Doctoral Dissertation

Recognizing Actions from Videos by Unsupervised Deep Neural Networks

(教師なしディープニューラルネットワークを用いた動画像からの人間動作認識)

Jiaxin Zhou

Graduate School of Science and Engineering,
Saitama University

Supervisor: Professor Takashi Komuro

December 2022

Abstract

In this thesis, we aim to study the problem of learning actions from videos by deep neural networks that are trained with less or no manual labels, which belongs to unsupervised learning. Supervised learning methods obtained great improvements in action classification by learning abundant manual and well-labeled data, however, it is labor and time-consuming to manually make such data, and it is also subjective and difficult to decide how exactly the annotations of action videos should be. Therefore, incomplete or even automatically-generated supervision signals that can help with action recognition are expected.

We first propose a method to recognize fall actions from videos without fine-grained labels, in which annotations of fall actions are not needed by utilizing learning of abundant Activity of Daily Life (ADL) videos. The first variational auto-encoder (VAE) of the method learns representations of ADL videos only by compressing those videos, and the second VAE gathers representations of ADL data and fall action data into two clusters. The experimental results showed that our method achieved better generalization ability compared to methods using supervised learning with well-labeled data. When the method evaluated data that is different from training data on scenes, subjects, etc., it achieved a 10% improvement compared to supervised learning methods.

Then, we propose a method for general action representation learning using skeleton sequences, in which a structure-asymmetrical auto-encoder is used to learn spatiotemporal representations under the supervision of salient skeleton motion cues. Since the supervision signals are automatically generated by a program in advance, our method is unsupervised and does not rely on manual annotations to associate skeleton sequences with actions. The experimental results showed the effectiveness of the proposed representation learning, and improvements compared with skeleton-based generative learning methods. When the proposed network was fine-tuned with partially labeled data, our results also outperformed some fully-supervised methods.

Finally, we propose a method for general action representation learning which is trained with paired videos and skeleton sequences and is evaluated using videos only. The proposed neural network simultaneously implements the prediction of position relationships of movements with salient pixel-value changes and multimodality-contrastive learning between representations that are respectively extracted from videos and skeleton sequences. In addition to not relying on manual annotations to associate input data with actions, the method can save time and memory space for devices, because sparse parts of videos are taken as training data instead of entire videos, which are picked up according to probabilistic values of the size of pixel-value changes of movements. In experiments using supervised settings, the proposed method trained with sparse parts of videos that are picked up according to probabilistic values obtained 30% and 18% improvements in classification performance on two datasets compared to a method trained with entire videos. In experiments using unsupervised settings, our method achieved state-of-the-art performance. The experimental results demonstrate the superiority of the proposed method, which efficiently learns discriminative features.

Acknowledgments

I wish to express my most sincere gratitude and appreciation to my thesis supervisor Prof. Takashi Komuro for his invaluable guidance, patience, and encouragement during my Ph.D. course. If without his detailed advice and precious experience, this Ph.D. dissertation will not come true. I also learned techniques of academic communication, presentation, and writing papers from his invaluable guidance.

I also would like to express my great gratitude and appreciation to my dissertation committee: Prof. Tetsuya Shimamura, Prof. Atsushi Uchida, and Prof. Jun Ohkubo for their valuable feedback and insightful suggestions for this research.

Furthermore, I really thank Saitama University, Mitsubishi Corporation International Scholarship, and JASSO Scholarship. They provided great support on aspects of academics, life, and finance which is crucial for my Ph.D. study.

I also greatly appreciate and acknowledge direct or indirect help from every one of the laboratory and the university. I thank their kindness and patience which help me to overcome difficulties and solve problems throughout my Ph.D. course.

Last, but not least, I would like to give my special thanks and great respects to my parents. They sustain me living and studying abroad, and finally finishing my doctoral degree.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

With the development of deep learning, supervised learning methods obtained excellent performance in image classification. For example, in a competition named ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015, the method that uses a supervised deep neural network named residual network [30] surpassed human performance for the first time. However, supervised learning methods perform well when training with abundant, balanced, and well-labeled data. If supervised neural networks cannot be trained with such data, they will easily suffer from overfitting problems, and the performance underperforms, which leads to a series of research topics such as weakly-supervised learning, self-supervised learning, unsupervised learning, etc.

Human action recognition is a crucial topic in computer vision since it plays a fundamental role in a wide range of applications, such as video surveillance, human–computer interaction, video understanding for retrieval, etc. It is usually difficult to label abundant data with precise tags as time and labor-consuming such as taking thousands of hours of videos and annotating each frame with an action tag like falling, sitting, drinking water, or others. Besides, it is subjective and difficult to decide how exactly the annotations of the action of videos should be. Therefore, in this thesis, we aim to study the problem of learning human actions from videos by using deep neural networks with less, or even no manual label.

Videos are skeletons are most two data modalities that are used for action recognition. With the rapid development of convolutional neural networks (CNN), CNN-based methods have achieved significant success in action classification from videos [9, 81, 75, 88, 32]. Visual cues (e.g., RGB images and depth images) provide discriminative features for action recognition, whereas learned features also include bias from viewpoints, the appearance of actors, backgrounds, and many other factors that can adversely affect recognition performance. In some studies, skeletons are extracted from images, and then recurrent neural networks (RNN) are used to model skeleton movements in the temporal dimension for extracting more robust features of actions [39, 16, 50, 102, 74]. Not only can 3D joint coordinates be acquired from low-cost human skeleton capture devices (e.g., Kinect), but also there have been extensive studies of 3D human pose estimation algorithms [44, 68, 49] in recent years. Therefore, we utilize videos are skeletons for studying action recognition.

1.2 Literature Review

1.2.1 Learning Approach

Data and associated labels are utilized by supervised learning methods. Its goal is to learn a function (i.e. a neural network) that maps input data (e.g. images, videos, etc.) to labels. A part of the data has associated labels, and the rest has

Learning Approach	What kind of labeled data is it?			
	manual	partial	incomplete	pseudo
Supervised Learning	√			
Semi-supervised Learning	√	√		
Weakly-supervised Learning	√		√	
Self-supervised Learning	×	×	×	√
Unsupervised Learning	×	×	×	×

Table 1.1: Comparison of different learning approaches with labeled data.

not at all, which are utilized by semi-supervised learning methods [84]. Data with incomplete (i.e. not precise) but manual labels are utilized by weakly-supervised learning methods [101]. Data with pseudo labels which can be made by programs are utilized by self-supervised learning methods [52]. Unsupervised learning methods utilize data only. For self-supervised learning, since pseudo labels are made from original data, it usually is considered a type of unsupervised learning. A comparison of different learning approaches is shown in Table. 1.1.

1.2.2 Supervised Action Recognition

Before the age of deep learning, various hand-crafted descriptors [67, 72, 92] were proposed to represent the features of actions. In recent years, considering the powerful convolutional neural networks (CNNs) developed for classification, many 3D-CNN-based architectures [81, 9, 88] have been used to extract spatiotemporal features from RGB (or depth) videos for action recognition.

Deep learning methods based on recurrent neural networks (RNNs) also perform well for classifying sequential skeleton data. In order to better capture long-term contextual information of skeleton sequences, the physical structure of human skeletons was considered. Du et al. [16] and Evangelidis et al. [18] proposed a method in which the whole human skeleton is split into several parts according to the physical structure, these parts are fed into different neural networks, and finally, the outputs are fused hierarchically. Shahroudy et al. [74] and Zhu et al. [102] proposed novel network structures to capture co-occurrences of joints in actions (i.e., joints moving together in groups) to improve recognition performance.

Originally, the neural network, Transformer [85], is an improvement of RNN to process sequential data in the field of natural language processing, in which the relationship of tokens (i.e. words) are captured by using self-attention mechanism [61, 3]. Recently, the Transformer is introduced into the field of computer vision such as Vision Transformer (ViT) [14] and Video Vision Transformer (ViViT) [2], in which, images or videos are divided to sequences of patches and are processed by Transformer-like neural networks.

1.2.3 Unsupervised Action Representation Learning

Various pretext tasks are utilized for unsupervised action representation learning. Since supervision signals of learning with pretext tasks are automatically made from original data by programs, it is also called self-supervised learning.

Some studies focus on using vision data, such as images and optical flow, and mining correlations of spatial and temporal arrangements among frames. In [78]'s study, videos were used to learn motion patterns in temporal intervals, and it was proposed to generate missing frames, reconstruct input frames, and predict future frames simultaneously. In [54]'s study, it was proposed to learn visual features by predicting optical flow information from input RGB or depth videos. In [87]'s study, it was proposed to learn visual features from videos by regressing both motion and appearance statistics (e.g., the dominant color, the largest and smallest color diversity locations, the dominant orientation of the largest motion, etc.) in videos. Those methods learned features in short temporal intervals, but long-term motion dependencies were lost. In [60]'s study, it was proposed to learn visual features by predicting the correct order for input videos with shuffled frames. In [35]'s study, an extended method was proposed in which videos were given as space-time cubes, and were separated into several cuboid puzzle blocks and shuffled, and then their correct arrangements were predicted. Those methods have the disadvantage that the learning was strongly based on local semantic features, and long-term temporal features were neglected.

Some studies focus on using skeleton data, and mining correlations of spatiotemporal arrangements among joints. In [100, 79]'s studies, encoder-decoder-based networks were proposed to reconstruct skeleton data, where an encoder learns to compress skeleton data to latent representations, and a decoder learns to generate original skeleton data from the learned representations. In [47]'s study, a method with a multi-task training strategy was proposed, where the network reconstructed input skeletons, and simultaneously, the network also generated masked input skeletons and predicted the correct order of shuffled skeletons.

1.3 Objectives and Contribution

In this thesis, we propose three methods for the purpose of learning human actions by using deep neural networks that are trained with fewer or no manual labels. A comparison of the three proposed methods is shown in Fig. 1.1.

In the first method, we propose a framework for detecting fall actions from videos to solve the problem of imbalance between fall action data and Activity of Daily Life (ADL) data by utilizing weakly-supervised learning and unsupervised clustering learning. Since surveillance videos contain abundant activities of the daily life of the elderly, we utilize learning ADL data to recognize fall actions. The first variational auto-encoder (VAE) in the framework learns representations of ADL data by compressing videos, and the second VAE gathers representations of ADL data and fall action data into two clusters. The experimental results showed that our method achieved a promising level of accuracy and better generalization ability compared to methods using supervised learning with well-labeled data. This method utilizes supervised information of ADL data which are manually labeled, and its application is limited to binary classification tasks. Therefore, we proposed the second method to further reduce the use of manual labels which can be applied to multi-class classification tasks.

In the second method, we propose a framework for unsupervised representation learning of skeleton sequences by using a structure-asymmetrical auto-encoder that learns spatiotemporal representations under the supervision of salient skeleton motion cues. The supervision signals are automatically generated by a program. The structure-asymmetrical auto-encoder captures not only correlations of adjacent joints but also long-term motion dependencies by using the proposed unsupervised training, which leads to the advantage that similar movements are gathered

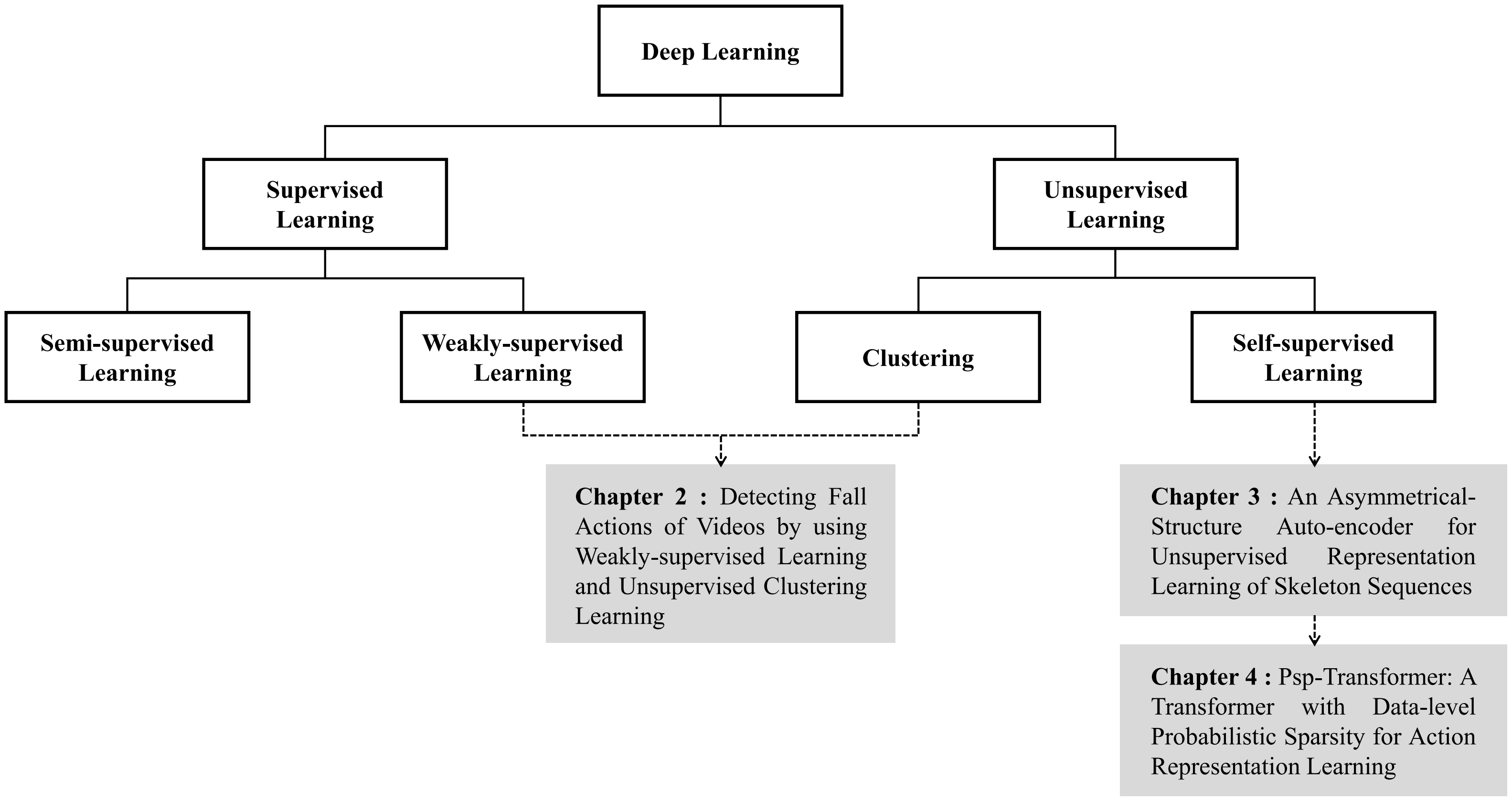

Deep Learning
Supervised Learning
Unsupervised Learning
Semi-supervised Learning
Weakly-supervised Learning
Clustering
Self-supervised Learning
Chapter 2 : Detecting Fall Actions of Videos by using Weakly-supervised Learning and Unsupervised Clustering Learning
Chapter 3 : An Asymmetrical-Structure Auto-encoder for Unsupervised Representation Learning of Skeleton Sequences
Chapter 4 : Psp-Transformer: A Transformer with Data-level Probabilistic Sparsity for Action Representation Learning

Figure 1.1: The positioning of the three proposed methods.

around the same cluster, whereas different movements are gathered around distinct clusters. The experimental results showed the effectiveness of the proposed representation learning, and improvements compared with skeleton-based generative learning methods. When the proposed network was fine-tuned with partially labeled data, our results also outperformed some fully-supervised methods. Since skeleton coordinates are not the only important thing for action recognition, and the appearance of objects and subjects are also important, we proposed the third method which utilizes both skeletons and videos.

In the third method, we propose a neural network for action representation learning which is learned from spatiotemporal signals of salient pixel-value changes and salient skeleton motion cues using both videos and skeleton sequences. The network simultaneously implements the prediction of position relationships of movements with salient pixel-value changes using a vision transformer and multimodality-contrastive learning between representations respectively learned from videos and skeleton sequences. In experiments using supervised settings, our proposed network obtained remarkable generalization ability and higher accuracies. In experiments using unsupervised settings, our method achieved state-of-the-art performance. The experimental results demonstrate the superiority of the proposed method which efficiently learns discriminative features.

Chapter 2

Detecting Fall Actions of Videos by using Weakly-supervised Learning and Unsupervised Clustering Learning

2.1 Introduction

Population aging is a widespread problem across the world and is very severe in highly developed countries. Since solitary elderly people are more likely to fall indoors and cannot obtain assistance in time, demands for stable fall-detecting systems are increasing. However, it is challenging to detect whether a person falls by using computer vision due to complicated real-life situations.

Until the age of deep learning, hand-crafted features were extracted from images and were used to detect fall actions. However, they are not sufficient to discriminate fall actions due to complicated human behaviors, viewpoints of cameras, and other factors. Recently, deep learning methods using supervised neural networks have proposed to detect fall actions. In those methods [22, 66], neural networks were used to classify input data as fall actions or normal actions by training with manually labeled data. If supervised neural networks are trained with enough, balanced and well-labeled data, they perform well. However, there is a problem in the field of fall detection that well-labeled data is not abundant, since it is labor-consuming to label each frame of hours of videos with tags of falling, sitting, drinking water, and other actions in daily life. There is another problem that quantities Activity of Daily Life (ADL) data and fall action data are imbalanced. In most videos, fall actions do not happed or happen only within several seconds, and the rest is about activities of daily life. Those problems can adversely affect performance of supervised neural networks.

Therefore, some researchers proposed utilizing the idea of anomaly detection [37] for fall detection, since there is sometimes imbalance between regular events and anomalous events like ADL data and fall action data. Some methods using unsupervised neural networks [99, 12] were proposed to overcome the imbalance between anomalous data and regular data. In these methods, auto-encoders (AE) that are a kind of unsupervised neural network and do not depend on well-labeled data were used to detect abnormal events. First, AE-based networks learn a distribution of regular videos by compressing and reconstructing regular videos. When training is finished, an abnormal sample is input to the networks, and still is reconstructed to be normal, which makes reconstruction errors large, and the sample is classified as an anomaly. Those methods belong to weakly supervised learning, which uses data with imprecise labels, since training data has imprecise labels, namely training data only contains regular videos.

In this study, we propose a framework for fall detection where a Variational Auto-encoder (VAE) [36] with 3D-convolutional residual blocks [29] learns a distribution of ADL videos, and another AE with fully connected layers learns to cluster representations that belong to the distribution of ADL videos or the other one of fall action videos into two distinct clusters. Besides, a region extraction technique for enhancing accuracy is proposed to make the VAE focus on human actions. In comparison with the study [65], the difference is our finding that a low ratio of a motion region to the entire image can adversely affect the performance of the neural network when using RGB images instead of depth and thermal images. Therefore, we use a region extraction technique to increase the ratio so that the neural network can focus on learning human motions. We also verify that a combination of weakly supervised learning and unsupervised cluster learning can be used to ease the lack of well-labeled data, and abundant ADL data can be taken good use of to overcome the adverse effect of imbalanced data. The proposed framework is expected to obtain better accuracies and generalization ability than methods using supervised learning with well-labeled data.

2.2 Related Research

2.2.1 Fall Detection

Until the age of deep learning, handcrafted features were used to detect fall actions, such as fitting an ellipse to a body [63, 46]. In these methods, whether a fall action happens or not is detected depending on variations of the short and long axis, the area, etc. of the fitted ellipse in videos. For example, if a vertical and thin ellipse becomes horizontal, it may indicate that a fall action happened. Such hand-crafted features are not sufficient to discriminate fall actions.

In some studies, handcrafted features were used as inputs to neural networks for for fall detection. For example, skeleton information is extracted by using Microsoft Kinect, treated as biomechanical features, and then are used as inputs to a recurrent neural network with long short-term memory units [95]. Since visual information is lost when extracting skeleton information, and the performance may become unstable, it would be more appropriate to directly use pixel-level information to training neural networks. Due to complicated human behaviors, viewpoints of cameras, and other factors, it is more reasonable to automatically extract features by using deep neural networks.

Recently, supervised learning methods using deep neural networks have been proposed to detect fall actions [22, 66, 1]. In these methods, a neural network is trained with manually labeled data to classify input data as a fall action or a normal action. Well-labeled data usually is precious since it is tiresome to label tags on a large amount of data, and lacking well-labeled data will lead to overfitting of supervised neural networks. Moreover, the amount of fall action data and the amount of ADL data are imbalanced. Therefore, a large amount of ADL data is abandoned to keep the balance and allow supervised neural networks to work normally, which further aggravates the lack of well-labeled data.

2.2.2 Weakly Supervised Learning

Since there is an imbalance between fall action data and ADL data like anomalous videos and regular videos, the idea of anomaly detection [37] was proposed to detect fall actions. Anomaly detection is a kind of weakly supervised learning method since training data with imprecise labels is used. In the case of fall detection, all training data is from ADL videos.

Zhao et al. [99] and Chong et al. [12] proposed a spatiotemporal auto-encoder to detect abnormal events. In these methods, AE-based networks were used to model regular video data, and the networks learn how to reconstruct regular videos. When training is finished, if an abnormal sample is input, the networks still try to reconstruct it to be a regular video, which makes reconstruction errors large, and the sample is classified as an anomaly. The higher the reconstruction error is, the more possibly an abnormal event happens. Nogas et al. [65] also proposed an AE for fall detection and conducted experiments using a fall dataset consisting of thermal and depth images.

2.2.3 Unsupervised Clustering Learning

Using handcrafted thresholds of reconstruction errors cannot detect fall actions well in various situations, but metrics learned for different situations such as clustering learning can overcome that weakness.

Some classical clustering methods such as k-means [56] and Gaussian Mixture Models [7] tend to suffer from the curse of dimensionality [5] when high-dimensional data such as videos are input. Various clustering methods such as spectral clustering [55], density-based clustering [17], etc., are proposed to take good used of more flexible distance metrics to process high-dimensional data. However, they lead to other problems such as memory and time-consuming.

With the development of deep learning, some methods assembled both representation learning and clustering learning. Xie et al. [93] proposed Deep Embedding Clustering (DEC) in which an auto-encoder compresses the dimensionality of input data, and minimized the KL divergence between predictions and auxiliary target distribution. DEC achieved progressive performance on clustering tasks. Jiang et al. [31] and Nat et al. [13] assumed that low-dimensional latent space of compressed input data follows a mixture of gaussian distribution and proposed Variational Deep Embedding (VaDE) and Gaussian Mixture VAE (GMVAE) respectively. We refer to [59] for a comprehensive literature study of clustering with deep learning.

2.3 Method

We propose a 3D-convolutional VAE as shown in the orange part of Fig. 2.1 that learns representations of ADL actions by reconstructing ADL videos since videos include lots of redundant information and should be reduced to lower dimensionality. Many previous studies simply used reconstruction errors of the VAE to detect fall actions. However, handcrafted thresholds of reconstruction errors are not scalable to new data that is different from training data, which means generalization ability is weak. Therefore, we propose another AE with fully connected laters as shown in the blue part of Fig. 2.1 to classify those action representations by using clustering learning.

2.3.1 Weakly Supervised Representation Learning

We adopt a VAE with 3D-convolutional residual blocks (omitted to ResVAE for short) as shown in the orange part of Fig.2.1 to model ADL videos and trained it by minimizing reconstruction errors, which are the mean square error between an input samples and reconstructed samples. A VAE is a kind of unsupervised deep

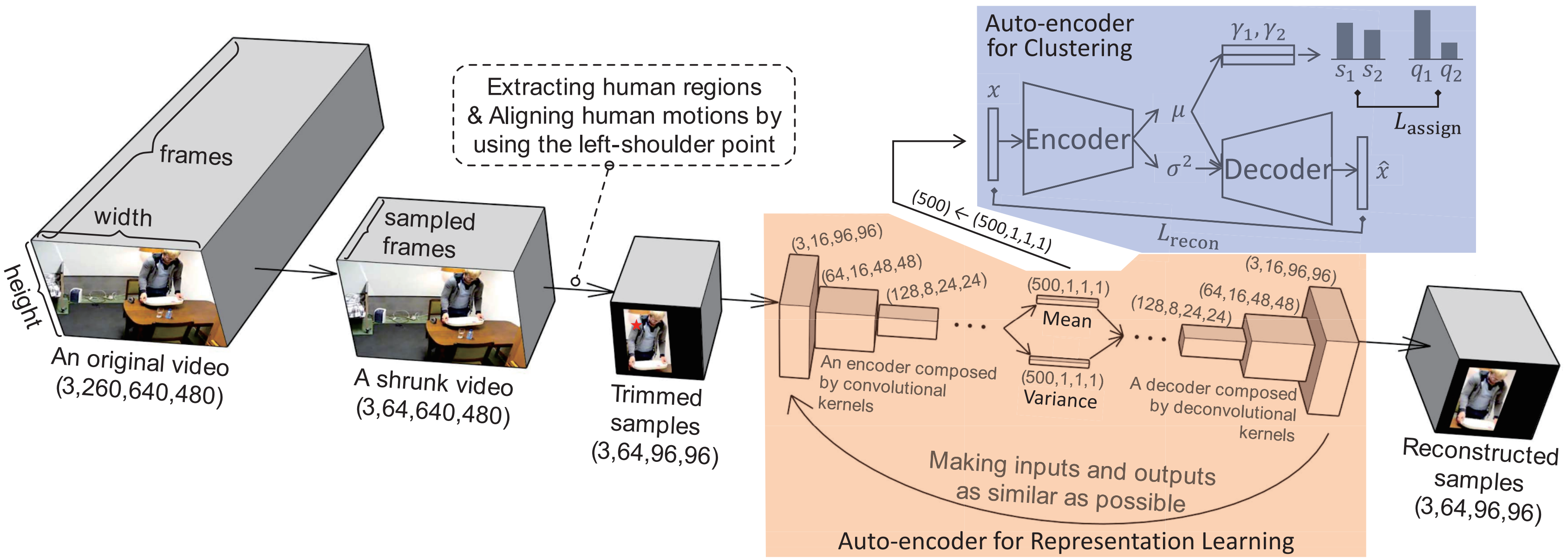


Figure 2.1: Overview of the proposed method that has three parts, pre-processing, the first auto-encoder and the second auto-encoder. In the part of pre-processing, human regions are extracted, and human motions are aligned by using a joint point (e.g. the left-shoulder). In the first auto-encoder, representations are learned by compressing and reconstructing videos. In the second auto-encoder, representations are gathered into two clusters.

learning architecture and is suitable for modeling training data without labels. However, the VAE is weakly supervised in this study, since we limit all training data to be ADL videos. To compare with AEs, which learn compressed representations of training data, VAEs learn the parameters of a probability distribution of latent variables. Using residual blocks in deep neural networks can avoid the problem of vanishing gradients when using deeper layers. In recent years, neural networks with 3D-convolution have been demonstrated as an effective model for learning local motion features from videos. We therefore combine a VAE, residual blocks, and 3D-convolution in our proposed method. Since the network is trained to reconstruct ADL videos, it learns a distribution of ADL actions in low-dimensional latent space. After training is finished, reconstruction errors of ADL samples are limited within a certain range. In contrast, the reconstruction error of an abnormal sample should be greater than that range.

2.3.2 Unsupervised Clustering Learning

Handcrafted thresholds of reconstruction errors are usually used to distinguish fall actions since fall action data is reconstructed worse than ADL data by the VAE in previous research [99, 12, 65]. However, it is more reasonable to directly cluster features in the high-dimensional latent space using a deep neural network. Therefore, we propose to use another VAE with fully connected layers (omitted to FCVAE for short) to learn cluster centers of distributions of ADL data and fall action data as shown in Fig. 2.1. Representations $\{x_1, x_2, ..., x_i, ...\}$, $x_i \in \mathbb{R}^D$, where D denotes dimensionality, are extracted by the trained encoder of 3D-convolutional VAE from both ADL data and fall action data and are taken as a set. Then, the FCVAE learns to distinguish between ADL data and fall action data by simultaneously learning two cluster centers denoted by γ_1, $\gamma_2 \in \mathbb{R}^D$ and reconstructing input data.

In the branch of clustering learning, cluster centers are firstly initialized by using k-means, and similarity s_{ij} between representation embeddings $\{\mu_1, \mu_2, ..., \mu_i, ...\}$,

$\mu_i \in \mathbb{R}^D$ and cluster centers in the high-dimensional latent space is calculated using Student's t-distribution following [93, 83]:

$$s_{ij} = \frac{(1 + \|\mu_i - \gamma_j\|^2)^{-1}}{\sum_{j'}(1 + \|\mu_i - \gamma_{j'}\|^2)^{-1}} \ ,$$

where $i = 1, 2, ..., N, \ j = 1, 2$. Those values of similarities are normalized between 0 and 1 using

$$s_{ij} \leftarrow \frac{s_{ij}}{\sum_{j'} s_{ij'}}$$

and compose a distribution (s_{i1}, s_{i2}) which represents label assignment possibility. We additionally apply a sharpening function to obtain pseudo ground truth [6, 26]

$$q_{ij} = f_{\text{sharpen}}(s_{ij}) = \frac{s_{ij}^2}{\sum_{j'} s_{ij'}^2}$$

which can reduce the entropy of assignment distribution, namely encouraging the network to make an assignment as certain as possible. We use a KL divergence loss between predicted assignment distribution (s_{i1}, s_{i2}) and the pseudo ground truth (q_{i1}, q_{i2}) to gradually reduce the entropy as follows:

$$L_{\text{assign}} = \sum_{i=1}^{N} \sum_{j=1}^{2} s_{ij} \text{log} \frac{s_{ij}}{q_{ij}} \ .$$

Clustering learning and reconstruction learning are conducted simultaneously. Reconstruction learning of x_i is needed since it implicitly makes middle representation embeddings be assigned based on input data. We use a mean squared error (MSE) loss during the reconstruction learning as follows:

$$L_{\text{recon}} = \sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2 \ .$$

Finally, the optimization objective is $L = \alpha L_{\text{assign}} + \beta L_{\text{recon}}$. The larger α is, the more obvious the clustering effect is. It is reasonable if α is much bigger than β. We test different α and β and use the best one.

2.3.3 Region Extraction for Focusing on Learning Human Motions

In our experience, neural networks will neglect motion information if entire images are used as training data. Since the ratio of the human region to the entire image is too low, the neural networks cannot sufficiently learn motion information. Thus, we decided to extract the human region from an entire image so that the neural network can focus on learning human motion, not the background. In a previous study [65], the researchers proposed an approach that does not extract a region of interest. A possible reason why they did not use region extraction is that most of the details of the background are fuzzy, or even disappear in thermal and depth images, and the details of the background did not adversely affect the learning of motion information. In the case of using RGB images, we need to extract the region of human motion to increase the ratio of the motion region to the entire image, so that the neural network can focus on learning human motion.

To extract the region of human motion, we adopted the AlphaPose estimator [23] to estimate key points of a human. As shown in Fig. 2.2, a minimal bounding rectangle that includes all estimated keypoint positions is obtained, and the human region is cut out using an appropriate rectangle that is a little larger than the minimal bounding rectangle according to those positions. Then we align motion regions based on the same point, namely, the left shoulder point, and place them on a square with a black background. It is necessary to align motion regions since quivering motions give a negative effect on the performance of neural networks for fall detection.

2.4 Experiment

We conducted experiments using a PC having a 4.2 GHz i7-7700K CPU, 16 GB RAM, and a GTX 1070 GPU. As shown in Table 3.1, we compared supervised learning networks and weakly supervised learning networks, using the same architecture with different numbers of layers, for training and evaluating on different datasets, original images, and cropped images.

TPR denotes the true positive rate, which is a measure of how many falling samples were classified correctly. TNR denotes the true negative rate, which is a

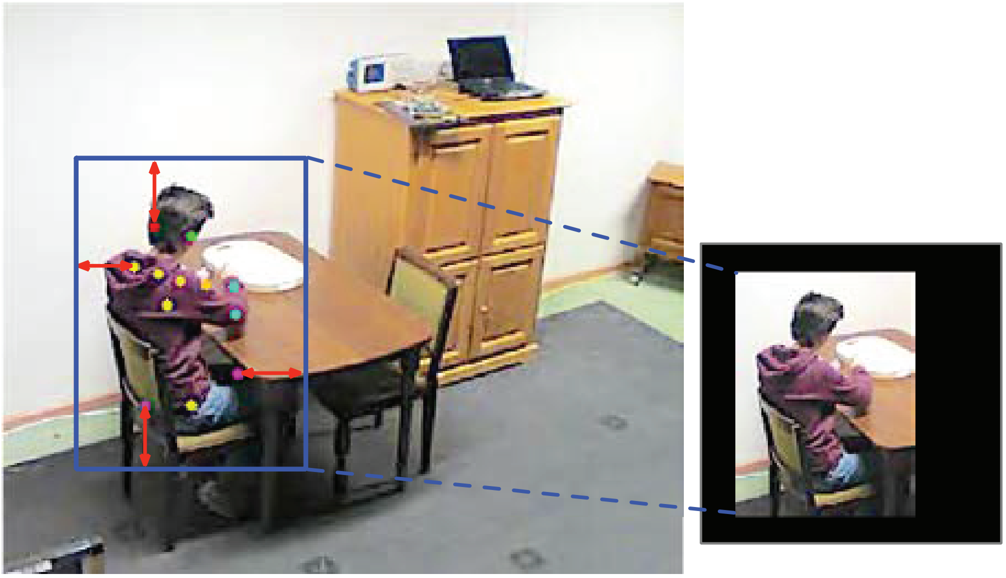
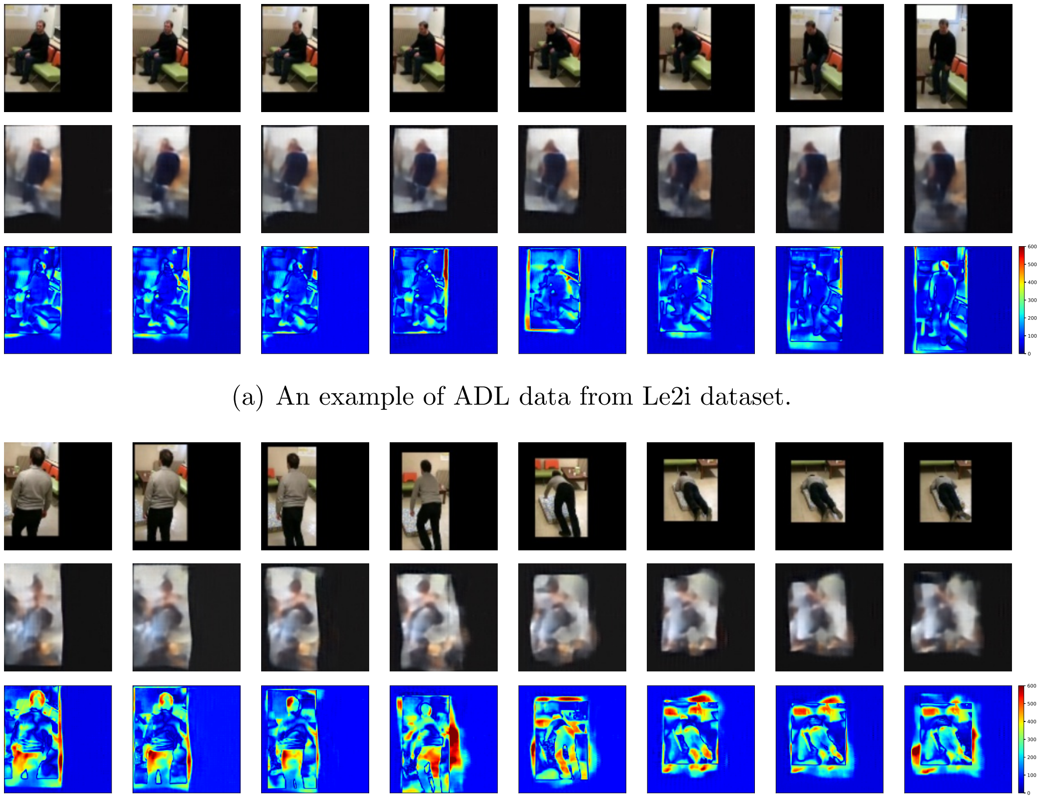


Figure 2.2: A example of human regions are extracted from an entire image by using a pose estimator.

measure of how many ADL samples are classified correctly. ACC denotes average accuracy. F1 denotes the F1-score, which is the harmonic average of the precision and recall. MCC denotes the Matthews correlation, coefficient which is a balanced measure of the quality of binary classifications. The closer an MCC value is to a positive one, the better the prediction is.

2.4.1 Dataset

We used the High Quality Simulated Fall Dataset (HQFD) [4] and the Le2i Fall Dataset (Le2i) [10]. ADL data was separated into several segments. Each segment included 16 frames which were uniformly sampled from 64 frames. The region of human motion was extracted from an entire image by using the AlphaPose estimator and was resized to 96 by 96 pixels. Regarding fall action data for evaluation, we manually trimmed each video to exactly include one fall action at first and then applied the same preprocessing operations as those on the ADL data. The data structure was in the form of (3, 16, 96, 96), which denotes 3 channels (RGB), 16 frames, a height of 96, and a width of 96. As training sets for weakly supervised learning, all samples were ADL data. As evaluation sets for weakly



(a) An example of ADL data from Le2i dataset.

(b) An example of fall action data from Le2i dataset.

Figure 2.3: Examples including input images (top), reconstructed images (middle), and heatmaps of reconstruction errors (bottom). For better showing, 16 frames are resampled to 8 frames, and best viewed in color.

supervised learning, besides all fall samples, there was also a moderate amount of ADL samples. Some examples of ADL data and fall action data are shown in Fig. 2.3.

HQFD. The HQFD dataset contains 275 fall videos and 85 ADL videos which range from 50 s to 35.5 min in duration. They were captured by RGB cameras from 5 different viewpoints. After preprocessing, there were 12266 ADL samples and 282 fall samples. An evaluation set for weakly supervised learning consisted of 282 fall samples and 300 ADL samples.

Le2i. The Le2i dataset contains 192 fall videos and 57 ADL videos which were captured by a single RGB camera and range from 10 s to 45 s in duration. After preprocessing, there were 834 ADL samples and 130 fall samples. Some videos after preprocessing with too few frames were abandoned. An evaluation set for weakly supervised learning consisted of 130 fall samples and 200 ADL samples.

2.4.2 Implementation

The encoder and decoder of the 3D-convolutional VAE respectively consisted of 8 residual blocks named ResVAE-18 or 16 residual blocks named ResVAE-34. Each residual block comprised two batch normalization layers, two activation layers, and two convolution layers (or transposed convolution layers). The networks were optimized using an Adam optimizer with a learning rate of 0.0001. The batch size was 32 in ResVAE-18 and 24 in ResVAE-34.

The encoder of the full-connected VAE (i.e. FCVAE) consisted of two full-connected layers with ReLU activation layers and another two full-connected layers with following dimensions: Input data (dim=512) $\rightarrow$ FC(dim=2048) $\rightarrow$ FC(dim=2048) $\rightarrow$ FC-μ(dim=512) & FC-σ^2(dim=512). The decoder consisted of two full-connected layers with ReLU activation layers and another full-connected layer with following dimensions: $z = \mu + \sigma \times \epsilon \sim N(0, I)$ (dim=512) $\rightarrow$ FC(dim=2048) $\rightarrow$ FC(dim=2048) $\rightarrow$ FC(dim=512). Cluster centers were also regarded as parameters and were jointly optimized with the FCVAE using an Adam optimizer with a learning rate of 0.0001 and a batch size of 64.

We chose an appropriate rectangle that was extended by 20 pixels in all directions from the minimal bounding rectangle to extract human regions. For the training of ResVAEs, networks were trained for 500 epochs. For the training of FCVAEs, networks were trained for 100 epochs. For training of standard ResNets, training was stopped if the value of the loss function was consecutively less than 0.00001 for ten epochs.

For evaluation using a threshold of reconstruction errors, the mean and variance of reconstructed errors of ADL samples in the evaluation dataset are calculated, and reconstruction errors of all samples are normalized using the mean and variance of training samples. The threshold was determined so that 85% of the ADL samples

Table 2.1: Experimental results of different networks by using evaluation of reconstruction error thresholds.

Method	Training Set (#Fall / #ADL)	Evaluation Set (#Fall / #ADL)	TPR (%)	TNR (%)	ACC (%)	F1 (%)	MCC
ResVAE-18 (proposed) (Weakly supervised, Cropped, Clustering)	HQFD (0/12266)	HQFD (282/300)	90	87.9	88.9	88.7	**0.778**
	HQFD (0/12266)	Le2i (130/200)	97.5	96	96.6	**95.5**	**0.928**
ResVAE-18 (proposed) (Weakly supervised, Cropped)	HQFD (0/12266)	HQFD (282/300)	94	83.7	88.7	**88.9**	**0.778**
	HQFD (0/12266)	Le2i (130/200)	92.3	84	87.3	85.1	0.748
ResVAE-34 (Weakly supervised, Cropped)	HQFD (0/12266)	HQFD (282/300)	80.5	83.3	82	81.2	0.639
	HQFD (0/12266)	Le2i (130/200)	92.3	84.5	87.6	85.4	0.753
ResVAE-18 (Weakly supervised, Non-cropped)	HQFD (0/12266)	HQFD (282/300)	13.5	83.7	49.7	20.6	-0.04
	HQFD (0/12266)	Le2i (130/200)	10.8	81.5	53.6	15.5	-0.105
ResVAE-34 (Weakly supervised, Non-cropped)	HQFD (0/12266)	HQFD (282/300)	58.9	83.7	71.6	66.8	0.44
	HQFD (0/12266)	Le2i (130/200)	10	84	54.8	14.9	-0.085
ResNet-18 (Supervised, Cropped)	HQFD (225/9812)	HQFD (57/2454)	5.3	100	97.8	10	0.227
	HQFD (225/240)	HQFD (57/60)	75.4	96.7	86.3	84.3	0.741
	HQFD (282/12266)	Le2i (130/200)	0.8	100	60.9	1.5	0.68
	HQFD (282/300)	Le2i (130/200)	93.8	79.5	85.2	83.3	0.717
ResNet-34 (Supervised, Cropped)	HQFD (225/9812)	HQFD (57/2454)	17.5	100	98.1	29.9	0.415
	HQFD (225/240)	HQFD (57/60)	47.5	98.3	73.5	63.5	0.535
	HQFD (282/12266)	Le2i (130/200)	2.3	100	61.5	4.5	0.119
	HQFD (282/300)	Le2i (130/200)	70.8	79.5	76.1	70	0.501

are always classified as normal samples. An unknown sample is classified as falling if its normalized reconstructed error is larger than a threshold.

For evaluation using clustering learning, we used unsupervised classification accuracy:

$$\mathrm{ACC} = \max_{m} \frac{1}{n} \sum_{i=1}^{n} 1\{l_i = m(c_i)\} \ ,$$

where l_i denotes the ground-truth label, c_i denotes the cluster assignment, and m denotes possible bijection functions between clusters and labels. Since there are only two classes, fall action data, and ADL data, there are two bijection functions.

2.4.3 Results and Analysis for Learning of ResVAE

Reconstructed data learned by the ResVAE and difference heatmaps between input data and reconstructed data were shown in Fig. 2.3. In heatmap images of ADL data shown in Fig. 2.3(a), it was seen that pixels of the subject and block edges are reconstructed well since motion changes smoothly, and blobs of reconstruction errors were rare. However, in heatmap images of Fig. 2.3(b), at the falling moment, many blobs of large reconstruction errors were produced, since the pose of falling down was a rare case in the training data, and ResVAE cannot reconstruct it well.

As shown in Table 2.1, in the experiments using imbalanced data, supervised

ResNets performed badly, which showed that imbalanced data was fatal for supervised learning. To maintain the balance, a lot of ADL data must be abandoned. The accuracy of supervised learning methods with balanced data sharply increased compared with those with imbalanced data. Since abandoning ADL data was unnecessary for weakly supervised learning methods, ResVAEs with cropped data had better performance than standard ResNets with cropped data.

For training and evaluation on different datasets with different persons and situations, the performance of ResVAEs with cropped data was still better than that of standard ResNets with cropped data. The ResVAEs showed good generalization ability since the proposed method adopts a kind of weakly supervised learning architecture.

Besides, the result of ResVAE-34 was worse than that of ResVAE-18 when training and evaluating on the same dataset. A possible reason was that ResVAE-34 may need more training for obtaining stable and better performance. Another possible reason was that overfitting happens in ResVAE-34 due to deeper layers.

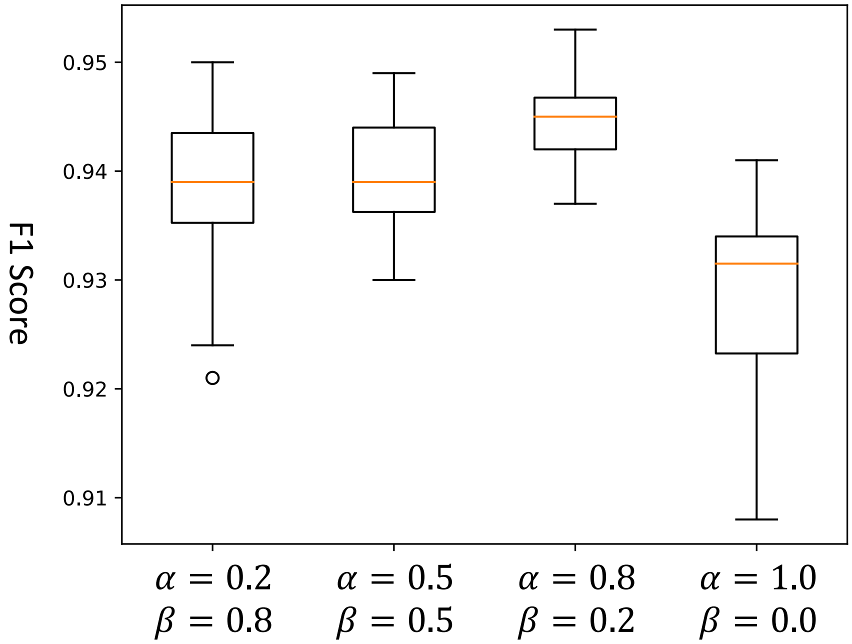


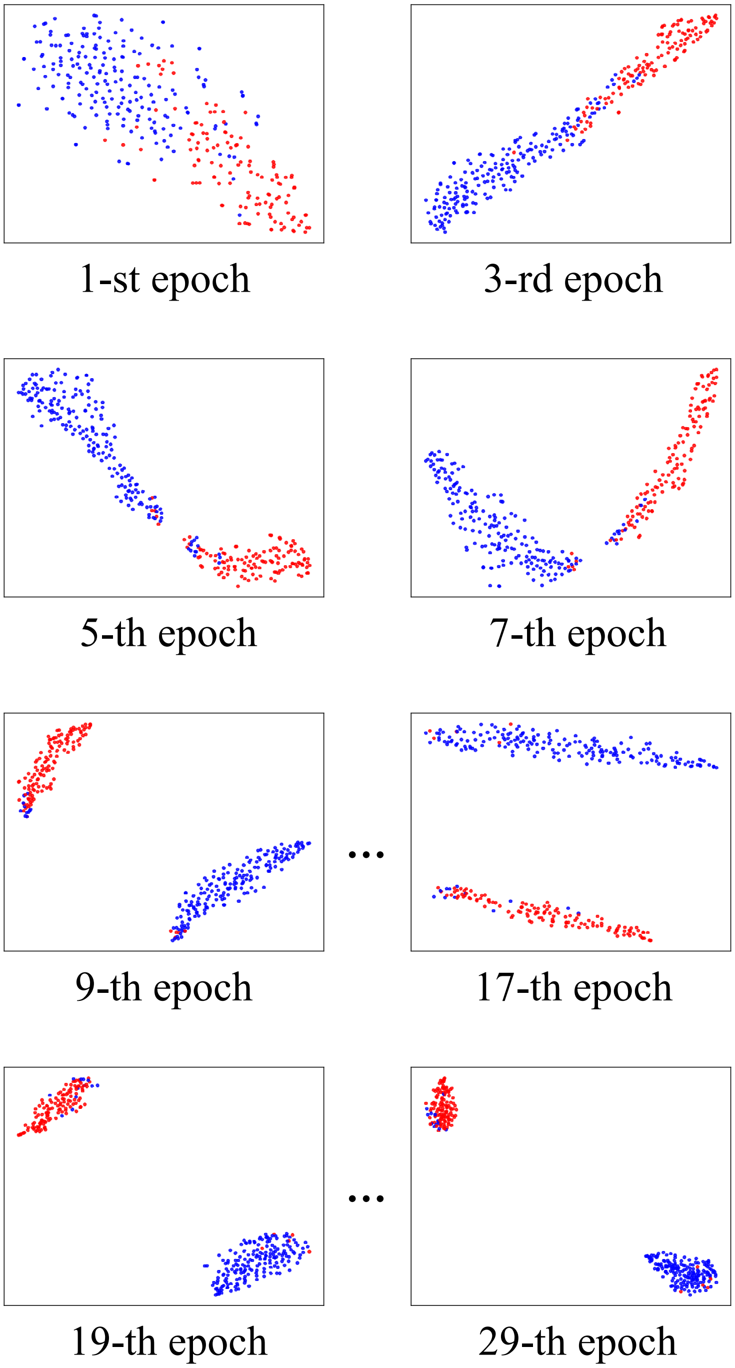
Figure 2.4: F1 scores by training the network with a loss function using different α and β.

2.4.4 Results and Analysis for Learning of FCVAE

F1 scores by training the network with a loss function using different α and β are shown in Fig. 2.4. The network obtained a lowest F1 score when a reconstruction loss was excluded, namely β was set to zero, which indicates reconstruction is necessary to keep a sample being gathered to a correct cluster.

Visualization results of representation embeddings learned by FCVAE during clustering learning are shown in Fig. 2.5. At the beginning (e.g. the first epoch), the boundary of ADL data and fall action data was unclear, and a part of sample points were mixed up. With training going on, similar representations were continuously gathered, and discrepant representations were continuously made distant. Finally, sample points formed two clear clusters, and classification results were obtained by using a metric of unsupervised classification accuracy.

As shown in Table 2.1, the proposed network with clustering learning that was trained on the HQFD dataset and evaluated on the Le2i dataset obtained a large improvement on the aspect of generalization ability It was shown that measurements learned by deep neural networks was superior to handcrafted thresholds of reconstrction errors.



1-st epoch

3-rd epoch

5-th epoch

7-th epoch

9-th epoch

17-th epoch

19-th epoch

29-th epoch

Figure 2.5: Two-dimensional visualization results of 500-dimensional latent representations during unsupervised clustering learning period. Best viewed in color.

2.5 Discussion

2.5.1 Weakly Supervised Learning vs. Supervised Learning

First, in the case of weakly supervised learning, abundant ADL data is an advantage because networks can be optimized better by using abundant data, and the performance can be improved. The accuracies of standard ResNet decrease when the networks detect fall actions with different persons and different situations, namely, evaluating networks on a different dataset. In contrast, our proposed method, which is based on a weakly supervised learning architecture, has advantages in dealing with such situations. After training, the encoder of the VAE encoded ADL data close to each other, and encoded the most of fall action data far away from ADL data, though fall action data were not included in the training set. When the network was evaluated on a different dataset (i.e., Le2i), we can find the same tendency that latent variables of ADL data and fall action data form clusters, which shows an advantage of using weakly supervised learning, i.e. good generalization ability.

Moreover, when standard ResNets with deeper layers were used, the performance decreased instead, which shows that the performance of supervised learning was not stable since less training data leads to overfitting. In contrast, our proposed method showed robustness and obtained a more stable performance. Thus, it is more appropriate to use a weakly supervised learning architecture for fall detection.

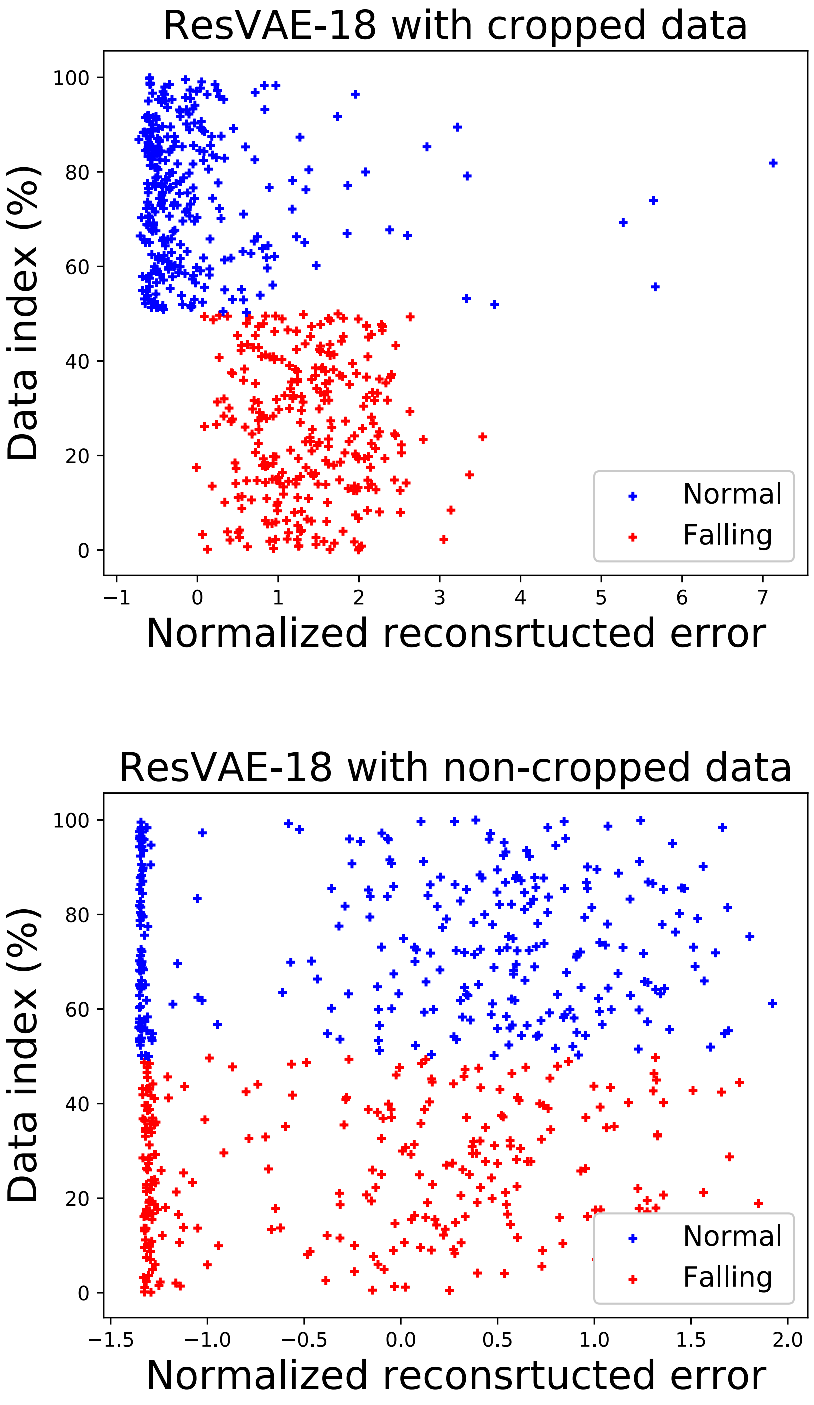


Figure 2.6: Normalized reconstructed errors that are evaluated on the HQFD dataset of ResVAE-18 with learning of cropped data and non-cropped data. Best viewed in color.

2.5.2 Cropped Image vs. Entire Image

The MCC value of ResVAE-18 with non-cropped data showed that its performance was no better than random prediction. Although adding up to 34 layers improved the performance, it was still much worse than that of ResVAEs with cropped data. This demonstrates that too little motion information in input images can adversely affect the performance of the networks, and therefore, the networks cannot sufficiently learn human motions. As shown in Fig. 2.6, the distribution of normalized reconstruction errors of ResVAE-18 with cropped data was more well-organized than that with non-cropped data. Thus, it is necessary to use a region extracting technique and align motions, which is more efficient than simply adding layers.

In heatmaps of reconstruction error of fall action data shown in Fig. 2.3, some areas with very high reconstruction errors (colored in red) appeared in edges of extracted human regions. We infer that since normal actions were generally not intense, the network learned reconstruction from the regular context of videos. However, fall actions were turbulent, which made it difficult for the network to reconstruct them.

2.6 Conclusion and Future Work

2.6.1 Conclusion

We proposed a method for detecting fall actions by using a 3D-convolutional VAE to learn a distribution of ADL data and use a fully-connected VAE of clustering learning to detect fall actions. We also proposed a technique for extracting a region of human motion from an entire image and aligning motions based on the same joint point so that the network can focus on learning human motions.

The results of experiments showed that our method, which is a type of weakly supervised learning, achieved a competitive level of accuracy and better generalization ability compared with supervised learning with well-labeled data. We demonstrated that the technique of extracting human regions and aligning motion had enhanced the accuracy of fall detection. We also discussed the advantages of using weakly supervised learning and region extraction. Weakly supervised learning methods can overcome imbalance between ADL data and fall action data, and obtain good generalization ability when the network is evaluated on different datasets. Using region extraction and motion aligning can make the networks focus on learning human motions.

2.6.2 Future Work

Our method has a limitation that the performance decreases when skeleton information is not extracted completely and accurately. In the future, more complex networks will have to be designed to extract features from non-preprocessed images so that the method will be less sensitive to extracted skeleton information.

Chapter 3

An Asymmetrical-Structure Auto-encoder for Unsupervised Representation Learning of Skeleton Sequences

3.1 Introduction

Human action recognition is a crucial topic in computer vision since it plays a fundamental role in a wide range of applications, such as video surveillance, human–computer interaction, video understanding for retrieval, etc. With the rapid development of convolutional neural networks (CNN), CNN-based methods have achieved significant success in action classification from videos [9, 81, 75, 88, 32]. Visual cues (e.g., RGB images and depth images) provide discriminative features for action recognition, whereas learned features also include bias from viewpoints, the appearance of actors, backgrounds, and many other factors that can adversely affect the recognition performance. In some studies, skeletons are extracted from images, and then recurrent neural networks (RNN) are used to model skeleton movements in the temporal dimension for extracting more robust features of actions [39, 16, 50, 102, 74]. Not only can 3D joint coordinates be acquired from low-cost human skeleton capture devices (e.g., Kinect), but also there have been extensive studies of 3D human pose estimation algorithms [44, 68, 49] in recent years. Hence, here we explore action representation learning based on 3D skeleton data.

Existing supervised learning methods have achieved excellent classification performance owing to strong supervision using a large amount of well-labeled data. Manually labeled data is extremely precious since it is laborious to annotate continuously generated data with tags. Neural networks easily suffer from overfitting if well-labeled data is lacked in training. Moreover, it is subjective and difficult to decide how exact the annotations of action videos should be. Therefore, it is worth studying to learn representations of motion dynamics from data itself. Some studies proposed using an encoder to learn representations from input skeleton sequences, and then using a decoder to reconstruct coordinates of skeleton sequences from given learned representations [57, 78, 79]. Classifying actions by given learned representations has been shown to be a valid approach, however, which easily allows networks to be made naive to reduce the dimensionality of input data so that similar skeleton movements are not clustered together.

Therefore, we propose a novel unsupervised action representation learning method that exploits a structure-asymmetrical auto-encoder to learn action representations from unlabeled data, and the learned representations are utilized for other tasks such as action recognition. In detail, a CNN-based encoder is trained to extract spatiotemporal features from pixelated images which are made from skeleton trajectories, and then extracted features are fed into an RNN-based decoder to generate salient skeleton motion cues. Not only does the CNN-based encoder naturally encode correlations of adjacent joints, but also long-term motion dependencies are implicitly encoded in the representation due to the supervision of salient skeleton motion cues in the RNN-based decoder. Thus, those learned representations are made separable in low-dimensional feature space, and are discriminated for action classification.

We also propose a type of feature to effectively represent motion dynamics, namely, salient skeleton motion cues, in which the 3D motion of each joint is normalized (i.e., direction is retained and magnitude is removed), and only joints and frames with salient motion are retained. Since the capability of the network is to focus on predicting salient information that mainly affects the recognition results, the proposed network can capture more essential motion dependencies.

The contributions of our study are as follows:

i) We propose a novel unsupervised network, a structure-asymmetrical autoencoder, to effectively learn action representations. A CNN-based encoder is used to explicitly extract local motion features, and an RNN-based decoder is used to implicitly encode long-term motion features. We show that our network gathers similar movements around the same cluster, and gathers different movements around distinct clusters in a low-dimensional feature space by using the unsupervised training of information transformation from images (spatial) to sequences (temporal).

ii) We propose a type of cue, salient skeleton motion cues, to effectively represent motion dynamics and serve the function of supervision signals in the proposed network. We show that our network captures more essential motion dependencies, since the network capability is made to focus on generating salient information by using the proposed supervision.

iii) We conducted experiments on the NTU RGB+D and NW-UCLA datasets. The experimental results showed the effectiveness of the proposed method for unsupervised representation learning. When training under unsupervised learning settings, Our method outperformed most previous methods. When the proposed network was fine-tuned with partial labeled data, our results still outperformed some fully supervised methods.

3.2 Related Work

3.2.1 Supervised Action Recognition

Except for CNN-based deep learning methods, RNN-based deep learning methods also perform well for classifying sequential skeleton data. In order to better capture long-term contextual information of skeleton sequences, physical structure of human skeletons were considered. A number of studies [102, 34, 51, 89, 15, 40] mentioned that the spatial structure of the human skeleton is an important clue for action recognition. Some authors attempted to convert skeleton sequences to images and used CNNs to classify actions. Du et al. [15] and Li et al. [40] proposed regarding 3D coordinates values on the x, y and z axes as three channels in an image with frame indices or joint indices corresponding to columns or rows, respectively. Ke et al. [34] proposed leveraging relative positions between four reference joints and other joints to obtain images instead of using absolute coordinate values. In addition, a view-invariant transformation Lee et al. [39] and Su et al. [79] was implemented on skeleton coordinates to improve the performance of detecting the same action captured by different viewpoints.

Recently, point cloud-based 3D action recognition methods have been developed and have shown good performance. Point cloud-based 3D methods are roughly divided into two categories, those using voxelated points [91] and those using raw points [21, 20, 45]. Wang et al. [91] proposed to convert point motions to a voxel set, and a PointNet++ was used to extract spatiotemporal feature from the voxel set. Fan et al. [20] proposed to use 4D convolution to features in point cloud videos, and a Transformer was used to learn the relationship of those spatiotemporal features. Fan et al. [21] and Li et al. [45] proposed methods that hierarchically present point clouds, where time (1D) and space (3D) of point cloud videos were decomposed as sequences of 3D point clouds.

3.2.2 Unsupervised Action Representation Learning

Although supervised learning methods show continuous improvement on recognition performance, unsupervised learning methods deserve to be studied since they do not rely on well-labeled training data. Some studies focus on unsupervised visual representation learning using RGB video or video with additional information, such as depth or optical flow.

Those methods and ours make a generative pretext task to guide the action representation learning, but the difference is asymmetrical architecture of the proposed networks, a CNN-based encoder followed with a RNN-based decoder. CNNs do not explicitly aggregate sequential information from extracted features, and we, therefore, take sequential data as supervision signals to enhance CNNs, and to guide the action representation learning. Those proposals were valid only for visual features in short temporal intervals; however, long-term motion dependencies were lost. Luo et al. [54] proposed an improved method in which a convolutional LSTM predicts optical flow information of future frames from input RGB or depth videos. Li et al. [41] showed that adding a camera-view discriminator in networks can improve performance since it helps networks to learn view-invariant representations. In addition, Fan et al. [19] proposed a vision-based mechanism that learns motion representation by eliminating content from videos, which makes representations of the same actions with different appearances can be gathered nearly in latent space.

Nowadays, coordinate positions of joints can be obtained efficiently thanks to the development of deep learning and convolutional neural networks [44, 68, 49]. Our method therefore focuses on learning action representations by using 3D skeleton sequences. Recently, some studies of unsupervised action recognition have focused on making networks reconstruct data including motion information. Zheng et al. [100] proposed an adversarial auto-encoder to reconstruct skeleton sequences, where an encoder learns to compress skeleton sequences to latent representations, and a decoder attempts to generate skeleton sequences from the representations, and a discriminator learns to distinguish the original inputs from the reconstructed skeleton sequences. Su et al. [79] proposed an RNN-based auto-encoder with novel training strategies to reconstruct skeleton sequences.

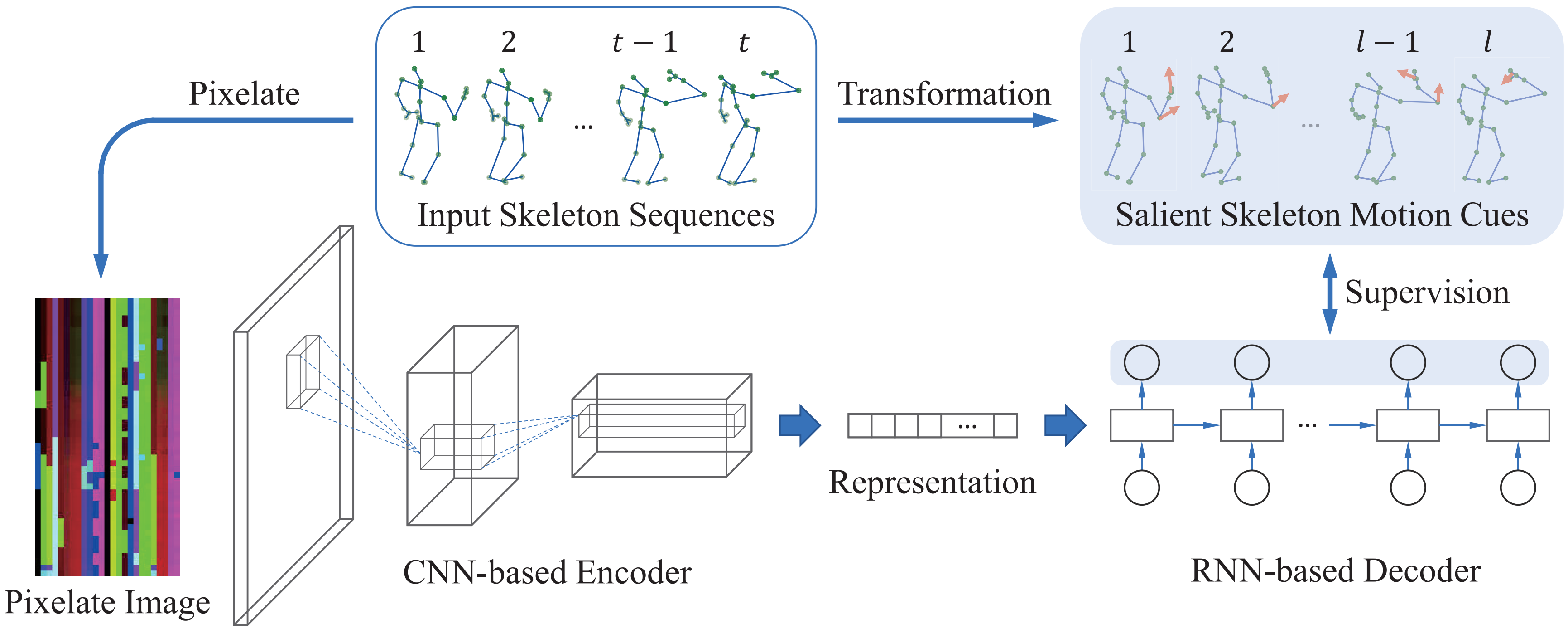


Figure 3.1: Overview of the proposed method. Both input data (i.e. pixelated images) and supervision signals (i.e. salient skeleton motion cues) are created from skeleton sequences. Pixelated images are created by taking coordinates as pixels in multiple channels, taking joint indices as a horizontal axis, and taking frame indices as a vertical axis. Salient skeleton motion cues are created by extracting salient parts of skeleton sequences by using a clustering method. The network predicts salient skeleton motion cues by given representations, in which a CNN-based encoder is used to learn representations from pixelated images, and a RNN-based decoder is used to generate salient skeleton motion cues by given the representations.

3.3 Method

The goal of our unsupervised learning framework is to learn action representations without human-labeled annotations, which allows the capture of both correlations of adjacent joints and long-term motion dependencies, and is sufficiently discriminative for classification. To achieve this, we propose a structure-asymmetrical auto-encoder, as shown in Fig. 3.1. In the network, a CNN-based encoder learns to extract features from skeleton trajectories which are treated as 3-channel images. After features are extracted, they are fed into an RNN-based decoder to generate salient skeleton motion cues. Therefore, not only does the CNN-based encoder naturally encode correlations of adjacent joints (i.e., spatial features) into the representations, but it also implicitly encodes long-term motion dependencies (i.e., temporal features) into the representations due to the supervision of salient skeleton motion cues. The representations will be used for action recognition.

3.3.1 Network Architecture

We propose a structure-asymmetrical auto-encoder to learn action representations from unlabeled data. The encoder, denoted as f_θ, is a CNN-based neural network with seven weight layers, as shown in Fig. 3.2. It is composed of three VGG-style blocks [76], two max pooling layers, an adaptive average pooling layer and a fully-connected layer. Each VGG-style block consists of two convolutional layers followed by a batch normalization layer and a ReLU activation layer. Each 3D skeleton sequence S with T frames and J joints and 3 axes is formulated as

$$S = \{s_1, s_2, ..., s_T\},\ s_t \in \mathbb{R}^{J\times 3}$$

before being input to the encoder needs to be transformed to a 3-channel image X where 3 axes are taken as 3 channels, and T frame indices and J joint indices correspond to H rows and W columns. Since CNNs have a strong capability for modeling correlations of neighboring pixels in images, we apply a pixelated transformation to skeleton sequences, and use a CNN-based neural network to

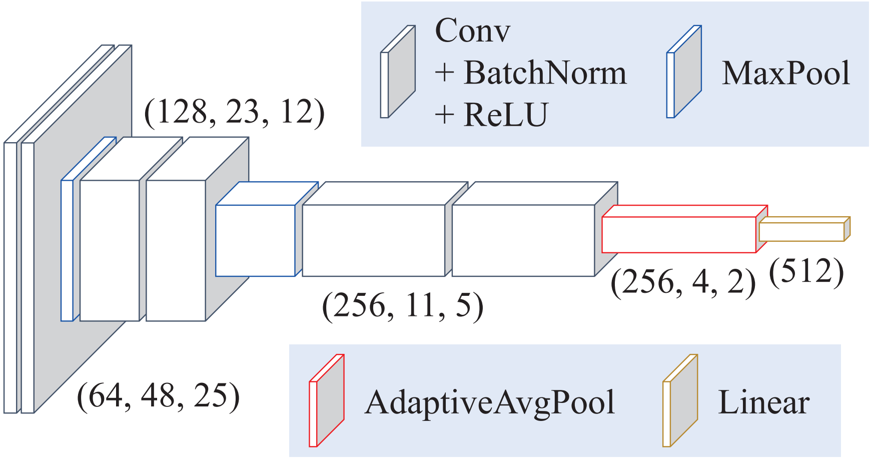


Figure 3.2: Detailed configuration of the CNN-based encoder. The size of the middle outputs is denoted as $C \times H \times W$, where C is the number of channels, H is the height, and W is the width. Finally, the encoder outputs a vector with a size of (512). Best viewed in color.

model the correlations of adjacent joints. Features extracted by the encoder f_θ, denoted as $f_\theta(X)$, will be taken as initial hidden states h_0, and are fed into the decoder.

The decoder, denoted as f_ϕ, is an RNN-based neural network with two weight layers, as shown in Fig. 3.3. It is composed of a gated recurrent unit, a fully connected layer and a Tanh activation layer. Since RNNs can naturally deal with the order of elements in a long sequence, we enable the decoder generate salient skeleton motion cues from a given initial hidden state h_0 so that long-term motion dependencies should be implicitly encoded in the given initial hidden states. The inputs of the decoder, denoted as x_t, are initialized to zeros. The decoder outputs are denoted by $\hat{O} = f_\phi(h_0)$, where each output has a variable length L that is

$$\hat{O} = \{\hat{o}_1, \hat{o}_2, ..., \hat{o}_L\}, \ \ \hat{o}_l \in \mathbb{R}^{J \times 3}.$$

Length L is always less than length T. As the training loss, we adopt mean square errors (MSE) between outputs $\hat{O}$ and the ground truth O, that is,

$$\mathcal{L} = \frac{1}{L} \sum_{l=1}^{L} \|o_l - \hat{o}_l\|^2.$$

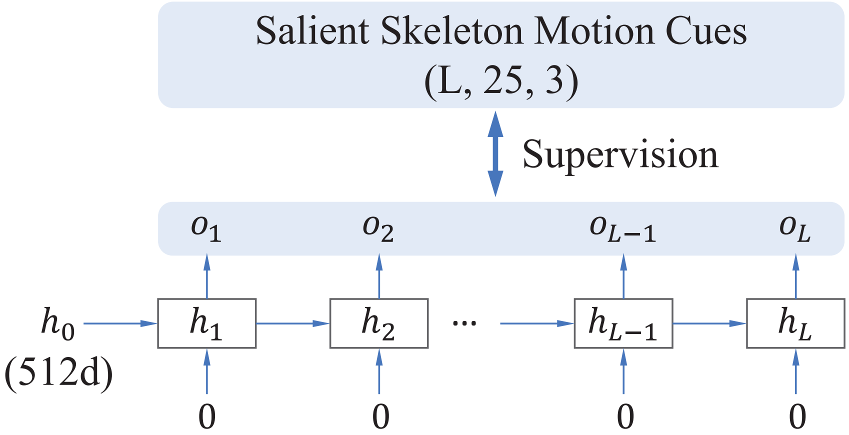


Figure 3.3: Detailed configuration of the RNN-based decoder, where h_t are hidden states, x_t are initial inputs, and x_t are outputs. The size of the initial hidden states is denoted as (512). The outputs and the supervision signals have a size with a variable length L, 25 joints and coordinates on 3 axes.

3.3.2 Salient Skeleton Motion Cues

We propose to take salient skeleton motion cues as supervision signals during training so that the network is made to capture essential motion dependencies. First, joints and frames with salient motion need to be identified. Most of the frames are redundant and are not clear enough to represent motion dynamics, which would adversely affect the representation learning if they played a role in supervision. Thus, frames without salient motion need to be removed.

Firstly, the variance of coordinates of each joint among all frames are calculated, and then those variance values of 25 joints are clustered by k-Means where $n\_clusters$ is set to two. The set of joints with larger variance values is retained, and the rest is dropped. An example skeleton sequence of a person drinking water is shown in Fig. 3.4, where only four frames have salient motions. Furthermore, the movements of most joints are meaningless, as shown in Fig. 3.5. Coordinate values are set to zero if they are less than a threshold. In this study, the threshold is a mean of the center of large values and the center of small values. Similarly, a summation of variances of retained joints for each frame are calculated, and then variance values of T frames are clustered by k-Means too, and the set of frames

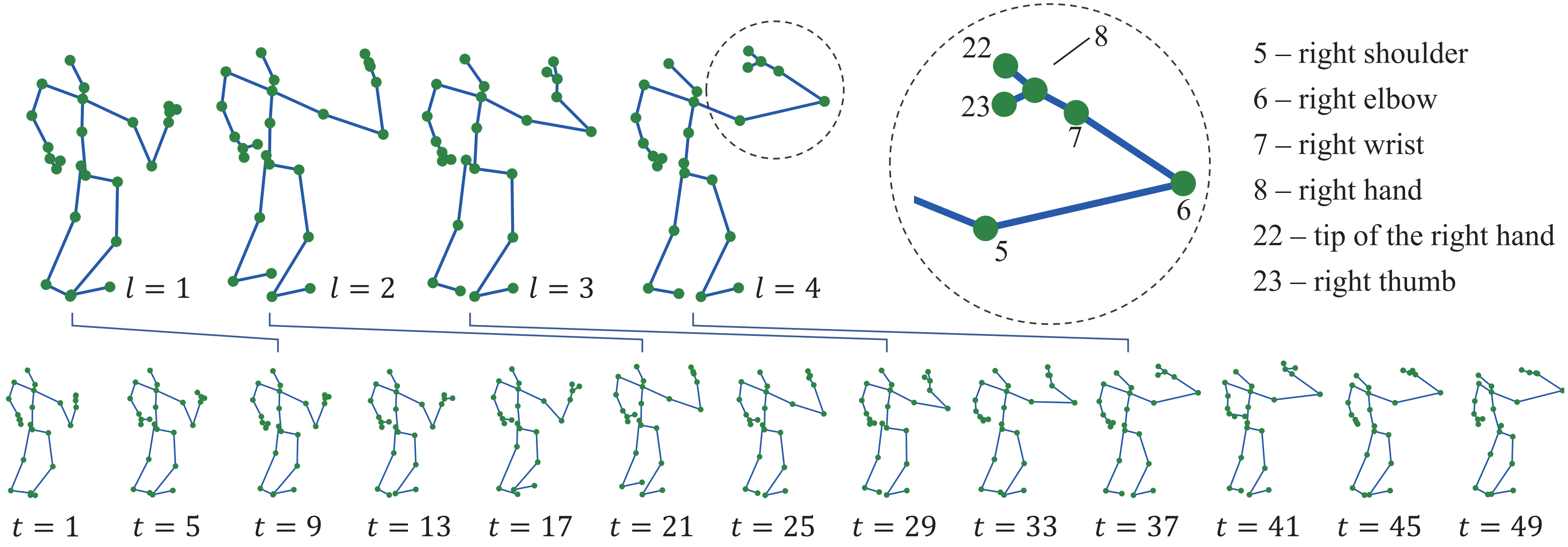


Figure 3.4: A skeleton sequence of a person drinking water. Four frames that have salient skeleton motions are highlighted. Partial frames are omitted for brevity, and best viewed in color.

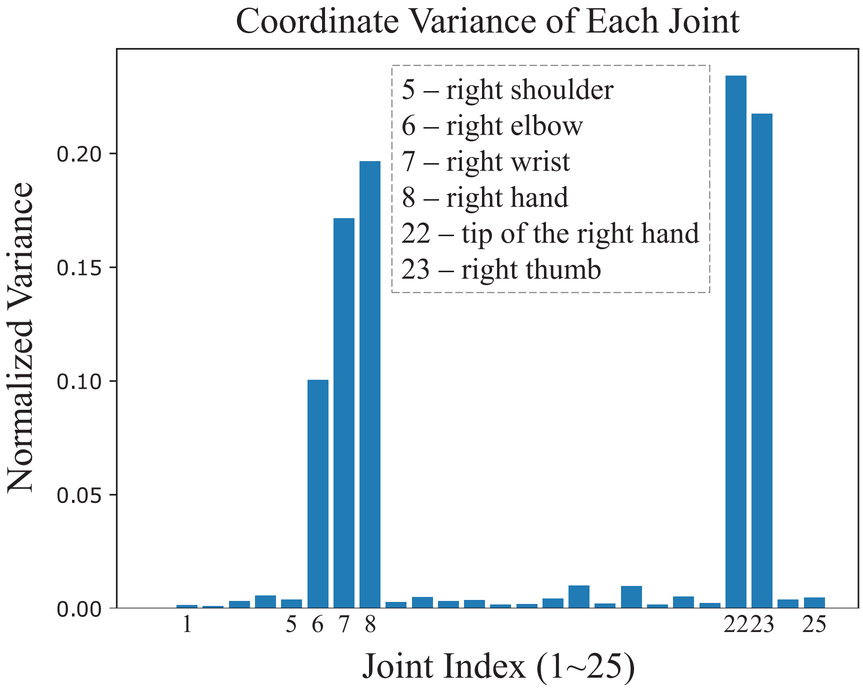


Figure 3.5: Coordinate variance of each joint for an example of drinking water. The sum of the coordinate variances of joints 6, 7, 8, 22 and 23 accounts for over 92% of the total.

with larger variance values is retained and have a length of L.

It is necessary to eliminate noise since some motions are too noisy to be meaningful for representation learning, and the capabilities of the neural network are wasted in predicting those motions. After that, motion features are obtained by

$$o_l = x_{l+1} - x_l,\ \ l \in \{1, 2, ..., L-1\}.$$

Then magnitude information in the motion features is eliminated by using

$$o_{l,j} \leftarrow \frac{o_{l,j}}{\|o_{l,j}\|^2},\ \text{if } o_{l,j} \neq 0;$$

that is, only direction information is retained. The magnitude information varies irregularly according to the viewpoint of the camera and the diversity of actors, which hinders accurate predictions and will adversely affect the performance. Therefore, it is also necessary to eliminate them.

There is another advantage of the proposed salient motion cues, namely, that the retained direction information is naturally rescaled within the range -1 to 1, which helps to make training stable. Since salient skeleton motion cues are filtered from primal coordinate positions, we expect that the proposed network can capture more essential motion dependencies and perform better than an auto-encoder in the reconstruction of primal coordinate positions [79].

3.4 Experimental Results

We conducted experiments on the NTU RGBD 60 and NW-UCLA datasets to evaluate our method. KNN classification was utilized to demonstrate that the learned representations are separable and discriminate for downstream tasks such as classification. We tested the proposed network with various configurations to show the advantages of our method. We also present a visualization result showing how the representations are distributed in low-dimensional feature space. We fine-tuned the proposed network with different percentages of labeled data to show that our method still performs well in a situation where human-labeled data was lacking. We also compared our method with prior state-of-the-art methods.

3.4.1 Dataset

NTU RGBD 60 [74]. This is a large-scale human action dataset that contains about 56000 samples for 60 action classes captured from 40 subjects and 3 viewpoints, such as clapping, drinking water, handshaking, etc. We only used sequential data in this dataset, in which each skeleton was 3-dimensional and had 25 joints. We used two evaluation protocols: Cross View and Cross Subject. In the Cross View protocol, samples belonging to cameras 2 and 3 were used for training, and samples belonging to camera 1 were used for testing. In other words, the training set included a front view and two side views of actions, whereas the test set included left and right 45-degree views of actions. In the Cross Subject protocol, 40 subjects were split into training and test groups so that each group included 20 subjects. We evaluated our method using both protocols.

Northwestern-UCLA (NW-UCLA) [86]. This dataset contains 1494 samples for 10 action classes where each skeleton has 20 joints captured from 10 subjects and 3 viewpoints. The training set consists of samples from viewpoints 1 and 2, and the test set consists of samples from viewpoints 3.

3.4.2 Implementation

Pre-processing. Before training the network, all skeleton coordinates were preprocessed with a view-invariant transformation, as described in [79] and [39]. Since CNNs cannot deal with sequential data with a variable length, skeleton sequences were down-sampled to have 48 frames. Finally, the input data was changed to images with a size of $3 \times T \times J$, where values on three axes were assigned to three channels and were permuted before T and J. T denotes 48 frames corresponding to different columns, and J denotes 20 or 25 joints corresponding to different rows. L had a range from 4 to 22.

Training. During unsupervised training, we used an Adam optimizer with a learning rate of 0.0003 and a batch size of 128. The network was trained for 1000 epochs. The encoder output vectors with 512 elements, which were taken as final representations and were used for action recognition.

KNN Evaluation. We applied a KNN classifier with $k = 1$ (i.e. 1-nearest neighbor) on the learned representations output by the encoder to evaluate the action recognition performance of the proposed method. All sequences in the training dataset were used to assign classes similarly to another study reported in the literature [79]. Specifically, we used related code in the scikit-learn package [69] to assign classes and make predictions. Note that KNN classifiers do not require to learn extra weights for classification.

Fine-tuning Evaluation. It is usual practice to pre-train the system on a large-scale dataset and then fine-tune it with a few items of labeled data. To make predictions, we used a linear classifier (i.e., a fully-connected layer) attached after the frozen encoder. Parameters of the linear classifier were initialized using a uniform distribution and were jointly upgraded with pre-trained parameters of the encoder during fine-tuning. The encoder with a classifier was trained for 155 epochs using an AdamW optimizer [53] with a learning rate of 0.0003, a weight decay of 0.004, a batch size of 128, and a cyclic learning rate scheduler [77] with restarts including a restart period of 5, a multiplier of 2 and a cosine policy. In total, the network was trained ten times to calculate the average values and variance values. For training data each time, 16, 32, 64, 128 and 256 samples were randomly picked up form 60 classes and were fixed for epochs, which accounted for about 2.5%, 5%, 10%, 20% and 40% of the total, respectively.

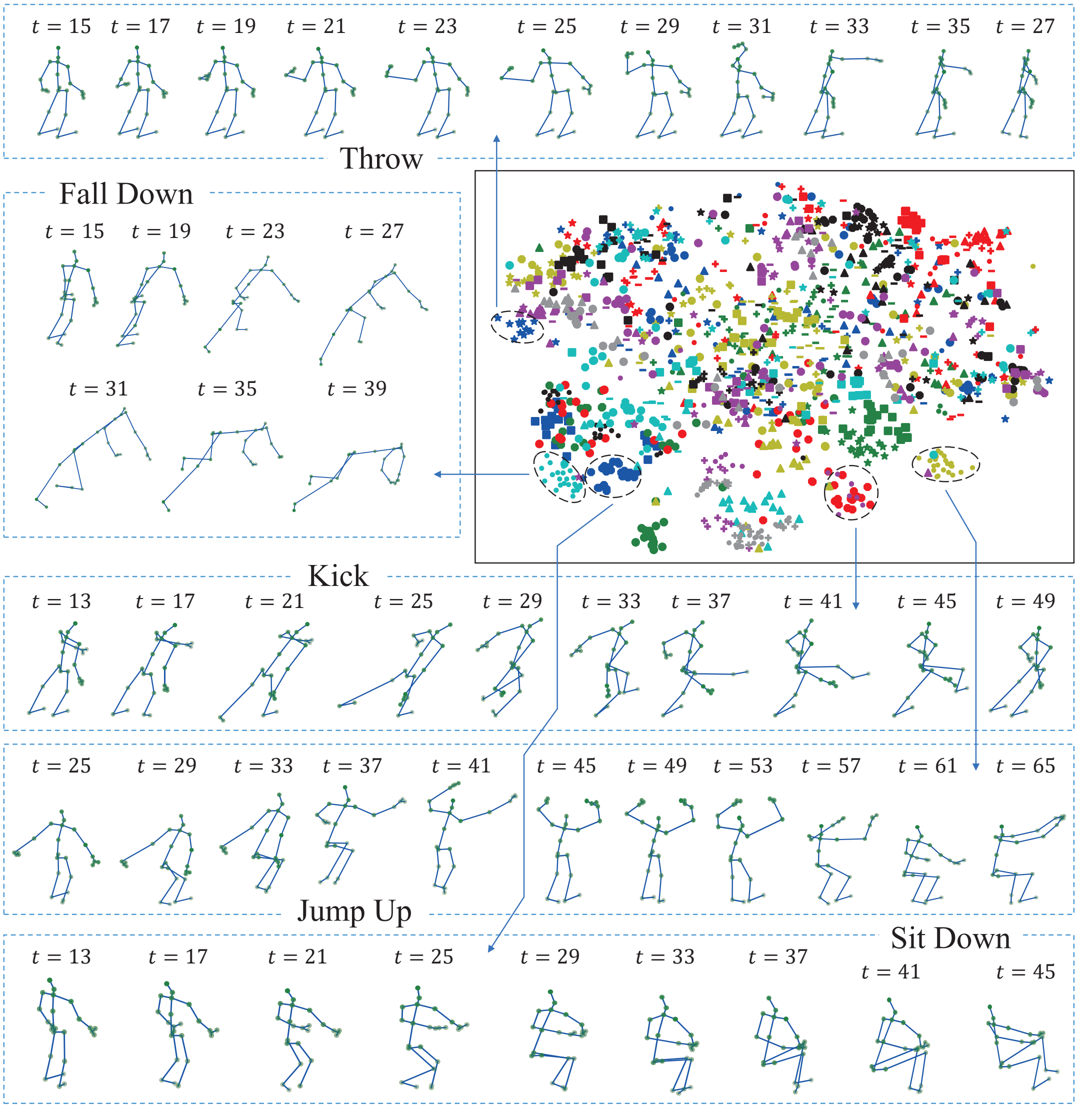


Figure 3.6: t-SNE visualization for the learned representations on NTU RGB+D dataset (60 classes) with the cross-view protocol. Different colors represent different classes, and best viewed in color.

3.4.3 Result

We first show a t-SNE visualization result of the learned representations in Fig. 3.6, where 512-dimensional vectors were embedded in 2D space. It is seen that some actions were clustered together in the feature space, such as throw (blue star), fall down (cyan dot), kick (red hexagon), jump up (yellow dot), sit down (blue hexagon), etc. Moreover, two similar actions, fall down (cyan dot) and sit down (blue hexagon), were clustered into distinct but near clusters, which shows

Amount of Training Labels	Baseline - Avg. ± Var.		Proposed - Avg. ± Var. (↑)	
	Cross View	Cross Subject	Cross View	Cross Subject
2.5% (16×60 classes)	47.69% ± 1.10	44.15% ± 0.28	56.03% ± 0.23 (+8.34)	51.45% ± 0.31 (+7.30)
5% (32×60 classes)	57.45% ± 0.59	52.61% ± 0.64	63.51% ± 0.22 (+6.06)	58.75% ± 0.10 (+6.14)
10% (64×60 classes)	66.16% ± 0.10	59.88% ± 0.35	70.10% ± 0.04 (+3.94)	63.70% ± 0.11 (+3.82)
20% (128×60 classes)	73.48% ± 0.05	66.41% ± 0.14	75.68% ± 0.08 (+2.20)	68.20% ± 0.09 (+1.79)
40% (256×60 classes)	79.85% ± 0.07	71.63% ± 0.10	80.55% ± 0.04 (+0.70)	72.14% ± 0.02 (+0.51)
100% (all labels*)	86.45% ± 0.02	76.98% ± 0.03	86.53% ± 0.03 (+0.08)	76.52% ± 0.02 (−0.46)

Table 3.1: Experimental results of training with partial labeled data. "Avg. ± Var. (↑)" means the average and variance of accuracies for ten iterations of training and improvements compared with baselines. *In total, there were 37113 training samples in the cross-view protocol, and 39649 samples in cross-subject protocol.

the effectiveness of the learned representations. However, in some cases at the top right corner, representations were mixed. It may be a possible reason that objects cannot be recognized since we did not use RGB images for training, and some actions based on recognizing objects cannot be represented well, such as putting on or taking off jackets, brushing or flicking hair, etc.

We also show results of the KNN classification in Table 3.2. On the first line, a symmetrical network using RNN-based encoder and decoder was trained to reconstruct skeleton coordinates, and it was taken as a baseline. When salient skeleton motion cues were taken as supervision signals, the performance was not significantly improved as shown on the second line. When using the proposed CNN-based encoder, the accuracies were improved as shown on the third line, and it was shown that the proposed structure-asymmetrical structure is effective to improve the performance. When using both proposals, our method obtained the best results as shown on the fourth line. CNNs do not explicitly aggregate

Method	CNN-based Encoder	Salient Skeleton Motion Cues	Cross View	Cross Subject
RNN-to-RNN			69.51%	48.17%
RNN-to-RNN		✓	67.91%	50.27%
CNN-to-RNN	✓		76.39%	52.75%
CNN-to-RNN	✓	✓	**77.50%**	**56.42%**

Table 3.2: Comparison of methods with different configurations using KNN classifiers. Network structures are denoted as ⟨encoder⟩-to-⟨decoder⟩.

Method	Modality	Unsupervised Learning Type	NTU RGBD 60		NW-UCLA
			Cross View	Cross Subject	
3s-CrossSCLR (LSTM) [43]	Skeleton	Contrastive Learning	62.8%	69.2%	-
3s-CrossSCLR (ST-GCN) [43]	Skeleton	Contrastive Learning	**77.8%**	83.4%	-
Skeleton Contrast [80]	Skeleton	Contrastive Learning	76.3%	**85.2%**	-
TS Colorization [97]	Point Cloud	Pretext Task	71.6%	79.9%	90.1%
TS+SS Colorization [97]	Point Cloud	Pretext Task	74.6%	82.6%	**91.1%**
TS+SS+PS Colorization [97]	Point Cloud	Pretext Task	**75.2%**	**83.1%**	-
LongT GAN [100]	Skeleton	Pretext Task	39.1%	52.1%	74.3%
P&C FW-AEC [79]	Skeleton	Pretext Task	50.7%	76.1%	84.9%
MS^{2}L [47]	Skeleton	Pretext Task	52.6%	-	76.8%
MCAE-MP [96]	Skeleton	Pretext Task	**74.7%**	65.6%	83.6%
Ours	Skeleton	Pretext Task	70.3%	**78.3%**	**87.4%**

Table 3.3: Comparison with state-of-the-art unsupervised methods on NTU RGBD 60 and NW-UCLA.

sequential information from extracted features, and we thought sequential data are good supervision signals to enhance CNNs. For this reason, more discriminative features can be extracted to represent actions. Therefore, it would be efficient to extract sequential features using an asymmetrical auto-encoder instead of labels.

We then evaluated our method by training with partial labeled data. The performance is shown in Table 3.1. The baseline methods used general supervised learning, and they suffer from overfitting when well-labeled data is not abundant. Our methods always performed better than the baselines when training with partial labeled data since the proposed network succeeded in learning motion patterns and associating skeleton sequences with actions. However, when labeled data was abundant, namely, when using 100% labeled data of the NTU RGB+D dataset, our method did not perform better than the baselines.

3.4.4 Comparison with State-of-the-art Methods

As shown in the third part of Table 3.3, our method learned better representations from unlabeled data than others on both datasets. A similar point of LongT GAN [100], P&C FW-AEC [79], MS^{2}L [47] and our method was to compress skeleton data to middle representations and then generate entire or partial skeleton data from these representations. However, we replaced skeleton data with salient skeleton motion cues as supervision signals which were abstracted from skeleton data.

Our method did not outperform methods using contrastive learning [43, 80]. Those methods learned action representations by distinguishing positive and negative sample pairs. If two data points belong to a positive pair, their distance is made as small as possible, otherwise as large as possible. Thus, given a dataset with N samples, the usable data amount for contrastive learning methods is $C_2^N = \frac{1}{2}N(N-1)$, which is much larger than N, and the usable data amount for ours is just N. Therefore, we did not directly compare our method with contrastive learning methods, and assigned those studies to another group.

Generative learning networks indirectly learn representations by a pretext task, and how good are the representations depends on quality of the pretext task. Contrastive learning networks directly learn representations by making distances of representations belonging to positive pairs close, which is a direct constraint to representations in the latent space. This difference is a possible reason why contrastive learning methods outperformed generative learning methods.

To compare our method with previous supervised methods, when the proposed network was fine-tuned with 100% labeled data, it outperformed a part of previous methods as shown in Table 3.4, [51], [16], [74], [50] and [39], but did not exceed the SOTA performance. When the proposed network was fine-tuned with 40% labeled data, it still outperformed the methods in [16], [74] and [50]. Thus, our method achieved the same level of performance with less labeled data during training by using the proposed unsupervised representation learning.

In Table 3.4, VA-CNN [98], which used very deep CNNs (152 layers) obtained significantly better results than other methods that used RNNs or LSTMs with just several layers. When using methods using 6-layer or 8-layer CNNs, the gap between accuracies were not significantly large. [40]'s method, which used near amount of layers, did not outperform ours under the cross-view protocol, whereas it outperformed ours under the cross-subject protocol. This is because different subjects lead to some scale variant of skeletons, and [40] proposed to normalize

Method	Networks	Layers	NTU RGBD 60 Cross View	 Cross Subject
Du et al. [16]	biRNN	5	63.97%	59.07%
Shahroudy et al. [74]	LSTM	2	70.27%	62.93%
Liu et al. [50]	ST-LSTM	2	77.70%	69.20%
Lee et al. [39]	TS-LSTM v2	unknown	81.25%	74.60%
Liu et al. [51]	AlexNet	8	82.56%	75.97%
Li et al. [40]	AlexNet	8	85.00%	80.20%
Li et al. [40]	ResNet	152	90.10%	84.30%
VA-RNN [98]	LSTM	5	87.60%	79.40%
VA-CNN [98]	ResNet	152	**93.40%**	**88.20%**
Ours (with 40% labeled data)	CNN	6	80.55%	72.14%
Ours (with 100% labeled data)	CNN	6	86.53%	76.52%

Table 3.4: Comparison with state-of-the-art supervised methods.

Method	NTU RGBD 60 (Cross Subject)	NW-UCLA
1% labeled data		
LongT GAN [100]	35.22%	18.22%
MS^{2}L [47]	33.10%	21.28%
Ours	**40.17%**	**32.83%**
10% labeled data		
LongT GAN [100]	62.03%	59.94%
MS^{2}L [47]	**65.17%**	60.45%
Ours	63.49%	**65.11%**

Table 3.5: Comparison with state-of-the-art semi-supervised methods.

coordinates on each axis respectively to solve that problem. In our method, joint coordinates were simply mapped to images without specific processing. As shown in Table 3.5, our method outperformed other semi-supervised methods in most scenarios, which demonstrated the effectiveness of proposed unsupervised representation learning.

3.5 Conclusion and Future Work

3.5.1 Conclusion

We proposed a novel method of unsupervised representation learning for skeleton-based action recognition. By training a novel structure-asymmetrical auto-encoder using the supervision of salient skeleton motion cues, our method achieved better performance compared with previous unsupervised methods, which showed that the auto-encoder learned separable representations. When fine-tuning the network after the proposed unsupervised representation learning, our method was able to keep the same performance level using less labeled data, which showed that the network effectively learned discriminate representations and associated them with actions.

3.5.2 Future Work

The proposed representation learning is unsupervised; however, we still classified actions with the aid of labeled data, and it was shown that higher accuracies were achieved by using more labeled data. It is still a challenge to directly classify actions without labels.

Chapter 4

Psp-Transformer: A Transformer with Data-level Probabilistic Sparsity for Action Representation Learning

4.1 Introduction

Human action recognition is a crucial topic in computer vision since it plays a fundamental role in a wide range of applications, such as video surveillance, human–computer interaction, video understanding for retrieval, etc.

With the development of convolutional neural networks (CNN), some vision-based recognition methods that need strong supervision and a large number of labeled data [9, 30] have achieved significant levels of performance in the past few years. CNNs are good at extracting local features, but neglect global integration. Therefore, non-local networks [90] were proposed to enhance CNNs and make them able to extract global features. Recently, self-attention-based Transformers [85] in natural language processing (NLP), which are designed to capture the global dependencies of every two words in sentences, have been introduced into the computer vision (CV) domain. Transformer-like networks [14, 2] have been applied to sequences of image patches or video blocks (i.e., tokens) treated the same way as words (i.e., tokens) in the NLP domain. Unfortunately, the self-attention mechanism is very time-consuming and requires much larger storage spaces due to the squared complexity of time and space.

RGB images provide discriminative features for action recognition; however, variations in viewpoint changes, background, and the appearances of people can adversely affect the performance. For this reason, skeleton data has attracted much attention since it is robust to those variations, and it includes high-level representations of human behaviors. In some studies [39, 16, 50, 102, 74], recurrent neural networks were proposed to model skeleton movements for capturing temporal relations of actions. Although skeleton data has the advantages of being lightweight and robust, a disadvantage is that specific devices such as depth cameras are needed to recognize skeletons. Hence, action representation learning based on RGB data is still worth exploring since it does not depend on specific devices.

Existing supervised methods have shown remarkable success owing to the use of strong supervision and a large number of labeled data. Well-labeled data is extremely precious since it is time-consuming to annotate a massive amount of videos with tags, and if labeled data is lacking during training, neural networks easily suffer from overfitting. Furthermore, it is difficult and subjective to decide how accurate the annotations of action videos should be. Thus, it is worth studying how to learn action representations from data itself. In many studies, [57, 79, 47], skeleton data is often compressed to low-dimensional representations, and then the input skeleton data is reconstructed from given learned representations.

To the best of our knowledge, only a few studies [78, 42, 54] explored unsupervised representation learning of actions from videos, using an approach in which representations are learned from several frames, and input frames are reconstructed to predict future frames. Therefore, we propose an action representation learning method using videos that exploits a transformer to learn action representations from parts of each video. In detail, each video volume is separated into several 3D blocks, and a certain number of blocks are picked up according to probabilistic values of how large pixel-value changes of the blocks are. For supervised settings, embeddings of blocks and their positions are taken as input data during training. For unsupervised settings, embeddings of blocks only are input to networks to learn middle representations, and the learned representations are utilized to predict the positions of input blocks. The learned representations can be used for other tasks, such as action recognition. The positions of blocks where pixel-value changes are large are crucial cues for learning representations of actions. Not only are representations more discriminative for action recognition, but also the computation time is reduced.

Furthermore, we propose a framework to implicitly fuse vision data and skeleton data in an unsupervised manner, which utilizes multimodal information for mining correct video–skeleton pairs. Indeed, skeleton data includes high-level representations of human behaviors, which are also crucial cues for learning. In a training batch, features extracted by an RNN-based encoder from skeleton data and features extracted by a CNN-transformer-based encoder from videos compose many positive and negative video–skeleton pairs, and they are classified as to whether they belong to the same video. Prediction of the positions of blocks that are picked up from videos and multimodality-contrastive learning are implemented simultaneously during training.

The contributions of our study can be summarized as follows:

i) We propose a transformer-based network for action recognition that takes sparse parts of videos instead of entire videos as training data. Thus, we design a scheme based on events of salient pixel-value changes to make input data sparse but indispensable.

ii) We demonstrate that the proposed network with the designed scheme can not only reduce the time required for training and testing but can also achieve a remarkable level of performance compared with general video transformers [2] under the same supervised training settings.

iii) We also propose a framework of multimodal-contrastive learning for unsupervised action representation learning that utilizes multimodal information for mining correct video–skeleton pairs and position prediction, which guides the learning of comprehensive representations.

iv) We evaluated the framework on action datasets, e.g., NTU-RGBD-60 and PKU-MMD-II, and achieved state-of-the-art results under unsupervised training settings.

4.2 Related Work

4.2.1 Supervised Action Recognition

Since 2D-convolution was extended to 3D-convolution [81], 3D-CNNs [9, 29, 24] have been used to recognize actions from videos. Those studies showed that CNNs are good at capturing the relationship of neighboring pixels, and remarkable performance can be achieved based on training with close to one million items of data [62, 27, 33]. However, there are two disadvantages: it is time-consuming to process vision data, and the global integration of local features is easily neglected.

Therefore, skeleton-based action recognition has attracted the interest of many researchers since the amount of data for skeletons is greatly reduced, which means that processing skeleton data is not time-consuming, and the negative effect of neglecting global integration is naturally reduced. Some studies [39, 50] showed that RNNs perform well for classifying sequential skeleton data since RNNs are good at capturing long-term contextual information of skeleton sequences. In those studies [16, 18], a whole human skeleton was split into several parts, the features of different parts were extracted by different RNNs, and the finally extracted features were fused hierarchically. They belong to a bottom-up manner of integrating local features by utilizing physical structures of human skeletons. However, there is a disadvantage that joint coordinates need to be extracted from videos in advance by using specific algorithms such as OpenPose [28] or devices such as Kinect cameras, which limits the applications of skeleton-based methods.

Recently, another approach known as self-attention-based Transformers [85], which was originally used to capture the relationship of words in the natural language processing domain, has been introduced to solve the problem of neglecting global integration in the computer vision domain. Vision data was separated into sequences of image patches or video blocks, and then the proposed network with a self-attention mechanism was used to capture the relationship of image patches or video blocks globally. In the usual approach to combine CNNs and Transformers to save time and reduce overfitting in many CV tasks [8, 25], features are first extracted by a CNN, a sequence of features is composed, and then a network with a self-attention mechanism is used to capture the relationship of features globally. Although this improves the problem of neglecting global integration, the problem of time-consuming computation becomes more serious since transformers contain many more trainable parameters than CNNs with the same number of layers. Therefore, we designed a scheme to pick up sparse but indispensable parts of videos to reduce the amount of time-consuming computation.

In addition, it has been proposed that a transformer be directly applied to skeleton data for action recognition [71]. In [58]'s study, skeleton data and image data were explicitly fused at a low level and were then fed into a transformer to classify actions. Skeleton data and image data were needed during both training and testing.

4.2.2 Unsupervised Action Representation Learning

A lot of different pretext tasks were proposed to improve the performance of unsupervised representation learning. Among various pretext tasks, a type of pretext tasks, contrastive learning, has shown promising performance. Some studies of contrastive learning, regardless of whether they use vision data or skeleton data, focus on making abundant positive and negative samples using data augmentation techniques and distinguishing them. In [94, 11]'s studies, images transformed from the same input were considered as positive pairs, and images transformed from dif-

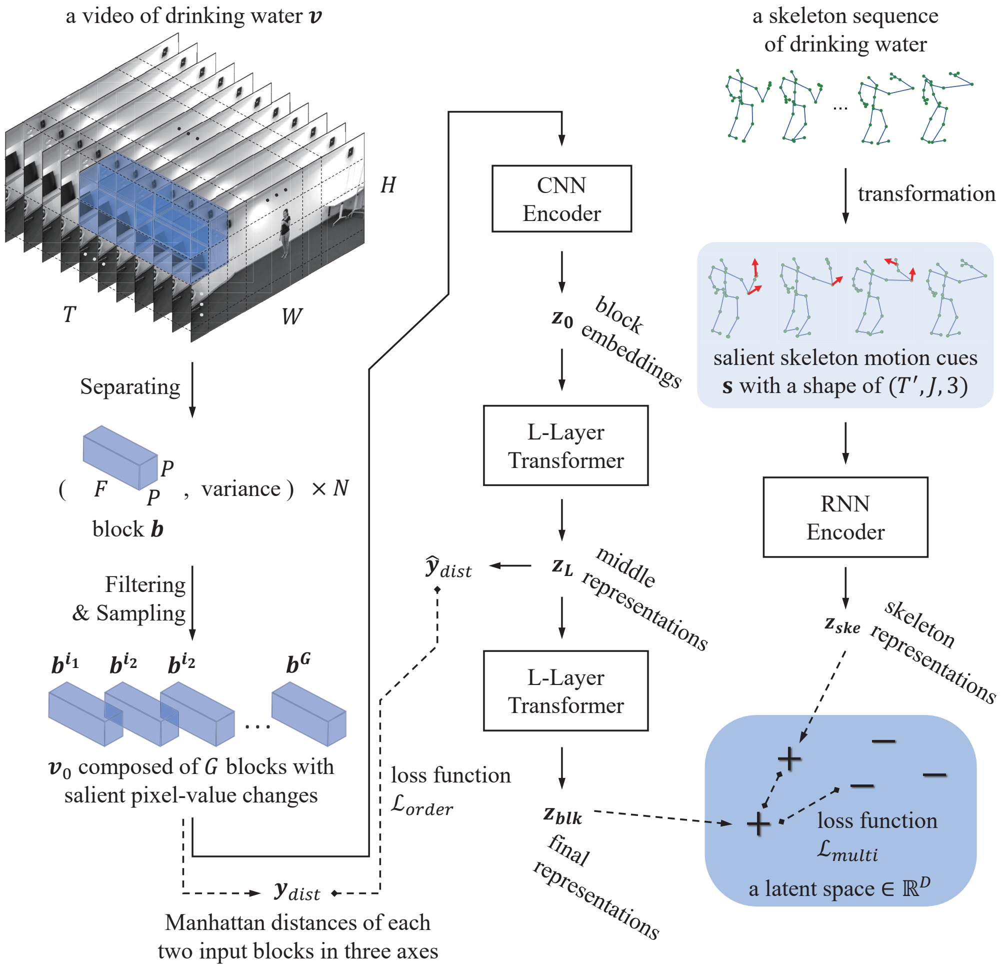

a video of drinking water $\boldsymbol{v}$
H
T
W
Separating
(F P P , variance) × N
block $\boldsymbol{b}$
Filtering & Sampling
$\boldsymbol{b}^{i_1}$ $\boldsymbol{b}^{i_2}$ $\boldsymbol{b}^{i_2}$ $\boldsymbol{b}^{G}$
...
$\boldsymbol{v}_0$ composed of G blocks with salient pixel-value changes
$\boldsymbol{y}_{dist}$
Manhattan distances of each two input blocks in three axes
loss function $\mathcal{L}_{order}$
$\hat{\boldsymbol{y}}_{dist}$
CNN Encoder
$\mathbf{z}_0$
block embeddings
L-Layer Transformer
$\mathbf{z}_L$
middle representations
L-Layer Transformer
$\mathbf{z}_{blk}$
final representations
a skeleton sequence of drinking water
...
transformation
salient skeleton motion cues
$\mathbf{s}$ with a shape of $(T', J, 3)$
RNN Encoder
$\mathbf{z}_{ske}$
skeleton representations
+
+
−
−
−
loss function $\mathcal{L}_{multi}$
a latent space $\in \mathbb{R}^D$

Figure 4.1: Overview of the proposed method. The method simultaneously implements prediction of position relationships of movements with salient pixel-value changes using a vision transformer and multimodality-contrastive learning between representations respectively learned from videos and skeleton sequences.

ferent inputs were considered as negative pairs, and the network was trained to distinguish positive pairs from negative pairs. In [43, 73, 80]'s studies, similarly, abundant positive and negative pairs were made, and mining positive pairs. The difference in our approach is that we implement contrastive learning between different modality data, namely, videos and skeletons.

4.3 Method

The goal of our method is to learn action representations without human-labeled annotations, which allows the capture of relationships of low-level movements in spatiotemporal dimensions. Since videos include much redundant information at the pixel level, which can adversely affect the performance of representation learning, we considered that the input data should be sparse but indispensable. To achieve this, we propose a transformer-like network based on events of salient pixel-value changes, as shown in Fig. 4.1. Only parts with salient movements of videos are fed into the network, which makes the network focus on learning essential relationships of movements in videos.

Moreover, we propose a framework of multimodality-contrastive learning that implicitly fuses features of videos and skeletons. Since skeleton sequences include high-level representations of human behaviors, we expect that networks can learn a better representation space by using multimodality-contrastive learning instead of learning using single-modality data.

4.3.1 Network Architecture based on Events of Salient Pixel-Value Changes

Each video has T frames and a resolution of $(W \times H)$ and is separated into several blocks $\mathbf{b} \in \mathbb{R}^{F \times P \times P}$, and N is the resulting number of blocks. Here F is the interval of frames, and $(P \times P)$ is a small patch size. F should be divisible by T, and P should be divisible by W and H. We refer to these blocks as $\mathbf{v}$

$$\mathbf{v} = [\mathbf{b}^1;\ \mathbf{b}^2;\ ...;\ \mathbf{b}^N],\ \mathbf{v} \in \mathbb{R}^{N \times F \times P \times P},\ N = \frac{T \times W \times H}{F \times P \times P}.$$

Variances of blocks are calculated, and then those variance values of N blocks are clustered by k-Means, where *n_cluster* is set to two. The set of blocks with larger variance values is retained, where the number of these blocks is M, and the rest are dropped out if the variance values are less than a threshold. In this study, the threshold is the mean of the center of large values and the center of small values.

For retained blocks, their variance values were normalized to have a summation of one and the normalized variance values were taken as probabilities to pick up G blocks, and we refer to them as input data $\mathbf{v}_0$.

$$\mathbf{v}_0 = [\mathbf{b}^{i_1};\ \mathbf{b}^{i_2};\ ...;\ \mathbf{b}^{G}],\ \mathbf{v}_0 \in \mathbb{R}^{G\times F\times P\times P}$$

The network is composed of a CNN encoder and a transformer encoder with $2L$ layers that uses a constant latent vector size D through all of its layers. Local spatiotemporal features are first extracted from blocks by a CNN encoder, and we refer to the features as block embeddings $\mathbf{z}_0$.

$$\mathbf{z}_0 = \text{CNN}(\mathbf{v}_0),\ \mathbf{z}_0 \in \mathbb{R}^{G\times D}$$

Unlike general transformers [14, 2, 85], position embeddings are not added to block embeddings z_0 here. The transformer encoder directly captures the global relationship of block embeddings. A transformer encoder layer consists of a multi-headed self-attention (MSA) layer, a feed-forward (FF) layer, and two LayerNorm (LN) layers placed before the MSA and FF layers. Residual connections are applied after the MSA and FF layers.

$$\mathbf{z'}_l = \text{MSA}(\text{LN}(\mathbf{z}_{l-1})) + \mathbf{z}_{l-1},\ l = 1\ ...\ L$$

$$\mathbf{z}_l = \text{FF}(\text{LN}(\mathbf{z'}_l)) + \mathbf{z'}_l,\ l = 1\ ...\ L$$

$\mathbf{z}_L \in \mathbb{R}^{M\times D}$ serves as the middle representation and is fed to a linear layer to predict the Manhattan distance y_{dist} of every two input blocks on three axes.

$$\hat{\mathbf{y}}_{dist} = \text{Linear}(\mathbf{z}_L),\ \hat{\mathbf{y}}_{dist} \in \mathbb{R}^{M\times M\times 3}$$

This can be treated as a pretext task that guides representation learning by predicting the spatiotemporal order from flat sequences of blocks. The loss function $\mathcal{L}_{order}$ is as follows:

$$\mathcal{L}_{order} = ||\mathbf{y}_{dist} - \hat{\mathbf{y}}_{dist}||_2.$$

Three dimensional positions $E_{pos} \in \mathbb{R}^3$ are embedded in D-dimensional space by using a linear layer and added to $\mathbf{z}_L$. We propose using 3D-aware position embeddings instead of learnable 1D position embeddings since we find significant performance gains with that change. The resulting sequence of middle representation $\mathbf{z}_L$ plus position embedding serves as the input to the rest of the L layers.

$$\mathbf{z}_L \leftarrow \mathbf{z}_L + \text{Linear}(E_{pos})$$

$$\mathbf{z'}_k = \text{MSA}(\text{LN}(\mathbf{z}_{k-1})) + \mathbf{z}_{k-1},\ k = L+1 \ ...\ 2L$$

$$\mathbf{z}_k = \text{FF}(\text{LN}(\mathbf{z'}_k)) + \mathbf{z'}_k,\ k = L+1 \ ...\ 2L$$

The representations $\mathbf{z}_{2L}$ learned from videos are taken an average on the dimension of M, and we refer to them as $\mathbf{z}_{blk} \in \mathbb{R}^D$.

4.3.2 Multimodality-Contrastive Action Representation Learning

We use additional representations learned from salient skeleton motion cues $\mathbf{s}$ to implement multimodality-contrastive learning. Salient skeleton motion cues are extracted from skeleton data, where frames without salient motions are dropped, and if joints do not have salient motions, their coordinates are set as zero.

Salient skeleton motion cues $\mathbf{s} \in \mathbb{R}^{T' \times J \times 3}$ that have T' frames and represent J joints with 3D coordinates are embedded in a D-dimensional space by using an RNN encoder. We refer to the embeddings, hidden states of the RNN, as skeleton representations $\mathbf{z}_{ske}$.

$$\mathbf{z}_{ske} = \text{RNN}(\mathbf{s}),\ \mathbf{z}_{ske} \in \mathbb{R}^D$$

There are positive video–skeleton pairs if $\mathbf{z}_{blk}$ and $\mathbf{z}_{ske}$ come from the same video. The loss function based on the noise contrastive estimation loss (InfoNCE) [82] is as follows:

$$\mathcal{L}_{multi} = -\log \frac{\exp(\mathbf{z}_{blk} \cdot \mathbf{z}_{ske}/\tau)}{\exp(\mathbf{z}_{blk} \cdot \mathbf{z}_{ske}/\tau) + \sum\limits_{\mathbf{u} \sim \mathcal{N}_{ske}} \exp(\mathbf{z}_{blk} \cdot \mathbf{u}/\tau)}\ ,$$

where τ is a temperature softening hyper-parameter, and $\mathcal{N}_{ske}$ is a set of skeleton representations that belong to negative video–skeleton pairs in a batch. Finally, the objective loss function is

$$\mathcal{L} = \mathcal{L}_{order} + \mathcal{L}_{multi}.$$

4.4 Experiments

4.4.1 Dataset

NTU RGB+D 60 Dataset (NTU-RGBD-60) [74]. This is a large-scale human action dataset that contains about 56000 videos for 60 action categories performed by 40 volunteers and captured from three viewpoints, such as handshaking, flicking hair, clapping, etc. We used both videos and skeleton sequences in this dataset. The skeletons were 3-dimensional and had 25 joints, and videos had a resolution of 1080 × 1920. We tested our method under the cross-view protocol, where 37113 samples belonging to cameras 2 and 3 were used for training, and 18887 samples belonging to camera 1 were used for testing.

PKU Multi-Modality Dataset Phase II (PKU-MMD-II) [48]. This is a new benchmark for multimodality 3D human action understanding and covers a wide range of human activities. It contains almost 7000 action instances for 49 action categories performed by 13 volunteers and captured from three viewpoints, such as putting on a hat, throwing something, wiping the face, etc. We used both videos and skeleton sequences in this dataset. The skeletons were 3-dimensional and had 25 joints, and videos had a resolution of 1080 × 1920. We tested our method under the cross-subject protocol in which the training dataset has 5295 samples, and the test dataset has 1612 samples.

4.4.2 Implementation

Pre-processing. For the proposed method, we used grayscale images. The original resolution of 1080×1920 was trimmed to 1024×1920 and was finally reshaped as 256×480. The frame interval F was set to 8, and the small patch size $(P \times P)$ was set to (16×16). The number of blocks with salient pixel-value changes was variable, from which 48 blocks were probabilistically selected to be input data. Salient skeleton motion cues were extracted from skeleton sequences and were resampled to have 32 frames. For baselines using entire videos, videos with a resolution of 1080×1920 were cropped to a size of 640×640 from the center, were reshaped to 128×128 and were uniformly resampled to have 32 frames. They can be separated into 256 blocks.

Networks. The proposed transformer had 8 layers with a size of 512 dimensions, where the structure of the layers followed [2]. Layers up to "Mixed_4b" in I3D [9] with pre-trained parameters were used as the CNN encoder, and 3-layer bidirectional gated recurrent units were used as the RNN encoder. For experiments using supervised learning, we used a 4-layer video vision transformer [2] (ViViT-4L) for a baseline, where the input data was blocks with a shape of $256 \times 8 \times 16 \times 16$. To compare with the baseline of the 4-layer ViVit, the proposed transformer was limited to have four layers, and blocks and position information were embedded in a 512 dimensional space by a linear layer and were added before the input. We report other details of the network architecture in the supplementary materials.

Training. For experiments using the proposed networks, we used a NoamOpt optimizer with a learning rate of 0.0003 and a weight decay of 0.3. It was trained for 1000 epochs. The temperature softening hyper-parameter τ was set to 0.1. For experiments using supervised learning, we used an Adam optimizer with a learning rate of 0.0004 and a weight decay of 0.004, and a cyclic learning rate scheduler with restarts including a restart period of 5, a multiplier of 2, and a policy of cosine. They were trained for 155 epochs. The batch size was set to 128 for all situations.

Evaluation. The proposed transformer-like network output 512 dimensional vectors $\mathbf{z}_{blk}$, which were taken as final representations for action recognition. We applied a KNN classifier with $k = 1$ (i.e., 1-nearest neighbor) on the learned

Table 4.1: Comparison of supervised action classification results using NTU-RGBD-60 and PKU-MMD-II. *We applied a network in which the architecture followed [2] and had four layers to save memory and time.

Method	NTU-RGBD-60	PKU-MMD-II
ViViT-4L* [2]	36.43%	15.76%
Ours	**67.2%**	**34.37%**

representations to evaluate the action recognition performance of the proposed method. All samples in the training dataset were used to assign classes similarly to another study [79]. Specifically, we used related code in the scikit-learn package [70] to assign classes and make predictions. Note that KNN classifiers do not require to learn extra weights for classification. For experiments using supervised learning, a fully-connected layer attached at the end of networks was used to make predictions.

4.4.3 Results

First, we show classification results obtained by using supervised learning. As shown in Table 4.1, our network largely outperformed the baseline 4-layer ViViT on both datasets. Experimental results showed that salient movements in videos and their position information were crucial clues to recognize actions, and the use of sparse parts of videos instead of entire videos did not adversely affect the performance.

As shown in Fig. 4.2, loss values of two networks converged to the same low level during training; however, there was a large gap in test accuracy between our network and the 4-layer ViViT, which indicates that a severe overfitting problem happened in the training of the 4-layer ViViT, which used entire videos and learnable 1D position embeddings. It was shown that the proposed probabilistic sparsity of input data and 3D-aware position embeddings can significantly improve the generalization ability, namely to obtain higher accuracies on test data after the end of training.

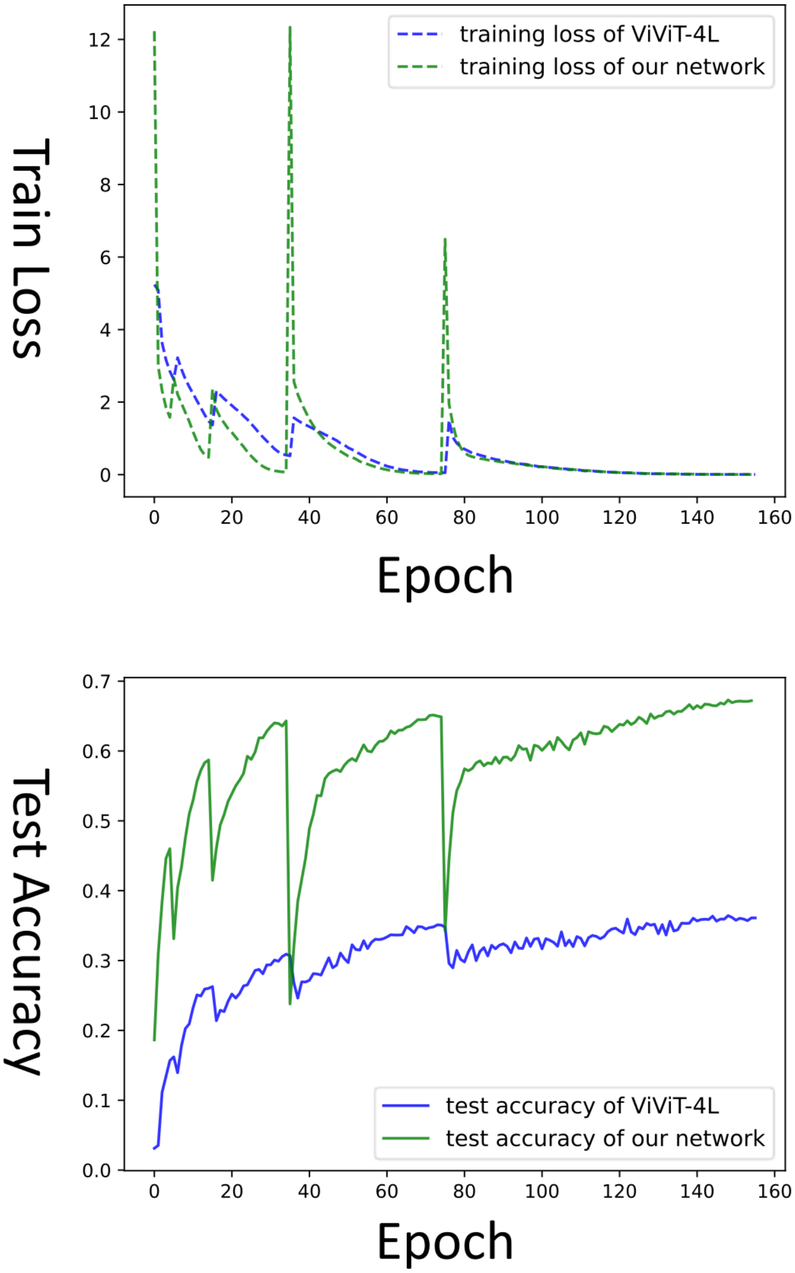


Figure 4.2: Curves of training loss and test accuracy on NTU-RGBD-60. Best viewed in color.

Next, we report the costs of training and testing during learning on NTU-RGBD-60. Compared with the the general 4-layer ViViT, the proposed method could reduce the computing time to about 25% of the original, and also was efficient in terms of GPU memory usage, as shown in Table 4.2. It was demonstrated that

Table 4.2: GPU peak memory usage, GPU training speeds, and GPU inference speeds during learning on NTU-RGBD-60. There were 128 samples in a batch. Speed-up multipliers relative to networks are given in parenthesis.

Method	ViViT-4L [2]	Ours
Layers	4	4
GPU peak memory usage (measured in GB; smaller is better)	20.87	7.38
GPU training speeds (measured in seconds per batch; lower is better)	1.726	0.43 (0.25×)
GPU inference speeds (measured in seconds per batch; lower is better)	0.561	0.137 (0.24×)

making input data sparse is a direct and effective way to reduce redundancy, which could be used to train neural networks with deeper layers and more data.

4.4.4 Comparison with State-of-the-art Methods

As shown in the third part of Table 4.3, our method achieved the best performance compared with methods tested on vision-modality data. Since videos could have different appearances even if they belong to the same action, a cluster of representations learned from videos only have many noises. For this reason, our method additionally used salient skeleton motion cues to fuse features of videos and skeletons by using multimodality-contrastive learning. By the proposed multimodality-contrastive learning, fused features were more discriminative than features extracted from single-modality data. Another difference from methods using RGB modality data is that we used sparse parts of videos instead of entire videos, and the spatiotemporal relationship of sparse parts became a supervision signal to guide representation learning.

Compared with methods tested on other modality data, as shown in the first and second part of Table 4.3, our method still outperformed all other methods. In the method using a pretext task of point cloud colorization, skeleton data is given as point clouds, the relationship of each two joints is broken and becomes guidance to help networks to represent actions. It is a similar point that our method exploited the relationship of each two salient movements in videos. It is

Table 4.3: Comparison of action recognition results with state-of-the-art unsupervised methods on NTU-RGBD-60.

Method	Modality of Test Data	NTU-RGBD-60
TS Colorization [97]	Point Cloud	79.9%
TS+SS Colorization [97]	Point Cloud	82.6%
TS+SS+PS Colorization [97]	Point Cloud	**83.1%**
LongT GAN [100]	Skeleton	52.1%
EnGAN-PoseRNN [38]	Skeleton	77.8%
AS-CAL [73]	Skeleton	64.8%
P & C [79]	Skeleton	76.3%
SeBiReNet [64]	Skeleton	79.7%
3s-CrosSCLR (LSTM) [43]	Skeleton	69.2%
3s-CrosSCLR (ST-GCN) [43]	Skeleton	83.4%
Thoker et al. [80]	Skeleton	**85.2%**
Li et al. [42]	Depth	63.9%
Luo et al. [54]	Depth	66.2%
Shuffle & Learn [60]	RGB	40.9%
Li et al. [42]	RGB	49.3%
Luo et al. [54]	RGB	56%
Ours	RGB	**86.44%**

worth studying video-based methods since they do not need specific devices during inference, such as laser scanners for capturing point clouds or depth cameras for obtaining depth information. In our method, skeleton features were implicitly fused into vision features during training. Therefore, vision features can be directly used for action recognition when performing inference. Notably, our method achieved better performance than former state-of-the-art methods.

As shown in Table 4.4, our method also outperformed methods tested on skeleton-modality data, which shows the effectiveness of the proposed representation learning method. Aside from the fact that we used sparse parts of videos, salient skeleton motion cues were another difference. They were extracted from primal joint coordinates by eliminating meaningless information, and the retained information essentially represents motion dynamics. Our method performed well where salient skeleton motion cues played an important role in guiding representation learning.

Table 4.4: Comparison of action recognition results with state-of-the-art unsupervised methods on PKU-MMD-II.

Method	Modality of Test Data	PKU-MMD-II
LongT GAN [54]	Skeleton	25.95%
P&C [79]	Skeleton	25.5%
MS^{2}L [47]	Skeleton	27.63%
Thoker et al. [80]	Skeleton	**36%**
Ours	RGB	**36.23%**

4.5 Conclusion

We proposed an efficient transformer-based network for action recognition that took sparse parts of videos instead of entire videos as training data and used 3D-aware position embeddings. Compared with general video vision transformers, the proposed method achieved better performance and generalization ability, and in addition, required less time and GPU memory.

We also proposed a framework of multimodality-contrastive learning for unsupervised action representation learning that utilizes multimodal information for mining correct video–skeleton pairs and position prediction. By multimodality-contrastive learning, our network learned more comprehensive representations from implicitly fused features of videos and skeletons. We evaluated the framework on action datasets, e.g., NTU-RGBD-60 and PKU-MMD-II, and achieved state-of-the-art results under unsupervised training settings.

Chapter 5

Conclusion and Future work

This chapter concludes our thesis works and shows the future work of our thesis.

5.1 Conclusion

In this thesis, we proposed three methods for action recognition by using deep neural networks that are trained with fewer or no manual labels.

In the first method, we proposed a framework to recognize fall actions from videos without fine-grained labels, in which annotations of fall actions are not needed by utilizing learning of abundant Activity of Daily Life (ADL) videos. The first variational auto-encoder (VAE) in the framework learns representations of ADL videos only by compressing those videos, and the second VAE gathers representations of ADL data and fall action data into two clusters. The experimental results showed that our method achieved better generalization ability compared to methods using supervised learning with well-labeled data.

In the second method, we propose a framework for general action representation learning using skeleton sequences, in which a structure-asymmetrical auto-encoder is used to learn spatiotemporal representations under the supervision of salient skeleton motion cues. Manual annotations are not needed during the training of the neural network. The experimental results showed the effectiveness of the proposed representation learning, and improvements compared with skeleton-based generative learning methods. When the proposed network was fine-tuned with partially labeled data, our results also outperformed some fully-supervised methods.

In the third method, we propose a neural network for general action representation learning which is trained with paired videos and skeleton sequences and is evaluated using videos only. The network learns representation by simultaneously predicting position relationships of movements with salient pixel-value changes and doing multimodality-contrastive learning between representations that are respectively extracted from videos and skeleton sequences. The experimental results demonstrate the superiority of the proposed method, which efficiently learns discriminative features.

5.2 Future work

Although manual annotations are not needed during representation learning by utilizing pseudo annotations that are automatically generated by programs, a small amount of labeled data still is needed for supervised fine-tuning of neural networks when methods are applied to different datasets. In the future, we will consider reducing the amount of manually-labeled data to zero.

Publications

Journal (Refereed)

- Jiaxin Zhou and Takashi Komuro, "PSp-Transformer: A Transformer with Data-level Probabilistic Sparsity for Action Representation Learning", *Computer Vision and Image Understanding (CVIU)* (under review)
- Jiaxin Zhou and Takashi Komuro, "Detecting Fall Actions of Videos by using Weakly-supervised Learning and Unsupervised Clustering Learning", *Lecture Notes in Computer Science*, Vol. xxxxx *(Advances in Visual Computing. ISVC 2022)*, pp. xxx-xxx (2022) (to appear)
- Jiaxin Zhou and Takashi Komuro, "An asymmetrical-structure autoencoder for unsupervised representation learning of skeleton sequences", *Computer Vision and Image Understanding (CVIU)*, Vol. 222, Article 103491 (2022)

Conference (Refereed)

- Jiaxin Zhou and Takashi Komuro, "Recognizing Gestures from Videos using a Network with Two-Branch Structure and Additional Motion Cues", *Proc. 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pp. 133-137 (2020)
- Jiaxin Zhou and Takashi Komuro, "Recognizing Fall Actions from Videos using Reconstruction Error of Variational Auto-encoder", *Proc. 26th*

IEEE International Conference on Image Processing (ICIP 2019), pp. 3372-3376 (2019)

Conference (Unrefereed)

- 周嘉欣，小室孝: マルチモーダルデータの暗黙的融合による動画像からのジェスチャー認識, 動的画像処理実利用化ワークショップ (DIA 2020) 予稿集, pp. 500-504 (2020)

References

[1] Suad Albawendi, Kofi Appiah, Heather Powell, and Ahmad Lotfi. Video based fall detection with enhanced motion history images. In *Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, pages 1–7, 2016.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6836–6846, 2021.

[3] Dzmitry Bahdanau, Cho Kyung, Cho Hyun, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[4] Greet Baldewijns, Glen Debard, Gert Mertes, Bart Vanrumste, and Tom Croonenborghs. Bridging the gap between real-life data and simulated data by providing a highly realistic fall dataset for evaluating camera-based fall detection algorithms. *Healthcare Technology Letters*, 3(1):6–11, 2016.

[5] Richard E Bellman. *Adaptive control processes: a guided tour.* 2015.

[6] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. In *Proceedings of the Advances in Neural Information Processing Systems.*

[7] Christopher M Bishop. *Pattern Recognition and Machine Learning.* 2006.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020.

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[10] Imen Charfi, Johel Miteran, Julien Dubois, Mohamed Atri, and Rached Tourki. Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and adaboost-based classification. *Journal of Electronic Imaging*, 22(4):041106, 2013.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*, pages 1597–1607, 2020.

[12] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *Proceedings of the International Symposium on Neural Networks*, pages 189–196, 2017.

[13] Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *CoRR*, abs/1611.02648, 2016.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In

Proceedings of the International Conference on Learning Representations, 2021.

[15] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In *Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition*, pages 579–583, 2015.

[16] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015.

[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[18] Georgios Evangelidis, Gurkirt Singh, and Radu Horaud. Skeletal quads: Human action recognition using joint quadruples. In *Proceedings of the 22nd International Conference on Pattern Recognition*, pages 4513–4518, 2014.

[19] Hehe Fan and Mohan Kankanhalli. Motion= video-content: Towards unsupervised learning of motion representation from videos. In *ACM Multimedia Asia*, pages 1–7. 2021.

[20] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 14204–14213, 2021.

[21] Hehe Fan, Xin Yu, Yi Yang, and Mohan Kankanhalli. Deep hierarchical representation of point cloud videos via spatio-temporal decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[22] Yaxiang Fan, Martin D Levine, Gongjian Wen, and Shaohua Qiu. A deep neural network for real-time detection of falling humans in naturally occurring scenes. *Neurocomputing*, 260:43–58, 2017.

[23] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017.

[24] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[25] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019.

[26] Yves Grandvalet and Yoshua Bengio. In L. Saul, Y. Weiss, and L. Bottou, editors, *Proceedings of the Advances in Neural Information Processing Systems.*

[27] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.

[28] Chen-zhi Guan. Realtime multi-person 2d pose estimation using shufflenet. In *Proceedings of the 14th International Conference on Computer Science & Education*, pages 17–21, 2019.

[29] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *Pro-*

ceedings of the IEEE International Conference on Computer Vision Workshops, pages 3154–3160, 2017.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[31] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

[32] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[33] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[34] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3288–3297, 2017.

[35] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019.

[36] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[37] B Ravi Kiran, Dilip Mathew Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36, 2018.

[38] Jogendra Nath Kundu, Maharshi Gor, Phani Krishna Uppala, and Venkatesh Babu Radhakrishnan. Unsupervised feature learning of human actions as trajectories in pose embedding manifold. In *Proceedings of the IEEE winter conference on applications of computer vision*, pages 1459–1467, 2019.

[39] Inwoong Lee, Doyoung Kim, Seoungyoon Kang, and Sanghoon Lee. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1012–1020, 2017.

[40] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin, and Mingyi He. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn. In *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops*, pages 601–604, 2017.

[41] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Proceedings of the Advances in Neural Information Processing Systems*.

[42] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Unsupervised learning of view-invariant action representations. volume 31, 2018.

[43] Linguo Li, Minsi Wang, Bingbing Ni, Hang Wang, Jiancheng Yang, and Wenjun Zhang. 3d human action representation learning via cross-view consistency pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4741–4750, 2021.

[44] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Proceedings of the Asian Conference on Computer Vision*, pages 332–347, 2014.

[45] Xing Li, Qian Huang, Zhijian Wang, Zhenjie Hou, and Tianjin Yang. Sequentialpointnet: A strong parallelized point cloud sequence network for 3d action recognition. *arXiv preprint arXiv:2111.08492*, 2021.

[46] Chih-Yang Lin, Shang-Ming Wang, Jia-Wei Hong, Li-Wei Kang, and Chung-Lin Huang. Vision-based fall detection through shape features. In *Proceedings of the IEEE Second International Conference on Multimedia Big Data*, pages 237–240, 2016.

[47] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2490–2498, 2020.

[48] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. Pku-mmd: A large scale benchmark for skeleton-based human action understanding. In *Proceedings of the Workshop on Visual Analysis in Smart and Connected Communities*, pages 1–8, 2017.

[49] Jun Liu, Henghui Ding, Amir Shahroudy, Ling-Yu Duan, Xudong Jiang, Gang Wang, and Alex C Kot. Feature boosting network for 3d pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):494–501, 2019.

[50] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *Proceedings of the European Conference on Computer Vision*, pages 816–833, 2016.

[51] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[52] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*.

[54] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2203–2212, 2017.

[55] Ulrike Von Luxburg. A tutorial on spectral clustering, 2007.

[56] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[57] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017.

[58] Vittorio Mazzia, Simone Angarano, Francesco Salvetti, Federico Angelini, and Marcello Chiaberge. Action transformer: A self-attention model for short-time pose-based human action recognition. *Pattern Recognition*, 124:108487, 2022.

[59] Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514, 2018.

[60] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Proceedings of the European Conference on Computer Vision*, pages 527–544, 2016.

[61] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. volume 27, 2014.

[62] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):502–508, 2019.

[63] Viet Anh Nguyen, Thanh Ha Le, and Thuy Thi Nguyen. Single camera based fall detection using motion and human shape features. In *Proceedings of the Seventh Symposium on Information and Communication Technology*, pages 339–344, 2016.

[64] Qiang Nie and Yunhui Liu. View transfer on human skeleton pose: Automatically disentangle the view-variant and view-invariant information for pose representation learning. *International Journal of Computer Vision*, 129(1):1–22, 2021.

[65] Jacob Nogas, Shehroz S Khan, and Alex Mihailidis. Deepfall: Non-invasive fall detection with deep spatio-temporal convolutional autoencoders. *Journal of Healthcare Informatics Research*, pages 1–21, 2018.

[66] Adrian Nunez-Marcos, Gorka Azkune, and Ignacio Arganda-Carreras. Vision-based fall detection with convolutional neural networks. *Wireless Communications and Mobile Computing*, 2017, 2017.

[67] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.

[68] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.

[69] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[70] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[71] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Skeleton-based action recognition via spatial and temporal transformer networks. *Computer Vision and Image Understanding*, 208-209:103219, 2021.

[72] Hossein Rahmani, Arif Mahmood, Du Q Huynh, and Ajmal Mian. Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition. In *Proceedings of the European Conference on Computer Vision*, pages 742–757, 2014.

[73] Haocong Rao, Shihao Xu, Xiping Hu, Jun Cheng, and Bin Hu. Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. volume 569, pages 90–109, 2021.

[74] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016.

[75] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. volume 27, pages 568–576, 2014.

[76] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[77] Leslie N Smith. Cyclical learning rates for training neural networks. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 464–472, 2017.

[78] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *Proceedings of the International Conference on Machine Learning*, pages 843–852, 2015.

[79] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9631–9640, 2020.

[80] Fida Mohammad Thoker, Hazel Doughty, and Cees GM Snoek. Skeleton-contrastive 3d action representation learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1655–1663, 2021.

[81] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks.

In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[82] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

[83] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[84] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

[85] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. volume 30, 2017.

[86] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014.

[87] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4006–4015, 2019.

[88] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision*, pages 20–36, 2016.

[89] Pichao Wang, Zhaoyang Li, Yonghong Hou, and Wanqing Li. Action recognition based on joint trajectory maps using convolutional neural networks.

In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 102–106, 2016.

[90] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 7794–7803, 2018.

[91] Yancheng Wang, Yang Xiao, Fu Xiong, Wenxiang Jiang, Zhiguo Cao, Joey Tianyi Zhou, and Junsong Yuan. 3dv: 3d dynamic voxel for action recognition in depth video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–520, 2020.

[92] Lu Xia and JK Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2834–2841, 2013.

[93] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the International Conference on Machine Learning*, pages 478–487, 2016.

[94] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. volume 33, pages 6256–6268, 2020.

[95] Tao Xu and Yun Zhou. Elders' fall detection based on biomechanical features using depth camera. *International Journal of Wavelets, Multiresolution and Information Processing*, 16(02):1840005, 2018.

[96] Ziwei Xu, Xudong Shen, Yongkang Wong, and Mohan S Kankanhalli. Unsupervised motion representation learning with capsule autoencoders. volume 34, 2021.

[97] Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C Kot. Skeleton cloud colorization for unsupervised 3d action representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 13423–13433, 2021.

[98] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1963–1978, 2019.

[99] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 1933–1941, 2017.

[100] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[101] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

[102] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.