Deep Learning Methods for Video-Based Human Activity Recognition in Industrial Settings

Behnoosh Parsa

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Ashis G. Banerjee, Chair

Steven L. Brunton

Santosh Devasia

Sawyer B. Fuller

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Deep Learning Methods for Video-Based Human Activity Recognition in Industrial Settings

Behnoosh Parsa

Chair of the Supervisory Committee:
Assistant Professor Ashis G. Banerjee
Mechanical Engineering

With increasingly high interest in assistive robots and smart surveillance systems, we need a powerful perception mechanism to be able to describe the events in a scene. However, achieving accurate perception models is not trivial, since, even for one perception task there are unlimited possible scenarios. Hoping to develop analytically driven models seems too optimistic for such systems; hence, *Supervised Learning* as a sub-field of function approximation has become very popular in robotic perception. Supervised learning is the task of learning a function that maps an input to an output based on example input-output pairs.

Scene understanding is even more involved when it comes to solving Human Action Recognition (HAR) problems. In HAR the task is to classify human activities from an image or determine atomic actions composing the activity in a video. In video-based HAR, there are exponentially many ways that humans can perform the same task. Besides, the variety in posture and speed at which people perform activities makes solving HAR tasks even more challenging. Therefore, models should be designed to learn common underlying spatial and temporal properties of human activity to achieve generalizability.

This thesis is dedicated to designing perception models for recognizing human actions and determining the ergonomic risk associated with them. Specifically, Part I focus on solving the Human Activity Segmentation (HAS) problem in long videos, which is the task of semantically segmenting long videos into distinct actions in an offline framework. In Part II, we present our designs

for solving online-HAR problems to recognize human activities in the observed batch of frames. Since, the performance of computer vision algorithms also depends on the quality and relevance of the training data, in Part I, we introduce a new dataset for an indoor object manipulation task called the University of Washington Indoor Object Manipulation (UW-IOM).

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AI: Artificial Intelligence

AS: Action Segmentation

AQA: Action Quality Assessment

C3D: 3D Convolutional Neural Networks

CNN: Convolutional Neural Networks

EAWS: European Assembly Worksheet

ED-TCN: Encoder-Decoder Temporal Convolutional Network

ENCODER: maps each node of a graph to a low dimensional vector (ENC(v))

ERA: Ergonomics Risk Assessment

FC: Fully Connected

GC: Graph Convolution

GNN: Graph Neural Networks

GGNN: Gated Graph Neural Network

GRU: Gated Recurrent Unit

HOI: Human Object Interaction

HRC: Human Robot Collaboration

HAR: Human Action Recognition

HAE: Human Activity Evaluation

HOI-GCN: Human Object Interaction Graph Convolutional Network

HBS-GCN: Human Body Structure Graph Convolutional Network

HPA: Human Postural Assessment

HAS: Human Activity Segmentation

HAR: Human Action Recognition

IMAGENET: A dataset that offers tens of millions of cleanly sorted images for various concepts

LSTM: Long-Short-Term-Memory

MPNN: Message Passing Neural Network

MTL: Multi-Task Learning

NLP: Natural Language Processing

NN: Neural Network

P-GCN: Pyramidal Graph Convolutional Network

PA: Postural Assessment

POSECNN: A deep architecture that represents very good feature for images including humans

RBF: Radial Basis Function

REBA: Rapid Entire Body Assessment

RNN: Recurrent Neural Network

RELU: Rectified Linear Unit (ReLU)

SIMILARITY FUNCTION: specifies how relationships in vector space map to relationships in the original network ($similarity(u, v)$)

SLA: Static Learning Algorithm

STL: Single-Task Learning

ST-HGCN: Spatio-Temporal Hierarchical Graph Convolutional Network

ST-GCN: Spatio-Temporal Graph Convolutional Network

ST-PGN: Spatio-Temporal Pyramid Graph Convolutional Network

TCN: Temporal Convolutional Network

TUM: Technical University of Munich

UW-IOM: University of Washington Indoor Object Manipulation

VGG16: A powerful deep architecture that in this thesis is pre-trained on ImageNet dataset

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor, Prof. *Ashis G. Banerjee* for the continuous support of my Ph.D. and research, for his patience, motivation, and allowing me to grow in multiple dimensions and expand knowledge beyond my Ph.D. research. I also appreciate Amazon Robotics as well as the Washington State Department of Labor and Industry for supporting my research.

I would also like to thank the rest of my Ph.D. committee members, Profs. *Steven Brunton*, *Santosh Devasia*, *Sawyer Fuller*, and *Mehran Mesbahi* for their insightful comments and encouragement, which incented me to widen my research from various perspectives. In addition, I would like to thank *Wanwisa Kisalang* the Mechanical Engineering Graduate Student Advisor for her kind support and the joy that she brings to the department.

I would like to thank all my collaborators at the Amplifying Movement and Performance Laboratory at the University of Washington, specifically Profs. *Sam Burden* and *Lillian Ratliff*. Many thanks to Dr. *Behzad Dariush* at Honda Research Institute (HRI-USA) and Dr. *Franziska Meier* at Facebook AI Research (FAIR) for their guidance during my internships.

I should thank the University of Washington (UW) for providing diverse resources for students and acknowledging students who leverage from the available opportunities to grow different dimensions of their personalities. I was honored to be selected as one of the Husky 100 cohort 2020 for leveraging the academic, entrepreneurship, and mindfulness resources as well as my efforts toward improving the experience of students on campus. And learning that UW recognizes my dedication to academic- as well as self-development was the most heartwarming moment of my Ph.D.

For the past two years, I have been taking Yoga classes at UW Recreation to help myself keep

alive the spiritual aspect of my life. This program has made me physically and mentally healthier. I truly appreciate *Danny Arguetty*'s hard work for creating the philosophy behind this program and recruiting exceptional instructors. All the good experience and the joy of sharing it with others encouraged me to enroll in the "Yoga Teacher Training" program at UW Recreation and now I am a Yoga instructor teaching at UW Recreation, which would not be possible without the support of the UW community.

In addition, I would like to thank my family: my dear parents, *Parvaneh* and *Javid*, and my sweet little sister *Niusha* for their unconditional love and support throughout my life. Last but not least, a special thanks to the love of my life *Siavash* who has always seen the best in me and encouraged me in any endeavor.

DEDICATION

to the light of my life,

to my dear mom, Parvaneh

Chapter 1

INTRODUCTION

1.1 Motivation

In toady's fast growing technology market, Artificial Intelligence (AI) applications are rapidly increasing, and among them HAR has received great interest. Applications of HAR are numerous and vary from anomaly detection in surveillance systems, self-driving cars, sport analytics, human robot interaction, and many more. In designing autonomous systems interacting with humans, the accuracy of the the systems' perception of human activities become even more crucial in order to insure safety.

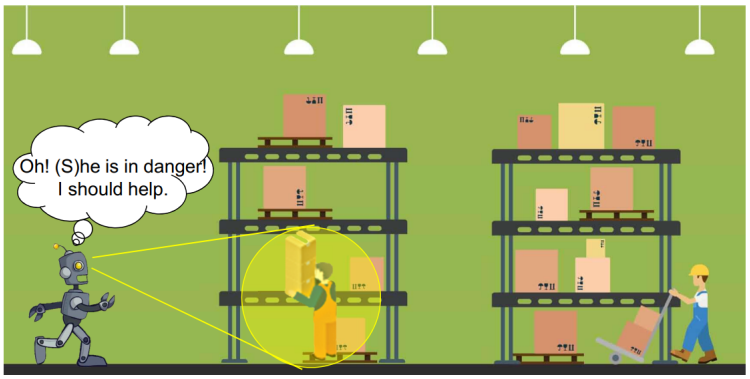


Figure 1.1: The scenario of a robot recognizing that a human operator is in danger of injuring him/herself and going to offer assistance.

Still there is a long journey to realize the dream of having robots helping humans out in difficult tasks. For a robot to be trusted to work near humans it has to be safe. These robots must be capable of understanding humans' behaviour/actions to perceive their needs. In addition, they must be able to respond/act accordingly. This thesis is motivated by the first requirement, and tries to explore the challenges regarding robot perceptions of human actions from video data.

1.2 Human Activity Recognition

Human Action Recognition is often associated with recognizing human activities in data that is coming from sensory data [18, 149]. In vision-based HAR the information is coming from video data. Human activity refers to the movement(s) of one(multiple) human body parts to accomplish a goal. In [144], human activities are broken down into 6 categories.

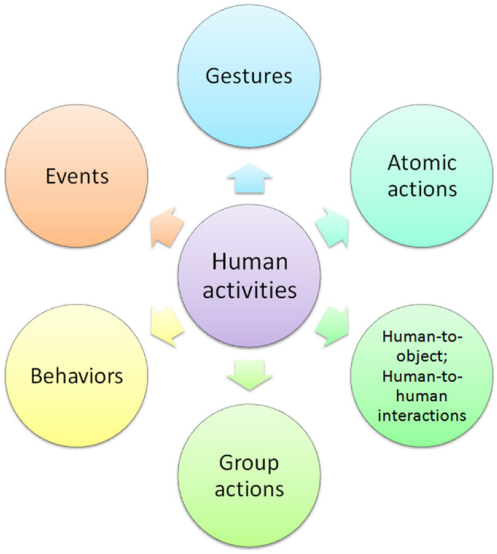


Figure 1.2: Decomposition of human activities. Picture is taken from [144].

Gestures are referred to primitive movements of the body parts of a person and may be a part of another action [157]. Group actions are activities performed by a group of people [141]. Human behaviors refer to movements of actions that are associated with specific emotions, personality, and psychological state mind [80]. Events are high-level activities describing social activities among a group of people [63]. Atomic actions are distinct movements/actions that comprise a more complex activity of a person [89]. Human-to-object or human-to-human interactions are human activities that involve other people or objects [102]. For any activity category of choice, HAR should be able to classify similar actions with same objective correctly regardless of the person or style.

As shown in Figure 1.3, HAR methods are categorized, based on the input modalities, in two main categories, Unimodal and Multimodal [144]. This thesis is focused on unimodal space-time

problems based on video data, which are also called vision-based HAR problems.

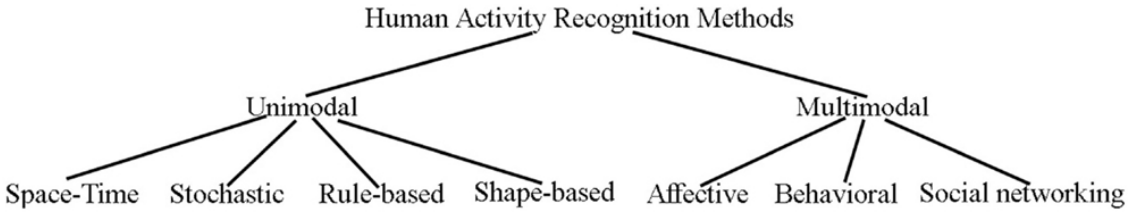


Figure 1.3: Hierarchical categorization of human activity recognition methods based on difference in data modalities. Uni-modal refers to methods using one source of information and multimodal methods use multiple sources of data. Picture is taken from [144].

.

Other ways of categorizing HAR methods is based on feature extraction process (Figure 1.4). Methods based on Handcrafted features rely on human insight and prior knowledge about data to extract discriminating features. Designing hand crafted features is time-consuming but it often results in accurate performance on the selected dataset. Recently, feature learning has started to become popular in vision-based HAR applications. Besides, many of the learning-based action recognition approaches rely on the end-to-end learning which consists of transformations from pixel-level to action classes. For more information refer to [9].

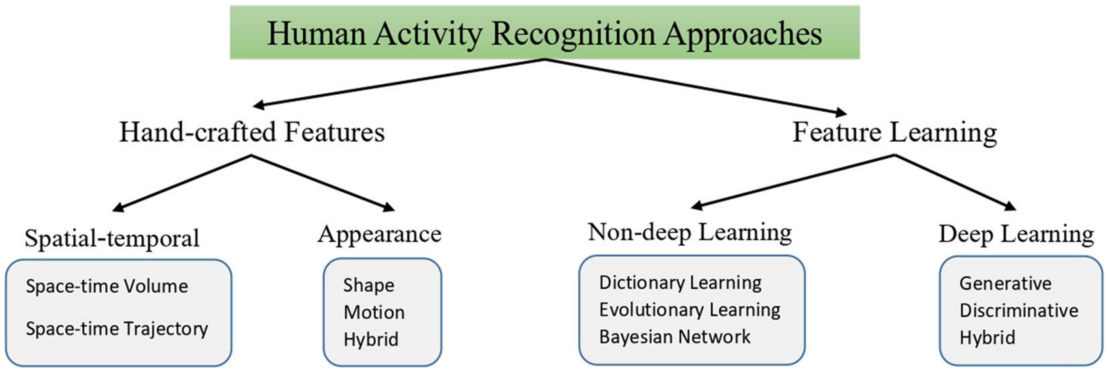


Figure 1.4: Vision-based Human activities recognition approaches. Picture is taken from [9].

.

Vision-based HAR has become an important topic in the computer vision community. It is involved in the development of many important applications such as virtual reality [122], human computer interaction (HCI) [63], security [125], video surveillance and home monitoring [5, 110, 113, 120, 121, 154]. A wide range of the activity recognition methods are directly linked to a specific application domain, hence, many efforts has been put to review the research trajectory in the field of vision-based HAR [9, 59, 113, 149, 161].

Many approaches in video-based HAR report state-of-the-art results for recognizing short-term actions in manually clipped videos. The main downside of these approaches is that they are not applicable in real world settings. The challenge is different when it comes to understanding daily tasks in long-term videos. Long videos contain several complex activities and these activities are difficult to model because each individual perform these task in their own way. Another challenge is that the starting and ending time of each task are overlapping and often result of many False Negatives or False Positives predictions, which is studied in [88].

1.3 Why Deep Learning

In solving HAR problems, the goal is to find a mapping from the robot's observations (video stream) to robot's perception (activity labels) of human actions, given a set of $\{$observations, correct perceptions$\}$. In mathematics, finding a mapping function is referred to as a function approximation problem.

Many often, regardless of the research domain, we face the problem of function approximation. In general, function approximation is to select a function among a well-defined class that closely approximates our observations. This is a general terminology and one can break it down into two major classes of one, where the target function is known and when it is unknown. The former has been treated by the numerical analysis which is a branch of the approximation theory [3], which is concerned with how functions can be approximated with simpler function and how to characterize the error properties.

In the second class, the explicit formula of the function is unknown and only a set of paired points are available. In this case depending on the domain (input) and codomain (output) of the

target function several techniques have been developed for approximation. For example, if the function operates on the real numbers, we can use techniques like regression analysis or curve fitting to find the input-output mapping. While if the codomain is a finite set we need to use classification techniques.

Consider the challenging task of robot scene understanding, specifically HAR. This task can be formulated as a problem of finding a function that maps video frames to a notion of action, which is a mapping from a matrix or a sequence of matrices to a label (categorical data). This is an extremely involved process because there are infinite features one can extract only from an image even without considering the sequential dependencies between those images. It was only after the introduction of Neural Networks (NN) that approximating functions operating on the infinite domain became possible. Especially, with the Convolutional Neural Networks (CNN) we can reduce the information of an image or a video and extract lower dimensional features.

1.4 Overview and Contributions of the Thesis

As mentioned earlier in this chapter, to achieve real-world application capability we need to develop algorithms that can tackle the challenges come with long-term videos. Therefore, in this thesis two important class of task in HAR are considered. First, the Human Activity Segmentation (HAS), and Second, early-/online-HAR problems.

Part I: Human Activity Segmentation

Temporal segmentation of human activities into atomic actions is central to the understanding and building computational models of human motion and activity recognition [134]. The main difficulty of HAS stems from the large intra-person physical variability and the irregularity in periodicity of human activities. In addition, there are exponential number of possible movement combination.

In this thesis, we add the flavour of Human Activity Evaluation (HAE) to the HAS problem. In that we are interested in temporally segment the videos into semantically distinct actions and be able to determine the ergonomics risk associated with each activity. Ergonomics Risk Assessment

(ERA) is the process of evaluating human movements to calculate a risk score indicating the level of ergonomics risk (more details can be found in Chapter 2).

In Chapter 3, we present a new dataset we collected, UW-IOM, that includes activities like picking-up a box, reaching, bending, walking, and etc. In addition we use the skeleton detected using a Kinect sensor to evaluate a popular industry metric for ERA called Rapid Entire Body Assessment (REBA) for each frame. The REBA scores are averaged for each atomic action and are integrated into the activity labels. A Encoder-Decoder Temporal Convolutional Network (ED-TCN) is implemented to semantically segment the videos. Multiple feature representation are used as the backbone of the algorithm and the comparison of the results is reported.

In Chapter 4, we present a novel Multi-Task Learning (MTL) approach for learning to both HAS and ERA task simultaneously. Our proposed framework comprises a Graph Convolutional Network (GCN) backbone and an Encoder-Decoder Temporal Convolutional Network (ED-TCN) for the activity segmentation head and a Long-Short-Term-Memory (LSTM)-based head for activity assessment. The contribution of our work is threefold. First, we introduce a novel combination of GCN with ED-TCN for activity segmentation in long videos that outperforms state-of-the-art results on the UW-IOM dataset. Second, our MTL-emb method initiates a line of research for more informed activity assessment by fusing activity embedding with spatial features for ERA. Third, we present a way to use the skeletal information for activity assessment in a Multi-Task Learning (MTL) framework that may enable generalization across a variety of environments and leverage anthropometric information.

Part II: Online-Human Activity Recognition

This part of the thesis is devoted to the online-HAR. In Chapter 5, we propose a novel real-time Spatio-Temporal Pyramid Graph Convolutional Network (ST-PGN) for action recognition that enables the use of features from all levels of the skeleton feature hierarchy. ST-PGN, designed with a feature pyramid architecture enables the model to capture the correlation between body parts, rather than hand-coding body-part relations. We test the performance of the model on two public benchmark datasets typically used for postural assessment (TUM and UW-IOM) as well as Ki-

netics and NTU-RGBD datasets. We show that the algorithm is also able to learn the transitions between actions and is suitable for real-time applications. As compared to the state-of-the-art algorithms such as ST-GCN [156], our model has fewer graph convolution kernels without sacrificing performance. Finally, we enhance the pipeline with postural assessment methods (REBA [43]) that use the online action recognition output of our model to produce ergonomics risk estimates. We propose, this combined action-risk architectural design as a first step towards automated assessment of musculoskeletal disorders in occupational safety.

In Chapter 6, first, we introduce a novel hierarchical graph structure to handle temporal non-Euclidean datasets, in which the nodes are non-Euclidean themselves. Second, Inspired by Radial Basis Function (RBF), we define an adjacency function in real-time for the Human Object Interaction Graph Convolutional Network (HOI-GCN). Third, We leverage our ST-HGCN architecture to solve an early activity recognition problem that, to the best of our knowledge, has not been explored in the context of HOI. We evaluate our algorithm on the UW-IOM dataset, and show that it not only significantly outperforms the state-of-the-art, but also converges faster than the already existing methods. In addition, we demonstrate how our proposed method can be applied to any custom dataset—possibly created for real industrial applications.

In the next chapter, provides the required background knowledge for the technical chapters of the thesis. we summarize the related research on HAS, online-/early-HAR, ERA, and GCN. And finally, in chapter 7 we discuss the key takeaways from the presented research and the potential future directions.

Chapter 2

LITERATURE REVIEW

Human Action Recognition (HAR) is an important topic in computer vision that represents a broad category of problems dealing with identifying human activity in images or videos. In this chapter we review works related to Human Activity Segmentation (HAS) and early-HAR. HAS focuses on semantically segmenting a video into predefined classes of human activity. On the other hand, early-Human Action Recognition (HAR) is the task of identifying human activities after the observation of a few frames from a video. The application of these categories has extended to solving Human Activity Evaluation (HAE) tasks, in which the goal is to determine how well activities are performed. Later in the chapter we discuss the developments for this application, and specifically talk about automated Ergonomics Risk Assessment (ERA).

2.1 Human Activity Recognition and Object Interaction

2.1.1 Human Action Recognition

Human Action Recognition (HAR) is the task of identifying the human activity within an image or a video. Video-based HAR itself can be categorized into three major problems. The first category is video action segmentation (sometimes referred to as action detection or offline action recognition), which is the task of localizing action labels in untrimmed videos or classifying the entire video clip with one label [15, 126, 135, 139, 140, 156, 164]. There has been significant effort on solving this category of problems [15, 47, 62, 132]. Usually, the preferred method for modeling the temporal properties of the videos is to use variations of temporal convolutions [6, 64], since they can better aggregate long temporal relations compared to RNN-based methods.

The second category of video-based HAR is online action recognition [99]. In these problems, the goal is to identify actions in an ongoing, partially observed videos that include multiple action

classes [77, 78, 131]. In Chapter 5 and 6, our focus would be mainly on this type of problems due to the relevance of industrial applications, for instance, analyzing surveillance videos. We seek a scheme that performs well on any sample clip from a video sequence, regardless of the number of activity classes.

Regarding the feature representation, skeletal action recognition has recently received immense attention in the computer vision community. Human pose and relative motion of the body parts convey key information about the human action. Many efforts have been dedicated to improve the human dynamic modeling using GCN and leveraging that in solving HAR problems [66, 128, 156, 160]. The advantage of using skeleton-based methods is that they are robust to variations in illumination or color. Nonetheless, the drawback of using such methods is that crucial scene information are often missing. In many activities, the differentiating factor is an element within the scene, namely, an object that is being manipulated.

Despite the significance of the embedded information within the scene, only a few methods illustrate a way to incorporate the dynamic scene representation into the action recognition problems. Parsa et al. [99] propose a fusion mechanism using VGG16 network to combine scene feature with the one for the pose. Zhou and Chi [167] present a method that captures the relationship between the object and the body part in a *object-bodypart graph* and the relationship between body parts with a *human-bodypart graph* for solving HOI detection in images. To the best of our knowledge, no methodology has been developed for skeleton-based online-HAR that incorporates dynamic scene information using a GCN. In Chapter 6, we propose a ST-HGCN scheme to solve online-HAR. Our model not only addresses the relations between human body parts, but also incorporates dynamic scene information.

Other attempts to address online early action recognition for indoor datasets are [77, 78, 131], which use PKU-MMD dataset [76] and OAD dataset [71]. However, the focus of those works is on modelling the temporal evolution of poses and early prediction of future actions. Rather than predicting future pose streams, our aim is to instead classify incoming pose streams. It is imperative to capture local label transitions (*reaching* to *pickup*) by exploiting subtle pose cues and temporal sequence understating. Moreover, as evidenced by our ST-GCN [156] experiments, offline models

do not translate well to online settings. Hence, in Chapter 6 we propose a hierarchical architecture that can do these tasks jointly in an online manner.

2.1.2 Skeleton-based Action Recognition

With advances in reliable pose estimation models [14, 117, 163], skeleton only action recognition has gained popularity [66, 117, 128, 156, 160]. Those methods have shown to be robust to variations in illumination and scene, and are typically context agnostic. One limitation of previous methods is that they do not use the necessary features from scene context or object handling which give more meaning to the actions (e.g. walking on crosswalk means crossing verses walking indoors, lifting box vs lifting rod). Using scene only cues limits models from capturing complex pose dynamics and relative pose structure changes (e.g. hand moving in relation to torso means reaching for object). In this regard [156] is, to our knowledge, the first method to operate on a local pose structure graph. For more literature on various action recognition techniques we recommend [146].

Skeleton-based action recognition methods not only facilitate the design of generalizable solutions for various environments, they also provide the opportunity to study downstream applications such as human performance analysis. The information about human-objects relations inferred from HOI, on the other hand, create the opportunity to design algorithms for better scene understanding such as process fidelity assessment in manufacturing. In this work, we leverage the skeleton-based activity representation as well as the human-object relations to design a generalizable algorithm for solving online-HAR problems. Here, we summarize the related work to our proposed ST-HGCN method in the literature of HAR and HOI.

GCNs was developed to process data belonging to non-Euclidean spaces [151]. GCNs are the most intuitive choice for human body kinematics since the commonly-used independent and identically distributed random variable assumption is not applicable. Spatio-Temporal Graph Convolutional Networks (ST-GCN) introduced a powerful tool for analyzing human motions in videos, and has been utilized in several computer vision applications [54, 69, 128, 156]. However, most of these works focus on solving Human Action Recognition (HAR) problems. Recently, [99] in-

troduced a Spatio-Temporal Pyramid Graph Network (ST-PGN) for early action recognition. They also used the predicted activity labels to enhance ERA that was computed using 3D skeletal reconstruction. In this work, we leverage a GCN backbone to learn the joint embedding and use that to directly predict the ergonomics risks rather than solving it as a separate problem.

Graph convolution network (GCN) is a powerful method for processing non-Euclidean spaces [151]. Since the skeleton structure is inherently represented as a graph with nodes and connections, GCN is increasingly being used for analyzing human motion for different applications. Spatio-temporal graph convolutions add another dimension to GCN by applying convolutions over spatial domain, and temporal convolutions (TCN) over the time domain in a sequential manner. Most related work in skeleton based action recognition include [54, 69, 128, 156]. The first three papers focus on graph convolution on temporal skeleton sequences. However, they do not model the hierarchical parts structure in graphs.

Recently, Kim et al. [54] introduced a two-stream method for human action recognition. They used a human pose stream based on ST-GCN and an object-related pose stream which is achieved by training an object detector on the set of objects of their interest. Similarly, in chapter 5, we fuse the object/context features along with pose dynamics. However, we focus on enhancing the skeleton features and treat objects as features from VGG16. We propose an alternative strategy for fusion inspired by GRU. The focus is to avoid confusion between objects handled in the labels, for instance, pose configuration for picking up a rod and picking up a box look similar and can be classified incorrectly.

2.1.3 Human Object Interaction

Human Object Interaction (HOI) is the task of inferring relationships between human and objects, such as "lifting a box" or "washing a car". In general, the task is to detect the triplet of $(human, verb, object)$, however, the HOI problems are often laid out such that an object affordance is predicted—in addition to the human's activity. Inference of such complex relations requires meticulously crafted datasets such as HICO-DET [17], V-COCO [39], and CAD-120 [60] just to name a few. The features that represent the scene elements—including the human—are of-

ten emanated from the feature representations driven from object detection backend. Also, human pose has rarely been leveraged in HOI problems. Recently, Wan et al. [145] proposed a method that utilizes human body parts and objects relation to enhance the performance of their HOI detection model on images.

Another aspect of the HOI that has not been fully studied is a dynamic scene, in which the number of objects can be different in every frame. Qi et al. [112] propose a Graph Parsing Neural Network for solving temporal HOI problems and they use CAD-120 [60] to test their spatio-temporal algorithm. CAD-120 dataset has a particular characteristics where, each activity is represented by an average of 8 frames, and also, same number of objects are present in each clip. Furthermore, the order of the objects within CAD-120 remains fixed which, in turn, simplifies the construction of the adjacency matrix. This dataset is not applicable to our setup, as our algorithm is specifically designed for long videos. In Chapter 6, we present a ST-HGCN, which makes it possible to tackle HOI for an online setting and with a dynamic scene.

2.2 Human Activity Evaluation

Also known as Action Quality Assessment, HAE focuses on designing models that are able to learn a mapping between human body dynamics and the completion quality of the performed actions based on an accepted metric or a template sequence (refer to [67] for further literature on early methods with handcrafted features). The majority of deep learning approaches to HAE have focused on using 3D Convolutional Neural Networks (C3D) [139] and Pseudo-3D Networks (P3D) [152] to extract spatio-temporal features that are fed into a regression unit. One of the recent works in applications for physical therapy, [74], proposed a framework including performance metrics, scoring functions, and different neural network architectures for mapping joint coordinates to the activity score. Similarly, [97] used C3D to extract spatio-temporal features and conducted performance score regression using a LSTM unit for data from Olympic events. Despite the value of all these works in initiating the use of computer vision techniques for HAE in rehabilitation and sports, the proposed methods are highly dependent on the context of the video frames. Moreover, the learned mapping between the frames and the score does not incorporate the effect of human

body kinematics.

Recently, there have been efforts in leveraging human body kinematics in designing deep architectures for evaluating surgical skills [31]. This work uses 75 dimensional kinematic data (3D coordinates plus velocities) of two surgical tools being manipulated by surgeons and classifies the skill level into expert, intermediate, and novice. Joint relation graph has been utilized to assess the performance of athletes in Olympic events [93]. The proposed joint relation graph is a spatial GCN with node features that are outputs of I3D [15] on image patches containing the human joints.

Parmar and Morris's work [93] is the most similar work to our paper. They propose a multi-task framework utilizing spatio-temporal features to solve action recognition, commentary generation, and HAE score estimation for Olympic events. However, the focus of their work is on short video classification, where each clip includes only one activity, namely, diving of one athlete. In contrasts, our focus is on localizing actions in a long video while simultaneously inferring the ergonomics risk of human posture at every frame.

2.3 Ergonomics Risk Assessment

One of the key considerations for viable human-robot collaboration (HRC) in industrial settings is *safety*. This consideration is particularly important when a robot operates in close proximity to humans and assists them with certain tasks in increasingly automated factories and warehouses. Therefore, it is not surprising that a lot of effort has gone into identifying and implementing suitable HRC safety measures [159]. Typically, the efforts focus on designing collaborative workspaces to minimize interferences between human and robot activities [87], installing redundant safety protocols and emergency robot activity termination mechanisms through multiple sensors [87], and developing both predictive and reactive collision avoidance strategies [115]. These efforts have resulted in the expanded acceptance and use of industrial robots, both mobile and stationary, leading to increased operational flexibility, productivity, and quality [82].

A key factor in achieving safe HRC is accurate robotic perception of humans actions and their potential risks. Specifically, perceiving (assessing) the ergonomic risks of human actions is an extremely important topic that has not received much attention until recently [35, 70]. Unlike

other commonly considered safety measures, a lack of ergonomic safety does not lead to immediate injury concerns or fatality risks. It, however, causes or increases the likelihood of causing longterm injuries in the form of musculoskeletal disorders (MSDs) [42]. According to a recent report by the U.S. Bureau of Labor Statistics, there were 349,050 reported cases of MSDs in 2016 just in the U.S. [90], leading to tens of billions of dollars in healthcare costs.

Most organizations use conventional ergonomic risk assessment methods, which are based on observations and self-reports, making them error-prone, time consuming, and labor-intensive [133]. More recently, researchers have started exploring alternative sensor-based automated assessment methods. For example, Li et al. [68] used distributed surveillance cameras and body-mounted motion sensors for this purpose. Shafti et al. [124] used an RGB-D camera to extract the skeletal information of the arm and understand the safe range of arm motions and give feedback on the subjects performance during welding. Kim et al. [55] introduced a reconfigurable HRC workstation to monitor and adjust the ergonomic risks of working with power tools in real time using a stereovision camera.

From a methodological perspective, deep learning has become popular in assessing the risks of performing occupational tasks, especially in the construction industry [26, 29]. Outside of the construction sector, Abobakr et al. [2] employed deep residual convolutional networks (CNNs) to predict the joint angles of manufacturing workers from individual camera depth images. Mehrizi et al. [85] proposed a multi-view based deep perceptron approach for marker-less 3D pose estimation in the context of object lifting tasks. While all these works present useful advances and report promising performances, they do not provide a general-purpose framework to predict the ergonomic risks for any representative set of object manipulation tasks commonly performed in the industry.

Manufacturing assembly is another well studied area for ergonomics risk analysis [22, 79, 106, 109, 118, 130]. Rapid Entire Body Assessment (REBA) [44] and the European Assembly Worksheet (EAWS) [123] are two common ergonomics risk measures used in the industry. REBA is a tabular method created by experts in ergonomics by evaluating over 600 postural examples. REBA is a less qualitative measure for ergonomics risk assessment which takes the human joint

angles and computes a risk score. EAWS, however, is focused at the type of activity that is done during an assembly task. Both metrics are traditionally determined visually, by an expert observing the action.

One line of research is focused on using body-mounted motion sensors for automation of ergonomics risk assessment [68, 79]. Recently, [81] introduces a dataset which is very useful for studying ergonomics for collaborative robotics application. Another track focuses on using only a camera sensor for evaluating the safety of an activity. For instance, in [124] an RGB-D camera is used to find a safe range for arm movements and give feedback on the subjects' performance during welding. In [55] a camera is used to monitor and adjust the ergonomics risks of working with power tools in real-time. Moreover, in [100], an offline method is introduced to segment a video into semantically meaningful actions and report an ergonomics risk level for each action.

Improved ergonomics risk assessment can be attained by considering not only the posture, but also the action and object interaction. In this thesis, along with our effort to advance activity recognition algorithms, we present ways to leverage these HAR methods for automated ergonomics risk assessment. In Chapter 3, the ERA problem is taken as an action localization problem and Temporal Convolutional Network (TCN) is used to segment the videos into tasks with different risk labels. The ergonomics risk is computed offline and the dataset is labeled with high-, medium-, and low-risk labels. In Chapter 4, on the other hand, introduces a multi-task Human Postural Assessment (HPA) framework that predicts ergonomics risk directly from human pose with the help of HAS as an auxiliary task. In Chapter 5, we compute REBA frame-wise and use the recognition predictions to adjust the scores. The proposed activity recognition algorithm predicts the postures and actions, and identifies object interactions and the height at which the activity is being performed, which are important for REBA calculation.

Part I

HUMAN ACTIVITY SEGMENTATION

Chapter 3

ENCODER-DECODER TEMPORAL CONVOLUTIONAL NETWORK

Automated real-time prediction of the ergonomic risks of manipulating objects is a key unsolved challenge in developing effective human-robot collaboration systems for logistics and manufacturing applications. In this chapter, We present a foundational paradigm to address this challenge by formulating the problem as one of action segmentation from RGB-D camera videos. Spatial features are first learned using a deep convolutional model from the video frames, which are then fed sequentially to temporal convolutional networks to semantically segment the frames into a hierarchy of actions, which are either ergonomically safe, require monitoring, or need immediate attention. For performance evaluation, in addition to an open-source kitchen dataset, a new dataset was collected comprising twenty individuals picking up and placing objects of varying weights to and from cabinet and table locations at various heights. Results show very high (87-94)% F1 overlap scores among the ground truth and predicted frame labels for videos lasting over two minutes and comprising a large number of actions. In the following sections, We start by discussing the deep learning methods that We adapted for solving this problem, then, We introduce the datasets, and lastly We illustrate the experimental details and discuss the results.

3.1 Background

One of the key considerations for viable human-robot collaboration (HRC) in industrial settings is *safety*. This consideration is particularly important when a robot operates in close proximity to humans and assists them with certain tasks in increasingly automated factories and warehouses. Therefore, it is not surprising that a lot of effort has gone into identifying and implementing suitable HRC safety measures [159]. Typically, the efforts focus on designing collaborative workspaces to minimize interferences between human and robot activities [87], installing redundant safety

protocols and emergency robot activity termination mechanisms through multiple sensors [87], and developing both predictive and reactive collision avoidance strategies [115]. These efforts have resulted in the expanded acceptance and use of industrial robots, both mobile and stationary, leading to increased operational flexibility, productivity, and quality [82].

A key factor in achieving safe HRC is accurate robotic perception of humans actions and their potential risks. Specifically, perceiving (assessing) the ergonomic risks of human actions is an extremely important topic that has not received much attention until recently [35, 70]. Unlike other commonly considered safety measures, a lack of ergonomic safety does not lead to immediate injury concerns or fatality risks. It, however, causes or increases the likelihood of causing longterm injuries in the form of musculoskeletal disorders (MSDs) [42]. According to a recent report by the U.S. Bureau of Labor Statistics, there were 349,050 reported cases of MSDs in 2016 just in the U.S. [90], leading to tens of billions of dollars in healthcare costs.

Most organizations use conventional ergonomic risk assessment methods, which are based on observations and self-reports, making them error-prone, time consuming, and labor-intensive [133]. More recently, researchers have started exploring alternative sensor-based automated assessment methods. For example, Li et al. [68] used distributed surveillance cameras and body-mounted motion sensors for this purpose. Shafti et al. [124] used an RGB-D camera to extract the skeletal information of the arm and understand the safe range of arm motions during welding. Kim et al. [55] introduced a reconfigurable HRC workstation to monitor and adjust the ergonomic risks of working with power tools in real time using a stereovision camera.

From a methodological perspective, deep learning has become popular in assessing the risks of performing occupational tasks, especially in the construction industry [26, 29]. Outside of the construction sector, Abobakr et al. [2] employed deep residual convolutional networks (CNNs) to predict the joint angles of manufacturing workers from individual camera depth images. Mehrizi et al. [85] proposed a multi-view based deep perceptron approach for markerless 3D pose estimation in the context of object lifting tasks. While all these works present useful advances and report promising performances, they do not provide a general-purpose framework to predict the ergonomic risks for any representative set of object manipulation tasks commonly performed in

the industry.

Here, we present a first of its kind *end-to-end* deep learning system for ergonomic risk assessment during indoor object manipulation using camera videos. Our learning system is based on *action segmentation*, where an action class (with a corresponding risk label) is predicted for every video frame.

Representative works on this topic include that by Fathi et al. [30], who showed that state changes at the start and end frames of actions provided good segmentation performance. Kuehne et al. [61] used reduced Fisher vectors for visual (spatial) representation of every frame, which were then fitted to Gaussian mixture models. Huang et al. [46] presented a temporal classification framework in the case of weakly supervised action labeling. Ghosh et al. [33] recently developed a graph-based spatiotemporal CNN to exploit environmental cues for better segmentation. Along similar lines, our method uses a combination of spatial and temporal CNNs to achieve good segmentation performance.

In addition, we present a new benchmark dataset, called the University of Washington Indoor Object Manipulation (UW IOM) dataset, for vision-based ergonomic risk assessment studies. Given an acceptable ergonomic risk model, we then show that our end-to-end system satisfactorily predicts the risks of actions in test videos. The goal of our system, therefore, is to enable the collaborative robots to accurately detect the risky manipulation actions so that they can perform these actions, allowing the humans to instead engage in supervisory control or cognitively challenging tasks.

3.2 Ergonomic Risk Assessment Model

We use a well-established ergonomic model, known as the rapid entire body assessment (REBA) model [44], which is popularly used in the industry. The REBA model assigns scores to the human poses, within a range of 1-15, on a frame-by-frame basis by accounting for the joints motions and angles, load conditions, and activity repetitions. An action with an overall score of less than 3 is labeled as ergonomically safe, a score between 3-7 is deemed to be medium risk that requires monitoring, and every other action is considered high risk that needs attention.

Skeletal information for the TUM Kitchen dataset [136] is available in the bvh (Biovision Hierarchy) file format. We use the bvh parser from the MoCap Toolbox [12] in MATLAB to read this information as the XYZ coordinates of thirty three markers (joints and end sites) corresponding to every frame. For the UW IOM dataset, the positions of twenty five different joints are recorded directly in the global coordinate system for each frame using the Kinect sensor with the help of a toolbox [138] that links Kinect and MATLAB. For every frame, the vectors corresponding to different body segments such as fore-arm, upper-arm, leg, thigh, lower half spine, upper half spine, and so on, are computed. The extension, flexion, and abduction (as applicable) of the various body segments are computed by taking the projection of the two body segment vectors that constitute the angle on the plane of motion. These angles of extension, flexion, and abduction are used to assign the trunk, neck, leg, upper arm, lower arm, and wrist scores [44].

We define three different thresholds as a part of our implementation, namely, zero threshold, binary threshold, and abduction threshold. Zero threshold is used for trunk bending, such that any trunk bending angle less than this value is regarded as no bending. Binary threshold is defined to answer whether the trunk is twisted and/or side flexed. Trunk twisting and trunk side flexion less than this value are ignored. Abduction threshold, though similar to the binary threshold, is separately defined for shoulder abduction considering the considerably larger allowable range of abduction (about 150°) as against a smaller allowable range of trunk twisting. Due to the non-availability of rotation information of the neck, We assume that the neck is twisted when the trunk is twisted, which is not entirely unreasonable. The nature of the performed actions does not involve arm rotations, and they are ignored while computing the upper arm score.

The computed trunk, neck, leg, upper arm, lower arm, and wrist scores are used to assign the REBA score on a frame-by-frame basis using lookup tables [44]. The REBA scores assigned for each frame are then aggregated over all the actions and participants, so that We have a constant risk score for every frame that corresponds to a particular action. This aggregated value is also considered as the final REBA score for that particular action.

3.3 Deep Learning Models

3.3.1 Spatial Features Extraction

We adapt two variants of VGG16 convolutional neural network models [129] for spatial feature extraction. The first model is based on the VGG16 model that is pre-trained on the ImageNet database [24]. The second model involves fine-tuning the last two convolutional layers of the VGG16 base that is pretrained on ImageNet. In both the models, the flattened output of the last convolutional layer is connected to a fully connected layer with a drop-out of 0.5 and then fed into a classifier. We always use rectified linear units (ReLU) and softmax as the activation functions, and Adam [56] as the optimizer.

We also use a simplified form of the pose-based CNN (P-CNN) features [20] that only consider the full images and not the different image patches. Optical flow [11] is first computed for each consecutive pair of RGB datasets, and the flow map is stored as an image [34]. A motion network, introduced in [34], containing five convolutional and three fully-connected layers, is then used to extract frame descriptors for all the optical flow images. Subsequently, the VGG-f network [19], pre-trained on ImageNet, is used to extract another set of frame descriptors for all the RGB images. The VGG-f network also contains five convolutional and three fully connected layers. The two sets of frame descriptors are put together as arrays in the same sequence as that of the video frames to construct motion-based and appearance-based video descriptors, respectively. The appearance and motion-based video descriptors are then normalized and concatenated to form the final video descriptor (spatial features).

3.3.2 Video Segmentation Methods

We use two kinds of temporal convolutional networks (TCNs), both of which use encoder-decoder architectures to capture long-range temporal patterns in videos [64]. In the first network, referred as the encoder decoder-TCN, or ED-TCN, a hierarchy of temporal convolutions, pooling, and upsampling layers is used. The network does not have a large number of layers, but each layer includes a set of long convolutional filters. We use the ReLU activation function and a categorical

cross-entropy loss function with RMSProp [1] as the optimizer. In the second network, termed as dilated-TCN, or D-TCN, dilated upsampling and skip connections are added between the layers. We use the gated activation function as it is inspired by the WaveNet [142] and Adam optimizer. We also use two other segmentation methods for comparison purposes. The first method is bidirectional long short term memory, or Bi-LSTM[36], a recurrent neural network commonly used for analyzing sequential data streams. The second method is support vector machine, or SVM, which is extremely popular for any kind of classification problem.

3.3.3 Video Segmentation Performance Metrics

In addition to frame-based accuracy, which is the percentage of frames labeled correctly for the related sequence as compared to the ground truth (manually annotated), We report edit-score and F1 overlap score to evaluate the performance of the various methods. The edit-score [65] measures the correctness of Levenshtein distance to the segmented predictions. The F1 overlap score [65], combines classification precision and recall to reduce the sensitivity of the predictions to minor temporal shifts between the predicted and ground truth values, as such shifts might be caused by subjective variabilities among the annotators.

3.4 System Architecture and Datasets

3.4.1 System Architecture

We develop an end-to-end automated ergonomic risk prediction system as shown in Fig. 3.1. The Figure shows that the prediction works in two stages. In the first stage (top half of the Figure), which only needs to be done once for a given dataset, ergonomic risk labels are computed for each object manipulation action class based on the skeletal models extracted from the RGB-D camera videos. Simultaneously, the videos are annotated carefully to assign an action label to each and every frame. These two types of labels are then used to learn a modified VGG16 model for the entire set of available videos. In the second stage (bottom half of the Figure), during training, the exact sequence of video frames is fed to the learned VGG16 model to extract useful spatial

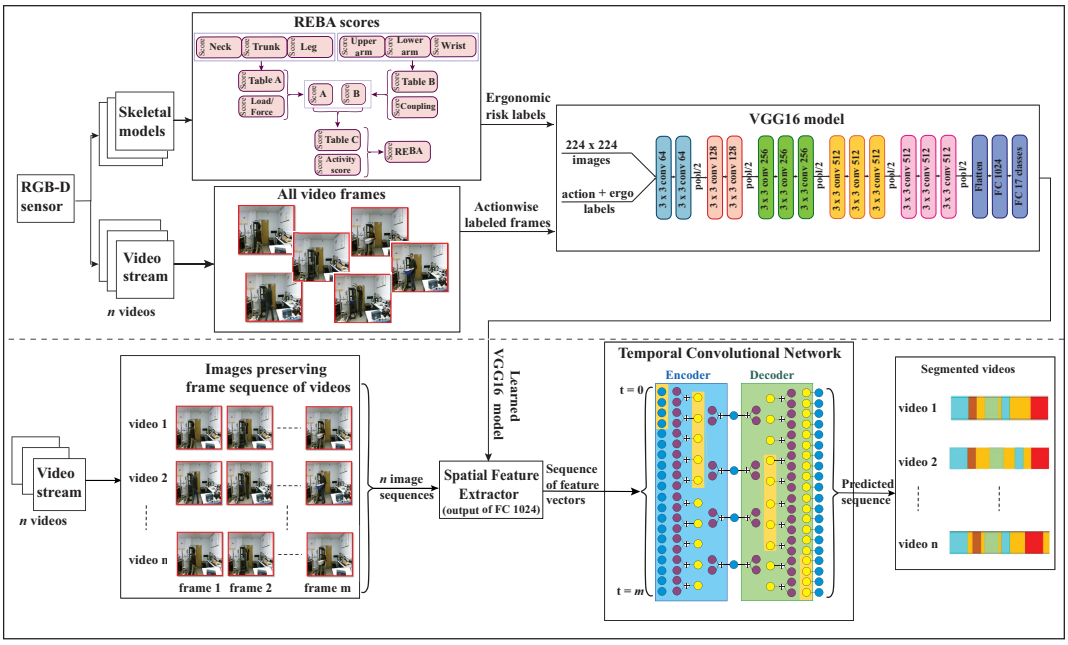


Figure 3.1: End-to-end ergonomic risk prediction system

features. The array of extracted features is then fed in the same order to train a TCN on how to segment the videos by identifying the similarities and changes in the features corresponding to actions executions and transitions, respectively. For testing, a similar procedure is followed except that the trained TCN is now employed to segment unlabeled videos into semantically meaningful actions with known ergonomic risk categories. It is possible to replace the VGG16 model by another deep neural network model that also accounts for human motions (e.g., P-CNN) to achieve slightly better segmentation performance at the expense of longer training and prediction times.

3.4.2 Datasets

TUM Kitchen Dataset

The TUM Kitchen dataset [136], consists of nineteen videos, at twenty-five frames per second, taken by four different monocular cameras, numbered from 0 to 3. Each video captures regular actions performed by an individual in a kitchen involving walking, picking up, and placing utensils

to and from cabinets, drawers, and tables. The average duration of the videos is about two minutes. The dataset also includes skeletal models of the individual through 3D reconstruction of the camera images. These models are constructed using a markerless full body motion tracking system through hierarchical sampling for Bayesian estimation and layered observation modeling to handle environmental occlusions [7]. We categorize the actions into twenty-one classes or labels, where each label follows a two-tier hierarchy with the first tier indicating a motion verb (close, open, pick-up, place, reach, stand, twist, and walk) and the second tier denoting the location (cabinet, drawer) or mode of object manipulation (do not hold, hold with one hand, and hold with both hands).

UW IOM Dataset

Considering the dearth of suitable videos capturing object manipulation actions involving awkward poses and repetitions, We collected our own dataset using an Institutional Review Board (IRB)-approved study. The dataset comprises videos of twenty participants within the age group of 18-25 years, of which fifteen are males and the remaining five are females. The videos are recorded using a Kinect Sensor for Xbox One at an average rate of twelve frames per second. Each participant carries out the same set of tasks in terms of picking up six objects (three empty boxes and three identical rods) from three different vertical racks, placing them on a table, putting them back on the racks from where they are picked up, and then walking out of the scene carrying the box from the middle rack. The boxes are manipulated with both the hands while the rods are manipulated using only one hand. The above tasks are repeated in the same sequence three times such that the duration of every video is approximately three minutes. We categorize the actions into seventeen labels, where each label follows a four-tier hierarchy. The first tier indicates whether the box or the rod is manipulated, the second tier denotes human motion (walk, stand, and bend), the third tier captures the type of object manipulation if applicable (reach, pick-up, place, and hold), and the fourth tier represents the relative height of the surface where manipulation is taking place (low, medium, and high). Representative snapshots from one of the videos are shown in Fig. 3.2. Each video is annotated manually using the ANVIL annota-

tion tool [58]. After annotating all the videos, the frames within the same class are extracted and checked for accuracy and consistency. The UW IOM dataset is available for free download and use at: https://data.mendeley.com/datasets/xwzzkxtf9s/draft?a=c81c8954-6cad-4888-9bec-6e7e09782a01.

3.5 Experimental Details and Results

3.5.1 Implementation Details

For each participant (video), We first compute the REBA score for all the frames. The zero threshold is set to 5° and the binary threshold is set to 10°. To avoid minor shoulder abductions from contributing substantially owing to Kinect tracking errors, the abduction threshold is chosen as 30°. For the UW IOM dataset, we compute the median of the REBA scores assigned to all the frames belonging to a particular action. We then take the median over all the participants to determine the final REBA score for that action.

The framewise skeletal information available for the TUM Kitchen dataset has a variable lag with respect to the video frames, i.e., the skeleton does not lie exactly on the human pose in the RGB image. Therefore, aggregating over actions and participants according to the RGB image annotations does not result in meaningful REBA scores. Therefore, the length of both the video annotations of the RGB frames and the framewise REBA scores are reduced to 100 using a constant step size of number of frames/100 for every video. Then the average REBA score is computed for every action in a particular video using the reduced video annotations and framewise scores. For safety considerations, the maximum score assigned to a particular action among all the videos is considered as the final REBA score for that action.

The pre-trained VGG16 model for spatial features extraction is trained for 200 epochs with 300 steps per epoch on the TUM Kitchen dataset, and 300 epochs with 300 steps per epoch on the UW IOM dataset with a step-size of 10^{-5}. The fine-tuned model is trained with the same number of epochs for the TUM Kitchen dataset but with 500 steps per epoch on the UW IOM dataset with 300 steps per epoch and a step-size of 10^{-7}. The number of training and validation samples for the

Figure 3.2: Representative video frames depicting actions with different ergonomic risk levels in our own UW IOM dataset

TUM Kitchen dataset are 24,052 and 5,290, respectively. For the UW IOM dataset, We train over 27,539 samples and validate over 6,052 samples. The models are learned using the TensorFlow machine learning software library [137] and Python-based Keras [53] neural network library as the backend. To implement the simplified P-CNN model, We modify the MATLAB package provided with [20].

We evaluate the performance of the four segmentation methods by splitting our datasets into five splits, in each of which, the videos are assigned randomly to mutually exclusive training and test sets of fixed sizes. For both the TCN methods, training is terminated after 500 epochs in each of the splits as the validation accuracy stops improving afterward. We use a learning rate of 0.001 for both the methods. D-TCN includes five stacks, each with three layers, and a set of $\{32, 64, 96\}$ filters in each of the three layers. Filter duration duration, defined as the mean segment duration for the shortest class from the training set, is chosen to be 10 seconds. Similarly, training for Bi-LSTM is terminated after 200 epochs for each split as the validation accuracy does not change any further. Bi-LSTM uses Adam optimizer with a learning rate of 0.001, softmax activation function, and categorical cross-entropy loss function. We choose a linear kernel to train the SVM and use squared hinge loss as the loss function. All the training and testing are done on a workstation running Windows 10 operating system, equipped with a 3.7GHz 8 Core Intel Xeon W-2145 CPU, GPU ZOTAC GeForce GTX 1080 Ti, and 64 GB RAM.

3.5.2 Results

Ergonomic Risk Assessment Labels

For the TUM Kitchen dataset, fifteen actions are labeled to be medium risk, while the remaining six are deemed as high risk. The high risk actions are associated with closing, opening, and reaching motions, although there is no perfect correspondence due to a lack of fidelity of the skeletal models on which the risk scores are based upon.

In case of the UW IOM dataset, three actions are labeled as low risks, eleven actions are considered medium risk, and the remaining three are identified as high risk. The high risk actions

Table 3.1: Comparative performance measures of different action segmentation methods on the TUM Kitchen dataset for camera # 2.

Method		Accuracy (%)	Edit score (%)	F1 overlap (%)
Pre-trained VGG16	**D-TCN**	73.74±4.57	78.7±6.50	83.88±4.52
	ED-TCN	**74.75±4.08**	**86.34±3.15**	**87.92±2.16**
	Bi-LSTM	62.55 ± 6.56	44.49±8.67	55.11±9.42
	SVM	59.55 ± 4.98	35.39±3.00	47.75±3.82
Fine-tuned VGG16	**D-TCN**	74.14±4.97	80.33±5.41	84.44±4.05
	ED-TCN	**74.32±4.06**	**84.96±4.37**	**87.29±2.78**
	Bi-LSTM	62.89± 6.17	47.15±8.67	57.75±9.02
	SVM	59.81 ± 5.10	35.8±3.54	47.67±4.35

include picking up a box from the top rack and placing objects (box and rod) on the top rack. Walking without holding any object, walking while holding a box, and picking up a rod from the mid-level rack while standing are regarded as low risk, i.e., safe actions. Fig. 3.2 shows the corresponding ergonomic risk labels for these different actions depicted in the video snapshots

Video Segmentation Outcomes

Table 3.1 provides a quantitative performance assessment of the two variants of our segmentation method on the TUM Kitchen dataset for camera # 2 videos. Both the variants perform satisfactorily with respect to all the three performance measures. In fact, the ED-TCN method achieves an F1 overlap score of almost 88%, which has not been previously reported for any action segmentation problem with more than twenty labels to the best of our knowledge. Our TCN methods also outperform Bi-LSTM and SVM substantially. Just for comparison purposes, it is interesting to note that the pre-trained and fine-tuned VGG16 models provide validation accuracy of 82.80% and 73.46%, respectively, during image classification. Fig. 3.3 demonstrates that regardless of whether the spatial features are extracted using a pre-trained or fine-tuned VGG16 architecture, both the TCN methods are able to segment the frames into the correct (or more precisely, same as the

manually annotated) actions substantially better than Bi-LSTM and SVM. In fact, the global frame-by-frame classification accuracy value is very high, between (86-91)%, using the TCN methods. Furthermore, both the TCN methods almost always predict the correct sequence of actions unlike the other two widely-used classification methods.

The difference in performance between the TCN and other two segmentation methods is even more pronounced in case of the UW IOM dataset, which includes a larger variety of object manipulation actions. As shown in Table 3.2, SVM performs rather poorly particularly with respect to edit score and F1 overlap values owing to over-segmentation and sequence prediction errors. Bi-LSTM performs somewhat better with the best results obtained using the spatial features generated from a simplified form of P-CNN. Interestingly enough, ED-TCN performs substantially better than D-TCN regardless of the spatial feature extraction method being used. This finding is also consistent with the results for different grocery shopping, gaze tracking, and salad preparation datasets presented in [64]. It happens most likely due to the ability of ED-TCN to identify fine-grained actions without causing over-segmentation by modeling long-term temporal dependencies through max pooling over large time windows. In fact, the edit scores for ED-TCN are close to 90% and the F1 overlap values are more than 93% when We use the fine-tuned VGG16 and P-CNN models. The performance measures are almost identical between the two models with P-CNN yielding marginally better results. For just the pre-trained VGG16 and fine-tuned VGG16 models, the validation accuracy is 75.97% and 73.86%, respectively, which are similar to the values for the TUM Kitchen dataset. Fig. 3.4 reinforces these observations on a representative UW IOM dataset video.

If one only uses the spatial features, image classification validation accuracy is either comparable to (for the TUM Kitchen dataset), or lower than the video segmentation test accuracy (for the UW IOM dataset). Noting that validation accuracy is typically greater than test accuracy for any supervised learning problem, one would expect segmentation accuracy to be much lower than the reported values in the absence of the temporal neural networks. On the other hand, segmentation performance depends quite a bit on the choice of the spatial feature extraction model, particularly in the case of the more challenging UW IOM dataset. This reinforces the intuition that both spatial

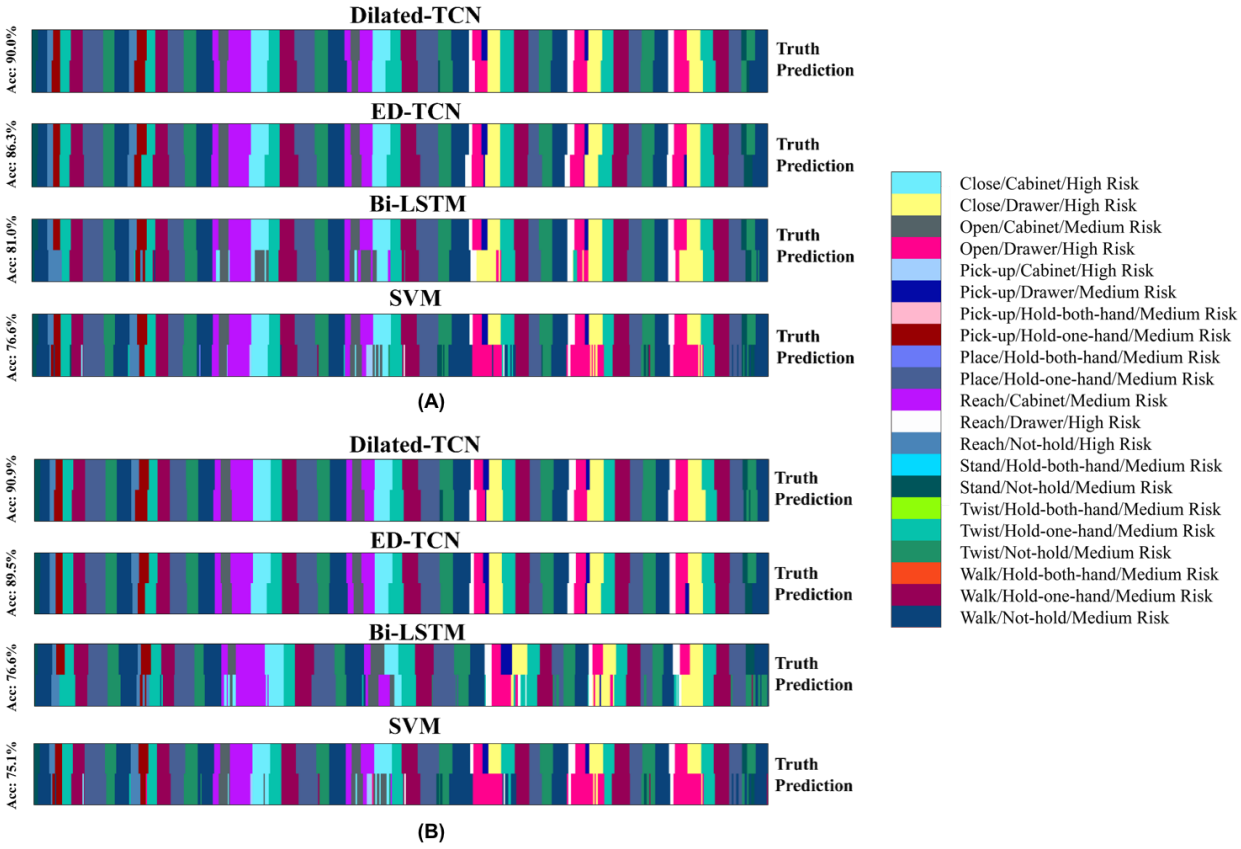


Figure 3.3: Performance comparison of various methods in action segmentation of a representative TUM Kitchen dataset video using (A) pre-trained VGG16 model and (B) fine-tuned VGG16 model. For each method, the upper row shows the ground truth (manually annotated) action labels, whereas the lower row depicts the corresponding predicted label.

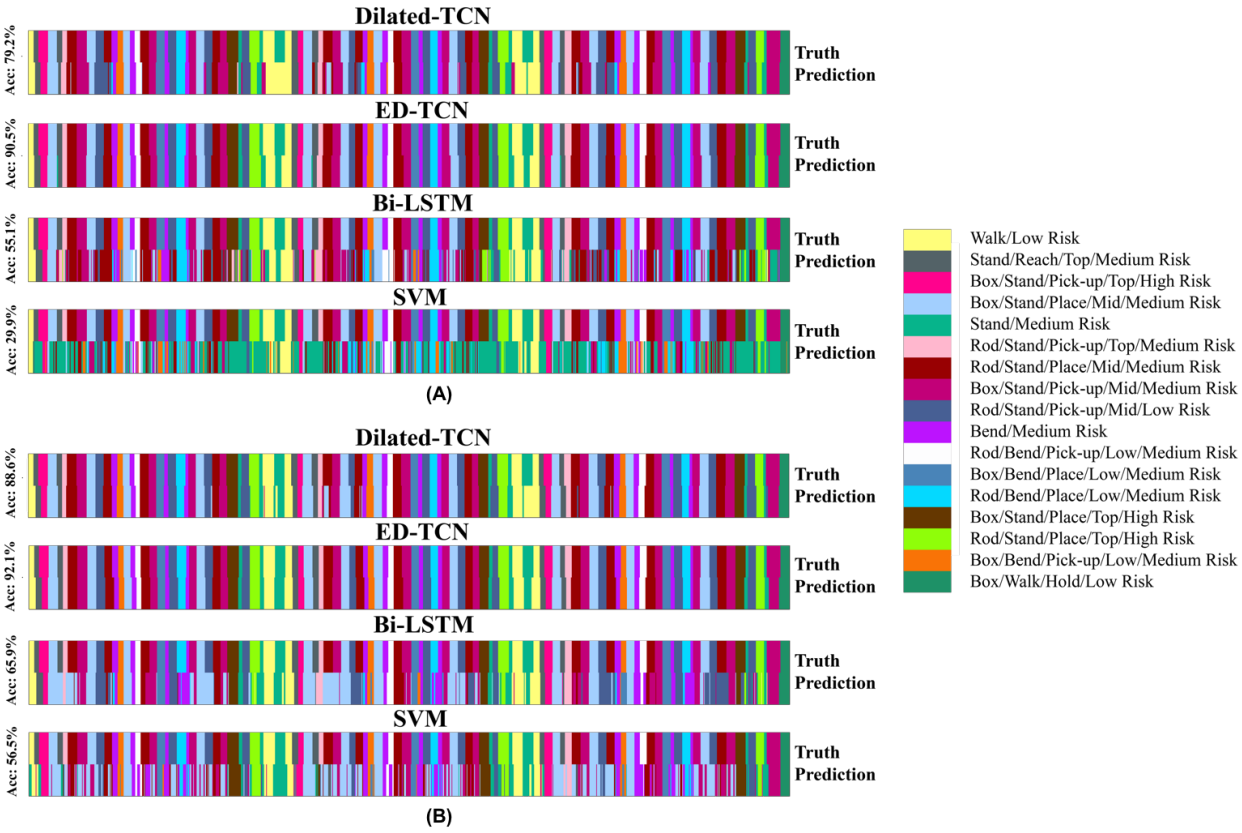


Figure 3.4: Performance comparison of various methods in semantic segmentation of a representative UW IOM dataset video using (A) pre-trained VGG16 model and (B) fine-tuned VGG16 model. For each method, the upper row shows the ground truth (manually annotated) action labels, whereas the lower row depicts the corresponding predicted label.

Table 3.2: Comparative performance measures of different action segmentation methods on the complete UW IOM dataset

Method		Accuracy (%)	Edit score(%)	F1 overlap (%)
Pre-trained VGG16	**D-TCN**	62.11±4.13	46.62±4.17	57.48±5.26
	ED-TCN	**78.76±3.65**	**82.96±3.33**	**87.77±2.51**
	Bi-LSTM	42.14 ± 5.45	23.76±1.50	29.71±3.72
	SVM	27.10 ±3.40	18.05±0.92	20.25±1.35
Fine-tuned VGG16	**D-TCN**	61.39±6.22	72.29±6.16	72.29±6.16
	ED-TCN	**86.46±0.50**	**88.52±1.17**	**93.24±0.58**
	Bi-LSTM	59.23±4.40	33.19±3.13	43.88±4.23
	SVM	42.10±3.33	20.61±0.89	27.56±1.92
Simplified P-CNN	**D-TCN**	81.72±2.82	74.01±5.13	82.23±4.80
	ED-TCN	**87.63±0.77**	**89.90±1.16**	**93.99±0.77**
	Bi-LSTM	71.38±4.97	75.33±7.41	80.45±7.55
	SVM	59.62±2.74	20.09±0.95	31.33±1.75

and temporal characteristics are important in analyzing long-duration human action videos.

It is not surprising to observe that the TCN methods perform better using edit score and F1 overlap score as the measure instead of global accuracy. As also reported in [64], accuracy is susceptible to erroneous and subjective manual annotation of the video frames, particularly during the transitions from one action to the next, where identifying the exact frame when one action ends and the next one begins is often open to individual interpretation. Both edit score and F1 score are more robust to such annotation issues as compared to accuracy, and, therefore, serve as better indicators of true system performance.

To further evaluate the general applicability of our action segmentation methods, We consider two additional test scenarios: TUM Kitchen videos taken from camera # 1, and a truncated UW IOM dataset comprising only one sequence of object manipulation actions per participant. Table 3.3 reports the action segmentation outcomes using just the fine-tuned VGG16 model since it yields better results than the pretrained VGG16 model on our regular test datasets. The trends are more or

Table 3.3: Comparative performance measures of different action segmentation methods on two additional video datasets.

Method		Accuracy (%)	Edit score (%)	F1 overlap (%)
TUM Kitchen with camera # 1	**D-TCN**	64.00±8.74	69.26±6.08	72.24±7.49
	ED-TCN	**69.83±4.66**	**84.69±3.49**	**83.43±3.09**
	Bi-LSTM	54.10 ± 9.37	40.64±8.06	48.33±9.44
	SVM	48.43 ± 8.87	30.34±6.25	38.46±7.88
UW IOM with non-repeated action sequence	**D-TCN**	74.04±2.53	62.91±3.32	72.91±3.01
	ED-TCN	**83.99±1.10**	**88.16±2.24**	**92.66±1.72**
	Bi-LSTM	58.93± 2.22	30.57±2.98	41.23±2.71
	SVM	40.62±1.72	21.05±1.94	26.98±1.92

less the same as in our regular datasets. The actual measures are almost identical for the complete and truncated UW IOM dataset. Thus, our methods seem to be robust to sample size, provided all the actions are covered adequately with a sufficient number of instances in the training set, and the actions occur in the same sequence in all the videos. The actual measures for our TCN methods are only slightly lower for the different TUM Kitchen dataset. Thus, performances appear to be independent of how the videos are recorded. The VGG16 validation accuracy is equal to 76.81% and 75.28% for the different TUM Kitchen and the truncated UW IOM dataset, respectively, which are, again, almost identical to the corresponding values for the regular TUM Kitchen and complete UW IOM datasets.

System Computation Times

In addition to characterizing the goodness of action segmentation, We am interested in knowing how long does it take to learn the spatial feature extraction models, to train the segmentation methods, and to compute the framewise action labels during testing.

The learning times for the pre-trained and fine-tuned VGG16 models are 20,844.11 seconds and 30,564.39 seconds, respectively, in case of the complete UW IOM dataset. As expected, the

learning time for the fine-tuned VGG16 model is somewhat lower and equal to 25,414.24 seconds in case of the truncated UW IOM dataset. For the TUM Kitchen dataset, the corresponding value is 31,753.18 seconds.

Using the fine-trained VGG16 model, in case of the complete UW IOM dataset, the overall training times are 252.73 $\pm$ 0.85, 237.76 $\pm$ 0.72, 2,172.23 $\pm$ 11.22, and 60.54 $\pm$ 1.54 seconds across the five data splits for the D-TCN, ED-TCN, Bi-LSTM, and SVM methods, respectively. The corresponding testing times are 0.10, 0.10, 1.09, and 0.09 seconds (the standard errors are negligible), respectively, for an average number of 8,261 frames, which implies that real-time action class prediction is highly feasible. These values are almost identical using the pre-trained VGG16 model. For the TUM Kitchen dataset, the overall training times are 91.19 $\pm$ 1.04, 74.53 $\pm$ 0.65, 619.21 $\pm$ 1.82, and 15.68 $\pm$ 0.67 seconds across the five data splits for the D-TCN, ED-TCN, Bi-LSTM, and SVM methods, respectively. The corresponding testing times are 0.03, 0.02, 0.33, and 0.03 seconds (negligible standard errors), respectively, for an average number of 6,311 frames.

Note that the TCN methods also have acceptable training times of the order of a few minutes for reasonably large datasets. This characteristic enables our system to adapt quickly to changing object manipulation tasks. On the other hand, the training times are considerably larger for Bi-LSTM, similar to the results reported in [64].

3.6 Discussion

In case of the more challenging UW IOM dataset, We observed that our TCN methods demonstrate better segmentation performance when spatial features are extracted using the fine-tuned VGG16 model instead of the pre-trained VGG16 model. Consequently, We decided to use P-CNN features to examine whether additional spatial features would further facilitate learning the temporal aspects of the videos for the action segmentation methods. As introduced in [20], P-CNN features are descriptors for video clips that are restricted to only one action per clip. All the frame features of a video clip are aggregated over time using different schemes that result in a single descriptor comprising information about the action in that clip. However, our goal is to process full-length videos with multiple actions. A single time-aggregated descriptor for an entire sequence of multiple

actions is not useful to us, as time aggregation results in the loss of important information about the sequence of actions as well as the transitions between the different actions. Hence, We skip the time aggregation step to obtain a video descriptor of the same length as the number of features in the full-length video.

Also, P-CNN features are originally generated by stacking normalized time-aggregated descriptors for ten different patches, i.e., five patches of the RGB image (namely, full body, upper body, left hand, right hand, and full image) and corresponding five patches of the optical flow image. These patches are cropped from the RGB and optical flow frames, respectively, using the relevant body joint positions. The missing parts in the patches are filled with gray pixels, before resizing them as necessary for the CNN input layer. This filling step is done using a scale factor available along with the joint positions for the dataset used in [20]. Such a scale factor is, however, not available for our TUM Kitchen and UW IOM datasets. On experimenting with various common values for this scale factor, We observe that it needs to be different for every video as each participant has a somewhat different body structure. Therefore, only the full image patches were used in the simplified form of P-CNN.

To further understand the potential impact and deployment challenges of our system, a proof-of-concept trials were performed, thank to Ekta SamanWe and Cameron Devine, using a Yaskawa HC10 collaborative robot equipped with an Intel RealSense D435 camera (see Fig. 5). Video recordings of ten human subjects (five males and five females), each performing the same set of seventeen actions as in the UW IOM dataset, are used to train the segmentation model consisting of the fine-tuned VGG16 and ED-TCN models. The RGB camera frames are captured and stored using the pyrealsense2 [49] library. While the videos are acquired at 30 Hz, only every third frame is retained. At run time, for two new test subjects (both males), the actions are segmented in groups of thirty frames at a time. Considering that ED-TCN requires the features for full-length videos to generate the predictions, We pad the feature vector by repeating the features of the 30^{th} frame. Such padding is done every time ED-TCN is given a new group of thirty frames to segment. The predictions are then communicated to a Programmable Logic Controller (PLC), which displays the outputs (action classes with ergonomic risk levels) on the robot teach pendant. This framework

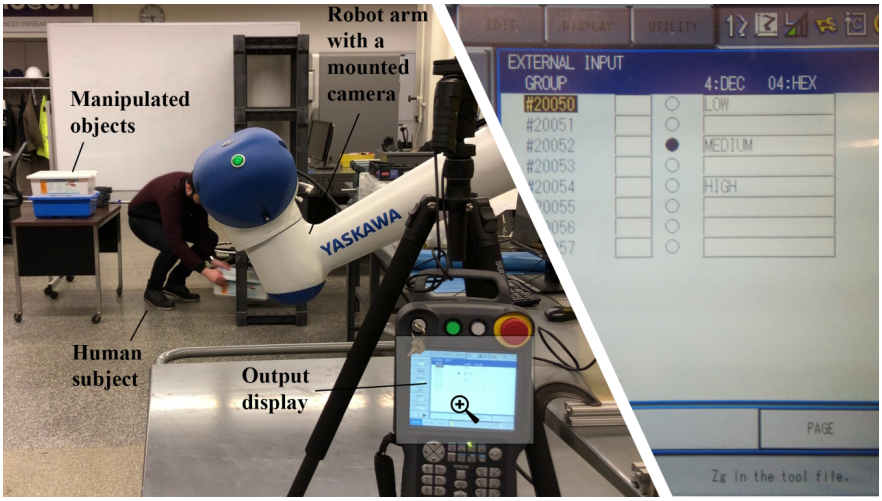


Figure 3.5: Output indicator corresponding to medium risk is turned on when a human subject performs a Box/Bend/Place/Low action.

predicts most of the test actions in the correct sequence with reasonably accurate action duration. However, the performance is worse than that in Section 3.5.2 due to frame downsampling and extra feature padding.

3.7 Summary

In this chapter, We presented an end-to-end deep learning system to accurately segment human actions and predict the corresponding ergonomic risks during indoor object manipulation using camera videos. This system comprises effective spatial features extraction and sequential feeding of the extracted features to temporal neural networks for real-time segmentation into meaningful actions. We believe that good overall performance is the cumulative effect of both the steps as is observed from our results for the different segmentation methods on the more challenging UW IOM dataset. This reinforces the intuition that both spatial and temporal characteristics are important in analyzing long-duration human action videos. The segmentation methods work well with just standard (RGB) camera videos, irrespective of how the spatial features are extracted, provided depth cameras are used to generate reliable ergonomic risk scores for all the possible actions

corresponding to a known object manipulation environment. Consequently, it makes our system useful for widespread deployment in factories and warehouses without requiring 3D cameras, body markers, and body-mounted sensors.

Despite the accurate performance of the proposed model, this method can be improved in multiple aspects. First, is to use less context-dependent features (VGG16 driven) to enhance the generalizability of the model. In Chapter 4, a multi-task learning framework is proposed that uses skeleton-based features for solving similar problem. Another opportunity for improvement is to move toward online-HAR (or early-HAR) and develop a learning method that would be capable of risk prediction on a frame-by-frame basis. This aspect is addressed in Chapter 5.

Chapter 4

MULTI-TASK LEARNING FOR ACTIVITY SEGMENTATION

We propose a new approach to Human Activity Evaluation (HAE) in long videos using graph-based multi-task modeling [98]. Previous works in activity evaluation either directly compute a metric using a detected skeleton or use the scene information to regress the activity score. These approaches are insufficient for accurate activity assessment since they only compute an average score over a clip, and do not consider the correlation between the joints and body dynamics. Moreover, they are highly scene-dependent which makes the generalizability of these methods questionable. We propose a novel multi-task framework for HAE that utilizes a Graph Convolutional Network backbone to embed the interconnections between human joints in the features. In this framework, we solve the Human Activity Segmentation (HAS) problem as an auxiliary task to improve activity assessment. The HAS head is powered by an Encoder-Decoder Temporal Convolutional Network to semantically segment long videos into distinct activity classes, whereas, HAE uses a Long-Short-Term-Memory-based architecture. We evaluate our method on the UW-IOM and TUM Kitchen datasets and discuss the success and failure cases in these two datasets.

4.1 Background

With the advancements in computer vision techniques, automated Human Activity Evaluation (HAE) has received significant attention. The aim of this category of problems is to design a computational model that captures the dynamic changes in human movement and measures the quality of human actions based on a predefined metric. HAE has been studied in a variety of computer vision applications such as sports activity scoring, athletes training [96, 103, 148], rehabilitation and healthcare [8, 95], interactive games [86, 162], skill assessment [27, 73], and workers activity assessment in industrial settings [99, 100]. Some of the earlier works on HAE used tradi-

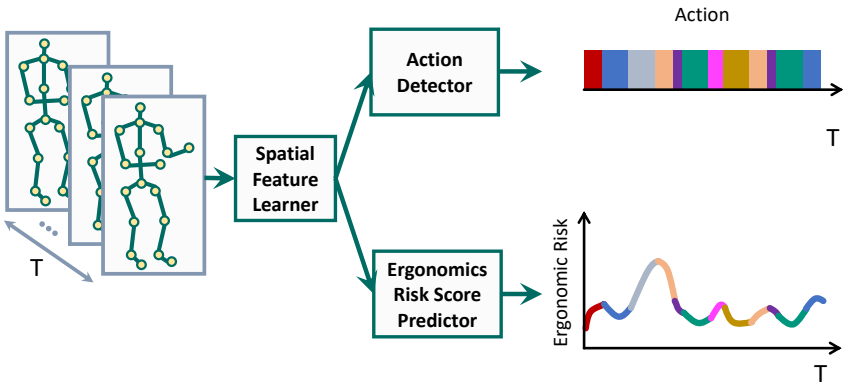


Figure 4.1: Multi-task activity segmentation and ergonomics risk assessment pipeline.

tional feature extraction methods for performance analysis [48, 107]. Recently, with the popularity of deep learning methods, a multitude of creative solutions have emerged for solving HAE problems. Among the proposed methods, some directly learn a mapping from images to a quality score [152]. As the activity quality is highly task-dependent a majority of research is focused on leveraging the available activity information in the learning process [96, 99]. Another approach has been to measure the deviation of a test sequence from a template sequence for determining the activity quality [94]. This approach is valuable when the performance of humans is evaluated based on how well they followed a fixed series of activities in a certain way such as in sport competitions or manufacturing operations.

There is another aspect of HAE that has received less attention despite its importance and potential impact on the safety and health of the society. Human Postural Assessment (HPA) is studied in various fields such as biomechanics, physiotherapy, neuroscience, and more recently in computer vision [79, 99, 100]. HPA is a subcategory of HAE that focuses on determining the quality of human posture using a ergonomics-based (or biomechanics-based) criteria. There are three major challenges in solving HPA problems: (1) the type of task and the object involved in the activity highly influence the risk level. (2) The repetition of certain movements can cause accumulated pressure on specific body parts. Therefore, it is important to analyze a video in a

frame-wise fashion to be able to capture repetition. (3) Everyone does not necessarily perform a task in the same way, hence, a successful algorithm should learn the relation between human joints dynamics and the corresponding ergonomics risk score.

This work is inspired by the importance of HPA problems and their significant impact on the health and safety of industrial workers. However, our approach is not limited to this specific application and it is a novel design that can benefit other aspects of HAE research. We leverage from consistent representation of human 3D pose and propose an end-to-end multi-task framework (Figure 5.2) that solves Human Activity Segmentation (HAS) as an auxiliary task to improve the HPA performance. Skeleton-based methods have been shown to provide the opportunity of developing more generalizable algorithms for various applications in HAR and prediction problems [117]. However, they have not been leveraged enough in HAE.

The work presented in this chapter, brings together activity segmentation and activity assessment using a novel multi-task learning framework. Our proposed framework comprises a Graph Convolutional Network (GCN) backbone and an Encoder-Decoder Temporal Convolutional Network (ED-TCN) for the activity segmentation head and a Long-Short-Term-Memory (LSTM)-based head for activity assessment. The contribution of our work is threefold. (1) We introduce a novel combination of GCN with ED-TCN for activity segmentation in long videos that outperforms state-of-the-art results on the UW-IOM dataset. (2) Our MTL-emb method initiates a line of research for more informed activity assessment by fusing activity embedding with spatial features for ERA. (3) We present a way to use the skeletal information for activity assessment in a Multi-Task Learning (MTL) framework that may enable generalization across a variety of environments and leverage anthropometric information.

4.2 Proposed Multi-Task Framework

In ERA, posture alone cannot accurately determine the risk level. The activity class contains information that is key to measure ergonomics risk. We, therefore, define HPA as a MTL problem consisting of an HAS and an HPA task (Figure 4.2). In the following sections, each component of our MTL model is described in details.

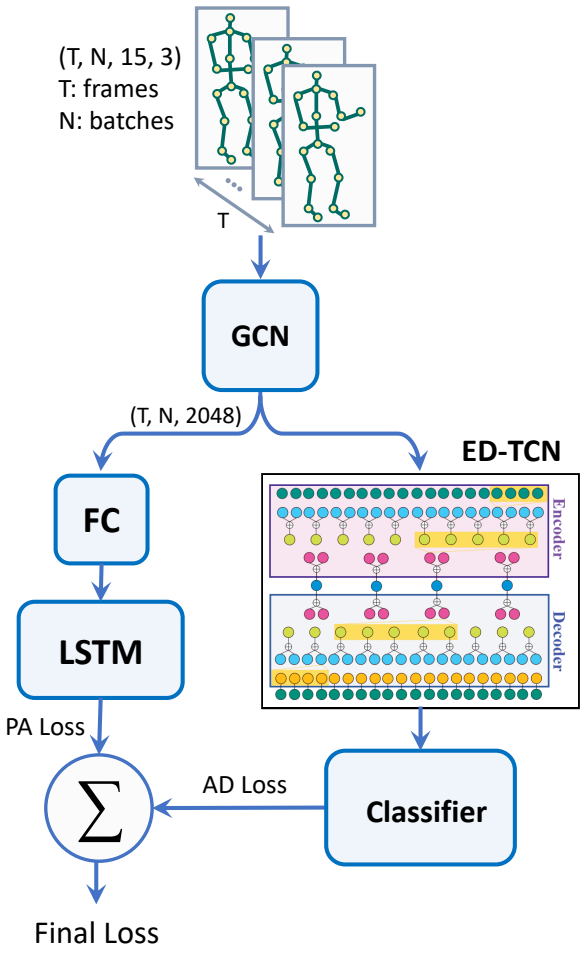


Figure 4.2: MTL network architecture.

4.2.1 Spatial Features

The inputs to our multi-task model are 3D joints locations, which is a form of structured data. Since GCNs are known to be powerful in representing structured data [166], our model uses a sequence of stacked GCNs as the backbone for spatial feature extraction, similar to the proposed structure in [156] except for temporal convolution. Just like a 2D convolutional layer, a stacked GCN allows better feature extraction for unstructured data such as graphs.

Given the input $\mathbf{x} \in \mathbf{R}^{D \times N}$, where D is equal to 3 as the joints are represented using (x, y, z)

coordinates and N is the number of joints, the adjacency matrix $\mathbf{A} \in \mathbf{R}^{N \times N}$, and the degree matrix $\hat{\mathbf{D}}$ with $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, a Graph Convolution (GC) can be written as,

$$\mathbf{f} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{x}^{\top} \mathbf{W} \ . \tag{4.1}$$

Here, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{I}$ is the identity matrix. For a graph with human skeletal structure, $\mathbf{A}$ is designed based on the anatomical connections among the joints. $\mathbf{W} \in \mathbf{R}^{D \times F}$ is the weight matrix that is to be learned. Hence, if the input to a GCN layer is $D \times N$, the output feature $\mathbf{f}$ is $N \times F$, where F is the chosen output feature size. In our proposed backbone, each GCN is followed by a ReLU activation. Moreover, the adjacency matrix is partitioned into three sub-matrices as described in [156] to better capture the spatial relations among the joints. Therefore, Equation (6.1) is written in a summation form for each GCN layer as:

$$\mathbf{f} = \sum_{a=1}^{3} \hat{\mathbf{D}}_a^{-\frac{1}{2}} \mathbf{A}_a \hat{\mathbf{D}}_a^{-\frac{1}{2}} \mathbf{x}^{\top} \mathbf{W}_a \ , \tag{4.2}$$

where a indexes each partition.

4.2.2 Encoder-Decoder Temporal Convolution for Human Activity Segmentation

In HAS problems, the task is to identify the activities that are happening in untrimmed videos and determine the corresponding initial and final frames [6, 32, 64, 100]. A popular approach that is inspired by works in audio generation and speech recognition [92, 153] is to use feed-forward (i.e., non-recurrent) networks for modeling long sequences. The main component of these methods is a 1D dilated causal convolution that can model long-term dependencies.

A dilated convolution is a filter that applies to an area larger than its length by skipping input values by a certain length [92]. A causal convolution is a 1D convolution which ensures the model does not violate the ordering of the input sequence. The prediction emitted by a causal convolution (that is $p(x_t|x_1, ..., x_{t-1})$) at time step t only depends on the previous data. Combining these two properties, dilated causal convolutions have large receptive fields and are faster than Recurrent Neural Networks (RNNs). Moreover, they are shallower than regular causal convolution due to dilation.

For the HAS task, inspired by [64, 92, 100] we design an ED-TCN-based on 1D dilated convolutions (Figure 4.2). Our design consists of a hierarchy of four temporal convolutions, pooling, and upsampling layers. The output of the ED-TCN followed by a Fully Connected (FC) layer and a `RelU` activation is fed to the classification layer.

In using ED-TCN for HAS [64, 100], the focus is on learning the temporal sequence and localizing activities. It is common to extract spatial features prior to training from an independent network like VGG16 [129] or ResNet [41]. Our proposed framework learns the spatial and temporal properties of the data in an end-to-end fashion. To our knowledge, this is the first attempt to use ED-TCN in an end-to-end architecture with a spatial feature detector. In addition, the combination of GCN with ED-TCN for solving HAS is a novel approach and it shows promising results.

4.2.3 Regression Module for Human Postural Assessment

We define HPA as a sub-category of HAE where the activity score is determined based on the safety of the posture. In HPA, the task is to find a mapping between the spatio-temporal features and ergonomics risk score. Our proposed regressor uses the shared spatial features coming from the GCN backbone. The GCN features go through a FC layer with `tanh` nonlinearity and are then fed into a stacked LSTM structure to predict the REBA scores.

4.2.4 Multi-Task Approach to ERA

MTL is a popular framework for end-to-end training of a single network for solving multiple related tasks. In these networks, a common backbone provides the data representation for branches responsible for learning a specific task. Usually in MTL, there is a main task plus multiple auxiliary tasks that complement the core task. For instance, in HAE, the main task is to determine the action quality. However, action quality is not independent of what action is carried out, which makes the HAS choice of auxiliary tasks natural for this kind of problems.

The supervision signals from the auxiliary tasks can be viewed as inductive biases [16] that limit the hypothesis search space and result in a more generalizable solution. The multi-task ap-

proach to HAE has been recently introduced by [96] for determining the quality of action in short clips from Olympic games.

In our work, the main task is to predict the REBA scores. However, the information about human action is closely related to its corresponding ergonomics risk. Therefore, the auxiliary task in this case is the HAS. The long duration videos pose an additional challenge, since, unlike most of the HAE datasets, both the activities and their risk scores vary over time. In a majority of sport HAE [96], a single activity score is predicted for a clip. Here, the HAS task consists of 17 and 20 actions for the UW-IOM and TUM datasets, respectively (see Section 4 for more information on the datasets). Therefore, in any video, activity localization and ERA task involves predicting a smooth function that shows how the risk is changing throughout the video.

We studied two different architectures for solving this MTL problem. In the first architecture, the heads corresponding to each task only share the GCN-driven features. In the second architecture, the output of the *Softmax* layer of the HAS head is fused to the feature going to the LSTM regressor.

We consider a weighted average of the HAS loss and the HPA loss as the overall multi-task HPA loss function,

$$\mathcal{L}_{HPA} = \sum_{t=1}^{T} \alpha(\mathbf{x}_t - \mathbf{y}_t)^2 + \beta|\mathbf{x}_t - \mathbf{y}_t|, \tag{4.3}$$

where $\mathbf{y}_t$ is the frame-wise ground truth REBA score and $\mathbf{x}_t$ is the model prediction. $|\cdot|$ is the $\mathcal{L}_1$ norm. α and β are weights to be learned. For HAS, we use cross-entropy loss between ground truth and model prediction,

$$\mathcal{L}_{HAS} = -\sum_{t=1}^{T}\sum_{c=1}^{Cl} \mathbf{y}_{t,c}\log(\mathbf{x}_{t,c}), \tag{4.4}$$

where Cl is the number of classes. The overall loss is the sum of all the losses,

$$\mathcal{L}_{MTL} = \mathcal{L}_{HPA} + \gamma\mathcal{L}_{HAS}, \tag{4.5}$$

where γ is to be learned.

4.3 Experiments

4.3.1 Datasets

Despite the impact of automated ERA on industry, research in this area has started gaining popularity only recently. As a result, only a few datasets are available that capture representative activities in industrial settings. In particular, two such datasets have been used in recent publications in this domain.

UW-IOM Dataset is a publicly available dataset of 20 videos by [100] that captures industry-relevant activities. This dataset has 17 action classes and labels are of four-tier hierarchy indicating the object, human motion, type of object manipulation (if applicable), and the relative height of the surface on which the activity is taking place. The longest video in this dataset has $2,384$ frames. We use the 3D poses for UW-IOM dataset from our earlier work [99].

TUM Kitchen Dataset has 19 videos consisting of daily activities in a kitchen. Learning with graph-based methods has been shown to be challenging on this datasets due to the similarity of human postures in multiple action classes [99]. We took labels provided by [100] so that we can compare our results with theirs. We used [104] to extract the 3D poses from the videos recorded by the second camera. The longest video in this dataset has $2,048$ frames.

The input features to our model are 3-dimensional key-points (x, y, z) of N *=15* joints, concatenated over time T. Hence, the resulting input tensor is of dimension $3 \times 15 \times T$. The output ground truth labels are frame-wise labels that have the dimension of $1 \times T$.

4.3.2 Ergonomics Risk Pre-processing

REBA method [44] computes a score describing the total body risk based on the joint angles and the properties of an action. The REBA scores are discrete integers from 1 (the minimum risk level) to 15 (the maximum risk level). In [100], the scores of all the subjects are averaged over the classes and a single score is reported for each activity class. We used the detected skeletons to compute the joint angles and obtained a frame-wise REBA score. However, the REBA profile then becomes a sequence of piece-wise constants, which is hard to learn by a regressor. Therefore, we smoothed

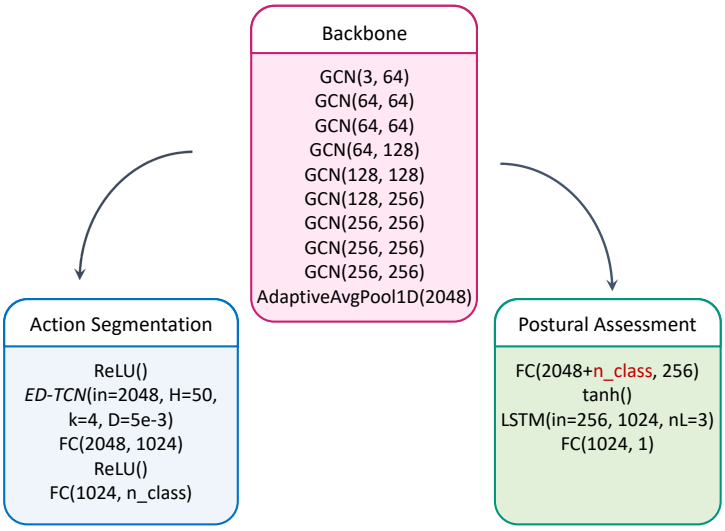


Figure 4.3: Detailed MTL-emb architecture. $GCN(in, out)$ is a GCN with edge-importance. ED-TCN has 4 hidden layers of size H with kernel size k and dropout of D. $FC(in, out)$ is a fully connected layer. n_{class} is the number of classes. The LSTM has nl layers.

the REBA sequence using the Python `UnivariateSpline` function to make it easier for the ERA regressor to learn the patterns. To help advance research in this area, the smoothed REBA scores along with the code are available on the project repository[1].

4.3.3 Implementation Details

All the networks were implemented in PyTorch [101]. The initial values of the loss function weight parameters α, β, and γ were set to 1. All the networks were trained using the Adam optimizer [56]. We implemented early-stopping and trained the model with different learning rates to find the best one (the best performing learning rate is shown in Table 4.3). The 20 videos in the UW-IOM dataset were randomly split into 15 and 5 for the training and validation set, respectively. For the TUM dataset, the training and validation sets include 15 and 4 videos, respectively.

[1] https://github.com/BehnooshParsa/MTL-ERA

GCN Backbone: The details of the GCN network is displayed in Figure 4.3. The output of the final GCN layer is of size $(N, T, 256, 15)$ that is flattened to $(N, T, 3840)$ and passed through an adaptive pool layer. Therefore, the feature that is fed to the rest of the network is of size $(N, T, 2048)$.

Action Segmentation Head: ED-TCN requires input batches to have the same temporal length. Hence, we defined a maximum length in both the training and validation sets, and masked the rest of the inputs with a value of -1 (thus, T corresponds to the maximum sequence length). The predicted sequence was unmasked before calculating the loss. The ED-TCN output goes through two fully connected layers with `RelU` activation and is used to compute the cross-entropy loss.

Postural Assessment Head: We evaluated the performance of two architectures for HPA. In one design, we fuse the Softmax output of the HAS head to the GCN features and call this model, *multi-task-emb*. The base design does not include fusion and we refer to that as *multi-task-base*. The spatial features (from the GCN backbone) are followed by a fully connected layer with `tanh` activation and sent to three layers of LSTM. The LSTM output is followed by a fully connected layer to predict the REBA scores and is sent to the regression loss function.

4.3.4 Evaluation Metrics

To measure the performance of the HAS network we use *F1-overlap score*, *segmental edit score*, and *Mean Average Precision* (MAP). F1-overlap score is essentially the harmonic mean of $Precision$ and $Recall$ and is computed using the following well known formula:

$$F_1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{4.6}$$

Edit score measures the closeness of the predicted sequence to the ground truth sequence. This metric penalizes if the order of the sequence and the number of action segments are not correct. The average precision is computed over all the classes and its mean is reported.

4.4 Results and Discussion

To evaluate the strength of our proposed multi-task approach in solving the HAS and HPA problems, we carry out two single-task experiments for the HPA task (STL-PA) and the HAS task (STL-AS). Another reason behind the STL-AS experiment is to investigate the power of our GCN model as a spatial feature extractor in solving HAS problems.

The STL-PA network has identical GCN backbone and LSTM design as the MTL network. The average MSE result is reported for the validation set in Table 4.1. It is clear from the results that the network cannot learn the sophisticated pattern of the REBA profile.

UW-IOM		TUM	
MSE	Sp. Corr. (%)	MSE	Sp. Corr. (%)
1.68 ±0.28	11.79 ±12.32	2.75 ±0.40	62.92±4.89

Table 4.1: Average MSE and Spearman's Coefficient of the activity score prediction over the validation videos using the STL-PA model.

4.4.1 Action Segmentation with GCN-ED-TCN

As discussed in Section 2, ED-TCN along with the input features derived from pre-trained networks, have been widely used for HAS. The idea is that given the input spatial features for every time-step of a sequence, this method can segment it into semantically similar pieces. Nonetheless, an end-to-end approach for learning both the spatial and temporal features in an HAS has not been explored with ED-TCN. While GCN models have been used both for activity classification [54, 156] and early action recognition [99], its capability has not been evaluated for HAS.

ED-TCN is used for HAS on the UW-IOM dataset in [100], where the authors compare three spatial feature extractors, namely, a pre-trained VGG16 on ImageNet [24], a fine-tuned version of VGG16 model, and a P-CNN model [20]. Our proposed GCN backbone extracts spatial features based on human pose only, but its performance is comparable with the state-of-the-art as shown in Table 4.2. Hence, we believe that pose-based features are more suitable for designing a gener-

Method	UW-IOM			TUM		
	mAP (%)	Edit score (%)	F1 overlap (%)	mAP (%)	Edit score (%)	F1 overlap (%)
ED-TCN / Pre-trained VGG16 [33]	-	88.52 ± 1.17	93.24 ± 0.58	-	86.34 ± 3.15	87.92 ± 2.16
ED-TCN / Fine-tuned VGG16 [33]	-	82.96 ± 3.33	87.77 ± 2.51	-	84.96 ± 4.37	87.29 ± 2.78
ED-TCN / Simplified P-CNN [33]	-	89.90 ± 1.16	93.99 ± 0.77	-		
GCN-ED-TCN (STL-AS)	49.61 ± 0.17	92.08 ± 1.18	92.33 ± 0.78	24.17 ± 11.99	67.53 ± 5.16	52.20 ± 22.02

Table 4.2: mAP, edit, and F1-overlap score represented using mean and standard deviation values over the test videos in the UW-IOM and TUM datasets for different methods and modalities solving the HAS task.

alizable algorithm. However, we should emphasize that generalizability comes with a price of the model not performing well when the pose information is poor or when the activities require similar postures, which is the case for the TUM dataset (Table 4.2).

Method	UW-IOM					
	MSE	Sp. Corr. (%)	mAP (%)	Edit score (%)	F1 overlap (%)	Learned Weights
MTL-base	0.72 ±0.14	66.68 ±4.89	76.0 ±8.51	88.36 ± 4.67	89.56 ±4.45	CrE: 0.70, MSE: 0.81, L1: 0.51 lr: 0.001
MTL-emb	0.61 ±0.36	55.18 ±6.57	74.45 ±10.36	91.59 ±1.23	92.03 ±2.54	CrE: 0.72, MSE: 0.85, L1: 0.64 lr: 0.001
TUM						
MTL-base	1.03 ±0.48	80.44 ±4.67	36.83 ±16.63	64.75 ±13.10	54.15 ±19.01	CrE: 0.95, MSE: 0.97, L1: 0.95 lr: 1e-04
MTL-emb	1.01 ±0.38	73.83 ±8.00	39.23 ±17.00	65.87± 9.13	58.24 ±11.23	CrE: 0.86, MSE: 0.89, L1: 0. 86 lr: 0.0005

Table 4.3: Results for the MTL network. mAP, edit, and F1-overlap scores are represented using mean and standard deviation values over the validation splits in the UW-IOM and TUM datasets for different activity segmentation methods and modalities. MSE and Spearman's coefficient show the model's performance in predicting the activity risk scores.

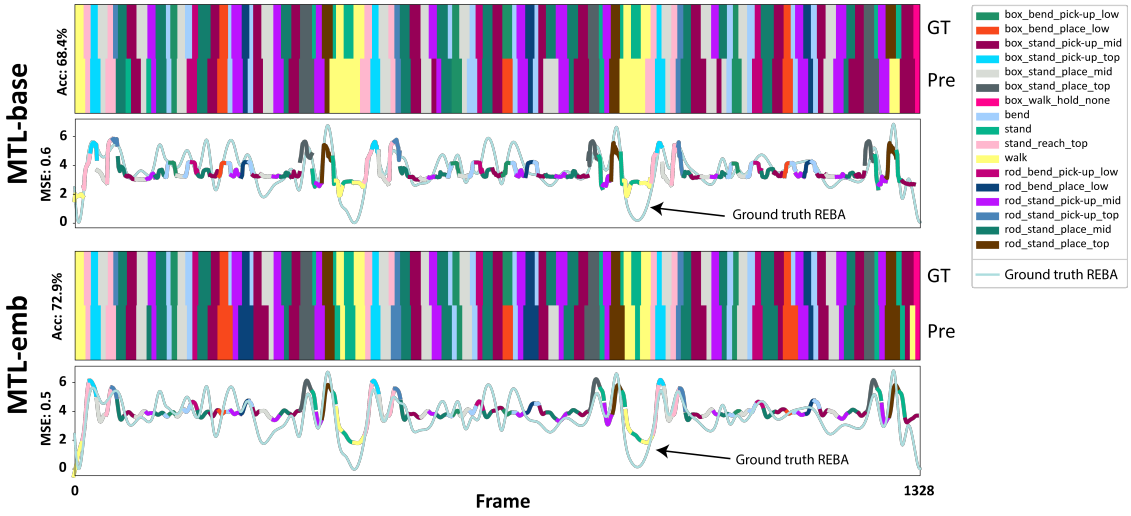


Figure 4.4: Visualization of HAS and REBA prediction result for a sample test video of UW-IOM dataset. The first and third plots (colored ribbons) are the segmentation results. In each ribbon the top-half is the ground truth and the bottom-half is the predictions by the network. The second and fourth plots depicting the ground truth REBA score and the network prediction. The network prediction is color-coded based on the activity class.

4.4.2 Single-task vs. Multi-task Approach

The substantial improvement in predicting the activity risk scores is evident when comparing the results in Table 4.1 with Table 4.3. We believe that the underlying reason behind this observation is that the REBA score is highly dependent on the type of activity, and learning an auxiliary HAS task can enhance the performance of the HPA head. However, the inverse dependency is not that strong. Our findings indicate that the STL-AS performs better than the MTL approach for HAS (comparing results in Table 4.2 and 4.3).

4.4.3 Fusion vs. No Fusion Approach

The main purpose of this experiment is to validate the idea that action information can improve REBA score predictions. Table 4.3 and Figure 4.4 show the MTL-base and MTL-emb results,

where improvements are observed when the HPA head has access to the `Softmax` output of the HAS head. In Figure 4.4, we see the highly nonlinear ground truth REBA scoreline (in solid light blue-green) and the corresponding predictions for each action by both the MTL networks. The figure suggests that the network with embedding predicts the REBA scores more accurately. On the contrary, the shared embedding model does not significantly improve the performance of the HAS head. Figure 4.5 depicts the difference in the confusion matrices of the two models. For simplicity, the off-diagonal elements are ignored. While there are small improvements in a few classes, the overall improvement is not substantial.

4.4.4 Failure Cases

Although we show that our MTL-emb and STL-AS methods perform well on the UW-IOM dataset and even better than using context heavy features such as VGG16, these models are not particularly successful on the TUM dataset. We present the confusion matrices for the UW-IOM and TUM datasets in Figure 4.6. In the following, we describe our insights on the performance of the models in detail.

The camera view in the TUM dataset is from the top. As a result, arm pose estimation quality is poor for the activities where the person's back is facing the camera and the arm is occluded such as for *pickup-drawer* and *close-drawer*. Another source of confusion is between *Pickup-hold-**both**-hands* and *Pickup-hold-**one**-hands* due to the fact that the poses are very similar.

Since the segmentation head is not very successful on the TUM dataset, the improvement in the REBA score prediction between the MTL-emb and MTL-base models is also not significant unlike in the case of the UW-IOM dataset. For the TUM dataset, fusing image-based features with the GCN can be potentially useful in decreasing the ambiguity in the GCN spatial descriptors, thereby, improving both the STL-AS and MTL results for REBA score prediction.

4.5 Summary

We introduce a graph-based multi-task learning approach for Human Postural Assessment and show that it outperforms the equivalent Single-Task Learning due to the importance of the ac-

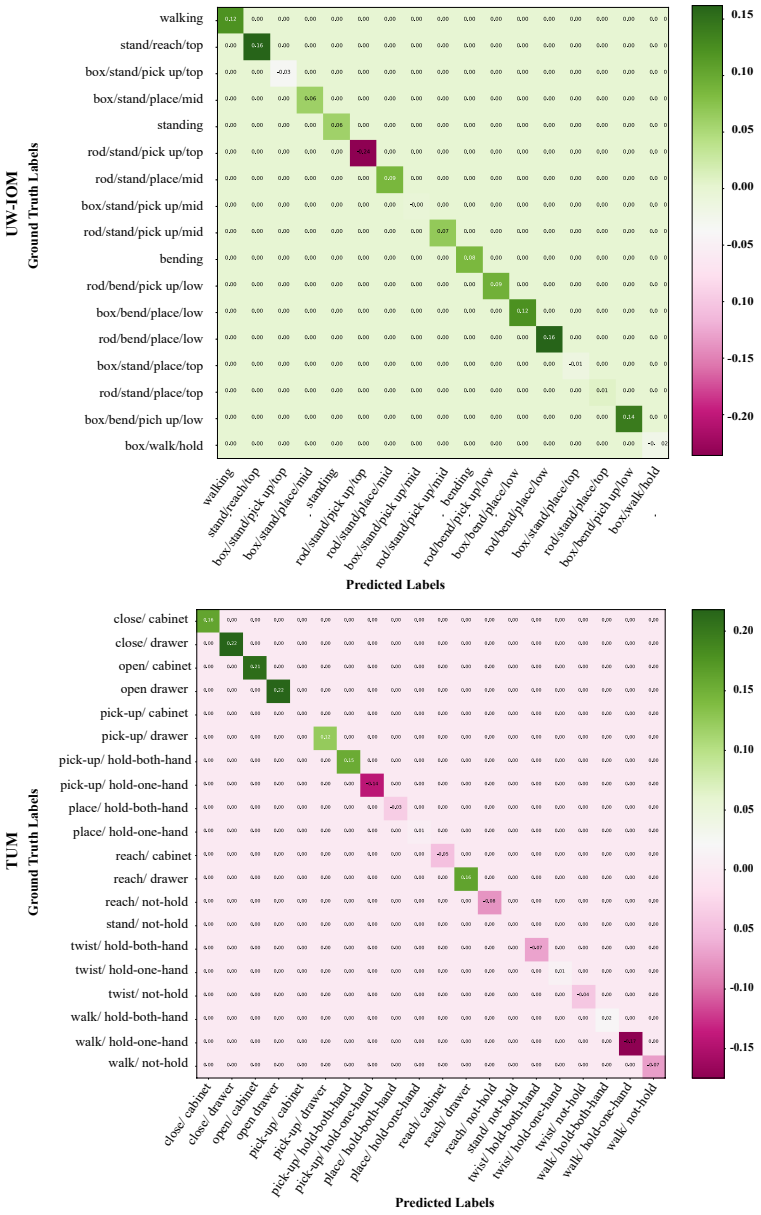


Figure 4.5: The difference in confusion matrices. The top and bottom matrices are for the UW-IOM and TUM dataset, respectively. The diagonal elements show the differences between the diagonal values of the MTL-emb and MTL-base confusion matrices and the off-diagonal elements are shown as ”0.0” for simplicity.

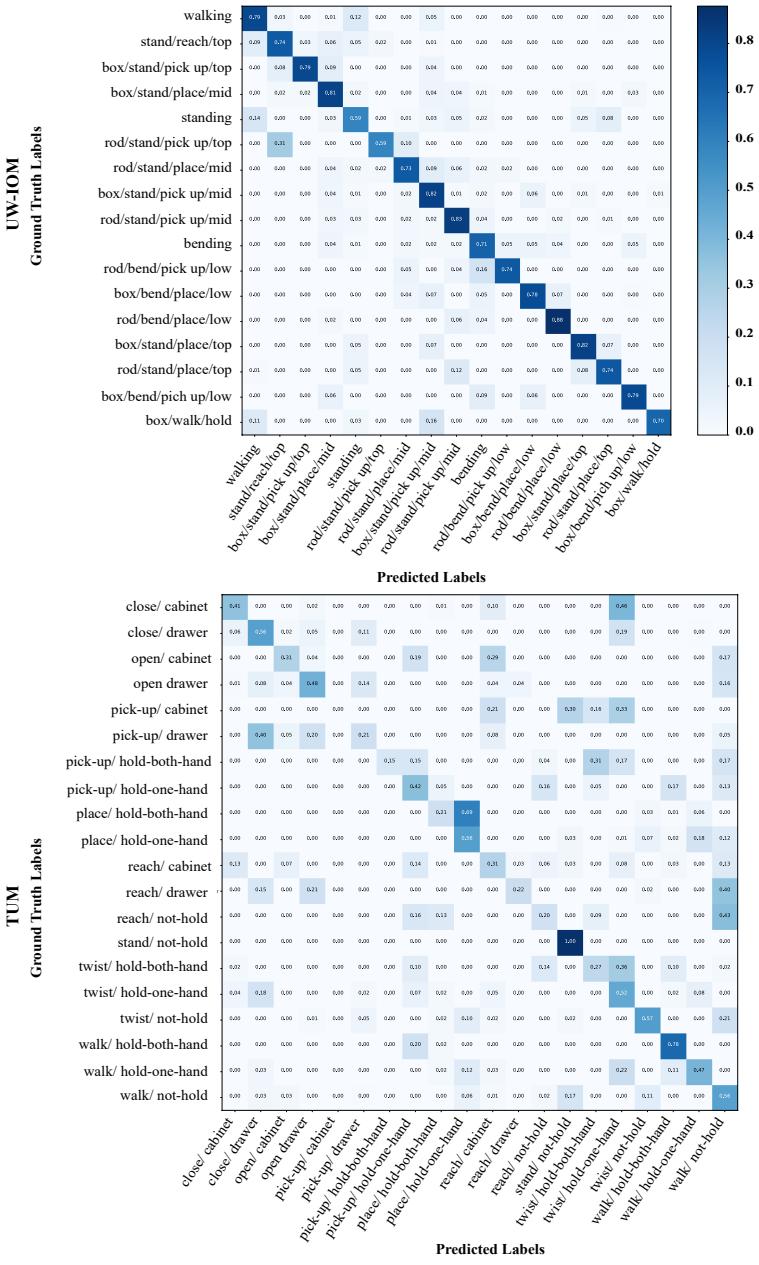


Figure 4.6: Confusion matrices using MTL-base. The top and bottom matrices are for the UW-IOM and TUM dataset, respectively.

tivity type in the risk associated with a posture. Human Postural Assessment tasks, specifically Ergonomics Risk Assessment, are more challenging than regular Human Activity Evaluation problems since the assessment has to happen in a frame-wise manner and is highly dependent on joint kinematics. Despite the challenge of tracking the intricacies of our risk assessment (REBA) profile, the proposed method shows competence in predicting the risk scores. More importantly, our work demonstrates the effectiveness of the GCN model as a spatial feature extraction backbone, compared to context-based features that have been traditionally used with ED-TCN for Human Activity Segmentation tasks. To showcase the weaknesses of this framework, we implemented our method on a challenging dataset (TUM) and discussed the failure cases.

Part II

EARLY HUMAN ACTIVITY RECOGNITION

Chapter 5

SPATIO-TEMPORAL PYRAMID GRAPH CONVOLUTIONAL NETWORK

Recognition of human actions and associated interactions with objects and the environment is an important problem in computer vision due to its potential applications in a variety of domains. Recently, graph convolutional networks that extract features from the skeleton have demonstrated promising performance. In this paper, we propose a novel Spatio-Temporal Pyramid Graph Convolutional Network (ST-PGN) for online action recognition for ergonomics risk assessment that enables the use of features from all levels of the skeleton feature hierarchy. The proposed algorithm outperforms state-of-art action recognition algorithms tested on two public benchmark datasets typically used for postural assessment (TUM and UW-IOM). We also introduce a pipeline to enhance postural assessment methods with online action recognition techniques. Finally, the proposed algorithm is integrated with a traditional ergonomics risk index (REBA) to demonstrate the potential value for assessment of musculoskeletal disorders in occupational safety.

5.1 Background

Human action recognition has been a widely studied research topic in computer vision for several decades. The task is to infer the human action and activity from still images or video frames. Solutions to this important and challenging problem have traditionally been applied to domains such as surveillance, entertainment, robotics, video retrieval, and intelligent driving assistance systems [91, 108, 165]. Recently, there are emerging applications that involve assessment of human performance for virtual fitness, health monitoring, training, and ergonomics risk assessment for occupational safety [38, 100, 111]. These applications have unique requirements that may involve simultaneous association of time varying pose with action and object interaction, and relating such

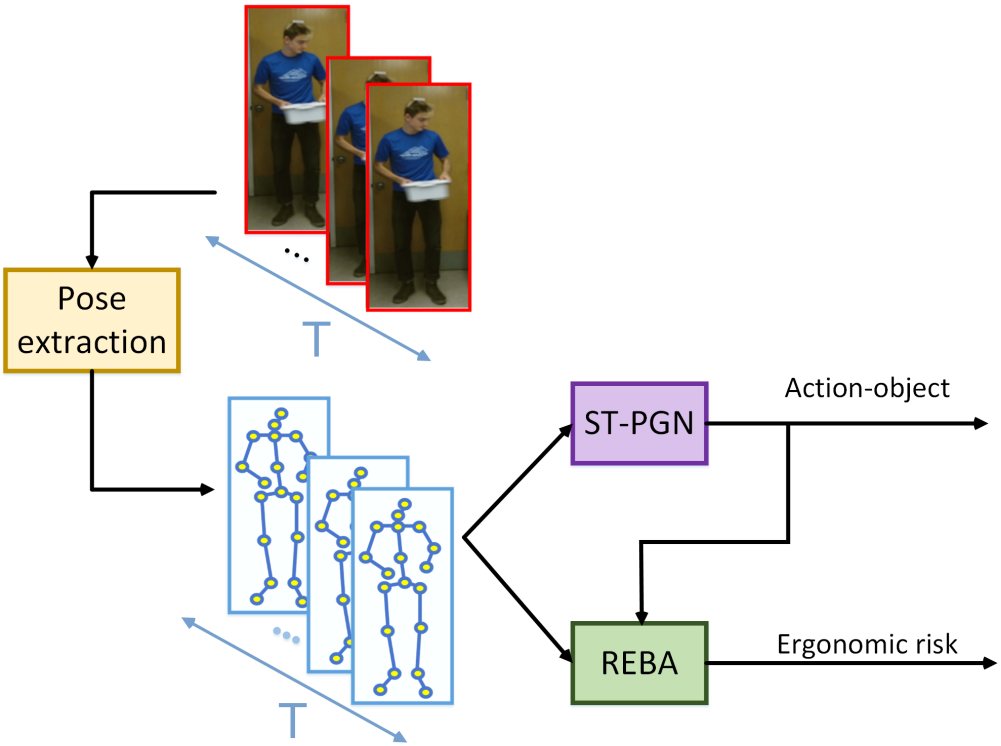


Figure 5.1: Our model (ST-PGN) takes a sequence of skeleton input produced by a pose extraction unit (like LCR-Net [117]) and does early action recognition. The skeleton sequence along with the activity labels go to the REBA computation unit to assess the ergonomics risk while testing.

information for computational modeling and prediction of various biomechanical indicators. Vision only systems are non-invasive and less expensive alternatives to study these problems as opposed to expensive drift prone motion capture systems and wearable sensors [21, 84]. Depending on the application, human action recognition can be formulated in an online or off-line setting. In most applications, processing is performed off-line, making use of the entire video sequence without strict limitations on computational resources. In such cases, the typical assumption is that the start and end points of the action is known [28, 83] and the training video is pre-segmented into various action classes. Recent advances in hardware and GPU performance has led to the emer-

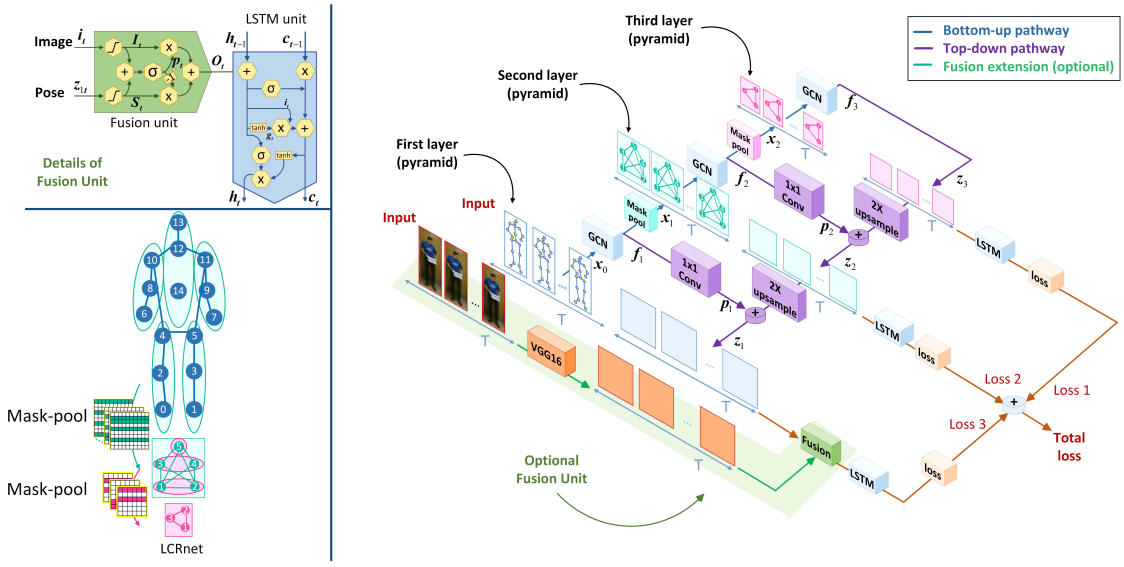


Figure 5.2: The Feature Pyramid Convolutional Graph Network pipeline.

gence of many online applications, where the requirement is to process video streams in real-time and without a priori knowledge of the transitions between actions [77, 78, 131]. Generalization of action recognition algorithms is a challenging and unsolved problem. Ideally, the method should generalize to various environments and deal with cluttered backgrounds, occlusions, and viewpoint variations. While end to end video to action classification have shown great promise, generalization is achieved through domain adaptation [13, 40] or using intermediate skeletal representations that are robust to these variations [116, 156]. In particular, skeleton-based features appear to produce favorable results since human pose is typically a consistent representation of action across people and context. Among them, recent work based on graph convolutional networks that extract meaningful features from the skeleton have achieved good performance [69, 156].

The work in this paper is inspired by emerging applications involving human performance assessment in various domains including health, fitness, rehabilitation, and occupational safety. In particular, we consider specific challenges for real-time ergonomics risk assessment in complex environments such as manufacturing assembly. The requirements include correlation of action with the time varying posture and associated ergonomics and biomechanical risk. The ultimate

goal is to produce reliable estimates of pose, action, and associated ergonomics indicators in order to identify the risk of musculoskeletal disorders associated with acute and repetitive tasks.

In this chapter, we propose a novel real-time Spatio-Temporal Pyramid Graph Convolutional Network (ST-PGN) for action recognition that enables the use of features from all levels of the skeleton feature hierarchy (Figure 5.2). ST-PGN, designed with a feature pyramid architecture enables the model to capture the correlation between body parts, rather than hand-coding body-part relations. We test the performance of the model on two public benchmark datasets typically used for postural assessment (TUM and UW-IOM) as well as Kinetics and NTU-RGBD datasets. We show that the algorithm is also able to learn the transitions between actions and is suitable for real-time applications. As compared to the state-of-the-art algorithms such as ST-GCN [156], our model has fewer graph convolution kernels without sacrificing performance. Finally, we enhance the pipeline with postural assessment methods (REBA [43]) that use the online action recognition output of our model to produce ergonomics risk estimates. We propose, this combined action-risk architectural design as a first step towards automated assessment of musculoskeletal disorders in occupational safety.

5.2 Proposed Spatio-Temporal Feature Pyramid Graph Convolution

In this work we introduce Saptio-Temporal Pyramid Graph Convolutional Network (ST-PGN). ST-PGN models the spatio temporal features of the skeletal structure using combinations of Pyramidal GCNs (PGNs) and Long-Short-Term-Memory Units (LSTMs). PGN is a novel way to process non-Euclidean skeletal data in a hierarchical form. Each feature representation in PGN hierarchy is used as an input to an LSTM unit to learn the temporal aspect of the input sequence (shown in Figure 5.2 and described in Section 5.2.4).

5.2.1 Graph Convolutional Network

Graph convolutional networks (GCN) [166] learn the layer-wise propagation operation that can be applied on structured data represented by a graph. To briefly introduce how GCNs work, assume we have an undirected graph with N nodes, a set of edges between nodes, an adjacency matrix

$\mathbf{A} \in \mathbf{R}^{N \times N}$, and a degree matrix $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. If $\mathbf{x} \in \mathbf{R}^{F \times N}$ represents the feature matrix of the graph ($\mathbf{x}_i \in \mathbf{R}^F$ is the feature vector of node i with size F), a linear formulation of graph convolution is,

$$\mathbf{f} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{x}_i{}^{\top} \mathbf{W}, \tag{5.1}$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{I}$ is the identity matrix and $\mathbf{W} \in \mathbf{R}^{F \times C}$ is the weight matrix. So, if the input to a GCN layer is $F \times N$ the output feature $\mathbf{f}$ is $N \times C$, where C is the chosen output size. As with any other convolution layer we can have a stack of GCNs each followed by a nonlinear function (such as ReLU) [57].

In this work, we are following the spatial configuration partitioning introduced in ST-GCN [156], therefore, $\hat{\mathbf{A}} = \sum_a \mathbf{A}_a$ and Equation 6.1 is written in a summation form.

$$\mathbf{f} = \sum_a \hat{\mathbf{D}}_a^{-\frac{1}{2}} \mathbf{A}_a \hat{\mathbf{D}}_a^{-\frac{1}{2}} \mathbf{x}^{\top} \mathbf{W}_a, \tag{5.2}$$

Equation 6.2 is represented for k^{th} level of the pyramidal hierarchy in line 3 of Algorithm 1. We hypothesize that a hierarchical graph convolution that operates on human joints, body parts and global structure would enrich the input representation.

5.2.2 Pyramidal Graph Architecture

Pyramidal Graph Convolutional Network (PGN) is a hierarchical GCN that produces different spatial features with semantic meaning at different levels. The input to the PGN is the skeleton with $\mathbf{N}$ joints represented by a tensor ($\mathbf{X}$) of dimension $F \times N \times T$, where T indicates time. Each GCN aggregates features along the spatial dimension using a specific adjacency matrix $\hat{\mathbf{A}}_k$ using Equation 6.2. Our PGN has three graph levels ($\hat{\mathbf{A}}_1$, $\hat{\mathbf{A}}_2$, $\hat{\mathbf{A}}_3$). The initial GCN works on the skeleton with $\hat{\mathbf{A}}_1$, which is constructed based on the skeleton connections and accompanied with an edge-importance matrix [1]. The subsequent graph levels represent the body parts and global structure respectively. Since the correlation between the nodes for higher level graphs is unknown,

[1] Edge-importance is a learnable mask on every layer of the spatio-temporal graph convolution. It has the same dimension as the adjacency matrix and learns to scale the contribution of a node's feature to its neighboring nodes based on the learned importance weight of each spatial graph edge.

$\hat{\mathbf{A}}_2$ and $\hat{\mathbf{A}}_3$ represent fully connected graphs and we let the edge-importance learn the correlations. Thus our model has a hierarchy of graphs with the base as the input skeleton and the top level a graph with three nodes representing arms, legs and middle part of the body. We refer to this hierarchical graph structure as a pyramidal graph architecture because it is large at the base and becomes smaller as we move to the top levels.

Symbol	Legend
N	Number of 3D Skeleton joints,(x, y, z) tuples
T	Time history of 80 samples
$\mathbf{c}_k$	Graph convolution(GCN) at each hierarchy k
$\hat{\mathbf{A}}_k$	Adjacency matrix at each hierarchy k
$\mathbf{X}_0$	Input Skeleton feature to first GCN ($3 \times N \times T$)
$\mathbf{g}_k$	Group Average Pool at each hierarchy k
$\mathbf{J}_k$	Pooling kernel at each hierarchy k
$\mathbf{f}_k$	Output of each GCN ($3 \times N \times F$)
$\mathbf{w}_k$	1×1 convolution operation
$\mathbf{u}_k$	Upsample and Add
$\mathbf{p}_k$	Output of 1×1 convolution
$\mathbf{z}_k$	Final features sent to LSTMs

Table 5.1: Description of the symbols used in Algorithms

5.2.3 Group Average Pool

A Group Average Pool (GAP) layer average-pools the features in a selected group of nodes or joints using a specific kernel ($\mathbf{J}_k$) for each level (line 4 of Algorithm 1). The resulting graph has nodes that represent a higher level body part (as shown in Figure 5.2). Therefore, every layer of the pyramid has a semantic meaning, from low to high level. In the bottom left corner of Figure 5.2, we show how the groups are defined in TUM and UW-IOM datasets.

More specifically, feature masking is inspired by [23] which is generally used in foreground background separation. Here, kernels are pre-determined matrices with ones or zeros. These

kernels are element-wise multiplied by the features to group only certain body parts one at a time. For example, the kernel has ones in the particular rows corresponding to those joints representing left arm $(7, 9, 11)$ and zero everywhere else. Hence, the masked features (features multiplied by the mask) all belong to the left arm. These features are average pooled as they belong to the same group. Multiple such combinations are used to group the joints into different parts. Similarly , parts are combined into global structure using another set of kernels. Such successive GCN-GAP combinations allows us to model the entire local and global motions jointly. We refer to this as the feature update rule (Algorithm 1), and later in Section 5.2.4, it is referred to as a bottom-up pathway.

5.2.4 Feature Pyramid Graph Convolutional Network

Feature pyramids have been an important component of object recognition algorithms [41, 75, 127]. The advantage of using pyramids is that it produces a multi-scale feature representation in which all feature levels are semantically strong. Especially in skeleton-based action recognition the correlation of body-parts can be very informative in recognizing actions. However, a pre-defined graph might not be sufficient to represent every sample. For example, in ST-GCN graph, there is no connection between hand and head, which is important in actions such as eating. Therefore, here we are generalizing the feature pyramid network to a GCN pyramidal feature hierarchy, and we believe that learning the correlations at different levels of the hierarchy enhances the performance of our model. Here feature pyramids are still valid in skeleton structure as global motion is a combination of local motion of parts and part motion is a combination of local motion of joints. Hence our feature pyramids aggregate joints, parts and global features jointly [155].

The feature pyramid networks consist of two pathways, a bottom-up and a top-down pathway. The **bottom-up pathway** is the feed-forward computation of the backbone GCN, which computes a feature hierarchy consisting of feature maps at different scales. The **top-down pathway** produces higher resolution features by up-sampling spatially larger, but semantically stronger, feature maps from higher pyramid levels. The top-down path is enhanced by the features produced in the bottom-up pathway through lateral connections. The features from the bottom-up pathway undergo a 1×1

Algorithm 1 Feature Update Rule

1: $\mathbf{X}_0 \leftarrow \mathbf{X}$ {input skeleton distributed over time}

2: $k \leftarrow 1$ {iterator} $k \leq 3$

3: $\mathbf{f}_k = \boldsymbol{c_k}(\mathbf{X}_{k-1}; \hat{\mathbf{A}}_k)$ {GCN operation}

4: $\mathbf{X}_k = \begin{cases} \boldsymbol{g_k}(\mathbf{f}_k; \mathbf{J}_k) & \text{if } k < 3, \\ None & \text{otherwise.} \end{cases}$ {GAP operation}

5: k=k+1

{The $\mathbf{f}_k \forall k \in (1, 2, 3)$ are used as input features for the feature pyramid operations in Algorithm 2.}

Algorithm 2 Pyramid Update Rule

1: $k \leftarrow 1$ {iterator} $k \leq 3$

2: $\mathbf{p}_k = \begin{cases} \boldsymbol{w_k} \otimes \mathbf{f}_k & \text{if } k < 3, \\ \mathbf{f}_k & \text{otherwise.} \end{cases}$ {1×1 convolution}

3: $\boldsymbol{z}_k = \begin{cases} \boldsymbol{p}_k \oplus \boldsymbol{u}_k(\boldsymbol{p}_{k-1}) & \text{if } k < 3, \\ \boldsymbol{p}_k & \text{otherwise.} \end{cases}$

4: {Upsample & Add}

5: k=k+1

{The Following $\boldsymbol{z}_k \forall k \in (1, 2, 3)$ are used as input features for the temporal modelling using three separate LSTMs.}

conv layer to reduce channel dimensions and then are merged into the top-down pathway features by element-wise addition. The purple connections in Figure 5.2 shows this process, and it is described as the pyramid update rule.

5.2.5 Spatio-Temporal Modelling

Now we briefly summarize ST-PGN steps that are described in Algorithm 1-2 and Figure 5.2, and also describe major differences with respect to ST-GCN. The input skeleton ($\mathbf{X}_0$) goes through three levels of GCN and GAP, and the output of each level ($\mathbf{f}_k$) is aggregated with the upsample features through lateral connection and forms the final features ($\mathbf{z}_k$). Each pyramidal feature is passed through separate LSTMs to create three frame-wise activity predictions. As an ablation

study we either 1) average these three predictions and compute one loss or 2) compute three losses separately and average the predictions while testing. The latter gives us better performance. As a comparison, in ST-GCN, the input goes through a sequence of multiple GCN and TCN units so that the final feature embodies spatial and temporal properties of the input. A final feature that summarizes spatial and temporal properties is the key for video clip classification. However, when we need to recognize activities frame-wise, that strategy fails as it is shown in Section 6.4.1. Therefore, we extract the spatial features through PGN and send these features to individual LSTM units so that the temporal aspect is learned at different spatially semantic layers.

5.2.6 Optional Fusion Unit

To study the benefit of image features, we also perform experiments with image features concatenated along with skeleton pose features. We hope to avoid confusion in situations with object handling. Hence we extract VGG16 features from a crop image region around the human and fuse them with the final skeleton feature pyramid. Our fusion unit is inspired by GRU [25], that learns to weight the features before LSTM. We freeze the weights of the pre-trained network and only train the fusion unit along with the final LSTM layer. While the benefit of the image features are very minimal, for completeness, we will describe the fusion unit below.

At time t, let the image features and the final feature pyramid layer features be denoted by i_t and z_{1t}, respectively. Since the dimensions of these features do not match, we apply linear weights ($\boldsymbol{U_i}$, $\boldsymbol{U_z}$) to transform them into the same dimension and arrive at the transformed image and skeleton features I_t and $S(t)$ as shown in Equation 5.3. The terms W_i and W_z are learnt weights that are used to learn a gauging value (p_t) between the two features similar to the GRU. The weight p_t is squished to take on values $\in [0, 1]$ using a sigmoid operation. Finally this weights are multiplied to the incoming features.

$$I_t = \text{relu}(\mathbf{U}_i * i_t), S_t = \text{relu}(\mathbf{U}_z * z_{1t}) \tag{5.3}$$

$$p_t = \sigma(\mathbf{W}_i * I_t + \mathbf{W}_s * S_t), \tag{5.4}$$

$$\mathbf{O}_t = p_t I_t + (1 - p_t) S_t \tag{5.5}$$

Modalities	Backbones	UW-IOM			TUM		
		mAP (%)	Edit (%)	F1-overlap (%)	mAP (%)	Edit (%)	F1-overlap (%)
Skeleton (only)	*Frame based*	39.82 ± 1.45	29.26 ± 1.32	37.87 ± 1.82	29.79 ± 4.74	27.55 ± 2.89	32.63 ± 4.66
	LSTM [45]	79.35 ± 4.55	77.82 ± 6.34	85.32 ± 5.37	44.24 ± 5.97	56.46 ± 5.92	57.13 ± 8.24
	TCN [6]	57.72 ± 6.40	56.40 ± 5.36	64.78 ± 6.38	30.61 ± 5.40	51.07 ± 6.17	49.87 ± 11.01
	ED-TCN [64]	60.05 ± 4.89	81.73 ± 2.44	84.60 ± 2.64	28.89 ± 5.77	**56.75 ± 8.50**	55.92 ± 11.11
	ST-GCN [156]	66.94 ± 3.49	61.89 ± 3.56	71.08 ± 2.83	34.73 ± 5.98	53.88 ± 5.53	53.52 ± 7.09
	ST-GCN+IMP [156]	73.28 ± 4.30	67.21 ± 6.05	76.58 ± 4.95	34.93 ± 4.75	52.27 ± 3.99	52.60 ± 5.72
	GCN+LSTM+IMP	81.97 ± 7.34	72.25 ± 7.24	82.04 ± 6.08	45.92 ± 4.19	52.07 ± 4.01	55.26 ± 5.54
	ST-PGN+LSTM (ours)	86.33 ± 2.71	77.92 ± 2.44	86.83 ± 1.74	48.02 ± 4.68	55.31 ± 5.09	57.58 ± 6.38
	ST-PGN+LSTM+IMP (ours)	85.92 ± 1.62	77.75 ± 2.46	86.21 ± 1.91	42.74 ± 1.03	47.19 ± 6.39	51.14 ± 6.94
	ST-PGN+LSTM+IMP+ML (ours)	**87.03 ± 2.85**	**79.86 ± 2.15**	**87.95 ± 1.54**	**49.62 ± 6.10**	56.10 ± 4.98	**57.60 ± 6.03**
Image (only)	*Frame based*	51.62 ± 4.12	25.60 ± 1.55	34.17 ± 3.08	35.33 ± 5.26	28.33 ± 1.94	35.34 ± 2.65
	LSTM	66.50 ± 7.55	48.31 ± 5.90	57.81 ± 6.64	49.04 ± 7.03	52.64 ± 7.50	**58.60 ± 7.53**
Fusion	*Frame based+ Concat*	50.54 ± 1.55	27.57 ± 0.96	36.42 ± 2.09	41.70 ± 5.76	29.66 ± 1.25	36.04 ± 1.59
	LSTM+ Concat	83.55 ± 5.74	72.98 ± 7.32	77.89 ± 11.70	48.71 ± 9.42	**54.86 ± 6.83**	57.11 ± 8.81
	ST-PGN+LSTM+IMP+ML+GRU-Fusion (ours)	**87.05 ± 3.47**	**80.90 ± 2.06**	**88.08 ± 1.89**	**57.79 ± 6.43**	54.49 ± 5.59	58.35 ± 9.78

Table 5.2: mAP, edit, and F1-overlap score represented in mean and standard deviation over five splits in UW-IOM and TUM datasets for different methods and modalities. The best results in skeleton and fusion modality are shown in bold.

Where, $\mathbf{O}_t$ is the **weighted** feature that is sent as input into one LSTM unit. For Example, If p_t is 0.6 then the image features (I_t) is weighted higher and the skeletal features (S_t) are weighted lower ($1 - 0.6 = 0.4$).

5.3 Experiments

5.3.1 Datasets

In skeleton-based action recognition the skeletal structure is represented as a graph. For our vision only system, we use state of the art 3D skeleton estimation LCR-Net [117] to estimate poses for the TUM Kitchen and UW-IOM dataset. While the focus of our work is to evaluate our proposed method on an online setting , we also run experiments on Skeleton Kinetics and NTU-RGB datasets as shown in Appendix Section I.

UW-IOM Dataset is a new dataset introduced in [100] with the intention of capturing activities

that are common in warehouses. It consists of twenty videos (with average rate of twelve frames per second) of a sequence of object manipulation. The duration of every video is approximately three minutes. This dataset represents seventeen action classes and labels are of four-tier hierarchy indicating the object (box/rod), human motion (walk, stand, and bend), type of object manipulation if applicable (reach, pick-up, place, and hold), and the relative height of the surface where manipulation is taking place (low, medium, and high).

TUM Kitchen Dataset [136] consists of nineteen videos of a sequence of kitchen activities from four monocular cameras with the rate of twenty-five frames per second and the average duration of two minutes (we used camera 2). We use the provided two-tier labels by [100], which includes a motion verb (place, reach, stand), and a location (cabinet, drawer) or object manipulation mode (both-hands, one-hand) and creates a total of twenty-one activity classes.

5.3.2 Implementation Details

In our experiments, we sample a fixed length T=80 frames from each skeleton sequence as the input for online experiments. For offline experiments (NTU dataset and Skeleton Kinetics) we set the length T = 150 to cover the entire sequence for one label. We set the batch size to 128 and 32 for online and offline experiments respectively. In order to compare fairly with ST-GCN, the graph partitioning for the first adjacency matrix ($\hat{\mathbf{A}}_1$) is set to the same spatial strategy and partitioned into 3 subsets: the root node itself, centripetal group, and centrifugal group. However for the subsequent graphs $\hat{\mathbf{A}}_2$ and $\hat{\mathbf{A}}_3$ we assume that fully connected graph as initialization (all nodes are connected to every other node) and learn the edge importance weighting.

It should be noted that we do not modify the original ST-GCN model in terms of number of GCN or parameters. Our final model has only three GCN layers as opposed to the ten GCN-TCN components. More specifically the first GCN layer has 64 channels, second GCN has 128 and third has 256 channels. During training, we use the Adam optimizer [11] to optimize the network. We set the betas to 0.9 and 0.999 and set weight decay to zero. We split the training and validation using a **five** fold split in both TUM and UW-IOM. We report he mean and variance of all the splits in the results Table 6.1. We also do a grid search for learning rate(lr) from 0.1 to 0.001. On an

average, lr of 0.05 performs best on all the splits in both datasets.

5.3.3 Ergonomics Risk Assessment

Given the input skeleton ($\mathbf{X}$) and the recognized activity from PGN (Figure 5.1), we compute REBA. REBA assigns human posture scores, in the range 1-15, based on joint angles during an activity. First, a risk score is computed for lower and upper extremities and those scores are added to the task-related scores (coupling and load scores). In [100] the score over all subjects are averaged offline and one score is reported for each activity class. We are proposing a real-time subject specific REBA evaluation using pose and action information such as the weight of the object and the type of manipulation.

5.4 Results and Discussion

5.4.1 Baseline Models

GCN vs Non-GCN Methods. To see the benefit of temporal analysis we perform experiments that only take human skeleton or image as input. We use these features as inputs of a *TCN* [6], *ED-TCN* [64] and *LSTM* [45] model. Baselines are trained in an online fashion. We also perform frame based experiments to determine the efficacy of temporal modelling. It must be noted that no additional convolution or linear layers are used to transform the input pose.

ST-GCN variants. We showcase the original ST-GCN implementations modified to support online setting by removing the final average pooling layer. Most ST-GCN variants used for spatio-temporal modelling, support recursive GCN-TCN models that pool messages across the overall graph of full skeleton. We also replace the 1x1 TCN convolutions with LSTMs. We refer to this model as *GCN+TCN*. Edge Importance, as in the original work, is trained and is showcased as *ST-GCN+IMP*. LSTMs generally outperforms the TCNs to capture short transition changes in online fashion. Hence for the following experiments we choose to use LSTM as a primary temporal modelling source.

ST-PGN variants. Our models are showcased as pyramid-GCN (PGN) models. Similar to the previous section, we choose to train the edge importance for each of the sub graphs. The

predictions are averaged for *ST-PGN+LSTM* and *ST-PGN+LSTM+IMP* and used to compute a single loss. Alternatively, our final multi-loss (ML) model has three losses, one for each of the pyramids. These losses are averaged and propagated during training. During testing the model's predictions are averaged and used for evaluation. The results for this model is shown in Table 6.1 under *ST-PGN+LSTM+IMP+ML*.

Fusion Models. To evaluate the impact of adding contextual features, we use a fusion mechanism that learns the importance of each feature modalities through a gauging mechanism (p_t in top-left of Figure 5.2).

5.4.2 Performance Analysis

The UW-IOM dataset focus is on object manipulation tasks that involve picking up, placing, and carrying objects, as well as walking bending and standing. Therefore, when we look at the edge importance demonstrations in the top left of Figure 5.3, we see that left hand (L-hand), right hand (R-hand) and right hip (R-hip) are the most important nodes in the low-level edge importance heat-map. Also, at the high-level, the importance of arms is higher than the legs and spine. We achieve an overall +5% improvement in mAP, +2% improvement in F1-overlap (*ST-PGN+LSTM*) over the best baseline (*GCN+LSTM* and *LSTM*). However, we see an overall performance boost of +16% in Edit score and similar to our multi-loss model. Importantly, ST-PGN is more powerful in distinguishing pick-up and place. These activities are spatially very similar and differ primarily in temporal aspects. We do not see a huge benefit in Edit score using our image fusion. However, we see a minor improvement of 1% in the mAP and F1-overlap.

TUM kitchen dataset also includes object manipulation activities; however, it is focused on common daily activities in a kitchen. Looking at the low-level heat-map (top right in Figure 5.3) we observe that the hand, elbow, shoulder, and the neck joints have more importance. Looking at the high-level demonstration, we observe that the arms are more important than the legs and spine. We observe an overall improvement in mAP and F1-overlap using our models. However, a simple ED-TCN is slightly better at capturing the sequence and hence the Edit score is higher. Since the subjects move around in the scene, the significance of the lower body (*legs, hip*) is visibly higher

in the edge importance compared to UW-IOM.

The results reported in Table 6.1 show that using skeleton is sufficient to get equal or better performance as compared to image-only or the fusion of skeleton and image. In UW-IOM, the human is facing the camera; thus, the detected skeleton is accurate. However, since this is not the case in the TUM dataset, the image-based models perform better as compared to the skeleton only on TUM dataset. If the skeleton is accurate, the addition of the image does not seem to enhance the results significantly in these tasks.

5.4.3 Failure Cases

We showcase confusion matrices of our best *ST-PGN+LSTM+IMP+ML* model in Figure 5.4 of the previous section. While we see an overall performance increase on both UW-IOM and TUM datasets, the model cannot deal with confusion among similar classes. We showcase the skeleton only model as adding image features do not help significantly. We describe our insights in detail below.

UW-IOM Dataset Our models can differentiate between *box-handling* actions and *rod-handling* actions without the use of image features (The skeleton configuration differs and handling of these objects is distinct due to the object size and location). However, *Standing* and *walking* misclassifications occur especially when the subject's back faces the camera. Hence important hand motions that help to infer these actions are missed. Better self-occlusion handling is warranted. Confusion also occurs between *bending* actions such as *bending-place*. This is predominantly due to misclassifications in transitions between these actions since bending is followed by pickup action or preceded by place action. Since it is a challenge for human annotators to accurately label transitions, the edit score should avoid penalizing such transitions. **TUM Dataset** The camera view angle contributes to significant confusion between related classes. We choose the training-validation split with the lowest mAP score to analyze the results. The following observations are made:

1) The *pickup-drawer* and *close-drawer* are completely misclassified in this split. Once the drawer is closed, the pose estimation predicted the hand orientation and location using LCR-Net occlusion strategy [117]. However, the predicted pose is not always reliable, resulting in poor

performance due to incorrect pose input during training. 2) *Walk-not holding* is misclassified for the majority of the classes such as *reach-cabinet*, *reach-drawer*, *stand-hold-both-hands*, *stand-not-hold*. This is attributed to unbalanced class distribution, where most of the actions are walking. In future work, we plan to address biases introduced by data imbalance by introducing sampling strategies. 3) *Twisting* actions are very challenging to detect using vision only since we only measure poses in Cartesian coordinates. Adding rotation information should help the model detect twisting actions about certain body axes. 4) *Pickup-hold-both-hands* gets confused with either *Pickup-hold-**one**-hand* or *stand-hold-**both**-hands*. Confusion is primarily due to one hand either being occluded by the object being handled, or the pose configuration being too similar in pose configuration with *standing*. More key-points in the pose prediction models could help resolve such issues.

5.5 Summary

We proposed a novel Spatio-Temporal Pyramid Graph Convolutional Network (ST-PGN) for online action recognition. The method integrates the following: a) basic prior knowledge about the skeletal structure, b) hierarchical joint relationships and c) data-driven learning framework for online action based ergonomics risk assessment. The proposed approach addresses the simultaneous association of time-varying pose with action and objects interaction to enable downstream applications that involve computational modeling and prediction of various human performance metrics for ergonomics risk assessment.

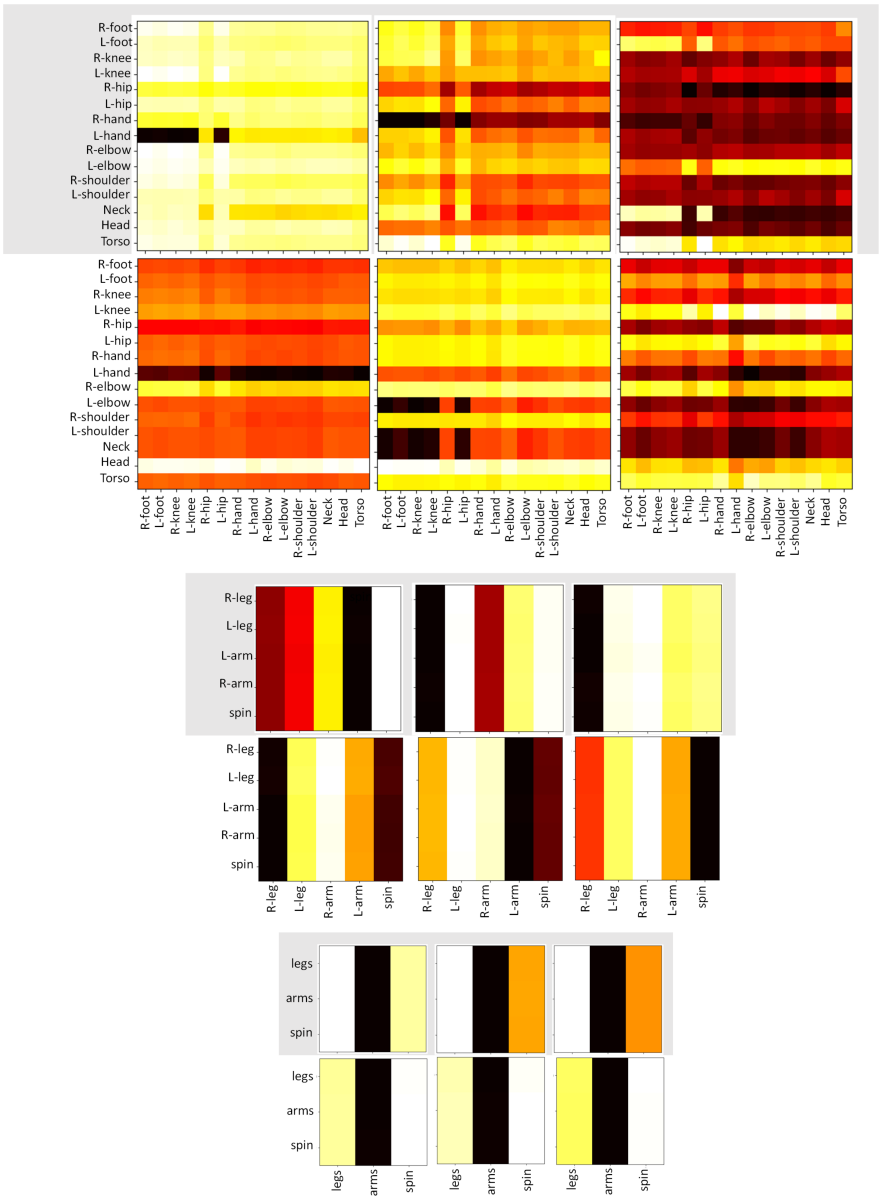
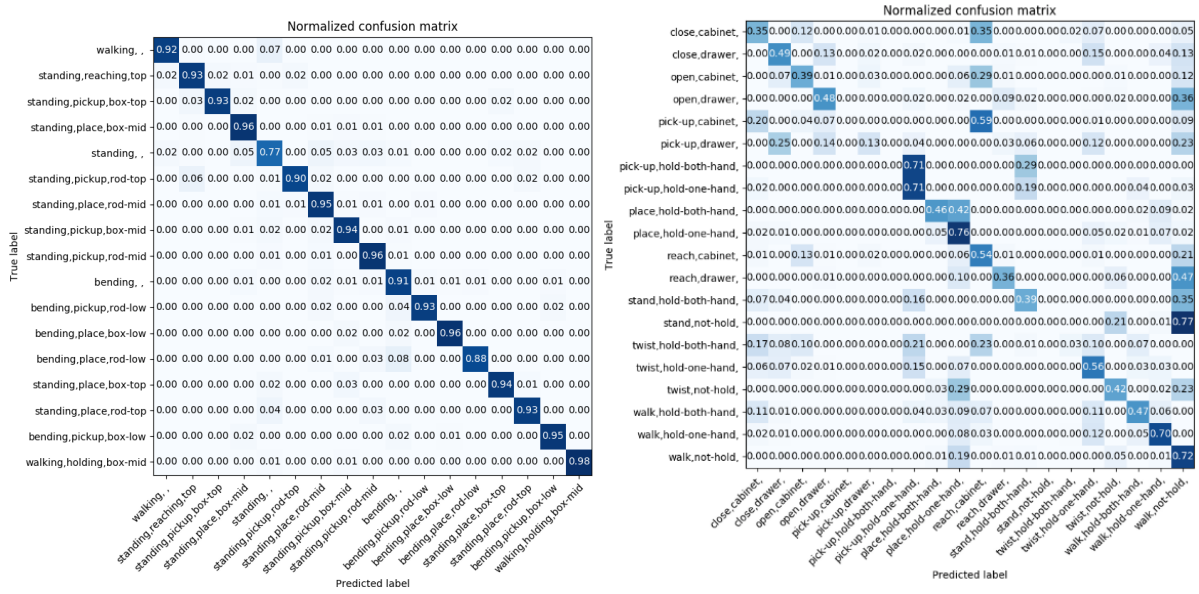


Figure 5.3: Three level learned edge importance heat-map in UW-IOM (shaded) and TUM datasets. Each row shows the edge importance of each level of graph pyramid and it is consistent with bottom-left of Figure 5.2. Every level of PGN consists of the sum of three edge importance multiplied by the adjacency matrix and node features.



(a) UW-IOM (b) TUM

Figure 5.4: Confusion Matrix of *ST-PGN+LSTM+IMP+ML* model. Larger figures are added in the Appendix section.

Chapter 6

SPATIO-TEMPORAL HIERARCHICAL GRAPH CONVOLUTIONAL NETWORK

In a HAR framework, objects or other components of the scenes are often the prime differentiating factors among human activities and such distinction becomes even more crucial in real-world applications of HAR, such as safety analysis or industrial process fidelity monitoring. This has led to a recent surge in using HOI graph representations. However, designing an algorithm based on context-specific features affects generalizability, whereas using fewer context-sensitive ones such as skeleton-based features has shown promising results. GCNs have been successfully used in action recognition problems for modeling human-object and human joints interactions. Nevertheless, the employment of both graphs for action recognition is unexplored in the literature. We propose a ST-HGCN for early action recognition in long videos. Our method is flexible in handling datasets with a variable number of objects in the scene as well as an unknown adjacency matrix. We evaluate our algorithm on the UW-IOM dataset and show that our method significantly outperforms the previous results on this dataset with faster convergence.

6.1 Background

The majority of human activities involve some level of interactions with the objects in the surrounding environments. Such scenarios result in spatio-temporal data structures, where suitable representation and understanding of such data structures allows us to incorporate domain specific knowledge in the learning frameworks that aim to recognize the human activities. This form of learning is especially important in computer vision applications such as human-robot collaboration and industrial scene inspections.

Indeed, the interactions between humans and their environments are inherently spatio-temporal

in the real world. For instance, during daily tasks such as house cleaning or food preparation, humans interact with several objects at different locations and different points of time, indicating dependencies over both space and time. Similar interdependence holds for the human body parts, where the individual parts (shoulders, arms, etc.) coordinate their motions through the joints to generate physically feasible postures. Such spatio-temporal interdependencies and coordinations have formed the basis for many graph-based algorithms in order to solve HOI and Human Action Recognition (HAR) problems [99, 112, 147, 167]. Scene-based features—which do not explicitly model the spatio-temporal relations—have also been widely used to learn graphical representations for both HOI and HAR. These features typically capture the scene-level appearance and contextual information (background, color, boundary, configuration, etc.) of both humans and the objects [4, 51, 119] and, sometimes, include the object affordances to learn how humans interact with them in a (physically) meaningful manner [50]. However, these methods have been mostly successful for offline semantic segmentation of human activities and online recognition of HOI from images and video clips.

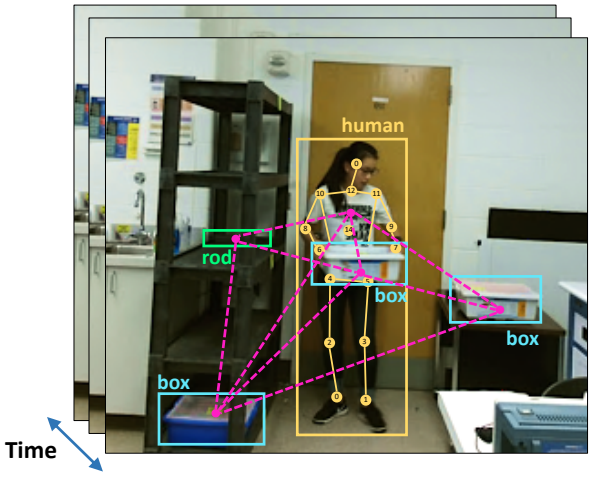


Figure 6.1: Demonstration of the hierarchical graph structure on a sample frame. The pink dashed-lines represent the human-object interaction graph and the yellow skeleton denotes the human body structure graph.

Our work, on the other hand, is inspired by the applications of HAR in industrial settings, where real-time interactions with objects play an important role. In particular, we consider the challenge of online activity recognition in warehouses and fulfillment centers, where the number of objects vary and the HOI graphs, often represented using adjacency matrices, are not known in advance. Our goal, therefore, is to create a flexible and fast converging algorithm that is able to identify the activities and determine the key objects during HOI from long videos. Such an algorithm opens up the possibility of performing key measurements such as ergonomics risk monitoring and process fidelity assessment in industrial settings. In both of these measures, the requirement includes identifying the correlations between the human poses and objects. Figure 6.1 demonstrates how we can leverage the graph representation of human pose along with a graph representation of the HOI in order to achieve this objective.

In this chapter, we propose a novel real-time Spatio-Temporal Hierarchical Graph Convolutional Network (ST-HGCN). ST-HGCN is designed with a multi-layered GCN that models HOI. The human node itself has a feature pyramid architecture that allows the model to learn the correlations between human body parts. The contributions of our work are threefold: (1) We introduce a novel hierarchical graph structure to handle temporal non-Euclidean datasets, in which the nodes are non-Euclidean themselves. (2) Inspired by RBF, we define an adjacency function in real-time for the Human Object Interaction Graph Convolutional Network (HOI-GCN). (3) We leverage our ST-HGCN architecture to solve an early activity recognition problem that, to the best of our knowledge, has not been explored in the context of HOI. We evaluate our algorithm on the UW-IOM dataset, and show that it not only significantly outperforms the state-of-the-art, but also converges faster than the already existing methods. In addition, we demonstrate how our proposed method can be applied to any custom dataset—possibly created for real industrial applications. All the steps and the source code will be available on the project's GitHub page.

6.2 Proposed Hierarchical Framework

In this work, we introduce Spatio-Temporal Hierarchical Graph Convolutional Network (ST-HGCN) for solving early action recognition problems. ST-HGCN models both the spatiotemporal features

of the skeletal structure and the HOI graph. To represent the skeletal structure of the human body, we use a Pyramidal Graph Convolutional Network (P-GCN) inspired by [99]. To capture the human object interaction within a scene we design a Human Object Interaction Graph Convolutional Network (HOI-GCN) with an RBF-based adjacency matrix. Finally, an LSTM unit is used to learn the temporal relations throughout the videos (shown in Figure 6.2 and described in Section 6.2.5). Before we proceed to the details of the model, it is essential to describe the input structure.

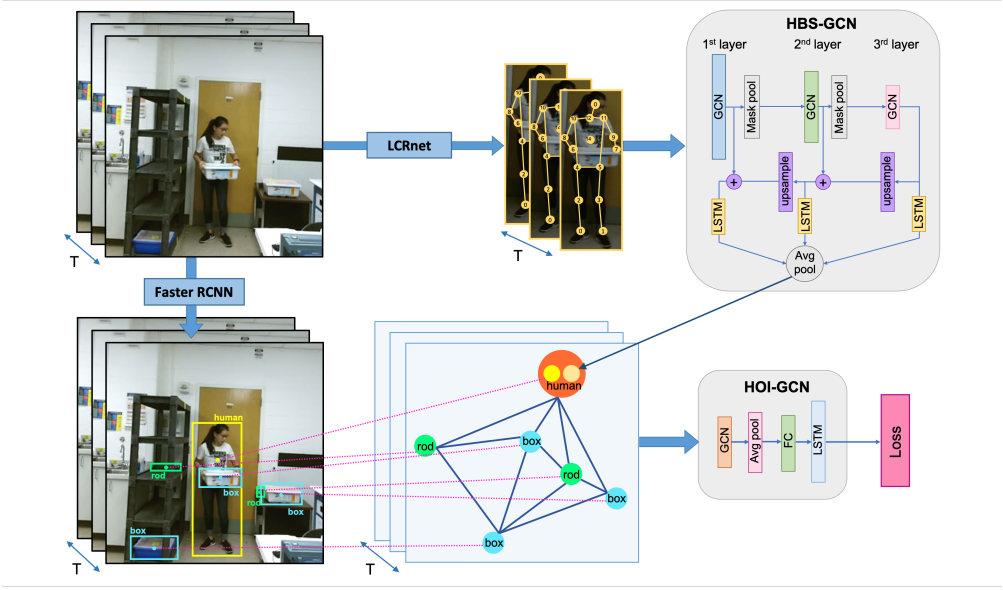


Figure 6.2: Demonstration of our Spatio-Temporal Hierarchical Graph Convolutional Network (ST-HGCN) scheme. The human 3D pose is detected using the LCRnet [117], and passes through HBS-GCN—a pyramid architecture inspired by [99]. In parallel, a Faster R-CNN [114] backbone detects the objects and the human and returns the bounding boxes and a feature vector representing the Region of Interest (ROI). In our *ST-HGCN-earlyFusion+LSTM* model, the human features from HBS-GCN are fused with the Faster R-CNN features and in the late fusion model (*ST-HGCN+LSTM*), they are fused after the `Avg pool` layer. *ST-HGCN+LSTM* is our best performing model.

6.2.1 *Model Input Structure*

The input to the model comprises two main parts. First, the skeleton structure of the human body, which consists of a sequence of the xyz location of $N = 15$ joints ($\mathbf{X} \in \mathbf{R}^{F \times N \times T}$, where $F = 3$ and T is the number of frames). Second, the objects bounding box coordinates along with the Faster R-CNN [114] features ($f_o \in \mathbf{R}^{1024}$) that represent each object (described in Section 6.3.3). A bounding box is defined by a tuple containing (x_0, y_0) and (x_1, y_1) coordinates, where x_0, y_0 are the coordinates of the image's top left corner and x_1, y_1 denote the bottom right. The size of the second set of features is $1024 + 4 = 1028$. Note that this feature is also computed for the detected human.

6.2.2 *Graph Convolutional Network*

GCN is proved to be an efficient tool for representing non-Euclidean data structures [166]. In this work, we are using an undirected graph to represent the human skeleton structure as well as the HOI graph. To briefly describe the machinery of GCNs, consider a graph with N nodes, a set of edges between nodes encoded by an adjacency matrix $\mathbf{A} \in \mathbf{R}^{N \times N}$. The degree matrix is then denoted by $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. Let $\mathbf{x}_i \in \mathbf{R}^F$ represent the feature of each node, then $\mathbf{x} \in \mathbf{R}^{F \times N}$ represents the feature matrix of the graph and a linear formulation of graph convolution is,

$$\mathbf{f} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{x}^\top \mathbf{W}, \tag{6.1}$$

where the output feature $\mathbf{f}$ is $N \times C$ with C as the output feature size. We define $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ where $\mathbf{I}$ is the identity matrix and $\mathbf{W} \in \mathbf{R}^{F \times C}$ denotes the weight matrix. Similar to the design of 2D convolutional layers, we can have a stack of GCNs each followed by a nonlinear function—say, `ReLU`—to capture more detailed representations [57].

For the human skeleton structure, we follow the spatial configuration partitioning introduced in ST-GCN [156]. Therein, the adjacency matrix is comprised of three matrices, hence $\hat{\mathbf{A}} = \sum_a \mathbf{A}_a$ and Equation 6.1 is cast as a summation,

$$\mathbf{f} = \sum_a \hat{\mathbf{D}}_a^{-\frac{1}{2}} \mathbf{A}_a \hat{\mathbf{D}}_a^{-\frac{1}{2}} \mathbf{x}^\top \mathbf{W}_a, \tag{6.2}$$

6.2.3 Human Body Structure Graph

Feature pyramids have been leveraged mainly in object recognition algorithms [41, 75, 127] to produce a multi-scale feature representation with semantically strong meaning. Recently, Parsa et al. [99] brought the feature pyramid architecture in skeleton-based action recognition and showed that this design can more effectively capture the correlation of body parts. We use a similar Pyramidal Graph Convolutional Network (P-GCN) as [99] to represent the skeletal structure of the human body. P-GCN consist of two pathways: bottom-up and top-down. The **bottom-up** pathway is a feed-forward process to compute a feature hierarchy consisting of feature maps at different scales using a series of GCNs. The **top-down** pathway is the process of up-sampling spatially larger—but semantically stronger—feature maps from higher pyramid levels resulting in higher resolution features.

The input skeleton representation to the bottom-up pathway, $\mathbf{X} \in \mathbf{R}^{F \times N \times T}$, consists of the xyz location of $N = 15$ human body joints with $F = 3$ and T indicates time. Each level of P-GCN has a specific adjacency matrix ($\hat{\mathbf{A}}_k$) that is defining the connection between the nodes at that level and each GCN aggregates features using Equation 6.2. The initial GCN works on the skeleton with $\hat{\mathbf{A}}_1$, constructed based on the skeleton connections. The second graph represents the body parts and the third one the global structure. Each adjacency matrix is accompanied with an edge-importance matrix[1]. A Group Average Pool (GAP) layer average-pools the features in a selected group of nodes using a specific kernel so that the corresponding graph has semantically meaningful nodes. This successive design of GCN-GAP combinations models the entire local and global motions jointly.

To enhance the features in the top-down pathway, the output features of the bottom-up pathway (3 layers) undergo a 1×1 conv layer to reduce channel dimensions and then they are merged into the top-down pathway features by element-wise addition (as described in [99]). In the rest of the paper, we refer to Human Body Structure Graph Convolutional Network as HBS-GCN.

[1]Edge-importance is a learnable mask with the same dimension as the adjacency matrix and learns to scale the contribution of a node's feature to its neighboring nodes based on the learned importance weight of each spatial graph edge.

6.2.4 Human Object Interaction Graph

Performing an activity, humans often interact with multiple objects within a scene where the interactions are under the influence of the relative position of objects with respect to each other and the human. Therefore, local interactions are as important as the global ones. This, hence, motivates the design of a GCN to capture the local and global representations in a scene.

We propose a HOI-GCN with N_{hoi} nodes consisting of a human and the objects detected in the scene. To encourage local attention in our network, we construct an adjacency matrix with elements representing the relative distance of nodes. Then this adjacency matrix is normalized to the interval of $[0, 1]$ and undergoes an RBF function that prioritizes closer objects by creating values closer to one as in,

$$a_{ij}^{hoi} = \exp\left(-c{d_{ij}}^2\right), \tag{6.3}$$

where a_{ij}^{hoi} is the ij^{th} element of the adjacency matrix A^{hoi}, d_{ij} is the normalized distance between the nodes i and j, c is a constant that is dependent on the scene setup ($c = 5$ for UW-IOM dataset), and exp is the exponential function. The adjacency matrix was accompanied by an edge importance matrix to allow the algorithm learn the more sophisticated nodal relations from the data.

6.2.5 Spatio-Temporal Modeling

In this section, we describe in details how the HBS-GCN and HOI-GCN are combined and used in the temporal modeling of the video sequences. The output features of the top-down pathway ($\mathbf{f}_k$) are passed through an LSTM and result in a feature ($\mathbf{z}_k$) that captures the temporal aspect of the human body movement. We use an `adaptive average-pool` mechanism to convert the three features to one feature vector of size 1028. Then this feature is fused to the Faster R-CNN feature corresponding to the human and undergoes a linear transformation followed by an activation unit (`ReLU`). The resultant feature for the human node along with the features representing other objects is used as the input to the HOI-GCN in order to represent the scene spatial features. The output of the HOI-GCN undergoes a linear transformation followed by a `ReLU` activation unit and is sent to an LSTM unit to create frame-wise activity predictions.

6.3 Experiments

6.3.1 UW-IOM Dataset

Our proposed method is designed for online-HOI on long custom videos. The most relevant dataset that satisfied these conditions is the newly released UW-IOM [100]. This dataset consists of twenty videos capturing activities that are common in warehouses, and each video is approximately three minutes. There are seventeen action classes and the labels consist of the object (box/rod), human motion (walk, stand, and bend), type of object manipulation if applicable (reach, pick-up, place, and hold), and the relative height of the surface where manipulation is taking place (low, medium, and high).

To extract the human 3D pose, we use LCR-Net [117]. Moreover, we use Faster R-CNN [114] to detect the objects, as UW-IOM does not provide the bounding boxes and features representing the objects.

6.3.2 HOI-GCN Node Representations

We fine-tune the Faster R-CNN [114] with Detectron2 [150] backbone pre-trained on ImageNet to detect the human and objects in the UW-IOM dataset. The result of the detection algorithm is the 2D coordinate of the objects. However, we also require the underlying features that is used by the network during the detection process, while in the current implementation of Faster R-CNN, those features are inaccessible. Hence, we implement a function to extract the output of the fully connected layer after the `ROI Pooler` layer and before the `Softmax`, and use that as the object representations.[2]

6.3.3 Implementation Details

In our experiments, we sample a fixed length T=80 frames from each video sequence, and use a batch size of 128. Inspired by [156], the adjacency matrix applied on the skeletal data at the first layer of the P-GCN ($\hat{\mathbf{A}}_1$) is partitioned into 3 subsets: the root node itself, centripetal group, and

[2]Full detail of the implementation with the code will be available on the project GitHub page.

centrifugal group. For the two other layers of the pyramid network similar to [99], we initialize $\hat{\mathbf{A}}_2$ and $\hat{\mathbf{A}}_3$ as if we have a fully connected graph and allow the edge importance learn the weighting of the nodes. The feature pyramid architecture in HBS-GCN has 64 channels in the first GCN layer, 128 in the second GCN, and 256 channels in the third one. The GCN in the HOI-GCN has 1024 channels.

We design two different architectures to find the best strategy for fusing the feature coming from HBS-GCN and Faster R-CNN for the human-node shown in Figure 6.2. In the first design we fuse the features after the `Avg pool` unit in HOI-GCN. In the second design, we concatenate the two features and pass them through a linear layer to make the size consistent with the object-nodes and then send the HOI graph to the HOI-GCN. We call this design *ST-HGCN-earlyFusion*.

We used the same training and validation videos used in [99] to be able to compare our results, however, since we are interested to determine the generalizability of our model, we keep out video number 12 for testing. Thus we have 15 training and 4 validation and 1 test videos. We report the mean and variance of all the validation splits in the results in Table 6.1. We use the Adam optimizer [56] to optimize the network and perform a grid search for learning rate (lr) from $1e-2$ to $5e-5$. On an average, an lr of $5e-5$ performs best on all the splits. We use early stopping with 10-step patience based on the F1-overlap and Edit score.

6.4 Results and Discussion

To validate the effectiveness of our ST-HGCN model, we performed two experiments. The first one is without the human skeleton information from the HBS-GCN, which we refer to as *HOI-GCN+LSTM* in Tables 6.1 and 6.2. The second experiment is our hierarchical graph convolution design that fuses the features from HBS-GCN and Faster R-CNN for the human and runs a HOI-GCN on the scene. We call this experiment *ST-HGCN+LSTM*. We also consider two different architectures for fusing the human features. One choice is to fuse after HOI-GCN has aggregated the Faster R-CNN feature of the scene (*ST-HGCN+LSTM*), and the other is to fuse before the aggregation happens (*ST-HGCN-earlyFusion+LSTM*). In the following sections, we discuss the performance of these methods in more details.

Backbones	UW-IOM		
	mAP (%)	Edit (%)	F1-overlap (%)
ST-GCN [156]	66.94 ± 3.49	61.89 ± 3.56	71.08 ± 2.83
ST-GCN+IMP [156]	73.28 ± 4.30	67.21 ± 6.05	76.58 ± 4.95
ST-PGN+LSTM+IMP+ML [99]	87.03 ± 2.85	79.86 ± 2.15	87.95 ± 1.54
ST-PGN+LSTM+IMP+ML+GRU-Fusion [99]	87.05 ± 3.47	80.90 ± 2.06	88.08 ± 1.89
HOI-GCN+LSTM (ours)	80.08 ± 11.82	87.35 ± 3.11	89.18 ± 4.02
ST-HGCN-earlyFusion+LSTM (ours)	83.50 ± 9.96	87.81 ± 3.23	89.09 ± 3.08
ST-HGCN+LSTM (ours)	**88.40 6.32**	**88.84 ± 3.23**	**90.29 ± 1.64**

Table 6.1: mAP across classes, edit, and F1-overlap score represented using mean and standard deviation over the validation set in the UW-IOM dataset for different methods and modalities. The best results are shown in bold. The results for our best model was achieved after 12 epoch with the learning rate of $5e-5$. It should be mentioned that we consider four validation videos and kept one video for testing, and the results on the test video are shown in Table 6.2.

Backbones	UW-IOM		
	mAP (%)	Edit (%)	F1-overlap (%)
HOI-GCN+LSTM (ours)	78.37 ± 17.95	72.22	75.60
ST-HGCN-earlyFusion+LSTM (ours)	83.83 ± 16.84	82.95	80.33
ST-HGCN+LSTM (ours)	**85.88 ± 12.52**	**82.67**	**82.70**

Table 6.2: mAP across classes, edit, and F1-overlap score represented using mean and standard deviation over the test set (video number 12) in the UW-IOM dataset for different methods and modalities. The best results are shown in bold.

6.4.1 Performance Analysis

HBS-GCN vs. HOI-GCN vs. ST-HGCN

In Table 6.1, we compare the result of our ST-HGCN model on UW-IOM dataset with the other reported results on this dataset. Parsa et al. examined the idea of fusing scene information driven from a VGG16 network to the skeleton-based features in [99]. Although our model only has partial information from the scene, it outperforms the *ST-PGN+LSTM+IMP+ML+GRU-Fusion* results. This is a key observation emphasizing the importance of modeling the human object relation in online-HAR problems. Our HBS-GCN has a similar architecture to the ST-PGN network presented in [99]. Comparing our best model results with *ST-PGN+LSTM+IMP+ML* in Table 6.1, we see that incorporating the HOI graph enhances the skeleton-only results. Our early- and late-fusion models perform similarly. However, the late fusion of HBS-GCN features results in the best performance (*ST-HGCN+LSTM*). In Table 6.2, we show the performance of our models on the test set. It is insightful to see even with close performance of HOI-GCN with the other models with skeleton information, this model is significantly less generalizable on an unseen video.

In Figure 6.5, we demonstrate the average edge importance matrices across three matrix components of the adjacency matrix corresponding to each level of the pyramid network. Similarly, we showcase the edge importance for the HOI graph in Figure 6.3. The UW-IOM dataset includes object manipulation tasks that involve picking-up and placing objects from a shelf at different heights. Therefore, in Figure 6.5, we observe that the correlations between the legs and arms are more important. The majority of participants in this dataset are right-hand dominant. This is interestingly reflected in the resulting edge importance matrices as *R-leg* and *R-arm* are found to be more important.

Looking at Figure 6.3 that shows the objects edge importance weights for *ST-HGCN+LSTM*, an interesting observation is that the correlation of the objects matter more as compared to the objects features themselves (brighter off-diagonal entries). There is an exception for *obj 7*, and we think this is because that object is mainly absent and network needs to pay close attention to when it is detected meaning the feature is non-zero. *obj 1* to *obj 3* are mainly the boxes in the scene

and they are almost always present. Therefore, the corresponding off-diagonal entities are brighter (more important).

A similar plot is drawn for the *HOI-GCN+LSTM* in Figure 6.4. We see the results are reversed as compared to *ST-HGCN+LSTM* (Figure 6.3). We believe that this reversal is because the human feature has no advantage over the other objects, and they are all driven from the Faster R-CNN backbone. Therefore, any object can be selected as the most important node. Another interesting observation is that the diagonal elements are more important than the off-diagonal elements, implying that the objects themselves play a more important role in solving the recognition task rather than the correlations among the objects.

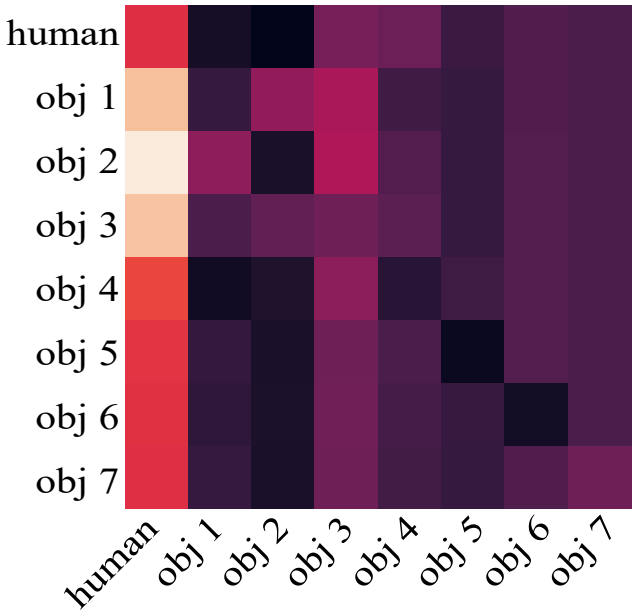


Figure 6.3: Learned edge importance heat-map for the ST-HGCN model. Each row shows the edge importance corresponding to each node of the HOI graph. The HOI-GCN part of the network uses the edge importance multiplied by the adjacency matrix to aggregate node features. The level of brightness shows higher values and is an indication of the importance.

Late Fusion vs. Early Fusion

The idea behind designing this experiment is to determine the most effective place in the network to include the HBS-GCN human feature. In the early fusion model (*ST-HGCN-earlyFusion+LSTM*), we fuse the two features for the human node before the HOI-GCN aggregates node features, and in the late fusion model (*ST-HGCN+LSTM*) the HBS-GCN feature later fused to the scene feature coming from the HOI-GCN. Although the results in both Table 6.1 and 6.2 are close, it is slightly in favor of *ST-HGCN+LSTM*, and we believe this is due to the higher pose information preserved in the feature vector that goes to the classifier.

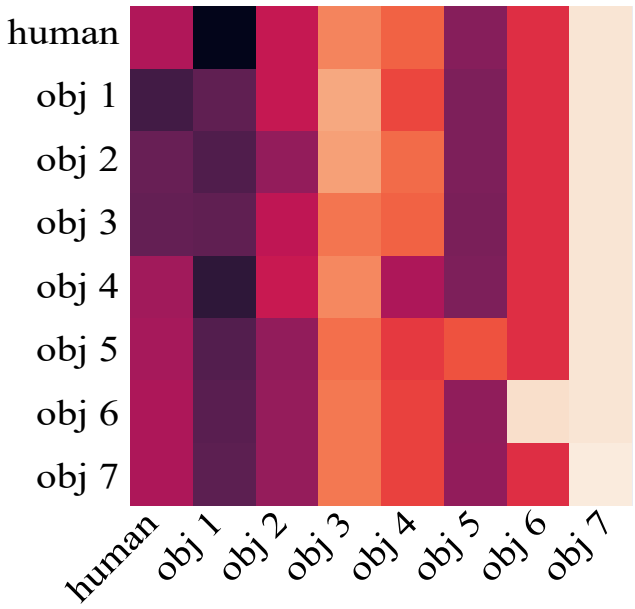


Figure 6.4: Learned edge importance heat-map for the HOI-GCN model. Each row shows the edge importance corresponding to each node of the HOI graph. HOI-GCN uses the edge importance multiplied by the adjacency matrix to aggregate node features. The level of brightness shows higher values and is an indication of the importance.

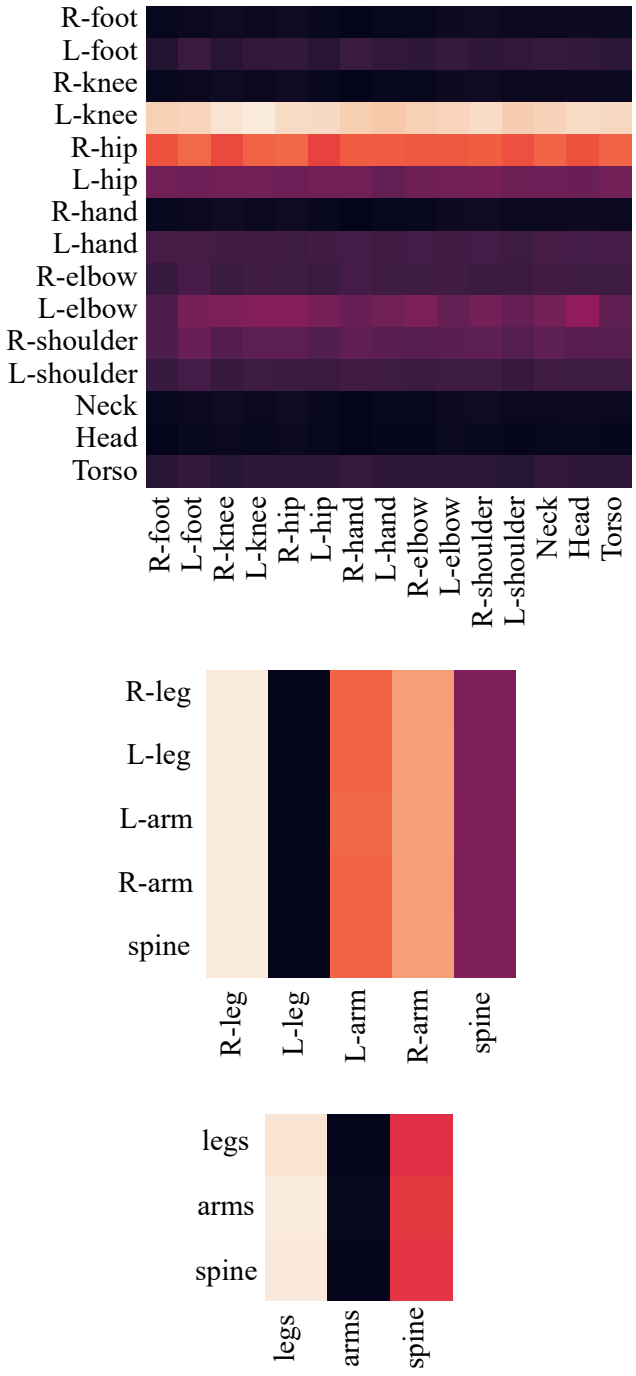


Figure 6.5: Three level learned edge importance heat-map in UW-IOM shown as a heat-map. Each row shows the edge importance of each level of graph pyramid as described in [99]. Every level of the pyramid consists of the sum of three edge importance multiplied by the adjacency matrix and node features. The level of brightness shows higher values and is an indication of the importance.

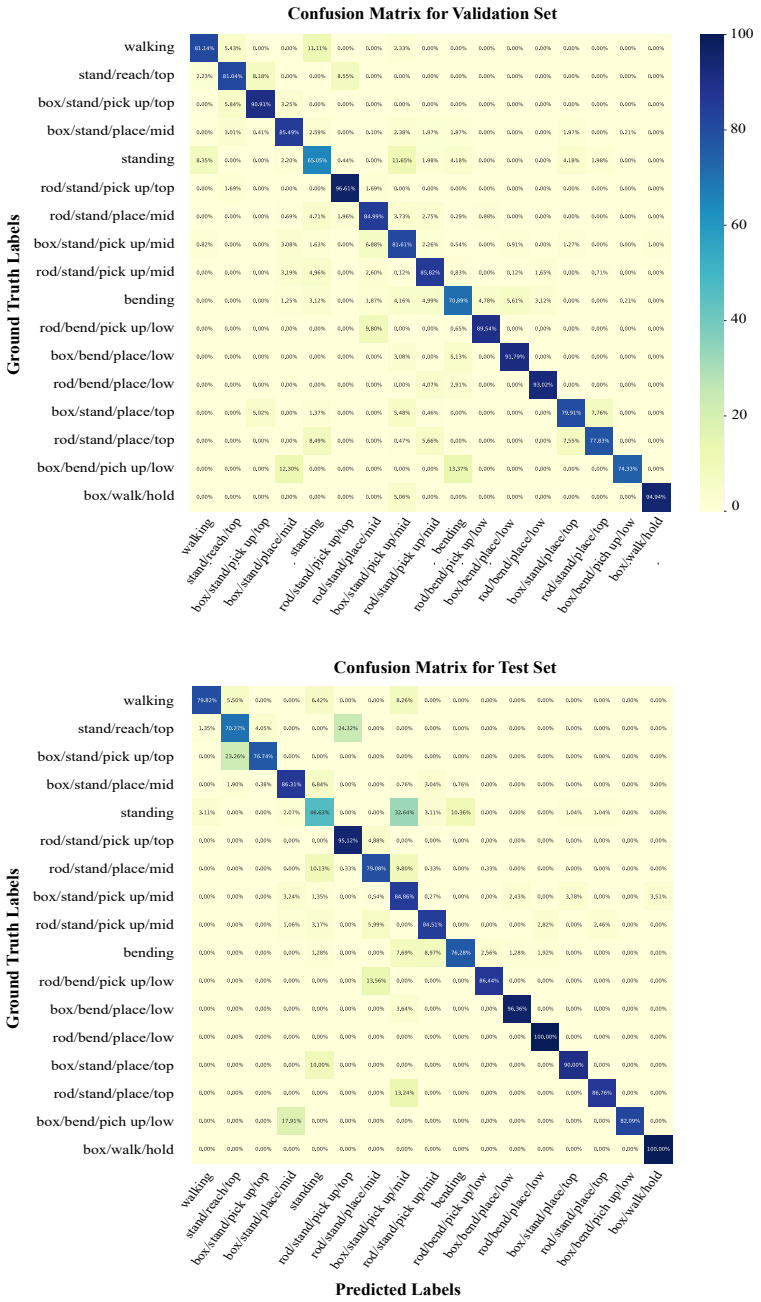


Figure 6.6: Confusion matrix for *ST-HGCN+LSTM* model. The top figure is for the validation set and the bottom figure for the test set.

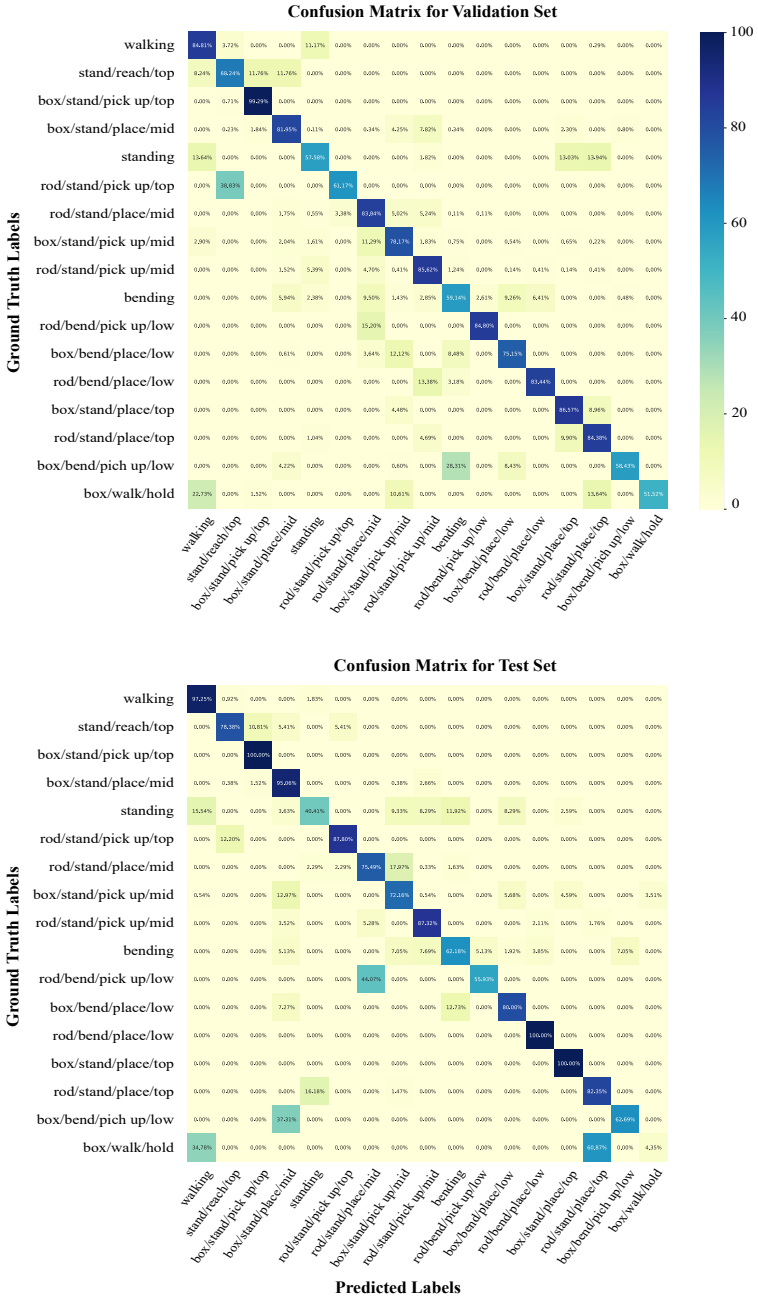


Figure 6.7: Confusion matrix for *HOI-GCN+LSTM* model. The top figure is for the validation set and the bottom figure for the test set.

6.4.2 Failure Cases

We show the confusion matrices for our *ST-HGCN+LSTM* model performance on both the validation and test set in Figure 6.6. The bottom confusion matrix demonstrates the significance of the performance. In addition, as observed from the results in Table 6.2, the HOI-GCN generalizes poorly to the test set.

Our models can successfully classify box-handling and rod-handling actions. However, *Standing* and *walking* are more confusing, especially at the transitions between two classes. This is even more evident when the hands of the subjects are hidden from the camera due to self-occlusion. Similarly, *bending* is confused with *bending/place* and *bending/pick up* since the transitions between these actions are not necessarily annotated consistently in this dataset. In the *HOI-GCN+LSTM* model (Figure 6.7), more confusion is observed for actions in which the pose carries a lot of information, such as *rod/pick-up* and *box-pickup*, where the hand position is the main differentiator.

6.5 Summary

In this chapter, we propose using a Spatio-Temporal Hierarchical Graph Convolutional Network (ST-HGCN) to leverage the human object interaction information in a skeleton-based online action recognition setup. Our algorithm can potentially be employed for any custom dataset and with all the components released on the project website. We compare the results of our method to two other designs as well as the state-of-the-art results on UW-IOM dataset to show how human object interaction information can enhance the results of an online-HAR model.

Chapter 7

CONCLUSIONS AND FUTURE WORK

In this thesis we studied Human Action Recognition (HAR) in video data. Part I is about Human Activity Segmentation (HAS) in which the goal is to semantically segment long videos into activity classes and identify the beginning and end of each segment. Part II is dedicated to early-/online-HAR problems in which inference is made to identify human actions in the latest observed T frames. Usually, T is a small number in the range of 50 to 100 frames, which makes it a challenging problem. All presented studies are done having the industry application in mind. Specifically, they are dedicated to facilitate automation in ergonomics risk assessment. When real world application is a key drive of projects then the generalizability of the developed algorithms become even more essential. In this regard, this chapter focuses on discussing the key learning points from the presented studies.

7.1 Discussion

In Chapter 3, we present an end-to-end deep learning system to accurately segment human actions and predict the corresponding ergonomic risks during indoor object manipulation using camera videos. This system comprises effective spatial features extraction and sequential feeding of the extracted features to temporal neural networks for real-time segmentation into meaningful actions. We believe that good overall performance is the cumulative effect of both the steps as is observed from our results for the different segmentation methods on the more challenging UW IOM dataset. This reinforces the intuition that both spatial and temporal characteristics are important in analyzing long-duration human action videos. The segmentation methods work well with just standard (RGB) camera videos, irrespective of how the spatial features are extracted, provided depth cameras are used to generate reliable ergonomic risk scores for all the possible actions correspond-

ing to a known object manipulation environment. Consequently, it makes our system useful for widespread deployment in factories and warehouses without requiring 3D cameras, body markers, and body-mounted sensors.

Despite the accurate performance of the proposed model, this method can be improved in multiple aspects. First, is to use less context-dependent features (VGG16 driven) to enhance the generalizability of the model. In Chapter 4, a multi-task learning framework is proposed that uses skeleton-based features for solving similar problem. Another opportunity for improvement is to move toward online-HAR (or early-HAR) and develop a learning method that would be capable of risk prediction on a frame-by-frame basis. This aspect is addressed in Chapter 5.

In Chapter 4, we introduce a graph-based multi-task learning approach for Human Postural Assessment and show that it outperforms the equivalent Single-Task Learning due to the importance of the activity type in the risk associated with a posture. Human Postural Assessment tasks, specifically Ergonomics Risk Assessment, are more challenging than regular Human Activity Evaluation problems since the assessment has to happen in a frame-wise manner and is highly dependent on joint kinematics. Despite the challenge of tracking the intricacies of our risk assessment (REBA) profile, the proposed method shows competence in predicting the risk scores. More importantly, our work demonstrates the effectiveness of the GCN model as a spatial feature extraction backbone, compared to context-based features that have been traditionally used with ED-TCN for Human Activity Segmentation tasks. Although the focus of this work is on Ergonomics Risk Assessment, we believe that our Multi-Task Learning approach can be applied to many other action and skill assessment problems. The mapping of skeletal representation to the activity score using GCN is a new approach for solving ERA, which can initiate a new direction by exploiting the natural connection between posture and activity risk/quality.

In Chapter 5, We proposed a novel Spatio-Temporal Pyramid Graph Convolutional Network (ST-PGN) for online action recognition. The method integrates the following: a) basic prior knowledge about the skeletal structure, b) hierarchical joint relationships and c) data-driven learning framework for online action based ergonomics risk assessment. The hierarchical graph embeddings preserve the semantic meaning of skeletal keypoint descriptors. The learned skeletal features

are fused with image features using a fusion mechanism that can model motion and context jointly. The proposed approach addresses the simultaneous association of time-varying pose with action and objects interaction to enable downstream applications that involve computational modeling and prediction of various human performance metrics for ergonomics risk assessment.

Some open issues remain. First, generalization concerning other skeletal joint representations (Lie [143], Quaternion [105]) and camera viewpoint changes have not been addressed. Furthermore, while we outperform state of the art on online action recognition we are only comparable to the state of the art in offline action datasets (NTU-RGBD and Skeleton-Kinetics). The effect of the action label distribution on our models need to be studied further. In future work, we hope to address these issues with improved context fusion, long-term temporal modeling, and biomechanically consistent human pose representations [168]. In Chapter 6, we propose using a Spatio-Temporal Hierarchical Graph Convolutional Network (ST-HGCN) to leverage the human object interaction information in a skeleton-based online action recognition setup. Our algorithm can potentially be employed for any custom dataset and with all the components released on the project website. We compare the results of our method to two other designs as well as the state-of-the-art results on UW-IOM dataset to show how human object interaction information can enhance the results of an online-HAR model.

We believe that our ST-HGCN model introduces a new perspective in solving both HAR and HOI problems. As discussed earlier, the HOI graph has not been considered in solving online-HAR problems, and the human skeleton information has not been leveraged in HOI problems. In addition, HOI methods have not been implemented for handling long videos. Hence, we anticipate our method to inspire more efforts in this line of research.

7.2 Towards Generalization

A generalizable neural network shows reliable performance on unseen data in addition to the training set. The question of what metric should be used for measuring generalizability of both Machine Learning and Deep Learning models have been a research topic itself [37, 52, 158]. In the case of video analysis, generalizability has two aspects to it; one is the capability to perform on sequences

with different temporal properties that share similar features as the training data. Second, is the capability to capture spatial representation in unseen videos. Prior knowledge about data helps with designing richer spatial and temporal representation and achieve better performance on unseen or adversarial data. In the following sections we discuss our observations from the research presented in previous chapters and the possible directions for key component of HAR problem, such as features and training algorithm.

7.2.1 Spatial and Temporal Features

Good feature representations result in a better differentiated search space and make it easier for the learning algorithm to distinguish different classes of activities. Context rich features such as pixel-based features often show the best performance on the training set, but depending on how the feature extractor is designed their performance on test set suffers. In Chapter 3, we use a VGG16 backbone to extract the spatial features in each frame and we observed that despite the good performance of models with pixel-based features, they do not necessarily generalize well. For HAR, specifically, these features do not represent human body dynamics. Optical Flow which is the pattern of apparent motion of objects, humans, and surfaces in a visual scene caused by the relative motion between an observer and a scene is often used to capture human movement in a scene. Optical follow is a measure for quantifying the difference between two frame of a video from pixel values. Although using optical flow is effective in many applications, it has a few limitations when it comes to HAR. Optical flow is most useful when the camera and the background are stationary. Like any other 2D pixel based feature extractor it is sensitive to view point and how the human body is oriented with respect to the camera.

In Chapter 4, 5, and 6, we demonstrated the skeleton-based features, and how they can enhance the generalizability of models. Skeleton-based features provide the opportunity to incorporate a more realistic model of human movement in to a deep learning model and since to posture of performing an a activity is similar across individuals it creates a more generalizable model. 3D skeleton information can be represented with Lie Algebra and make the model learn the human joints trajectories [10]. To further enhance the quality of Lie algebra-based feature representation,

we can design convolutional networks inspired from Rotation Matrices used in Lie algebra mappings [72]. The downside of using skeleton-based feature is that we lose information about the scene, and in many cases key elements in classifying human activities are objects or scene information. We explored incorporation of objects with skeleton information in Chapter 6. To design well representative features one should balance the use of context-independent and context heavy features. For modeling the temporal representation, when it comes to HAS, ED-TCNs variations a very powerful to capture the temporal transitions between activities, and we observed that for early-HAR, LSTM-based designs perform better.

7.2.2 Learning Algorithms

In this thesis we demonstrate the power of Graph Convolutional Networks (GCNs) in capturing complex non-Euclidean data structures with interactions. In Chapter 4-6, we used variations of GCNs to represent both the human skeleton and the scene graph. GCNs learn the dynamic of the interaction between input elements and at the time of inference in addition to spatial positioning of elements they look for the interactions.

If the goal is to move towards generalizable HAR algorithms for real-life scenarios, we need to leverage prior knowledge about the task as much as possible. In addition, if all the activities are not predefined for the application in hand, we need to transition from Static Learning Algorithms (SLAs) to Dynamic Learning Algorithms (DLAs) to be able to handle unseen activities. SLAs are algorithms trained offline and used in online inference, whereas DLAs are algorithms that get updated as new observations are made. The challenge with DLAs is that we would need a labeling mechanism to label new activities. One possible getaway with the issue of lacking labels is to use few-shot learning and/or to bring in other modalities of training such as Natural Language Processing and use knowledge graphs to create labels for unseen activities with a better accuracy.

BIBLIOGRAPHY

[1] CS University of Toronto CSC321 Winter 2014 - lecture six slides.

[2] Ahmed Abobakr, Darius Nahavandi, Julie Iskander, Mohammed Hossny, Saeid Nahavandi, and Marty Smets. A kinect-based workplace postural analysis system using deep residual networks. In *IEEE Int. Syst. Eng. Symp.*, pages 1–6, 2017.

[3] Naum I Achieser. *Theory of approximation*. Courier Corporation, 2013.

[4] Aniket Agarwal, Ayush Mangal, et al. Visual relationship detection using scene graphs: A survey. *arXiv preprint arXiv:2005.08045*, 2020.

[5] Svitlana Antoshchuk, Mykyta Kovalenko, and Jürgen Sieck. Gesture recognition-based human–computer interaction interface for multimedia applications. In *Digitisation of Culture: Namibian and International Perspectives*, pages 269–286. Springer, 2018.

[6] 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.

[7] Jan Bandouch and Michael Beetz. Tracking humans interacting with the environment using efficient hierarchical sampling and layered observation models. In *IEEE Int. Comput. Vis. Workshops*, pages 2040–2047, 2009.

[8] Renato Baptista, Michel Goncalves Almeida Antunes, Djamila Aouada, and Björn Ottersten. Video-based feedback for assisting physical activity. In *12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2017.

[9] Djamila Romaissa Beddiar, Brahim Nini, Mohammad Sabokrou, and Abdenour Hadid. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, 79(41):30509–30555, 2020.

[10] Malek Boujebli, Hassen Drira, Makram Mestiri, and Imed Riadh Farah. Rate-invariant modeling in lie algebra for activity recognition. *Electronics*, 9(11):1888, 2020.

[11] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Eur. Conf. Comput. Vis.*, pages 25–36, 2004.

[12] Birgitta Burger and Petri Toiviainen. MoCap toolbox - A Matlab toolbox for computational analysis of movement data. In *Sound Music Comput. Conf.*, pages 172–178, 2013.

[13] Pau Panareda Busto, Ahsan Iqbal, and Juergen Gall. Open set domain adaptation for image and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):413–429, 2018.

[14] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:1812.08008*, 2018.

[15] 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 (CVPR)*, pages 6299–6308, 2017.

[16] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[17] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 381–389. IEEE, 2018.

[18] Benjamin Chasnov, Momona Yamagami, Behnoosh Parsa, Lillian J Ratliff, and Samuel A Burden. Experiments with sensorimotor games in dynamic human/machine interaction. In *Micro-and Nanotechnology Sensors, Systems, and Applications XI*, volume 10982, page 109822A. International Society for Optics and Photonics, 2019.

[19] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Mach. Vis. Conf.*, 2014.

[20] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-CNN: Pose-based CNN features for action recognition. In *IEEE Int. Conf. Comput. Vis.*, pages 3218–3226, 2015.

[21] Teunis Cloete and Cornie Scheffer. Benchmarking of a full-body inertial motion capture system for clinical gait analysis. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pages 4579–4582, 2008.

[22] Ana Colim, Paula Carneiro, Nélson Costa, Pedro M Arezes, and Nuno Sousa. Ergonomic assessment and workstation design in a furniture manufacturing industry—a case study. In *Occupational and Environmental Safety and Health*, pages 409–417. 2019.

[23] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3992–4000, 2015.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 248–255, 2009.

[25] Rahul Dey and Fathi M Salemt. Gate-variants of gated recurrent unit (GRU) neural networks. In *Proceedings of the IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1597–1600, 2017.

[26] Lieyun Ding, Weili Fang, Hanbin Luo, Peter ED Love, Botao Zhong, and Xi Ouyang. A deep hybrid learning model to detect unsafe behavior: integrating convolution neural networks and long short-term memory. *Autom. Construction*, 86:118–124, Feb. 2018.

[27] Hazel Doughty, Dima Damen, and Walterio Mayol-Cuevas. Who's better? who's best? pairwise deep ranking for skill determination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6057–6066, 2018.

[28] 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 (CVPR)*, pages 1110–1118, 2015.

[29] Weili Fang, Lieyun Ding, Hanbin Luo, and Peter ED Love. Falls from heights: A computer vision-based approach for safety harness detection. *Autom. Construction*, 91:53–61, Feb. 2018.

[30] Alireza Fathi and James M Rehg. Modeling actions through state changes. In *IEEE Conf. Comput. Vis. Pattern Recognit*, pages 2579–2586, 2013.

[31] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Evaluating surgical skills from kinematic data using convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 214–221. Springer, 2018.

[32] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Fine-grained action segmentation using the semi-supervised action gan. *Pattern Recognition*, 98:107039, 2020.

[33] Pallabi Ghosh, Yi Yao, Larry S Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. *arXiv preprint arXiv:1811.10575*, 2018.

[34] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 759–768, 2015.

[35] Alireza Golabchi, SangHyeok Han, JoonOh Seo, SangUk Han, SangHyun Lee, and Mohamed Al-Hussein. An automated biomechanical simulation approach to ergonomic job analysis for workplace design. *J. Construction Eng. Manag.*, 141(8):04015020, Jan. 2015.

[36] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional LSTM networks for improved phoneme classification and recognition. In *Int. Conf. Artif. Neural Networks*, pages 799–804, 2005.

[37] Shuyue Guan and Murray Loew. Analysis of generalizability of deep neural networks based on the complexity of decision boundary. *arXiv preprint arXiv:2009.07974*, 2020.

[38] Xiaonan Guo, Jian Liu, and Yingying Chen. Fitcoach: Virtual fitness coach empowered by wearable mobile devices. In *Proceedings of the IEEE Conference on Computer Communications*, pages 1–9, 2017.

[39] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, 2015.

[40] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10905–10914, 2019.

[41] 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 (CVPR)*, pages 770–778, 2016.

[42] Martin Helander. *A guide to human factors and ergonomics*. CRC Press, 2005.

[43] Sue Hignett and Lynn McAtamney. Rapid entire body assessment (reba). *Applied Ergonomics*, 31(2):201–205, 2000.

[44] Sue Hignett and Lynn McAtamney. Rapid entire body assessment. In *Handbook of Human Factors and Ergonomics Methods*, pages 97–108. CRC Press, 2004.

[45] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[46] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *Eur. Conf. Comput. Vis.*, pages 137–153, 2016.

[47] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[48] Winfried Ilg, Johannes Mezger, and Martin Giese. Estimation of skill levels in sports based on hierarchical spatio-temporal correspondences. In *Joint Pattern Recognition Symposium*, pages 523–531. Springer, 2003.

[49] IntelRealSense. Intelrealsense/librealsense, Apr. 2019.

[50] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5308–5317, 2016.

[51] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10236–10247, 2020.

[52] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*, 2018.

[53] Keras-Team. keras-team/keras, Feb. 2019.

[54] Sunoh Kim, Kimin Yun, Jongyoul Park, and Jin Young Choi. Skeleton-based action recognition of people handling objects. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 61–70. IEEE, 2019.

[55] Wansoo Kim et al. Adaptable workstations for human-robot collaboration: A reconfigurable framework for improving worker ergonomics and productivity. *IEEE Robot. Autom. Mag.*, pages 1–1, 2019.

[56] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[58] Michael Kipp. ANVIL - a generic annotation tool for multimodal dialogue. In *Eur. Conf. Speech Commun. Tech.*, pages 1367–1370, 2001.

[59] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230*, 2018.

[60] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):14–29, 2015.

[61] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *IEEE Winter Conf. Appl. Comput. Vis.*, pages 1–8, 2016.

[62] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2556–2563, 2011.

[63] Tian Lan, Yang Wang, Weilong Yang, Stephen N Robinovitch, and Greg Mori. Discriminative latent models for recognizing contextual group activities. *IEEE transactions on pattern analysis and machine intelligence*, 34(8):1549–1562, 2011.

[64] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 156–165, 2017.

[65] Colin Lea, René Vidal, and Gregory D Hager. Learning convolutional action primitives for fine-grained action recognition. In *IEEE Int. Conf. Robot. Autom.*, pages 1642–1649, 2016.

[66] 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 (ICCV)*, pages 1012–1020, 2017.

[67] Qing Lei, Ji-Xiang Du, Hong-Bo Zhang, Shuang Ye, and Duan-Sheng Chen. A survey of vision-based human action evaluation methods. *Sensors*, 19(19):4129, 2019.

[68] Chunxia Li and SangHyun Lee. Computer vision techniques for worker motion analysis to reduce musculoskeletal disorders in construction. In *Reston, VA: ASCEProceedings of the 2011 ASCE International Workshop on Computing in Civil Engineering, Miami, Florida, June 19-22, 2011— d 20110000*. American Society of Civil Engineers, 2011.

[69] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3595–3603, 2019.

[70] Xinming Li, SangHyeok Han, Mustafa Gül, and Mohamed Al-Hussein. Automated post-3D visualization ergonomic analysis system for rapid workplace design in modular construction. *Autom. Construction*, 98:160–174, Jan. 2019.

[71] Yanghao Li, Cuiling Lan, Junliang Xing, Wenjun Zeng, Chunfeng Yuan, and Jiaying Liu. Online human action detection using joint classification-regression recurrent neural networks. In *European Conference on Computer Vision (ECCV)*, pages 203–220. Springer, 2016.

[72] Yanshan Li, Tianyu Guo, Xing Liu, and Rongiie Xia. Skeleton-based action recognition with lie group and deep neural networks. In *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, pages 26–30. IEEE, 2019.

[73] Zhenqiang Li, Yifei Huang, Minjie Cai, and Yoichi Sato. Manipulation-skill assessment from videos with spatial attention network. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[74] Yalin Liao, Aleksandar Vakanski, and Min Xian. A deep learning framework for assessing physical rehabilitation exercises. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(2):468–477, 2020.

[75] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[76] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*, 2017.

[77] Jun Liu, Amir Shahroudy, Gang Wang, Ling-Yu Duan, and Alex C Kot. SSNet: scale selection network for online 3D action prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8349–8358, 2018.

[78] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention LSTM networks for 3D action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1656, 2017.

[79] Adrien Malaisé, Pauline Maurice, Francis Colas, and Serena Ivaldi. Activity recognition for ergonomics assessment of industrial tasks with automatic feature selection. *IEEE Robotics and Automation Letters*, 4(2):1132–1139, 2019.

[80] Hector P Martinez, Georgios N Yannakakis, and John Hallam. Don't classify ratings of affect; rank them! *IEEE transactions on affective computing*, 5(3):314–326, 2014.

[81] Pauline Maurice, Adrien Malaisé, Clélie Amiot, Nicolas Paris, Guy-Junior Richard, Olivier Rochel, and Serena Ivaldi. Human movement and ergonomics: An industry-oriented dataset for collaborative robotics. *The International Journal of Robotics Research*, 38(14):1529–1537, 2019.

[82] Iñaki Maurtua, Aitor Ibarguren, Johan Kildal, Loreto Susperregi, and Basilio Sierra. Human-robot collaboration in industrial applications: Safety, interaction and trust. *Int. J. Adv. Robot. Syst.*, 14(4):1–10, Jul. 2017.

[83] Effrosyni Mavroudi, Divya Bhaskara, Shahin Sefati, Haider Ali, and René Vidal. End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1558–1567, 2018.

[84] Alberto Mazzoldi, Danillo De Rossi, Federico Lorussi, EP Scilingo, and R Paradiso. Smart textiles for wearable motion capture systems. *AUTEX Research Journal*, 2(4):199–203, 2002.

[85] Rahil Mehrizi, Xi Peng, Zhiqiang Tang, Xu Xu, Dimitris Metaxas, and Kang Li. Toward marker-free 3D pose estimation in lifting: A deep multi-view solution. In *IEEE Int. Conf. Autom. Face Gesture Recognit.*, pages 485–491, 2018.

[86] Meng Meng, Hassen Drira, and Jacques Boonaert. Distances evolution analysis for online and off-line human object interaction recognition. *Image and Vision Computing*, 70:32–45, 2018.

[87] George Michalos, Sotiris Makris, Panagiota Tsarouchi, Toni Guasch, Dimitris Kontovrakis, and George Chryssolouris. Design considerations for safe human-robot collaborative workplaces. *Procedia CIRP*, 37:248–253, Dec. 2015.

[88] Farhood Negin, Michal Koperski, Carlos F Crispim, Francois Bremond, Sehan Coşar, and Konstantinos Avgerinakis. A hybrid framework for online recognition of activities of daily living in real-world settings. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 37–43. IEEE, 2016.

[89] Bingbing Ni, Vignesh R Paramathayalan, and Pierre Moulin. Multiple granularity analysis for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 756–763, 2014.

[90] U.S. Bureau of Labor Statistics. Back injuries prominent in work-related musculoskeletal disorder cases in 2016, Aug 2018.

[91] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3153–3160, 2011.

[92] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[93] Jia-Hui Pan, Jibin Gao, and Wei-Shi Zheng. Action assessment by joint relation graphs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6331–6340, 2019.

[94] German I Parisi, Sven Magg, and Stefan Wermter. Human motion assessment in real time using recurrent self-organization. In *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)*, pages 71–76. IEEE, 2016.

[95] Paritosh Parmar and Brendan Tran Morris. Measuring the quality of exercises. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2241–2244. IEEE, 2016.

[96] Paritosh Parmar and Brendan Tran Morris. What and how well you performed? a multitask learning approach to action quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–313, 2019.

[97] Paritosh Parmar and Brendan Tran Morris. Learning to score olympic events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28, 2017.

[98] Behnoosh Parsa and Ashis G Banerjee. A multi-task learning approach for human activity segmentation and ergonomics risk assessment. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2021.

[99] Behnoosh Parsa, Athma Narayanan, and Behzad Dariush. Spatio-temporal pyramid graph convolutions for human action recognition and postural assessment. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1069–1079. IEEE, 2020.

[100] Behnoosh Parsa, Ekta U Samani, Rose Hendrix, Cameron Devine, Shashi M Singh, Santosh Devasia, and Ashis G Banerjee. Toward ergonomic risk prediction via segmentation of indoor object manipulation actions using spatiotemporal convolutional networks. *IEEE Robotics and Automation Letters*, 4(4):3153–3160, 2019.

[101] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[102] Alonso Patron-Perez, Marcin Marszalek, Ian Reid, and Andrew Zisserman. Structured

learning of human interactions in tv shows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2441–2453, 2012.

[103] Fotini Patrona, Anargyros Chatzitofis, Dimitrios Zarpalas, and Petros Daras. Motion analysis: Action detection, recognition and evaluation based on motion capture data. *Pattern Recognition*, 76:612–622, 2018.

[104] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[105] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. *arXiv preprint arXiv:1805.06485*, 2018.

[106] L Peppoloni, A Filippeschi, E Ruffaldi, and CA Avizzano. A novel wearable system for the online assessment of risk for biomechanical load in repetitive efforts. *International Journal of Industrial Ergonomics*, 52:1–11, 2016.

[107] Hamed Pirsiavash, Carl Vondrick, and Antonio Torralba. Assessing the quality of actions. In *European Conference on Computer Vision*, pages 556–571. Springer, 2014.

[108] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[109] Gessieli Possebom, Airton dos Santos Alonço, Sabrina Dalla Corte Bellochio, Tiago Gonçalves Lopes, Dauto Pivetta Carpes, Rafael Sobroza Becker, Antonio Robson Moreira, Tiago Rodrigo Francetto, Fernando Pissetti Rossato, and Bruno Christiano Corrêa Ruiz Zart. Comparison of methods for postural assessment in the operation of agricultural machinery. *Journal of Agricultural Science*, 10(9), 2018.

[110] Adam Poulos, Cameron Brown, Daniel McCulloch, and Jeff Cole. Context-aware augmented reality object commands, October 17 2017. US Patent 9,791,921.

[111] Andrea Prati, Caifeng Shan, and Kevin I-Kai Wang. Sensors, vision and networks: From video surveillance to activity recognition and health monitoring. *Journal of Ambient Intelligence and Smart Environments*, 11(1):5–22, 2019.

[112] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 401–417, 2018.

[113] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial intelligence review*, 43(1):1–54, 2015.

[114] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the Conference on Neural Information Proceesing Systems (NeurIPS)*, pages 1–9, 2015.

[115] S Robla-Gómez, Victor M Becerra, Jose R Llata, Esther Gonzalez-Sarabia, Carlos Torre-Ferrero, and Juan Perez-Oria. Working together: A review on safe human-robot collaboration in industrial environments. *IEEE Access*, 5:26754–26773, Nov. 2017.

[116] Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 3108–3116, 2016.

[117] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3433–3441, 2017.

[118] Akram Sadat Jafari Roodbandi, Forough Ekhlaspour, Maryam Naseri Takaloo, and Samira Farokhipour. Prevalence of musculoskeletal disorders and posture assessment by qec and inter-rater agreement in this method in an automobile assembly factory: Iran-2016. In *Congress of the International Ergonomics Association*, pages 333–339, 2018.

[119] Michael S Ryoo, AJ Piergiovanni, Juhana Kangaspunta, and Anelia Angelova. Assemblenet++: Assembling modality representations via attention connections. *arXiv preprint arXiv:2008.08072*, 2020.

[120] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, Zahra Moayed, and Reinhard Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, 172:88–97, 2018.

[121] Mohammad Sabokrou, Mohammad Khalooei, and Ehsan Adeli. Self-supervised representation learning via neighborhood-relational encoding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8010–8019, 2019.

[122] K Martin Sagayam and D Jude Hemanth. Hand posture and gesture recognition techniques for virtual reality applications: a survey. *Virtual Reality*, 21(2):91–107, 2017.

[123] Karlheinz Schaub, Gabriele Caragnano, Bernd Britzke, and Ralph Bruder. The european assembly worksheet. *Theoretical Issues in Ergonomics Science*, 14(6):616–639, 2013.

[124] Ali Shafti, Ahmad Ataka, Beatriz Urbistondo Lazpita, Ali Shiva, Helge A. Wurdemann, and Kaspar Althoefer. Real-time robot-assisted ergonomics. In *IEEE Int. Conf. Robot. Autom.*, 2019.

[125] Jing Shao, Kai Kang, Chen Change Loy, and Xiaogang Wang. Deeply learned attributes for crowded scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4657–4666, 2015.

[126] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1058, 2016.

[127] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 761–769, 2016.

[128] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1227–1236, 2019.

[129] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learning Representations*, 2014.

[130] Ashish Kumar Singh, ML Meena, Himanshu Chaudhary, and GS Dangayach. Ergonomic assessment and prevalence of musculoskeletal disorders among washer-men during carpet washing: guidelines to an effective sustainability in workstation design. *International Journal of Human Factors and Ergonomics*, 5(1):22–43, 2017.

[131] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3637–3646, 2017.

[132] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[133] Peregrin Spielholz, Barbara Silverstein, Michael Morgan, Harvey Checkoway, and Joel Kaufman. Comparison of self-report, video observation and direct measurement methods for upper extremity musculoskeletal disorder physical risk factors. *Ergonomics*, 44(6):588–613, Jun. 2001.

[134] Ekaterina H Spriggs, Fernando De La Torre, and Martial Hebert. Temporal segmentation and activity classification from first-person sensing. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 17–24. IEEE, 2009.

[135] Waqas Sultani and Mubarak Shah. What If We Do Not Have Multiple Videos of the Same Action?–Video Action Localization Using Web Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1077–1085, 2016.

[136] Moritz Tenorth, Jan Bandouch, and Michael Beetz. The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *IEEE Int. Conf. Comput. Vis. Workshops*, pages 1089–1096, 2009.

[137] Tensorflow. tensorflow/tensorflow.

[138] Juan R Terven and Diana M Córdova-Esparza. Kin2. a kinect 2 toolbox for matlab. *Sci. Comput. Prog.*, 130:97–106, 2016.

[139] 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 (ICCV)*, pages 4489–4497, 2015.

[140] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018.

[141] Khai N Tran, Ioannis A Kakadiaris, and Shishir K Shah. Part-based motion descriptor image for human action recognition. *Pattern Recognition*, 45(7):2562–2572, 2012.

[142] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *Speech Synthesis Workshop*, 2016.

[143] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 588–595, 2014.

[144] Michalis Vrigkas, Christophoros Nikou, and Ioannis A. Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, 2015.

[145] Bo Wan, Desen Zhou, Yongfei Liu, Rongjie Li, and Xuming He. Pose-aware multi-level feature network for human object interaction detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9469–9478, 2019.

[146] Pichao Wang, Wanqing Li, Philip Ogunbona, Jun Wan, and Sergio Escalera. Rgb-d-based human motion recognition with deep learning: A survey. *Computer Vision and Image Understanding*, 171:118–139, 2018.

[147] Wenguan Wang, Hailong Zhu, Jifeng Dai, Yanwei Pang, Jianbing Shen, and Ling Shao. Hierarchical human parsing with typed part-relation reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8929–8939, 2020.

[148] Kokum Weeratunga, Anuja Dharmaratne, and Khoo Boon How. Application of computer vision and vector space model for tactical movement classification in badminton. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 76–82, 2017.

[149] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer vision and image understanding*, 115(2):224–241, 2011.

[150] Yuxin Wu, Alexander Kirillov, Francisco Massa, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[151] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[152] Xiang Xiang, Ye Tian, Austin Reiter, Gregory D Hager, and Trac D Tran. S3d: Stacking segmental p3d for action quality assessment. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 928–932. IEEE, 2018.

[153] Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. The microsoft 2017 conversational speech recognition system. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5934–5938. IEEE, 2018.

[154] Wenkai Xu and Eung-Joo Lee. A novel method for hand posture recognition based on depth information descriptor. *KSII Transactions on Internet & Information Systems*, 9(2), 2015.

[155] Zhenjia Xu, Zhijian Liu, Chen Sun, Kevin Murphy, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Unsupervised Discovery of Parts, Structure, and Dynamics. 2018.

[156] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[157] Yang Yang, Imran Saleemi, and Mubarak Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1635–1648, 2012.

[158] Houpu Yao. *Robust and Generalizable Machine Learning Through Generative Models, Adversarial Training, and Physics Priors*. PhD thesis, Arizona State University, 2019.

[159] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias. Safety in human-robot collaborative manufacturing environments: Metrics and control. *IEEE Trans. Autom. Sci. Eng.*, 13(2):26754–26772, Apr. 2016.

[160] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2117–2126, 2017.

[161] Shugang Zhang, Zhiqiang Wei, Jie Nie, Lei Huang, Shuang Wang, and Zhen Li. A review on human activity recognition using vision-based method. *Journal of healthcare engineering*, 2017, 2017.

[162] Weichen Zhang, Zhiguang Liu, Liuyang Zhou, Howard Leung, and Antoni B Chan. Martial arts, dancing and sports dataset: A challenging stereo and multi-view dataset for 3d human pose estimation. *Image and Vision Computing*, 61:22–39, 2017.

[163] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1077–1085, 2017.

[164] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Recognize actions by disentangling components of dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6566–6575, 2018.

[165] Wu Zheng, Lin Li, Zhaoxiang Zhang, Yan Huang, and Liang Wang. Relational network for skeleton-based action recognition. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 826–831, 2019.

[166] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

[167] Penghao Zhou and Mingmin Chi. Relation parsing neural network for human-object interaction detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 843–851, 2019.

[168] Youding Zhu, Behzad Dariush, and Kikuo Fujimura. Kinematic self retargeting: A framework for human pose estimation. *Computer Vision and Image Understanding*, 114(12):1362–1375, 2010.