Pose2Act: Transformer-based 3D Pose Estimation and Graph Convolution Networks for Human Activity Recognition

by

Dias Aimyshev

Submitted to the Department of Computer Science or Data Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science or Data Science

at the

NAZARBAYEV UNIVERSITY

June 2023

Author .....
Department of Computer Science or Data Science
07.04.2023

Certified by .....
Adnan Yazici
Full Professor
Thesis Supervisor

Accepted by .....
Vassilios D. Tourassis
Dean, School of Engineering and Digital Sciences

Pose2Act: Transformer-based 3D Pose Estimation and Graph Convolution Networks for Human Activity Recognition

by

Dias Aimyshev

Submitted to the Department of Computer Science or Data Science
on 07.04.2023, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science or Data Science

Abstract

The rise of deep learning has brought significant attention to two tasks in computer vision: pose estimation and human activity recognition. While human activity recognition has various applications in IoT systems, pose estimation is critical for motion tracking and prediction in virtual and augmented realities, robotics, and other fields. Despite being distinct tasks, they are closely linked, and this study focuses on merging pose estimation, which generates body joint coordinates, and skeleton-based activity recognition, which operates on the given joints. The study uses a visual transformer for 3D pose estimation, viewing joints as spatial features and neighboring frames as temporal features. Meanwhile, graph convolution networks are used for activity recognition based on a 3D skeleton, which has produced state-of-the-art results. However, these outcomes are based on 3D coordinates generated by motion capture systems and have limitations in their applicability and robustness. To overcome these limitations, the two models are merged into a single End2End network. The proposed approach is enhanced by applying various data transformations, modifications, pre-training, and fine-tuning of different architecture components. The research achieves a 90.3% activity recognition cross-subject accuracy score on the NTU RGB+D test dataset, comparable to the state-of-the-art using generated 3D input, and outperforms other models using 2D input by predicting 3D coordinates in the process.

Thesis Supervisor: Adnan Yazici
Title: Full Professor

Acknowledgments

First of all, I would like to express my gratitude to my supervisor Professor Adnan Yazici. He gave me initial direction and often guided me throughout the process of the research. I would like to extend my thanks to my colleague Aldiyar Bolatov. From our discussions, we came up with many research ideas that found their application in the thesis. Additional thanks to Professor Adnan Yazici and the Office of the Provost for providing us with a proper workspace and hardware to conduct research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of IoT systems has led to a considerable increase in the use of computer vision for pose estimation and human activity recognition. These two tasks have a shared requirement for skeleton data. Pose estimation involves recognizing body joints from video inputs, and due to the growing use of 3D technology in virtual and augmented reality, there is a surge of interest in extracting 3D coordinates from pose estimation.

Activity recognition, on the other hand, can be achieved by utilizing various types of information such as sensor data, RGB video, or depth maps. While RGB video is a more practical option due to its lack of dependence on additional equipment beyond conventional cameras, recent studies have indicated that skeleton data possess a distinct advantage over RGB videos in terms of robustness to noise, background interference, and lighting [10].

According to recent studies, skeleton-based activity recognition models are highly effective despite their compact size, producing state-of-the-art outcomes. Nevertheless, these models can only be utilized with 3D data acquired through Microsoft Kinect cameras [30] or intricate motion capture systems [15]. As a result, it is reasonable to integrate these models with a pose estimation task, which can take RGB video as input and anticipate the skeleton data.

Although 2D joint detection has been extensively studied, state-of-the-art results for activity recognition are obtained using 3D skeleton data. When 2D data is used

instead, the model's performance decreases significantly due to the loss of information from the z-axis [10]. Therefore, a more advantageous approach is to extract 3D coordinates from 2D video. This can be achieved by using a reliable 2D joint extractor along with a 2D to 3D conversion model. Several studies have indicated that acquiring 2D coordinates first and then converting them to 3D is preferable to directly searching for 3D coordinates from the video [47, 12, 13]. As a result, it is feasible to construct a single model that takes advantage of the availability of RGB cameras and the high performance of 3D skeleton-based models.

Graph convolution networks are utilized for activity recognition, and transformers are used for 3D pose estimation. These solutions share similar principles as they both involve learning spatiotemporal relationships of body joints within a single frame and across multiple frames. To achieve this, HAR models use graph convolution networks to learn from spatial features and temporal convolution networks to learn from temporal features. GCN effectively represents the human skeleton as a graph, which yields high results. For learning from sequential data of skeleton data, TCN outperforms RNN and LSTM. Transformers are successful in both parts of 3D pose estimation and outperform other methods for learning spatiotemporal relationships of sequence-to-sequence and sequence-to-frame mapping. In a similar vein, HAR implements an attention mechanism to identify the most significant relationships between single and multiple frame joints.

A unified model was developed to address the limitations of the state-of-the-art 3D skeleton HAR model on 2D data. The model comprises two components: a 3D pose estimation module and a HAR module. To account for spatiotemporal relationships, graph convolution networks, and temporal convolution networks were utilized, similar to previous HAR studies [20, 9]. A graph was created using the hierarchy decomposition method from modified edge sets for each center of mass and employed an attention module to identify the most significant edges for joint relationships. The findings showed that combining different data streams via a six-way combination of bones and joints for the three centers of mass was the most effective.

The 3D pose estimation model comprises a temporal encoder, a simple multi-layer perceptron, and the original transformer architecture proposed in [31, 46]. A dedicated strived transformer was used instead of a simple regression module to predict a central frame, as the temporal encoder is lightweight [31]. To accommodate the dataset structure, a sequential model combination was performed, where 3D pose estimation was performed on a sequence of overlapping frames to predict 64 central frames, followed by the use of the HAR model to classify activities based on these predictions.

The findings showed that combining different data streams via a six-way combination of bones and joints for the three centers of mass has been effective. To identify the most significant edges for joint relationships, a graph is created using the hierarchy decomposition method from modified edge sets for each center of mass and employed an attention module.

The proposed unified model is capable of addressing the limitations of the state-of-the-art 3D skeleton HAR model on 2D data. It takes advantage of the availability of RGB cameras and the high performance of 3D skeleton-based models, utilizing GCN, TCN, and transformer architectures to learn spatiotemporal relationships of body joints and achieve state-of-the-art results in activity recognition.

The contributions of this study are as follows:

- An innovative End2End model called Pose2Act has been developed using advanced techniques and models from the latest research [20, 31, 46, 9, 46]. This model works by first converting 2D input to 3D coordinates and then predicting activity tags without relying on motion capture systems to generate 3D skeletons for human activity recognition (HAR). Instead, it predicts 3D skeletons using well-established 2D pose prediction models. Our approach has the added advantage that it has the potential to improve model performance.

- In order to determine the optimal ratio of 3D pose frames to HAR window size, a series of experiments was conducted. However, because of the structure of the NTU RGB+D dataset and its reliance on state-of-the-art 3D pose estima-

tion models in the regression part, it was not feasible to put them sequentially without modifications. Therefore, a method was provided to overlap the input frames to produce an output of reasonable size for the HAR model.

- The proposed model, which relies on a 2D projection approach, has yielded remarkable results on the NTU-RGB+D dataset. It has surpassed other models that utilize 2D input and has performed equally well as conventional 3D skeleton-based models that necessitate 3D input. While the experimental results of this model are slightly inferior to those generated by 3D models, its scalability, adaptability, and capacity to enhance scores make it a compelling option.

The remaining chapters of the thesis are structured as follows: Chapter 2 covers the background on pose estimation, activity recognition, and state-of-the-art deep learning techniques. Chapter 3 reviews the latest research on pose estimation and activity recognition tasks. Chapter 4 presents the methodology used to tackle these tasks. Chapter 5 details the experiments conducted and their results. Lastly, Chapter 6 offers a summary of the entire thesis.

Chapter 2

Background

2.1 Human Activity Recognition

The human activity recognition task can be categorized into different approaches by researchers. These approaches include sensor-based, which relies on sensor readings; video-based, which uses video streams as data; and skeleton-based, which involves a set of coordinates of the human body or skeleton as the data.

2.1.1 Sensor-based

The rapid development of sensor-based activity recognition is attributed to the growth of IoT. This is due to sensors causing fewer data privacy issues. Wearable and ambient sensors have been the focus of research in this area. Currently, the state-of-the-art is automatic feature extraction using deep learning models [3], which have been combined with classification to create end-to-end learning. CNN, LSTM, and Autoencoders are the main models used in this field. CNN is effective in feature extraction, Autoencoders excel in unsupervised learning and noise robustness, and LSTM is used to address irregular sampling issues. However, recognizing complex activities and multiple users is still a challenge for this type of sensor [3].

Regarding wearable sensors, the current state-of-the-art are feature-level fusion algorithms. Using multiple sensors has led to better results, and deep learning models

have outperformed traditional machine learning methods [29]. CNN, DBN, LSTM, and Autoencoders are commonly used models. One example of a state-of-the-art model is a combination of CNN and bidirectional LSTM [27]. which is most suitable for long-term activity recognition. Another recent study focused on using 3D data from multiple sensors by applying multidimensional convolution networks [41]. This approach is said to allow better extraction of spatio-temporal features from different sources and sensors.

2.1.2 Approaches to Video-based Activity Recognition

The video-based approach to activity recognition can benefit from some of the findings in wearable sensor-based recognition, as RGB cameras can be included as a type of sensor. However, typically, researchers prefer to distinguish between these two approaches.

Video-based activity recognition involves several stages, including human detection, segmentation, feature extraction, and classification [32]. This task can be described as the classification of video frames. Like sensor-based recognition, the current research focus is on automatic feature extraction using deep learning. However, some studies attempt to use hand-crafted features combined with neural networks, such as the Gaussian Mixing Model and magnitude of Optical Flow to detect and segment human shape, and then use stacked sparse autoencoders for classification [11]. This method achieves comparable results to state-of-the-art models.

The survey categorizes deep learning approaches for video-based activity recognition into CNN and RNN-based [32]. The CNN approach involves a stream of multiple CNN for spatial and temporal features, which are then combined in the final layers, and it also includes a sequential approach. For example, one study used 3D CNN to extract spatial-temporal interest points [37]. This method achieves similar results to the other approach, which will be described shortly. The advantage of 3D CNN is that it uses filters on the entire video rather than separate frames [32].

The RNN-based approach combines CNN with LSTM in various ways [32]. LSTM can learn high-level features extracted by CNN. This method performs better for

video classification because LSTM can learn from correlated frames, but it is also more resource-intensive.

Some studies aim to combine multiple methods, such as video and sensors. For example, one paper proposes using a visual transformer for acceleration sensors and camera data [2]. By concatenating the results of both methods, the authors achieve state-of-the-art results.

2.1.3 Approaches to Skeleton-based Activity Recognition

Skeleton-based activity recognition is closely linked to pose estimation. To obtain data for skeleton-based models, either sensor data or output from a pose estimation model is required. Some researchers include this approach in the video-based approach since the information is obtained from the video source [32]. This is achieved by applying a pose estimation algorithm to obtain joint positions. Alternatively, data can be obtained using sensors, making the approach more similar to sensor-based recognition. For example, one study proposed using Microsoft Kinect cameras to obtain joint coordinates [30]. Compared to RGB input, skeleton data is more robust to noise such as lighting or background [9, 35]. With regard to sensors, some studies have shown that predicted joint coordinates are more accurate than using sensors [10].

2.2 Pose Detection

2.2.1 2D Pose Detection

In computer vision, 2D pose detection refers to the process of identifying and localizing the positions of body joints or a skeleton in RGB video or separate frames. Deep learning-based methods are commonly used for 2D pose detection, where joint features are extracted from input images and regression is used to estimate joint locations [4]. Recent research has focused on improving the accuracy and efficiency of 2D pose detection, such as developing multi-stage architectures for better detail capture and

localization accuracy [36]. Other approaches include using temporal information to track body movements over time and utilizing spatial relationships between joints for robustness in challenging scenarios [42].

2.2.2 3D Pose Detection

There are two ways to obtain 3D skeleton data from pose detection, either directly from the RGB video source or by converting 2D to 3D coordinates. Currently, the latter method is considered state-of-the-art, as it achieves better performance when combined with state-of-the-art 2D pose detectors [47, 12, 13]. Detecting 2D coordinates first is generally more accurate and easier due to the availability of large annotated datasets [4]. Furthermore, 2D to 3D models are lightweight, making it easier to conduct experiments and compare metrics using the same 2D pose detector.

2.3 Transformers

Originally designed for natural language processing tasks, the transformer architecture is based on a self-attention mechanism that allows the model to attend to different parts of the input sequence when computing the output at each position [39]. Compared to recurrent neural networks (RNNs) and long short-term memory (LSTM) models that process input sequences sequentially, the transformer's self-attention mechanism is more flexible and efficient. The transformer is composed of an encoder and a decoder, with each layer in both consisting of attention and feed-forward layers. The multi-head self-attention mechanism allows the model to attend to different parts of the input sequence simultaneously, while the feed-forward network applies a non-linear transformation to each position independently. The output of each sub-layer is fed into a residual connection and layer normalization operation before being passed to the next layer. Figure 2-1 shows the architecture of the original transformer model.

The transformer architecture has been adapted for computer vision tasks as well. One such adaptation is the Visual Transformer, which processes an image as a grid of

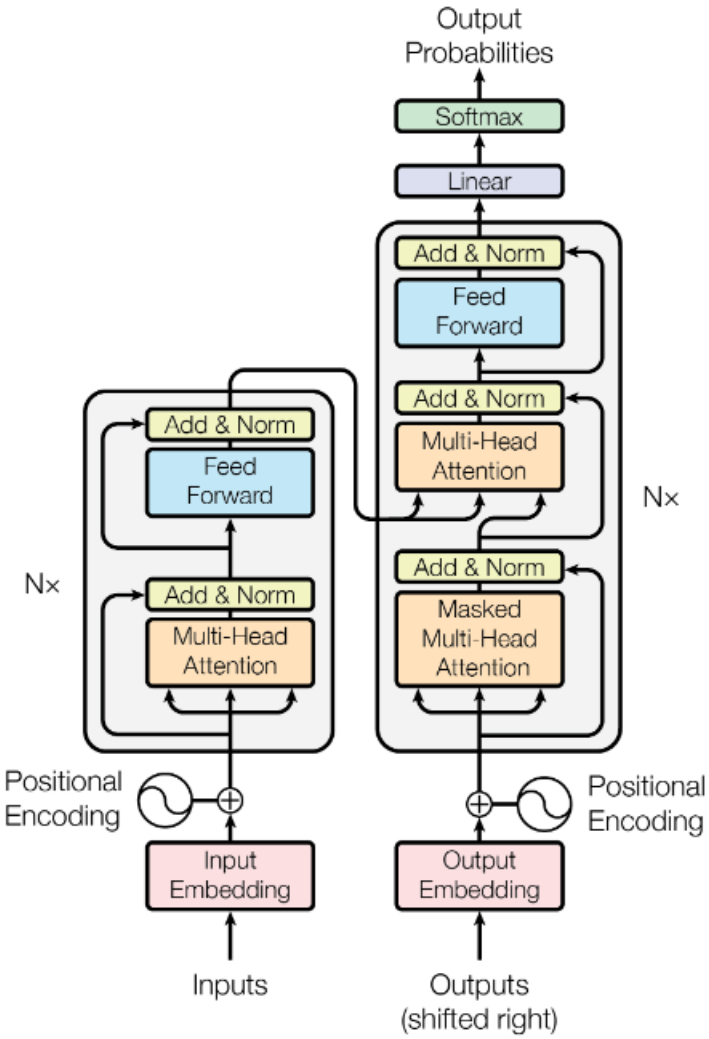

Figure 2-1: Architecture of vanilla transformer [39]

patches, treating each patch as a sequence of tokens similar to words. This approach encodes the relative positions of the patches to capture spatial information using a mechanism [8]. Another variation, called the strided transformer, has been developed specifically for pose estimation tasks. It replaces fully connected layers with strided convolutions to enable a deeper network and reduce sequence length. Figure 2-2 illustrates the architecture of the strided transformer [21].

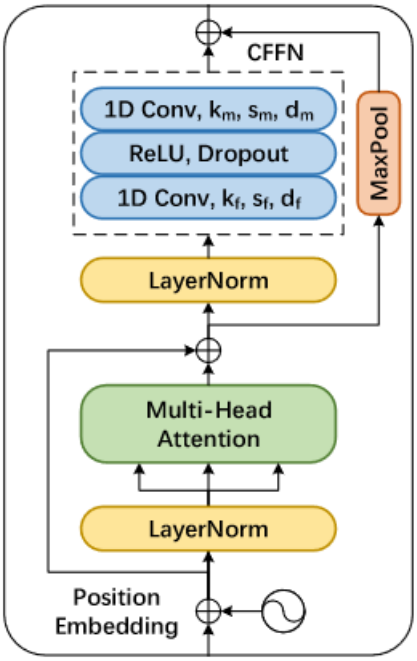

Figure 2-2: Modifications of architecture by strided transformer [21]

2.4 Graph Convolutional Network

Graph Convolutional Network (GCN) is a type of neural network designed specifically for processing data in graph form. Its architecture is composed of multiple layers, each of which conducts a graph convolution operation on node features, taking into account the features of neighboring nodes [16]. This enables the GCN to acquire representations that capture the graph's structural information. In addition, the GCN incorporates a pooling operation that aggregates the features of neighboring nodes to produce a summary representation of the graph.

There have been several advancements and modifications made to the GCN. The Graph Attention Network is one such development, which introduces attention mechanisms to dynamically weigh the contributions of each neighbor node during the convolution operation [40]. This enhances the GCN's ability to learn more expressive node representations and improve performance. Another modification is the Edge-Conditioned Convolution, which introduces a learnable edge filter to generalize the convolution operation on graph-structured data [34]. This makes it possible to

perform flexible and expressive feature learning for a variety of graph-related tasks.

Chapter 3

Related Works

3.1 3D Pose Estimation

Initial attempts at 3D pose estimation tasks using deep learning techniques were carried out using CNN and RNN. Although CNN focused on spatial correlations and RNN learned temporal relationships, both had limitations in utilizing spatio-temporal features [47]. As a result, recent research has shifted towards utilizing transformers and GCNs, which have demonstrated state-of-the-art results.

3.1.1 Transformers

Transformers are a state-of-the-art solution that can be used in various ways depending on the type of research being conducted. For example, in video-based activity recognition, the spatial-temporal structure of transformer architecture has been widely adopted.

One approach is to directly apply transformers to 2D joint coordinates. However, in order to optimize the computation of attention between all joints, researchers have proposed encoding local relationships between the 2D joints in all frames and analyzing the global dependencies between spatial features [47]. The model consists of two parts: a spatial transformer module that works with joints and a temporal module that works with frames. The regression module predicts the 3D pose for the

center frame. The mean per-joint position error (MPJPE) on the Human3.6M dataset is 44.3 [47]. Another attempt to improve performance was made by adding Cross-Joint Interaction and Cross-Frame Interaction [12], which resulted in an MPJPE of 43.7 on the Human3.6M dataset and better feature representations.

One approach involves integrating transformers with epipolar geometry to enhance 2D pose estimation through 3D-informed features [13, 25]. While transformers yield impressive outcomes, The applicability of the epipolar geometry is restricted to multi-view pose estimation, thereby limiting its potential.

In a recent study by Zhang et al. [46], a novel architecture was proposed that employed two transformer blocks. The temporal block was utilized to model the motion of each joint, which enabled the learning of temporal correlations between frames. The spatial block was used in the same way as in the previous method. According to the authors, this new architecture resulted in better spatio-temporal feature encoding compared to the previous method, which relied on the center frame. The authors also reported achieving state-of-the-art results, with an MPJPE of 39.8 on the Human3.6M dataset [46]. By accounting for the motion of the joints, the new method achieved better spatio-temporal correlation and improved inference speed. The authors noted that other sequence-to-sequence models, such as GCN, resulted in overly smooth global modeling ability between input and output sequences [46].

To improve MPJPE scores, researchers have attempted to sequence all predicted 3D skeletons to map them to a single central frame. One of the early attempts involved using a multi-layer perceptron to reduce the output of the transformer [45]. This module, called Regression Head, has also been utilized in other recent studies [46, 44]. Another alternative approach involves using a strided transformer [31, 21]. As a result, a more accurate central frame is constructed, leading to state-of-the-art scores.

3.1.2 Graph Convolution Networks (GCN)

Graph Convolution Networks (GCNs) are a cutting-edge approach used to capture the relationships between skeleton joints, which is similar to previous methods. In

GCNs, nodes of a graph are considered as joints, while edges represent connections between them. This makes GCNs naturally applicable for both 3D pose estimation and Human Action Recognition (HAR), given that the structure of the data is a time series of joint coordinates [45].

Researchers in this field attempted to capture non-local dependence of distant joints. They presented a paper inspired by the attention module used in transformers, which introduced a channel squeezing fusion layer to suppress noise from distant joints. This approach fused the squeezed information with close joints to deal with complex poses and enhance the performance of other graph methods. An alternative approach using conditional convolution and directed graph network achieved slightly worse results but still managed to improve other graph approaches (MPJPE on Human3.6M dataset 47.9 vs 41.1) [14]. Some studies also directly used attention modules with GCNs to weigh the importance of each node based on its spatial and temporal relationships with other nodes, mainly in HAR tasks [20, 9]. However, this approach did not receive much development [43].

3.2 Human Action Recognition (HAR)

The field of skeleton-based HAR has seen continuous development of various approaches over time, including Convolutional Neural Networks (CNN) and Graph Convolution Networks (GCN). Recent studies show that GCN with Temporal Convolution Network (TCN) module and 3D CNN have surpassed all other solutions in terms of performance [10, 20, 9, 38].

3.2.1 GCN for skeleton-based action recognition

Graph Convolution Networks (GCN) have become the leading method for skeleton-based action recognition because of their effectiveness in modeling the relationship between joints and their connectivity in a human skeleton. GCN assigns each node in the graph a feature vector that is convolved with the feature vectors of its neighboring nodes [43]. By aggregating the output features of GCN layers, the model accurately

classifies the action performed in the video.

Another widely used approach for skeleton-based tasks is the integration of attention modules, which are used to select significant joints for specific actions and locate joints between frames [35, 20]. When combined with GCN, attention modules achieve the best results, as the module is also used to connect distant joint edges [20].

To train on spatial features and capture temporal relations in HAR, researchers use a graph convolution network in combination with a Temporal Convolution Network (TCN) [20, 9]. TCNs are popular in activity recognition because they can model temporal dependencies in sequential data. They consist of a 1D convolutional layer applied to the input sequence, followed by multiple residual blocks. These blocks include a dilated convolutional layer and two normalization layers, allowing TCNs to capture long-term dependencies and facilitate the training of deep networks [1]. Recent studies explore the use of TCNs in skeleton-based activity recognition, investigating the effect of different input representations such as 2D projections and bone vectors on model performance [43, 9, 20]. The combination of GCN and TCN achieves state-of-the-art results in learning spatio-temporal relationships in HAR.

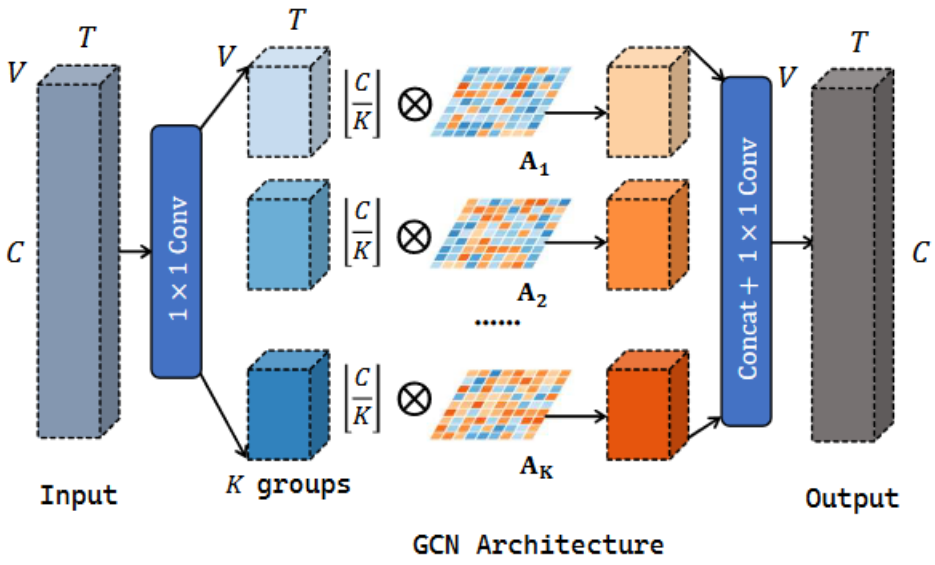

Figure 3-1: Architecture of GCN in HAR [9]

In addition to utilizing graph convolution layers within GCN, the utilization of

edge convolutions has been suggested [20]. The primary benefit of this approach is the ability to learn the connections between nodes that are not directly linked in the graph. This technique is also applied in a comparable fashion within the attention module. Unlike previous studies that attempted to quantify joint and frame degrees [35], edge sets are identified using this method, which has proven to be more effective in recognizing activities.

3.2.2 Graphs

At first, skeleton data graphs were created by manually connecting joints based on their natural connections [43]. However, recent studies indicate that these graphs fail to capture all the necessary relationships between nodes [38, 20]. To address this, a common approach is to incorporate dynamic graph learning by adding layers [38, 9, 33]. Another proposed approach is to construct graphs based on different centers of mass [20, 7]. In this method, graphs are built as follows:

- For each center of mass there is a pre-defined set of edges.
- These sets are used to construct a rooted tree.
- The adjacency matrix is formed from the tree.
- All nodes from neighboring sets are then connected, to include fully connected edges from the tree.

According to the authors, this allows coverage of additional joint relationships.

3.2.3 Ensembling in Skeleton-based Human Activity Recognition

Ensembling techniques are commonly used in skeleton-based Human Activity Recognition (HAR) to improve accuracy. This method involves training the same model on different data streams and combining the results. The ensemble was initially introduced to capture short and medium-term relationships in skeleton data [18] and

has since been utilized in state-of-the-art models, with the most common approach being a 4-way ensemble [5, 9, 38]. The data streams used in this ensemble method include bones, joints, bone motion, and joint motion. Attempts to improve ensembling have included finding optimal coefficients through training small 1D-CNNs [38] and adding more data streams, such as multi-model representation, which considers relations between joints and bones [7], or the same data for different centres of mass [20]. Ensembling has shown to improve results by 3-4

3.3 Multitask Learning in Human Activity Recognition and Pose Estimation

Some research focuses on combining activity recognition and pose estimation into a single model through multitask learning. This approach combines the processes into an end-to-end model and aims to optimize the system. One paper proposed predicting and refining pose and action in parallel [24] to speed up inference compared to the traditional sequential approach of building action recognition on top of pose estimation. This was achieved by using a prediction block where pose and action are predicted and injected into the network. The pyramid residual module was used to upscale and downscale features after prediction blocks, allowing for a trade-off between accuracy and speed. Another method to handle these tasks proposed skip frames based on their difference [11]. Pose estimation is run on separate frames, while action recognition is run on the video sequence.

Chapter 4

Methodology

4.1 Network Merging

The NTU-RGB+D dataset has a unique structure that made it challenging to adopt the parallelization method described earlier to improve inference speed [24]. This is because the pose estimation model only predicts the central frame, requiring a stack of such predictions. As a result, we decided to build a Human Activity Recognition (HAR) model on top of the pose estimation model. This approach is similar to the one described in the paper by Gnouma et al. [11], where individual frame pose predictions are made, and a sequence of frames is used for activity labeling. The entire process of taking 2D key points as input and predicting the activity is outlined in Algorithm 1.

4.2 Transformers in 3D Pose Estimation

As stated in the Related Works chapter, Transformers are currently known for their exceptional performance in various tasks, including 3D pose estimation. Figure 4-1 illustrates the proposed model's architecture, which consists of three main parts in series: an encoder and two transformer blocks. The input size includes the batch size, (x,y)-coordinates, sequence of frames, joints, and person. In the MPI-INF 3DHP dataset, there is only one person in each sample, while the NTU-RGB+D dataset

Algorithm 1 The flow of 3D pose estimation and HAR

Input: $Model\_Pose$: model for 3D pose estimation, $Model\_HAR$: a model for HAR, $NTU\_dataset$: preprocessed NTU RGB+D dataset in Pytorch, where each sample has a shape (dimensions, window, number of joints, number of people)

Output: $output\_har$: activity label predictions

$in\_channels \leftarrow 2$ ▷ Number of input channels, for 2D input is 2
$out\_channels \leftarrow 3$ ▷ Number of output channels, for 3D output is 3
$num\_joints \leftarrow 17$ ▷ Number of joints in skeleton data
$num\_classes \leftarrow 60$ ▷ Number of activity labels
$frames \leftarrow 27$ ▷ Length of frame sequence for central frame prediction
$window \leftarrow 64$ ▷ Number of frames for 1 activity sample prediction
▷ Initialize 3D pose and HAR models, in End2End viewed as a single step
$model\_pose \leftarrow Model\_Pose(in\_channels, out\_channels, window\_size, num\_joints, frames)$
$model\_har \leftarrow Model\_HAR(out\_channels, num\_classes, window)$
$Data\_loader \leftarrow torch.DataLoader(NTU\_dataset)$
$3D\_joints \leftarrow []$ ▷ Intermediary 3D pose estimation storage
for $data$ in $Data\_loader$ **do**
  for $idx \leftarrow 0$ to $window$ **do**
    if $data \neq 0$ **then** ▷ If sample is not empty
▷ Divide sample into overlapping sequences of desired length (27, 81, 243, etc.) for 3D pose prediction of central frame
      $3D\_output \leftarrow model\_pose(data[:, :, idx : idx + frames, :, :])$
      $3D\_joints \leftarrow 3D\_output$
    else if $data = 0$ **then** ▷ NTU RGB+D dataset contains zero data for a second person for single person activities
      $3D\_joints \leftarrow zeros()$ ▷ 3D pose estimation model would still try to predict 0 data, so just fill output data with 0
  $output \leftarrow model\_har(3D\_joints)$ ▷ Activity prediction
  $output\_har \leftarrow max(output)$ ▷ Pick activity label with highest confidence score

includes samples with two persons.

The encoder aims to capture spatial features between joints within a frame, and it employs a simple structure of multi-layer perception. Research has shown that using an encoder, which is lighter than a transformer, for learning spatial correlations is sufficient, and when combined with other solutions, it achieves better results than using the transformer described in the paper [46].

The first transformer captures temporal features between the frames, and the number of frames may vary, but originally it was proposed to use 9, 27, or 81 frames [47]. The model uses up to 81 frames for the MPI-INF-3DHP dataset, which is the maximum seen. The vanilla transformer architecture described in the paper [39] is used.

The second transformer takes all frames with predicted 3D coordinates and maps them into a single frame. Instead of using a vanilla transformer, a strided transformer is employed. The main difference between them is that strided convolutions replace perceptron [21]. This solution proposed by Shan *et al.* [31] is the most accurate in terms of MPJPE, but it has some limitations when combined with the HAR part, which will be discussed in the Merging Networks section.

The model is trained in two stages following the procedure of Shan *et al.* [31]. Firstly, the pre-training of the encoder and the first transformer, and then the fine-tuning of the entire 3D pose model is conducted.

4.2.1 Training Stages

During the pre-training stage, temporal and spatial masking techniques were used [31]. These techniques involve masking 70% of the joints and frames for the encoder and temporal transformer to learn to reproduce the masked inputs. This approach was adapted because training the model directly resulted in poorer scores and longer convergence times. Once the pre-training was completed, the weights of the epoch with the best scores were saved, and the model was fine-tuned.

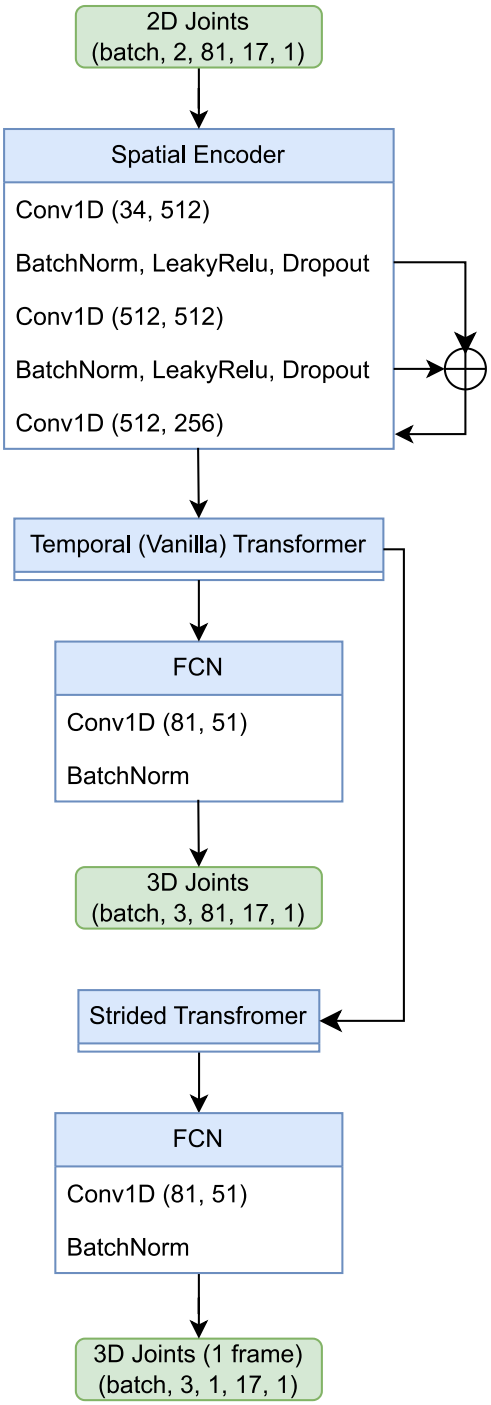

Figure 4-1: Architecture of Pose Estimation Model

4.2.2 Normalized MPJPE

Upon reviewing the literature, it was observed that the mean per joint position error (MPJPE) is the standard evaluation metric used by most researchers for assessing the

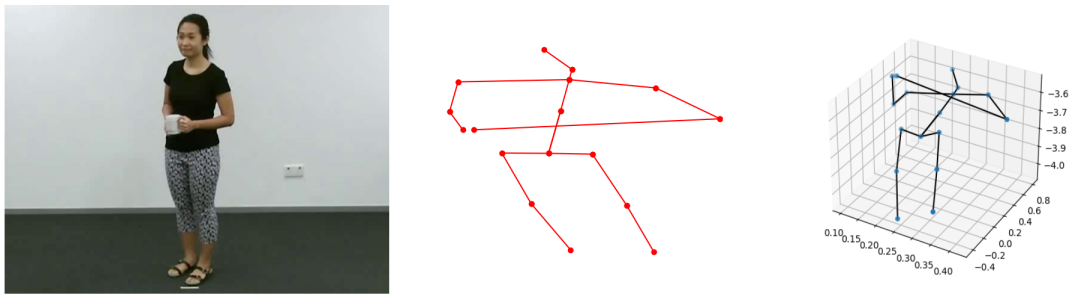

Figure 4-2: One frame of video with corresponding 2D projection and 3D coordinates of NTU RGB+D dataset [30]

performance of 3D pose estimation models. MPJPE is calculated using the following formula for a single frame:

$$E_{MPJPE}(f, S) = \frac{1}{N} \sum_{1}^{N} ||pred_{f,S}(i) - target_{f,S}(i)||_2 \tag{4.1}$$

At the end, the average is computed for all frames, using the formula where f represents the frame number and N denotes the number of frames in the skeleton S.

The loss function formula is utilized in both stages of training, with the distinction being that pre-training employs 2D coordinates for prediction and target.

A research paper on motion capture [28] proposed the use of normalized MPJPE due to scale ambiguity for monocular reconstruction. Scale ambiguity is characteristic of different 3D objects having 2D projections of similar sizes. In the thesis, this ambiguity is expressed as the reconstruction of different 3D coordinates from the same 2D inputs.

The target is normalized over the joint axis, preserving the overall shape of the skeleton. This sets it apart from a regular error.

The Experiments and Results section will detail the practical need to normalize the skeleton data.

4.3 Joints

In the context of 3D pose estimation, the MPI-INF-3DHP dataset comprises a single sample that includes a frame featuring 17 joint coordinates. Each joint is defined by

its corresponding 3D coordinates (x, y, z). Meanwhile, the NTU RGB+D dataset also follows a comparable structure, but with 25 joints that are indexed differently, as depicted in Figure 4-3.

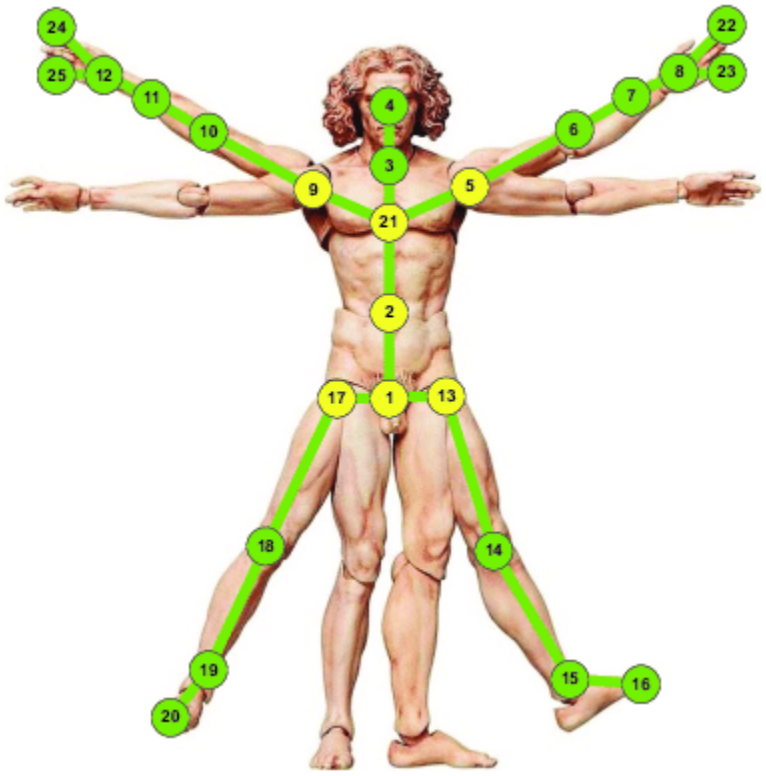

Figure 4-3: Visualization of skeleton data of NTU RGB+D dataset [22]

4.4 Bones

In each dataset, there exists a distinctive collection of joint pairs and their respective "parent" nodes that create a bone structure. While this information is exclusively employed for human activity recognition (HAR), it is disregarded in pose estimation. The use of bones is not novel, and combining bone and joint data streams can enhance the accuracy of HAR, as indicated in [6]. In this thesis, bones are represented by vectors that are derived by subtracting the coordinates of one joint from its corresponding paired joint.

4.5 Graph Convolution Network

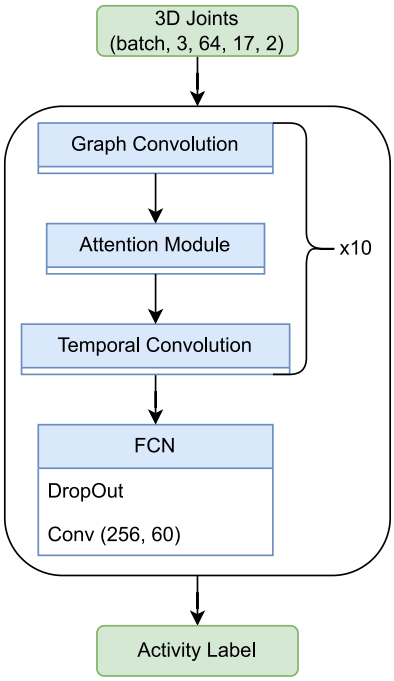

Figure 4-4: Architecture of HAR Model

Layers	In channels	Out channels
1	3	64
2-4	64	64
5	64	128
6-7	128	128
8	128	256
9-10	256	256

Figure 4-5: Number of channels for repeated layers in HAR architecture

To tackle the HAR aspect of the project, a graph convolution network was deemed suitable. Following the approach proposed by state-of-the-art models such as [20, 9], the same method was adopted. The primary principle is akin to that of the pose estimation task: the graph convolution network is trained on temporal features (graph), the temporal convolution network is trained on temporal features (frames), and an attention module is employed, much like a transformer, to identify the most crucial edge sets. The GCN utilized a set of four parallel convolution layers, whose results were then fed into the attention module. The TCN, on the other hand, comprised four 2D convolution layers and two temporal convolution layers.

4.5.1 Graph

The HD-Graph approach [20] was implemented to generate a graph structure. The skeleton was decomposed using a hierarchy set of pre-defined edges. Due to modifications made to the dataset, these edge sets were adjusted accordingly. Despite some

joints being excluded (e.g., fingers, toes), resulting in less information for edge sets, experiments demonstrated that accuracy did not significantly decrease. This outcome could be explained by the fact that the NTU-RGB+D 60 dataset comprises actions that are not heavily reliant on these specific joints.

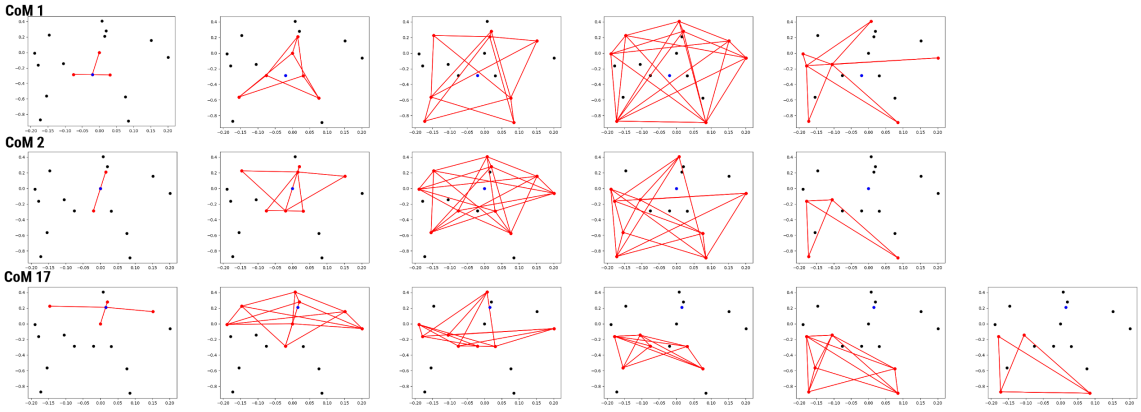

Figure 4-6: Visualization of edge sets for each center of mass

4.5.2 Ensemble method

As outlined in [20], configurations that incorporate both joints and bones result in inferior outcomes, thus it is reasonable to exclude them. Instead, the proposed approach involves ensembling solely joint and bone streams, but for three distinct centers of mass, using equal contributions represented by coefficients [0.5, 0.5, 0.5, 0.5, 0.5, 0.5]. These centers of mass correspond to joints 1, 2, and 21, as illustrated in Figure 4-3. Each center of mass employs a unique hierarchy set of edge sets, which can be observed in Figure 4-6. The experiments revealed that individual data streams perform similarly, while ensembling significantly enhances the final accuracy.

Chapter 5

Experiments and Results

5.1 Datasets

Several experiments were conducted using popular datasets for 3D pose estimation and activity recognition. Initially, two datasets were planned for 3D pose estimation: MPI-INF-3DHP and Human3.6M, as they are both large and widely used benchmarks in state-of-the-art works [46, 31, 14]. However, due to Human3.6M not being publicly available, only MPI-INF-3DHP was used. For activity recognition, the NTU-RGB+D 60 dataset was chosen due to its size and potential for future expansion to the NTU-RGB+D 120 dataset (same format, twice the data).

The NTU-RGB+D dataset [30] consists of 56880 videos of 60 action classes, with each video having 30 frames per second. The dataset provides 3D skeletons, their 2D projections, and labels. The skeleton is composed of 25 joints of the human body captured by Microsoft Kinect V2 cameras. Some activities involve 2 people, so each sample of the dataset contains skeleton data for 2 persons, with the second person's data being filled with zeros if not involved. To address the issue of incomplete skeleton data, the input is checked before passing it to 3D pose estimation.

MPI-INF-3DHP [26] contains more than a million frames captured by cameras from 14 views. Each sample provides 2D and corresponding 3D coordinates of 17 body joints. The dataset was divided into training and testing sets, with 1054462 and 24891 frames, respectively. To enable comparison between the two datasets, the

number of joints in the NTU-RGB+D dataset was reduced, resulting in the joint configuration shown in Figure 5-1.

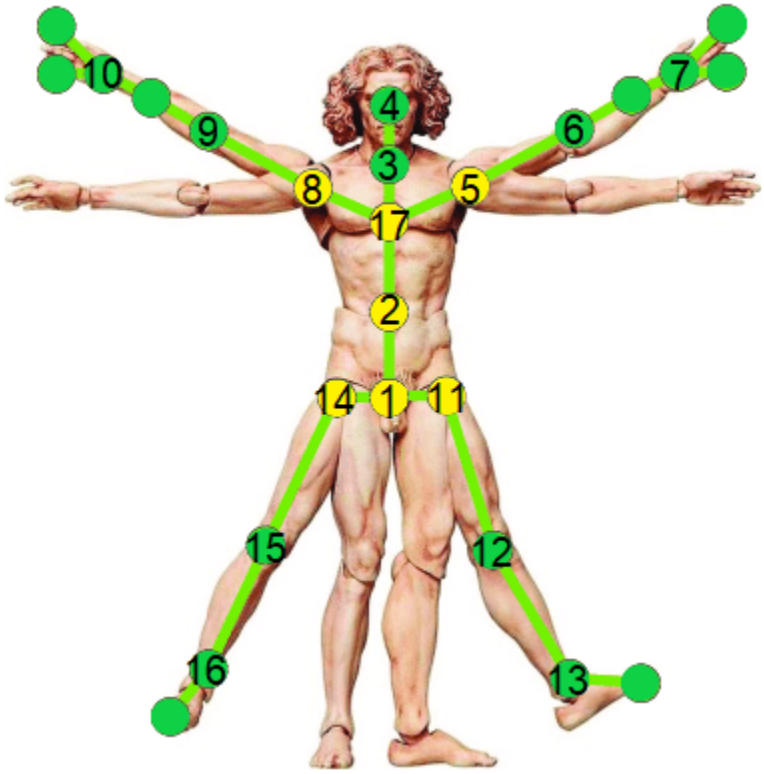

Figure 5-1: Visualization of modified skeleton data

For the proof of concept, only a cross-subject benchmark was used, meaning that the training set of 40320 samples did not include people from the testing set of 16560 samples. As is typical with this dataset, cross-subject accuracy scores are lower than cross-view scores [46, 31].

5.2 Pose Estimation Features

In 3D pose estimation for the MPI-INF-3DHP dataset, each batch contains 81 consecutive 2D (x,y) coordinates for 17 joints and corresponding 3D coordinates for one frame, which serves as the target. When switching to another dataset, the data was permuted to match the joint order shown in Figure 5-1. For the NTU RGB+D dataset, 2D projections are used as input, with the central frame (number 40 out of

81) serving as the target.

Pre-train stage on MPI dataset	15
Fine-tune stage on MPI dataset	30
Train on MPI, test on NTU dataset	200
Train on NTU, test on NTU dataset	0.1
Train on MPI and NTU, test on NTU dataset	0.1

Table 5.1: MPJPE scores of 3D pose estimation

In the first strategy, the model was trained on the MPI-INF-3DHP dataset in both stages. This was based on the rationale that the dataset was designed specifically for this task, is larger (1 million vs 50 thousand frames), and covers more cases. As reported in Table 5.1, this strategy performed well on the MPI-INF-3DHP dataset with a state-of-the-art score of 30mm, but its performance on NTU RGB+D was poor. A preliminary conclusion was reached that MPI-INF-3DHP does not cover all cases.

Training on NTU RGB+D and both datasets yielded suspiciously low results with a 100 magnitude lower than the state-of-the-art. To investigate this, normalized MPJPE was used. The explanation is that the magnitude of values in the NTU RGB+D dataset is smaller, which resulted in lower scores. However, upon examining the output, the shapes of predicted skeletons were observed to be too far from the target. Normalized MPJPE was then employed, revealing that the score for the original model was 0.05mm against 0.1mm, i.e., 10 times lower.

The reason why training on NTU RGB+D did not work is attributed to transformers needing large datasets and performing better when pre-trained [17]. The unsuccessful training on both datasets implies that there is a difference in the data, which prevents generalization on both sets.

5.3 Pre-processing

To investigate the data, several steps were taken to identify any differences. First, a visual inspection of the skeletons did not reveal anything noteworthy. However,

an analysis of the numeric values of the coordinates revealed that their magnitudes varied significantly. Additionally, the authors of MPI-INF-3DHP set the coordinates of joint 1 (as shown in Figure 5-1) to 0 [31].

To address these differences, the decision was made to centralize the data by subtracting the coordinates of joint 1 from every joint, effectively placing joint 1 at the origin (0,0,0). The data was then normalized along the joint axis, which preserved the shape of the skeletons and set both datasets to the same scale. A similar procedure was employed in some previous works [10, 38]. The subsequent steps included pre-training on MPI-INF-3DHP, pre-training on NTU RGB+D, fine-tuning on MPI-INF-3DHP, and fine-tuning on NTU RGB+D. The results are presented in Table 5.2.

Pre-train stage on MPI dataset	15
Pre-train stage on NTU dataset	14
Fine-tune stage on MPI dataset	0.05
Fine-tune stage on NTU dataset	0.06

Table 5.2: Normalized MPJPE scores of 3D pose estimation after pre-processing

The final training results are almost identical to the original performance, indicating a close match. Additionally, training with normalized data resulted in higher normalized MPJPE scores compared to normalizing the results and taking MPJPE scores.

5.4 HAR Features

For HAR, only the NTU RGB+D dataset was used as the MPI-INF-3DHP dataset did not have any activity labels. As with the previous section, the input comprised the (x,y,z) coordinates of 17 joints across consecutive frames. The number of frames varied between 64, 81, and 144, depending on the task. Table 5.4 shows the difference in accuracy scores.

The experimental results demonstrate that a slight 0.2% decline in accuracy was observed when the number of joints was reduced. This loss in accuracy was subsequently recovered by increasing the frame window. Nevertheless, it was important to

25 joints and window size 64	90%
17 joints and window size 64	89%
17 joints and window size 81	90%
2D 17 joints and window size 64	85.16%

Table 5.3: Accuracy scores of HAR for joints stream, the center of mass at joint 1

note that the pose estimation depended on 81 frames, and therefore the initial plan was to combine the models using these 81 frames. Additionally, while incorporating the 3D pose module, the accuracy score for 2D joint input was assessed, exposing a reduced accuracy of 85% due to the absence of z-dimension information.

5.5 End2End Data Processing

Regarding End2End Data Processing, the initial approach involved using the predictions for 81 2D to 3D conversions before combining them into a single frame. However, the model only achieved a 75% accuracy due to output and individual predictions being far from the desired target. Consequently, the window size of frames was increased to 144. As discussed in the pose estimation section, predictions are generated for the central frame with an index of 40 out of 81. To obtain 64 frames for activity recognition, each frame requires 39 frames preceding it and 41 frames following it, allowing for pose estimation from 2D to 3D for each of the central 64 frames.

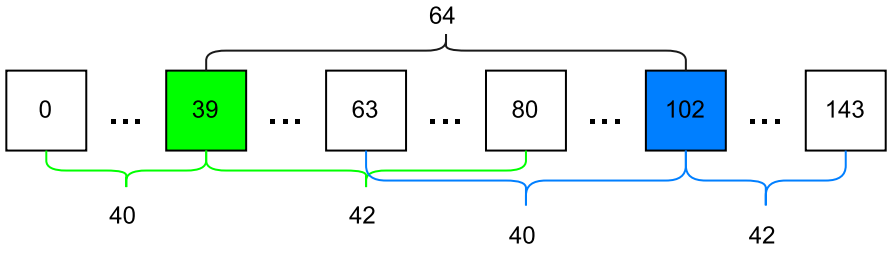

Figure 5-2: The method to get frames for HAR

This approach has the disadvantage of having fewer frames available for each activity label. However, the advantage is that there is no need to retrain individual models with a new flow. As the subsequent frames are similar, the pose estimation

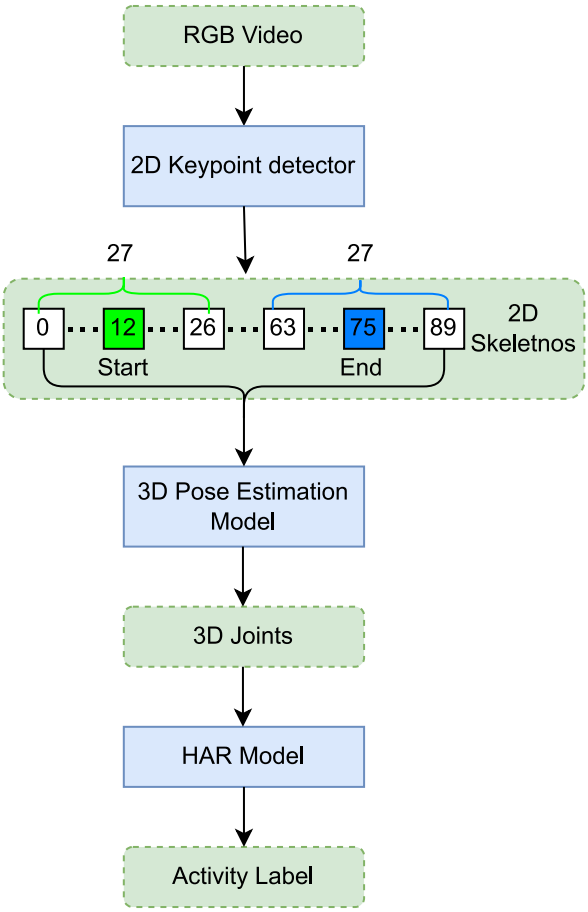

Figure 5-3: The whole process of 2D joints to activity recognition, including keypoints detection

model trained as usual achieves a similar MPJPE score on the test set generated in this manner, with a difference of only 0.0006 mm between scores of 0.0446 mm and 0.044 mm.

Despite its high accuracy, the activity prediction model could not train effectively, reaching only 70% due to a significant loss of frames (55%). To address this issue, the 3D pose estimation model was retrained to function on 27 frames. To account for the smaller frame size, the first transformer's number of layers was increased from 3 to 4. The resulting MPJPE score for this retrained model is 0.0607 mm, as depicted in figure 5-3.

5.6 Ensembling

In order to implement 6-way ensembling, the model was trained using 6 distinct configurations incorporating joints and bones information for 3 different centers of mass. The outcomes for each individual configuration as well as the ensembling results are displayed in table 5.5. As previously discussed, the data was pre-processed in the same manner as pose estimation.

Joint stream, the center of mass at joint 1	89.09%
Joint stream, the center of mass at joint 2	89.29%
Joint stream, the center of mass at joint 17	88.45%
Bone stream, the center of mass at joint 1	88.08%
Bone stream, the center of mass at joint 2	87.75%
Bone stream, the center of mass at joint 17	87.61%
6-way ensemble	92.9%

Table 5.4: Accuracy scores of HAR for 17 joints, 64 frames on generated 3D data

As previously mentioned, due to the frame reduction and 3D conversion, the final scores are on average 2% lower. Nevertheless, the performance after ensembling surpasses the results reported in a paper proposing an alternative approach for 2D projections on the NTU-RGB+D dataset [10].

Model	Input	Accuracy
Center of mass 1	2D joint stream	87.25%
Center of mass 2	2D joint stream	87.14%
Center of mass 17	2D joint stream	86.94%
Center of mass 1	2D bone stream	86.49%
Center of mass 2	2D bone stream	85.90%
Center of mass 17	2D bone stream	85.07%
Pose2Act (6-way ensemble)	2D projection	90.3%
MS-G3D [10, 23]	2D projection	86.8%
HD-GCN [20]	2D projection	87.1%
PoseConv3D [10]	2D projection	89.2%
DGNN [33]	3D skeleton	89.9%
MS-OP-AGCN [38]	3D skeleton	91.12%
HD-GCN [20]	3D skeleton	93.4%

Table 5.5: Final accuracy scores

Table 5.5 shows that state-of-the-art models (MS-OP-AGCN, HD-GCN) achieve the highest results on generated 3D-skeleton data. In this case, the proposed solution can only outperform the older model (DGNN). However, this is also true for any model predicting activity from 2D projections. For the same most accurate model (HD-GCN), the difference between 2D and 3D input is 6.3%, while the difference with the proposed method is only 3.1%.

Using a model that performs well only on 3D input data is less practical and has limited applications because obtaining 3D data requires special sensors like the Microsoft Kinect v2 cameras used in the NTU-RGB+D dataset. In contrast, obtaining 2D data only requires an RGB camera and a 2D pose estimation model. Additionally, the study found that using predicted data could improve HAR results compared to sensors on average by 1%.

From the confusion matrix shown in Figure 5-4, it is apparent that certain activ-

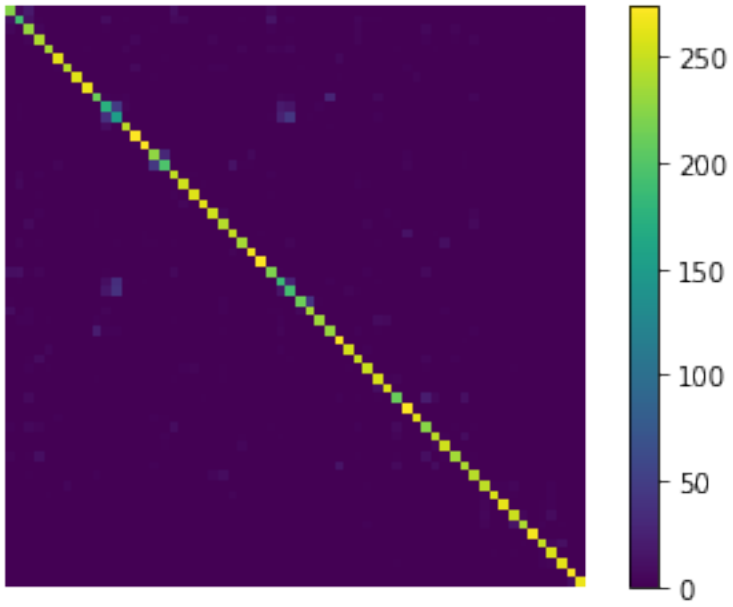

Figure 5-4: Confusion matrix for Pose2Act predictions

ities have lower results. Specifically, reading, writing, playing with the phone, and typing on a keyboard have the lowest scores. Since all of these actions involve hands, it is likely that the lower scores are due to the exclusion of finger joints. Including finger data in the dataset may lead to an improvement in overall accuracy. However, it is also possible that these activities are more complex, and models in general struggle to learn them.

5.7 Discussion

In Table 5.6, various models are compared, and it is evident that the proposed model performs exceptionally well in predicting both 3D input data and 2D data. While models that utilize generated 3D data achieve the highest scores, their practicality is limited. Furthermore, the effectiveness of GCN-based solutions depends heavily on the ensembling technique used. To address this issue, our future plans include comparing models based on their lightweight properties, such as the number of parameters and FLOPs.

	Pose2Act	MS-G3D [10, 23]	HD-GCN 2D [20]	PoseC-onv 3D [10]	HD-GCN 3D [20]	DGNN [33]	MS-OP-AGCN [38]
Year	2023	2020	2022	2022	2022	2019	2021
Produced 3D data	predicted	2D only	2D only	pre-dicted	generated	gener-ated	generated
Ensembling	6-way	2-way	6-way	2-way	6-way	2-way	5-way
Data Streams	joint and bone for 3 CoM	joint, bone	joint, and bone for 3 CoM	joint, limb	joint and bone for 3 CoM	joint, bone	joint, bone, joint and bone motion, joint orientation
Accuracy	90.3%	86.8%	87.1%	89.2%	93.4%	89.9%	91.12%
Parameters	7.0M	-	-	-	-	-	-
FLOPs	1.7G	-	-	-	-	-	-
Methods	Transform-ers, GCN, TCN	GCN, TCN	GCN, TCN	3D CNN	GCN, TCN	GCN, TCN	GCN, CNN
Importance of edges	Attention module	Multi-Scale Aggrega-tion	Attention module	Atten-tion module	Attention module	Adap-tive graph	Adaptive graph
Integration with 2D pose estimator	yes	no	no	yes	no	no	no
Real-life application	full	full	full	full	limited	limited	limited

Table 5.6: Comparison of different models

There was a concern that using convolution block as spatial encoder as shown in figure 4-1 could have a negative impact on performance. There was an attempt to replace this part with additional transformer. As a result, the inference time increased several times, while the accuracy score only decreased. The results of training are displayed in the table 5.7. In comparison with original score of 32.2mm, the error increased more than 2 times. Furthermore, additional study is required.

Stages	Dataset	MPJPE score (mm)
Stage I	MPI INF	15.34
Stage I	NTU RGB+D	41.82
Stage II	MPI INF	67.91

Table 5.7: The results of modified pose estimation model with transformer as spatial encoder

It is worth noting that modifying datasets, such as normalization and centering, leads to a longer convergence time for the HAR model. Achieving an accuracy of 89% percent takes 180 epochs instead of 90 epochs. This may be due to the fact that after normalization, the scale of coordinates is smaller, making it more challenging for

the model to capture the changes in each frame. Additionally, further investigation is required, and more data augmentation may be beneficial, as some overfitting is observed. At some point, the training accuracy reaches 100% for all configurations.

The experiments showed that the performance of the pose estimation model on a new dataset is low, requiring further training. However, after retraining on a new dataset, the performance of the original data decreases. This issue can partially be addressed by proper fine-tuning, as described in previous works [19]. However, this drawback suggests that the transformer's performance in the 3D pose estimation task depends heavily on the dataset and may not work as well in real-life conditions.

Normalization of poses another problem for real-life applications, as the model becomes less scale invariant. Additionally, to visualize pose estimation results, reconstruction to the original scale is necessary. This issue can be resolved in the future by applying a larger scaling factor in data augmentation after normalization and keeping the original scale coefficients to restore proper sizes.

The pose estimation is a bottleneck in the End2End flow since only one frame out of a sequence is predicted. As a result, the number of frames required to predict one activity increases from 64 to 144 (depending on window size), from 2.13 seconds of video to 4.8 seconds. While using state-of-the-art models is reasonable for research purposes, it would be more advantageous to use a smaller window size of frames (9) or a model that can accurately predict 3D coordinates for individual frames, such as a sequence-to-sequence approach instead of a sequence-to-frame approach, for real-life applications.

Finally, the model's use in real-time applications is still limited due to the ensembling method used. Each configuration requires a separately trained model, resulting in six models in a six-way ensembling. As a result, the GPU must run six different models, which requires significant VRAM. Every solution is highly dependent on ensembling, as shown in recent research. Additionally, incorporating a 2D pose estimation model to predict 2D coordinates is necessary, but these models are larger and require even more VRAM. Thus, high-end GPUs are required to use such solutions.

In future tasks, it would be beneficial to consider using predicted 2D key points

instead of relying solely on sensor readings. To achieve this, one potential approach is to utilize a 2D pose estimation model, which can accurately estimate the positions of key points in a given image. Previous research has shown that using predicted data results in higher accuracy scores, as demonstrated in studies such as [10, 9]. The whole plan for the future works can be summarized by the figure **??**.

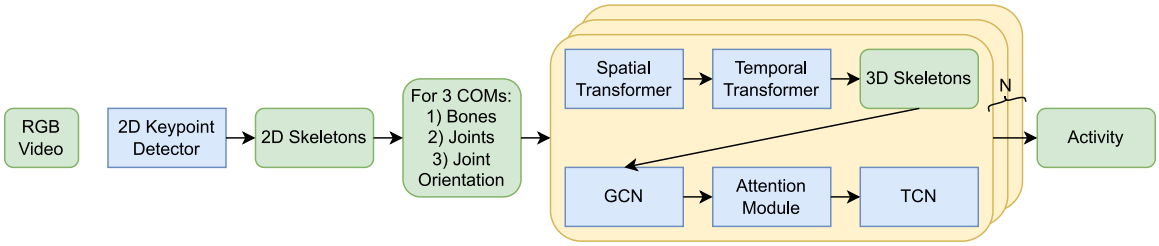

Figure 5-5: The plan for the future work

Furthermore, it may be advantageous to prioritize the development of a more robust model that is capable of operating on a smaller number of frames. By focusing on a heavier model with stronger predictive power, it may be possible to achieve higher accuracy scores and more precise predictions with a reduced computational workload. This approach could help to streamline the overall process and reduce the resources required for future tasks.

Chapter 6

Conclusion

The objective of this thesis is to expand the scope of skeleton-based human activity recognition models by incorporating 3D pose estimation. The proposed approach outperformed activity recognition on 2D projections of the NTU RGB+D dataset by utilizing a hierarchically decomposed graph, a graph convolution network, and two transformers for the 3D pose estimation task. The model was trained using joints and bones information for three different centers of mass, and the results were combined using ensembling. However, the use of generated 3D data in other HAR solutions still surpasses the proposed model due to information loss. Nonetheless, the proposed model marks a significant milestone in computer vision and has the potential to enhance the performance of various applications that rely on 2D input data.

Future work may include integrating a 2D pose estimation component, improving scaling invariance, enhancing the sequence-to-sequence 3D conversion, testing the method on different datasets, and refining the ensembling of data streams. The proposed model can be easily applied to datasets of various sizes and types without significant changes to its architecture. Additionally, it can be customized to suit different tasks and domains, such as pose estimation, action recognition, and activity analysis.

Bibliography

[1] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[2] Marius Bock, Michael Moeller, Kristof Van Laerhoven, and Hilde Kuehne. Wear: A multimodal dataset for wearable and egocentric video activity recognition. *arXiv preprint arXiv:2304.05088*, 2023.

[3] Damien Bouchabou, Sao Mai Nguyen, Christophe Lohr, Benoit LeDuc, and Ioannis Kanellos. A survey of human activity recognition in smart homes based on iot sensors algorithms: Taxonomies, challenges, and opportunities with deep learning. *Sensors*, 21(18):6037, 2021.

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[5] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368, 2021.

[6] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 536–553. Springer, 2020.

[7] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20186–20196, 2022.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Dg-stgcn: Dynamic spatial-temporal modeling for skeleton-based action recognition. *arXiv preprint arXiv:2210.05895*, 2022.

[10] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978, 2022.

[11] Mariem Gnouma, Ammar Ladjailia, Ridha Ejbali, and Mourad Zaied. Stacked sparse autoencoder and history of binary motion image for human activity recognition. *Multimedia Tools and Applications*, 78(2):2157–2179, 2019.

[12] Mohammed Hassanin, Abdelwahed Khamiss, Mohammed Bennamoun, Farid Boussaid, and Ibrahim Radwan. Crossformer: Cross spatio-temporal transformer for 3d human pose estimation. *arXiv preprint arXiv:2203.13387*, 2022.

[13] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7779–7788, 2020.

[14] Wenbo Hu, Changgong Zhang, Fangneng Zhan, Lei Zhang, and Tien-Tsin Wong. Conditional directed graph convolution for 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 602–611, 2021.

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[17] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[18] 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.

[19] Jaejun Lee, Raphael Tang, and Jimmy Lin. What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*, 2019.

[20] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoon Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:2208.10741*, 2022.

[21] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*, 2022.

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

[23] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152, 2020.

[24] Diogo C Luvizon, David Picard, and Hedi Tabia. Multi-task deep learning for real-time 3d human pose estimation and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2752–2764, 2020.

[25] Haoyu Ma, Liangjian Chen, Deying Kong, Zhe Wang, Xingwei Liu, Hao Tang, Xiangyi Yan, Yusheng Xie, Shih-Yao Lin, and Xiaohui Xie. Transfusion: Cross-view fusion with transformer for 3d human pose estimation. *arXiv preprint arXiv:2110.09554*, 2021.

[26] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017.

[27] Ohoud Nafea, Wadood Abdul, Ghulam Muhammad, and Mansour Alsulaiman. Sensor-based human activity recognition with spatio-temporal deep learning. *Sensors*, 21(6):2141, 2021.

[28] Mirela Ostrek, Helge Rhodin, Pascal Fua, Erich Müller, and Jörg Spörri. Are existing monocular computer vision-based 3d motion capture approaches ready for deployment? a methodological study on the example of alpine skiing. *Sensors*, 19(19):4323, 2019.

[29] Sen Qiu, Hongkai Zhao, Nan Jiang, Zhelong Wang, Long Liu, Yi An, Hongyu Zhao, Xin Miao, Ruichen Liu, and Giancarlo Fortino. Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges. *Information Fusion*, 80:241–265, 2022.

[30] 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.

[31] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. In *Computer Vision–ECCV 2022: 17th European Confer-*

ence, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V, pages 461–478. Springer, 2022.

[32] Vijeta Sharma, Manjari Gupta, Anil Kumar Pandey, Deepti Mishra, and Ajai Kumar. A review of deep learning-based human activity recognition on benchmark video datasets. *Applied Artificial Intelligence*, 36(1):2093705, 2022.

[33] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019.

[34] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017.

[35] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[36] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[37] Arslan Syed, Eman A Aldhahri, Muhammad Munawar Iqbal, Abid Ali, Ammar Muthanna, Harun Jamil, and Faisal Jamil. Intelligent 3d network protocol for multimedia data classification using deep learning. *arXiv preprint arXiv:2207.11504*, 2022.

[38] Chingizkhan Tangirbergenov, Adnan Yazici, and Enver Ever. Multi-stream orientation and position based adaptive graph convolutional network for skeleton based activity recognition. 2021.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[41] Yuhao Wang, Hongji Xu, Lina Zheng, Guozhen Zhao, Zhi Liu, Shuang Zhou, Mengmeng Wang, and Jie Xu. A multi-dimensional parallel convolutional connected network based on multi-source and multi-modal sensor data for human activity recognition. *IEEE Internet of Things Journal*, 2023.

[42] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[43] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[44] Xinwei Yu. (fusionformer): Exploiting the joint motion synergy with fusion network based on transformer for 3d human pose estimation. *arXiv preprint arXiv:2210.04006*, 2022.

[45] Ailing Zeng, Xiao Sun, Lei Yang, Nanxuan Zhao, Minhao Liu, and Qiang Xu. Learning skeletal graph neural networks for hard 3d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11436–11445, 2021.

[46] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13232–13242, 2022.

[47] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11656–11665, October 2021.