning. Farhadi et al. [86] proposed a depth image refinement technique based on the Intel RealSense D455 camera, aimed at enhancing the camera's usability in underwater environments. The method first captures depth images using the depth camera and generates relative depth images based on the recorded color images. Then, a mapping process is applied between the relative depth data and the RealSense depth images to produce visually enhanced and highly accurate depth images.

2.3 Action Classification Methods

Action classification is a fundamental task in computer vision, aiming to automatically identify and categorize human actions from video or sensor data [28]. This section provides a technical overview of the three fundamental types of skeleton-based data action classification methods: RNN, CNN, and GCN.

2.3.1 RNN-based Action Classification

RNNs, with their ability to maintain temporal dependencies across sequences, fulfill the need for modeling sequential data, making them particularly useful for tasks involving time-series analysis. RNN-based methods [91, 92, 93, 94, 95], as shown in Figure 2.10, typically represent skeletal data as a sequence of coordinate vectors, with each vector representing a joint of the human body. A representative approach is spatial-temporal long short term memory proposed by [91]. They extended contextual dependency to spatialtemporal domain to analyze the sources of action-related information within the input data over both domains concurrently. Considering that RNN with LSTM can learn feature representations and model long-term temporal dependencies automatically, Zhu et. al [92] proposed a fully connected deep long short term memory network with regularization terms to learn co-occurrence features of skeleton joints. However, these methods use RNN to handle raw skeletons only focus on the contextual dependency in the temporal domain and neglect the construction of spatial information. Wang et al. [93] proposed a novel two-stream RNN architecture to model both temporal dynamics and spatial configurations for skeleton based action recognition.

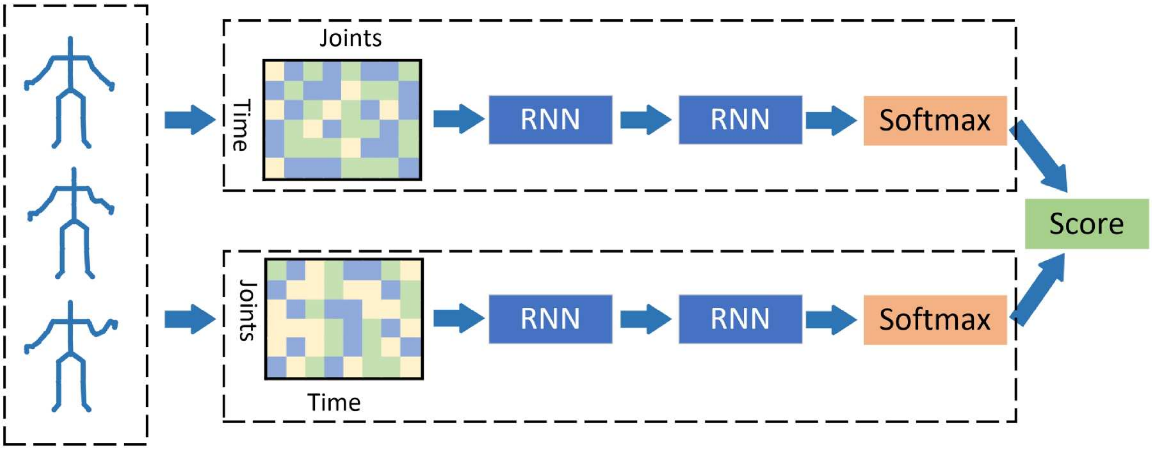


Figure 2.10: Illustration of the RNN-based action classification method.

2.3.2 CNN-based Action Classification

CNNs effectively detect spatial hierarchies in data by leveraging local connectivity and shared weights, making them well-suited for extracting spatial features from images and videos. CNN-based methods [30, 29, 96, 97, 98], as shown in Figure 2.11, are generally more popular than RNN-based because the CNNs can easily learn high-level semantic information with their excellent feature extraction ability and are easier to train than RNNs. Liang et al [29] proposed the three-stream CNN method, which includes a pariware feature fusion startegy to fully take advantage of the complementary and diverse nature among three features. Recently, Duan et al. [98] proposed a novel framework PoseC3D, which takes as input 2D poses obtained by modern pose estimators. It has demonstrated remarkable success in skeleton-based action recognition, leading to notable performance improvements. However, it's important to recognize that their inherent design, tailored for grid-based data, hinders their effectiveness in handling graph data and fully leveraging its underlying topology.

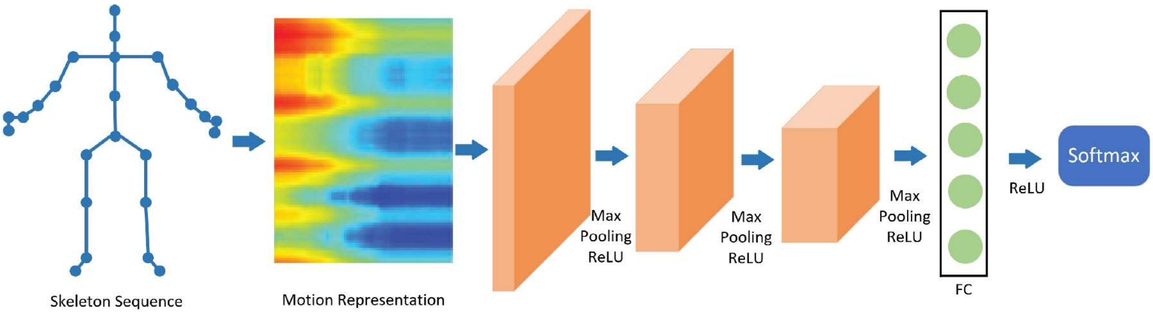


Figure 2.11: Illustration of the CNN-based action classification method.

2.3.3 GCN-based Action Classification

The emergence of GCNs marks a paradigm shift, as they utilize graph structures to model complex relationships between non-Euclidean data. By capturing spatial and temporal dependencies through graph convolution operations (as shown in Figure 2.12), GCNs have demonstrated superior performance in skeleton-based action recognition. Scarselli et al. [99] introduced a trainable GCN designed to process graph-structured data based on the topological structure of human skeletal joints. Lin et al. [100] proposed the SlowFastGCN framework, which employs ST-GCN to model the spatio-temporal information of human skeletons, utilizing a slow stream to capture static semantics and a fast stream to capture fine-grained motions. Monti et al. [101] modified the spectral domain GCN to develop a more efficient spatial domain GCN that operates directly on graph vertices. Yan et al. [34] introduced a novel ST-GCN model. This model utilizes only the first-order information, representing the coordinates of the joints, to create a spatial-temporal graph with joints as vertices and natural connectivities as edges. Following this work, Shi et al. [35] introduced 2s-AGCN model, which effectively captures the crucial second-order information related to the lengths and directions of bones within the skeleton data. This incorporation of second-order information has proven to be highly informative and beneficial for improving action recognition performance. Subsequent to these pioneering works, researchers [25, 102, 103], aiming to harness the full potential of spatial dimension features, introduced a variety of distinct model architectures. In more recent times, Qin et al. [104] offer a novel method for fusing higher-order features, in the form of spatial angle, into graph convolutional networks. This enables the model to effectively capture the intricate relationships between joints and bones.

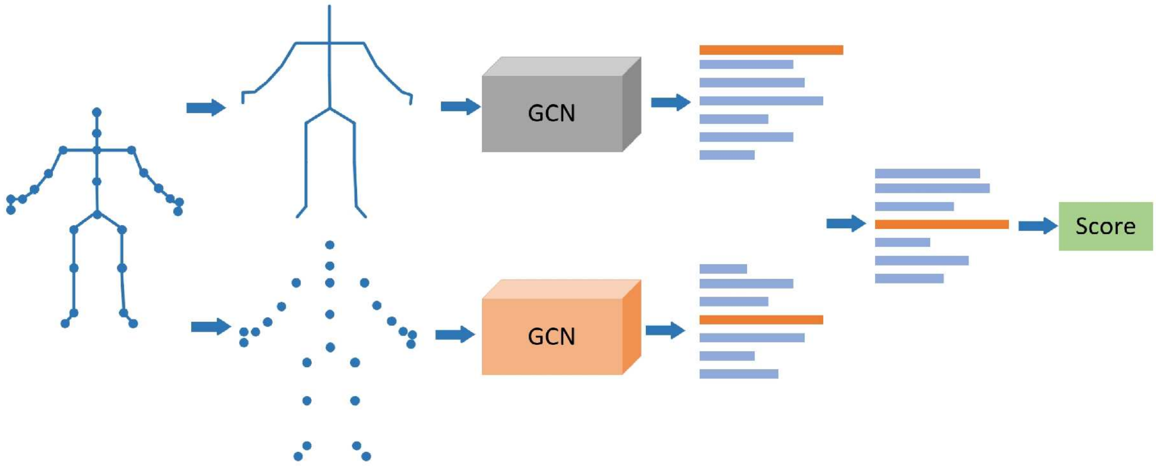


Figure 2.12: Illustration of the GCN-based action classification method.

2.4 Human Action Recognition

This section explores two important applications of human action recognition, including action recognition in elderly care and action recognition in smart education. Although both elderly care and smart education involve human action recognition, the nature and objectives of the target actions differ significantly. In elderly care, actions such as falling are typically involuntary and may indicate medical emergencies, requiring timely and sensitive detection to ensure safety. In contrast, actions like hand-raising in smart education are intentional and controlled, serving as indicators of student engagement and classroom participation.

2.4.1 Action Recognition in Elderly Care

Action recognition plays a pivotal role in enhancing the safety and quality of life for the elderly, a demographic that often requires continuous monitoring due to a higher risk of falls that could necessitate immediate medical attention. Jang et al. [2] proposed a network called Four Stream Adaptive CNN (FSA-CNN), which has three main properties: robustness to spatio-temporal variations, inputadaptive activation function, and extension of the conventional two-stream approach. Shu et al. [105] proposed a Expansion-Squeeze-Excitation Fusion Network (ESE-FN), which learns modal and channel-wise ESE attentions for attentively fusing the multi-modal features in the modal and channel-wise ways. Zin et al. [106] introduced a system that integrates feature extraction methods from previous works and utilizes depth frame sequences provided by depth cameras. The system locates individuals by extracting different RoI from UV-disparity maps. Tabbakha et al. [107] introduced the development and testing of a wearable device featuring motion detection and indoor localization based on a random forest classifier. The action recognition phase utilizes a gyroscope and an accelerometer to detect various types of movements. Among the methods discussed, only several methods [106, 107] involved actual model deployment and testing. The remaining methods did not implement the proposed algorithms in practical settings. Although Zin et al. [106] employ depth cameras, they only use depth information for locating individuals and do not fully utilize them in the action recognition phase. Instead, the action recognition phase of their method involves a strategy of randomly sampling frame sequences, a process that can lead to inconsistent results and potentially miss critical motions, reducing the overall effectiveness and accuracy of the action recognition.

2.4.2 Action Recognition in Smart Education

As educational technologies evolve towards personalized learning and data-driven feedback, action recognition in smart education becomes increasingly important. Recognizing specific actions, such as hand-raising, plays a key role in assessing student engagement, facilitating classroom interaction, and optimizing teaching strategies. Therefore, some studies have focused on detecting hand-raising. Zhou et al. [108] proposed an algorithm for recognizing hand-raising actions. It accomplishes the recognition of hand-raising through three tasks: hand detection, pose estimation, and heuristic matching. Si et al. [109] integrated a feature pyramid into their model architecture, thus designing a region-based fully convolutional network to detect hand-raising gestures. This design somewhat improves the detection capability for low-resolution hand gestures. The method proposed by Liao et al. [110] includes two stages: pose estimation and gesture recognition. The goal of the pose estimation stage is to estimate human joint positions, which are then used to recognize gestures through predefined relationships among the joints. Similarly, Lin et al. [3] employed a process akin to that of Liao et al. [110], utilizing CNNs to classify actions. Although these methods have achieved commendable results, they are all based on RGB videos, which lack depth information and are highly susceptible to variations in lighting and clothing. Additionally, these methods typically segment an action into frames to perform recognition, which can lead to fragmentation of contextual information and potential loss of continuity in action sequences.

Chapter 3

Object Detection Algorithm Based on Cascade Network

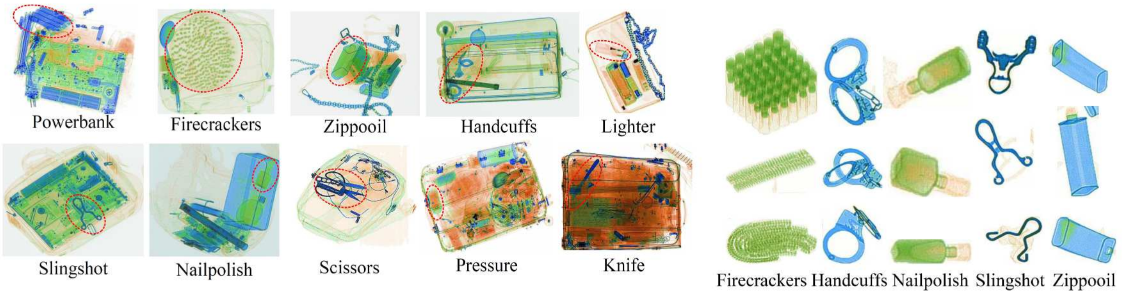
In this chapter, a cascade network-based object detection algorithm is proposed and applied to two major tasks: prohibited item detection and human detection. Although these two tasks involve different target objects, they fundamentally belong to the same category of object detection and face similar challenges, such as complex background interference and scale variations. Therefore, this chapter takes prohibited item detection as a representative case to analyze the existing problems and challenges in object detection. The proposed method is then validated on both prohibited item detection and human detection tasks to demonstrate its effectiveness.

3.1 Introduction

As urban populations and crowd densities at public transportation hubs grow, security inspection is becoming increasingly important in protecting public safety [111]. Security inspection machine is the most widely used security inspection equipment [112]. It uses X-ray technology to scan a traveler's package and generate an irradiation image in real time. Currently, most of the work of security inspection still relies on highly trained security staff to carefully identify by eye whether there are any prohibited items in the irradiation image [113, 114]. As security staff fulfills a demanding occupation, being in a high-pressure work environment for elongated periods may cause false detection or missed detection of prohibited items, which may seriously threaten public safety [115]. Moreover, frequent shift changes consume many human resources and increase labor costs.

With the substantial development of artificial intelligence technologies, automatic security inspection of prohibited items has become possible in recent years. Machine learning and deep learning algorithms are the main methods for prohibited item detection within X-ray security inspection images. Muhammet et al. [44] utilize the Bag of Visual Word (BoVW) framework with SVMs structure to classify prohibited items. Mery et al. [45] proposed to use a method based on multiple X-ray views to detect regular prohibited items with very defined shapes and sizes. The main drawback of these machine learning approaches is the reliance on hand-crafted features that require manual engineering. Wang et al. [54] and Miao et al. [55] proposed a selective dense attention network and Class-balanced Hierarchical Refinement (CHR) approach, respectively. These deep learning methods have achieved better performance compared to machine learning methods.

However, three challenges still appeal to us in the prohibited items detection task. First, the X-ray security inspection dataset has an imbalanced distribution of categories. Deep learning as a standard data-driven technique, a balanced distribution of categories in the dataset is the cornerstone of the algorithm to achieve better performance. The dataset, as shown in Figure 3.1, consists of two parts provided by iFLYTEK CO.LTD: X-ray images of the entire package and X-ray images of a separate prohibited item. To alleviate the category imbalance problem, we propose to leverage the Poisson blending algorithm with the Canny edge operator to fuse an X-ray image of a separate prohibited item with an X-ray image of the entire package. This data enhancement method can naturally fuse the two X-ray images with minimal noise and increase the diversity and complexity of the samples.



(a) Example of X-ray images of the entire package with 10 categories of prohibited items. For clarity, we show one category per image.

(b) Example of X-ray images of a separate prohibited item.

Figure 3.1: The visualization examples of the X-ray security inspection dataset.

Second, diversity of prohibited item scales. The size of prohibited items in the same X-ray image varies. There is also variation in the size of the same type of prohibited items in different X-ray images. To address the challenge of detecting prohibited items at diverse scales, we propose an approach named Re-BiFPN. For comparison, a cascaded network [116] merges multiple detectors and leverages FPN [52] for feature extraction. Unlike the FPN, which mainly concentrates on managing multi-scale features through straightforward aggregation, Re-BiFPN presents a novel theoretical advancement. It establishes a recursive multi-scale structure and incorporates the coordinate information of prohibited items into the feature layers. This unique design allows our model to refine multi-scale information iteratively, enhancing multi-scale representations. Moreover, it equips the model with the ability to discern the relative positions and spatial relationships among prohibited items across different scales.

Third, the problem of overlapping prohibited items has been receiving the attention of most researchers, such as [115, 117, 118]. In prohibited item detection, however, no loss function is designed for this problem. Since the evaluation metric for the prohibited item detection task is IoU (Intersection over Union), the loss function in the original cascade network, which calculates the loss of the prediction box's four points, is unsuitable for this task. We designed a new loss function, Rep-CIoU, to make the model more robust to overlapping items, which considers the IoU between multiple prediction boxes and the centroid distance between the prediction box and ground-truth. It can effectively prevent multiple prediction boxes from filtering out by NMS (Non-Maximum Suppression) when the IoU generated by a particular prediction box and other surrounding prediction boxes is large or their centroids' distance is small.

In summary, the contributions of this thesis for object detection are as follows: (1)

This thesis proposes to utilize the Poisson blending algorithm with the Canny edge operator to fuse an X-ray image of a separate prohibited item with an X-ray image of the entire package, which can naturally fuse the two X-ray images with minimal noise and increase the diversity and complexity of the samples. (2) This thesis proposes the Re-BiFPN feature fusion method, which includes a CA-ASPP module and a recursive connection. The method can learn the coordinate information implicit in the feature maps while improving the network's ability to extract multi-scale features. (3) This thesis designs a new loss function, Rep-CIoU, to make the model more robust to overlapping items, which considers the IoU between multiple prediction boxes and the centroid distance between the prediction box and ground-truth. This loss function can effectively reduce missed detection due to overlap.

3.2 Proposed Cascade Network for Object Detection

3.2.1 Overall Architecture of the Cascade Network

Our method is a multi-stage target detection architecture consisting of a series of IoU threshold trained detectors that are continuously improved. The cascade process can continuously change the distribution of candidate boxes and re-sample them by adjusting the thresholds [116]. The overall framework of our method is shown in Figure 3.2.

We construct the training set by fusing the X-ray images of a single prohibited item with the X-ray images of the entire package. For clarity, we have artificially marked the location of a single prohibited item in the fused image with a red circle, as shown in the "Training set" of Figure 3.2. Our method employs ResNeXt-101(32x4d) [119] as the backbone network. The characters on the arrow and the white circles indicate the feature map of the backbone. The colored circles indicate the multi-scale features in our proposed Re-BiFPN. The up arrow in the Re-BiFPN indicates down-sampling; the down arrow indicates up-sampling; horizontal arrow and curved arrow indicate connection operations; the red arrow is recursive connection. The CA-ASPP is coordinate attention atrous spatial pyramid pooling module. RPN is the region proposal network. "pooling" is region-wise feature extraction. "B0" is proposals in all architectures. The "Rep-CIoU" is our proposed bounding box regression loss function. And "C" is classification loss function.

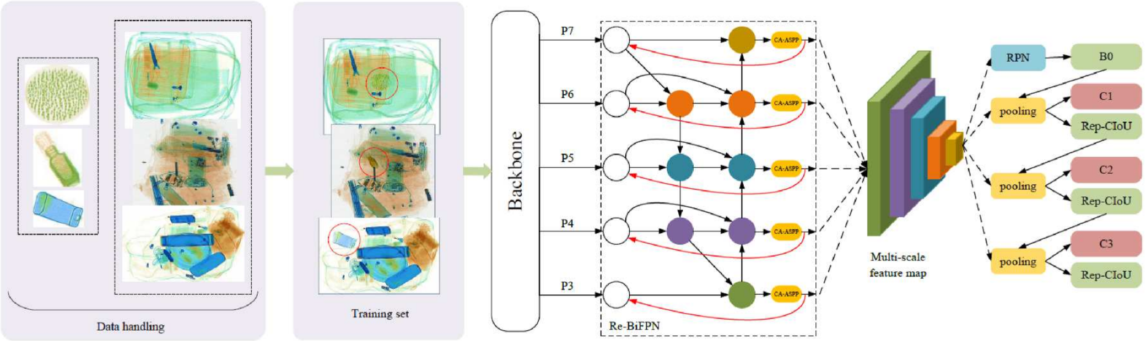
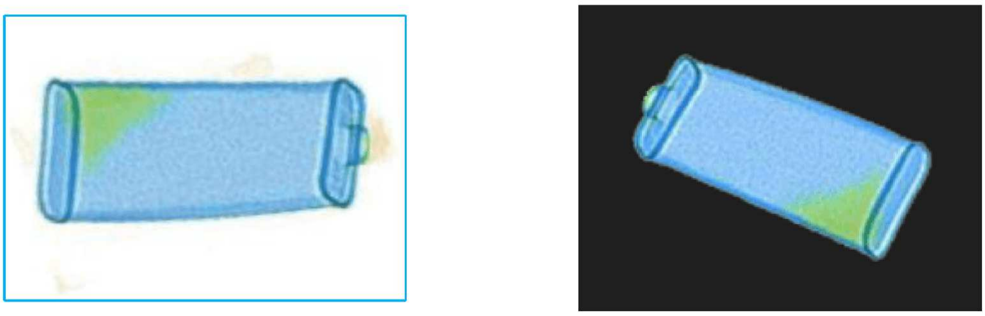


Figure 3.2: The overall framework of the cascade network.

3.2.2 Data Handling

The issue of category imbalance refers to a situation in a training dataset where there is a significant disparity in the number of samples among different categories. This imbalance may cause the model to overly optimize for the majority category while



(a) Source image.

(b) Image to be fused.

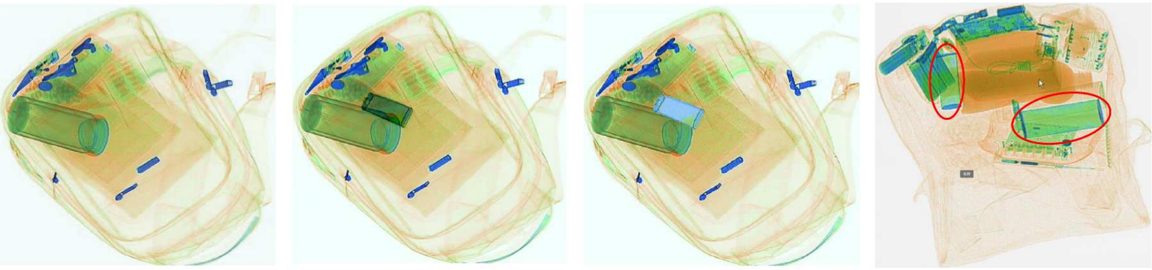
Figure 3.3: Contour detection is performed on the source image utilizing the Canny edge operator. For the clarity of the presentation, we have marked the borders of the source image in blue. After processing by the Canny edge operator, the boundary of the image to be fused is the contour of the prohibited item.

neglecting the minority category, thereby reducing the model's generalization capability. In X-ray security inspection dataset, category imbalance may stem from the much higher occurrence rates of certain prohibited items, such as knives and lighters, compared to others. Additionally, obtaining X-ray images of specific prohibited items, like fireworks and slingshots, becomes more challenging due to their dangerous nature and rarity. To address the imbalance distribution of categories, we propose utilizing the Poisson blending algorithm [120] in conjunction with the Canny edge operator. This method aims to mitigate the category imbalance issue by fusing X-ray images of individual prohibited items with images of entire packages.

The image in Figure 3.1b is source image for the Poisson blending. First, we leverage the excellent contour detection capability of the Canny edge operator to extract the contour information of the prohibited item, thus avoiding the introduction of out-of-contour noise. The operation on the source image is shown in Figure 3.3. The source image is randomly rotated or scaling. The random rotation range is $[0, 360^\circ]$. The range of random scaling is $\frac{1}{n}$ times the length and width of the original image, $n \in \{n|1 \leq n \leq 10, n \in Z\}$. Then contour detection is performed, and interfering parts other than the target contour is removed to obtain an image to be fused.

After contour detection, we utilize the Poisson blending algorithm to maximize the retention of gradient information of the image to be fused to make the fusion boundary smoother. The Poisson blending of Figure 3.3b and Figure 3.4a is performed. Finally, an image is obtained after Poisson blending as shown in Figure 3.4b.

Compared to the addition operation, as shown in Figure 3.4b and Figure 3.4c, our method offers superior image fusion quality and improved edge blending effects. This advantage stems from Poisson blending's consideration of gradient discrepancies



(a) Image before the Poisson blending.

(b) Image after the Poisson blending.

(c) Image obtained through addition.

(d) Actual image with occlusion in dataset.

Figure 3.4: An example of image fusion utilizing the Poisson blending with the Canny edge operator. For clarity, we also show the image obtained by utilizing the general addition operation, denoted as (c), the actual image with occlusion in the dataset, denoted as (d), and compare them to (b).

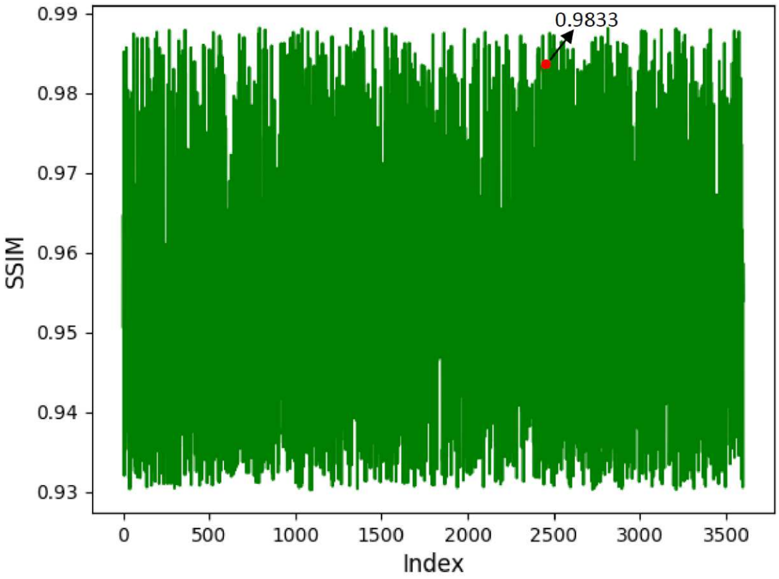
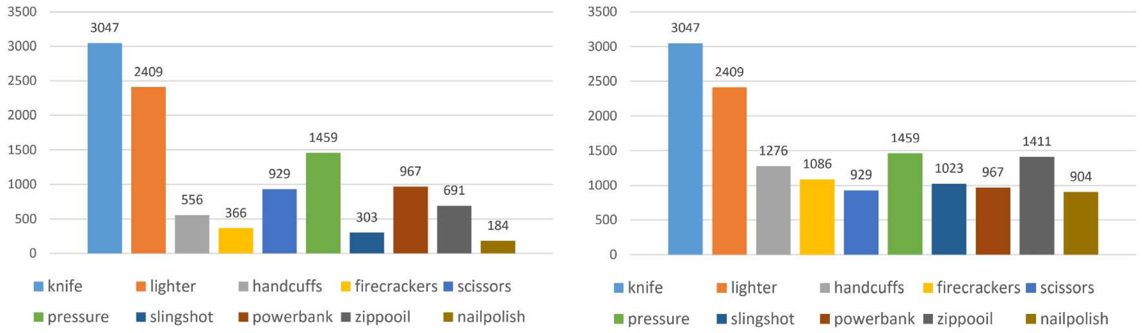


Figure 3.5: Line chart of Structural Similarity Index (SSIM) for each image before and after Poisson blending. The horizontal axis (Index) represents the image ID in the dataset, and the vertical axis (SSIM) indicates the similarity score between the original image and its Poisson-blended version. Each green bar corresponds to one image's SSIM value, with higher values indicating greater structural similarity. The maximum SSIM value reaches 98.82%, the minimum is 93.03%, with an average of 95.50%. The red-marked point represents the SSIM value of the image pair shown in Figure 3.4

between the target and source images, enabling a more natural fusion that avoids abrupt changes in edges and colors. Conversely, the addition operation merely involves straightforward pixel value summation, which might lead to less smooth and natural image fusion outcomes.



(a) Distribution of prohibited item categories before Poisson blending.

(b) Distribution of prohibited item categories after Poisson blending.

Figure 3.6: Distribution of prohibited item categories before and after Poisson blending.

In Figure 3.4d, we display an actual image with occlusion. Both the object marked in red in Figure 3.4d and the blended object in Figure 3.4b represent Zippooil. It's evident that the image generated by our method reproduces the occlusion nearly as accurately as the actual image. In addition, the image produced by our method maintains edge details similar to the actual image and prevents abrupt transitions in edges and colors.

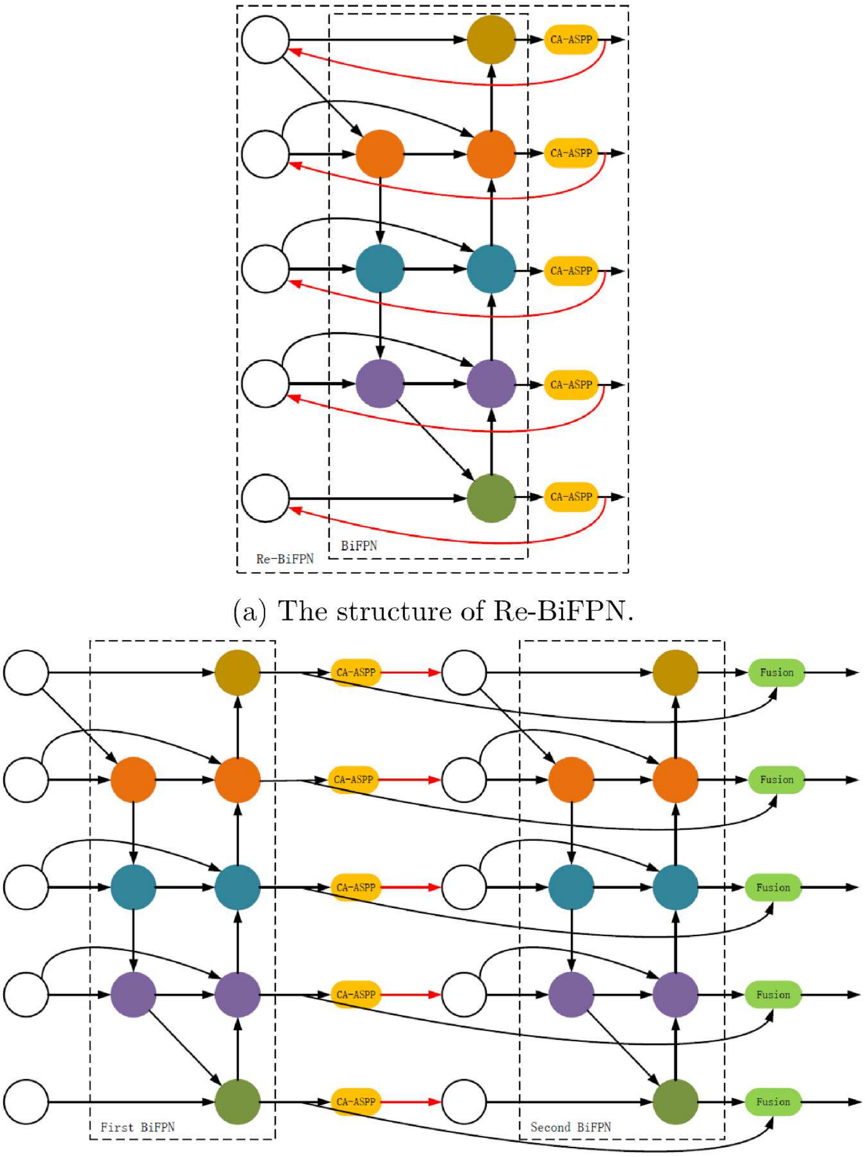
We employ the Structural Similarity Index (SSIM) to quantitatively assess the similarity and structural preservation between images before and after each Poisson blending operation, as depicted in Figure 3.5. SSIM is an effective metric for evaluating image quality and is frequently used to measure the resemblance between two images in terms of pixel-level differences, structural coherence, and textures. The findings, displayed in Figure 3.5, show high SSIM values. The highest SSIM value achieved is 98.82%, the lowest stands at 93.03%, and the average is 95.50%. These figures underscore a considerable similarity between the images before and after Poisson blending, especially concerning textures, structure, and intricate details.

We analyze samples both before and after the application of our method, as depicted in Figure 3.6. From this figure, it becomes evident that using our method results in a substantial increase in the sample counts for five categories: Firecrackers, Handcuffs, Nailpolish, Slingshot, and Zippooil. Moreover, aside from Knife and Lighter which are more commonly encountered and thus easier to collect in larger quantities, the sample distribution across categories appears relatively balanced. It's crucial to note that our method is only utilized during the model's training phase, while the evaluation is conducted using the original images.

3.2.3 Re-BiFPN Feature Fusion Module

To improve the network's ability to learn multi-scale features of prohibited items, we propose the Re-BiFPN feature fusion method, which achieves more efficient cross-scale connectivity and weighted feature fusion. The Re-BiFPN includes a recursive connection and a CA-ASPP module.

Figure 3.7a shows the structure of Re-BiFPN. Re-BiFPN connects the layers in



(a) The structure of Re-BiFPN.

(b) Unrolling Re-BiFPN.

Figure 3.7: The structure of Re-BiFPN and the expanded view. The white circles represent the feature maps extracted by backbone. The colored circles indicate the multi-scale features in the Re-BiFPN structure.

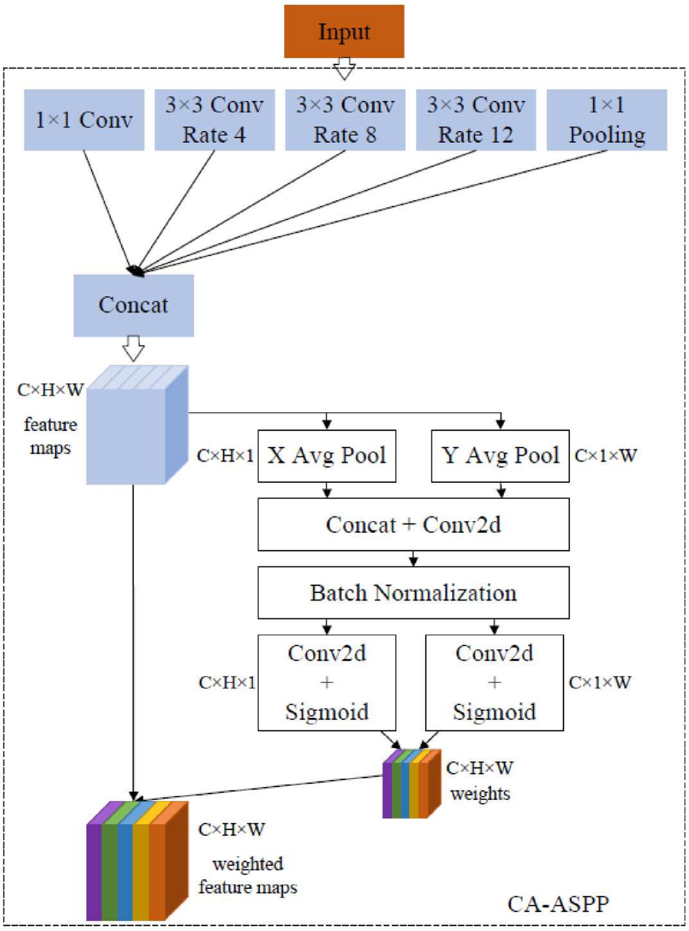


Figure 3.8: The structure of CA-ASPP module.

BiFPN to the bottom-up backbone network through additional recursive connections to form a recursive structure. The red arrow in Figure 3.7a is the recursive connection. Specifically, this recursive connection brings the features with rich multi-scale information back to the lower-level backbone network, which is not rich enough in multi-scale information, thus enhancing the representation of features to achieve efficient cross-scale connectivity and weighted feature fusion. Figure 3.7b is the expanded view of Figure 3.7a.

The structure of CA-ASPP module is shown in Figure 3.8. The CA-ASPP module takes the output features of the first BiFPN structure as input and converts them into the features used in the second bottom-up backbone network in Figure 3.7b. Simultaneously, it captures cross-channel, direction-aware, and position-sensitive information to help the model locate and identify prohibited items.

As shown in Figure 3.8, in addition to $1 \times 1 Conv$ and $1 \times 1 Pooling$, we set up $3 \times 3 Conv$ dilated convolution with the expansion ratio of 4, 8, and 12 for capturing the multi-scale information in the feature maps. And then, the two vectors are obtained by average pooling for horizontal and vertical directions, respectively. These two vectors with embedded direction-aware and position-sensitive information are encoded as two attention maps, each capturing the long-range dependencies of the input feature map along a spatial direction. The Concat operation and BN operation are performed on these two vectors. Next, the split operation is performed and the weights are obtained after the Sigmoid activation function. Finally, the weights are added to the $C \times H \times W$ feature maps.

Mathematical expression of the Re-BiFPN structure. Let P_i^{td} and P_i^{out} denote the intermediate feature layer and the output feature layer of the first BiFPN structure, respectively. *Resize* denotes up-sampling and down-sampling. Both w_i and $w_i^{'}$ denote learnable weights. P_i^{td} and P_i^{out} are calculated according to Eq. 3.1 and Eq. 3.2, respectively. Let R_i denote the feature transformation before connecting the features to the bottom-up backbone network. Let F_i^{td} represent the intermediate feature layer of the second BiFPN structure. Let F_i^{out} represent the output layer features of the second BiFPN structure. Then, the intermediate feature layer and output layer features of the second BiFPN structure can be derived according to Eq. 3.3 and Eq. 3.4, respectively. To prevent the divisor from being zero set ε in the formula to a small constant. The fusion module in Figure 3.7b is used to fuse P_i^{out} and F_i^{out} together. To further improve the efficiency, the feature fusion process of Re-BiFPN uses deeply separable convolution [121].

$$P_i^{td} = \begin{cases} \frac{w_1 P_i^{in} + w_2 Resize(P_{i+1}^{td})}{w_1 + w_2 + \varepsilon} & \text{if } i = 4, 5 \\ \frac{w_1 P_i^{in} + w_2 Resize(P_{i+1}^{in})}{w_1 + w_2 + \varepsilon} & \text{if } i = 6 \end{cases} \tag{3.1}$$

$$P_i^{out} = \begin{cases} \frac{w_1^{'} P_i^{in} + w_2^{'} Resize(P_{i+1}^{td})}{w_1^{'} + w_2^{'} + \varepsilon} & \text{if } i = 3 \\ \frac{w_1^{'} P_i^{in} + w_2^{'} P_i^{td} + w_3^{'} Resize(P_{i-1}^{out})}{w_1^{'} + w_2^{'} + w_3^{'} + \varepsilon} & \text{if } i = 4, 5, 6 \\ \frac{w_1^{'} P_i^{in} + w_3^{'} Resize(P_{i-1}^{out})}{w_1^{'} + w_3^{'} + \varepsilon} & \text{if } i = 7 \end{cases} \tag{3.2}$$

$$F_i^{td} = \begin{cases} \frac{w_1 R_i(P_i^{out}) + w_2 Resize(R_i(F_{i+1}^{td}))}{w_1 + w_2 + \varepsilon} & \text{if } i = 4, 5 \\ \frac{w_1 R_i(P_i^{out}) + w_2 Resize(R_i(P_{i+1}^{out}))}{w_1 + w_2 + \varepsilon} & \text{if } i = 6 \end{cases} \tag{3.3}$$

$$F_i^{out} = \begin{cases} \frac{w_1^{'}R_i(P_i^{out})+w_2^{'}Resize(F_{i+1}^{td})}{w_1^{'}+w_2^{'}+\varepsilon} & \text{if } i = 3 \\ \frac{w_1^{'}R_i(P_i^{out})+w_2^{'}F_i^{td}+w_3^{'}Resize(F_{i-1}^{out})}{w_1^{'}+w_2^{'}+w_3^{'}+\varepsilon} & \text{if } i = 4, 5, 6 \\ \frac{w_1^{'}R_i(P_i^{out})+w_3^{'}Resize(F_{i-1}^{out})}{w_1^{'}+w_3^{'}+\varepsilon} & \text{if } i = 7 \end{cases} \tag{3.4}$$

3.2.4 Rep-CIoU Loss Function

The loss function used in the original cascade network for bounding box regression is Smooth L1 loss, which has some limitations in the prohibited item detection task. When the Smooth L1 loss is used to calculate the bounding box of the target detection, the losses of the four points are first calculated independently and then summed to get the final bounding box loss. Since the evaluation metric for object detection tasks is IoU, and multiple detection boxes may have the same Smooth L1 loss while exhibiting significant differences in IoU, the Smooth L1 loss is not suitable in this case.

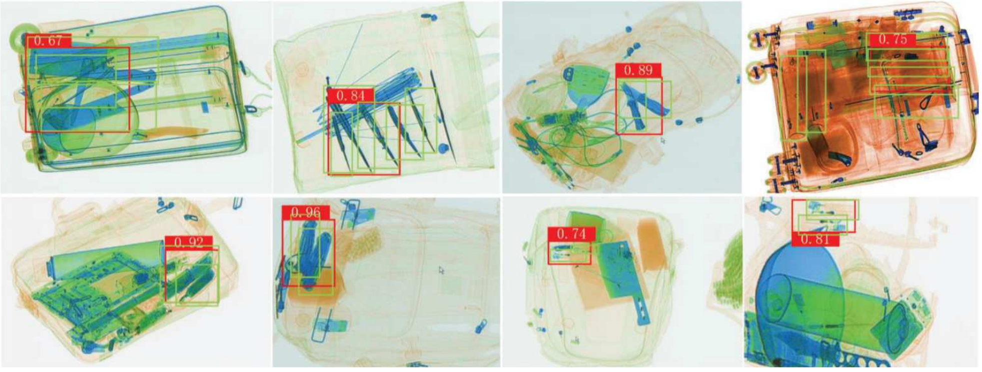


Figure 3.9: Example of visualization of prohibited item detection errors. Green boxes are correctly prediction boxes, while red boxes are false positives caused by overlapping. The confidence scores outputted by detectors are also attached. The errors usually occur when a prediction box shifts slightly or dramatically to a neighboring ground-truth object, or bounds the union of several overlapping ground-truth objects.

In addition, overlapping prohibited items is also a problem that needs attention. As shown in Figure 3.9, in the case of overlapping between multiple targets, the prediction boxes of multiple targets are regressed into one box. The reason is that the NMS algorithm filters out multiple prediction boxes because they are too close. To make each prediction box as close as possible to the ground truth while staying away from the regions of other targets, we propose the Rep-CIoU loss function of Eq. 3.5. Coefficients α and β act as the weights to balance the L_{CIoU} and the L_{Rep}.

$$L_{Rep-CIoU} = \alpha \cdot L_{CIoU} + \beta \cdot L_{Rep} \tag{3.5}$$

The L_{CIoU} loss term is expressed as Eq. 3.6. Where IoU is the ratio of the intersection and union of the prediction box and the ground-truth; b and b_{gt} denote the centroids of the prediction box and the ground-truth, respectively; ρ denotes the Euclidean distance; c denotes the diagonal distance of the minimum outer rectangle of the prediction box and the ground-truth; λ is a positive trade-off parameter, $\lambda = \frac{v}{(1-IoU)+v}$; v denotes the constraint on the geometric relationship of the prediction box, $v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2$, w, h, w^{gt}, h^{gt} represent the height and width of the prediction box and the height and width of the ground-truth, respectively.

The L_{Rep} loss term is expressed as Eq. 3.7. Where $Smooth_{ln}$ is a commonly used regression loss function, and its expression is Eq. 3.8; B^{P_i} and B^{P_j} denote the prediction box for the initial detection box P_i and P_j regressions; $\mathbb{1}$ is an identity function; ε is a small constant set to prevent the divisor from being zero.

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \lambda v \tag{3.6}$$

$$L_{Rep} = \frac{\sum_{i \neq j} Smooth_{ln}(IoU(B^{P_i}, B^{P_j}))}{\sum_{i \neq j} \mathbb{1}[IoU(B^{P_i}, B^{P_j}) > 0] + \varepsilon} \tag{3.7}$$

$$Smooth_{ln} = \begin{cases} -ln(1-x) & x \leq 0 \\ x & x > 0 \end{cases} \tag{3.8}$$

Rep-CIoU loss function considers not only the IoU between multiple prediction boxes but also the centroid distance between prediction box and ground-truth. The L_{Rep} loss term in Rep-CIoU represents the loss value generated between a prediction box and a prediction box that is adjacent and not the same target. Its purpose is to exclude other detection boxes with different targets, making the model more robust to overlapping items. It can be found from Eq. 3.7 that when the IoU distance between the target prediction box P_i and other surrounding prediction boxes P_j is larger, the generated loss is also larger. Therefore, the L_{Rep} loss term can effectively prevent multiple prediction boxes from being filtered out by the NMS algorithm because they are too close to each other, and thus reduce the missed detection due to overlapping.

3.3 Experiments

3.3.1 Datasets

For prohibited item detection, the dataset includes ten types of prohibited items: Knife, Scissors, Lighter, Zippooil, Pressure, Slingshot, Handcuffs, Nailpolish, Powerbank, and Firecrackers. The visual presentation of the dataset is shown in Figure 3.1a. Figure 3.1b shows the images of five (Firecrackers, Handcuffs, Nailpolish, Slingshot, Zippooil) of the ten types of prohibited items after X-ray irradiation alone. Each of these five categories contains 200 images. The dataset comprises a total of 6,400 images, with 5,400 images of entire packages in X-ray and 1,000 images of individual prohibited items in X-ray. For the 5,400 images, the training set accounts for two-thirds and the test set accounts for one-third.

For human detection, the COCO dataset [122] provides a rich collection of annotated images, including the ”person” category, which encompasses various human poses, occlusions, and background variations. Due to its extensive annotations and real-world scene diversity, COCO serves as a benchmark dataset for training and evaluating human detection models. As a large-scale dataset specifically designed for object detection, COCO contains over 200,000 images and 1.5 million object instances covering 80 object categories. It provides high-quality bounding box annotations, making it widely used for training and evaluating deep learning-based object detection models. Compared to earlier datasets like Pascal VOC [123], COCO offers a more diverse set of objects in realistic environments, introducing challenges such as occlusions, scale variations, and cluttered backgrounds, which contribute to improving model generalization.

3.3.2 Evaluation Metric

The mAP (mean Average Precision) is commonly used to evaluate the performance of target detection algorithms. The AP (Average Precision) is used to measure the accuracy of a certain category. The AP of all categories is averaged as mAP, and the expression is Eq.3.9. Where N is the number of categories, AP_c is the AP of category c.

$$mAP = \frac{1}{N} \cdot \Sigma AP_c \tag{3.9}$$

In the COCO dataset, mAP$_{50}$ and mAP$_{75}$ represent the mAP at fixed IoU thresholds of 0.5 and 0.75, respectively. Additionally, the evaluation is conducted based on different object sizes: mAP_S represents small objects (area $< 32^2 = 1024$ pixels), mAP_M represents medium-sized objects (area between 32^2 and $96^2 = 1024$ to 9216 pixels), and mAP_L represents large objects (area $> 96^2 = 9216$ pixels). This comprehensive evaluation helps assess model performance across different object scales.

3.3.3 Implementation Details

The experimental environment in this chapter is shown in Table 3.1. To control the experimental variables, we used a 32-group ResNeXt-101 (32x4d) network as the backbone network with a pre-trained model. We visualize and analyze the aspect ratio of the ground-truth of the training set images as shown in Figure 3.10, so it is appropriate to set the *Anchor_Ratio* parameters in the RPN network to [0.4, 0.6, 0.8, 1.0, 2.0, 3.0].

Table 3.1: Experimental environment parameters

Name	Environment parameters
System	Linux 4.4.0-130-generic x86_64
GPU	TeslaV100-SXM2
RAM	16GB
Framework	Pytorch1.3

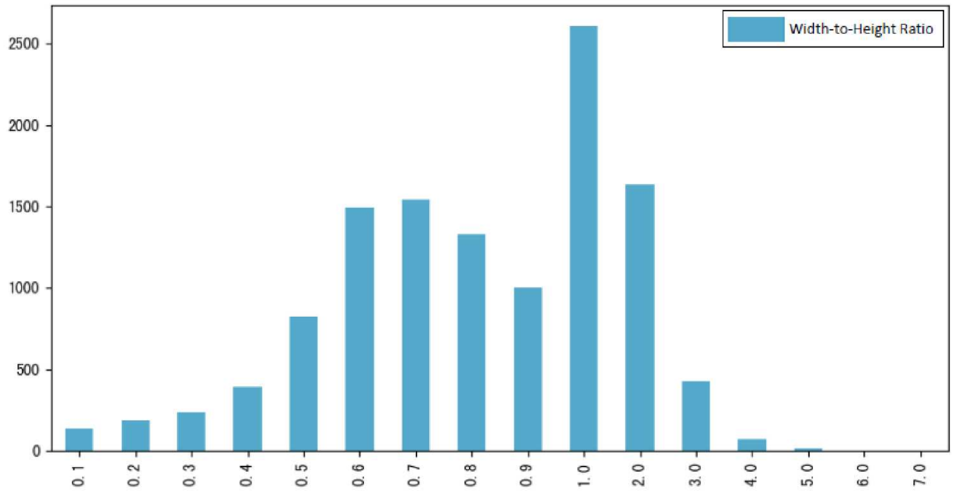


Figure 3.10: Statistics on the number of ground-truth aspect ratios in the dataset. The network parameters are adjusted based on the statistics to make the network more suitable for the dataset.

Table 3.2: Experimental results of comparing different weighting coefficients in Rep-CIoU loss function.

α	β	mAP(%)
0.3	0.7	81.9
0.4	0.6	81.9
0.5	0.5	82.3
0.6	0.4	**82.6**
0.7	0.3	82.2

3.3.4 Experimental Results and Analysis

Before the ablation experiments, we perform parametric experiments of the Rep-CIoU loss function. To verify the best performance of the Rep-CIoU loss, coefficients α and β act as the weights to balance the L_{CIoU} and the L_{Rep}. The parametric experiments are based on the original Cascade R-CNN algorithm combined with the proposed Rep-CIoU loss function to perform comparison experiments with different weighting coefficients. Table 3.2 shows our results with different settings of α and β. It can be concluded from Table 3.2 that different weighting coefficients have different effects on the algorithm accuracy. Empirically, α=0.6, β=0.4 yields the best performance.

Next, to illustrate the impact of our method on detection performance, we set up an ablation experiment with **the original Cascade R-CNN** [116] which employs the FPN and Smooth L1 loss function as the baseline. The evaluation metric is mAP, and the results of the ablation experiments are shown in Table 3.3. It can be seen that our proposed Poisson blending combined with the Canny edge operator method, the Re-BiFPN feature fusion method, and the Rep-CIoU loss function improved 1.5, 1.6, and 0.8 percent, respectively, which seems the summation 3.9 percent could be improved theoretically. Table 3.3 also shows the AP for each category in the ablation experiments. It can be found that the AP of some prohibited items in each ablation experiment has improved, indicating that our method can effectively improve the accuracy of prohibited items.

3.3.4.1 Ablation Studies

Comparison of detection accuracy. Our method is the baseline combination with the Poisson blending method, the Re-BiFPN feature fusion method, and the Rep-CIoU loss function. As can be seen from Table 3.3, the AP has increased in most categories. The APs of Nailpolish and Firecrackers reach above 90%, and the APs of Lighter, Pressure, Slingshot, Handcuffs, and Powerbank also reach above 80%. Although the

Table 3.3: Comparisons of the AP and mAP when adding baseline with the Poisson blending, Re-BiFPN, and Rep-CIoU.

Method	Knife	Scissors	Lighter	Zippooil	Pressure
Baseline	73.2	75.4	82.2	73.1	87.4
Baseline w/ Poisson	75.8	74.4	82.4	81.4	88.5
Baseline w/ Re-BiFPN	75.8	74.1	84.1	**78.7**	88.4
Baseline w/ Rep-CIoU	75.2	75.2	82.5	72.3	88.4
Ours	**78.8**	**78.4**	**84.7**	78.2	**89.4**

Table 3.3: Comparisons of the AP and mAP (Continued).

Method	Slingshot	Handcuffs	Nailpolish	Powerbank	Firecrackers	mAP(%)
Baseline	71.9	87.5	94.9	82.3	90.1	81.8
Baseline w/ Poisson	75.2	**87.6**	100	81.9	86.1	83.3
Baseline w/ Re-BiFPN	72.3	87.3	100	84.3	89.2	83.4
Baseline w/ Rep-CIoU	74.0	87.3	96.1	85.5	89.2	82.6
Ours	**81.0**	87.3	**100**	**87.3**	**90.9**	**85.6**

AP of Knife, Scissors, and Zippooil does not reach 80%, it is still a good improvement compared to the baseline. Compared with the baseline, the mAP of our method gets 85.6%, which is an improvement of 3.8%. Slingshot, Knife, Zippooil, and Nailpolish have the most noticeable improvement, with 9.1%, 5.6%, 5.1%, and 5.1%, respectively. Our method improves the mAP by 3.8 percent compared to the baseline. Although the theoretical improvement, 3.9 percent, is not reached, 3.8 percent can be considered as a reasonable good improvement.

In the comparison between the Baseline and Baseline with Poisson, we note that the accuracy has improved for all categories that had an increase in sample counts, except for Firecrackers. As depicted in Figure 3.11, the volume of Firecrackers is larger than that of other categories, which might be related to its decrease in accuracy. For the categories Knife, Lighter, and Pressure, where the sample counts remained unchanged, their performance has also shown improvement. The enhanced accuracy

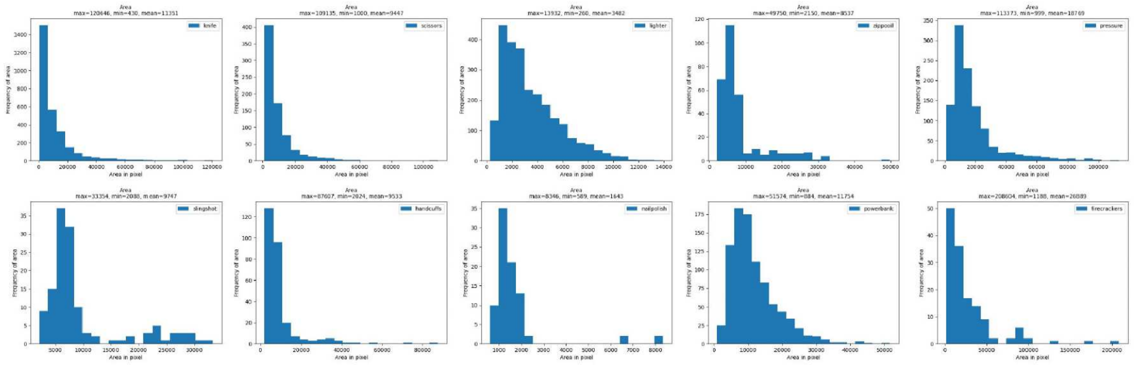


Figure 3.11: Histogram of scale distribution for each category. In the subplots, the horizontal axis represents the pixel area of prohibited items, while the vertical axis represents the frequency. The headers “max”, “min”, and “mean” respectively indicate the maximum, minimum, and average values of prohibited item pixel areas within each category. Please note the scales of the horizontal and vertical axes in each subplot.

for these categories could be attributed to their ample number of samples, and the improved accuracy in other categories likely reduces the risk of misclassification by the model.

In the comparison between Baseline and Baseline with Re-BiFPN, referencing Table 3.3 and Figure 3.11, we note improvements in accuracy for several categories, including Knife, Lighter, Zippooil, Pressure, Slingshot, Nailpolish, and Powerbank. However, there was no observed improvement in accuracy for the three other categories: Handcuffs, Firecrackers, and Scissors. Drawing from the insights provided by Figure 3.6 and Figure 3.11, the decrease in accuracy for Handcuffs and Firecrackers could be attributed to their limited sample counts and a broad range of scales. The efficacy of the Re-BiFPN method might be hindered by this factor, given its need for a significant volume of data to adeptly learn a variety of multi-scale features. The decrease in accuracy for Scissors might be due to complex occlusions between the Scissors and the background and overlapping texture details, as illustrated in Figure 3.1a for Scissors.

Comparison of detection effect. Figure 3.12 shows the detection effect of our method and the baseline. In the image, the closer the yellow box is to the prohibited item, the better the algorithm is at locating the prohibited item. Under the condition that the labels are correct, the scores of the labels are positively correlated with the classification ability of the algorithm. Combining Table 3.3 and Figure 3.12, we can find that the baseline (the original Cascade R-CNN) has missed detections for categories Knife, Lighter, Zippooil, Nailpolish, and Powerbank, and however our

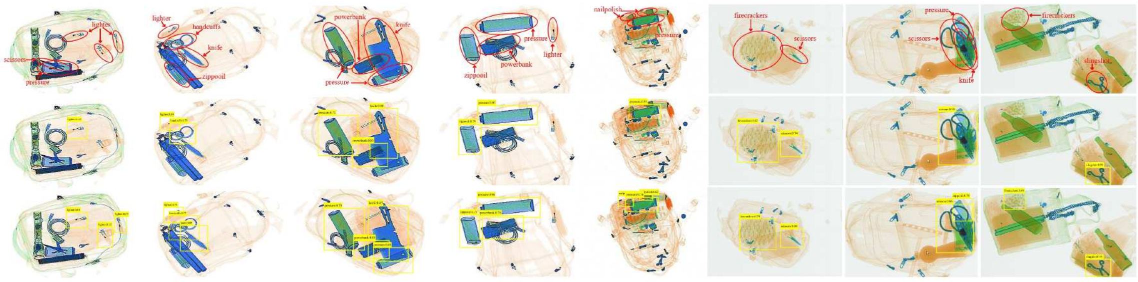


Figure 3.12: Comparison of detection effect between our method and the baseline. The first row shows the input images. For clarity, we artificially highlight the prohibited items in red in the input image. The second row shows the detection effect of the baseline. The third row shows the detection effect of our method.

method can detect all these missed prohibited items. Moreover, even for the categories of Firecrackers, Handcuffs, Pressure, and Scissors that can be detected by the baseline, our method's localization and classification effects are better than the baseline.

In the comparative experiments, we benchmarked against SOTA (State-of-the-Art) methods from various domains. The work by Miao [55] represents the SOTA method in the field of foreground-background separation techniques. Similarly, the method proposed in [124] stands as the SOTA method among single-stage approaches, while Wang's method [54] is recognized as the SOTA method among two-stage approaches. Zhang et al.'s technique [56] has demonstrated noteworthy performance in prohibited item detection tasks. Additionally, the method introduced by Mery [45] is a prominent representative within the domain of machine learning-based methods. These benchmarks enable us to conduct comprehensive comparative analyses to showcase the efficacy of our proposed method.

3.3.4.2 Comparison with Other SOTA Approaches in Prohibited Item Detection

Comparison of detection accuracy. As can be seen from Table 3.4, the detection accuracy of our method is respectively 12.5%, 7.4%, 4.9%, 4.7%, and 4.5% higher than that of the five control groups. Among the five comparison methods, the accuracy of the algorithm proposed by Mery et al. [45] based on machine learning is only 73.1%. In contrast, the detection accuracy of our method in this paper reaches 85.6%.

Comparison of detection effect. Figure 3.13 shows the detection effect of our method and other comparison algorithms. As shown in Figure 3.13, the method proposed by Mery et al. [45] shows a severe wrong detection for Lighter. Moreover, Handcuffs, Nailpolish, Scissors, and Slingshot, have missed detection. Although Pressure and Firecrackers can classify correctly, their prediction boxes do not fully

Table 3.4: Comparison with other methods on prohibited item dataset.

Method	Knife	Scissors	Lighter	Zippooil	Pressure
Mery et al. [45]	65.3	71.8	76.5	61.5	74.8
Zhang et al. [56]	71.7	66.8	78.5	68.5	83.2
Wang et al. [54]	69.4	73.5	81.3	72.6	82.5
Wang et al. [124]	71.7	75.4	78.5	72.6	83.2
Miao et al. [55]	75.4	73.5	77.9	74.5	85.5
Ours	**78.8**	**78.4**	**84.7**	**78.2**	**89.4**

Table 3.4: Comparison with other methods on prohibited item dataset (Continued).

Method	Slingshot	Handcuffs	Nailpolish	Powerbank	Firecrackers	mAP(%)
Mery et al. [45]	72.4	75.5	80.0	78.6	74.6	73.1
Zhang et al. [56]	77.5	86.5	80.0	83.4	86.3	78.2
Wang et al. [54]	80.5	84.7	95.0	82.5	85.2	80.7
Wang et al. [124]	78.4	86.5	90.0	**87.3**	85.2	80.9
Miao et al. [55]	79.6	82.5	95.0	82.5	84.8	81.1
Ours	**81.0**	**87.3**	**100**	**87.3**	**90.9**	**85.6**

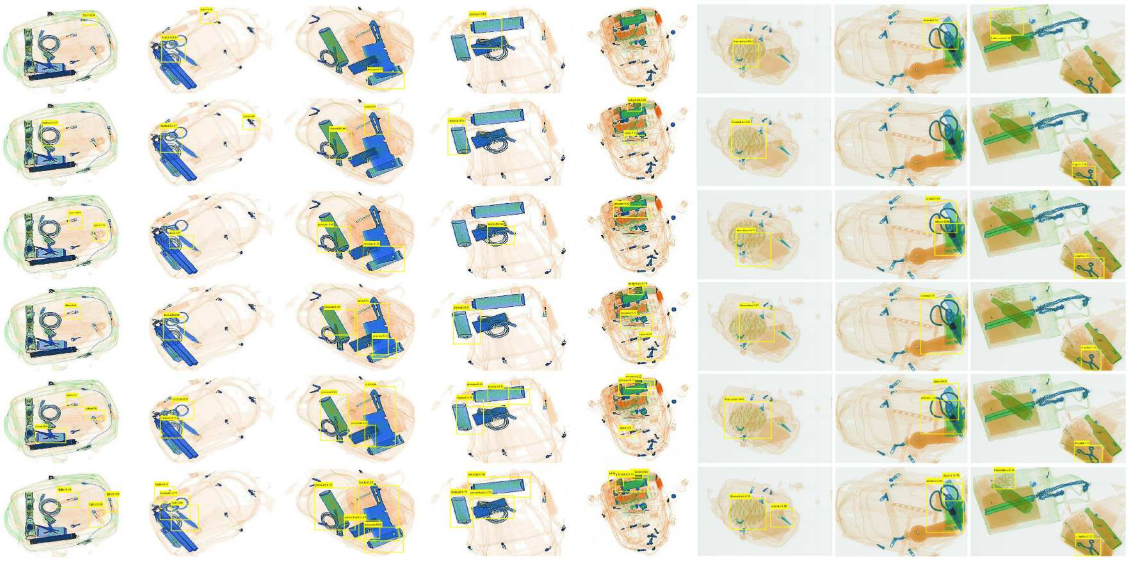


Figure 3.13: Comparison of detection effect between our method and other mainstream algorithms. The input image is shown in the first row of Figure 3.12. The first to fifth rows represent the detection effect of the five control groups in Table 3.4. The last row shows the detection effect of our method.

circle the prohibited items. The method proposed by Zhang et al. [56] also has some wrong and missed detections for Lighter, Handcuffs, Pressure, and Firecrackers. The method proposed by Wang et al. [54] has good detection for Lighter and Scissors. But Handcuffs and Pressure both have some wrong detection. Knife and Firecrackers need to be located precisely. The method proposed by Wang et al. [124] is a representative one-stage target detection algorithm. However, Knife, Lighter, and Pressure have some detected that need to be corrected. The method proposed by Miao et al. [55] has better classification accuracy for Lighter, Scissors, Pressure, Zippooil, and Slingshot. However, it can be seen from the figure that the prediction boxes of Lighter, Pressure, and Zippooil do not accurately locate the location of the prohibited items.

The last row shows the detection effect of our method. It performs better for Lighter, Handcuffs, and Scissors, which are more prone to wrong and missed detection. In addition, our method is more accurate in locating Pressure, Firecrackers, Zippooil, and Pressure. In summary, it can be seen from Table 3.4 and Figure 3.13 that our method has better localization precision and higher classification accuracy in this work.

3.3.4.3 Comparison with Other SOTA Approaches in Human Detection

To validate the effectiveness of our method in human detection, we conducted experiments on the COCO dataset. Notably, for the COCO dataset, we did not apply the data handling steps described in Section 3.2.2.

Comparison of detection accuracy. The experimental results on the COCO dataset are shown in Tables 3.5 and 3.6, which compare our method with other approaches on the validation and test sets, respectively. The evaluation metrics include mean Average Precision (mAP) across different IoU thresholds, as well as AP scores for small, medium, and large objects.

From Table 3.5, it can be observed that our method achieves the highest mAP of 49.3% on the COCO validation set, outperforming all competing methods. Notably, our approach also achieves the best results across all specific IoU thresholds, including mAP_{50} (67.3%) and mAP_{75} (51.7%), indicating superior detection performance across varying levels of localization precision. Additionally, our method performs particularly well in detecting small ($\text{mAP}_S = 29.5\%$), medium ($\text{mAP}_M = 51.8\%$), and large ($\text{mAP}_L = 53.1\%$) objects, demonstrating strong adaptability across different object scales.

Similarly, Table 3.6 presents results on the COCO test set, where our method maintains a leading performance with an overall mAP of 49.5%, again surpassing other approaches. Compared to the best-performing approach, our model achieves a +0.4% improvement in overall mAP, a +1.7% increase in mAP_{50}, and a +0.1% increase in mAP_{75}. The results for small, medium, and large objects further confirm the robustness of our approach across various object sizes.

The superior performance of our method can be attributed to its enhanced feature extraction and localization capabilities, which enable more accurate object detection in complex scenes. The improvements in small-object detection highlight the effectiveness of our model in handling occlusions and low-resolution instances, which are common challenges in real-world scenarios. Overall, these results demonstrate that our approach significantly improves detection accuracy while maintaining strong generalization across different object categories and scales.

Table 3.5: Comparison with other methods on the COCO dataset (validation set).

Method	mAP	mAP_{50}	mAP_{75}	mAP_S	mAP_M	mAP_L
Mery et al. [45]	36.3	52.8	40.5	20.5	40.8	43.1
Zhang et al. [56]	39.5	60.8	49.5	25.5	45.2	47.2
Wang et al. [54]	45.4	63.5	48.3	27.6	48.5	50.7
Wang et al. [124]	48.7	61.4	51.5	27.6	48.2	51.9
Miao et al. [55]	47.4	65.5	50.9	28.5	49.5	51.1
Ours	**49.3**	**67.3**	**51.7**	**29.5**	**51.8**	**53.1**

Table 3.6: Comparison with other methods on the COCO dataset (test set).

Method	mAP	mAP_{50}	mAP_{75}	mAP_S	mAP_M	mAP_L
Mery et al. [45]	37.1	53.3	40.3	20.2	40.2	42.8
Zhang et al. [56]	40.2	61.6	49.1	25.1	44.7	45.5
Wang et al. [54]	45.8	64.1	48.1	27.3	48.1	48.6
Wang et al. [124]	49.1	62.0	51.4	27.5	47.9	48.7
Miao et al. [55]	47.8	66.1	50.6	28.2	49.1	48.2
Ours	**49.5**	**67.8**	**51.5**	**29.4**	**51.2**	**51.1**

Comparison of detection effect. To further evaluate the effectiveness of our method, we compare its human detection results with five other approaches, as shown in Figure 3.14.

In the first column, where a skier is present in a snowy environment, our method exhibits superior detection performance for small and distant objects. Several other methods either fail to detect the person or produce incomplete and inaccurate bounding boxes. In contrast, our approach successfully identifies the person with a well-localized bounding box, highlighting its robustness in handling small-scale objects.

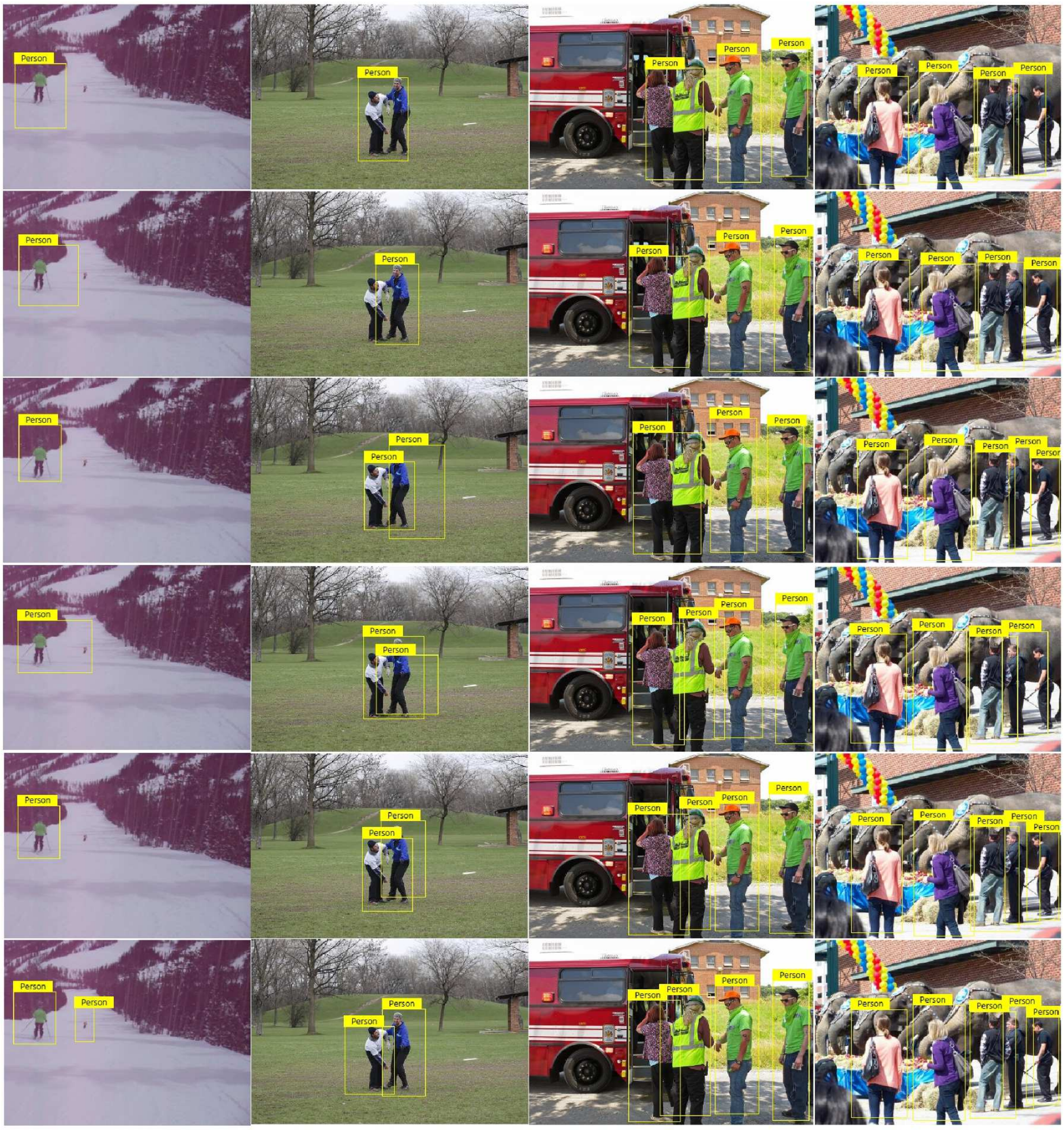


Figure 3.14: Comparison of human detection effect between our method and other approaches. The last row presents the results obtained by our method, while the upper rows correspond to the methods of Mery et al. [45], Zhang et al. [56], Wang et al. [54], Wang et al. [124], and Miao et al. [55], respectively.

The second column presents a scenario where two individuals are standing close together. Some methods struggle to detect both individuals accurately, leading to either missed detections or bounding boxes that fail to properly separate the two persons. Our method, however, correctly distinguishes both individuals, demonstrating better localization and spatial awareness in human detection.

The third and fourth columns feature crowded scenes where multiple people are present, posing challenges such as occlusion and close proximity between individuals. Several existing methods either miss detections or produce incorrect bounding boxes that merge multiple persons into a single detection. Our approach effectively identifies each person, ensuring that the bounding boxes are correctly placed without significant overlap errors. This highlights the advantage of our method in handling occluded individuals and dense environments.

3.4 Conclusion

In this chapter, we discussed three challenges faced in prohibited item detection within X-ray security inspection images: (a) the imbalance distribution of categories, (b) diversity of prohibited item scales, and (c) overlap between items. For (a), we proposed to leverage the Poisson blending algorithm with the Canny edge operator approach to increase the diversity and complexity of the samples. For (b), we proposed the Re-BiFPN feature fusion method, which consists of a CA-ASPP module and a recursive connection. The CA-ASPP module extracts the location information from the multi-scale feature maps. The recursive connection feeds the multi-scale feature maps processed by the CA-ASPP module to the bottom-up backbone layer. For (c), a Rep-CIoU loss function is designed to address the overlapping problem in X-ray images.

In the ablation experiments, our proposed Poisson blending combined with the Canny edge operator method, the Re-BiFPN feature fusion method, and the Rep-CIoU loss function improved 1.5, 1.6, and 0.8 percent, respectively. Comparison experiments show that our method can successfully identify ten kinds of prohibited items such as Knife, Scissors, etc., and achieved 83.4% of mAP, which is superior to the baseline (the original Cascade R-CNN) and other mainstream methods.

In addition, we conducted extensive experiments on human detection using the COCO dataset to verify the effectiveness of our proposed method. The experimental results demonstrate that our approach significantly outperforms previous methods in detecting humans under various challenging conditions, such as small-scale objects, occlusion, and dense crowds. Specifically, our model achieved an mAP of 49.5% on the COCO test set, surpassing other state-of-the-art methods. The qualitative analysis further confirms that our model effectively distinguishes individuals in crowded scenes, correctly localizes small and distant persons, and improves detection precision. These advantages highlight the superiority of our method in real-world human detection applications and provide a solid foundation for the human detection step in human action recognition systems.

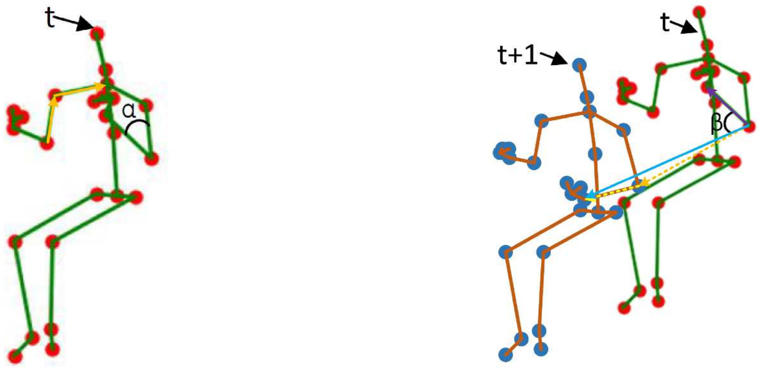
Chapter 4

Skeleton Temporal Fusion Graph Convolutional Network

In this chapter, we take the most challenging case of elderly action recognition as a representative example to analyze the problems and challenges in action recognition. Elderly action recognition is particularly difficult because many elderly individuals exhibit small movement amplitudes and long action durations. To address these challenges, we propose a novel Skeleton Temporal Fusion Graph Convolutional Network (STF-GCN) for skeleton-based action recognition, which effectively models advanced temporal feature representations. Specifically, the STF-GCN employs three encoding strategies to integrate two types of temporal feature representations. Additionally, we introduce a Skeleton Temporal Fusion (STF) module that emphasizes temporal feature representations by alternating between large and small kernel convolutions, enabling various effective receptive fields. Finally, we validate the effectiveness of the proposed method through extensive experiments on both elderly action recognition and general action recognition tasks, demonstrating its robustness and adaptability.

4.1 Introduction

Skeleton-based action recognition (SAR) holds significant importance within the field of human action recognition. The acquisition of precise 3D skeleton data has been greatly facilitated by sensors, such as Microsoft Kinect [17] and human pose estimation algorithms [73, 125]. Skeleton data offers numerous advantages, including its concise representation, computational efficiency and robustness against variations in human appearance [126]. Due to these unique characteristics, SAR has garnered increased attention among researchers.



(a) Spatial dimension features.

(b) Temporal dimension features.

Figure 4.1: Illustration of spatial and temporal dimension features. In (a), the red dot signifies the joint feature, while the orange vector denotes the bone feature. α represents the spatial angle feature at frame t. In (b), the blue vector indicates the temporal motion feature, derived from both the orange vector (indicating the joint's displacement between frame t and t+1) and the yellow vector (showing the bone feature in frame t+1). β is the temporal angle feature, formed by the temporal motion feature and the bone feature at frame t (the purple vector).

Elderly action recognition is a more challenging task than general action recognition [128] due to the fact that many elderly people move with small amplitude and long duration of actions, making many otherwise distinctly different actions similar. Advancements in skeleton-based graph convolutional networks have progressed from spatial node analysis [34], to adding bone vectors [35], and finally, to integrating spatial angles for richer spatial relationship insights [104]. However, while previous methods have made significant advancements in spatial encoding, as illustrated in Figure 4.1a, they often do not sufficiently account for the critical temporal dimension details. This oversight becomes particularly significant in the context of elderly action recognition, where the small amplitude of movements necessitates a more detailed and comprehensive understanding of temporal dimension features, such as temporal motion and temporal angle (as depicted in Figure 4.1b), to accurately capture the nuances between different elderly actions. The nuances refer to the subtle distinctions and fine-grained details between different elderly actions.

For actions of the elderly, which typically have a prolonged duration, the long-range temporal dependencies are crucial for accurate classification. In many existing research methodologies [30, 34, 35, 104], due to the fixed temporal kernel sizes across all layers, there is a tendency to prioritize short-range temporal dependencies, consequently often overlooking the importance of modeling long-range temporal dependencies within extended frame sequences. This limitation confines these methods to a narrow temporal scale, potentially failing to fully exploit the characteristics of the temporal dimension. For instance, in the action sequence "taking off a jacket" (as depicted in Figure 4.2), the similarity in joint coordinates from one frame to the next makes the discernment of subtle differences extremely challenging. In contrast, long-range temporal dependencies, encompassing features of the temporal dimension, provide a holistic view of the action and play a key role in offering comprehensive insights.

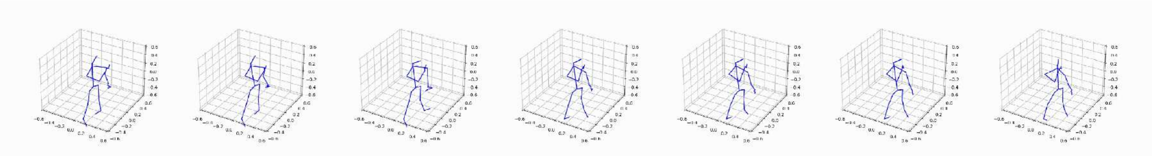


Figure 4.2: For clarity, we visualized a partial sequence of "taking off a jacket". We observed a similarity in joint coordinates between adjacent frames. However, by expanding the temporal receptive field to compare the first and last frames, the differences become more pronounced.

In this chapter, we propose a Skeleton Temporal Fusion Graph Convolutional Network (STF-GCN), distinguished by its innovative Temporal Dimension Encoding (TDE) and five Skeleton Temporal Fusion (STF) modules, each specifically designed for TDE. By "harnessing time", we emphasize our model's strategic focus on capturing and analyzing temporal dynamics essential for action recognition. TDE integrates temporal motion and angle features into the graph convolutional network, employing three encoding strategies, general, reinforced and united. These features and strategies enable the model to intricately extract and analyze complex motion patterns and information from the temporal dimension of skeleton sequences. Consequently, our model boasts an enhanced ability to discern subtle action differences and exhibits increased sensitivity to motion nuances. Further enhancing the framework, each STF module includes an alternating structure of large and small kernel convolutions, aimed at fully exploiting the potential of TDE while expanding the effective receptive field. This design facilitates the advanced integration of various features and significantly boosts the model's capability to perceive complex motion patterns and handle long-range temporal dependencies, thereby improving action recognition capabilities.

Our main contributions can be summarized as follows: (1) We propose two temporal dimension features, namely temporal motion and angle features, and three distinct encoding strategies, all aimed at facilitating a more detailed and comprehensive understanding of skeletal data. (2) We propose a STF-GCN that integrates our innovative STF module, which emphasize temporal feature representations to significantly enhance nuanced motion perception and effectively capture spatio-temporal patterns across a variety of ranges. (3) Evaluation on both elderly action recognition and general action recognition tasks demonstrates the outstanding performance of STF-GCN, particularly in distinguishing similar actions.

4.2 Proposed Skeleton Temporal Fusion Graph Convolutional Network

We first present the overall architecture of our STF-GCN. Next, we introduce and formulate the proposed TDE. Finally, we provide details about the main components of STF-GCN.

4.2.1 Overall Architecture

The STF-GCN, our proposed architecture depicted in Figure 4.3, leverages the temporal and spatial dimensions of skeletal data to advance action recognition. Central to our approach is the detailed representation of the human skeleton across successive frames, enhancing our understanding of both static postures and dynamic joint movements. We introduce a dual encoding scheme: the TDE delineates the progression of actions over time (elaborated in Section 4.2.2), while the Spatial Dimension Encoding (SDE) assesses the joint configuration within each frame [34, 35, 104]. Together, these encoded dimensions form the input to the architecture's main processing stream, illustrated in Figure 4.3 and detailed in Section 4.2.3, which consists of five STF modules for refining feature representations toward precise action classification. Finally, the action categories are obtained through a Global Average Pooling (GAP) layer and a Fully Connected (FC) layer. Our proposed method is applicable for recognizing actions of elderly individuals, which are characterized by slow execution and prolonged durations.

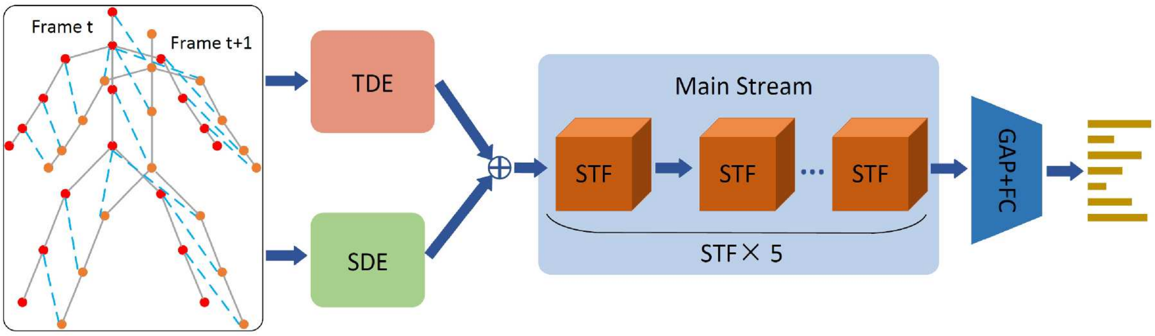


Figure 4.3: The overall architecture of STF-GCN. The TDE, i.e. temporal dimension encoding, which captures temporal dependencies, is elaborated in Section 4.2.2. The SDE, i.e. spatial dimension encodig, is dedicated to representing joint [34], bone [35] and spatial angle [104] features. The intricacies of the main stream, which includes five STF modules that process these encoded features, are detailed in Section 4.2.3.

4.2.2 Temporal Dimension Encoding

In this part, we delineate our proposed TDE, which incorporates two temporal dimension features and three distinct encoding strategies.

4.2.2.1 Temporal Motion and Angle Features

We define $J = (x, y, z)^T$ as a 3D joint coordinate. In frames t and $t + 1$, we consider four joints o^t, o^{t+1}, p^t and p^{t+1}. The proposed temporal dimension features comprise the following two main components.

Temporal motion feature. The vector $\overrightarrow{b}_{o^t,o^{t+1}} = (x_{o^{t+1}} - x_{o^t}, y_{o^{t+1}} - y_{o^t}, z_{o^{t+1}} - z_{o^t})$ indicates the displacement and direction information from o^t to o^{t+1}, as depicted by the orange vector in Figure 4.1b. Similarly, the vector $\overrightarrow{b}_{o^{t+1},p^{t+1}} = (x_{p^{t+1}} - x_{o^{t+1}}, y_{p^{t+1}} - y_{o^{t+1}}, z_{p^{t+1}} - z_{o^{t+1}})$ represents the bone feature between o^{t+1} and p^{t+1} in frame $t + 1$, which is represented by the yellow vector in Figure 4.1b. We define the temporal motion feature of joint o at frame t as M_o^t (depicted as the blue vector in Figure 4.1b), which is formulated in Eq.4.1. Given that each sequence comprises F frames, we set the temporal motion feature of the F-th frame to 0.

$$M_o^t = \begin{cases} \overrightarrow{b}_{o^t,o^{t+1}} + \overrightarrow{b}_{o^{t+1},p^{t+1}} & \text{if } t < F \\ 0 & \text{if } t = F \end{cases} \tag{4.1}$$

Through the temporal motion features, we capture not only the directional movement from joint o^t to o^{t+1} but also from o^{t+1} to p^{t+1}. This approach offers a more comprehensive understanding of the relative motions between joints in the skeletal sequence, effectively detailing the evolving relationships and dynamic changes among them.

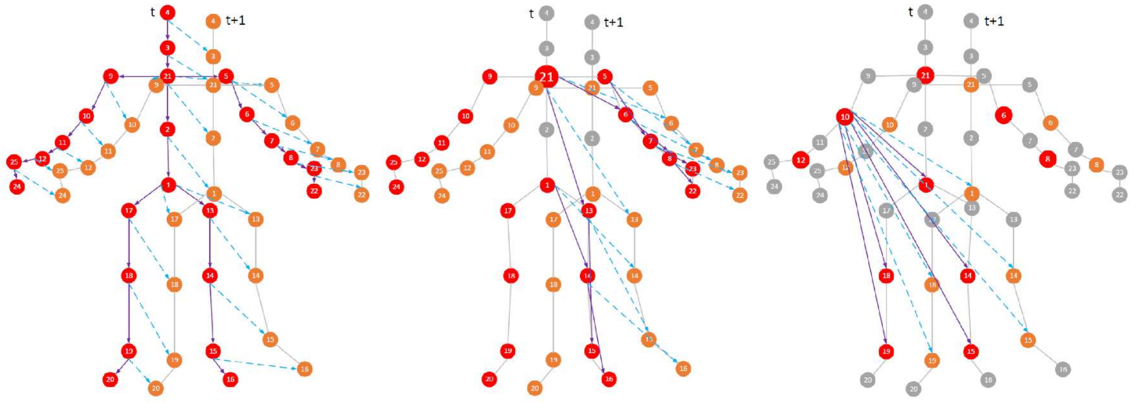
Temporal angle feature. Similar to the vector $\overrightarrow{b}_{o^{t+1},p^{t+1}}$, the vector $\overrightarrow{b}_{o^t,p^t}$ (depicted as the purple vector in Figure 4.1b) represents the bone feature between o^t and p^t in frame t. The vector $M_o^t = \overrightarrow{b}_{o^t,p^{t+1}}$ is the aforementioned temporal motion feature. Let β denote the angle between the two vectors, $\overrightarrow{b}_{o^t,p^t}$ and $\overrightarrow{b}_{o^t,p^{t+1}}$. We define the temporal angle feature $A(o^t)$ for joint o^t as shown in Eq.4.2.

$$A(o^t) = \begin{cases} 1 - \cos\beta = 1 - \dfrac{\overrightarrow{b}_{o^t,p^t} \cdot \overrightarrow{b}_{o^t,p^{t+1}}}{|\overrightarrow{b}_{o^t,p^t}| \cdot |\overrightarrow{b}_{o^t,p^{t+1}}|} & \text{if } t < F \\ 0 & \text{if } t = F \end{cases} \tag{4.2}$$

As β increases from 0 to π radians, the feature value consistently and continuously increases without any decreasing or fluctuating trends. This inherent monotonic relationship with β holds potential advantages for action recognition. By considering the continuous and consistent feature variations with angle β, the model can better capture the underlying motion patterns and body poses associated with different actions.

4.2.2.2 Encoding Strategies

Having clarified the two temporal dimension features, we now turn our attention to the proposed three distinct encoding strategies, specifically designed to harness the full potential of the temporal dimension features.



(a) General encoding of human skeleton.

(b) Reinforced encoding of human limbs.

(c) United encoding of human limbs.

Figure 4.4: Temporal dimension features using three distinct encoding strategies. Red and orange dots represent skeletal joints at frames t and $t+1$, respectively. Blue vectors symbolize temporal motion features; angles between blue and purple vectors are temporal angle features. (b) For clarity, features of only the right arm and leg are depicted. (c) Visualized features between joint No. 10 of the upper body and the joints of the lower body.

General encoding of human skeleton. As depicted in Figure 4.4a, the general encoding of the human skeleton is rooted in the natural connections between skeletal joints. By analyzing the temporal dimension relationships of each joint across two distinct frames, this encoding strategy comprehensively captures the subtle temporal movement and angle variations that spatial dimension features may overlook. Specifically, the temporal angle is formed by the vector (shown as the purple vector) that connects a joint at frame t to its neighboring joints, and the temporal motion feature (portrayed as the blue vector) pertinent to that joint.

Reinforced encoding of human limbs. Different actions typically manifest distinct limb motion patterns and positional variations. Additionally, the positions and movements of human limbs often convey semantic cues related to the performed actions.

Algorithm 1 Reinforced encoding of human limbs

```
Require: The limb joints of the human body at frame t, $R_1^t, R_2^t, R_3^t, R_4^t$   ▷ The right
    arm from joint 21 to 22, the left arm from joint 21 to 24, the right leg from joint
    21 to 16, and the left leg from joint 21 to 20.
Ensure: Reinforced encoding features
for i = 1to 4 do                                  ▷ Iterate through $R_1^t, R_2^t, R_3^t, R_4^t$
    for j = 1 to len($R_i^t$) - 2 do              ▷ j controls vector head
        for k = 3 to len($R_i^t$) do              ▷ k controls vector tail
            $joint1 \leftarrow R_{i,j}^t$
            $joint2 \leftarrow R_{i,j}^{t+1}$
            $joint3 \leftarrow R_{i,j+k}^t$   ▷ Iterate from the third joint, the features between
    the joint No. 21 and the second joint of the limb belong to the general encoding
            $joint4 \leftarrow R_{i,j+k}^{t+1}$
            $vector1 \leftarrow GetVector(joint1, joint3)$
            $vector2 \leftarrow GetMotionVector(joint1, joint2, joint4)$   ▷ Temporal
    motion
            $temporal\ angle \leftarrow GetAngle(vector1, vector2)$   ▷ Temporal angle
        end for
    end for
end for
```

Therefore, we introduce the reinforced limb encoding scheme, designed to capture the intricate characteristics and semantic nuances inherent in limb motions. As depicted in Figure 4.4b, we designate joint No. 21 as the root node and select the limb joints of the human body (shown as red and orange dots) to constitute four joint sequences: left arm, right arm, left leg and right leg. For a more detailed examination using two distinct frames as an example, refer to Algorithm 1. Algorithm 1 is suitable for actions dominated by either the upper limbs or lower limbs.

United encoding of human limbs. Human actions typically involve intricate interactions and coordination among various limbs. Furthermore, the synchronized motions of these limbs can indicate the consistency and coherence of specific actions. To capture these inter-limb relationships and enhance the model's capacity to distinguish between diverse actions, we introduce the united encoding approach for human limbs. As depicted in Figure 4.4c, considering the shoulder, elbow and wrist joints account for the primary range of motion in the upper limbs, we select joint No. 21 from the upper body, accompanied by both elbow and wrist joints, as representative of the upper body's movements. Similarly, we choose the joint No. 1, along with both knee and ankle joints, to represent the lower body's motions. Subsequently, we apply united encoding to these selected joint sequences. For a detailed examination using

Algorithm 2 United encoding of human limbs

```
Require: The limb joints of the human body at frame t, $U_1^t, U_2^t$   ▷ $U_1$ represents the upper body joints, including joints No. 12, No. 10, No. 21, No. 6 and No. 8. $U_2$ represents the lower body joints, including joints No. 19, No. 18, No. 1, No. 14 and No. 15.
Ensure: United encoding features
for i = 1 to len($U_1^t$) do   ▷ i controls upper body list
    $joint1 \leftarrow U_{1,i}^t$
    $joint2 \leftarrow U_{1,i}^{t+1}$
    for k = 1 to len($U_2^t$) do   ▷ j controls lower body list
        $joint3 \leftarrow U_{2,j}^t$   ▷ Iterate from the third joint, the features between the joint No. 21 and the second joint of the limb belong to the general encoding
        $joint4 \leftarrow U_{2,j}^{t+1}$
        $vector1 \leftarrow GetVector(joint1, joint3)$
        $vector2 \leftarrow GetMotionVector(joint1, joint2, joint4)$   ▷ Temporal motion
        $vector3 \leftarrow GetVector(joint3, joint1)$   ▷ The direction is opposite to vector1
        $vector4 \leftarrow GetMotionVector(joint4, joint2, joint1)$   ▷ The direction is opposite to vector2
        $temporal\ angle1 \leftarrow GetAngle(vector1, vector2)$
        $temporal\ angle2 \leftarrow GetAngle(vector3, vector4)$   ▷ Temporal angle
    end for
end for
```

two distinct frames as an example, refer to Algorithm 2. Algorithm 2 is designed for actions involving coordinated movements of all limbs.

4.2.3 Skeleton Temporal Fusion Module

STF module architecture. As depicted in Figure 4.5, the STF module is meticulously designed to process input features through a multi-layered architecture. The input, represented by a tensor of dimensions $C \times T \times V$, undergoes successive transformations by the STF module.

The STF module, central to the feature refinement process, starts with a Spatial Graph Convolution (SGC) [24], which meticulously captures the spatial relationships between joints within frames, delineating the complex interactions in skeletal structures. Following the SGC, the module integrates two Temporal Receptive Field (TRF) blocks, each designed to leverage temporal relationships across multiple frames, facilitating the analysis of movement evolution over time. Integrated with these TRF blocks are residual connections that guarantee the retention of crucial information. As data progresses through the five cascaded STF modules, the evolving feature map

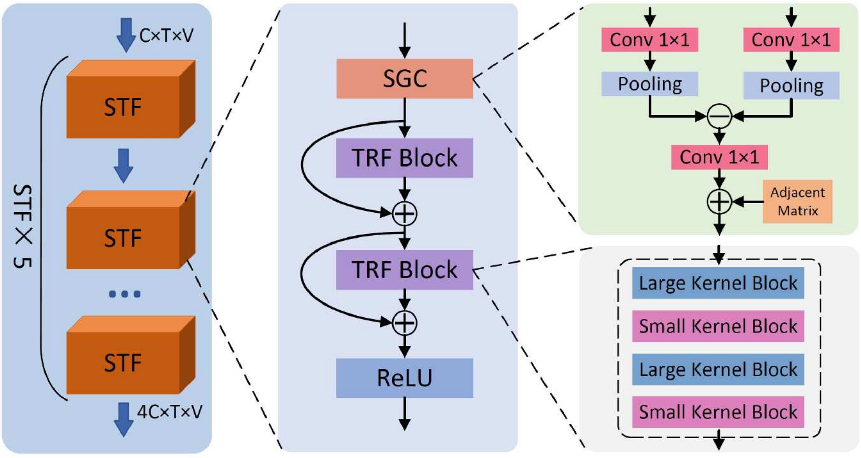


Figure 4.5: Overview of STF module. $C \times T \times V$ represents the dimensions of the input features, where C stands for the number of channels, T for the number of temporal frames and V for the number of joints. The principal components of STF encompass the Spatial Graph Convolution (SGC) [24], the Temporal Receptive Field (TRF) block. $\oplus$ represents concatenation. $\ominus$ represents pairwise subtraction.

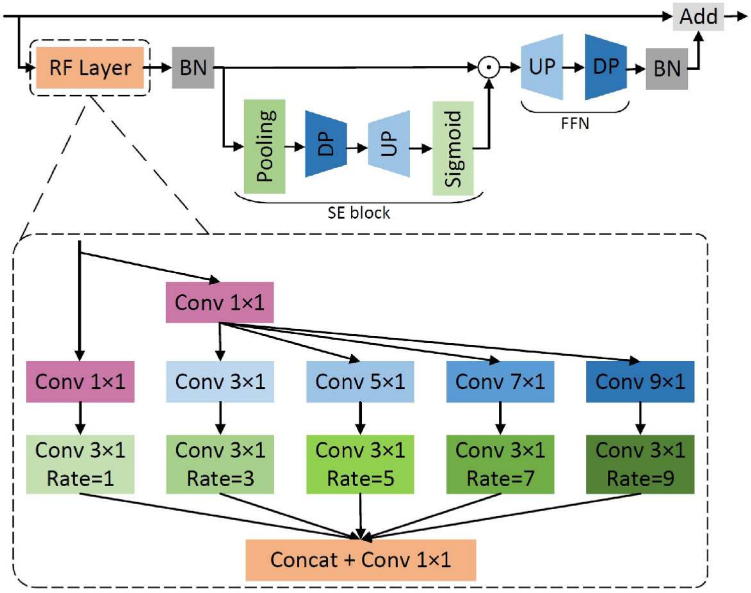


Figure 4.6: Architectural design of Large Kernel Block. It comprises an Receptive Field (RF) layer, an SE (Squeeze-and-Excitation) block [127], an FFN (Feed Forward Network) and BN (Batch Normalization). The only difference between a small kernel block and a large kernel block is that the former replaces the RF layer with a Conv 3×1. The symbol $\odot$ represents element-wise multiplication.

dimensions highlight the module's multi-scale processing ability, essential for identifying complex movement patterns over time and discerning subtle differences in motion.

TRF block architecture. Within each TRF block, as shown on the right side of Figure 4.5, large kernel and small kernel blocks are arranged in an alternating cascading sequence. This configuration, supported by empirical evidence, advocates for the combined use of large and small kernel convolutions. The small kernels excel at identifying fine-grained and small-scale temporal patterns. Conversely, large kernels are pivotal for capturing sparse and extensive temporal features, vital for analyzing time series with significant relational dependencies.

The large kernel block is illustrated in Figure 4.6. The only difference between a small kernel block and a large kernel block is that the former replaces the RF layer with a Conv 3×1. Initially, the RF layer applies a convolution operation to the input data. The sequence of convolutions, starting from Conv 1×1 to Conv 9×1, with increasing dilation rates from 1 to 9, allows for an expansive temporal analysis, covering both immediate and extended temporal contexts. The parallel pathways process the input through Conv 3×1 kernels at varying dilation rates, enabling the block to assimilate temporal features with different granularities. SE block [127] is an efficient structure that perform both inter-channel communications and temporal aggregations to increase the depth.

4.3 Experiments

To comprehensively evaluate the effectiveness of our proposed method, we conducted experiments on the ETRI-Activity3D (EA3D) dataset [2], which focuses on elderly action recognition and presents unique challenges such as small motion amplitudes and long action durations. To further demonstrate the generalization capability of our approach, we extended our investigations to the NTU-RGB+D 120 (NTU120) dataset [128], a widely used benchmark for general action recognition known for its diversity and representativeness. This dual evaluation allows us to validate the effectiveness of our method across both elderly and general action recognition tasks.

4.3.1 Datasets

EA3D is currently the largest elderly action recognition dataset collected in real-world surveillance environments, designed specifically to address the challenges of recognizing actions performed by elderly individuals. It comprises 112,620 samples collected from 100 participants using 8 synchronized sensors. The dataset is categorized into 55 action classes, which encompass a wide range of activities such as individual actions (e.g., walking, sitting down, standing up), human-object interactions (e.g., using a walker, picking up objects), and multiperson interactions (e.g., helping others, social greetings). These categories are tailored to capture the nuanced and subtle motions often observed in elderly populations, making it particularly valuable for research focused on elderly care and monitoring. To maintain consistency and reproducibility, we followed the data selection criteria established by Jang et al. [2] for training and testing.

NTU120 dataset [128] is one of the most widely used large-scale skeleton-based action recognition datasets, known for its extensive coverage of diverse general actions. It is an extension of the NTU60 dataset [129], with 57,367 additional skeleton sequences, bringing the total to 113,945 samples across 120 action classes. These actions encompass a wide spectrum of activities, including daily activities (e.g., eating, drinking), mutual actions (e.g., shaking hands, hugging), health-related actions (e.g., sneezing, falling down), and interaction with objects (e.g., reading, writing). The dataset is captured from 106 subjects and 32 different camera setups. To provide a comprehensive evaluation, NTU120 introduces two standard benchmarks: the Cross-Subject (X-sub120) and Cross-Setup (X-set120) settings. In the Cross-Subject benchmark, the 106 subjects are divided into training and evaluation sets, with 63,026 samples used for training and 50,922 samples for evaluation. This partitioning ensures that actions performed by different individuals are represented in both sets. In the Cross-Setup benchmark, videos recorded from 16 different camera setups are divided into 54,471 training samples and 59,477 evaluation samples. This configuration is designed to evaluate the model's robustness to variations in viewpoints and environmental settings.

4.3.2 Evaluation Metric

To evaluate the performance of our action recognition model, we adopt Accuracy (Acc) as the evaluation metric, which is widely used in action recognition tasks. Acc Eq.4.3 measures the proportion of samples for which the model's highest confidence prediction matches the ground truth label.

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \tag{4.3}$$

4.3.3 Implementation Details

We use SGD optimization with 0.1 as the base learning rate and a weight decay of 0.0005. All the models are trained on two GeForce RTX 3090 with a batch size of 200, using ReduceLROnPlateau to update the learning rate. For the datasets under consideration, the EA3D and NTU120, the number of epochs were configured to 100 and 200, respectively. Following the data preprocessing protocol established by [105, 22], we set the value of F in Eq.4.1 and Eq.4.2 to 64, a setting that is applicable for both elderly and general actions.

4.3.4 Experimental Results and Analysis

4.3.4.1 Ablation Studies

As detailed in Section 3.2, we proposed two temporal dimension features, temporal motion feature (denoted as Mot) and temporal angle feature (denoted as Ang), and three distinct encoding strategies, general encoding, reinforced encoding and united encoding. We employ the suffixes -A, -B and -C to denote these three encoding strategies, respectively. For instance, Mot-A signifies the temporal motion feature using general encoding, whereas MotAng-B denotes the combination of both temporal motion and angle features via the reinforced encoding strategy. We systematically evaluate the contributions of proposed TDE by initially removing this component

Table 4.1: Ablation study of temporal motion feature on EA3D.

STF-GCN	Acc. (%)
w/o TDE	92.0 (-)
+ Mot-A	92.1 (↑0.1)
+ Mot-B	91.9 (↓0.1)
+ Mot-C	91.8 (↓0.2)
+ Mot-A + Mot-B	**92.2 (↑0.2)**
+ Mot-B + Mot-C	92.1 (↑0.1)
+ Mot-A + Mot-C	92.1 (↑0.1)
+ Mot-A + Mot-B + Mot-C	**92.2 (↑0.2)**

from our STF-GCN model, followed by the gradual reintroduction of various temporal dimension features and encoding strategies to assess their individual impacts.

Effect of temporal motion feature. We initially evaluated the efficacy of our proposed temporal motion feature using various encoding strategies, with the results presented in Table 4.1. While employing either -B or -C independently led to a decline in accuracy, a remarkable improvement was observed when the two were combined. This suggests that the synergy between -B and -C captures a richer and more comprehensive representation of the temporal motion feature. Notably, the model that incorporated all three encoding strategies (-A, -B and -C) achieved the highest performance.

Effect of temporal angle feature. Similarly, we assessed the effectiveness of our proposed temporal angle feature using various encoding strategies, as shown in Table 4.2. From a performance enhancement standpoint, the temporal angle feature offers richer and more comprehensive action recognition information than the temporal motion feature, leading to a more significant improvement in accuracy. Furthermore, both the model that incorporated all three encoding strategies and the one that combined -B and -C achieved optimal performance.

Synergistic effect of temporal features. To verify the efficacy of the two temporal dimension features under various encoding strategies, we conducted the experiments outlined in Table 4.3. Similar to results from separate experimental contexts, models employing a single encoding strategy displayed slightly inferior performance compared to those integrating two encoding strategies. Furthermore, incorporating all three encoding strategies provided a modest performance increase over models using just two, achieving an accuracy of 92.9%.

Table 4.2: Ablation study of temporal angle feature on EA3D.

STF-GCN	Acc. (%)
w/o TDE	92.0 (-)
+ Ang-A	92.3 (↑0.3)
+ Ang-B	92.2 (↑0.2)
+ Ang-C	92.1 (↑0.1)
+ Ang-A + Ang-B	92.4 (↑0.4)
+ Ang-B + Ang-C	**92.5 (↑0.5)**
+ Ang-A + Ang-C	92.1 (↑0.1)
+ Ang-A + Ang-B + Ang-C	**92.5 (↑0.5)**

Table 4.3: Ablation study of temporal motion and angle features on EA3D.

STF-GCN	Acc. (%)
w/o TDE	92.0 (-)
+ MotAng-A	92.5 (↑0.5)
+ MotAng-B	92.3 (↑0.3)
+ MotAng-C	92.0 (↑0.0)
+ MotAng-A + MotAng-B	92.7 (↑0.7)
+ MotAng-B + MotAng-C	92.7 (↑0.7)
+ MotAng-A + MotAng-C	92.2 (↑0.2)
+ MotAng-A + MotAng-B + MotAng-C	**92.9 (↑0.9)**

Subsequently, we systematically evaluate the effect of each component within the proposed STF-GCN.

Effect of TRF block design. Table 4.4 consolidates the results of different TRF block designs, which involved a variety of stacking sequences of large kernel blocks and small kernel blocks, to scrutinize their collective impact on model performance. Our model's design adheres to empirically proven evidence and the guidelines elucidated in [130, 131]. The experimental outcomes indicate that the L-S-L-S configuration surpasses other variations, achieving the highest accuracy on EA3D dataset. This optimal sequence underscores the effectiveness of alternating expansive receptive fields of large kernel blocks with the fine-grained sensitivity of small kernel blocks.

Effect of different kernel sizes and dilated rates. We systematically altered the kernel sizes and dilation rates, adopting configurations ranging from smaller sets (1, 3, 5) to more expansive combinations (1, 3, 5, 7, 9, 11), in pursuit of optimizing the

Table 4.4: Performance comparison of different TRF block designs on EA3D. L: Large kernel block; S: Small kernel block.

Model	Params.	Acc. (%)
L	1.97M	91.6
S	2.02M	91.0
S-L	2.34M	91.8
L-S	2.34M	92.2
L-S-S	2.71M	92.4
L-S-S-S	3.08M	92.4
L-L-S	2.66M	92.5
L-S-L-S	**3.03M**	**92.9**
L-S-S-L-S-S	3.77M	92.7
L-L-L-L	2.93M	91.2
S-S-S-S	3.13M	92.3

Table 4.5: Performance comparison of RF layer with different kernel sizes and dilated rates on EA3D.

Kernel	Rate	Params.	Acc. (%)
1,3,5	1,3,5	3.10M	92.4
1,3,5,7	1,3,5,7	3.07M	92.6
1,3,5,7,9	**1,3,5,7,9**	**3.03M**	**92.9**
1,3,5,7,9,11	1,3,5,7,9,11	3.00M	92.5
5,7,3,3,3	1,2,3,4,5	3.05M	92.1

model's receptive field. Table 4.5 reveals that our configuration with kernel sizes and dilation rates of 1, 3, 5, 7 and 9 outperformed not only the various configurations we tested but also surpassed the default setting in [130], which utilized a sequence of 5, 7, 3, 3, 3. The superior performance of our optimal kernel configuration underscores the significance of fine-tuning the receptive field to match the temporal complexity inherent in the skeletal movements.

Effect of the number of TRF blocks. We explore the impact of the number of TRF blocks. As presented in Table 4.6, the model equipped with two TRF blocks exhibits the best performance. When the number of TRF blocks increases beyond this configuration, a decline in performance is observed. This degradation may be attributed to the additional TRF blocks extracting irrelevant information, thereby introducing redundancy in the capture of temporal dependencies.

Table 4.6: Performance comparison of the number of TRF blocks on EA3D.

TRF	Params.	Acc. (%)
1	2.34M	92.0
2	**3.03M**	**92.9**
3	3.72M	92.5

Table 4.7: Performance comparison of the number of STF modules on EA3D.

STF	Params.	Acc. (%)
3	1.91M	91.5
5	**3.03M**	**92.9**
7	4.14M	92.6

Effect of the number of STF stacks. Our STF-GCN is constructed by stacking several STF modules to enhance its ability to capture complex movement patterns over time and to discern subtle differences in motion. In this evaluation, we investigate the effect of varying the number of STF modules on STF-GCN's performance. As demonstrated in Table 4.7, an architecture with five stacked STF modules achieves the best results on EA3D dataset. Increasing the number of STF modules beyond this point leads to a degradation in accuracy, likely due to an overfitting issue.

4.3.4.2 Comparison with Other SOTA Approaches in Elderly Action Recognition

Building upon our STF-GCN, we developed two additional models named -hand and -leg, each designed to target specific skeletal regions. The STF-GCN model analyzes all 25 skeletal joints. In contrast, the -hand and -leg models utilize only the 13 hand joints and 9 leg joints, respectively, as inputs. The -ens model represents an ensemble of these three models. The ensemble method combines the predictions from the -hand, -leg, and STF-GCN models to leverage their complementary strengths, thereby enhancing the overall accuracy and robustness of action recognition. Notably, both the -hand and -leg models employ three STF modules and exclude the use of reinforced and united encoding. To ensure a fair comparison, we adopt methods specifically designed for elderly action recognition in Table 4.8, while Table 4.9 presents methods tailored for general action recognition. This distinction is necessary due to the unique challenges associated with each task, ensuring that the evaluation remains both meaningful and equitable.

The experimental results, presented in Table 4.8, demonstrate that both the

Table 4.8: Performance comparison on EA3D.

Category	Methods	EA3D (%)
RNN	IndRNN [132]	79.3
	Beyond Joint [133]	79.1
CNN	HCN [96]	88.0
	DBL [134]	88.4
	FSA-CNN [2]	90.6
GCN	ST-GCN [34]	86.8
	Motif ST-GCN [135]	89.9
	ESE-FN [105]	88.6
Ours	STF-GCN	92.9
	STF-GCN-hand	91.5
	STF-GCN-leg	67.7
	STF-GCN-ens	**93.8**

STF-GCN and -hand models outperform other skeleton-based methods in elderly action recognition, with the -ens model achieving state-of-the-art (SOTA) performance. Specifically, our STF-GCN and -hand models surpass the current SOTA, FSA-CNN, by 2.3% and 0.9% respectively, while our -ens model exceeds it by 3.2%. These achievements are attributed to our innovative TDE and STF modules. The STF-GCN-ens model significantly outperforms both RNN- and CNN-based approaches and also surpasses other GCN-based methods, thereby establishing a new benchmark in the field.

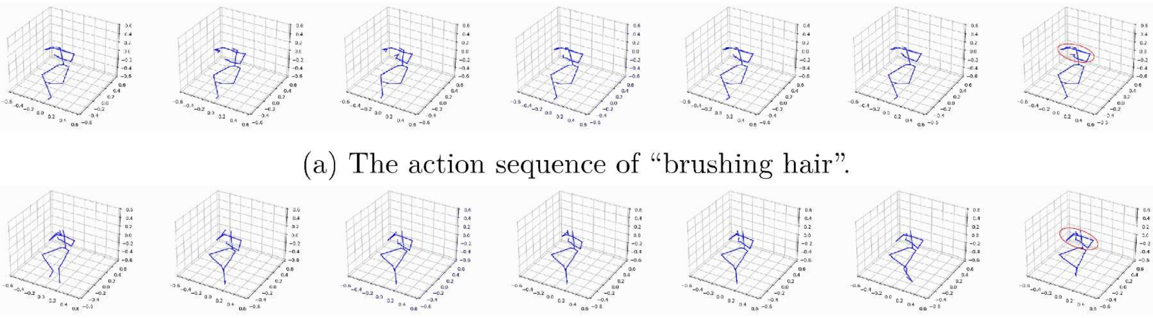
4.3.4.3 Comparison with Other SOTA Approaches in General Action Recognition

To evaluate the generalization of the proposed STF-GCN, we also conducted comparative experiments on NTU120 dataset, which are aimed at general action recognition tasks, comparing the proposed STF-GCN against other advanced methods. As shown in Table 4.9, the STF-GCN-ens model achieves SOTA performance on the X-sub120. Although it does not achieve the optimal performance on the X-set120, it trails the best result by only 0.2%.

Our method has achieved the best performance in elderly action recognition, benefiting from our focus on the subtle and prolonged movements characteristic of this demographic. However, there is room for improvement in general action recognition, which involves a broader range of dynamic movements. This discrepancy indicates that our method's performance is dependent on specific dataset characteristics.

Table 4.9: Performance comparison on NTU120.

Methods	X-sub120 (%)	X-set120 (%)
2s-GCA [136]	73.0	73.3
LSTM-IRN [137]	77.7	79.6
ST-GCN [34]	70.7	73.2
AS-GCN [138]	77.9	78.5
2S-AGCN [35]	82.9	84.9
FTF-GCN [104]	83.2	83.7
EfficientGCN-B4 [25]	88.3	89.1
STF [139]	88.9	89.9
HD-GCN(2-ens) [22]	89.1	**90.6**
IGFormer [140]	85.4	86.5
FG-STForm [141]	89.0	**90.6**
STF-GCN	87.2	89.0
STF-GCN-hand	85.6	87.5
STF-GCN-leg	49.8	50.5
STF-GCN-ens	**89.5**	90.4



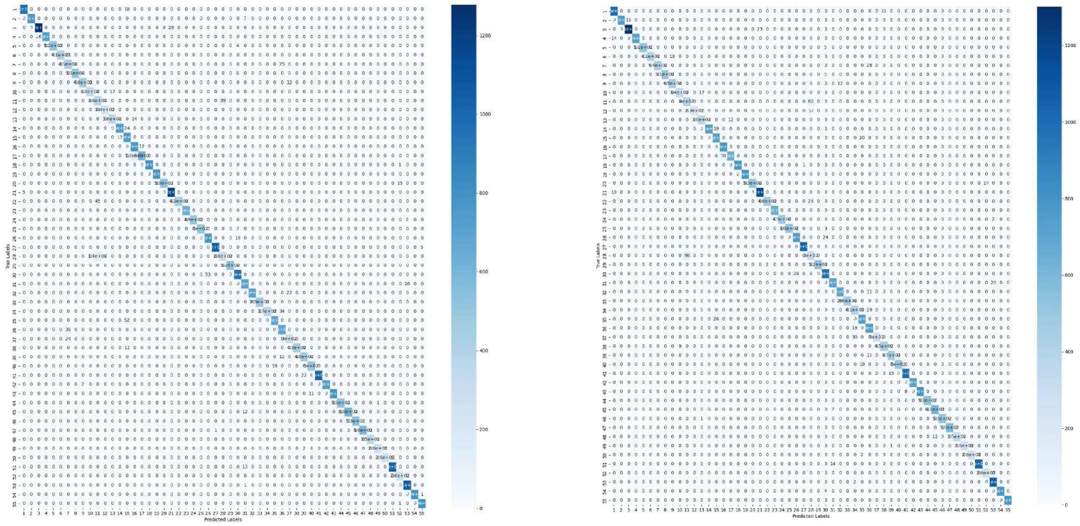
(a) The action sequence of “brushing hair”.

(b) The action sequence of “blow drying hair”.

Figure 4.7: Visual examples of similar elderly actions, “brushing hair” vs “blow drying hair”. The actions “brushing hair” and “blow drying hair” share common characteristics, as both involve interactions between the hands and the head.

Table 4.10: Performance comparison of elderly actions with similar patterns on EA3D dataset. The “Action” column represents ground truth labels, while the “Similar action” column indicates actions with similar motion trajectories.

Action	STF-GCN w/o TDE (%)	STF-GCN (%)	Similar Action
brushing hair	97.85	98.61 (↑0.76)	blow drying hair
blow drying hair	80.84	88.86 (↑8.02)	brushing hair
wiping face with a towel	85.85	89.62 (↑3.77)	blow drying hair
washing a towel by hands	53.32	57.69 (↑4.37)	washing hands
washing hands	72.35	75.19 (↑2.84)	washing a towel by hands
washing the dishes	80.49	88.07 (↑7.58)	washing a towel by hands
brushing teeth	85.79	89.47 (↑3.68)	smoking
smoking	88.01	92.07 (↑4.06)	brushing teeth
putting on a jacket	99.87	99.87 (↑0)	taking off a jacket
taking off a jacket	98.86	99.62 (↑0.76)	putting on a jacket
talking on the phone	78.30	84.55 (↑6.25)	rubbing face with hands
rubbing face with hands	89.67	90.04 (↑0.37)	talking on the phone
drinking water	90.01	93.30 (↑3.29)	taking medicine
taking medicine	91.42	93.06 (↑1.64)	drinking water



(a) Confusion matrix of STF-GCN without TDE.

(b) Confusion matrix of STF-GCN.

Figure 4.8: The confusion matriexs obtained by STF-GCN without TDE and STF-GCN on the EA3D dataset. The horizontal axis represents the predicted labels, while the vertical axis represents the true labels.

4.3.5 Results Analysis for Similar Actions

This part highlights the capability of our model to discern actions that are typically considered closely similar patterns. For example, Figure 4.7 illustrates one group of the typical actions, i.e., “brushing hair” vs “blow drying hair”, in which the skeletal motion trajectories and amplitude are remarkably similar. This similarity challenges the model without TDE, yet, by incorporating the TDE, our model significantly improves discrimination between these actions. The results presented in Table 4.10 and the confusion matrix depicted in Figure 4.8 further validate the superior performance of our model in achieving more accurate recognition of elderly actions.

The ability to distinguish between highly similar actions is crucial in real-world applications, especially when only skeleton-based data is available. Accurately recognizing subtle differences not only reduces misclassification but also improves the system’s reliability in critical scenarios such as elderly care, where distinguishing between intentional and unintentional behaviors is essential for timely intervention and appropriate response.

4.4 Conclusion

In this chapter, we proposed a Skeleton Temporal Fusion Graph Convolutional Network (STF-GCN) to address two key challenges in skeleton-based action recognition, which are small amplitude and long duration of actions typical among elderly individuals. The framework of the STF-GCN can be summarized through two main insights. First, we proposed a temporal dimension encoding that encompasses two types of temporal features and three encoding strategies. This approach enabled comprehensive extraction and analysis of subtle motion patterns from the temporal dimension of skeletal sequences. Second, we proposed a Skeleton Temporal Fusion (STF) module to highlight temporal feature representations. This module employed an alternating structure of large and small kernel convolutions to enhance the perception of subtle movements and effectively capture temporal patterns across various scales.

To validate the effectiveness of the proposed method, we conducted extensive experiments on both elderly action recognition and general action recognition benchmarks. Experimental results demonstrated that STF-GCN achieved 93.8% accuracy on the ETRI-Activity3D dataset for elderly action recognition, surpassing other competitive methods. Furthermore, on the NTU-RGB+D 120 dataset for general action recognition, our model achieved 89.5% accuracy on the cross-subject (X-sub120) benchmark and 90.4% accuracy on the cross-setup (X-set120) benchmark. These results highlight the strong generalization capability of STF-GCN across different action recognition scenarios, proving its robustness and effectiveness in both elderly and general action recognition tasks. Additionally, we analyzed the model's ability to distinguish between highly similar actions, which is particularly important in elderly action recognition due to the subtle nature of movements.

Chapter 5

Depth Camera-based Human Action Recognition System

This chapter builds upon the object detection algorithm from Chapter 3 and the action classification algorithm from Chapter 4 to develop a depth camera-based action recognition system. Furthermore, it explores the innovative applications of this system in two key domains: elderly care and smart education. In the field of elderly care, the system focuses on detecting and analyzing falls, which is crucial for enhancing safety measures and supporting independent living among elderly individuals. In the domain of smart education, our depth camera-based model is specifically designed for real-time recognition of students' hand-raising actions, providing valuable insights into classroom engagement and enabling educators to adjust their teaching strategies accordingly.

5.1 Introduction

Human action recognition is a challenging and engaging research task within computer vision. It has a wide range of applications including video understanding, smart surveillance, robotics, industrial automation, healthcare, and education [2, 3, 142]. In recent years, many researchers have dedicated efforts to recognizing and analyzing human actions from RGB videos [18, 19]. However, methods based on RGB often fail to achieve satisfactory results in practical applications due to their inability to robustly handle environmental noise such as changes in viewpoint, lighting conditions, background colors, and clothing [20, 21].

The rapid advancement of depth camera technology has opened up new possibilities for action recognition. Depth cameras, such as Azure Kinect DK [85], utilize Time-of-Flight (ToF) technology to capture depth information, enhancing the understanding of complex environments and interactions [22]. Unlike RGB-based, methods that leverage three-dimensional coordinates of human joints extracted from depth cameras are inherently robust to variations in lighting, viewpoint, background, and clothing. This robustness primarily stems from the sensing modality rather than the model architecture itself [23, 24]. Despite these advantages, technologies based on depth cameras, object detection, and GCNs have not yet been fully developed or widely applied in specific practical domains, such as elderly care and smart education.

In elderly care, medical surveys have shown that falls are a leading cause of both fatal and non-fatal injuries among the elderly [143]. The incidence of falls among the elderly ranges from 32% to 42%, and timely medical intervention after a fall can reduce the risk of death by 80% [144, 145]. Therefore, accurately and effectively monitoring elderly falls is of great importance. However, due to the sudden nature of falls and the rapidity of the falling process, traditional monitoring methods often fail to capture the entire event in real-time. This necessitates the adoption of more advanced technological means, such as depth cameras, which can precisely capture the entire process of a fall and provide detailed three-dimensional spatial information, thereby laying a solid foundation for the accurate identification of falls. Furthermore, by utilizing GCNs to analyze these complex three-dimensional data, we can extract key features from structured spatial relationships and the continuous temporal dimension, achieving more accurate recognition and analysis.

In smart education, recognizing students' hand-raising behavior plays a crucial role as it helps analyze classroom engagement to assess teaching processes and optimize educational strategies [146]. Traditionally, many studies have used two-dimensional cameras, i.e., RGB videos, to record and analyze student behavior to evaluate teaching quality and student attitudes [147]. However, this method inherently lacks depth information of the targeted objects and cannot robustly handle environmental noise. Depth camera technology, which provides richer three-dimensional spatial information, has the potential to significantly enhance the accuracy and reliability of student action recognition, yet its application in smart education is still in an exploratory stage. Depth cameras, combined with the analytical power of GCNs, offer a promising avenue for capturing and understanding complex student behaviors through enhanced three-dimensional data analysis.

Depth Camera
Depth Camera

(a) Elderly care.

(b) Smart education.

Figure 5.1: Applications of depth cameras in real-world scenarios. (a) Application of depth cameras in elderly care, focusing on the recognition and analysis of falling actions. (b) Application of depth cameras in smart education, concentrating on detecting students' hand-raising actions.

This chapter presents an action recognition system built using depth cameras, object detection, and GCNs, with a focus on two key application scenarios: elderly care and smart education, as illustrated in Figure 5.1. We introduce a methodology that integrates these technologies to develop a depth camera-based action recognition system designed for real-time motion recognition. Our approach involves deploying depth cameras in real-world environments to capture video and skeleton data, which are then processed using the object detection algorithm from Chapter 3 and the action classification algorithm from Chapter 4 for human detection and action classification. To facilitate real-time applications, we convert the trained model into the ONNX format, significantly enhancing its portability and deployment efficiency on depth cameras. Extensive experiments were conducted on fall detection in elderly care and hand-raising detection in educational settings. In elderly care, the system can enable timely emergency alerts by detecting falls or abnormal behaviors, potentially reducing response time and improving patient safety. In educational settings, automated recognition of student gestures such as hand-raising can support classroom engagement analytics, assistive teaching, and adaptive learning environments.

5.2 Proposed Depth Camera-based Human Action Recognition System

In this section, we present a human action recognition system based on depth camera. The system aims to accurately recognize human actions by integrating depth sensing technology, object detection (Chapter 3), and action classification (Chapter 4). While the proposed system is applied to both elderly care and smart education scenarios, the nature of the actions involved in these two domains differs significantly. In elderly care, the target actions, such as falling, are typically unintentional and demand rapid recognition due to their potential medical urgency. In contrast, in smart education, the system focuses on identifying intentional gestures such as hand-raising, which serve as indicators of cognitive participation and classroom interaction.

5.2.1 System Architecture

The proposed depth camera-based human action recognition system integrates RGB and depth sensors to effectively capture human motion and classify actions in real

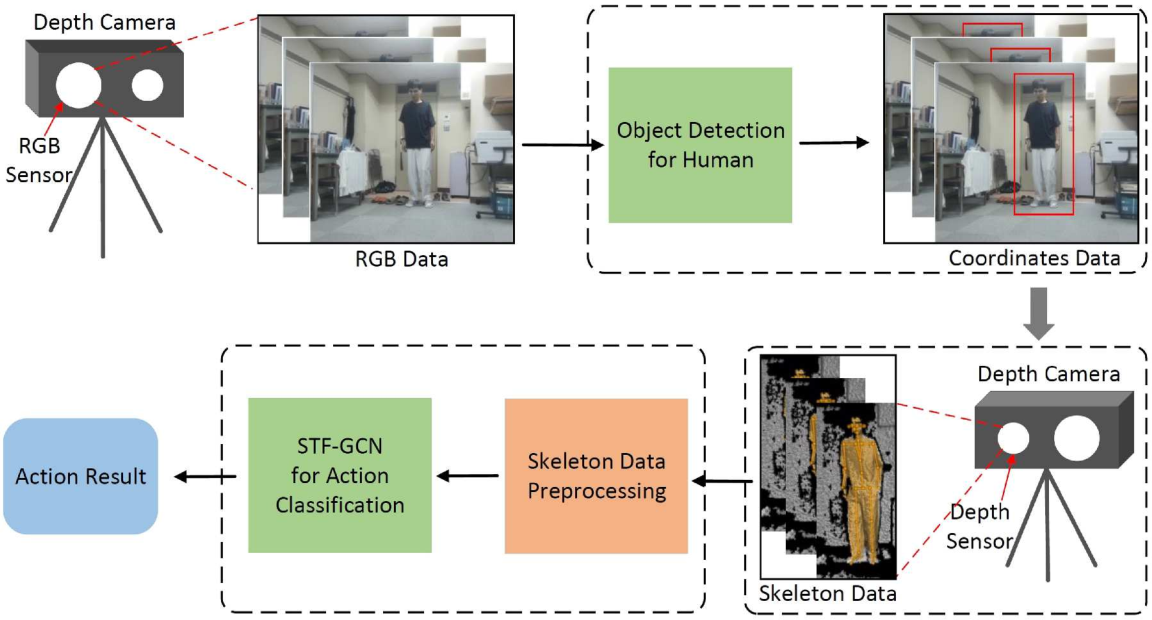


Figure 5.2: Architecture of the depth camera-based human action recognition system. The system leverages an depth camera to capture both RGB and depth data. The RGB data undergoes object detection to identify human subjects and extract coordinate data. Meanwhile, the depth sensor retrieves skeleton data, which is preprocessed before being passed to the STF-GCN model for action classification. The final output is the recognized action result.

time. The system follows a well-defined pipeline comprising three primary stages: human detection, skeleton extraction, and action classification, as illustrated in Figure 5.2.

The human detection stage aims to detect human subjects in an environment using RGB data captured by the RGB sensor of the depth camera. The system applies our proposed cascade network (see Chapter 3 for details), which leverages multi-stage refinement to enhance detection accuracy. The detected bounding box coordinates are then extracted, providing crucial positional data for subsequent processing. This step ensures that only relevant human motion is analyzed in later stages.

In the skeleton extraction stage, a depth sensor embedded in the depth camera is used to capture skeleton data corresponding to the detected human figures. The extracted depth-based skeletal representation transforms raw depth images into structured skeletal data, allowing precise tracking of human joint movements. This stage enables a more detailed and spatially accurate analysis of human actions, bridging the gap between visual input and structured motion modeling.

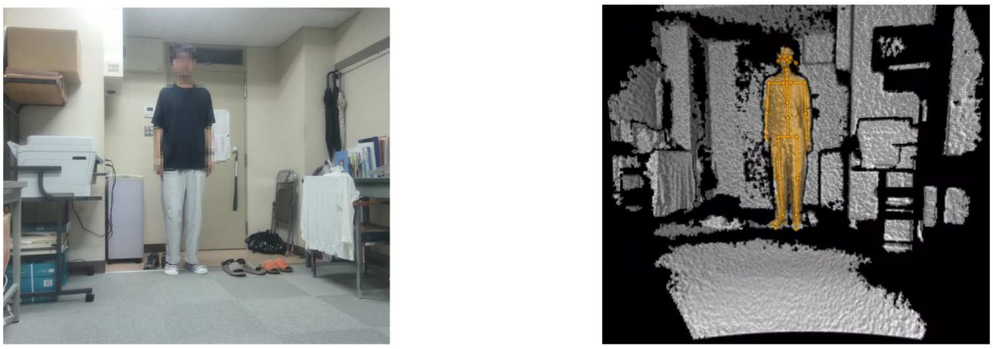
The final action classification stage processes the extracted skeletal data for action recognition. First, the skeleton data preprocessing module refines and normalizes the skeletal representations to ensure consistency. The preprocessed data is then fed into our Skeleton Temporal Fusion Graph Convolutional Network (STF-GCN) (see Chapter 4 for details). STF-GCN is designed to capture both spatial and temporal dependencies in skeletal motion, enabling robust recognition of various actions. It incorporates a temporal encoding mechanism and a spatial-temporal fusion module, enhancing the system's ability to capture fine-grained motion patterns and recognize both short-term and long-duration actions with high accuracy. The final action recognition results serve as the system's output.

5.2.2 Skeleton Data Preprocessing

This system utilizes the Femto Bolt depth camera. Figure 5.3 demonstrates visual examples of the depth camera's RGB field of view (Figure 5.3a) and infrared intensity visualization (Figure 5.3b) from a distance of 1.5 meters from the front. It is evident that the RGB field of view does not contain spatial information about the scene, such as the distance of objects. In contrast, Figure 5.3b represents the intensity of infrared reflection captured by the depth camera, where the grayscale variations indicate the strength of the reflected infrared signal. Depth information is stored as pixel-wise attributes in the depth data. And then, using the API provided by Azure Kinect

Table 5.1: Joints mapping from Kinect DK output to GCN input.

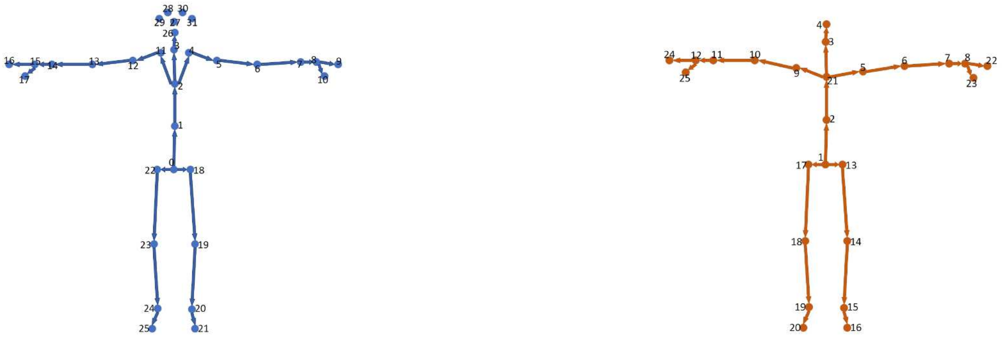
Joint Name	Kinect DK Joint Number	GCN Input Joint Number
Pelvis	0	1
Spine Nanal	1	2
Neck	3	3
Head	26	4
Shoulder Left	5	5
Elbow Left	6	6
Wrist Left	7	7
Hand Left	8	8
Shoulder Right	12	9
Elbow Right	13	10
Wrist Right	14	11
Hand Right	15	12
Hip Left	18	13
Knee Left	19	14
Ankle Left	20	15
Foot Left	21	16
Hip Right	22	17
Knee Right	23	18
Ankle Right	24	19
Foot Right	25	20
Spine Chest	2	21
Handtip Left	9	22
Thumb Left	10	23
Handtip Right	16	24
Thumb Right	17	25



(a) RGB field of view.

(b) Infrared intensity visualization.

Figure 5.3: Visualization of depth camera views.



(a) Azure Kinect DK format.

(b) Input format for GCNs.

Figure 5.4: Comparison of depth camera-derived human skeletal outputs and GCNs input formats.

Algorithm 3 Convert Kinect skeletons to GCN skeletons

```
Require: Input: $Kinect\_Skeletons$   // A one-dimensional vector composed of 32
    joints captured by the depth camera.
Ensure: Output: $GCN\_Skeletons$   // A one-dimensional vector composed of 25
    GCN joints.
$kinectToGCNMap \leftarrow [0, 1, 3, 26, 5, 6, 7, 8, 12, 13, 14, 15, 18,$
                         $19, 20, 21, 22, 23, 24, 25, 2, 9, 10, 16, 17]$
Initialize $GCN\_Skeletons$ as an empty vector of floats.
Initialize $gcnJointsTemp$ as an array of floats of size $JointCount \times 3$.   ▷ Each
    joint is represented by (x, y, z)
for each person in $Person\_Count$ do
    for each $skeletonPair$ in $Kinect\_Skeletons$ do
        for $gcnIndex$ from 0 to $Joint\_Count - 1$ do
            $kinectIndex \leftarrow kinectToGCNMap[gcnIndex]$
            $gcnJointsTemp[gcnIndex \times 3 + 0] \leftarrow$
                $skeletonPair.person.joints[kinectIndex].position.xyz.x$
            $gcnJointsTemp[gcnIndex \times 3 + 1] \leftarrow$
                $skeletonPair.person.joints[kinectIndex].position.xyz.y$
            $gcnJointsTemp[gcnIndex \times 3 + 2] \leftarrow$
                $skeletonPair.person.joints[kinectIndex].position.xyz.z$
        end for
        Append $gcnJointsTemp$ to $GCN\_Skeletons$
    end for
end for
```

Body Tracking SDK [148], we can capture 32 human body joints (marked in yellow in Figure 5.3b). Figure 5.4a displays the specific meanings of these 32 joints.

According to preprocessing protocols established by prior research [22, 105, 24], which have been proven effective and widely adopted, the input joint count for GCNs

Algorithm 4 Transform a one-dimensional vector to a five-dimensional tensor

```
Require: Input: $GCN\_Skeletons$
Ensure: Output: $GCN\_Tensor$    // A five-dimensional tensor
Initialize $rearranged\_data$ as a vector of floats with size
        $Batchsize \times Channels \times Frame\_Count \times Joint\_Count \times Person\_Count$
Initialize $index \leftarrow 0$
for $channel$ from 0 to $Channels - 1$ do
    for $frame$ from 0 to $Frame\_Count - 1$ do
        for $joint$ from 0 to $Joint\_Count - 1$ do
            for $person$ from 0 to $Person\_Count - 1$ do
                Compute $original\_index$ as:
        $original\_index \leftarrow frame \times Joint\_Count \times Person\_Count \times Channels +$
            $joint \times Channels + person \times Channels \times Joint\_Count + channel$
                Set $rearranged\_data[index] \leftarrow GCN\_Skeletons[original\_index]$
                Increment $index$
            end for
        end for
    end for
end for
Convert $rearranged\_data$ to $GCN\_Tensor$ using torch::from_blob function
```

is set at 25, as shown in Figure 5.4b. Since the number and sequence of the 32 joints captured by the depth camera do not align with the input format required by the GCNs, it is necessary to map these captured joints to the 25 joints required by the GCN. This mapping is detailed in Table 5.1, which provides a comprehensive cross-reference of each joint from the depth camera output to its corresponding joint in the GCN input.

Based on the mapping relationships outlined in Table 5.1, we first use Algorithm 3 to convert the one-dimensional vector of 32 joints captured by the depth camera into a one-dimensional vector containing 25 GCN joints. The input format for the GCN is a five-dimensional tensor, i.e. [Batchsize, Channels, Frame_Count, Joint_Count, Person_Count]. Subsequently, we employ Algorithm 4 to transform the one-dimensional vector of 25 joints into a five-dimensional tensor suitable for GCN input.

After successfully constructing the five-dimensional tensor, we can then use our proposed STF-GCN to perform model training and inference on the action data captured by the depth camera.

5.3 Experiments

Our experiments focus on fall detection in elderly care and hand-raising detection in smart education. Initially, we train our proposed STF-GCN using publicly available datasets [2, 129] and use it as a pre-trained model. Subsequently, we utilize transfer learning to fine-tune our model on fall and hand-raising data collected by the depth camera. Finally, the trained model is converted into an ONNX format, which can be deployed on depth cameras for testing and evaluation.

5.3.1 Datasets

ETRIActivity3D (EA3D) [2] is currently the largest elderly action recognition dataset collected in real-world monitoring environments, comprising 112,620 samples of 50 elderly individuals and 50 young adults performing actions across 8 synchronized sensors. The elderly participants range in age from 64 to 88 years, while the young adults are around 20 years old. All videos are categorized into 55 types of actions.

NTU-RGB+D 60 (NTU 60) [129] is a large-scale laboratory indoor dataset provided by [15], comprising 60 action categories. The authors of the dataset recommend two benchmarks: (1) Cross-Subject (X-sub), which includes 40,320 training samples and 16,560 evaluation samples, dividing 40 subjects into two groups. (2) Cross-View (X-view) uses videos captured by cameras 2 and 3 as training samples (37,920 videos) and videos captured by camera 1 as evaluation samples (18,960 videos).

We collected fall and hand-raising data by recording videos, each lasting one minute. Following the data preprocessing protocols established in [24, 105], we set the Frame_Count in Algorithms 3 and 4 to 64, a setting that is suitable for both falling and hand-raising actions. We created 88 samples from one-minute videos using a stride of 20 frames. To increase the quantity and diversity of samples, we also produced 58 samples from one-minute videos using a stride of 30 frames. Thus, a single one-minute video could generate 146 samples. We collected a total of 6,000 samples using the depth camera, with 3,000 samples for each type of action to fine-tune the STF-GCN model. Of these, 2,500 samples were used as the training set and 500 samples as the test set.

5.3.2 Evaluation Metrics

We evaluate the performance of our system using several key metrics: FPR (False Positive Rate), FNR (False Negative Rate), Acc (Accuracy), PT (Preprocessing

Time) and IT (Inference Time). These metrics collectively assist in assessing the effectiveness, efficiency and operational speed of our system.

FPR is defined as the proportion of negative cases that were incorrectly classified as positive. It represents the probability of falsely identifying a negative instance as positive and is calculated using the Eq 5.1. FNR is the proportion of positive cases that were incorrectly classified as negative. It measures the probability of failing to identify a positive instance and is calculated as Eq 5.2. Acc measures the proportion of true results (both true positives and true negatives) among the total number of cases examined. It provides an overall effectiveness of the model and is expressed as Eq 5.3. PT and IT specifically measure the time our system takes during the data preprocessing and model inference phases, respectively, highlighting the operational efficiency.

$$FPR = \frac{FP}{FP + TN} \tag{5.1}$$

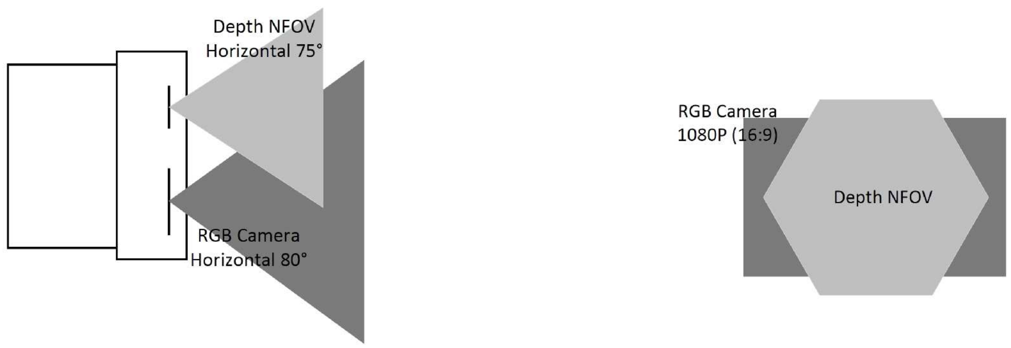
$$FNR = \frac{FN}{FN + TP} \tag{5.2}$$

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.3}$$

5.3.3 Implementation Details

For the depth camera settings, Figure 5.5 illustrates the specific setup details during the data collection process. Specifically, the RGB camera was set to 1080p mode with a frame rate of 30. In depth mode, we used the NFOV Unbinned (640×576) mode, also at a frame rate of 30. The distance between the camera and the subject ranged from 1.5 to 2.5 meters, which aligns with the optimal operating conditions of the depth camera. And the camera was positioned 70 cm above the ground.

For the training phase, programming was conducted in Python on Ubuntu 22.04. We use SGD optimization with 0.1 as the base learning rate and a weight decay of 0.0005. All the models are trained on two GeForce RTX 3090 with a batch size of 200, using ReduceLROnPlateau to update the learning rate. For fall detection, we initially pre-trained using the EA3D dataset, setting epochs to 100, and then fine-tuned the model with datasets collected by the depth camera. Similarly, for hand-raising recognition, we pre-trained using the NTU 60 dataset with epochs set at 100, followed by fine-tuning with datasets collected by the depth camera. The



(a) Field of view of depth and RGB.

(b) Field of view from 1.5m front.

Figure 5.5: Field of view for depth and RGB cameras (the perspective seen by the sensors) and field of view at a distance of 1.5 meters from the front.

fine-tuned model is in PyTorch format. Finally, we convert the PyTorch model into an ONNX format model that can be deployed on a depth camera.

For the inference phase, programming was conducted in C++ using Microsoft Visual Studio Community 2019. The computer used for this purpose was equipped with a 12th Gen Intel(R) Core (TM) i9-12900 and 32GB of RAM. The depth camera employed was the Femto Bolt, which is a collaborative production by Microsoft and Orbbec.

5.3.4 Results and Analysis of Fall Detection

Our proposed system first employs a cascade network-based object detection model to accurately locate individuals. Once detected, a depth camera extracts skeleton keypoints, which are then processed through a data preprocessing pipeline. Finally, the preprocessed skeleton data is classified using our action recognition model, STF-GCN.

Initially, we evaluated STF-GCN on the EA3D dataset, achieving an accuracy of 92.9% (Table 5.2), with classification results visualized in the confusion matrix (Figure 5.6). To further enhance performance specifically for fall detection, we fine-tuned the model using our collected fall data, boosting its accuracy to 97.6%. These results underscore not only the effectiveness of STF-GCN but also the importance of targeted fine-tuning for domain-specific applications.

After fine-tuning, the STF-GCN model size is 8.07MB. We then convert the fine-tuned model into the ONNX format, making it deployable on a depth camera, with the converted model size reduced to 1.55MB. Similarly, the cascade network-based

Table 5.2: Accuracy of our action recognition model on EA3D dataset before and after fine-tuning specifically for fall detection.

Setting	Acc.
Initial Testing on EA3D	92.9%
After Fine-tuning for Fall Detection	97.6%

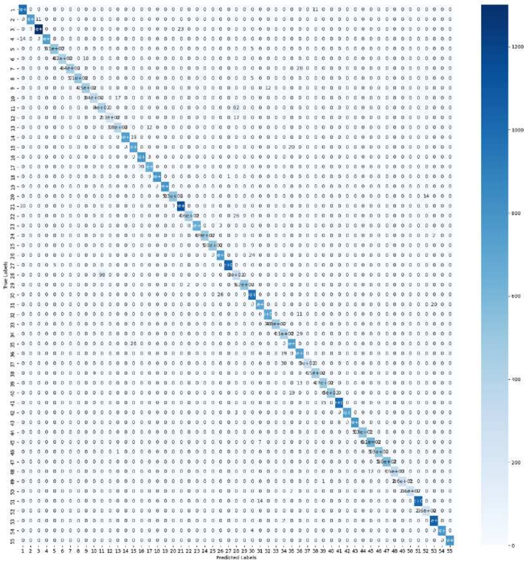


Figure 5.6: The confusion matrices of our action classification model on EA3D. The horizontal axis represents the predicted labels, while the vertical axis represents the true labels.

object detection model, originally 375MB, is also converted into the ONNX format, reducing its size to 72MB for deployment on the depth camera.

After deploying these ONNX models to the depth camera, we conducted real-time performance tests in real-world scenarios. Due to the limited detection range of the depth camera, these tests were performed with only one individual. Specifically, we simulated falls of an elderly person from four different orientations including front, back, left, and right. The results presented in Table 5.3 demonstrate our system's effectiveness and efficiency from various orientations. The system achieves 100% accuracy in both the Left and Right orientations, illustrating the robustness and reliability of our fall detection system in these views. However, the accuracy for Front and Back orientations does not reach 100%, primarily due to increased false positive rates in the Front and elevated false negative rate in the Back orientation. These discrepancies highlight challenges in accurately detecting falls when the individual is facing towards or away from the camera, which may be influenced by the depth

Table 5.3: Fall detection accuracy and processing times from various orientations.

Metric	Front	Back	Left	Right	Avg.
FPR	25.0%	0	0	0	6.25%
FNR	0	33.3%	0	0	8.33%
Acc.	92.9%	92.3%	100%	100%	96.3%
PT	0.148s	0.136s	0.139s	0.138s	0.14s
IT	0.047s	0.043s	0.047s	0.044s	0.045s

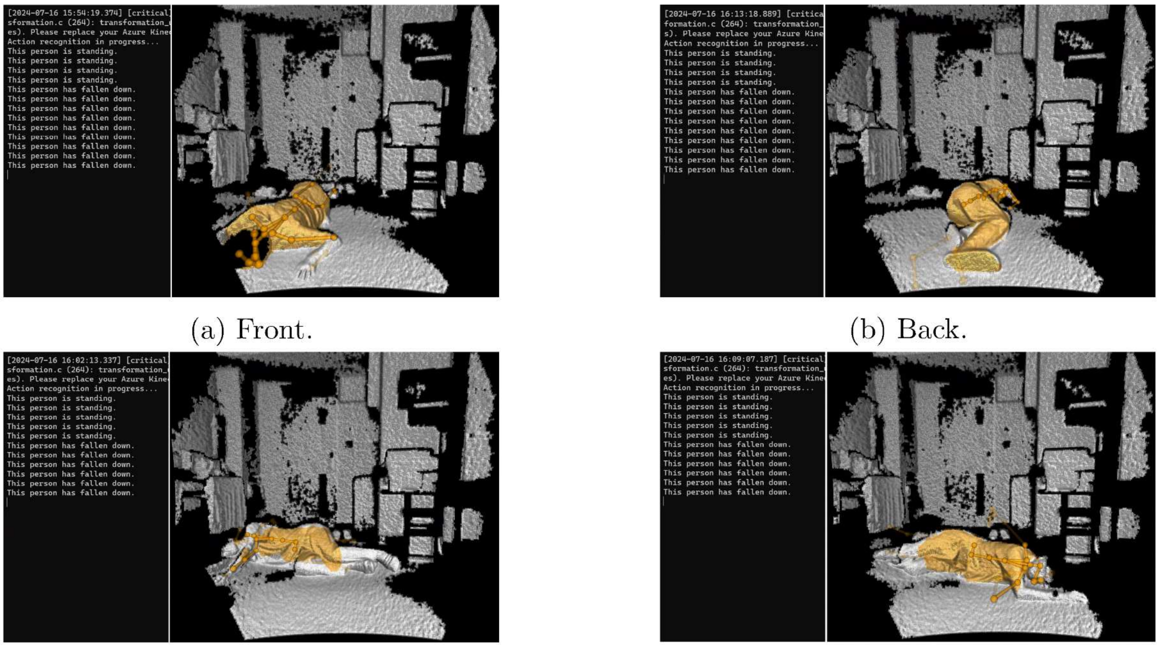
camera's angle and the overlap of body parts in these orientations. Table 5.3 also details preprocessing and inference times, further highlighting the operational efficiency of our system in real-world scenarios. These metrics collectively demonstrate the capability to our system in real-world scenarios.

Figure 5.7 provides visual examples of the detection process from these various orientations. These images demonstrate the system's real-time response and its ability to accurately recognize falls, which is crucial for ensuring the safety of the elderly. To distinguish falls from other actions such as sleeping, the system analyzes short-term temporal dynamics of joint movement. Falls are typically characterized by a sudden and rapid change in body posture, whereas sleeping involves gradual and controlled transitions. Our model captures this difference by leveraging temporal patterns in the skeleton sequence, enabling reliable classification of fall events.

Building upon these results, we plan to deploy the proposed fall detection system on domestic service robots for continuous and real-time elderly monitoring in home environments. With the integration of depth cameras and onboard edge computing, the system can detect critical incidents such as falls and automatically trigger emergency alerts or assistive actions.

5.3.5 Results and Analysis of Hand-raising Detection

Similar to the fall detection task, we first evaluated the STF-GCN model for hand-raising detection on NTU 60 X-sub and NTU 60 X-view, achieving accuracies of 91.4% and 95.9%, respectively, as shown in Table 5.4. The classification results on the NTU 60 datasets are further illustrated in the confusion matrices in Figure 5.8. To further optimize the model specifically for hand-raising actions, we fine-tuned it by retraining with our collected hand-raising dataset, which improved its accuracies to 95.1% and 98.5%. These results highlight the model's enhanced performance and demonstrate the effectiveness of targeted fine-tuning for specialized tasks.



```
[2024-07-16 15:54:19.374] [critical]
sformation.c (264): transformation_
es). Please replace your Azure Kine
Action recognition in progress...
This person is standing.
This person is standing.
This person is standing.
This person is standing.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
```

(a) Front.

```
[2024-07-16 16:13:18.889] [critica
formation.c (264): transformation_
s). Please replace your Azure Kine
Action recognition in progress...
This person is standing.
This person is standing.
This person is standing.
This person is standing.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
```

(b) Back.

```
[2024-07-16 16:02:13.337] [critical
sformation.c (264): transformation_
es). Please replace your Azure Kine
Action recognition in progress...
This person is standing.
This person is standing.
This person is standing.
This person is standing.
This person is standing.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
```

(c) Left.

```
[2024-07-16 16:09:07.187] [critical
sformation.c (264): transformation_
es). Please replace your Azure Kine
Action recognition in progress...
This person is standing.
This person is standing.
This person is standing.
This person is standing.
This person is standing.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
This person has fallen down.
```

(d) Right.

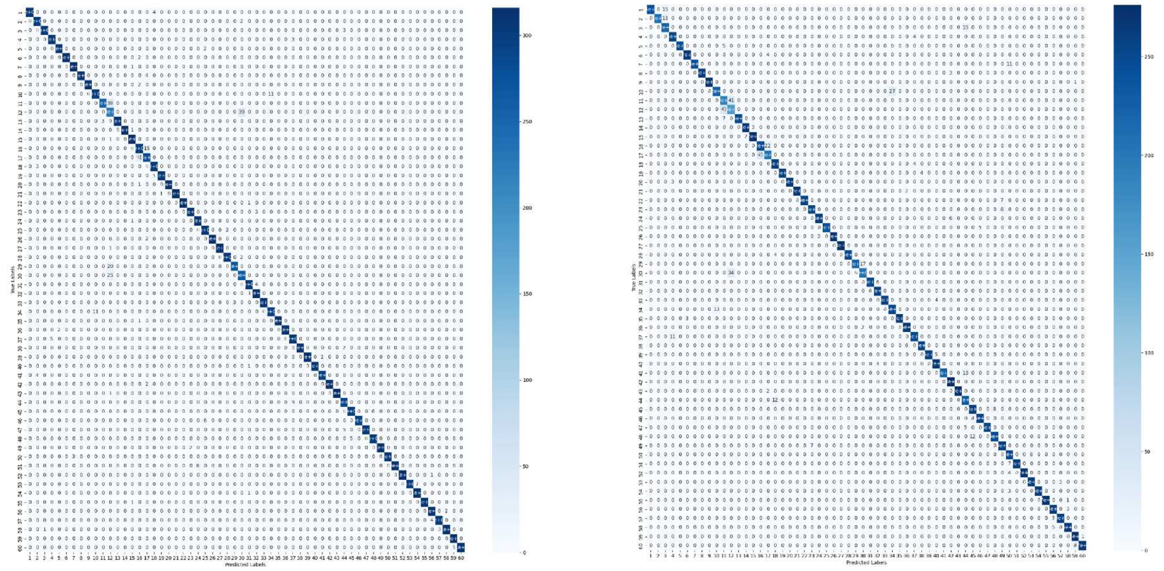
Figure 5.7: Visual examples of fall detection conducted from four orientations. Each figure (a) Front, (b) Back, (c) Left and (d) Right displays the detection outcome: "This person is standing" when the individual is upright, and "This person has fallen down" following a fall.

Since the two fine-tuned models listed in Table 5.4, we selected the model with the higher accuracy to be converted into an ONNX model for deployment on a depth camera. The sizes of the fine-tuned model and the ONNX model are 8.07MB and 1.55MB, respectively. Similarly, the cascade network-based object detection model , originally 375MB, is also converted into the ONNX format, reducing its size to 72MB for deployment on the depth camera.

After deploying these ONNX models to the depth camera, we conducted real-time performance tests in real-world scenarios. Due to the limited detection range of the depth camera, tests were performed only in scenarios involving one person and two people. Specifically, we simulated the detection of student hand-raising actions in a classroom setting and recorded the number of hand-raising in real-time. Table 5.5 presents the FPR, FNR, Acc, PT and IT for scenarios involving one and two people. For one person, the system achieved a higher accuracy of 93.1% with lower false positive and false negative rates compared to the two-people scenario, where accuracy dropped to 86.2%. This indicates the system's efficiency in simpler scenarios, while highlighting challenges in more complex settings with multiple individuals.

Table 5.4: Accuracy of the action recognition model on NTU 60 dataset before and after fine-tuning specifically for hand-raising.

Setting	Acc.
Initial Testing on NTU 60 X-sub	91.4%
After Fine-tuning for Hand-raising	95.1%
Initial Testing on NTU 60 X-view	95.9%
After Fine-tuning for Hand-raising	98.5%



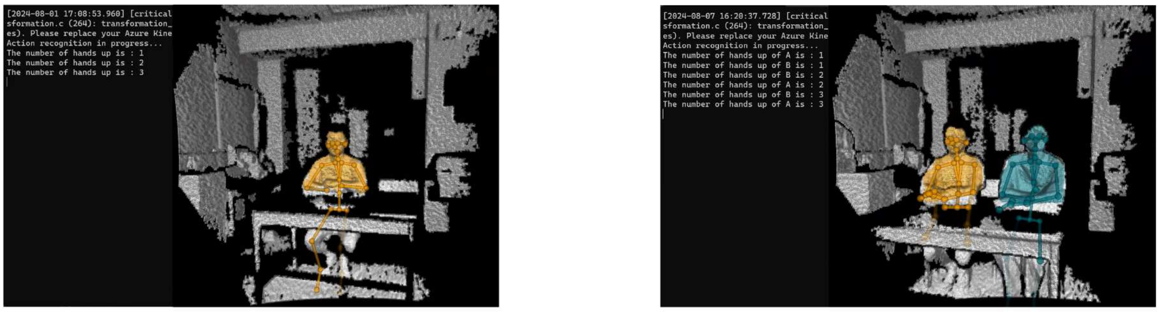
(a) Confusion matrix on NTU60 X-view.

(b) Confusion matrix on NTU60 X-sub.

Figure 5.8: The confusion matrices of our action recognition model on NTU60 X-view and NTU60 X-sub. The horizontal axis represents the predicted labels, while the vertical axis represents the true labels.

Table 5.5: Hand-raising detection accuracy and processing times for one and two scenarios.

Metric	One person	Two people	Avg.
FPR	5.9%	13.3%	9.6%
FNR	8.3%	14.3%	11.3%
Acc.	93.1%	86.2%	89.7%
PT	0.142s	0.137s	0.14s
IT	0.02s	0.039s	0.03s



```
[2024-08-01 17:08:53.960] [critical
sformation.c (264): transformation_
es). Please replace your Azure Kine
Action recognition in progress...
The number of hands up is : 1
The number of hands up is : 2
The number of hands up is : 3
```

(a) One person.

```
[2024-08-07 16:20:37.728] [critical
sformation.c (264): transformation_
es). Please replace your Azure Kine
Action recognition in progress...
The number of hands up of A is : 1
The number of hands up of B is : 1
The number of hands up of B is : 2
The number of hands up of A is : 2
The number of hands up of B is : 3
The number of hands up of A is : 3
```

(b) Two people.

Figure 5.9: Visual examples of hand-raising detection for one person (a) and two people (b). The left side of each sub-plot displays real-time updates by our system that records the number of hand-raising actions detected.

Figure 5.9 provides visual examples of hand-raising detection tests conducted using a depth camera for scenarios involving (a) one person and (b) two people. On the left side of each sub-plot, the system's real-time updates are displayed, which include the number of hand-raising actions detected. These images illustrate the system's capability to accurately monitor and record hand-raising actions, showcasing its real-time effectiveness in different individuals settings.

5.4 Conclusion

In this chapter, we demonstrated the effectiveness and efficiency of our proposed system in two key applications: fall detection in elderly care and hand-raising detection in smart education. In both cases, our system exhibited high accuracy, low latency, and strong generalization capability in real-world scenarios. (a) Fall detection: Our system initially achieved 92.9% accuracy during the pre-training phase. After fine-tuning for specific fall scenarios, the accuracy improved to 97.6%. In real-world tests, the system maintained an average accuracy of 96.3%, with a false positive rate of only 6.25% and a false negative rate of 8.33%. Moreover, it demonstrated a rapid response time, with an average inference time of just 0.045 seconds. (b) Hand-raising detection: Through targeted fine-tuning, the system's accuracy improved from 95.9% to 98.5%. In real-world tests, the system maintained an average accuracy of 89.7%, with a false positive rate of 9.6% and a false negative rate of 11.3%. The average inference time was only 0.03 seconds, ensuring real-time performance. These results highlight the potential of our depth-based action recognition system for real-world applications in elderly care and smart education, demonstrating high accuracy, efficiency, and robustness in practical deployment.

Chapter 6

Conclusions and Future Works

In the previous chapters, this thesis has introduced a cascade-based object detection method (Chapter 3), a Skeleton Temporal Fusion Graph Convolutional Network (STF-GCN) for action classification (Chapter 4), and a depth camera-based human action recognition system (Chapter 5). This chapter summarizes the key contributions of this research and discusses potential directions for future work.

6.1 Conclusions

In this thesis, we developed an action recognition system that integrates cascade-based object detection, GCN-based action classification, and depth camera-based skeletal extraction. By deploying both the object detection model and action classification model on a depth camera, we constructed a real-time action recognition system. Our proposed system has been extensively evaluated in various application scenarios, including fall detection in elderly care and hand-raising detection in smart education, demonstrating its effectiveness and robustness.

For object detection, we addressed three major challenges: the imbalance distribution of categories, the diversity of object scales, and the overlap between objects. To overcome these challenges, we introduced Poisson blending combined with Canny edge processing to enhance data diversity, proposed Re-BiFPN for multi-scale feature fusion, and designed Rep-CIoU loss to effectively handle overlapping objects. Our proposed approach achieved an mAP of 83.4% for prohibited item detection, surpassing other state-of-the-art methods. Additionally, we validated our method on the COCO dataset for human detection, where it achieved 49.5% mAP on the test set, also outperforming other leading approaches. Qualitative analysis further confirms that our model effectively distinguishes individuals in crowded scenes, accurately localizes small and distant persons, and enhances detection precision. These advantages underscore the superiority of our method in real-world human detection applications and establish a robust foundation for the human detection step in human action recognition systems.

For action classification, we proposed the Skeleton Temporal Fusion Graph Convolutional Network (STF-GCN) to address two key challenges in skeleton-based action recognition: small-amplitude movements and long-duration actions. First, we introduced a temporal dimension encoding that incorporates two types of temporal features and three encoding strategies. This approach enables comprehensive extraction and analysis of subtle motion patterns from the temporal dimension of skeletal sequences. Second, we proposed the Skeleton Temporal Fusion (STF) module to enhance spatiotemporal feature representations. This module employs an alternating structure of large and small kernel convolutions to improve the perception of fine-grained movements and effectively capture temporal patterns at multiple scales. To validate the effectiveness of the proposed method, we conducted extensive experiments on both elderly action recognition and general action recognition benchmarks. The experimental results demonstrate that STF-GCN achieved 93.8% accuracy on the ETRI-Activity3D (EA3D) dataset for elderly action recognition, outperforming other competing methods. Additionally, on the NTU-RGB+D 120 dataset for general action recognition, our model achieved 89.5% accuracy on the cross-subject (X-sub120) benchmark and 90.4% accuracy on the cross-setup (X-set120) benchmark. These results highlight the strong generalization capability of STF-GCN across different action recognition scenarios, proving its robustness and effectiveness in both elderly and general action recognition tasks.

For skeleton data acquisition, we utilized a depth camera to stably and in real-time capture human skeleton data. To ensure efficient deployment, we converted both the cascade-based object detection model and the STF-GCN action classification model into a lightweight ONNX format and deployed them on the depth camera.

Our depth camera-based action recognition system operates in three steps: (1) Real-time human detection: The cascade-based object detection model detects human positions in real-time. (2) Skeleton data acquisition: The depth camera's depth sensor extracts skeleton data. (3) Action classification: The preprocessed skeleton data is fed into the STF-GCN model to classify actions, enabling real-time action recognition.

We demonstrated the effectiveness and efficiency of our proposed system in two key applications: fall detection in elderly care and hand-raising detection in smart education. In both cases, the system exhibited high accuracy, low latency, and strong generalization ability in real-world scenarios. (a) Fall detection: In real-world tests, the system achieved an average accuracy of 96.3%, with a false positive rate of only 6.25% and a false negative rate of 8.33%. Additionally, it demonstrated rapid response capabilities, with an average inference time of just 0.045 seconds. (b) Hand-raising detection: The system maintained an average accuracy of 89.7%, with a false positive rate of 9.6% and a false negative rate of 11.3% in real-world tests. The average inference time was only 0.03 seconds, ensuring real-time performance. These results highlight the potential of our depth camera-based action recognition system in practical applications such as elderly care and smart education, demonstrating high accuracy, efficiency, and robustness in real-world deployment.

These results highlight the potential of our depth camera-based action recognition system in practical applications such as elderly care and smart education, demonstrating high accuracy, efficiency, and robustness in real-world deployment. For fall detection, the proposed system can be deployed on domestic service robots to enable continuous and real-time monitoring of elderly individuals in home environments. For smart education, the system can be installed on classroom cameras or integrated into educational robots to support automatic recognition of student behaviors and enhance interactive teaching.

6.2 Future Works

Building upon the work presented in this thesis, several avenues for future research can be explored to further enhance our proposed action recognition system and extend its applicability to more real-world scenarios.

First, we plan to improve the generalization capability of our system by incorporating domain adaptation techniques. Since real-world environments often introduce variations in camera angles, lighting conditions, and background noise, we aim to explore unsupervised domain adaptation and self-supervised learning strategies to reduce the performance gap between different deployment environments.

Second, we intend to optimize the efficiency of our system by further refining the lightweight model deployment on edge devices. While our current implementation using the ONNX format ensures real-time performance, future work will focus on model quantization, pruning, and knowledge distillation to improve computational efficiency without sacrificing accuracy. Additionally, integrating advanced hardware acceleration techniques, such as TensorRT optimization, will help further reduce inference time and enhance real-time processing capabilities.

Third, we aim to extend the range of recognized actions to support a wider variety of applications. In addition to fall detection and hand-raising recognition, we plan to incorporate more interactive and context-aware actions relevant to elderly care and smart classroom. This will require collecting a more diverse dataset and designing models capable of handling more complex motion patterns.

Furthermore, we plan to enhance the interpretability of our action recognition models by integrating explainable AI (XAI) techniques. This will provide deeper insights into the decision-making process of our system, making it more transparent and reliable, especially for critical applications such as elderly fall detection and security monitoring.

In addition to the current application domains, future work may explore applications in areas such as public safety surveillance, industrial safety monitoring, and wildlife behavior analysis, where robust and real-time human or animal action recognition is equally important.

By addressing these aspects, we hope to push the boundaries of real-time action recognition and contribute to the broader adoption of intelligent human action recognition systems in real-world applications.

Bibliography

[1] M. B. Shaikh, D. Chai, S. M. S. Islam, and N. Akhtar, "From CNNs to Transformers in Multimodal Human Action Recognition: A Survey," ACM Transactions on Multimedia Computing, Communications and Applications, 2024.

[2] J. Jang, D. Kim, C. Park, M. Jang, J. Lee, and J. Kim, "ETRI-Activity3D: A Large-Scale RGB-D Dataset for Robots to Recognize Daily Activities of the Elderly," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 10990-10997, 2020.

[3] F.-C. Lin, H.-H. Ngo, C.-R. Dow, K.-H. Lam, and H. L. Le, "Student behavior recognition system for the classroom environment based on skeleton pose estimation and person detection," Sensors, vol. 21, no. 16, p. 5314, 2021.

[4] S. Das et al., "Toyota smarthome: Real-world activities of daily living," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 833-842.

[5] H. Zhu, R. Vial, and S. Lu, "Tornado: A spatio-temporal convolutional regression network for video action proposal," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5813-5821.

[6] K. A. Kibria, A. S. Noman, M. A. Hossain, M. S. I. Bulbul, M. M. Rashid, and A. S. M. Miah, "Creation of a Cost-Efficient and Effective Personal Assistant Robot using Arduino & Machine Learning Algorithm," in 2020 IEEE Region 10 Symposium (TENSYMP), 2020: IEEE, pp. 477-482.

[7] J. Pustejovsky and N. Krishnaswamy, "Embodied human computer interaction," KI-Künstliche Intelligenz, vol. 35, no. 3, pp. 307-327, 2021.

[8] S. Guan, "Skeleton-based Human Action Recognition: From 3D Pose Estimation to Action Recognition," 2023.

[9] D. Wu, “Video-based similar gesture action recognition using deep learning and GAN-based approaches,” 2019.

[10] C. Zhang, “Human Activity Analysis using Multi-modalities and Deep Learning. The City College of New York,” 2016.

[11] R. Kaur and S. Singh, “A comprehensive review of object detection with deep learning,” Digital Signal Processing, vol. 132, p. 103812, 2023.

[12] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, “Vitpose: Simple vision transformer baselines for human pose estimation,” Advances in Neural Information Processing Systems, vol. 35, pp. 38571-38584, 2022.

[13] Y. Li et al., “Tokenpose: Learning keypoint tokens for human pose estimation,” in Proceedings of the IEEE/CVF International conference on computer vision, 2021, pp. 11313-11322.

[14] C. Zheng et al., “Deep learning-based human pose estimation: A survey,” ACM Computing Surveys, vol. 56, no. 1, pp. 1-37, 2023.

[15] R. Liu, J. Shen, H. Wang, C. Chen, S.-c. Cheung, and V. Asari, “Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 5064-5073.

[16] Q. Peng, C. Zheng, and C. Chen, “A Dual-Augmentor Framework for Domain Generalization in 3D Human Pose Estimation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 2240-2249.

[17] Z. Zhang, “Microsoft kinect sensor and its effect,” IEEE multimedia, vol. 19, no. 2, pp. 4-10, 2012.

[18] D. Lee, J. Lee, and J. Choi, “CAST: cross-attention in space and time for video action recognition,” Advances in Neural Information Processing Systems, vol. 36, 2024.

[19] S. Grover, V. Vineet, and Y. Rawat, “Revealing the unseen: Benchmarking video action recognition under occlusion,” Advances in Neural Information Processing Systems, vol. 36, 2024.

[20] C. Feichtenhofer, “X3d: Expanding architectures for efficient video recognition,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 203-213.

[21] C.-Y. Wu et al., “Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 13587-13597.

[22] J. Lee, M. Lee, D. Lee, and S. Lee, “Hierarchically decomposed graph convolutional networks for skeleton-based action recognition,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 10444-10453.

[23] L. Wang and P. Koniusz, “3mformer: Multi-order multi-mode transformer for skeletal action recognition,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5620-5631.

[24] N. Trivedi and R. K. Sarvadevabhatla, “Psumnet: Unified modality part streams are all you need for efficient pose-based action recognition,” in European Conference on Computer Vision, 2022: Springer, pp. 211-227.

[25] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, “Constructing stronger and faster baselines for skeleton-based action recognition,” IEEE transactions on pattern analysis and machine intelligence, vol. 45, no. 2, pp. 1474-1488, 2022.

[26] M. M. Islam and T. Iqbal, “Hamlet: A hierarchical multimodal attention-based human activity recognition algorithm,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020: IEEE, pp. 10285-10292.

[27] J. Liu, C. Chen, and M. Liu, “Multi-modality co-learning for efficient skeleton-based action recognition,” in Proceedings of the 32nd ACM International Conference on Multimedia, 2024, pp. 4909-4918.

[28] K. Alomar, H. I. Aysel, and X. Cai, “RNNs, CNNs and Transformers in Human Action Recognition: A Survey and A Hybrid Model,” arXiv preprint arXiv:2407.06162, 2024.

[29] D. Liang, G. Fan, G. Lin, W. Chen, X. Pan, and H. Zhu, “Three-stream convolutional neural network with multi-task and ensemble learning for 3d action recognition,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2019, pp. 0-0.

[30] M. H. Javed, Z. Yu, T. Li, T. M. Rajeh, F. Rafique, and S. Waqar, "Hybrid two-stream dynamic CNN for view adaptive human action recognition using ensemble learning," International Journal of Machine Learning and Cybernetics, pp. 1-10, 2022.

[31] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 203-213.

[32] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in International conference on machine learning, 2015: PMLR, pp. 843-852.

[33] N. u. R. Malik, S. A. R. Abu-Bakar, U. U. Sheikh, A. Channa, and N. Popescu, "Cascading pose features with CNN-LSTM for multiview human action recognition," Signals, vol. 4, no. 1, pp. 40-55, 2023.

[34] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in Proceedings of the AAAI conference on artificial intelligence, 2018, vol. 32, no. 1.

[35] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 12026-12035.

[36] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, 2001, vol. 1: Ieee, pp. I-I.

[37] P. Viola and M. J. Jones, "Robust real-time face detection," International journal of computer vision, vol. 57, pp. 137-154, 2004.

[38] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), 2005, vol. 1: Ieee, pp. 886-893.

[39] D. G. Lowe, "Object recognition from local scale-invariant features," in Proceedings of the seventh IEEE international conference on computer vision, 1999, vol. 2: Ieee, pp. 1150-1157.

[40] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 4, pp. 509-522, 2002.

[41] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in 2008 IEEE conference on computer vision and pattern recognition, 2008: Ieee, pp. 1-8.

[42] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, “Cascade object detection with deformable part models,” in 2010 IEEE Computer society conference on computer vision and pattern recognition, 2010: Ieee, pp. 2241-2248.

[43] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, pp. 1627-1645, 2009.

[44] M. Baştan, “Multi-view object detection in dual-energy X-ray images,” Machine Vision and Applications, vol. 26, no. 7, pp. 1045-1060, 2015.

[45] D. Mery, G. Mondragon, V. Riffo, and I. Zuccar, “Detection of regular objects in baggage using multiple X-ray views,” Insight-Non-Destructive Testing and Condition Monitoring, vol. 55, no. 1, pp. 16-20, 2013.

[46] S. Akçay, M. E. Kundegorski, M. Devereux, and T. P. Breckon, “Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery,” in 2016 IEEE International Conference on Image Processing (ICIP), 2016: IEEE, pp. 1057-1061.

[47] M. M. Roomi, “Detection of concealed weapons in x-ray images using fuzzy k-nn,” International Journal of Computer Science, Engineering and Information Technology, vol. 2, no. 2, pp. 187-196, 2012.

[48] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.

[49] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp. 1904-1916, 2015.

[50] R. Girshick, ”Fast r-cnn,” in ICCV, 2015, pp.1440-1448.

[51] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137-1149, 2016.

[52] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117-2125.

[53] S. Akcay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, “Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery,” IEEE transactions on information forensics and security, vol. 13, no. 9, pp. 2203-2215, 2018.

[54] B. Wang, L. Zhang, L. Wen, X. Liu, and Y. Wu, “Towards real-world prohibited item detection: A large-scale x-ray benchmark,” in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 5412-5421.

[55] C. Miao et al., “Sixray: A large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2119-2128.

[56] Y. Zhang, Z. Su, H. Zhang, and J. Yang, “Multi-scale prohibited item detection in X-ray security image,” Journal of Signal Processing, vol. 36, no. 7, pp. 1096-1106, 2020.

[57] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759-8768.

[58] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781-10790.

[59] Q. Hou, D. Zhou, and J. Feng, “Coordinate attention for efficient mobile network design,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 13713-13722.

[60] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.

[61] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, “Unitbox: An advanced object detection network,” in Proceedings of the 24th ACM international conference on Multimedia, 2016, pp. 516-520.

[62] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, “Repulsion loss: Detecting pedestrians in a crowd,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7774-7783.

[63] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 658-666.

[64] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU loss: Faster and better learning for bounding box regression,” in Proceedings of the AAAI conference on artificial intelligence, 2020, vol. 34, no. 07, pp. 12993-13000.

[65] Z. Zheng et al., “Enhancing geometric factors in model learning and inference for object detection and instance segmentation,” IEEE transactions on cybernetics, vol. 52, no. 8, pp. 8574-8586, 2021.

[66] A. Toshev and C. Szegedy, “Deeppose: Human pose estimation via deep neural networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1653-1660.

[67] B. Xiao, H. Wu, and Y. Wei, “Simple baselines for human pose estimation and tracking,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 466-481.

[68] J. Wang, X. Long, Y. Gao, E. Ding, and S. Wen, “Graph-pcnn: Two stage human pose estimation with graph pose refinement,” in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23 – 28, 2020, Proceedings, Part XI 16, 2020: Springer, pp. 492-508.

[69] Y. Cai et al., “Learning delicate local representations for multi-person pose estimation,” in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, 2020: Springer, pp. 455-472.

[70] S. Yang, Z. Quan, M. Nie, and W. Yang, “Transpose: Keypoint localization via transformer,” in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 11802-11812.

[71] Y. Yuan et al., "Hrformer: High-resolution vision transformer for dense predict," Advances in neural information processing systems, vol. 34, pp. 7281-7293, 2021.

[72] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 5386-5395.

[73] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7291-7299.

[74] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016, pp. 4724-4732.

[75] S. Kreiss, L. Bertoni, and A. Alahi, "PifPaf: Composite Fields for Human Pose Estimation," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11969-11978, 2019.

[76] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 5693-5703.

[77] M. Kocabas, S. Karagoz, and E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 417-433.

[78] S. Li and A. B. Chan, "3d human pose estimation from monocular images with deep convolutional neural network," in Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12, 2015: Springer, pp. 332-347.

[79] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2602-2611.

[80] G. Pavlakos, X. Zhou, and K. Daniilidis, "Ordinal depth supervision for 3d human pose estimation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7307-7316.

[81] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2640-2649.

[82] M. Wang, X. Chen, W. Liu, C. Qian, L. Lin, and L. Ma, "Drpose3d: Depth ranking in 3d human pose estimation," arXiv preprint arXiv:1805.08973, 2018.

[83] H. Ci, C. Wang, X. Ma, and Y. Wang, "Optimizing network structure for 3d human pose estimation," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 2262-2271.

[84] Z. Zou and W. Tang, "Modulated graph convolutional network for 3D human pose estimation," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 11477-11487.

[85] Z. Zhang, "Microsoft kinect sensor and its effect," IEEE multimedia, vol. 19, no. 2, pp. 4-10, 2012.

[86] H. Farhadi Tolie, J. Ren, M. J. Hasan, and S. Kannan, "Enhancing underwater situational awareness: RealSense camera integration with deep learning for improved depth perception and distance measurement," Artificial Intelligence for Security and Defence Applications II, 2024.

[87] S. Lee, "Depth camera image processing and applications," in 2012 19th IEEE International Conference on Image Processing, 2012: IEEE, pp. 545-548.

[88] B. Kang, S.-J. Kim, S. Lee, K. Lee, J. D. Kim, and C.-Y. Kim, "Harmonic distortion free distance estimation in ToF camera," in Three-Dimensional Imaging, Interaction, and Measurement, 2011, vol. 7864: SPIE, pp. 28-36.

[89] N. Andriyanov et al., "Intelligent system for estimation of the spatial position of apples based on YOLOv3 and real sense depth camera D415," Symmetry, vol. 14, no. 1, p. 148, 2022.

[90] J. Gai, L. Xiang, and L. Tang, "Using a depth camera for crop row detection and mapping for under-canopy navigation of agricultural robotic vehicle," Computers and Electronics in Agriculture, vol. 188, p. 106301, 2021.

[91] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition," in European Conference on Computer Vision, 2016.

[92] W. Zhu et al., "Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks," in AAAI Conference on Artificial Intelligence, 2016.

[93] H. Wang and L. Wang, "Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3633-3642, 2017.

[94] C. Xie et al., "Memory Attention Networks for Skeleton-Based Action Recognition," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, pp. 4800-4814, 2018.

[95] J. Liu, A. Shahroudy, G. Wang, L.-y. Duan, and A. C. Kot, "Skeleton-Based Online Action Prediction Using Scale Selection Network," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, pp. 1453-1467, 2019.

[96] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation," in International Joint Conference on Artificial Intelligence, 2018.

[97] Y. Xu, J. Cheng, L. Wang, H. Xia, F. Liu, and D. Tao, "Ensemble one-dimensional convolution neural networks for skeleton-based action recognition," IEEE Signal Processing Letters, vol. 25, no. 7, pp. 1044-1048, 2018.

[98] H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 2969-2978.

[99] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE transactions on neural networks, vol. 20, no. 1, pp. 61-80, 2008.

[100] C.-H. Lin, P.-Y. Chou, C.-H. Lin, and M.-Y. Tsai, "SlowFast-GCN: A Novel Skeleton-Based Action Recognition Framework," in 2020 International Conference on Pervasive Artificial Intelligence (ICPAI), 2020: IEEE, pp. 170-174.

[101] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5115-5124.

[102] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling gcn with dropgraph module for skeleton-based action recognition," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16, 2020: Springer, pp. 536-553.

[103] Z. Chen, S. Li, B. Yang, Q. Li, and H. Liu, "Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition," in Proceedings of the AAAI conference on artificial intelligence, 2021, vol. 35, no. 2, pp. 1113-1122.

[104] Z. Qin et al., "Fusing higher-order features in graph neural networks for skeleton-based action recognition," IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 4, pp. 4783-4797, 2022.

[105] X. Shu, J. Yang, R. Yan, and Y. Song, "Expansion-squeeze-excitation fusion network for elderly activity recognition," IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 8, pp. 5281-5292, 2022.

[106] T. T. Zin et al., "Real-time action recognition system for elderly people using stereo depth camera," Sensors, vol. 21, no. 17, p. 5895, 2021.

[107] N. E. Tabbakha, W.-H. Tan, and C.-P. Ooi, "Elderly action recognition system with location and motion data," in 2019 7th International Conference on Information and Communication Technology (ICoICT), 2019: IEEE, pp. 1-5.

[108] H. Zhou, F. Jiang, and R. Shen, "Who are raising their hands? Hand-raiser seeking based on object detection and pose estimation," in Asian Conference on Machine Learning, 2018: PMLR, pp. 470-485.

[109] J. Si, J. Lin, F. Jiang, and R. Shen, "Hand-raising gesture detection in real classrooms using improved R-FCN," Neurocomputing, vol. 359, pp. 69-76, 2019.

[110] W. Liao, W. Xu, S. Kong, F. Ahmad, and W. Liu, "A two-stage method for hand-raising gesture recognition in classroom," in Proceedings of the 2019 8th international conference on educational and information technology, 2019, pp. 38-44.

[111] T. Franzel, U. Schmidt, and S. Roth, "Object Detection in Multi-view X-Ray Images," in DAGM/OAGM Symposium, 2012.

[112] D. Mery, E. Svec, and M. Arias, "Object Recognition in Baggage Inspection Using Adaptive Sparse Representations of X-ray Images," in Pacific-Rim Symposium on Image and Video Technology, 2015.

[113] J. Ding, S. Chen, and G. Lu, "X-ray security inspection method using active vision based on Q-learning algorithm," Journal of Computer Applications, vol.38, no.12, pp.3414–3418, 2018.

[114] D. Mery, E. Svec, M. Arias, V. Riffo, J. M. Saavedra, and S. Banerjee, "Modern Computer Vision Techniques for X-Ray Testing in Baggage Inspection," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, pp. 682-692, 2017.

[115] Y. Wei, R. Tao, J. Z. Wu, Y. Ma, L. Zhang, and X. Liu, "Occluded Prohibited Items Detection: An X-ray Security Inspection Benchmark and De-occlusion Attention Module," Proceedings of the 28th ACM International Conference on Multimedia, 2020.

[116] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6154-6162, 2017.

[117] F. Shao, J. Liu, P. Wu, Z. Yang, and Z. Wu, "Exploiting foreground and background separation for prohibited item detection in overlapping X-Ray images," Pattern Recognit., vol. 122, p. 108261, 2022.

[118] T. Hassan, S. H. Khan, S. Akçay, Bennamoun, and N. Werghi, "Cascaded Structure Tensor Framework for Robust Identification of Heavily Occluded Baggage Items from Multi-Vendor X-ray Scans," arXiv: Computer Vision and Pattern Recognition, 2019.

[119] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987-5995, 2016.

[120] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," ACM SIGGRAPH 2003 Papers, 2003.

[121] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1800-1807, 2016.

[122] T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” in European Conference on Computer Vision, 2014.

[123] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” International journal of computer vision, vol. 88, pp. 303-338, 2010.

[124] Z. Wang, H. Zhang, Z. Lin, et al., “Prohibited Items Detection in Baggage Security Based on Improved YOLOv5,” 2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI). IEEE, 2022.

[125] H. Fang et al., “AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, pp. 7157-7173, 2022.

[126] Y. Kong and Y. R. Fu, “Human Action Recognition and Prediction: A Survey,” International Journal of Computer Vision, vol. 130, pp. 1366 - 1401, 2018.

[127] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation Networks,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132-7141, 2017.

[128] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-y. Duan, and A. C. Kot, “NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, pp. 2684-2701, 2019.

[129] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, ”NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1010-1019, 2016.

[130] X. Ding et al., “UniRepLKNet: A Universal Perception Large-Kernel ConvNet for Audio, Video, Point Cloud, Time-Series and Image Recognition,” 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5513-5524, 2023.

[131] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, “Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition,” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 13339-13348, 2021.

[132] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, “Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5457-5466, 2018.

[133] H. Wang and L. Wang, “Beyond Joints: Learning Representations From Primitive Geometries for Skeleton-Based Action Recognition and Detection,” IEEE Transactions on Image Processing, vol. 27, pp. 4382-4394, 2018.

[134] J. Hu, W. Zheng, J. Pan, J. Lai, and J. Zhang, “Deep Bilinear Learning for RGB-D Action Recognition,” in European Conference on Computer Vision, 2018.

[135] Y.-H. Wen, L. Gao, H. Fu, F.-L. Zhang, and S. Xia, “Graph CNNs with motif and variable temporal block for skeleton-based action recognition,” in Proceedings of the AAAI conference on artificial intelligence, 2019, vol. 33, no. 01, pp. 8989-8996.

[136] J. Liu, G. Wang, L.-y. Duan, K. Abdiyeva, and A. C. Kot, “Skeleton-Based Human Action Recognition With Global Context-Aware Attention LSTM Networks,” IEEE Transactions on Image Processing, vol. 27, pp. 1586-1599, 2017.

[137] M. Perez, J. Liu, and A. C. Kot, “Interaction Relational Network for Mutual Action Recognition,” IEEE Transactions on Multimedia, vol. 24, pp. 366-376, 2019.

[138] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, “Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3590-3598, 2019.

[139] L. Ke, K.-C. Peng, and S. Lyu, “Towards to-at spatio-temporal focus for skeleton-based action recognition,” in Proceedings of the AAAI conference on artificial intelligence, 2022, vol. 36, no. 1, pp. 1131-1139.

[140] Y. Pang, Q. Ke, H. Rahmani, J. Bailey, and J. Liu, “Igformer: Interaction graph transformer for skeleton-based human interaction recognition,” in European Conference on Computer Vision, 2022: Springer, pp. 605-622.

[141] Z. Gao et al., “Focal and global spatial-temporal transformer for skeleton-based action recognition,” in Proceedings of the Asian Conference on Computer Vision, 2022, pp. 382-398.

[142] L. Pan, J. Lu, and X. Tang, "Spatial-temporal graph neural ODE networks for skeleton-based action recognition," Scientific Reports, vol. 14, 2024.

[143] X. Zheng, J. Cao, C. Wang, and P. Ma, "A High-Precision Human Fall Detection Model Based on FasterNet and Deformable Convolution," Electronics, 2024.

[144] X. Wang, J. Ellul, and G. Azzopardi, "Elderly Fall Detection Systems: A Literature Survey," Frontiers in Robotics and AI, vol. 7, 2020.

[145] L. Ren and Y. Peng, "Research of Fall Detection and Fall Prevention Technologies: A Systematic Review," IEEE Access, vol. 7, pp. 77702-77722, 2019.

[146] O. Díaz-Parra et al., "Smart Education and future trends," Int. J. Comb. Optim. Probl. Informatics, vol. 13, pp. 65-74, 2022.

[147] S. Mouti and H. Al-Chalabi, "A smart system for student performance assessment (SPA)," Scientific Reports, vol. 14, 2024.

[148] Microsoft, "Azure Kinect Body Tracking SDK 1.1.x documentation," [Online]. Available: https://microsoft.github.io/Azure-Kinect-Body-Tracking/release/1.1.x/index.ht ml. [Accessed: Feb. 6, 2025].