Doctoral Dissertation

A Manual Assembly Process Virtual Training System with Automatically Generated Augmented Feedback

Raveekiat Singhaphandu

Supervisor: Van-Nam Huynh

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Knowledge Science
September 2024

Dissertation Abstract

Manual assembly training traditionally relies on experienced operators to guide trainees through task demonstrations, trials, evaluations, and discussions. This method, while effective, is limited by the availability of experts. Current virtual training systems (VTS) focus on delivering rich multimedia content for task demonstrations, reducing dependence on experts. However, these systems often lack automated, comprehensive, augmented feedback for trainees.

This research introduces EXAMINER (**EX**pert Independent Manual **A**sse**M**bly V**I**rtual Trai**NER**), a system that objectively evaluates and provides feedback on trainees' motor and cognitive skills in manual assembly tasks. By automating feedback, EXAMINER enhances training accessibility and reduces reliance on experts. This study explores the digitization of human skills, objective measurement techniques, and the integration of these elements into an effective training system. The proposed system is evaluated for its ability to deliver appropriate feedback based on trainee performance, aiming to improve training outcomes and adoption rates.

The resulting framework consists of the following components: skill digitization, skill comparison, feedback provider, and multimedia training material. The implementation focuses on the first three components, ensuring their seamless integration. The framework implementation utilized methodologies for skill digitization using a video camera, employing standard and contemporary techniques such as deep learning in computer vision for human pose estimation, recurrent neural networks for activity recognition, and computer vision for contextual sensing. Each underlying subcomponent shows promising performance.

The digitization process is critical because it is the foundation for subsequent skill analysis and comparison between trainees and experts. In analyzing these operations, the study takes a novel approach, employing algorithms such as edit distance and dynamic time warping to identify and quantify skill differences. This methodology enables a more in-depth understanding of manual assembly cognitive and motor skill differences.

Another contribution of this research is the introduction of the I-MA task data model. This model enhances the framework's adaptability across diverse training scenarios and revolutionizes how information is systematically organized and utilized within I-VTS. The modular design of the framework, emphasizing interconnected yet distinct components, significantly enhances system flexibility and scalability, catering to a wide range of training needs and environments.

In summary, this research offers a comprehensive, flexible, and efficient I-VTS framework, representing a significant leap forward in virtual training systems. The framework utilizes advanced digitization techniques, detailed skill analysis, and user-friendly augmented feedback to address current gaps in I-MA training and establish a new standard for future developments in the field.

Keywords: Deep learning, Computer vision, Manual assembly, Virtual training, Industry 4.0

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Training human operators to perform manual assembly(MA) with desired assembly skill, judgment, and dexterity is crucial in Industrial 4.0 [108]. MA is an action of composing previously manufactured parts into a complete product using a human operator [30]. It exists in an assembly unit that requires the flexibility of a human operator to produce a product with a small lot size and highly customizable variations. A trainee requires face-to-face training offered by the expert as they can directly judge and guide them to reach a desirable MA skill level. It introduces various limitations, including

1. **Scheduling Conflicts**: The activity must occur when both parties are available, making it difficult to coordinate schedules.

2. **Limited Training Scale**: One expert can only handle a few trainees at a time, requiring careful observation and evaluation, thus limiting the scalability of training.

3. **Productivity Disruption**: Allocating experts to conduct training can lead to reduced productivity in their regular roles.

4. **Uncertainty and Regulations**: Disruptions from events like pandemics can necessitate reduced close-contact activities, impacting training schedules and methods.

5. **Consistency and Quality**: Variability in training quality due to differences in expert instructors can potentially lead to inconsistent training outcomes.

6. **Resource Intensive**: High resource consumption, including time, personnel, and physical materials, can be costly.

7. **Trainee Learning Pace**: Difficulties in accommodating different learning paces of trainees, as face-to-face training might not allow for personalized, self-paced learning.

A virtual training system (VTS) is a paradigm where the simulation of conventional training is used to solve any previously introduced limitation. It aims to reduce face-to-face physical training. Before this chapter directly states the problem, motivation, and objective of this dissertation, it first introduces related concepts, including

- Manual assembly under Industry 4.0,
- Industrial skill training,
- Virtual training system.

1.1 Manual Assembly Under Industry 4.0

The term ”Industry 4.0” was introduced in 2011 as it refers to the next industrial revolution that is about to take place. The first industrial revolution in the 18^{th} century was the introduction of steam power and mechanical production facilities. The second industrial revolution in the 1870s was the introduction of electricity, mass production, and an assembly line. The third revolution around the 1970s was an introduction to a computer and automation known as ”the digital revolution.” The communication between people, machines, and resources is a fundamental aspect of Industry 4.0. This paradigm shift transitions from centrally controlled to decentralized production processes, incorporating critical components to form a smart factory [64]. Key elements include:

- *Internet of things*: Enables sensors and actuators to interact and cooperate with corresponding components, enhancing connectivity and data exchange.
- *Cyber-physical system*: Integrate computational and physical processes, merging the virtual and physical worlds for improved efficiency and real-time monitoring.,
- ***Smart factory***: Resulting from the integration of IoT and CPS, a smart factory is context-aware, assisting people and machines in executing tasks and providing optimized decisions through interconnected computer systems operating in the background.

These components collectively enhance the flexibility, efficiency, and intelligence of manufacturing processes, driving the evolution of modern industry.

Every industrial revolution impacts all related parties significantly. The third industrial revolution dramatically transformed product supply by lowering costs and offering a wider range of mass-produced products [80]. Consequently, manufacturing firms face lower margins due to increased competition, as competitors can also mass-produce similar products. To stay competitive, firms may adopt product personalization as an answer to diverse needs. For instance, a specific automotive part can fit various models with slight component changes. Adapting the assembly process to meet flexible demands is neither economical nor simple to automate, thus requiring the flexibility of manual labor. Laborers must adapt quickly to changing product demands. The Industry 4.0 smart factory aids operators by providing necessary information in various formats to assist the manufacturing process. For example, information can be digitized as an interactive step-by-step electronic manual, automatically served to the operator when needed [111].

In summary, Industry 4.0 marks a significant advancement in manufacturing, enabling personalized production while maintaining efficiency and competitiveness. The integration of IoT, CPS, and smart factories enhances adaptability and responsiveness to market demands, supporting the rapid evolution of industrial practices. This revolution benefits manufacturing firms by improving margins and productivity and empowers workers with advanced tools and information for enhanced performance.

1.2 Manual Assembly Industrial Skill Training

Training an inexperienced operator in a particular MA task typically requires face-to-face instruction from an experienced expert to achieve desirable dexterity. Dexterity comprises both cognitive and motor skills [6]. Cognitive skills for manual assembly involve an individual's ability to recall the assembly process and recognize the state of the assembly, including the sequence of steps, materials, and tools used [122]. Motor skills are crucial for manual assembly tasks, necessitating precise movements and coordination to assemble components accurately and efficiently. The basic MA skill training is typically conducted in the following steps:

1. The expert demonstrates the assembly task while the trainee observes and memorizes the operation sequences.

2. The trainee performs the task under the expert's instruction, observation, and guidance. The expert evaluates the trainee's performance and progressively reduces guidance to encourage independent learning.
3. The trainee must pass a dexterity assessment, meeting the desired cognitive and motor skills as evaluated by the expert.
4. The training is repeated if the trainee fails to meet the expected cognitive and motor skills or dexterity.

In summary, effective MA skill training requires a structured approach. This method ensures that trainees develop both cognitive and motor skills through observation, guided practice, and evaluation. This method also ensures that trainees can achieve the necessary dexterity to perform assembly tasks accurately and efficiently, thereby maintaining high standards in the manufacturing process.

1.3 Virtual Training System

Virtual Training Systems (VTS) offer innovative solutions for training in various fields by leveraging technology to create immersive and interactive learning environments. These systems are particularly valuable in situations where physical interaction between the trainer and trainee is limited or impractical. This section explores the properties and applications of VTS, highlighting their effectiveness and current limitations in motor performance training. VTS is a training paradigm that includes the following properties:

- The trainer and trainee can be physically or temporally separated [7].
- The training takes place within a simulated or augmented environment [52].

VTS is effective for training that does not require physical interaction. Standardized tests or quizzes are typically used to measure learning outcomes, resulting in summative assessments or terminal feedback. However, VTS is not suitable for all types of learning [89]. Motor performance training requires formative assessments, such as manual assembly (MA) tasks, sports, entertainment, and medical procedures. Experts must provide concurrent and terminal feedback to improve the trainee's performance. Currently, VTS allows experts in different geographical areas to receive video recordings of a trainee's performance. The expert can then evaluate and provide feedback, either concurrently or later, improving training outcomes.

Moreover, VTS in MA research and commercialization heavily uses immersive multimedia training materials for introductory training. Immersive materials, such as virtual reality (VR) [73], [95], [113], [118] and augmented reality (AR) head-mounted displays [31], [39], [40], visualize complex assembly processes and structures. Using these technologies as introductory training mitigates dependency on experts during the demonstration phase and allows for scalable training.

In summary, VTS offers a flexible and scalable solution for training in various environments, particularly when physical interaction is not essential. It leverages immersive technologies to provide rich, interactive content, thus reducing the reliance on experts and enabling trainees to learn and improve their skills more efficiently.

1.4 Problem Statement

In industrial manual assembly, training inexperienced operators typically requires extensive face-to-face interaction with experienced experts. This conventional training approach presents several limitations that hinder efficiency and scalability. While Virtual Training Systems (VTS) offer potential solutions, they also come with their own challenges, particularly in maintaining training effectiveness and accessibility.

Despite VTS reducing dependency on human experts, particularly during the introductory phase, experts are still required for subsequent conventional training. This introduces several limitations:

1. Training must occur when both parties are available.
2. The training scale is limited; one expert can only manage one trainee at a time, requiring careful observation and evaluation.
3. Allocating experts for training disrupts overall productivity.
4. Training can be disrupted by unforeseen events and regulations, such as during a pandemic when close contact activities are restricted.

Due to limited expert availability, training offers and durations are constrained, leading to unproductive waiting times for trainees who cannot perform MA tasks without proper training. VTS currently addresses these issues by:

- Enabling geographical separation of experts and trainees through the bi-directional transmission of necessary information and training materials.

- Scaling introductory training using immersive and multimedia training mediums.

Although VTS can significantly reduce expert dependency during the introductory phases, it still relies on experts to provide performance observation, evaluation, and feedback remotely or at a different time. Additionally, the high cost and complexity of current VTS setups can limit accessibility, particularly for small and medium-sized enterprises.

In summary, while VTS addresses several limitations of traditional training methods, it still faces challenges related to expert availability, training scalability, and cost.

1.5 Motivation

Based on the problem statement, this dissertation aims to address the issue of limited expert availability for face-to-face training by reducing the necessity for expert supervision and evaluation during manual assembly tasks. The final system will enable trainees to perform manual assembly using the VTS without relying on an expert, mitigating the pain points highlighted in the problem statement that hinder VTS adoption in the manufacturing industry.

Additionally, this dissertation will document the process of creating a framework for realizing a VTS for manual assembly tasks. The resulting framework will assist interested parties in designing and implementing VTS tailored to their specific use cases. As VTS consists of various components requiring integration, some components are readily available for implementation, while others may require modifications or need to be developed from scratch.

An important aspect of this dissertation is to make VTS economically accessible. By using cost-effective, off-the-shelf components, such as video cameras and personal computers, the system reduces the total cost of ownership, making it accessible for small and medium-sized enterprises. This approach ensures that VTS can be widely adopted, offering significant economic benefits by reducing the need for expensive, proprietary hardware and extensive expert involvement.

This dissertation will explore the available components, incorporate and modify them as needed, and implement new components to meet the specific requirements of our use case. This process will involve extensive testing, modification, and implementation to ensure the system's effectiveness and suitability.

1.6 Objective

This dissertation aims to develop and propose a comprehensive framework for an expert-independent Virtual Training System (VTS) for Manual Assembly (MA), named **EXAMINER** (**EX**pert Independent Manual **A**sse**M**bly V**I**rtual Trai**NER**). The objectives are:

1. **Integration and Innovation:** To integrate various concepts such as assembly state context sensing, performance comparison between subjects, augmented feedback for cognitive and motor skill learning, and multimedia training material into a cohesive framework.
2. **Evaluation:** To evaluate the effectiveness of EXAMINER's components through a case study of industrial-like robot parts assembly, focusing on their ability to capture and assess trainee performance.
3. **Economic Accessibility:** To ensure the system is cost-effective and accessible, making it suitable for small and medium-sized enterprises by utilizing off-the-shelf components.
4. **Reduction in Expert Dependency:** To reduce the need for expert supervision and evaluation, thereby addressing the limitations of traditional face-to-face training and improving the scalability and efficiency of MA training.

The goal is to establish EXAMINER as a viable solution for effective and autonomous MA training, enhancing training methodologies and accessibility across various industrial contexts.

The introduction chapter outlines the significance of training human operators for manual assembly (MA) in Industry 4.0. MA involves assembling manufactured parts into a product, necessitating flexibility due to small lot sizes and customizable variations. Traditional face-to-face training with experts is essential but presents several limitations, such as scheduling conflicts, limited scalability, productivity disruption, and high resource consumption. Virtual Training Systems (VTS) offer potential solutions by reducing reliance on expert presence, though challenges remain in maintaining training effectiveness and accessibility.

The chapter introduces key concepts, including:

1. **Manual Assembly Under Industry 4.0:** This section discusses the Industrial Revolution's historical context and evolution, emphasizing the role of IoT, cyber-physical systems, and smart factories in enhancing manufacturing flexibility, efficiency, and intelligence.

2. **Manual Assembly Industrial Skill Training:** This section highlights the structured approach to training MA skills, focusing on the development of cognitive and motor skills through expert-led demonstrations, guided practice, and assessments.

3. **Virtual Training System:** This section explores VTS as a solution for training in various fields, emphasizing its properties and applications and the use of immersive multimedia training materials to scale introductory training phases.

The **Problem Statement** section identifies the limitations of traditional training and the partial solutions offered by VTS. Despite its benefits, VTS still depends on experts for performance observation, evaluation, and feedback, with additional challenges in cost and complexity limiting accessibility.

The **Motivation** section outlines the goal to mitigate expert availability issues by creating an expert-independent VTS, documenting the framework process, and ensuring economic accessibility using off-the-shelf components.

The **Objective** section aims to propose a comprehensive framework for an expert-independent VTS, named EXAMINER, integrating various concepts, evaluating its components through a case study, ensuring economic accessibility, and reducing expert dependency.

The chapter establishes the foundation for developing an expert-independent VTS for MA, addressing traditional training limitations, and proposing a framework to enhance training methodologies and accessibility in the manufacturing industry.

1.7 Outline

After the introduction chapter, the dissertation is structured as follows

- Chapter **2** - *Related work*,
- Chapter **3** - *Requirement Engineering*,
- Chapter **4** - *System Analysis and Design*,
- Chapter **5** - *Digitization of Operator Skills*,
- Chapter **6** - *Comparison of Digitized Skill*,
- Chapter **7** - *Automatically Generated Augmented Feedback for Reporting a Training Outcomes*, and

- Lastly, chapter **8** - *Conclusion, Discussion, Limitation, and Future Work*

Chapter 2

Related Work

This chapter summarized related work on Industrial VTS(I-VTS), highlighting the distinctive features and the fundamental concepts required to implement it. Topics are as follows,

1. Industrial virtual training system,
2. Virtual training environment,
3. Augmented feedback for skill learning,
4. Skill comparison between the expert and trainee,
5. Information acquisition from the industrial space,
6. Multi-media training material, and
7. Industrial virtual training system framework

2.1 Industrial Virtual Training System

The primary motivation for creating an Industrial Virtual Training System (I-VTS) is to replace traditional face-to-face training methods and eventually reduce dependency on human experts, especially during the introductory phase of training [67]. Early systems promoted the idea that they could offer more repetition and enhance understanding by utilizing multiple presentation mediums, ultimately leading to improved training outcomes in terms of the learning curve and reduced errors [31], [40], [102].

I-VTS permits the delivery of training through different immersive and non-immersive presentation mediums, providing greater access to training and resulting in more satisfying outcomes for trainees. Most closely related systems can be categorized as Manual Assembly (MA) assistant systems. These systems assist operators by providing context-sensitive assembly guidance through interactive assembly manuals [66], [77]. While MA assistant systems are installed on the production assembly line to assist operators during tasks, I-VTS is exclusively dedicated to training, enabling trainees to practice and learn independently without disrupting ongoing production [66].

Training an operator for MA requires mastering dexterity. Dexterity is comprised of a manual assembly of cognitive and motor skills, **shortened as cognitive and motor skills**. Cognitive skills involve the mental capacity to perceive and execute appropriate MA operations, including recognizing assembly sequences, parts, tools, locations, and corresponding motions [21]. The I-VTS for cognitive skills operates as follows:

1. Captures experts' knowledge using multimodal sensing capabilities.
2. Transfers the expert's knowledge to the trainee through multimedia electronic training material.
3. Conducts a summative evaluation of the trainee's performance by recording and comparing it to a predefined evaluation template or the expert's recorded performance [38], [70].

Motor skills involve the ability to perform precise movement trajectories of limbs and joints within a consistently desirable operation time and with appropriate force [18]. Training such skills requires additional capabilities, including:

1. Measuring and comparing the trajectory, force, and rotational velocity of joints objectively [31], [50], [82].
2. Providing extrinsic or augmented feedback to guide the trainee in improving motor performance [94].
3. Using tangible task-related objects, either genuine parts or replicas, to simulate the touch and feel of real assembly tasks [115].

I-VTS can circumvent limited training repetitions by replacing the expert with technology. Research communities and businesses have utilized conventional training mediums, summative assessments, immersive displays, and context-sensing systems to enable trainees to practice dexterity skills without waiting for an expert's demonstration, instruction, or evaluation.

Combining these technologies allows trainees to receive continuous and immediate feedback, enhancing their learning and performance.

It is essential to distinguish between systems designed for assistance and those intended solely for skill learning. Assistance systems provide instruction while the operator performs tasks, whereas skill learning systems are used exclusively during the learning phases. An assistance-based system can be adapted for learning by gradually reducing the information provided to prevent trainees from becoming overly dependent on augmented reality (AR) features.

In summary, I-VTS represents a significant advancement in training methodologies for manual assembly tasks. By leveraging immersive technologies and providing continuous, objective feedback, I-VTS offers a scalable and effective solution for developing trainees' necessary cognitive and motor skills, enhancing overall training efficiency and outcomes.

2.2 Virtual training environments

VTS utilizes various technological advancements to develop customized training environments that tackle specific challenges. These environments span from fully virtual to entirely non-virtual settings, each specifically designed to enhance the learning experience based on different limitations and demands. This subsection examines the unique characteristics and advantages of fully virtual, semi-virtual, and non-virtual training environments.

Firstly, the fully virtual environment requires trainees to participate solely in a simulated digital setting [18], [39], [52], [73], [86], [95], [113]. This configuration is most advantageous when there are restrictions on accessing real parts or actual production environments. It heavily depends on immersive technologies to accurately recreate actual interactions, making it appropriate for situations where safety, cost, or accessibility are concerns.

Second, the semi-virtual environment is an intermediary between completely virtual and non-virtual environments, combining aspects from both. The system combines physical and digital elements, such as 3D-printed replicas or digital simulations of actual objects, with the physically simulated or actual working environment [40], [70], [94], [102], [118]. This hybrid approach facilitates a flexible training experience, allowing for accurate simulations of crucial tasks while incorporating some physical interaction with realistic elements.

Finally, the non-virtual environment involves training in a genuine setting, either in a specifically designated area that mimics a real work environment or in an existing workplace utilized for training purposes. Trainees engage with genuine components and equipment, which may be designated for training purposes [31], [38], [39]. This environment is most beneficial for tasks demanding hands-on experience with materials and conditions.

VTS provides a wide range of training environments to meet the requirements and constraints imposed by their physical or economic environments. The fully virtual environment is ideal for scenarios with limited physical access, as it uses immersive technologies to simulate real-world interactions accurately. The semi-virtual environment bridges physical and digital elements, providing a balanced and adaptable training experience. Meanwhile, the non-virtual environment provides the most direct engagement with real-world materials and settings, making it ideal for tasks requiring extensive hands-on practice. Together, these environments enable a comprehensive training approach, ensuring learners can effectively acquire the necessary skills.

2.3 Augmented Feedback for Skill Learning

The concept of augmented feedback in motor training has been extensively reviewed, demonstrating its ability to significantly improve motor skill acquisition in various domains, including sports and dancing [29], [35]. Augmented feedback comprises two main aspects: technique and strategy. Techniques refer to the ways feedback is presented, such as visual, auditory, and haptic, while strategies pertain to the timing and frequency of feedback delivery, such as concurrent, terminal, or hybrid feedback.

Techniques of Augmented Feedback:

- **Haptic Feedback:** This involves feedback that stimulates the sense of touch and can restrain movement by applying opposing forces. For example, vibration is used in VTS to simulate the sensation of touching a machine [31].

- **Auditory Feedback:** Audio signals capture the operator's attention, often in the form of alarms. In VR-based human-robot collaboration training, auditory alarms emphasize safety and continuous operation in hazardous areas [73].

- **Semantic Feedback:** This includes descriptive text or speech feedback. For instance, positive feedback for step completion and error descriptions for incorrect part orientation are used in PCB assembly training [47].

- **Visual Feedback:** Visual cues, such as color codes and guided visual indicators, show the current training status and operation conditions. These cues can indicate correct/incorrect actions, minor/major mistakes, and warnings [73], [86], [87], [95].

Strategies of Augmented Feedback:

- **Concurrent Feedback:** Provided during task performance, this type of feedback can instruct the trainee in real time. However, excessive reliance on guidance can lead to dependency, so it is important to reduce guidance gradually based on training outcomes [31].

- **Terminal Feedback:** This feedback is given after task performance, summarizing the trainee's performance. It is a straightforward form of feedback that reports either knowledge of performance or outcomes [14], [28].

- **Hybrid Feedback:** A combination of concurrent and terminal feedback, this approach provides real-time guidance and post-performance summaries, enhancing the overall training process [11].

Augmented feedback is crucial for I-VTS and MA assistant systems. For example, haptic feedback can guide movements with adjusted assistance levels based on trainee experience, reducing errors in machining operations [31]. In PCB I-MA training, concurrent semantic feedback provides positive reinforcement for correct steps and descriptive error feedback for mistakes [47]. In VR-based human-robot collaboration training, auditory warnings and visual alarms emphasize safety and promote continuous operation [73].

MA assistant systems can also function as VTS by offering modes that solely evaluate the operator's performance without providing assistance. Typically, these systems provide concurrent feedback with guided visual cues. However, in evaluation mode, the system assesses training outcomes without guidance, focusing on the operational context [86], [87], [95].

Implementing an augmented feedback training system requires careful planning to prevent trainee dependency on the system. Feedback frequency and detail should be gradually reduced to ensure trainees develop the necessary skills independently. This strategic approach confirms the critical role of augmented feedback in enhancing training effectiveness across different settings.

2.4 Skill Comparison Between Recorded Expert and Trainee

Ensuring an objective evaluation of a trainee's cognitive and motor skills in manual assembly (MA) is essential. The standard for this evaluation can either be semantically programmed [34], [49], [59], [61] or digitized from an expert's recorded performance [75], [76]. Digitizing these skills typically requires different techniques due to their unique challenges. The section addresses the comparison of cognitive and motor skills separately. In addition, the section also address industrial vs. non-industrial skill comparison.

2.4.1 Cognitive Skills Comparison

Cognitive skills involve comparing the assembly step, parts used, and their location or orientation captured by various sensors. These skills can be digitized by capturing the differences in the physical context of MA task. For example, systems can automatically digitize human skeletal coordinates and the state of an assembly workpiece, allowing for comparisons across different subjects [37], [75]. Most virtual training environments implement sequential virtual assembly simulators, where the assembly sequence is strictly predefined. Any errors must be corrected before proceeding, ensuring trainees adhere to the correct sequence and method.

2.4.2 Motor Skills Comparison

Motor skills are more challenging to program and transfer semantically. Instead, these skills can be captured as time-trajectory data and compared across subjects. This approach is prevalent in sports and dance, where performance outcomes are directly proportional to knowledge performance [15], [16]. For instance, a baseball swing virtual trainer uses wearable sensors to record the swing's three-dimensional acceleration. The virtual trainer then provides terminal feedback by visualizing the time-trajectory comparison graph between the trainee and a professional batter [39]. Similarly, a virtual dance trainer compares kinematic features such as joint angles and rotations, reporting differences from a target movement as a score [15]. In industrial settings, tasks like composite material layup require strict adherence to location and motion within a set time frame. These tasks can benefit from motor skill comparisons using motion trackers to capture and digitize operation time and transitions between areas [76].

2.4.3 Industrial vs. Non-Industrial Skill Comparison

Evaluating the trainee's cognitive and motor skills objectively is essential for effective skill development in both industrial and non-industrial contexts. However, the methods and requirements for skill comparison in these two domains can vary significantly. In industrial settings, the focus is on tasks like manual assembly, where precision and adherence to specific protocols are critical. In contrast, non-industrial contexts such as sports, entertainment, and rehabilitation often emphasize different aspects of skill performance, like coordination and force application. This section explores the distinct approaches to skill comparison in these two areas.

- **Industrial Skill Comparison:** This involves comparing digitized cognitive and motor skills specific to industrial tasks. For example, in a composite material layup, the operator must perform the task precisely at a specific location and time, which requires digitizing and comparing motor skills [76].

- **Non-Industrial Skill Comparison:** The comparison is widely used in sports, entertainment, and rehabilitation. In these areas, the focus is often on the performer's motor skills and ability to follow a desired movement pattern. For example, in sports, knowledge performance is closely linked to the outcome of actions, like a baseball swing, which requires precise coordination of limbs and joints. External feedback from experts is crucial for correct performance interpretation, especially when external factors like wind can affect outcomes [22], [33]. In entertainment, dancers must synchronize their movements with music, and virtual training systems can help them match their motions to a lead dancer's template, assessed externally [15], [35]. Rehabilitation involves patients performing exercises accurately, often at home, with virtual training systems providing objective performance feedback to guide their recovery [72], [120].

In summary, skill comparison is vital for both industrial and non-industrial training environments. Cognitive skills are evaluated by comparing physical contexts, while motor skills are assessed through time-trajectory comparisons. This approach reduces the need for expert intervention, as performance can be objectively measured and feedback provided. The direct comparison method, although beneficial, often requires expert interpretation of the results for the trainee's benefit.

2.5 Information Acquisition for Virtual Training System

There are various ways to acquire information for a Virtual Training System (VTS), including vision-based and non-vision-based methods. Vision systems use computer vision techniques to extract and detect relevant features from image sensors. A VTS requires the ability to sense the physical context and use it to compare an operator's performance against each other or specified templates. For fine motor skills focusing on hand movements, video capture, and hand articulation techniques are beneficial [72], [110]. Some assistance or training systems also employ depth or range imaging technology to obtain body pose or assembly station context, such as the depth profile of each assembly stage and the ability to track objects present [61], [66], [77].

In contrast, non-vision systems use data from various body contact sensors and sensors attached to the assembly station to acquire information [23]. While non-vision techniques are faster and less computationally expensive, they are intrusive to the operator. The emergence of commodity VR headsets and AR head-mounted display systems (HMD) has accelerated the implementation of hybrid systems that fuse vision and non-vision inputs. Generally, AR-HMDs contain various vision and non-vision sensors to sense context, such as image, range, acceleration, angular velocity, and magnetic field intensity. These systems can articulate hand location and movement using additional sensors [39], [61], [73], [77], [95], [113], [118].

This section reviews enabling technologies that can acquire the training's human and physical contextual information, including human pose estimation, activity recognition, and context recognition.

2.5.1 Human Pose Estimation

This section further introduces vision-based human-related information acquisition commonly adopted in the VTS. Vision-based operator performance acquisition is digitizing the operator into the machine's comprehensible form of information. It is possible to estimate the location of the target human's body joints, parts, and limbs in multi-dimensional Cartesian coordinates. Depending on the use cases and the enabling technology, it is performed in 3-D or 2-D. This section will introduce enabling technologies, including the state-of-the-art motion capture system, the commodity system being phased out, and deep learning on monovision color cameras, a current research trend.

- *Motion-Capture System* - Motion-capture system (MOCAP) is an optical-based system targeting professional usage [104]. It is primarily used in the entertainment and professional industry, including gaming, movies, virtual reality, and professional sports, to capture precise motion and facial expressions. The system usually requires a studio-like environment with dozens of high-speed infrared (IR) cameras around the scene toward the target. The camera captures IR reflection or emittance from IR markers placed on the target's joints, limbs, and face landmarks. The location of each IR reflector then fused, forming a human skeleton.

- *Commodity System* - The commodity system is the system that is commercially available to the mainstream user, including Microsoft Kinect and Intel Realsense. These systems mainly consist of a color camera and a depth-sensing mechanism. Either projected IR patterns [32] or time-of-flight (TOF) [9] concepts can sense depth. The first concept is to project the triangular IR pattern onto the scene and then use another IR camera to capture the deviation of the pattern. The deviation of the triangle properties is used to calculate depth. The latter concept employs the TOF mechanism by measuring the time emitted IR dots travel from the emitter to the target and back to the IR camera. After obtaining the depth information, it is used to subtract the background, resulting in a human silhouette. The silhouette is then fused with the color camera's information and passed to the estimation model, resulting in a human pose. The system was sufficient for general use, requiring less precision. They are popular amongst academics but lack general adoption and are currently discontinued for the mainstream user. Today, both manufacturers have stopped releasing new products for the mainstream market.

- *Deep Learning on Monovision Color Camera* - This section will introduce current trends in the research community that aim to substitute the commodity system being phased out. Nowadays, the related research under HPE employs the concept of deep learning in computer visions with the data from a color camera represented in the form of an image frame or a sequence of image frames [91]. The model works by first recognizing the human on the scene, then identifying each human's joints' location, and lastly, constructing a skeletal structure for each recognized human [65]. For instance, it first recognizes all possible body joints and forms interconnected joints as limbs. Lastly, the whole human skeleton [63]. The selection of the mentioned model is highly based on the nature of the problem. However, some models work oppositely. The scene with a single human will work best with the first model. In contrast, the latter model will perform faster in settings with more humans. This study chooses to employ this technique to get the pose information for the activity recognition model

2.5.2 Human Activity Recognition

Human activity recognition(HAR) is a challenging and highly active research topic; hence, it has a significant practical implication in the cyber-physical system. There are various ways to perform it, and it can be categorized in multiple ways. However, there is no general agreement on the categorizing methodology. This report considers widely discussed categories, including the type of input and related recognition model for each input type, including wearable sensor-based and vision-based input.

- *Wearable sensor-based input* - The wearable sensor-based input or contact-based input is usually located on the recognizing target. For instance, an accelerometer is a sensor that provides the acceleration data m/s^2 in three-dimensional space. The sensor has been primarily used on smartphones and fitness trackers for a decade. It can successfully recognize basic or general human activity, such as sitting, walking, lying, etc., using only a single accelerometer with a proper recognition model. In addition, the transitional activity, including sit-to-stand, stand-to-lie, or else, can also be recognized [57]. In addition, placing an additional accelerometer on the target at the different locations to increase recognition space is possible. For the last decades, there are several common classifiers in HAR including k-nearest neighbors [25], Discriminant Analysis [19], Naïve Bayes [20], Support Vector Machine [74], Hidden Markov Models [24], and combination of classifiers as Joint Boosting [97]. However, these methods require feature engineering. The data requires a domain expert to preprocess and fit it into the model. At current, there are various successful attempts using multilayer perceptron (MLP) or ANN, including CNN [83], recurrent neural network (RNN) [78], and hybrid models such as the combination of CNN and RNN [105]. In contrast to common classifiers, the main benefit of an ANN classifier is its generality, which is the ability to automatically realize the features directly from the data and later infer them.

- *Vision-based Input* - The vision-based input or the remote method is the vision sensor usually located further away, facing the recognizing target. One of the main benefits of the input is the elimination of wearable tracking devices that can be cumbersome for some tasks. However, this type of input usually requires additional computer vision tasks to extract features before performing the recognition task. For instance, the HAR model can consider the shape of the changes in human bounding silhouette [46]. However, obtaining a silhouette itself is challenging. One emerging method is to transform the vision-based input into a skeletal pose using the previously popular commodity HPE system and deep learning-based HPE. The estimated pose is then proceeding to the machine learning (ML)-based HAR model [84]. This section will further investigate and introduce the related model based on deep MLP or deep learning.

- *Deep Learning for Human activity recognition* - Currently, various deep-learning architectures are suitable for HAR based on Multi-dimensional Skeletal-based HPE. For instance, CNN, RNN, and hybrid models combine CNN and RNN. In addition, autoencoder [60], and self-attention [99] architecture also being under focuses amongst academic.

2.5.3 Acquisition of Station Context

MA assistance systems and VTS must automatically detect the physical context and use it as input for further processing. This input is crucial for comparing an operator's performance against other operators or predefined templates and delivering context-based instruction.

- *Vision-Based Information Acquisition:* Vision-based sensors use computer vision techniques to extract relevant features from images. They are essential in semi-virtual and non-virtual environments for capturing visual data accurately. Microsoft Kinect and Nintendo WiiMote have been popular choices, with Kinect used for pose estimation and depth-based context sensing, and WiiMote for gesture-based interaction [61], [66], [77]. Despite their discontinuation, alternatives like Intel RealSense and other motion-sensing devices are available.

- *Non-Vision Information Acquisition:* Uses body contact sensors and sensors attached to the assembly station. These sensors are faster and less computationally expensive but can be intrusive. Wearable devices with head-mounted cameras provide a hybrid approach, measuring kinematic characteristics and providing a viewpoint for analysis [38].

The "Information Acquisition for Virtual Training System" section details various methods for capturing data essential for VTS operations, focusing on vision-based and non-vision-based techniques. Vision-based systems employ computer vision to extract features from images, which are useful for fine motor skills and assembly context. Non-vision systems use body contact sensors and sensors attached to assembly stations, offering faster and less computationally intensive data acquisition but at the cost of being more intrusive. Hybrid systems, leveraging advancements in VR and AR, combine these methods for comprehensive data capture. Key enabling technologies include human pose estimation, human activity recognition, and context recognition. These technologies are vital for digitizing and comparing operator performance, ultimately enhancing training efficiency and effectiveness in VTS.

2.6 Multimedia Training Material

A multimedia training medium is essential for transferring knowledge of manual assembly tasks. It can serve as introductory training for new trainees and as a guide for experts assembling new products, allowing for scaling out training without relying solely on expert demonstrations. Virtual Training Systems (VTS) for manual assembly focus on utilizing immersive training materials, such as Augmented Reality (AR) and Virtual Reality (VR), to demonstrate complex assembly processes. These technologies reduce the need for expert demonstrations and help trainees understand hidden structures within enclosures. Conventional materials, including paper-based manuals, process illustrations, video recordings, and 3-D animations, are typically used for simpler assembly tasks. This section addresses types of immersive multimedia training materials, including:

- **Augmented Reality (AR)**: Augmented Reality (AR) enhances the real-world environment by overlaying digital information, such as visual cues, 3-D models, and text descriptions, onto physical objects and surroundings. AR can be delivered through various devices, including handheld tablets, optical see-through head-mounted displays (HMDs), and in-situ projection systems. These technologies provide interactive and context-sensitive guidance, making it easier for trainees to understand and perform complex assembly tasks. By visualizing hidden structures and offering real-time feedback, AR significantly improves the efficiency and effectiveness of manual assembly training. Each of the delivering techniques can be further elaborated as follows:
  - **Handheld Tablets:** AR on tablets uses the camera to capture images and augment training information, such as visual cues, 3-D models, and short text descriptions, onto the display image. The

information adapts based on the assembly's location, orientation, and perceived state [31]. This technology allows trainees to visualize and understand the assembly process interactively, improving their comprehension and retention of the information.

  - **Optical See-Through Head-Mounted Displays (HMD):** These devices provide the same augmented information without the need for users to hold the device [77]. This hands-free approach enhances the training experience by allowing trainees to focus on the assembly task while receiving real-time guidance and feedback.
  - **In-Situ Projection:** In-situ projection involves projecting visual aids directly onto the work environment or assembly parts, creating an interactive and intuitive training experience. This method allows trainees to receive guidance without wearing any device, as the information is projected onto their workspace. This approach can highlight specific assembly steps, parts, and tools, providing immediate visual context [66].

- **Virtual Reality (VR)**: Virtual Reality (VR) creates a fully immersive digital environment, allowing users to experience and interact with a simulated world. Using VR headsets, trainees are transported into a virtual space replicating real-world scenarios and assembly tasks. VR is particularly effective for training in hazardous environments, where operators can practice without exposure to actual risks. It enhances spatial awareness and muscle memory, ensuring correct part selection and placement during assembly. The immersive nature of VR helps trainees build muscle memory and spatial awareness, leading to more efficient and accurate performance during actual assembly tasks. VR's immersive environment is beneficial for introductory training in hazardous settings, where operators must be aware of heavy tool motions and potential risks [73]. By immersing trainees in a controlled virtual environment, VR training can safely replicate dangerous situations, enabling operators to practice and develop their skills without exposure to actual hazards. VR is also useful for training in assembly sequences, ensuring correct part selection and placement [68], [95], [113].

In summary, multimedia training materials are critical for effectively conveying manual assembly knowledge from experts to trainees. By incorporating visual aids such as video recordings and animations, these materials enhance trainees' understanding. Research communities continue to introduce and refine immersive display technologies for industrial training. These technologies reduce the cognitive load associated with separate assembly instructions and enable the simulation of hazardous work environments and the visualization of hidden structures within a workpiece's enclosure. As a result, multimedia training materials play a vital role in improving the efficiency, safety, and effectiveness of manual assembly training in modern industrial settings.

2.7 Industrial virtual training system framework

To enhance understanding among researchers and practitioners, the development of a structured framework is essential. Such frameworks provide a systematic approach to realizing Industrial Virtual Training Systems (I-VTS), enhancing consistency, efficiency, and effectiveness in implementation. They also facilitate a deeper comprehension of the operational mechanics of these systems. In the context of I-VTS, these frameworks, often presented as diagrams and detailed specifications, can be categorized into three primary types based on their focus: hardware-software mapping and information flow, data communication governance, and skill development.

First, numerous studies provide hardware-software mapping and information flow diagrams [18], [23], [38], [59], [61], [70], [76], [77], [94], [95], [113], [119]. These visualizations crucially describe the interaction between hardware and software components within I-VTS, illustrating the system's architecture and the pathways of information flow. They enable stakeholders to clearly understand the proposed systems' essential techniques, components, and key features.

Second, several papers focus on data communication governance within I-VTS, outlining the structure of the data exchanges [23], [34], [38], [47], [61]. This includes frameworks that use knowledge-level models, such as ontologies, which provide flexibility in system implementation and future scalability. Others employ predefined data standards to support common industrial maintenance and assembly tasks, ensuring interoperability and consistency across implementations.

Third, a few studies address frameworks related to skill development, particularly the cognitive skills necessary for effective industrial maintenance and assembly performance [70], [76]. These frameworks are vital for designing training systems that focus on cognitive abilities required for managing complex industrial tasks.

Overall, these frameworks play a crucial role in articulating the complex structure of I-VTS, ensuring that the systems are functional and adaptable to future advancements and changes in industrial requirements.

2.8 Summary

The related work chapter reviews the literature on virtual training systems (VTS), focusing on industrial use cases. It introduces related systems and categorizes them based on their motivation and objectives. Recent research has advanced the affordability and availability of immersive display technologies like AR and VR, reducing cognitive load by providing immersive presentations of training material. The chapter identifies critical enabling technologies such as information acquisition from the assembly station and skill comparison between expert templates and trainees. It also discusses the incorporation of augmented feedback techniques in motion-related training, such as sports and dance. To the best of the authors' knowledge, no existing I-VTS for industrial manual assembly (I-MA) provides augmented feedback based on digitized skill comparisons. The underlying techniques, including human pose estimation, activity recognition, and sequence and motion comparison, will be detailed in the methodology chapters.

Chapter 3

Requirement Engineering

The process of requirements engineering is for dealing with the problem's complexity. It focuses on describing the purpose of the system concerning the user's views on the system, which results in the specification of the system that can be understood by the stakeholders [13]. The visionary scenarios will be developed and proposed based on the observed as-is scenario. Then, the use cases and requirements are identified in the following sections.

3.1 Proposed Solution

This dissertation develops a I-VTS called "**EX**pert Independent Manual **A**sse**M**bly V**I**rtual Trai**NER**" (EXAMINER). The system uses the available technological concepts with optimized adjustment interconnected, creating a I-VTS that can automatically provide extrinsic augmented feedback reporting the training outcomes. The hardware setup of the system, as visualized in figure 3.1 shows an example of a vision-based EXAMINER that use a front-facing camera to sense the assembly context by pointing it to the assembly scene and the operator.

The system is capable of fulfilling the following tasks,

- The system can provide extrinsic augmented feedback of knowledge performance by comparing the digitized performance in cognitive and motor skill comparison.
- The digitization of performance uses context sensing capabilities, including the color vision camera or else that is not intrusive to the digitized target.
- The application's graphical interface allows an expert to include and

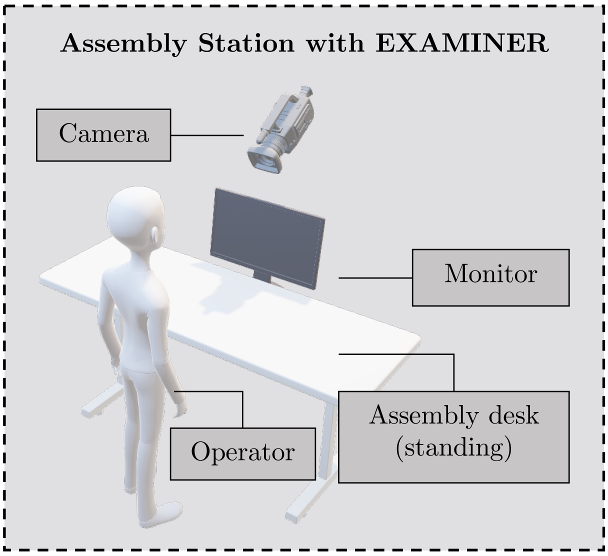


Figure 3.1: A vision-based EXAMINER.

present the additional training material, including descriptive text, still images or graphics, and recording video or animation for the trainee.

The proposed solution can benefit a trainee or an expert in industrial assembly in the following ways.

- A trainee can repeatedly study and train using the system. The system can eliminate the need for in-person, face-to-face training and assistance, which requires both parties to be available.
- Knowledge performance of an expert can be digitized and stored
- The Manufacturing industry can reduce productivity loss by allocating an expert to offer face-to-face training.
- The training can be scaled out and geographically distributed without relying on the number of experts available.

3.2 Scenario

A scenario is a concrete, focused, informal description of a feature of the system from the viewpoint of an actor [13]. In this section, the as-is scenarios describe a currently observed situation. Later, the visionary scenarios were developed and proposed to the stakeholders as a future system.

3.2.1 As-is Scenario

A manufacturing firm recently recruited new employees to help fulfill the customers' demands. The firm is about to release a new variation of the company's popular product, which was designed by the product design team. The team consists of multi-disciplinary experts from various sections of the firm. Together, they help develop the product prototype and translate it into a version that can be mass-produced. The knowledge co-created in the design phase has to be explicitly transferred to the production team at the factory. Here, the transfer of knowledge is done through one-on-one training with the expert. As of current, everything related to the assembly of a new product is readily available, including the material, assembly station, instruction manual, and video guidance. If the company has an expert operator to spare, they can directly assemble the product for the market. However, all experts are currently trying to fulfill the demands of the current generation of the product even though the company has already hinted to their customers that the new variation will launch soon. It only helps a slight decrease in demand. Only a few experts are available to offer one-on-one training to the new employees. As the firm does not want the new product release to suffer from the general availability, they decided to allocate more experts from the ongoing production lines to offer training by sacrificing the output of the current product. As a result, the current product starts to sell out, causing a supply-demand gap resulting in the loss of income. The firm hopes that there is some methodology to intelligently provide the training for the newcomers without relying too much on experts.

3.2.2 Visionary Scenario

The firm decided to equip the assembly station with the vision-based EXAMINER, consisting of a camera and a touch input display screen. As the firm has already started shifting to Industrial 4.0, all of this required equipment for EXAMINER was equipped and interconnected to the computation facility for other purposes. The firm only has to implement and tweak the EXAMINER to fit their needs based on the guidelines provided by the EXAMINER framework and then install them on the computational facility. The firm uses EXAMINER to capture the expert's manual assembly knowledge and encode it in cognitive and motor manual assembly skills (shortened as cognitive and motor skills). In addition to the encoded knowledge, EXAMINER attached the multimedia training material curated by the expert. The whole package of knowledge and material is then shipped and installed to the designated training destination. At the destination, the training manager makes sure that the system is ready to be used by the newcomers or the trainees. Once it is ready and meets the requirements, including ergonomics, safety, and space setup, the manager allows the trainee to train with the system. The system provides training material and instruction to the trainee. The trainee carefully navigates around the material and performs trail assembly along with the instruction at their own pace. Once the trainee is ready, the system turns from presentation to evaluation mode. Here, the trainee's performance is recorded and concurrently evaluated. At the end of the training iteration, the trainee received the automatically generated feedback on the knowledge performance and their intrinsic feedback, which is shown as the outcomes of the assembly. The firm can continue production without sacrificing the number of experts to offer one-on-one training. The training also scales out to all available assembly cells transformed to offer training. Training, however, is still being supervised by the training manager. However, a manager can handle multiple trainees at a time.

3.3 Use Case

A use case specifies all possible scenarios for a given piece of functionality; it helps clarify the system's requirements. The identified use cases are summarized in the Unified Modeling Language(UML) use case diagram shown in figure 3.2, which gives an overview of all identified use cases and the relationships among actors and use cases. The system consists of two actors. First, the expert who creates the assembly knowledge and related training materials by recording the assembly task and editing them by attaching training material. Second, the trainee, who consumes the training package, performs MA training and uses feedback provided by the system to improve the knowledge performance.

3.3.1 Use case 1: Record assembly task

The first use case has both experts and trainees as primary actors. They use the system to record their assembly skill. Once the recording is finished, the system automatically transcribes by digitizing the recorded performance into a step-by-step manual assembly process. The system requires digitizing cognitive and motor skills. Activity and context recognition enable the digitization of cognitive performance. At the same time, the latter requires an automatic recording of operation time and motion trajectory. Figure 3.3 provides an additional use case to the original figure 3.2 focusing on the expert.

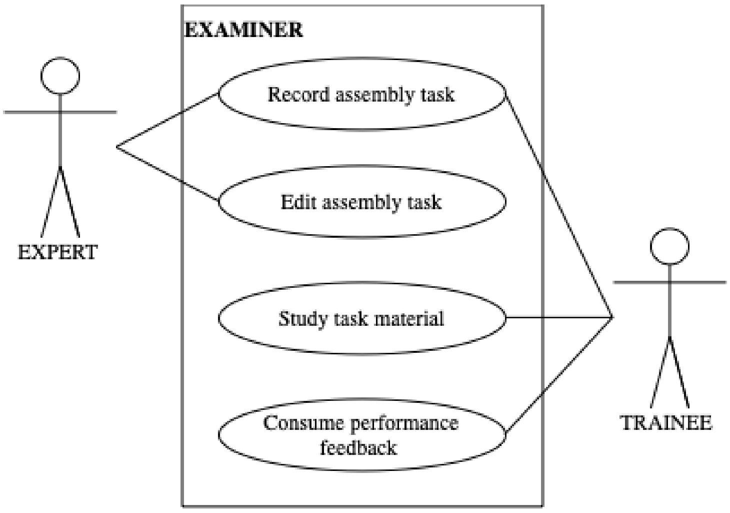


Figure 3.2: The overall use case for both expert and trainee.

Use case name	Record assembly task
Participating actor	Expert and trainee
Entry condition	The actor ready to record their MA knowledge performance for a particular task. The assembly cell was set up using the layout as instructed and consisting required material and tools for the task.
Flow of events	1. The actor start the system's recording tool. 2. The actor perform the MA of the task. 3. The actor stop the recording once the assembly is completed.
Exit condition	The raw performance were digitally saved in the persistent storage and digitized as cognitive and motor performance.

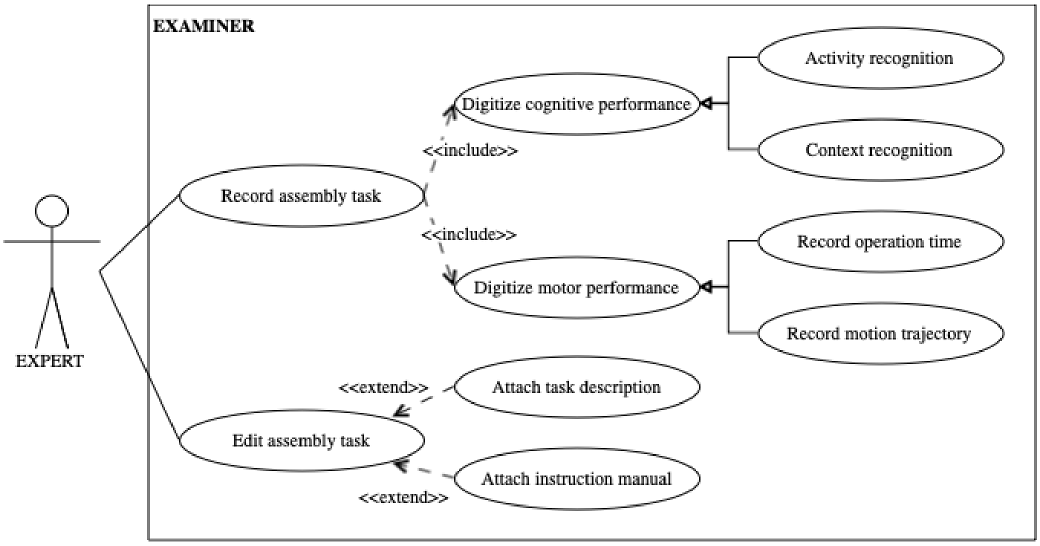


Figure 3.3: Extension of use case from figure 3.2, focusing on expert's use cases.

3.3.2 Use case 2: Edit assembly task

The expert later edits the recorded assembly task by associating additional information to each step or context, including the related training material and information the system cannot capture, including the name of the task and step. It serves as the information to be later visualized for both parties. As suggested by the diagram figure 3.3, the extension relationship indicates that the attachment of the task description and instruction manual is optional.

3.3.3 Use case 3: Study task material

The third use case has the trainee as a primary actor. Here, the trainee uses a training package for a particular MA task curated by the expert. Here, the trainee may navigate through each assembly step in the material while performing a trail assembly at their own pace. The judgment of knowledge outcomes is solely intrinsic in this phase.

3.3.4 Use case 4: Consume performance feedback

Once the trainee feels confident with the knowledge outcomes, they use the system to record their assembly performance as in the first use case. Once the digitization is complete, the performance will be compared with the expert's

Use case name	Edit assembly task
Participating actor	Expert
Entry condition	The performance recorded were successfully digitized by the underlying system.
Flow of events	1. The actor reviews each digitized step. 2. The actor includes each of missing context that is not digitized by the system including task name, and step name. 3. Optionally, the actor may include additional material including text description, images, and video or animation that is externally created.
Exit condition	The application added the context to digitized MA task and the electronic training material to each step if available, lastly saved in a persistent storage for later distribution.

Use case name	Study task material
Participating actor	Trainee
Entry condition	The training package was installed on the training system located at assembly station and the station is setup for the actor.
Flow of events	1. The actor study thorough the training material. 2. The actor perform trail on each assembly step.
Exit condition	Thec actor feel confident with the MA task intrinsicly judging by the knowledge outcomes.

Use case name	Consume performance feedback
Participating actor	Trainee
Entry condition	The actor feel confident to be evaluate by the system, and the assembly cell is return to the original state ready to be using for the recording of actor's performance.
Flow of events	1. The actor start the system's recording tool and perform the MA task. 2. Once the material is exhausted or reaching a target quantity, the actor stop the recording. In concurrently, the system digitized the actor skills. 3. The actor select the expert's MA performance to be evaluate. 4. The system evaluates and provide extrinsic augmented feedback to the actor.
Exit condition	The actor received the extrinsic augmented feedback from the system reporting the knowledge performance.

template. The system compares both cognitive and motor skills. In the end, the comparison result is reported using the extrinsic augmented feedback template that is easily interpretable. Trainees consume the feedback and incorporate it with their intrinsic feedback from the knowledge outcomes of the MA task. The trainee uses extrinsic and intrinsic feedback to improve their knowledge performance. Figure 3.4 provides an additional use case as discussed in the original figure 3.2 focusing on the trainee.

3.4 Functional Requirement

Functional requirements (FR) or functional specifications define a function consisting of a system's input, behavior, and output or its underlying components. The requirement can be documented as precisely as each calculation occurs or up to a high level that defines what should be accomplished. It serves as communication between the system designer and the implementer. Failing a functional requirement renders the system incomplete. This section

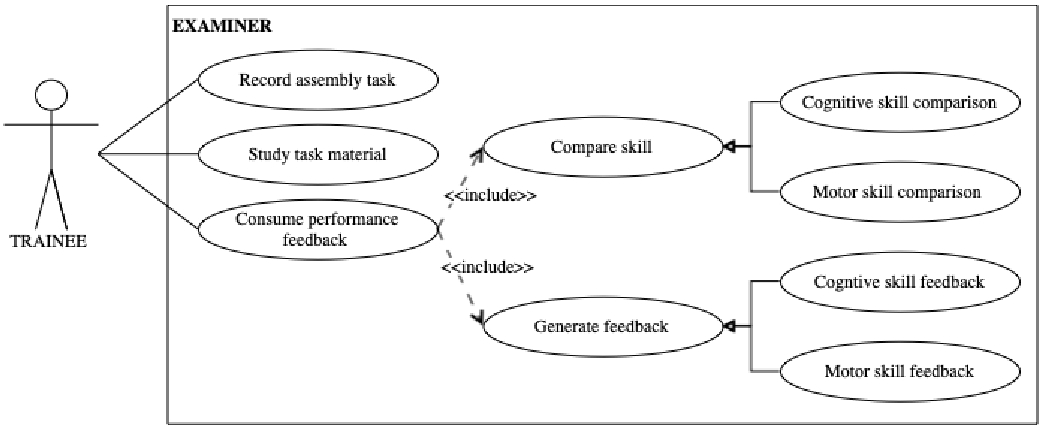


Figure 3.4: Extension of use case from figure 3.2, focusing on trainee's use cases.

introduces the high-level functional requirement of EXAMINER.

1. *Human motion capture* - The system must be able to capture and record the motion of an operator moving at the tracking area.

2. *Manual assembly activity recognition* - The system must detect the manual assembly-related activity. This includes the activity of **reaching** for material, tools, and area, **retracting** item to the assembly area, **using tool** to assembly, and **assembling** using hands only.

3. *Assembly context capture* - The system must capture state changes of MA-related context. This includes changes of availability as present and not present of material, tools, and the final product at the assembly station.

4. *Manual assembly step recognition* - The system must identify the step given the recognized activity and the MA-related context.

5. *Recording of operation time* - The system must be able to record the operation time of each assembly step automatically.

6. *Attaching of multimedia training material* - The system allows the user to attach multimedia training material, including video, pictures, and description text created with external tools.

7. *Presenting of multimedia training material* - The system allows the user to consume the multimedia training material associated with each assembly step.

8. *Comparing of cognitive skill* - The system must be able to use the recording of the performance of each subject and perform the cognitive skill comparison. It compares the sequence and similarity of each performed step without considering operation time and motion trajectory.

9. *Comparing of motor skills* - The system must be able to use the recording of the performance of each subject and perform the motor skill comparison. It compares the similarity of each performed step with the considering operation time and motion trajectory.

10. *Providing extrinsic feedback of knowledge outcomes* - The system allows the user to consume the extrinsic feedback of knowledge performance from the comparison of it with the expert's template. The feedback is in a form that is easily interpretable, including scores, grades, pass/failed evaluations, and descriptive recommendations.

3.5 Non-Functional Requirement

The non-functional requirement describes the aspect of the system that is not directly related to its functional behavior. This is usually how the system looks and feels to the user as a system's quality. It is also known as a quality requirement. The non-functional requirements(NFR) of EXAMINER are described in the following subsections below.

1. Performance

   - The system maintained the frame rate at 30 frames per second in any video and graphic display of any media presented in the graphical user interface.
   - From a user perspective, there should be no noticeable delay between reality and the live video feed display.

2. Supportability

   - The module must provide an interface for supporting different sources of data streams from different sensor vendors.
   - The system must be able to run on a dedicated personal computer or an edge computing device. If required, the heavy computational task, including human pose estimation, can be transmitted to a dedicated system.

3.6 Summary

The requirements engineering chapter describes the overall purpose of EXAMINER, as explained by the visionary scenario. The use cases are based on four primary scenarios: record assembly tasks, edit assembly tasks, study task material, and consume performance feedback. The scenarios are then generalized into use cases. From these use cases, functional and non-functional requirements are gained. The proposed system is built on the assumption of these requirements. In the next chapter, the conceptual framework of the proposed system is formalized in an analysis model.

Chapter 4

Conceptual Framework of Proposed System

From the requirements elicited in the previous section, Various tools aid the transformation of the requirement into the system in an organized and structured way. These tools also help to communicate the structure of the system to stakeholders. The dissertation continues to employ the concept of system analysis and design(SAD) by documenting it using Unified Modeling Language(UML) [8]. UML is a modeling language widely used for designing, communicating, and documenting software system design. This dissertation uses the conceptual data model, activity diagram, and component decomposition diagram to visualize the structure of EXAMINER and to demonstrate the process. In addition, off-the-shelf components and a selection of programming languages are also being elaborated on. The section begins with an introduction to the framework and its sketch.

4.0.1 Introduction to EXAMINER framework

EXAMINER simulates an MA training session conducted at an MA station. Generally, there are two types of MA stations: single MA stations and inline MA stations. At a single MA station, an MA operator independently completes a multi-step MA operation with varying operation lengths for each assembly step. The assembly can be performed either standing or sitting, depending on the required range of motor trajectory and hand motion precision. Compared to an inline MA station, the single station is used in all enterprises, from large to small. In contrast, an inline MA station is commonly used in large enterprises to produce large quantities of products.

EXAMINER is designed to operate as a portable I-VTS in the form of station add-ons. It primarily comprises a context-sensing device, a multimedia presentation device, and a computing resource. EXAMINER continuously senses the MA station and deduces MA context. It provides enhanced terminal extrinsic feedback to the trainee by objectively comparing their MA skills performance to the recorded template. The feedback assists the trainee in achieving a knowledge performance and facilitates the trainee's incorporation of the received feedback with personal intuition concerning the knowledge of a result.

Introducing a comprehensive I-VTS framework offers several key benefits:

1. **Component Identification:** The framework helps interested parties identify the system's underlying components, providing a clear understanding of how different parts interact and contribute to the overall functionality.

2. **Standardization:** It sets a standard for future work under the same paradigm, ensuring consistency and compatibility across different implementations. The standard is crucial for advancing research and development in this area.

3. **Structured Approach:** By providing a structured approach, the framework facilitates the systematic design, implementation, and evaluation of similar systems, promoting best practices and enhancing the quality of outcomes.

4. **Guidance for Development:** The detailed description of each component and their interconnections guides developers and researchers, streamlining the process of creating and improving virtual training systems.

This introduction to the EXAMINER framework outlines its configuration and operation at an MA station, designed to support both standing and sitting assembly tasks. The framework's adaptability across different enterprise sizes highlights its utility in enhancing MA training effectiveness. By detailing the setup and the technological components involved, this section sets the stage for further details of EXAMINER's design in subsequent sections.

4.1 Conceptual Data Model

Conceptual data models or domain models visualize a zoom out of the overall system showing major parts [4]. It directly transforms the requirement into a decoupling of underlying things important to fulfilling the requirement. This thesis visualized the data model of EXAMINER in the figure 4.1. A manual assembly task or training package consists of a strictly sequential assembly step (ST) list. Each step is composed of another sequential list of primitive steps (PR) and the step instruction manual. The step instruction manual includes multi-media material that assists the trainee in training each step. The primitive step is the concrete activity with the material, tools, and workpiece, defined as context at location (LO). The activity is a limited set of motion trajectories, and limb(s) (LI) perform the motion. The context is defined as the transition type between the present (P) not to present (NP) and vice-versa of the interested region. The training package is ready to be used by filling in all the data in all parts.

EXAMINER introduces an MA data structure, enabling the persistent storage, transfer, comparison, and utilization of digitized MA operations across various components within EXAMINER system. This ensures that the system is compatible and consistent across different implementations. Furthermore, the proposed MA data structure is a crucial aspect of the EXMINER framework describing the data object and its flow within the system.

At its core, MA_α denotes a finite set, with members ordered by a natural number index, $\mathbb{N}$ and α serves as an index parameter of MA_α with $\alpha \in \mathbb{N}$. This set comprises a sequence of assembly steps ST_β, where β ranges from 1 to n. Each ST_β represents an ordered assembly step, from the first step when $\beta = 1$ to the final step when $\beta = n$. Each ST_β is a set comprising of finite ordered primitive steps PR_γ, with γ ranging from 1 to n. Formally, MA_α can be represent as

$$MA_\alpha = \{ST_1^{(\alpha)}, ST_2^{(\alpha)}, ST_3^{(\alpha)}, \ldots, ST_n^{(\alpha)}\} \tag{4.1}$$

In this formulation, MA_α is the set containing all of its assembly steps from ST_1 to ST_n, and α serves as an index to access different assembly step sets. Each ST_β represents an assembly step within the sequence. An operator must perform MA_α completely in strict ST_β order with expected quality and a consistent, desirable operation time. An expert defines the step order through a well-established guideline, an experiment, a personal intuition, and prior experience. The ST_β set is represented as:

$$ST_\beta^{(\alpha)} = \{PR_1^{(\beta)}, PR_2^{(\beta)}, PR_3^{(\beta)}, \ldots, PR_n^{(\beta)}\} \tag{4.2}$$

In this formulation, ST_β represents an ordered set of its primitive steps PR_γ within the ST_β. Here, PR_γ is an object defined as:

$$\begin{aligned} PR_\gamma^{(\beta)} = \{&LI^{(\gamma)}, ACT^{(\gamma)}, LO_{init}^{(\gamma)}, \\ &LO_{dest}^{(\gamma)}, TOPT^{(\gamma)}, TRAJ^{(\gamma)}\} \end{aligned} \tag{4.3}$$

The members of the primitive step object can be categorized as follows:

- Cognitive skill data
    - LI represents the limb that performs the primitive step PR_γ. It can be the left hand (LH), right hand (RH), or both hands (LRH) in cases where the assembly activity requires the use of both hands.
    - ACT represents the action associated with PR_γ, including reaching and retracting for picking and placing, assembly for fitting and aligning, and tool use for screwing.
    - LO_{init} and LO_{dest} are location identifications associated with the movement of the limb LI from an initial location LO_{init} to a destination location LO_{dest}.
- Motor skill data
    - $TOPT$ represents the measured operation time of the primitive step PR_γ.
    - $TRAJ$ represents the multi-dimensional trajectory of the limb LI in either $\mathbb{R}^2$ or $\mathbb{R}^3$ space, expressed as a time series.

Additionally, it should be noted that although the primitive steps PR_γ within ST_β appear sequential in the notation, an operator can execute PR_γ concurrently with $PR_{\gamma+1}$ and subsequent steps, subject to the availability of limbs. For instance, the operator may simultaneously pick an assembly part and place another part with each available hand. EXAMINER interprets this circumstance as distinct.

In summary, the primitive step object PR_γ comprises cognitive skill data, including information about the limb involved (LI), the action performed (ACT), and the initial and destination locations (LO_{init}, LO_{dest}), as well as motor skill data, including the operation time ($TOPT$) and the trajectory of the limb ($TRAJ$) as summarized in 4.2.

4.2 Dynamic Model

The behavior of the data model can be documented using UML activity diagrams. An *activity diagram* is a UML notation representing the behavior of a system concerning activities. The diagram is separated into two as there are two main actors in EXAMINER. The first diagram concerns expert as in figure 4.3, the latter concerning 4.4. Each diagram consists of two swim

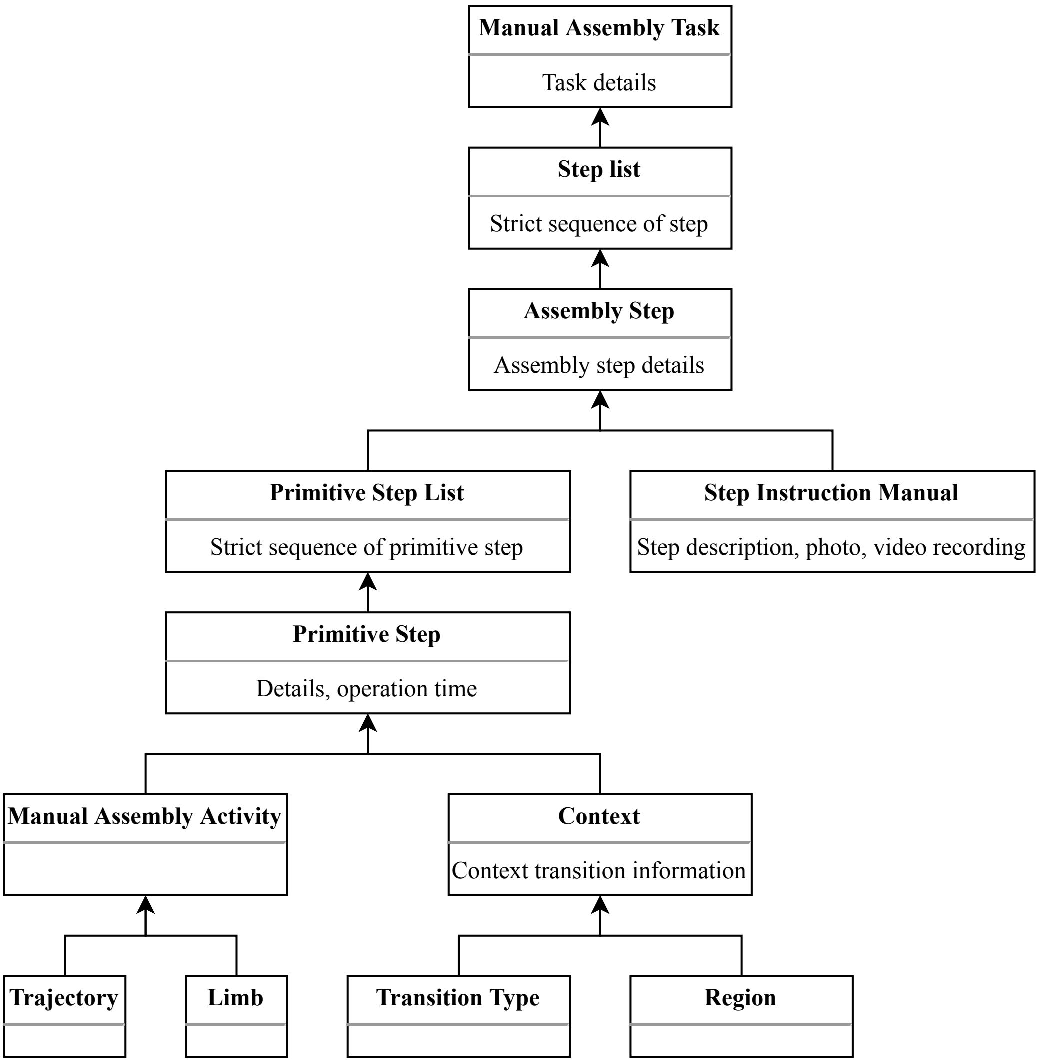


Figure 4.1: A conceptual data model of EXAMINER.

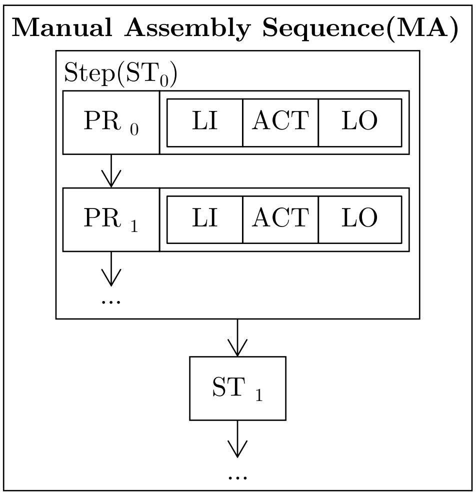


Figure 4.2: Proposed data structure of an MA task. Where activity is denoted as *ACT* and Region denoted as *LO*.

lanes, the actor and the EXAMINER system. It indicates the flow of activity from start to finish. EXAMINER consists of two main activities performed by different actors as follows,

1. Expert record and edit assembly task as in figure 4.3, the activity diagram depicts the process of recording and editing an assembly task performed by an expert using the EXAMINER system.

   (a) Expert Actions:

   - The expert begins by performing the assembly task.
   - Once the expert stops performing, they edit the assembly task.
   - If editing is required, the expert edits the assembly task before releasing the training material.

   (b) EXAMINER System Actions:

   - Concurrently, EXAMINER records the performance of the expert.
   - The system processes the recorded data through various modules:
     - **Human Pose Estimation:** Identifies and tracks the human body joints and limb positions.
     - **Activity Recognition:** Recognizes the activities performed during the assembly.

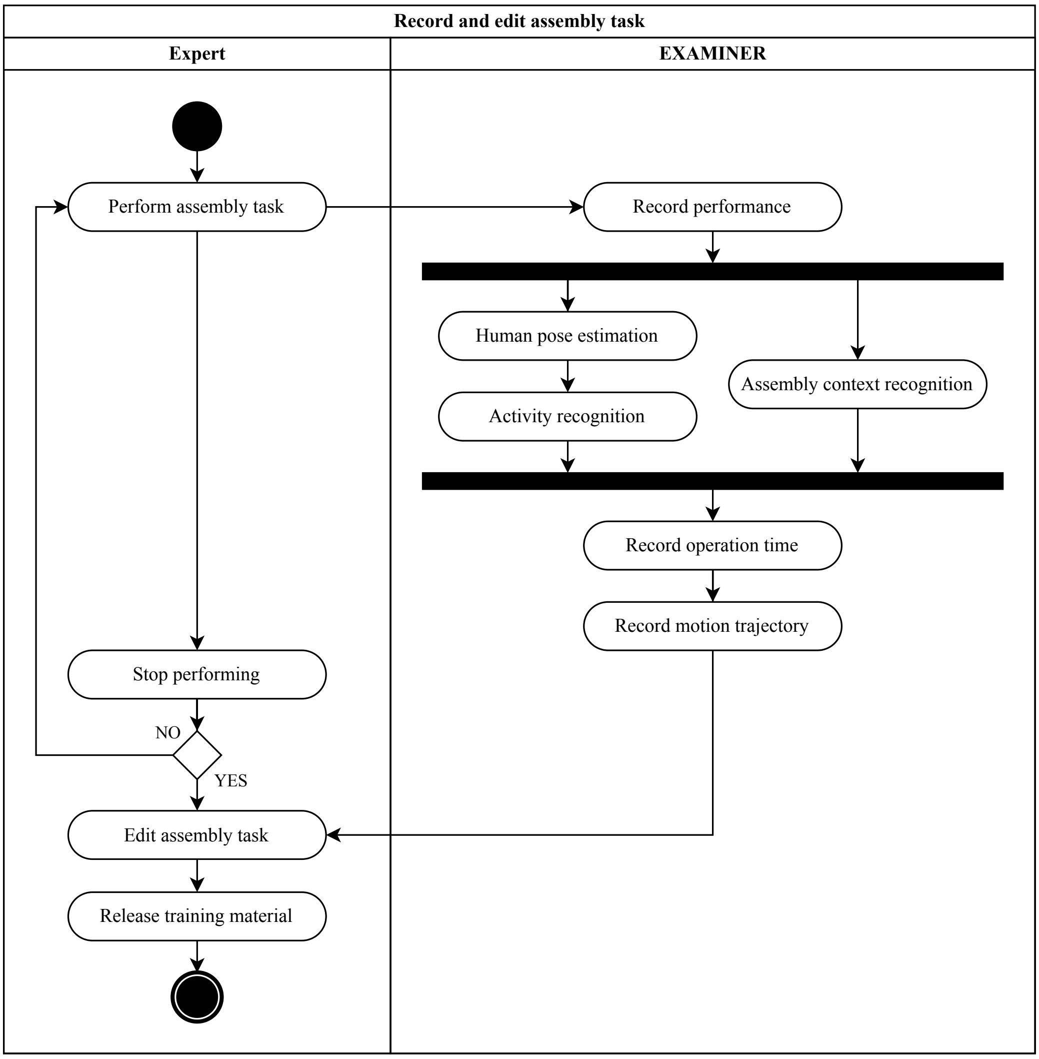


Figure 4.3: The activity regarding record and edit assembly task performed by an expert.

  - **Assembly Context Recognition:** Identifies the context of the assembly operations.
  - The system then records operation time and motion trajectory.

  This process ensures that the expert's performance is accurately captured and processed, allowing for the creation of comprehensive training materials. The recorded data is essential for providing detailed feedback and training guidance.

2. Trainee undergoes training as in figure 4.4. The activity diagram shows the training process for a trainee using the EXAMINER system.

   (a) Trainee's Activities:

   - Consume Training Material: The trainee starts by studying the provided training materials.
   - Ready to Perform Evaluation?: The trainee decides whether they are ready for evaluation.
   - Perform Assembly Task: The trainee proceeds to perform the assembly task.
   - Stop Performing: Upon completion, the trainee determines if they are done with the task.
   - Consume Extrinsic Augmented Feedback: The trainee receives feedback based on their performance.
   - Pass the Training?: The trainee checks if they have passed the training.

   (b) EXAMINER's Activities:

   - Present Training Material: EXAMINER presents the training materials to the trainee.
   - Record Performance: Records the trainee's performance during the assembly task.
   - Human Pose Estimation and Activity Recognition: Analyzes the trainee's body movements and recognizes the activities.
   - Assembly Context Recognition: Monitors the trainee's assembly task context.
   - Record Operation Time and Motion Trajectory: Logs the operation time and movement path.
   - Load Digitized Expert's Template: Loads the expert's performance template for comparison.

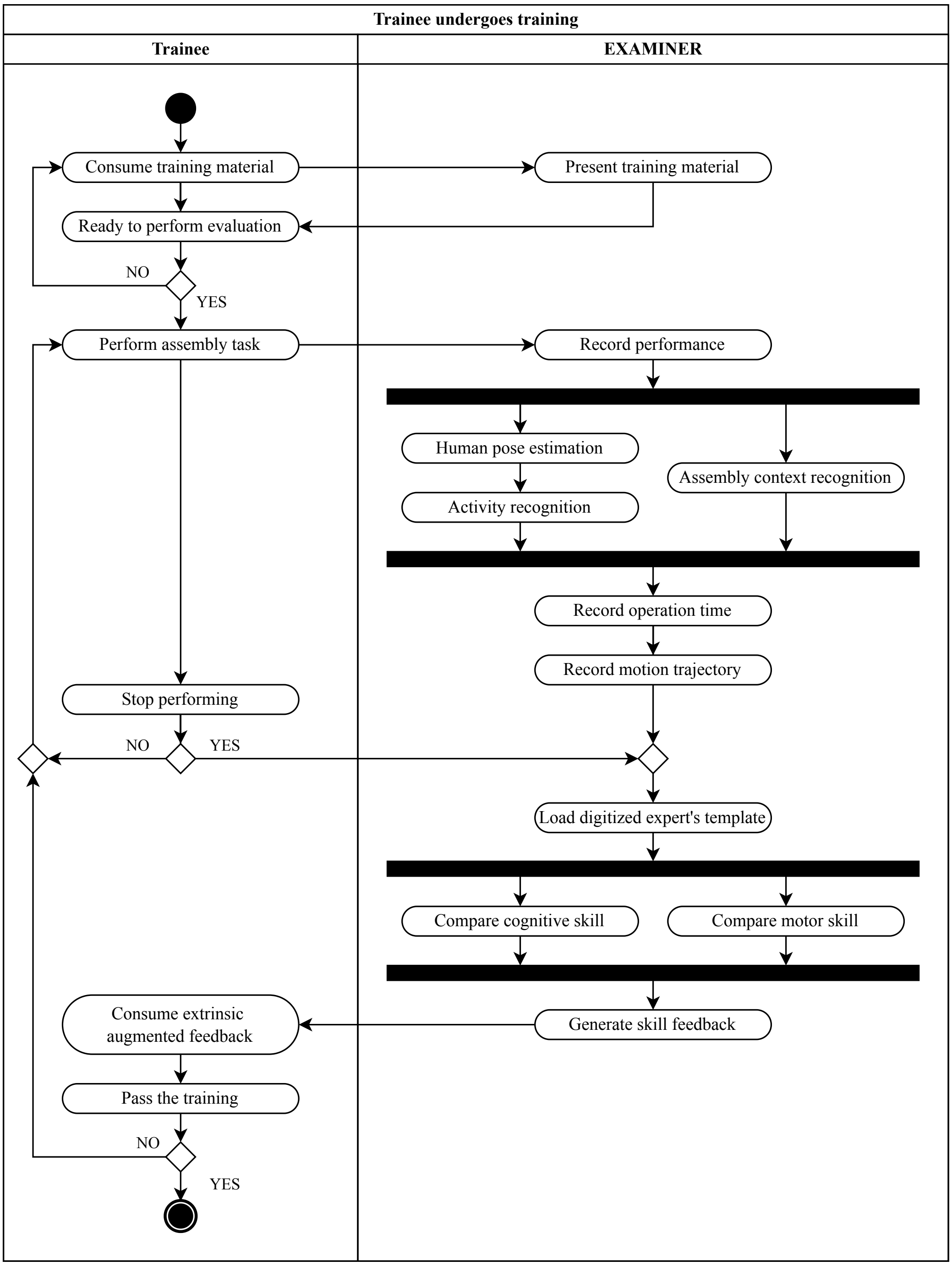


Figure 4.4: The activity regarding a trainee undergoing training.

- Compare Cognitive and Motor Skills: Evaluate the trainee's skills against the expert's template.
- Generate Skill Feedback: Produces feedback to guide the trainee's improvement.

These diagrams collectively illustrate the comprehensive training and evaluation process using the EXAMINER system, highlighting both the expert's role in preparing training materials and the trainee's experience during the training process.

4.3 System design and architecture

The previously discussed analysis model is transformed into a system design in the system design section. This model clearly describes design goals, subsystem decomposition, and system-building strategies.

4.3.1 Design Goals

The system design is driven by the following overall design goals, including,

- **Usability**
  It should be easy and intuitive for a user to navigate the graphical user interface to accomplish the task.

- **Non-Obtrusive**
  The system must be non-obstructive to the user while digitizing the skill. This means that the user's hand and arm can move freely in the camera's field of view. It requires no intermediate application controls during the recording.

- **Response Time**
  The system must appear responsive to user input. The computation time process must be presented and visualized as a progress bar or, if possible, a countdown and elapsed timer.

4.3.2 Subsystem Decomposition

In this section, the dissertation elaborates on mapping the system's components to different subsystems, which will later be mapped to actual hardware devices.

Addressing the architecture of the EXAMINER framework is crucial as it defines the interconnection of major components. This architecture not only addresses the functionality of each component but also ensures their effective integration. EXAMINER system is separated into four distinctive features that form an interconnection of major components resulting in a system architecture as depicted in Figure 4.5, including:

- Skill digitization,
- Skill comparison,
- Feedback providing, and
- Multimedia training material

Skill digitization

Skill digitization is a component responsible for converting input from a non-invasive commodity data acquisition sensor, such as a color video camera, into a piece of objectively comparable information on MA cognitive and motor skills. The skill digitization component comprises several sub-components that work collaboratively to achieve the common goal of digitizing the operator's MA skills.

The video obtained from a color camera is split into two data streams. The first data stream is routed to a sub-component that performs motion capture of human performance. The motion capture sub-component uses the data stream to detect joint locations in multidimensional Cartesian coordinates $\mathbb{R}^n$ where $n \in \{2, 3\}$. The output of the sub-component, in the form of a time series of the joint's location, is then used as an input for the activity recognition sub-component. The activity recognition sub-component recognizes MA activities, including reaching, retracting, tool use, and assembly. Concurrently, the second data stream goes to a context capture sub-component, which continuously recognizes the state of an MA object associated with the activity. The object's state in the assembly station is either present or not present. The output from the context capture sub-component indicates the object's state and the time at which the object's state changes. The output from the context capture sub-component is then split into two streams. The first stream is for the step recognition sub-component. The sub-component incorporates the previously recognized MA activities and the state of an MA object from the activity recognition and context capture sub-components, creating an MA step as an output. The second stream is for a

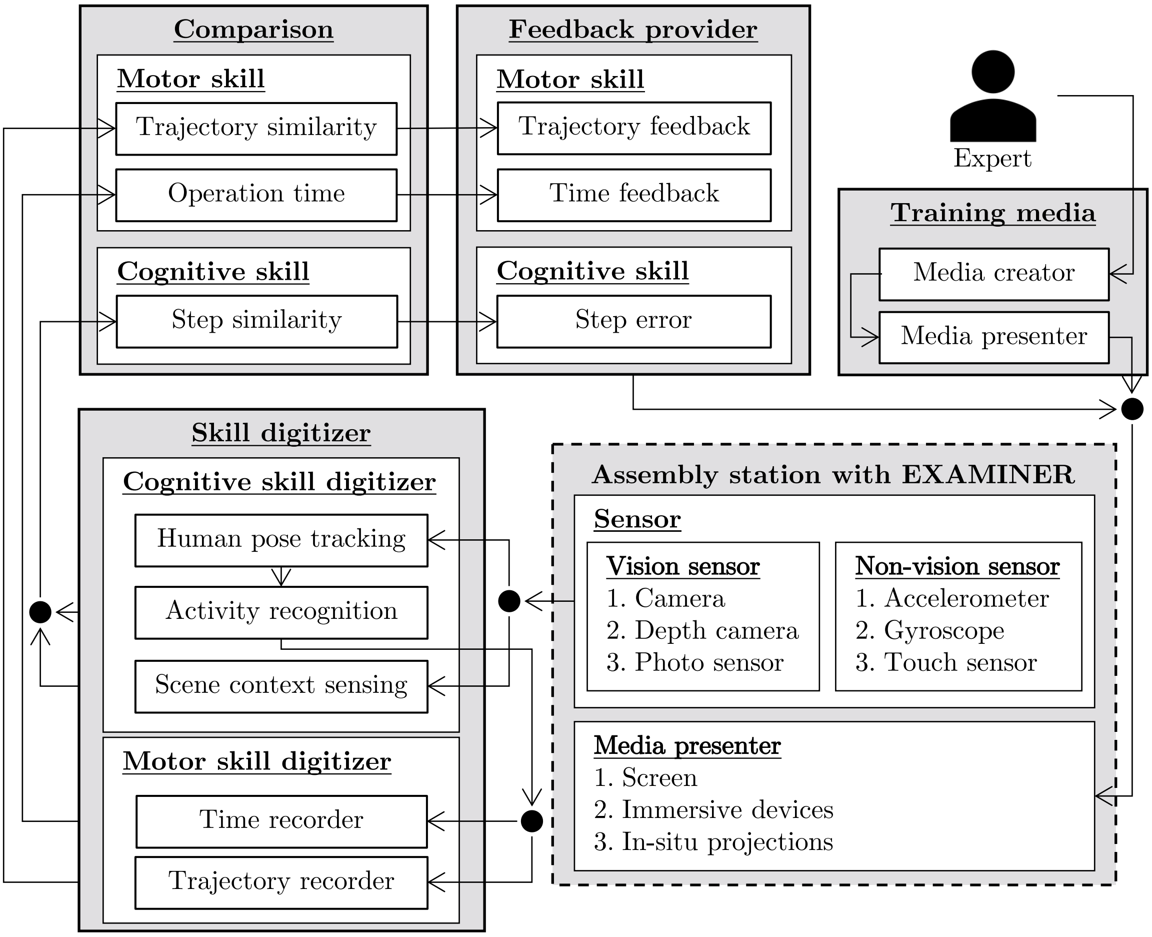


Figure 4.5: Proposed conceptual framework of EXAMINER. It consists of four major components and a physical system as an assembly station.

recording of the operation time sub-component where the name is suggested; it records the operation time at each assembly step.

Finally, the output of the step recognition and recording of the operation time sub-component is the operator's sequences of MA step and operation time, respectively. The output can now be used to compare the skill with the specified template using a skill comparison component or as the comparison template itself.

Skill comparison

The skill comparison component compares the trainee's previously digitized MA skills to the expert's digitized template. The comparison seeks to highlight operators' differences and similarities in MA skills. MA skills comprise MA cognitive skills and MA motor skills. EXAMINER defines MA cognitive skill as an operator's ability to perform strictly sequential step-by-step MA operation sequence with the appropriate action on an assembly part, including:

- Picking up or selecting the correct part,
- Incorporating parts with a correct location and orientation and
- Using a tool to assist the incorporation if required.

MA motor skills refer to the operator's ability to complete the assembly in a consistent, desirable time with an appropriate motion.

EXAMINER introduces sub-components for the skill comparison component, which compares digitized skills by identifying differences and similarities. A difference in operation sequence sub-components receives input from the step recognition sub-component. The sub-component compares and outputs MA operation sequence differences from the desired template. A difference in operation time and motion similarity are two measurements used to compare motor skills. First, a difference in operation time sub-component compares the difference in operation time from a specified target using an output from the sub-component. In concurrent, a motion similarity sub-component directly uses the output of the motion capture sub-component as a time series of the joint's location to measure similarity with the template.

Following the comparisons, the skill comparison component's outputs are passed to the feedback-providing component in the form of differences in MA step sequence, operation time of each step, and assembly motion similarity.

Feedback providing

The skill comparison component, as previously stated, produces three types of outputs: differences in operation sequence, operation time, and motion similarity. In general, these outputs can be used directly as augmented feedback to both the trainee and the expert, as has been done in previous studies. Most of the time, the expert interprets the system's augmented feedback to the trainee. On the other hand, EXAMINER introduces a feedback-providing component. It converts outputs of the skill comparison component into a suitable format specified by the expert for direct consumption by the trainee.

The feedback-providing component will use a matching rule to match the input and then provide augmented feedback. The expert must provide a matching rule and the corresponding augmented feedback output for each matched rule for each input, including differences in operation sequence, operation time, and motion similarity. EXAMINER proposed two types of augmented feedback: a grading and a semantic description of an error. First, the grading feedback intends to report differences in operation time and motion similarity. The grade can be given in letters, a numerical range, or a percentage. A matching rule for grading, also known as a grading scheme, is based on an expert's intuition and is flexible depending on the nature of the MA operation. For example, the operation that requires an exact match in operation time and motion may choose a binary pass or fail grade. Second, the semantic description of an error seeks to report differences in operation sequence in the form of a brief text. For example, a skip or missing MA operation step in the MA operation step sequence will be reported as a short text indicating the type of error and at which step.

Finally, the trainee employs augmented feedback provided by the feedback-providing component and self-realized intrinsic feedback to improve outcomes during the next training iteration.

Multimedia training material

Multimedia training material is a component that displays MA multimedia training material. The component enables an expert to attach additional traditional and immersive mediums and information that the system's skill digitization component cannot automatically digitize. A conventional medium may include an image, video, animation, and text describing the MA operation sequence. On the other hand, an immersive medium can range from semi-immersive, such as AR, to fully immersive, such as VR. Consequently, the expert improves the comprehensiveness of the digitized MA operation and generates training materials.

Training materials and an expert template of the digitalized MA are then packaged and sent to the trainee. The trainee must be capable of consuming and navigating the content using the appropriate media display and input technologies. For example, the conventional medium is displayed on a computer monitor. The interaction may be conducted using a touch-sensing interface or a hand-held pointing device, such as a touchscreen monitor and computer mouse. The immersive medium may necessitate an AR-OHMD, video see-through HMD (VST-HMD), and VR. Each device offers a distinct method of interaction with the immersive medium. The trainee can consume the training material at their discretion without relying on face-to-face training with an expert. The trainee may attempt the MA operation by referring to the provided materials. When the trainee is prepared, EXAMINER assesses their knowledge performance. After addressing the system architecture, the off-the-shelf components are addressed.

4.3.3 Off-the-Shelf Components

The off-the-shelf component is an external component that was carefully selected and used in the system as it was initially implemented. It supports services provided by the components described in subsystem decomposition. The digitization component uses vision-based two-dimensional human pose estimation called AlphaPose to aid the digitization task. This section first introduces the concept of human pose estimation. It later provides the rationale for the selection of the off-the-shelf components.

Human Pose Estimation

The input of the two-dimensional (2D) HPE model is a frame(F) of video(V). Given τ is a temporal space, a video V is an n-tuple $(F_1, F_2, \ldots, F_\tau)$ where F_τ represent an image frame F at τ. Each F is a $x \times y$ matrix with $|x \cdot y|$ depends on the camera's resolution representing $\mathbf{R}^2$ of image pixel $p_{x,y}$ object encoded by an RGB video camera. The output of 2-D HPE is the probability of joints for each human represented in F at $p_{x,y}$. The estimated number of joints solely depends on the dataset used for training. The MSCOCO 2017 dataset contains approximately 64,000 images with at least one human with joint annotations as $J_{k_{x,y}}$ [42]. Where k is the annotation class as listed in table 4.1. By providing F to the 2D-HPE model, the model estimates the location of all possible $J_{k_{x,y}}$ as visualized in figure 4.6.

2D-HPE is a challenging and highly competitive research topic as it has practical contributions in other areas. Currently, there are various ready-to-

Head	Nose	
	Left Eye	Right Eye
	Left Ear	Right Ear
Upper Body	Left Shoulder	Right Shoulder
	Left Elbow	Right Elbow
	Left Wrist	Right Wrist
Lower Body	Left Hip	Right Hip
	Left Knee	Right Knee
	Left Ankle	Right Ankle

Table 4.1: The table listed all J_k under MSCOCO 2017 dataset.

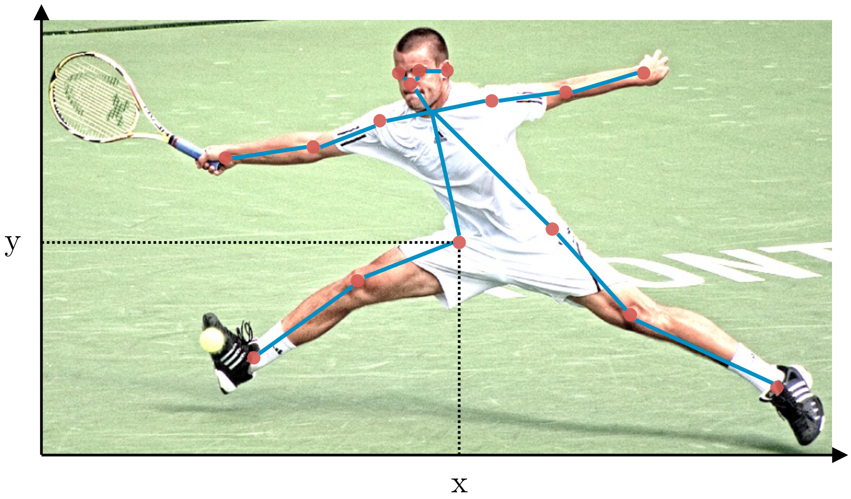


Figure 4.6: Visualization of HPE output under MSCOCO 2017 dataset.

Method	**Head**	**Shoulder**	**Elbow**	**Wrist**	**Hip**	**Knee**	**Ankle**	μ
OpenPose	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6
Stacked Hourglass	92.1	89.3	78.9	69.8	76.2	71.6	64.7	77.5
AlphaPose	91.3	**90.5**	**84**	**76.4**	**80.3**	**79.9**	**72.4**	**82.1**

Table 4.2: AlphaPose yields the best average precision (AP) on the target joints including shoulder, elbow, and wrist.

be-use models, including OpenPose [62], [63], [81], [90], Stacked Hourglass [54], and AlphaPose[65], [85], [88]. For simplicity, the EXAMINER employs one of the 2D-HPE models that yield a sufficiently good result and is ready to be used out of the box, called AlphaPose. The performance comparison is listed on the table 4.2. The model selection is solely based on the report performance on MS COCO. It works in a multi-phase manner. The significant phases consist of estimating the human bounding box in F and estimating each key point, resulting in joints' probability heatmaps for each bounding box. The bounding box estimation uses a spatial transformer network where they can recognize the object, in this case, a human. Then, it performs the image's geometric transformation, resulting in the accurate boundary for a single-person pose estimator(SPPE). After obtaining the boundary, the model uses SPPE to estimate the $J_{k_{x,y}}$ on each boundary.

Stacked Hourglass Networks is one of the common network architectures selected for providing the probability of an object's location in F [54]. It is a form of encoder and decoder artificial neural network; the encoder extracts the input into a downsampled feature matrix. However, it lacks spatial information, hence requiring a decoder to preserve it. A decoder upsamples the feature matrix using the nearest neighbor technique. It performs element-wise addition by using the information transported from early layers. The output is a probability heatmap highlighting the joints' location in F, which later transforms into exact coordinates by selecting coordinates with the highest probabilities of each presented joint. Formally, by providing the input V to the 2-D HPE, it provides the output as joint location presented in F as

$$\text{2D-HPE}(V) = (\{J_{k_{x,y}}, \ldots\}_{\tau_0}, \ldots) \tag{4.4}$$

The skeletal-based two-dimensional human pose estimation employed by EXAMINER provides a practical and efficient method for capturing human joint positions using standard video equipment. This technology supports the framework's ability to analyze and digitize human motion without invasive tracking devices. This approach aligns well with the framework's objectives by focusing on simplicity, cost-effectiveness, and sufficient accuracy for coarse movement differentiation, setting a strong foundation for the subsequent cognitive skill digitization process. These estimated joint locations will be further utilized in the activity recognition sub-component, enhancing the framework's capability to translate physical movements into manual assembly activity data.

4.3.4 Programming Language

EXAMINER software was implemented using the Python3 programming language. Python was designed to be interpretable on the modern operating system(OS), including Microsoft's Windows, Apple's macOS, Linux, and Unix. Academics and enterprises recommend Python as it allows rapid prototyping while sacrificing some performance. In addition, there is an abundantly ready-to-use code library and documentation, reducing development efforts. The selection of OS also requires the implementer to consider the OS compatibility based on the external library as they might couple with some specific binary that is not compiled on the target OS.

4.4 Summary

The chapter on the proposed system's conceptual framework employs system analysis and design to transfer the requirement into a framework that can act as a communication medium between parties. The dissertation employs the following tools to realize the conceptual framework of EXAMINER, including,

- Conceptual data model to represent the structure of MA,
- Dynamic model to represent the flow of activity in EXAMINER,
- Subsystem decomposition to decompose the system into smaller components connecting with each other.

In addition, the chapter introduces the ready-to-use HPE library, the programming languages selected for implementation, and the proposal's evaluation later. The next chapter will emphasize the solution domain, starting with digitizing operator skills.

Chapter 5

Digitization of operator skills

The digitization of operator skills is centered on transforming unstructured camera sensor data into objectively assessable MA data. First, the proposed EXAMINER MA data structure will be defined, and then the MA operation step (MA step) and its motion time trajectory will be digitized. This chapter is divided into three sections listed below:

1. **Cognitive skill digitization** is the section that provides implementation details for the cognitive skill digitization group of sub-components, including MA step recognition and MA context recognition. The section also introduces a non-invasive motion capture technology.

2. **Motor skill digitization** is the section that provides implementation details for motor skill digitization. Furthermore, it addresses the rationale for decoupling motor skill digitization from step digitization.

3. **Evaluation** is the section that provides an evaluation of digitization components.

5.1 Cognitive Skill Digitization

Digitizing the ST_n is recognizing PR(s) to form ST and later MA. EXAMINER employs the hybrid recognizer, consisting of ACT and context recognition. As suggested in the related work section, there are various ways to acquire information. EXAMINER employs one of the least intrusive user methods, with commodity hardware, for ACT recognition. Here it applies an off-the-shelf component to perform deep learning in computer visions on color video from a commodity camera to estimate the location of joints in Cartesian plane $\mathbf{R}^2$. EXAMINER uses it as a data transformation step before applying the ACT recognizer algorithms later. The method is commonly known as skeletal-based human pose estimation(HPE). The dissertation provided the theory and rationale behind selecting these components under the subsection 4.3.3. This section continues with using the output of 2D-HPE(V) from 4.4.

5.1.1 Activity Recognition

Previously, an input V was processed frame by frame using HPE, resulting in the location of joints. The output of 2D-HPE is in n-tuple format forming the time series location of the joint in $\mathbf{R^2}$. As such, action recognition tasks from 2D-HPE output may apply any supervised machine learning algorithm. EXAMINER aims to recognize MA-related ACT. The definition of an ACT must be precisely defined to reduce possible human error during ACT labeling. Based on the education robot assembly case study, the ACT consists of reach, retract, assembly, and tool use.

Other assembly activities may require additional ACT classes to extend to previously introduced classes. The pipeline of a supervised activity recognition model creation is as follows,

- Data labeling,
- Feature selection,
- Feature pre-processing,
- Feature scaling,
- Feature engineering
- Feature segmentation,
- Model selection and training, and
- Model inference

The elaboration of each pipeline step is as follows.

Data Labeling

Due to the unavailability of a publicly accessible manual assembly (MA) dataset suitable for creating an MA activity recognition model, data labeling must be performed manually. Data labeling involves the annotation process to specify the precise timestamp range T, from τ_{start} to τ_{end}, for a given PR_{γ}, where $start < end$. The labeler watches the video and associates the PR_n to the frame F_{τ} from initialization to termination. Any activity that does not adhere to the predefined definitions is labeled as a null class, denoted as ϵ.

The number and definition of activities vary according to the manual assembly operation. However, the following activities are identified as common occurrences in manual assembly by EXAMINER:

1. **Reach**: The use of a limb (either left hand (LH), right hand (RH), or both hands (LRH)) originally in the initial location (LO_{init}) to reach the destination location (LO_{dest}).
2. **Retract**: The pulling back of the limb from the destination location (LO_{dest}) to the initial location (LO_{init}).
3. **Assembly**: The action of incorporating parts using both hands (LRH).
4. **Tool Use**: The action of incorporating parts using both hands (LRH) with the assistance of a hand tool.

After completing data labeling, the pipeline proceeds to feature preprocessing, focusing on treating noisy data.

Feature selection

The COCO dataset provides an estimation of 17 body joints from the head to the lower body as listed in table 4.1. Activity recognition only considers upper body joints for the hand-only MA task at standing or sitting assembly station. As of this, six of 17 joints are in consideration. Here, the visualization in figure 5.1 shows the obstruction of the lower body part. However, other configurations requiring the operator to walk, use feet, or operate the floor-standing control panel will require a camera setup at different locations to cover a larger scene area.

Feature pre-processing

A skeletal-2D Human Pose Estimation (HPE) system has the drawback of processing each input frame F independently, which can result in jittery outputs. Jitter occurs when the approximated joint JT at coordinates (x, y) is not consistently precise, giving the impression that the joint's motion is shaking or oscillating. In the worst-case scenario, a stationary person may appear to be shaking. Using the inferred joint positions to train the model may introduce false-positive data points. Commonly, post-processing procedures are employed to eliminate skeletal-2D HPE jitters. Smoothing filters can be utilized both online and offline. An online or real-time filter does not

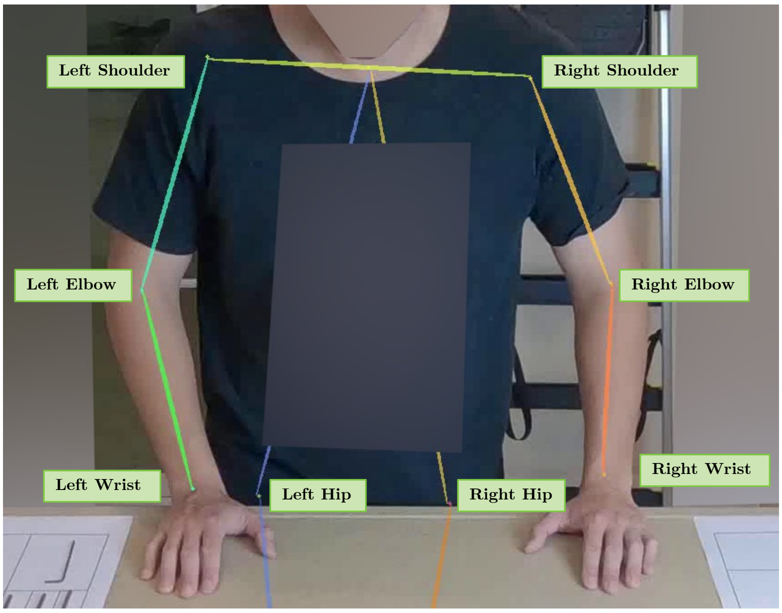


Figure 5.1: Visualization of output from a 2D-HPE where only the upper body is visible.

know in advance what value it will process, and a combination of high-pass and low-pass filters is commonly used because it reduces both high and low oscillations. On the other hand, an offline filter uses all values as input, allowing the filter algorithms to view all values before smoothing them.

The Savitzky-Golay filter is utilized by EXAMINER to reduce jitters. Figure 5.2 illustrates that it is one of the offline filters capable of smoothing down noisy inferencing results. The subsequence right wrist trajectory is shown in $\mathbb{R}^2$ over 30 frames, representing the recording of the performer in the assembly scene reaching for the product with their right hand, as digitized by HPE. In contrast, the original HPE appears to oscillate, necessitating the application of a smoothing filter. The Savitzky-Golay filter accomplishes polynomial least-squares fitting by approximating data points under each sliding filter and taking the estimated data point at the center. Given filter length FIL and filter order FIO, the selection of $FIO < FIL$ results in smoothing; hence, $FIO + 1 = FIL$ represents a polynomial interpolation of the data in FIL. All smoothing approaches result in some data loss, especially when the data includes rapid activity changes.

After removing noise and potentially erroneous data, the pipeline proceeds to feature scaling, focusing on data transformation to reduce bias from different measurements.

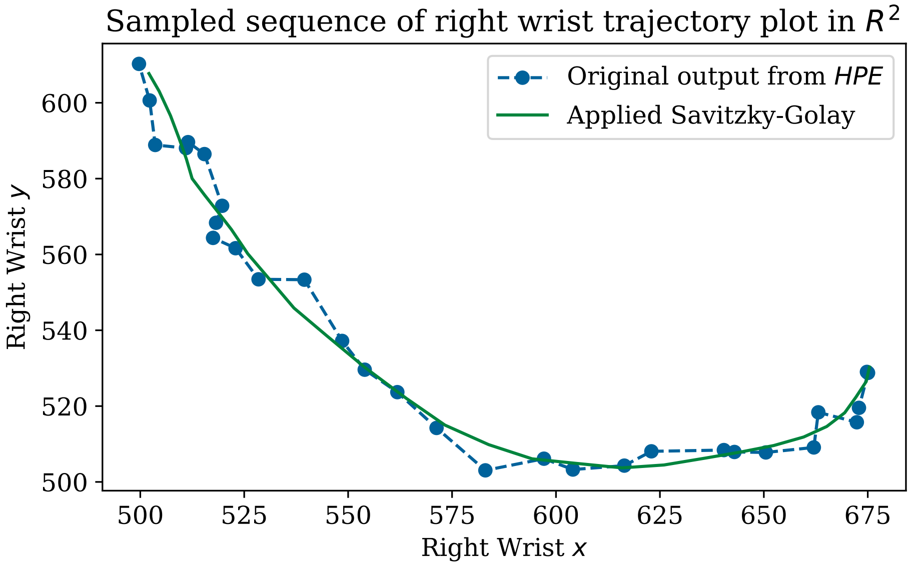


Figure 5.2: The subsequence right wrist trajectory with length 30 frames plotted in $\mathbf{R^2}$. It shows the performer's recording using the right hand, reaching for the object in the assembly scene as digitized by HPE. However, the original HPE appears jitter, requiring a smoothing filter to mitigate.

Feature scaling

Different measurements among features may lead to bias in training a classifier; hence, it is recommended to scale features. Various methods are available for scaling, including normalization and standardization. Normalization, such as the min-max scaling, can be performed as follows.

$$J'_{k_i} = \frac{J_{k_i} - J_{k_{min}}}{J_{k_{max}} - J_{k_{min}}} \tag{5.1}$$

Here, k denotes the joint, and i denotes the channel, either x or y. The min-max scaling requires knowing the $J_{k_{max}}$ and $J_{k_{min}}$ in advance; otherwise, the scaling value will not stay under $J_k \in [0, 1]$. Another scaling method is to standardize or center the features. One of the popular methods is to make each of the features have a zero mean by performing Z-score normalization as follows.

$$J'_{k_i} = \frac{J_{k_i} - \overline{J_k}}{\sigma_{J_k}} \tag{5.2}$$

The selection of either normalization or scaling solely depends on the classification algorithms. Even though most deep learning architecture does not require feature scaling, doing so will help the model to converge faster, taking less time to train them [98]. The pipeline proceeds to feature engineering, focusing on data transformation to generate a feature vector.

Feature engineering

Feature engineering is essential in preparing human pose estimation (HPE) data for deep learning models, which may struggle with raw coordinate input due to body scale and position variation. This variation is caused by differences in individual anthropometry and the various positions that subjects may take within a scene. The process converts the coordinates into a more consistent feature space to address these issues.

EXAMINER opted for three commonly used methods to transform the feature outlined in [107], including

- Normalized distances of each joint from the center of the pose account for positional variance on the scene. In this method, the original joint location is subtracted from the center of the pose and normalized to reduce variance from a different anthropometry. The normalization is typically based on the height of the pose, and the resulting distance can be directly divided.
- The angles of rotatable joints provide a feature independent of scene position and body rotation. This is especially useful when comparing the orientation of body parts across subjects. The rotatable joints generally include shoulders, elbows, hips, and knees.
- Normalized displacement of joints between frames, reflecting the movement while invariant to translation and scaling. The normalization is relative to the height of the pose, ensuring that the motion is measured consistently regardless of the subject's size or distance from the camera.

The feature vector resulting from these transformations includes 26 features for normalized distances, 26 for displacements, and four for joint angles, totaling 56 features. The feature vector excludes joints from the lower body part as they are not visible in the scene. The pipeline now proceeds to feature segmentation.

Feature segmentation

Feature segmentation is crucial in preparing sequential data for machine learning and deep learning models. In the context of EXAMINER, an overlapping sliding windows segmentation technique is employed to partition the

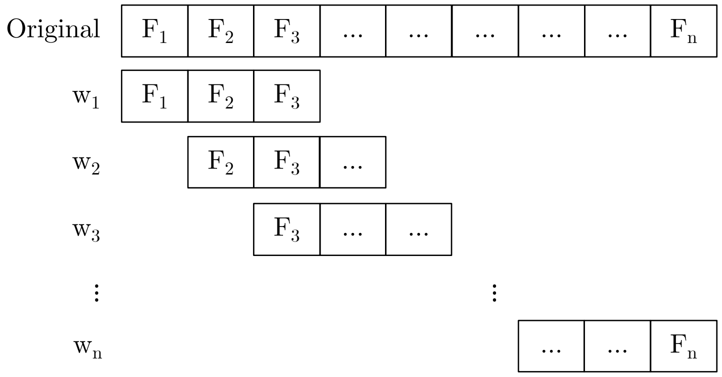


Figure 5.3: The original time series data is segmented by using the sliding windows of length three with the sliding of one resulting in multiple w_n.

input sequence of joints location of the whole video V into smaller windows (w_ι) with overlapping [100]. Figure 5.3 visualizes an overlapping sliding windows segmentation technique, given a sequential data sequence F_τ, where $\tau = 1, 2, 3, \ldots, T$, the overlapping sliding windows feature segmentation divides F_τ into a set of windows or segments $W = \{w_1, w_2, w_3, \ldots, w_n\}$, where $\iota \in \mathbb{N}$ represents the index of the segment staring from 1 to n. Each segment w_ι is of fixed size $|w_\iota|$; it is typically small enough to capture short activities without disregarding them. The overlap can be defined as a percentage of the window size or a specific number of data points to shift the window at each step. It ensures that each data point is included in multiple segments, capturing the temporal dependencies and preserving the continuity of the sequential data.

For example, given a sequential frame F sequence of a video V, the sliding windows segmentation technique would generate segments w_ι such that:

$$\begin{aligned}
w_1 &= F_{\tau:\tau+|w_1|} \\
w_2 &= F_{\tau+\delta:\tau+\delta+|w_2|} \\
w_3 &= F_{\tau+2\delta:\tau+2\delta+|w_3|} \\
&\vdots \\
w_n &= F_{\tau+(n-1)\delta:\tau+(n-1)\delta+|w_n|}
\end{aligned}$$

where δ represents the shift or overlap between consecutive windows; for demonstration purposes, EXAMINER assumes equal-size windows, hence $|w_1| = |w_2| = |w_3| = \ldots = |w_n|$. Sliding window feature segmentation preserves the temporal relationship between data points, allowing for capturing short-term activities or patterns within the sequential data. Additionally, this segmentation strategy generates multiple data samples, enhancing the model's training process.

Before proceeding with model selection and training, it is important to acknowledge that a single w_ι may contain multiple activity labels (ACT). This is because, in manual assembly tasks, operators may simultaneously perform multiple activities. As a result, the recognition problem needs to be treated as a multi-label classification, and it can be represented by the following equation

$$Y_\iota = (y_1, y_2, y_3, ..., y_n) \in \{\text{true}, \text{false}\}^{\mathbb{N}} \tag{5.3}$$

where y_κ represents the occurrence of an activity (ACT) either true or false, and κ serves as an identifier for each activity. Furthermore, since the sliding windows feature segmentation can include partial activity transitions from the previous ($w_{\iota-1}$) and next ($w_{\iota+1}$) segments, a majority voting technique is employed to determine the final label for w_ι. However, in the event of a tie in the voting process, the label for w_ι is a multi-label, indicating the presence of multiple activities within w_ι. Subsequently, after the feature segmentation step, the pipeline advances to model selection and training.

Model selection and training

Based on the given $HPE(V)$ input, various machine learning models are suitable for classifying ACT, including support vector machine, k-nearest neighbor, hidden Markov model, and random forest. Multi-layer perceptron as deep neural network architecture also receives attention from researchers. As the data is temporal, a variation of recurrent neural network(RNN) such as a Long short-term memory(LSTM) should yield a satisfactory result. LSTM is a recommended choice from several published research [55], [69]. Before providing further details on LSTM model implementation, tuning of hyperparameters, and model training, this dissertation provides some basics on deep neural networks.

Convolution neural network A convolution neural network(CNN) learns the convolution filters or weights to obtain the features map of the input. It has proven to be successful in learning the features of an input image. It can be applied for an image recognition task [26]. However, it cannot encode the location and orientation of the object [54]; some additional layers and measures must be implemented. A CNN's architecture generally consists of a convolution layer, pooling layer, and fully connected layer [79]. The explanation for each layer is as follows:

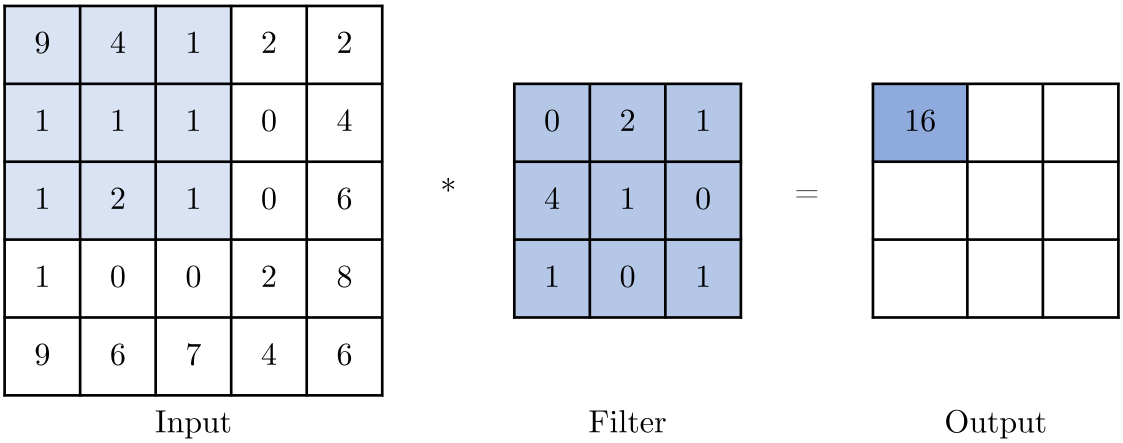


Figure 5.4: An example of convolution operation.

- *Convolution layer* - The convolution layer received input in the form of a tensor as (number of inputs) x (input height) x (input width) x (number of input channels). The input is then multiplied with $m \times n$ convolution filter or kernel, resulting in a feature map. A feature map is in the form of (number of inputs) x (feature map height) x (feature map width) x (feature map channels). It is also trivial to have multiple feature maps to learn different features. Figure 5.4 provides an illustrated example of a convolution filter. The $1 \times 5 \times 5 \times 1$ input matrix is being multiply with 3×3 convolution filter. Here the stride (or the shift of kernel on input) is one, and the padding is not in use. The final result is in the form of $1 \times 3 \times 3 \times 1$ feature map. The $Output[0][0]$ can be obtained by performing calculation $(9 \times 0) + (4 \times 2) + (1 \times 1) + (1 \times 4) + (1 \times 1) + (0 \times 1) + (1 \times 1) + (2 \times 0) + (1 \times 1)$.

- *Pooling layer* - The pooling layer reduces the dimension of an output or feature map from the convolution layer. By doing so, it reduces the location sensitivity of the features. There are three types of pooling layers, max pooling, average pooling, and global pooling [101].

  - **Max pooling:** calculates the maximum value in each patch of each feature map highlight the most present feature as in figure 5.5. Here, the patch is 2×2, and the stride is 2 or equal to the size of the patch. The result of the patch contains $9, 10, 11, 12$ is 12.

  - **Average pooling:** calculates the average value in each patch of each feature map as in figure 5.6. Here, the patch is 2×2, and

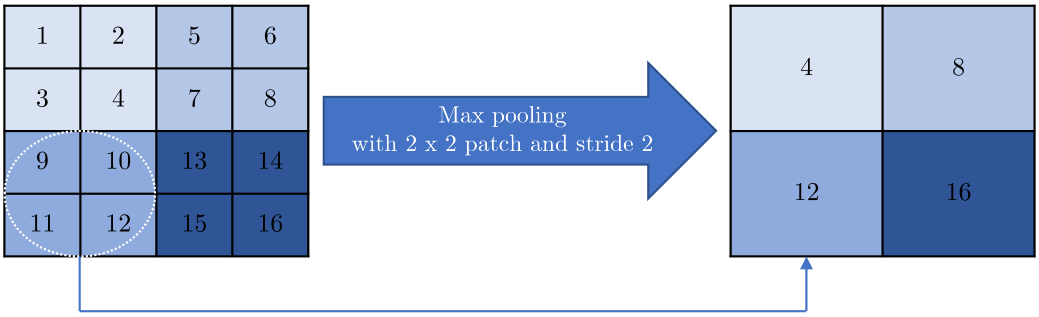


Figure 5.5: An example of a max pooling on a feature map.

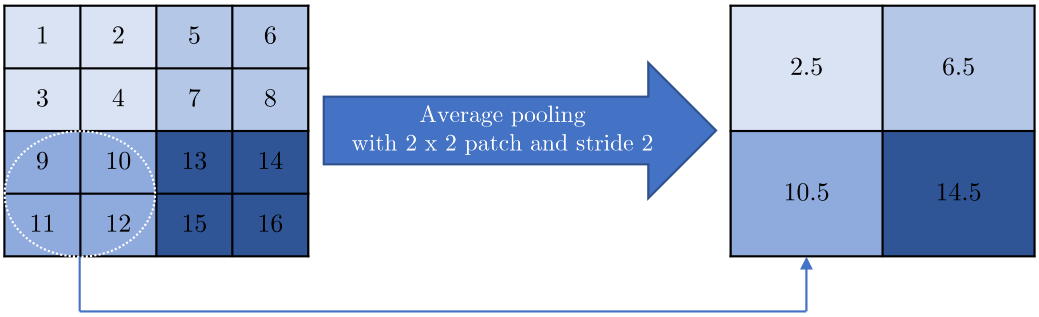


Figure 5.6: An example of an average pooling on a feature map.

the stride is 2 or equal to the size of the patch. The result of the patch containing $9, 10, 11, 12$ is 10.5 or the average among the values inside the patch.

- **Global pooling:** downsamples the entire feature map to a single value as in figure 5.7. Here is an example of a global average pooling on four feature maps. The patch is 4×4 or equal to the size of a feature map. It results in a vector of length equal to the number of feature maps.

- *Fully connected layer* - The fully connected layer is for providing the classification output. It is the same as a traditional multi-layer perceptron(MPL). Here, all the inputs from the previous layer are connected to every activation unit of the next layer forming the hidden layer. A fully connected layer may consist of more than a single hidden layer. The hidden layer then connects to the output layer providing probability for each class. Figure 5.8 is an example of a fully connected layer. The input is the flattening of the pooling layer. Input is then passed through the hidden layer, and it contains an activation function consisting of weight and bias. Lastly, each hidden layer output is connected to

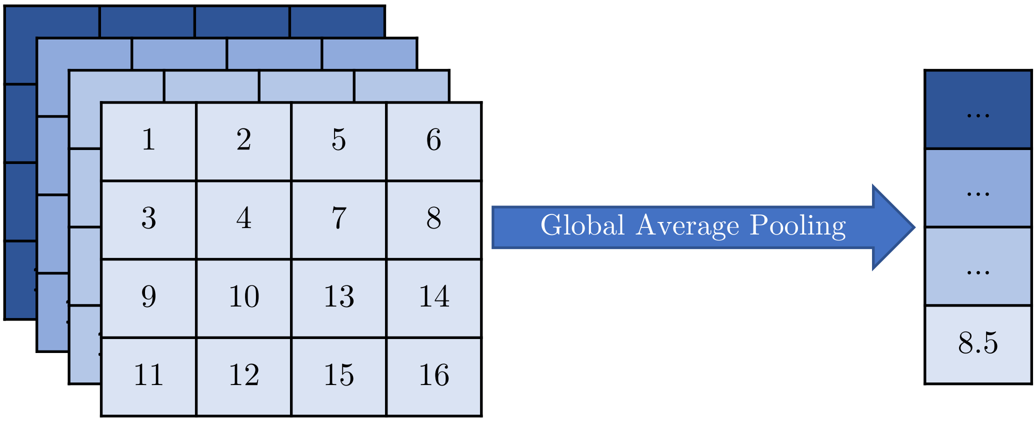


Figure 5.7: An example of a global average pooling on four feature maps.

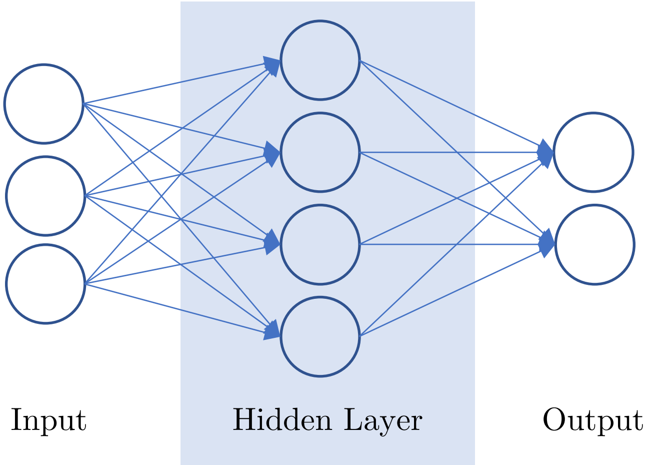


Figure 5.8: An example of a fully connected layer.

the output layer, which reports the probability for each class. The output layer consists of two nodes indicating the two possible classification classes, e.g., positive and negative.

- *A typical CNN architecture* - A simple CNN architecture can be realized by combining convolution, pooling, and fully connected layers. Figure 5.9 is an example of a CNN architecture consisting of a single convolution layer with pooling and a fully connected layer with a single hidden layer. The architecture received a single channel input of size 6×6. By applying 3×3 convolution with stride 1, the output of the convolution layer is in the size of 4×4. The output is then pooled by 2×2 patch with a stride of 2, resulting in 2×2 output. The output

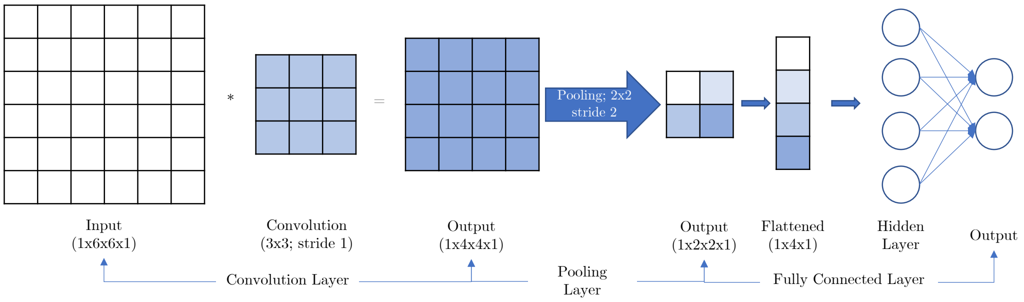


Figure 5.9: An example of a CNN architecture that consists of a single convolution layer with pooling and a fully connected layer.

then proceeds to the fully connected layer, which performs the final classification by providing class probabilities.

- *CNN architectures for human activity recognition* - Currently, the primary usage of CNN is mainly related to computer visions. However, it is also possible to encode the sequence of sensor reading as suitable tensor input to CNN architecture by sampling data streams into subsequences of equal size windows. The data from sensors may consist of multiple channels; for instance, the data from the accelerometer usually consists of x, y, and z axes. Here, the modeler may model the input into (number of equal size windows) × (length for each window) × (number of channels) × one or else. There are various CNN-based HAR proposals, and they can be categorized by the length of the input, types of warble sensors, prepossessing techniques, and CNN architecture. For instance, a typical CNN as in figure 5.9 with the input from an accelerometer length of 64 and 3 channels can recognize activity including jog, walk, walk up, walk down, sit, and stand [36]. The transformation and combination of input from various sensors such as gyroscope and accelerometer can also be applied for creating activity images and introducing additional hidden layers [58]. Various hyperparameters of CNN, including the number of hidden layers, filter size, and pooling size, are also evaluated for finding the best combinations that yield the best accuracy [41].

Recurrent neural network RNN is an MLP where the input is treated as a temporal sequence from t. In contrast with a feed-forward network, for instance, a fully connected layer in CNN, each perceptron's weight and biased parameters also share from $t-1$ to t as in figure 5.10. The sharing of the parameter indicates the dependency of each value in the sequence of

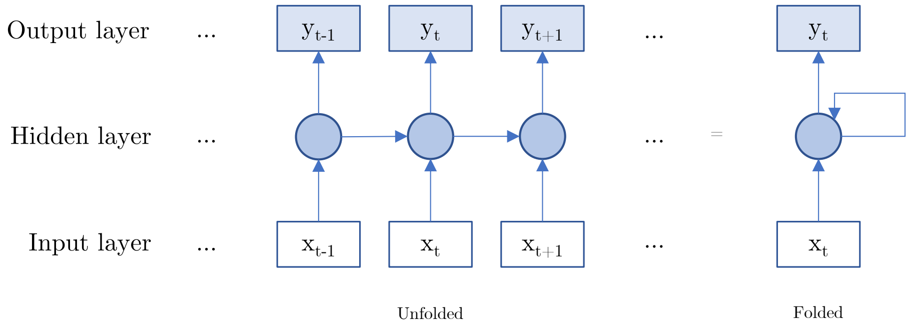


Figure 5.10: A visualization of the recurrent neural network shows that the unfolded version of the RNN emphasized parameter sharing.

input. A typical RNN architecture comprises an input layer, a single hidden layer, and an output layer. The arrow from perceptron at $t-1$ to t indicates that the parameter is being shared. The RNN architecture is many-to-many, or the input and output lengths are equal. It receives a sequence $x_{t_{0:n}}$ of single-channel input and provides a sequence $y_{t_{0:n}}$ as output.

There are various RNN architectures; LSTM is a popular variation of RNN due to its ability to handle the problem of vanishing gradients [96]. The problem arises during backpropagation, where only the last perceptrons and hidden layers usually receive significant weight updates. This update decreases from t to $t-1$, eventually leading to no or tiny update at early perceptrons and hidden layers. LSTM solves this by introducing an additional mechanism into its perceptron or cell, gradually increasing the weight update between layers. In addition, LSTM can forget or ignore the parameter from the cell and hidden layer, allowing it to ignore the irrelevant long-term dependencies.

- *RNN architectures for human activity recognition* - Human activity can be measured and encoded in the form of Spatio-temporal; for example, the movement of a limb can be encoded as a trajectory or changes of measurements, including acceleration and orientation in a temporal domain. A typical pure RNN-based architecture is usually concatenated with a dense layer for classification tasks. A Deep RNN-LSTM with three hidden layers is usually implemented as a baseline comparison [51]. An example of a deep RNN with three hidden LSTM layers is presented in figure 5.11. It receives a single-channel temporal sequence input. Outputs from the final LSTM layer are usually forwarded to the fully connected layer for performing classification. Multiple channel inputs are separately treated by separating each LSTM layer for

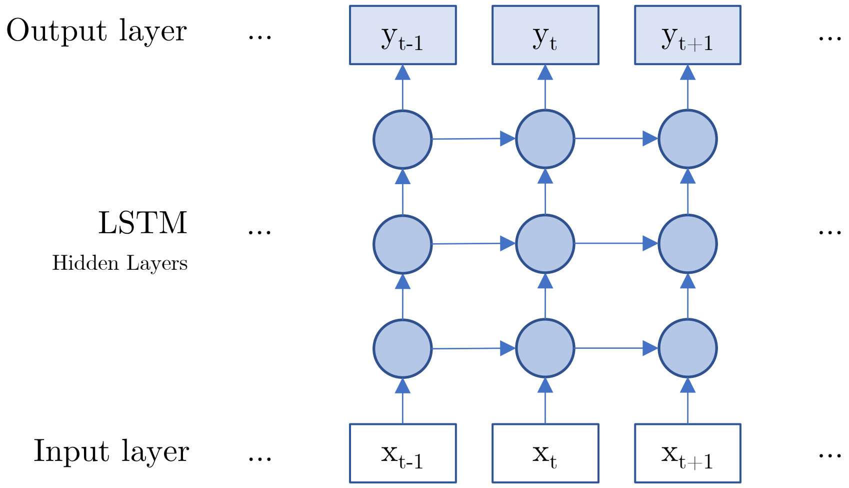


Figure 5.11: A visualization of the unfolded version of the LSTM.

each channel; the output combines using a fully connected layer. Some variations for realizing ensembled architecture can also be performed using train-validation-test model selection and split [69].

Hybrid combination of deep learning network The combination of deep learning architecture is mainly from an assumption that the different types of architecture handle different features. For instance, CNN handles the spatial features, while RNN handles temporal features. The main difference between architecture proposals can be categorized mainly by the order of the architecture components. For instance, CNN and then RNN or the RNN then CNN. The CNN-LSTM is when the input is passed through the CNN layers, and then the output feature map is flattened for the LSTM layers. Lastly, the output from RNN is passed through the fully connected or dense layer [55], [105], [106]. On the opposite, the LSTM-CNN is also proposed [112].

A baseline LSTM architecture is selected for implementation to demonstrate EXAMINER case-study implementation. HyperBand is selected for fine-tuned hyper-parameter models, including the number of layers, nodes, and drop out [71]. The tuning was performed based on the assumption that input characteristics are different from the baseline model as follows

- *Activity*, most of the state-of-the-art (SOTA) model was developed based on the common human daily activity. EXAMINER, in contrast, focuses on MA-related activity.

Component	Parameter	Value
LSTM	Number of layer	3
	Number of nodes	448
Training	Dropout	0.7
	MaxNorm	2
	Learning Rate	0.01

Table 5.1: The table summarize the optimized LSTM model's hyperparameters including the parameter for training the model.

- *Input length* - most of the architecture develops based on the publicly available action recognition dataset [27]. The dataset records the common daily activity, which, on average, has a longer sequence than MA activity.
- *Sensor* - SOTA ACT recognition models mainly employ multi-modal sensor input, including gyroscope and accelerometer. Here, pose information from a vision-based sensor has different characteristics as it reports trajectory, not acceleration and orientation.

After performing hyper-parameter tuning, the summarized architecture is presented in the table 5.1 together with the visualized model architecture in figure 5.12. Here, the visualization shows a single channel input. An extension to multiple-channel or multi-modal requires horizontal scaling or independently stacking the LSTM hidden layers. The output from each channel later combines and reduces using a pooling layer and provides class probabilities using a fully connected layer [17].

Model inference

The model inference is the process of providing unseen data as input to the recognition model. The model assigns the corresponding label with the highest class probability to the input by doing so. It either provides the correct classification or misclassifies in comparison with the ground truth. Continuous activity recognition model performance can be analyzed differently depending on the application. For instance, the performance can be measured by correctly identifying the activity and its boundary. It is also trivial to assume that the model is not perfect and will produce some errors as EXAMINER has to compare digitized skills later and provide feedback. The error from the activity recognition model will propagate through components and provide false feedback. Without considering activity boundary,

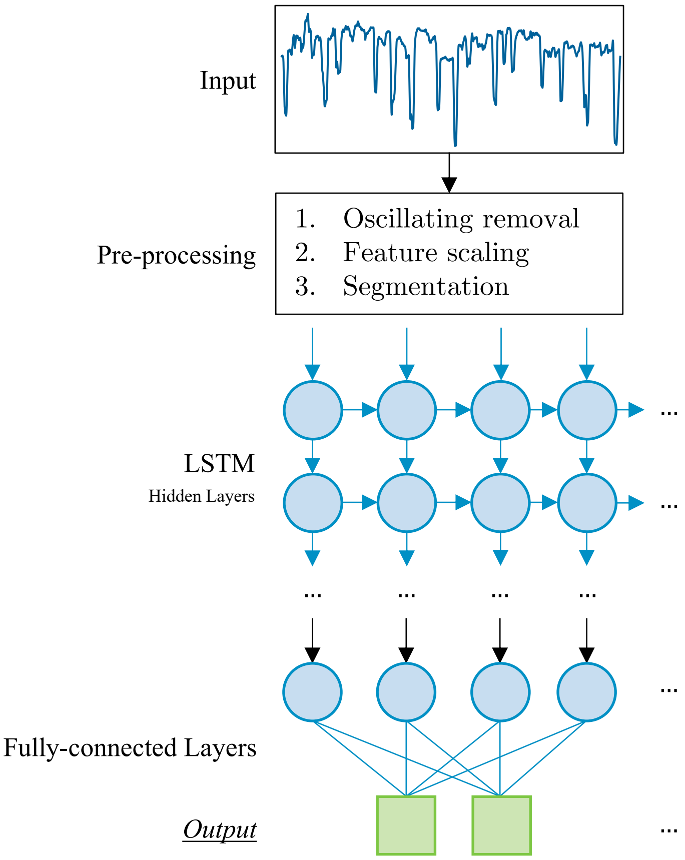


Figure 5.12: A visualization of LSTM architecture for single channel input activity recognition.

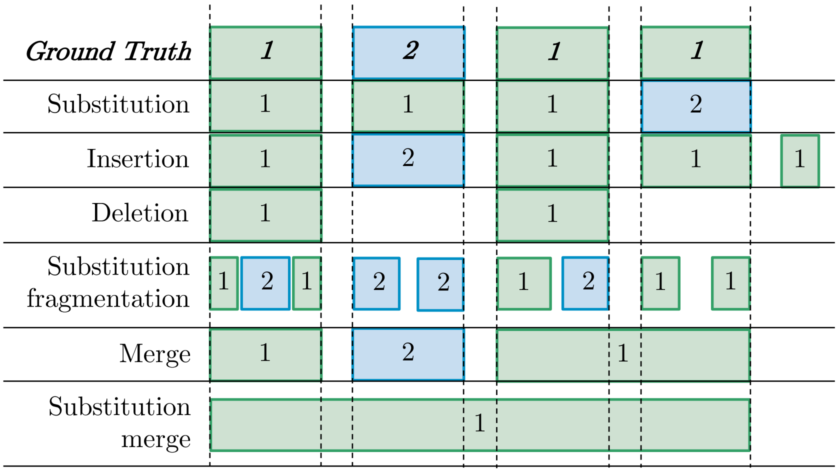


Figure 5.13: The visualization of possible error introduces by inference ACT recognition model.

as will be discussed later, five main types of error and two combination errors are introduced in the literature as visualized in figure 5.13 [10].s

These error measurement forms are introduced in addition to the general evaluation and ranking matrix, including the confusion matrix, f-measure, and receiver operating characteristic.

Even though humans can ignore false augmented feedback because it conflicts with personal intuition or intrinsic feedback, these introduced errors should be considered and mitigated because they cause nuance to the user. For instance, the experimenter may revisit each step in the pipeline, tweak parameters, and perform the tasks differently to compare the changes in model performance.

5.1.2 Assembly Space Context Sensing

However, the ACT is only a primitive activity lacking location context. The later stage of step digitization compares the differences in context to define the state of the assembly-related item. Commonly, on the assembly surface, there are four fixed spaces, including the assembly area, parts area, tool area, and the submission area, as visualized in figure 5.14.

Vision-based activity context sensing is research under image understanding. Setting up the area is usually carefully done by the expert. It mainly focuses on motion efficiency, organization, and ergonomics, which are not the focus of this dissertation. Most of the proposal aims to describe the given input by combining the perceived contexts from various sensing models. For

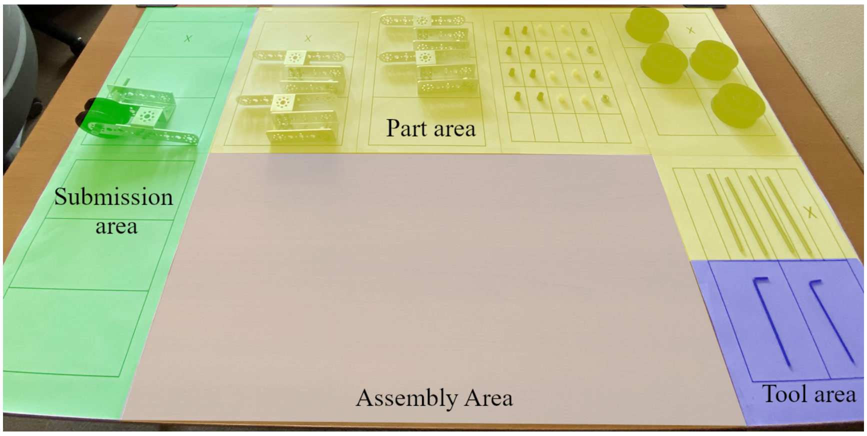


Figure 5.14: A simulated assembly cell at the perspective of the operator, consisting of non-overlapping spaces.

instance, the activity recognition model can be combined with the object recognition model to describe the activity context [48]. There are various ways to perform the sensing task visually. For instance, activation of the region of interest(ROI) registers activity in the surveillance camera. The ROI is a sub-region in F specified for further analysis defined as a closed set on $\mathbf{R^2}$ consisting of $p_{x,y}$ on F. The technique is proven to work in the assembly space set up with a material organization bin as the activation region can be located at the bin's boundary [66]. EXAMINER employs context comparison on the ROI instead. It ensures the context state, including present and not present, of tools and material. Background and object segmentation is a possible technique to detect changes in image context. It is one of the topics heavily studied in computer vision. In the simplest form, image binary thresholding is a technique that can perform an image's segmentation. It changes the grayscaled $p_{x,y}$ to 0 or 255 based on the preset threshold pixel values. By doing so, the presence of the assembly object in the ROI_n is going to appear as a white silhouette with the $p^T_{x,y} = 255$ as the color of the object appears to be contrasted with the scene. Additional techniques, including contour detection, can also be applied to detect the silhouette's shape and boundary. However, additional state handling or computer vision techniques may be required if the object and the assembly surface fail to meet the mentioned condition. EXAMINER assumes that all area tools and parts are spreading out and not overlapping for demonstration purposes. Each item is going to have its designated area on the surface. Before doing so, this dissertation introduces the concept of automatic detection of the region of interest. It also presents some of the underlying concepts applied in context sensing.

Automatically Detection of the Region of Interests

The camera location may change, or there is a slight difference in the scene setup. The user can deploy the system remotely and separately. Automated Region of Interests Calibration is a process where the rectangle ROIs are semi-automatically recognized and calibrated on the given input F scene. It is required to be done once at every physical change in the setup. The summarized process is as follows,

1. **Remove background noise**
   The selected sampling frame F is filtered using an edge-preserving smoothing to remove background noise called a bilateral filter. The image sensor is subjected to Gaussian noise during acquisition, resulting in some grains in an obtained F. These noises can cause the image processing techniques in the later step to produce random behavior; hence, they need to be filtered out. There are various noise reduction techniques. The simplest way is to blur the image using Gaussian blur. The noise is significantly filtered out, but the edge feature is lost. A bilateral filter is introduced to tackle this by changing the filter kernels based on the shape of the input [5]. For instance, instead of using the same Gaussian kernel on all $p_{x,y}$, the Bilateral filter introduces a range weight term that varies by $p_{x,y}$ to the Gaussian kernel.

2. **Convert to grayscale**
   As the color information is not necessary to detect the ROI as the boundary is presented by a black line with low intensity, the $F^{filtered}$ is converted to grayscale to generalize the intensity on each $p_{x,y}$. A grayscale image is a representation of an amount of intensity ι on each $p_{x,y}$. For each $p_{x,y}$ a grayscale pixel $\iota_{x,y}$ can be obtained by applying the sum of the product based on the NTSC formula as

   $$\iota_{x,y} = (0.299\iota_1 + 0.587\iota_2 + 0.114\iota_3) \tag{5.4}$$

   Where ι_n represents the intensity value for red, green, and blue, respectively. By applying on all $p_{x,y} \in F$, an F^{grayed} is obtained.

3. **Construct a binary image**
   A binary image construction is a step requires prior to the detection of the image contours, here it converts the eight bits unsigned intensity value of $p_{x,y}$ into either 0 or 255 using the predefined threshold values.

For instance, the black line with very low intensity is going to represent as 0, otherwise 255, this can be however swapped based on the implementation preferences. A threshold pixel $p_{x,y}^{thresh} \in 0, 255$ from a grayscale pixel $\iota_{x,y}$ can be obtained as follows

$$p_{x,y}^{thresh} = \begin{cases} 0, & \iota_{x,y} > T_{x,y} \\ 255, & otherwise \end{cases} \tag{5.5}$$

4. **Detect contour** In comparison with the edge detection, where it performs the differential between the neighbors $p_{x,y}$ looking for the drastic changes in intensity and joining them together as a curve. Contours detection, in contrast, joins all the neighbor's pixels having the same color or intensity, forming a close [2]. However, contour detection is a method that should be considered if the image contains a shape that has to be recognized. It requires an earlier process of transforming the input frame to represent black and white so that the contour detection algorithm can detect the boundary of the same color.

5. **Polygons approximation** Approximate polygons use the output from closed contours to construct the polygons. The output is defined as a set of all close polygon chains with each vertex's coordinates $\mathbf{R^2}$.

6. **Removes irrelevant polygons** It is a process to filter out irrelevant polygons that do not meet the requirement to be an ROI. For instance, the number of geometric sides(edges) and size must be under the predefined boundary. For instance, the rectangular ROI should have four edges and a size in a suitable range. Only a polygon that has four points is considered. Otherwise, the polygons are discarded.

7. **Inspects result** The expert must carefully inspect the resulting ROI from the process, as the process may fail to include all ROIs. The compression introduced by the camera encoding automatically applies anti-aliasing on F, making the intersection of lines grayed out. Once it is done, the ROI coordinates data is obtained. The expert later specifies the context information of each ROI.

Assembly State Transition

EXAMINER handles two types of context: the availability of items relevant to MA, and the positioning of limbs in relation to the region of interest (ROI). The assembly space context capture subcomponent provides transition information when the availability state of either item or limb(s) at the

ROI changes from present (P) to not present (NP), or vice versa. This information is later used to record $TOPT$. To achieve this, vision-based context sensing is employed, leveraging the existing camera installation for skeletal-2d-HPE. In Figure 5.14, EXAMINER is configured with four predefined areas on the assembly surface: the assembly area, the parts area, the tool area, and the submission area, each bounded by geometric boundaries. This dissertation first addresses the methodology for recognizing the transition moment of item availability shortened as the item's availability module, followed by the recognition of limb positioning, as the underlying mechanisms for obtaining these two contexts differ.

Item's availability module Identifying transitional moments in the availability state of MA-related items involves handling state changes in the parts, tools, and submission areas. To achieve this, this sub-component may employ various computer vision techniques, which can be categorized into neural network-based and non-neural approaches. Models like YOLO (You Only Look Once) [56], SSD (Single Shot MultiBox Detector) [53], and Faster R-CNN [44] represent the neural network category and are highly effective for real-time object detection, recognizing, and precisely locating objects within images.

Despite these advantages, neural network approaches require extensive data labeling and training and typically demand substantial computational resources. While these models are invaluable in contexts requiring advanced object recognition and localization, their implementation is excessive in settings with static or predictable environments.

A non-neural approach is advantageous for the MA station under EXAMINER, with its fixed physical configuration and specific geometric boundaries. This method focuses on image preprocessing, feature extraction, and detection, which are well-suited to EXAMINER-controlled conditions. Hence, it meets the requirements without the additional complexity and resource demands of neural network models.

The following is an implementation of non-neural network approaches to identify the transitional moment of MA-related items. In the initial step of image preprocessing, all $p_{(x,y)} \in F$ is converted to grayscale $GRAY(p_{(x,y)})$, and bilateral filtering is then used to perform edge-preserving smoothing. Converting $p_{(x,y)}$ to grayscale reduces by at least two-thirds the memory usage and time complexity of the subsequent image processing step while preserving the intensity of all color spaces. Then, edge-preserving smoothing reduces image noise introduced by the camera sensor while maintaining the sharpness of the edge features.

Second, image feature extraction employs an inverted binary thresholding. It is a technique for image segmentation typically employed when the background is plain and contrasts with the object. The opted image feature extraction operation involves the conversion of the grayscale value $GRAY(p_{(x,y)})$ to either 0 or 255, based on the threshold value $THLB$. Since the setup assumes a white background, the binary thresholding of the image can be achieved using the following equation:

$$p'_{(x,y)} = \begin{cases} 0 & GRAY(p_{(x,y)}) > THLB \\ 255 & \text{otherwise} \end{cases} \tag{5.6}$$

This equation is applied to all pixels $p_{(x,y)}$ in image F. Consequently, the presence of the assembly object in the image will be represented as a white silhouette against a black background. Additional techniques, such as contour detection, can determine the shape and boundary of the silhouette. However, suppose the object and the assembly surface do not meet the earlier requirement. In this case, additional state handling or another computer vision technique may be required.

Thirdly, the proposed approach performs image segmentation by defining regions of interest (ROI_λ), where λ serves as an identifier for each object's location (LO). Each ROI_λ represents a closed set in $\mathbb{R}^2$ and is constructed from the pixels $p_{(x,y)}$ in image F. These regions of interest are designated for subsequent analysis and further processing. Under consideration of EXAMINER, ROI_λ is an expert's predefined geometric boundary for each assembly item, tool, and area. Each item is assigned a specific location on the assembly surface. For demonstration, the system assumes that all objects, tools, and parts are dispersed, non-overlapping, and positioned within a predetermined, fixed boundary. In addition, there are no mistakes caused by placing any item not belonging to the ROI_λ. This predefined geometric boundary ensures that the objects of interest are isolated within their respective boundary regions.

Lastly, the proposed approach focuses on recognizing state transition $STRAN \in \{NPtoP, PtoNP\}$ of ROI_λ, and records the transition moment as

$$H_\eta = (T^{(\eta)}, STRAN^{(\eta)}, ST^{(\eta)}) \tag{5.7}$$

where $\eta \in \mathbb{N}$ represents the order index of the transaction starting at $\eta = 1$. Here, T is the specific moment when $STRAN_\eta$ occurs, and ST_η represents the outcome of $STRAN_\eta$ either P(present) or NP(not-present). To maintain a comprehensive record of these transactions, a transaction list is defined as:

$$TXN_{ROI,\lambda} = \{H_1, H_2, H_3, ..., H_n\} \tag{5.8}$$

To populate the transaction list $TXN_{ROI,\lambda}$, the recognition module continuously compares the summation of pixel values $p'(x,y)$ within the region of interest ROI_λ at a specific moment T. The summation function $\sigma(\lambda, T)$ is defined as follows:

$$\sigma(\lambda, T) = \sum_{p'(x,y) \in ROI_{\lambda,T}} p'_{(x,y)} \tag{5.9}$$

In Equation (5.9), the moment T can take on two possible values: $T = \tau$, indicating the present frame, and $T = \tau - c$, representing c frames in the past. By doing so, the $\sigma(\lambda, T)$ with an object is going to have higher values as it consists of more $p'_{(x,y)} = 255$ than the empty one as demonstrated in Figure 5.15. The figure visualizes an ROI_λ going through both types of $STRAN_\eta$, demonstrating the action of the operator picking the tool and returning it to its original location after use. The comparison between the present moment $T = \tau$ and the past moment $T = \tau - c$ can be quantified using the difference function $\Delta(\lambda, c)$, defined as:

$$\Delta(\lambda, c) = \sigma(\lambda, \tau) - \sigma(\lambda, \tau - c) \tag{5.10}$$

The equation calculates the difference between the summation of pixel values within the region of interest ROI_λ at the present moment and the past moment separated by a temporal distance of c frames, allowing for the detection of state transitions of ROI_λ.

In the initialization phase, each H_1 in the transaction list $TXN_{ROI,\lambda}$ is populated with the tuple $(1, \emptyset, ST)$. The state ST is determined based on the difference between the summation of pixel values within ROI_λ at the first moment, $\sigma(\lambda, 1)$, and a prior measurement $\omega_\lambda \in \mathbb{N}$ is a measurement when an object belongs to ROI_λ is (P) using Equation (5.9). The state ST is assigned as follows:

$$ST = \begin{cases} P, & |\sigma(\lambda, 1) - \omega_\lambda| \approx 0 \\ NP, & \text{otherwise} \end{cases} \tag{5.11}$$

If the absolute difference between $\sigma(\lambda, 1)$ and ω_λ is approximately zero, indicating availability of object in ROI_λ, the state ST is assigned as P. Otherwise, if there is a significant difference, the state ST is assigned as NP, indicating an absence of the object in ROI_λ at the beginning of the assembly process. Later when $T > 1$, the transaction list $TXN_{ROI,\lambda}$ is appended with H_η with $STRAN \in \{PtoNP, NPtoP\}$. First, to recognize $PtoNP$ the function having a precondition that ST in the latest H_η is P:

$$PtoNP = \begin{cases} \text{true}, & \omega_{\lambda,L^-} \leq \Delta(\lambda, c) \leq \omega_{\lambda,U^-} \\ \text{false}, & \text{otherwise} \end{cases} \tag{5.12}$$

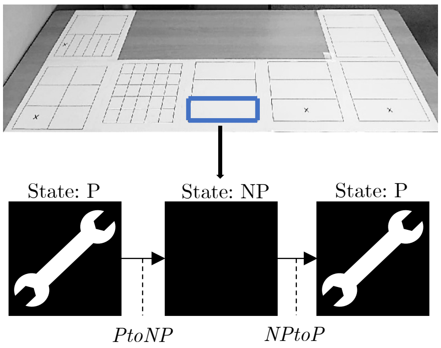


Figure 5.15: State transition of a hand tool from P to NP and back to P.

Here, ω_{λ,L^-} and ω_{λ,U^-} are predefined lower (L) and upper (U) bound constants, respectively, in negative integer $\mathbb{Z}^-$. These values can be obtained from prior measurements of the difference between $\sigma(\lambda, T)$ at the moment when the object is NP and the moment when the object is P. The introduction of L and U makes the transition more robust against any possible overshoot and undershoot of $\sigma(\lambda, T)$ that occurs during the state transition. While $ROI_{\lambda,\tau}$ is in state P, the function continuously compares $\sigma(\lambda, T)$ between the present moment ($T = \tau$) and an earlier moment ($T = \tau - c$). Since ROI_λ with an object present contains more $p'_{x,y} = 225$ values compared to when it is absent, $\sigma(\lambda, T = \tau)$ is significantly smaller than at $T = \tau - c$. In such cases, $H\eta = (\tau, PtoNP, NP)$ is recorded in the transaction list to signify the transition. On the other hand, the transition from NP to P occurs with the precondition that the latest H_η is NP. The condition for recognizing the NP to P transition is as follows:

$$NPtoP = \begin{cases} \text{true}, & \omega_{\lambda,L^+} \leq \Delta(\lambda, c) \leq \omega_{\lambda,U^+} \\ \text{false}, & \text{otherwise} \end{cases} \tag{5.13}$$

In this case, ω_{λ,L^+} and ω_{λ,U^+} are predefined upper (U) and lower (L) bound constants, respectively, in positive integer $\mathbb{Z}^+$.

EXAMINER proposed the digitized method that can perform operation time digitization automatically by utilizing a traditional computer vision pipeline. As a result, it is necessary to manually adjust the parameters of each computer vision process under the pipeline so that the silhouette of parts and tools is clearly in contrast with the assembly desk. Based on the introduced computer vision pipeline, two processes under the pipeline require parameter adjustment, including Bilateral filtering and binary thresholding. Three parameters under Bilateral filtering must be adjusted, including $Diameter$, $SigmaColor$, and $SigmaSpace$. The $THLB$ can be estimated

Table 5.2: Parameters of Computer Vision Pipeline

Pipeline	Parameter	Value
Bilateral filtering	*Diameter*	5
	SigmaColor	40
	SigmaSpace	40
Adaptive binary thresholding	*adaptiveMethod*	Mean
	blockSize	15
	constant	4

manually or automatically for binary thresholding using Otsu's method or OpenCV's adaptive thresholding. OpenCV's adaptive thresholding is chosen based on the experiment setup to contrast the scene's parts and tools. The thresholding method divides the image into smaller regions and calculates a threshold for each region. Because the environment under recording lacks a dedicated light source, this method mitigates the varying light conditions in different parts of the image. Three adaptive thresholding parameters will be set: $adaptiveMethod$, $blockSize$, and $constant$. The parameter values for Bilateral filtering and adaptive binary thresholding are shown in Table 5.2. In this dissertation, the parameter was manually adjusted to reduce image noise and improve the edge feature of parts and tools without compromising any necessary image features.

Based on the proposed method of determining ST, $PtoNP$, and $NPtoP$ in Equations (5.11), (5.12), and (5.13) for each ROI, the experiment is required to determine the c of Equation (5.10), ω_λ, $\omega_{\lambda,L}$ and $\omega_{\lambda,U}$. The experiment uses $c = 50$ or a one-second period. Table 5.3 provides an example of the initial state ST decision boundary and the transactional period from present not to present $PtoNP$ manually obtained for each ROI_λ. Each decision boundary for each ROI_λ is different due to the object's size and the camera's distance. Generally, the bigger objects appear on the camera, the higher the value for the decision boundary.

Up to this point, the module can register a state transition $TXN_{ROI,\lambda}$ of part, tool, and submission area. The "Item's availability module" of the EXAMINER framework employs a non-neural approach to identify transitional moments in the availability state of MA-related items, offering a precise and resource-efficient solution. This method adeptly manages the state changes of parts, tools, and submission areas within a controlled environment by utilizing straightforward image processing techniques such as grayscale conversion, bilateral filtering, and binary thresholding. The selected techniques are particularly well-suited to the fixed and predictable setup at the MA station,

Table 5.3: Eaxmple of state transaction decision boundary for each ROI

ROI_λ	ST	$PtoNP$	
	ω_λ	ω_{λ,L^-}	ω_{λ,U^-}
Chasis1	1,000,000	-500,000	-700,000
Wheel1	822500	-1,000,000	-755,000
Bushing1	47,000	-94,000	-130,000
Rod1	161,000	-282,000	-328,000
Spacer1	19,000	-28,000	-61,200
Collar1	29,750	-55,000	-106,500
Hex1	101,000	-155,000	-188,000
Hex2	108,500	-161,000	-195,000

ensuring high accuracy and low computational demand. This approach not only simplifies the implementation but also aligns perfectly with the operational needs of EXAMINER, proving that the method is not only adequate but ideally optimized for monitoring state transitions in industrial assembly settings. By effectively recognizing and recording these transitions, the module provides a robust foundation for the subsequent processing stages, which are crucial for the comprehensive monitoring and analysis required in manual assembly tasks. Next, the dissertation addresses the implementation of recognition of the transition moment on the limb's position concerning ROI_λ shortened as the limb's availability module.

Limb's availability module The limbs in the assembly area detection module verify the spatial positioning of the limbs by evaluating the two-dimensional Cartesian coordinates of target limbs obtained from skeletal-2d-HPE. Specifically, it determines whether the coordinates of the targeted limb fall within the designated ROI_λ defined for the assembly area. The module records transition in the same form as introduced in Equation (5.8); however, λ must be any of an assembly area, and the methodology to obtain H_η especially, $TRANS$ and ST differs as the information can be directly leveraged from skeletal-2d-HPE.

The ray casting algorithm, also known as the crossing number algorithm or even–odd rule algorithm, is suitable for verification and was chosen due to its simplicity and efficiency in determining point-in-polygon inclusion. The method depicted in Figure 5.16 casts a ray from the point in the direction of the x-axis and counts the number of intersections between the ray and the polygon's edges. If the ray from point $JT_{(x,y)}$ where JT can be either left or right or both wrists intersect the polygon's edge at odd intervals, in this

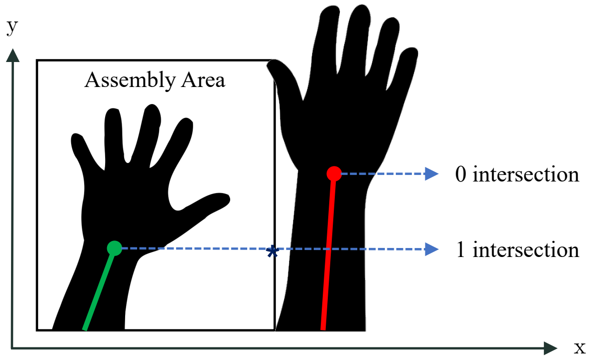


Figure 5.16: The left wrist's ray intersects the assembly area's edge, indicating that it is inside.

case, ROI_{λ} where λ is an assembly area, the $JT_{(x,y)}$ presented (P) within the ROI_{λ} otherwise it is not presented (NP).

The population of the transaction $TXN_{ROI,\text{"assembly area"}}$ can be achieved using the following steps. During the initialization phase, the transition moment H_1 is filled with the tuple $(1, \emptyset, ST)$. The state ST is obtained directly from the ray casting algorithm, either P or NP. As time progresses $(T > 1)$, the transaction list $TXN_{ROI,\lambda}$ is expanded by adding H_{η} with $STRAN \in PtoNP, NPtoP$. Two conditions must be met to populate the $PtoNP$ transition moment: the state ST in the latest H_{η} is P, and the output from the ray tracing algorithm is NP. Conversely, to populate the $NPtoP$ transition moment, the state ST in the latest H_{η} must be NP, and the output from the ray tracing algorithm is P. By doing so, the moment a limb enters or exits an assembly area can be recorded.

The Limb's Availability Module effectively uses the ray casting algorithm to verify whether the operator's limbs are within the designated assembly area, ensuring precise tracking of limb movements and transitions in a simple and straightforward manner.

All necessary data for digitizing cognitive skills, including $TOPT$, LO_{init}, and LO_{dest}, have been obtained from activity recognition and assembly space context recognition modules. The pipeline continues digitizing the $TOPT$ using the recently generated $TXN_{ROI,\lambda}$.

5.2 Motor Skill Digitization

To accurately evaluate the operator's motor skills, including a sub-component for recording operation time $(TOPT)$ is essential. This sub-component mea-

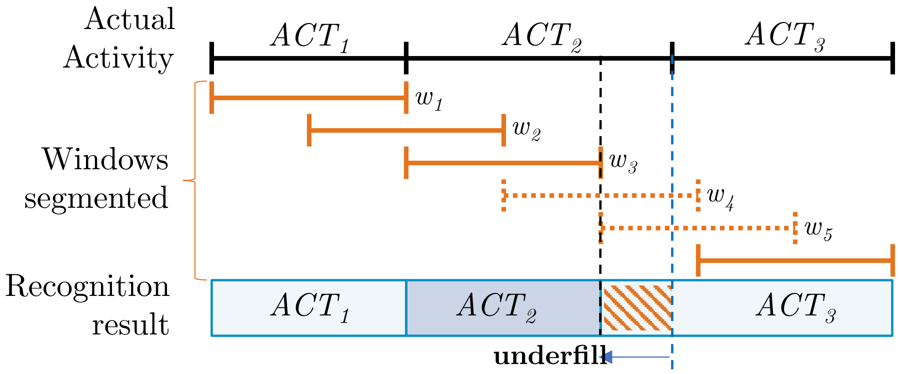


Figure 5.17: Ambiguity of ACT boundary

sures and records the time taken for each performed MA activity. The decision to incorporate a dedicated sub-component for recording operation time stems from the inherent ambiguity of activity boundaries in the MA activity recognition sub-component. Figure 5.17 illustrates the potential overlapping segments between consecutive activities. This ambiguity arises when an activity, such as ACT_2, exceeds the length of a single window, for instance, $|ACT_2| \geq |w_1|$, resulting in overlapping activity segments.

In the depicted scenario, windows w_4 and w_5 contain partial actual activity of both ACT_2 and ACT_3. Utilizing an activity recognition sub-component with the majority voting for result aggregation may preserve the correct sequence of activities. However, it may underestimate the duration of ACT_2 and overestimate ACT_3 as w_5 extends back to the duration of ACT_2.

Another key consideration is the frame rate of the video camera used for recording. Consumer video cameras typically operate at 15 to 120 frames per second (FPS). This means the time measurement error can be up to $\pm\frac{1000}{\text{FPS}}$ milliseconds. For time-sensitive applications, a high-speed or dynamic vision sensor event camera can be chosen to reduce measurement errors [12].

Industrial manual operation heavily utilized motion time study. It is a study aiming to reduce waste introduced by an operator's poorly optimized movement trajectory to perform a manual task. An observer will use a stopwatch to record the operator's $TOPT$ at each step to measure time. This enables the firm to precisely measure, record, and perform statistical analysis of $TOPT$. It is crucial as the firm will use this information to plan production. Stopwatch timing is a method prone to human error and must be carried out by a specific expert. Various studies attempt to automate this process by collecting and processing data from multiple vision and non-vision sensors installed on the assembly station. The models aim to detect the starting and stopping points of an activity. Because the camera was already installed in the simulated assembly station, EXAMINER chose a vision-based methodology. There are two types of detection models: supervised and unsupervised. A supervised recognition model uses available data to label and train the model. For example, a video frame F containing the starting and stopping point of the MA activity can be labeled and directly used to train the image recognition model [103]. Instead, the unsupervised model infers them from the learned frame features. For example, the output of skeletal-2d-HPE in time-series format can be inferred directly using search methods and cost functions to detect change points in time series [109].

EXAMINER precisely measures the operation time ($TOPT$) for each ST by utilizing the context state changes recently captured by assembly space context capture sub-components and the activity sequences recognized by the MA activity recognition sub-components to enhance accuracy and reduce redundancy, introducing an additional technique for measuring the operation time. The measurement focuses on adjusting ACT's beginning or ending points or both, which are classified as an adjustment at PR levels. Based on the evaluation case study for EXAMINER, the measurement can be divided into four primary measurement cases: a reach, a retracting, an assembly, and a tool use. Each instance is described as follows:

1. **A reach** - It is a primitive step (PR) in which the operator performs motion (ACT = "reach"). The action includes acquiring the component or tools at ROI_λ and submitting the final assembly. EXAMINER has identified five types of reach, each requiring a distinct measurement strategy for operation time ($TOPT$).

   (a) **An initial reach** is a special case that marks the beginning of an MA iteration. Here, the operator reaches for the first part or tools at ROI_λ, where λ is an identifier for that particular part or tool. Duration is measured from the moment when ($ACT = \epsilon$) changes to (ACT = "reach") to the moment when ($ACT \neq$ "reach"). An adjustment of the start and stop points of the activity using $TXN_{ROI,\lambda}$ can be obtained as follows:

   - **Start point**: Use the value T of $H_2 = (T, PtoNP, NP) \in TXN_{ROI,\text{"assembly area"}}$.
   - **Stop point**: Use the value T of $H_2 = (T, PtoNP, NP) \in TXN_{ROI,\lambda}$ given that λ is not either "assembly area" or "submission area."

   Using H_η, with $\eta = 2$, is specific to the EXAMINER's case study, implying that it considers values from transactions following the initialization (where $\eta = 1$) and assumes that at the beginning any of the hand is located in the assembly area. The actual implementation may vary depending on the specific MA.

(b) **A reach follows assembly or retract** - It is a motion in which the operator reaches for ROI_λ where $\lambda \in \{\text{"part area"}, \text{"tool area"}\}$ after performing an assembly or retracting the previous part/tool. The latter usually happens after reaching the first part of MA iteration. The duration is measured from $(ACT = \text{"retract"}, \text{"assembly"}, \epsilon)$ changes to $(ACT = \text{"reach"})$ to $(ACT \neq \text{"reach"})$. An adjustment of the start and stop points of the activity can be obtained as follows:

- **Start point**: Use the value T of $H_\eta = (T, \text{"PtoNP"}, NP) \in TXN_{ROI,\text{"assembly area"}}$, where $\eta > 3$. Using H_η, with $\eta > 3$, implies that it considers the value from the fourth transaction, meaning that after the initialization, another two transactions happen under an assembly area.
- **Stop point**: Use the value T of $H_\eta = (T, \text{"PtoNP"}, NP) \in TXN_{ROI,\lambda}$, where $\eta > 1$, and λ can be either the "part area" or the "tool area. This means that the operator picked a tool/part from ROI_λ. Using H_η, where $\eta > 1$, implies that it considers the value after the initialization.

(c) **A reach following tool use** - It is a motion where the operator returns the tool to its original ROI_λ. The duration is measured from $(ACT = \text{"tool use"}, \epsilon)$ becomes $(ACT = \text{"reach"})$ and then $(ACT \neq \text{"reach"})$. An adjustment of the start and stop points of the activity can be obtained as follows:

- **Start point**: Use the value T of $H_\eta = (T, \text{"PtoNP"}, NP) \in TXN_{ROI,\text{"assembly area"}}$, where $\eta > 3$.
- **Stop point**: Use the value T of $H_\eta = (T, \text{"NPtoP"}, P) \in TXN_{ROI,\text{"tool area"}}$, where $\eta > 1$.

(d) **A reach for submission of a finished assembly** - As the name suggests, is a motion of placing a finished product in the designated ROI_λ. The duration is measured from the moment when $(ACT = \text{"retract"}, \text{"assembly"}, \text{"tool use"}, \epsilon)$ becomes $(ACT = \text{"reach"})$ and then $(ACT \neq \text{"reach"})$. An adjustment of the start and stop points of the activity can be obtained as follows:

- **Start point**: Use the value T of $H_\eta = (T, \text{"PtoNP"}, NP) \in TXN_{ROI,\text{"assembly area"}}$, where $\eta > 3$.

- **Stop point**: Use the value T of $H_\eta = (T, \text{"NPtoP"}, P)$ $\in TXN_{ROI,\text{"submission area"}}$, where $\eta > 3$.

(e) **A reach following reach** - It is a motion in which the operator performs a second reach immediately after submitting a completed assembly or returning the tool. As it is possible for the operator to perform a subsequent reach after the reach for submission, the MA activity recognition system is not designed to distinguish between successive reaches. As of that, the measurement disregards T from the MA activity recognition. With the prerequisite that ACT = "reach", an adjustment of the start and stop points of the activity can be obtained as follows:

- **Start point**: Use the value T of $H_\eta = (T, \text{NPtoP"}, P) \in TXN_{ROI,\lambda}$, where $\eta > 3$, and λ is either "part area" or "tool area."
- **Stop point**: Use the value T of $H_\eta = (T, \text{"PtoNP"}, NP) \in TXN_{ROI,\lambda}$, where $\eta > 3$, and λ is either "part area" or "tool area."

2. **A retract** - It is a primitive step in which the operator performs (ACT = "retract") immediately after a reach. Currently, EXAMINER recognizes one type of retract as a retract after reach, and duration is measured from the moment activity recognition recognized ($ACT = \text{"retract"}l, \epsilon$) to the moment when ($ACT \neq$ "retract"). An adjustment of the start and stop points of the activity can be obtained as follows:

   - **Start point**: Use the value T of $H_\eta = (T, \text{"PtoNP"}, NP) \in TXN_{ROI,\lambda}$, where $\eta > 1$, and λ is either "part area" or "tool area."
   - **Stop point**: Use the value T of $H_\eta = (T, \text{"NPtoP"}, P) \in TXN_{ROI,\text{"assembly area"}}$, where $\eta > 2$. Using H_η, with $\eta > 2$, implies that it considers the value from the third transaction, meaning that after the initialization, a transaction happens under an assembly area.

3. **A tool use** - Is is a primitive step in which the operator performs (ACT = "tool use") after a retract. Currently, EXAMINER recognizes one type of tool use that happens after a retract (and ϵ), and duration is measured from the moment activity recognition recognized (ACT = "tool use") to the moment when ($ACT \neq$ "tool use"). An adjustment of the start and stop points of the activity can be obtained as follows:

- **Start point**: Use the value T of $H_\eta = (T, \text{"PtoNP"}, NP) \in TXN_{ROI,\text{"tool use"}}$, where $\eta > 1$.
- **Stop point**: Use the value T of $H_\eta = (T, \text{"NPtoP"}, P) \in TXN_{ROI,\text{"assembly area"}}$, where $\eta > 2$.

4. **An assembly** - Is is a primitive step in which the operator performs ($ACT = \text{"assembly"}$) immediately after a retract. Currently, EXAMINER recognizes one type of assembly that happens after a retract, and duration is measured from the moment activity recognition is recognized ($ACT = \text{"assembly"}$) to the moment when ($ACT \neq \text{"assembly"}$). An adjustment of the start and stop points of the activity can be obtained as follows:

   - **Start point**: Use the value T of $H_\eta = (T, \text{"PtoNP"}, NP) \in TXN_{ROI,\lambda}$, where $\eta > 1$, and λ is either "part area" or "tool area"
   - **Stop point**: Use the value T of $H_\eta = (T, \text{"NPtoP"}, P) \in TXN_{ROI,\text{"assembly area"}}$, where $\eta > 2$.

The introduced methodology for $TOPT$ measurement offers the ability to adjust activities' (ACT) starting and ending points earlier recognized by the MA activity recognition sub-component by combining state transaction information from the assembly space context capture. The measurement is introduced based on EXAMINER's case study covering basic MA operation. An actual implementation may require further investigation for any additional activity transition that is not covered by EXAMINER's operation time digitization. By obtaining precise time measurements for each activity, the overall effectiveness of the MA system can be improved, allowing for in-depth analysis and skill evaluation of the operator.

Until now, the information needed to finalize MA has been digitized by introduced sub-components. Including ACT by activity recognition, LO, and $TOPT$ by context capture sub-component and recording of operation time sub-component. For $TRAJ$ and LI, motion capture was already predigitized. A boundary T creates a subsequence $TRAJ$ from a time series $JT_{(x,y)}$. However, the digitization of MA is a semi-automatic process. The expert must add more information that EXAMINER cannot digitize, especially for the expert's template. The data includes the name of each assembly-related tool and part associated with each ROI, the name and description of the ST, and the name of the MA. The expert may also provide a text description of each step and media, such as photos and videos, to create a training medium. Later, the dissertation will refer to an expert's template as a template. After the elaboration of techniques under digitization of operator skill, the dissertation continues with the evaluation of proposed techniques.

5.3 Evaluation

The evaluation section evaluates subcomponents of EXAMINER. The evaluation objectively assesses performance for skill digitization components, including deep learning-based MA activity recognition, assembly space context capture, and step recognition. The dissertation first introduces the details of the experiments, then the evaluation metrics used, and finally, it reports the performance evaluation results of each target component.

5.3.1 Experiment

The experiments subsection aims to provide the experiment details for each skill digitization component separately. It first provides details about hardware and software setup, followed by data collection, and then the specific experiment details on each component.

Hardware and software setup

This study uses the following hardware and software configurations for performing experiments and evaluation. The setup includes resources for training the manual assembly activity recognition model and running the EXAMINER application, highlighting both computational and application-specific environments.

Hardware setup Multiple instances of Google Colab were used to accelerate and parallelize the training of the manual assembly activity recognition models with hyperparameter tuning. Each Google Colab environment provided an Intel Xeon CPU with 2 vCPUs, 13GB of memory, and an NVIDIA T4 GPU with 15GB of memory. A personal PC was used for the EXAMINER application, equipped with an AMD Ryzen 5 3600 CPU, 16GB of memory, and an NVIDIA GeForce RTX 2060 SUPER graphics card. The PC was connected to a professional-grade monitor, and a Logitech Streamcam web camera capable of recording the experiment in full HD 1080p resolution at 60 fps. The total hardware cost for the EXAMINER application is approximately 1,500 United States Dollars per station. This setup was chosen to leverage the high computational power required for deep learning model training and inference while ensuring public accessibility.

Software setup The EXAMINER application ran on a Windows 10 Education operating system. The programming language used was Python 3 due to its extensive support for scientific computing and machine learning. Essential libraries included Keras for deep learning-based MA activity recognition, OpenCV for context recognition tasks, and Tkinter for the graphical user interface. The 2D-skeletal-HPE inference using AlphaPose required PyTorch. For training the deep learning-based MA activity recognition model, Google Colab ran on Ubuntu 20.04 LTS, with Keras-Tuner used for hyperparameter optimization.

Following the hardware and software setup for the experiment, the subsection continues addressing the data collection method.

Data collection

As no publicly available MA dataset can be used to evaluate the digitization components, this research collected the data manually. Data collection was performed in a simulated semi-virtual environment. The environment is a laboratory environment that mimics the essentials of an actual environment. Here, the simulated MA standing single-cell setup consists of a table, assembly area, parts/tools area, and a submission area. The cell has a vision-based EXAMINER for recording the participant assembly iteration. The simulated MA scenario was employed due to COVID-19 when this experiment was conducted. Hence, the regulation limits the number of participants in this research. The scenario aims to simulate all possible outcomes of the education robot track wheel assembly case study. Here, the experiment was conducted by the paper's co-author with the approval consent to collect the video recording without face. The experiment asked to assemble the robot with a scenario mainly including a perfect assembly iteration and an erroneous iteration. For instance, a person must strictly adhere to instructions that produce a correct assembly or an assembly with one or more intentionally performed cognitive errors. A cognitive error mainly consists of omitting a step, substituting a step, and performing an extra step that is not in the general process. The scenario also included steps performed slower or faster than usual for the motor skills.

For each recording iteration, the participant must assemble five robot track wheels of eight parts and two hand tools for tightening each experiment iteration as visualized in Figure 5.18. The robot assembly comprises eleven steps $ST_{\alpha,\beta}$. Each $ST_{\alpha,\beta}$ is comprised of a series of $PR_{\alpha,\beta,\gamma}$. First, **reach** for an assembly part or hand tool from an ROI_{λ} and **retract** it to the assembly area. Second, an **assembly** or a **tool use** to assemble parts. A final act of reaching to return the tool and place the completed assembly

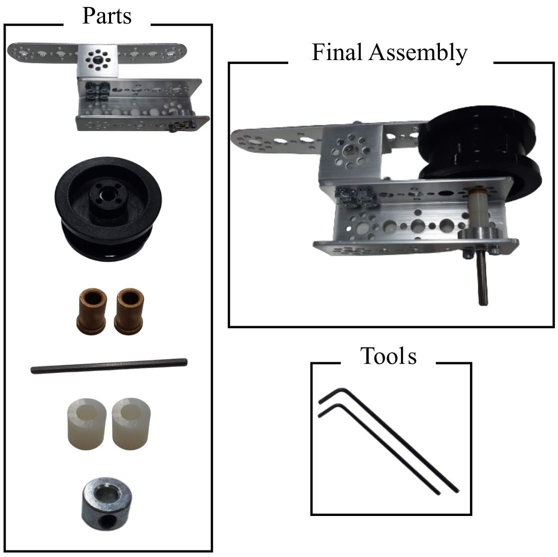


Figure 5.18: Educational robot track wheel with parts and tools.(Not to scale)

in a designated area. Each $ST_{\alpha,\beta}$ may have one or up to four $PR_{\alpha,\beta,\gamma}$ sequence. Considering all of $PR_{\alpha,\beta,\gamma}$, there are thirteen reach activities, ten retract activities, seven assembly activities, and two tool use activities for the assembly of the educational robot. Together with assembly scenarios, the participant executes five correct and 12 incorrect repetitions of assembly.

Automatically providing generated augmented feedback based on comparing digitized operator skills is a feature of EXAMINER's. Hence, three possible scenarios arise from the simulated assembly scenario, including a perfect performance in which the participant correctly executes a strictly $ST_{\alpha,\beta}$ sequence under a desirable $RTOPT_{\alpha,\beta,\gamma}$ with $dist_{\alpha,\beta,\gamma}(OID_{\varphi+1}, OID_{\varphi}) \approx 0$, and the other two scenarios of erroneous assemblies are further categorized as follows.

- The assembly with an incorrect $ST_{\alpha,\beta}$ sequence, the participant is requested to performs iterations of MA_{α}, with the following scenario:
  - Intentionally missing a $ST_{\alpha,\beta}$,
  - Intentionally inserting $ST_{\alpha,\beta} \setminus MA_{\alpha}$.
  - Intentionally substituting $ST_{\alpha,\beta}$ with $ST_{\alpha,\beta'}$, and

  By requesting the participant to perform the mentioned incorrect $ST_{\alpha,\beta}$ sequence iteration, the cognitive performance comparison and augmented feedback for the cognitive skill component of EXAMINER can be further evaluated.

- The assembly with fluctuation in motor performance, the participant is requested to experiment with including:
  - Intentionally performs $RTOPT_{\alpha,\beta,\gamma} < 0$, and
  - Intentionally performs $RTOPT_{\alpha,\beta,\gamma} > 0$.

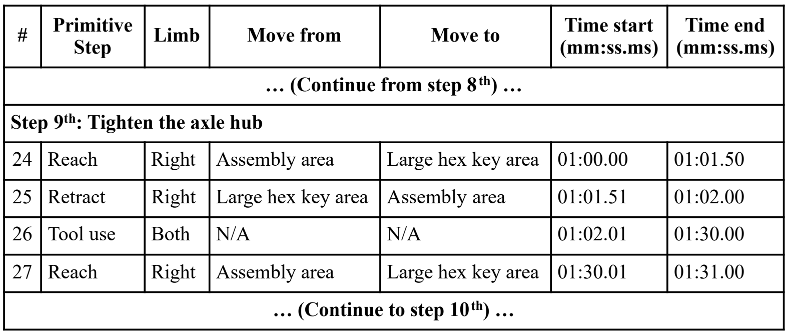
The collected data will be used separately to evaluate each of EXAMINER's digitization components, including assembly activity recognition and assembly space context capture. After the data collection process, the experiment subsection section proceeds with the ground truth annotation of collected data.

Ground truth annotation

Annotation of the ground truth is performed as follows. First, the expert examines the recording frame by frame with the non-linear video editor, using the keyboard's left and right arrow keys to navigate between frames and obtain the frame's time code. The time code is essential for documenting the beginning and ending times of a primitive step. It is generally under the format of Hours:Minutes:Seconds.FramesNumber. Milliseconds is obtained from FramesNumber $\times \frac{1000}{\text{FPS}}$. Second, the expert observes the recording closely and records each primitive action sequentially. The expert must record the following details in the provided recording template:

1. The action of primitive step.
2. The executed limb(s).
3. Movement of the executed limb(s) from the beginnings to the destination. Including any tools or parts employed.
4. Beginning and ending times for each of the primitive steps.

The experiment utilizes the operation time digitization section's definition of the start and end time or boundary of each primitive step. The expert then combines successive primitive steps into a single assembly step. Figure 5.19 represents the final document curated by the expert, in which the recorded information serves as the ground truth for evaluating the EXAMINER's underlying components. The sampled documents show the annotation of the 9^{th}-step in which the operator tightens the axel hub. This step comprises four



#	Primitive Step	Limb	Move from	Move to	Time start (mm:ss.ms)	Time end (mm:ss.ms)
… (Continue from step 8th) …						
Step 9th: Tighten the axle hub						
24	Reach	Right	Assembly area	Large hex key area	01:00.00	01:01.50
25	Retract	Right	Large hex key area	Assembly area	01:01.51	01:02.00
26	Tool use	Both	N/A	N/A	01:02.01	01:30.00
27	Reach	Right	Assembly area	Large hex key area	01:30.01	01:31.00
… (Continue to step 10th) …						

Figure 5.19: Ground truth recording template

primitive steps: reach, retract, tool use, and reach, respectively. The experiment section continues with the deep learning-based MA activity recognition experiment addressing the hyperparameter optimization of deep learning networks.

5.3.2 Basis of Evaluation

First, the dissertation explains basic evaluation metrics and a ranking matrix for multi-class supervised classification problems for balanced and imbalanced data sets. This includes confusion matrix, F-score, ranking matrix, and error measurement as mean squared error(MSE) or mean squared deviation(MSD).

Evaluation Metric

The confusion matrix is one of the most used metrics for the multi-class supervised classification problem. It is a $N \times N$ matrix where N is the number of classes being recognized. The matrix row represents the actual class, while each column is the recognized class. Before constructing a confusion matrix, some definitions must be clarified.

- True Positive (TP): the amount of actual positive class being recognized as positive

- True Negative (TN): the amount of actual negative class being recognized as negative

- False Negative (FN): the amount of actual positive class being recognized as negative

- False Positive (FP): the amount of actual negative class being recognized as positive
- Sensitivity/Recall: or true positive rate, the probability or proportion of a recognizer correctly recognizing positive class proportion to overall recognized classes.

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.14}$$

- Precision/Positive Predicted Value: it is the probability of a recognizer correctly recognizing the class corresponding to the actual occurrence.

$$Precision = \frac{TP}{TP + FP} \tag{5.15}$$

- Specificity: it is the probability or proportion of a recognizer correctly recognizing negative class proportion to overall true negative classes.

$$Specificity = \frac{TN}{TN + FP} \tag{5.16}$$

- Negative Predictive Value (NPV): it is the probability of a recognizer correctly recognizing a negative class corresponding to all negative recognition.

$$NPV = \frac{TN}{TN + FN} \tag{5.17}$$

- Accuracy: it is a proportion of correct recognition relative to all the samples, only used when the class is equally distributed.

$$Accuracy = \frac{TP + TN}{TotalNumberOfSamples} \tag{5.18}$$

Here is an example of $N = 2$, a binary confusion matrix consisting of positive and negative classes. For example, a positive class is reached for recognizing a reach activity, while a negative class is not reached. The confusion matrix is as follows.

The confusion matrix for the N class (C) is as follows.

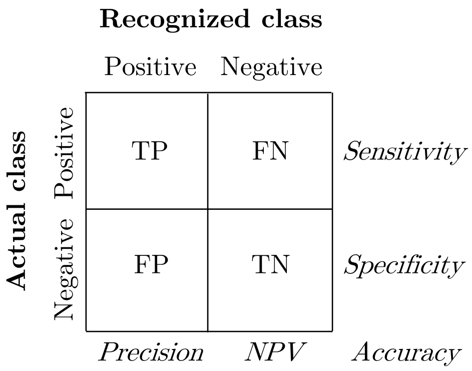


Figure 5.20: Binary classes confusion matrix.

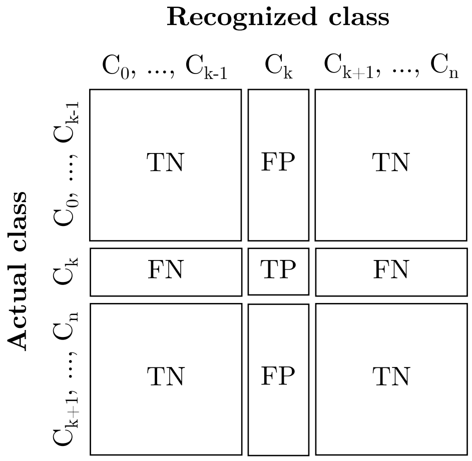


Figure 5.21: Multi-classes confusion matrix.

Performance evaluation for imbalance data

Standard matrix including accuracy (5.18) assumes a balanced class distribution. If the data is imbalanced or skewed, the errors will not be equally treated, resulting in misleading model performance. This subsection introduces some other popular matrices suitable for imbalanced data.

- F-measure or F-score: is the harmonic mean of precision (5.15) and recall (5.14). Precision is the probability of a recognizer correctly recognizing class corresponding to the real occurrence, and recall is the probability or proportion of a recognizer correctly recognizing positive class proportion to overall recognized classes—the F-measure addresses both concerns.

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.19}$$

5.3.3 Evaluation of Skill Digitization

In this subsection, the dissertation presents the performance evaluation of activity recognition and context capture as both sub-components work directly to digitize physical information. The evaluation reports begin with the deep learning-based MA activity recognition.

Deep learning-based MA activity recognition MA activity recognition is designed to identify human activities based on two-dimensional human pose estimation (2D-HPE) inputs. This process involves the application of a trained model that executes inference across each segment using a sliding window approach.

The LSTM model was utilized for manual assembly (MA) activity recognition, with experiments conducted on various models using the manual assembly activity data. Details of the dataset can be found in the appendix under the section "Manual Assembly Activity Dataset." Classes 0, 1, 2, and 3 represent reach, retract, assembly, and tool use, respectively. The fine-tuned LSTM model achieved the highest F-measures, as well as the best accuracy and macro/weighted averages, as shown in Table 5.4. The experiment's performance metrics were based on a test split of 0.2 from the original data. Additionally, the dissertation compares the confusion matrix (Figure 5.22), ROC-AUC as in figure 5.23, and PR-Curve as in figure 5.24 of a highly competitive model to LSTM which is LSTM-CNN.

The experimental results in Table 5.4 and Figure 5.22 provide a detailed comparison of various deep learning models (CNN-TUNED, LSTM-TUNED,

Model	F-Measure				accuracy	macro avg	weighted avg
	0	**1**	**2**	**3**			
CNN-TUNED	0.88	0.86	0.94	0.82	0.9	0.87	0.9
LSTM-TUNED	**0.97**	0.94	**0.97**	**0.9**	**0.96**	**0.95**	**0.96**
CNN-LSTM	0.93	0.91	0.95	0.84	0.93	0.91	0.93
LSTM-CNN	0.95	0.93	**0.97**	**0.9**	0.95	0.94	0.95

Table 5.4: Performance comparison on manual assembly activity recognition on various deep learning based model.

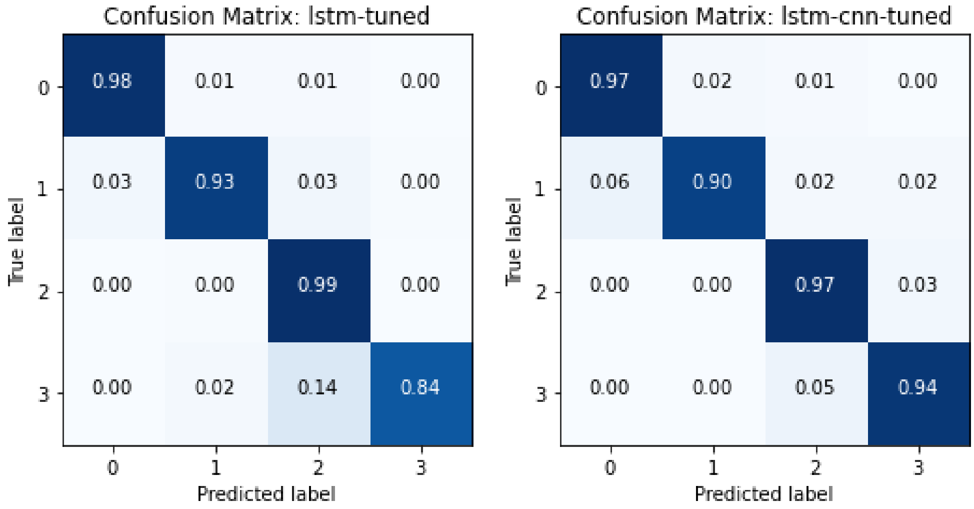


Figure 5.22: The multi-class confusion matrix comparison of LSTM and LSTM-CNN model.

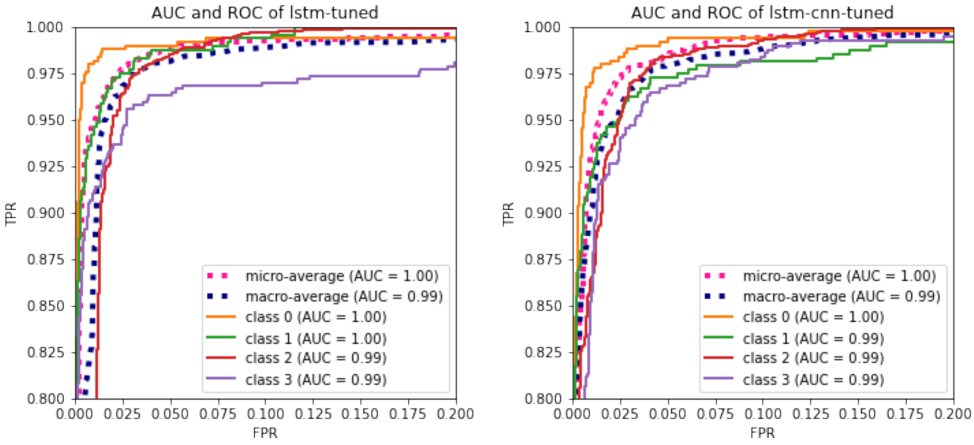


Figure 5.23: The top left corner zoom-in of ROC-AUC comparison of LSTM model and LSTM-CNN model.

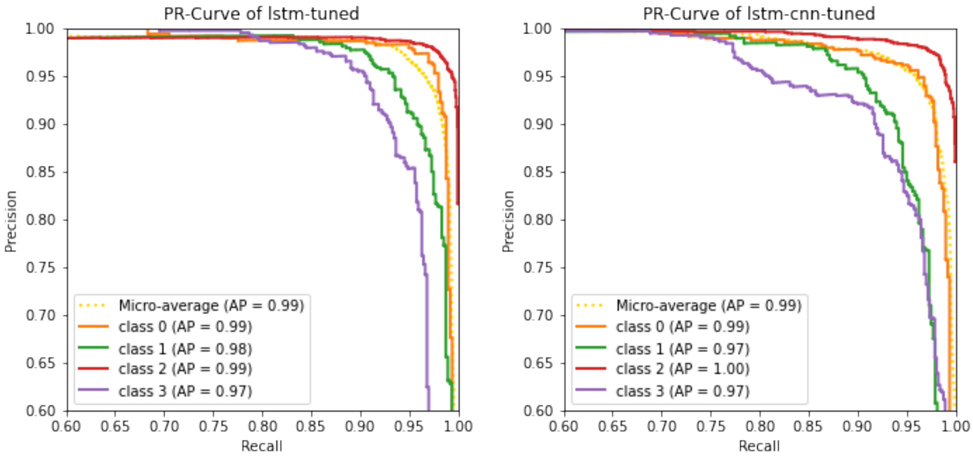


Figure 5.24: The top right corner zoom-in of PR-Curve comparison of LSTM model and LSTM-CNN model.

CNN-LSTM, and LSTM-CNN) for manual assembly activity recognition. The activities are categorized into four classes: 0 (Reach), 1 (Retract), 2 (Assembly), and 3 (Tool Use). The experimental results are discussed further as follows:

- F-Measure
  - **LSTM-TUNED** achieves the highest F-measure across all activity classes with scores of 0.97 (Reach), 0.94 (Retract), 0.97 (Assembly), and 0.9 (Tool Use).
  - **LSTM-CNN** also shows strong performance with F-measures of 0.95 (Reach), 0.93 (Retract), 0.97 (Assembly), and 0.9 (Tool Use), indicating competitive performance but slightly lower than LSTM-TUNED in some classes.
- Accuracy and Averages
  - **LSTM-TUNED** has the highest overall accuracy at 0.96 and the highest macro and weighted averages, both at 0.96.
  - **LSTM-CNN** follows closely with an accuracy of 0.95, a macro average of 0.94, and a weighted average of 0.95.
- Confusion Matrix Analysis Figure 5.22 compares the confusion matrices for LSTM-TUNED and LSTM-CNN models:

  - **LSTM-TUNED** demonstrates high precision with minimal misclassification across all classes. For example, it achieves a high score (0.98) for Class 0 (Reach) and maintains strong performance across other classes.
  - **LSTM-CNN** also shows strong performance but has slightly higher misclassification rates, particularly in Class 1 (Retract), where it scores 0.90 compared to LSTM-TUNED's 0.93.

- Key Takeaways

  1. **LSTM-TUNED** outperforms other models across all metrics, making it the best model for manual assembly activity recognition.
  2. **LSTM-CNN** is a strong competitor but falls slightly short in accuracy and F-measure compared to LSTM-TUNED.
  3. The confusion matrix analysis highlights that LSTM-TUNED has superior precision and lower misclassification rates, particularly in critical classes like Reach and Retract.

Overall, **LSTM-TUNED** is the most effective model for this task, providing high accuracy and reliable performance across different activity classes, thus making it the best choice for manual assembly activity recognition in this experiment.

Context capture The subcomponent effectively identifies frame numbers where an object within the Region of Interest (ROI) transitions from Present to Not Present (*PtoNP*) and Not Present to Present (*NPtoP*). This experiment's ground truth comprised 181 *PtoNP* iterations, 43 *NPtoP* iterations, and 234 non-transitional activities labeled as "none." We employed a direct subtraction method to determine the temporal offset, comparing the frame number identified by the subcomponent with the ground truth. The evaluation includes statistical analysis, a histogram, and a box plot to visualize deviations from the ground truth.

Of the 224 transactions involving either *PtoNP* or *NPtoP*, the subcomponent accurately identified 218 (97%). The average offset was approximately 4.93 frames, with a standard deviation of 6.47 frames. The observed offsets ranged from -5 to 36 frames. Half of the transitions had an offset of 2 frames or less. These statistics indicate a relatively high variability in the model's detection timing, with some instances of early and delayed detections. As shown in Figure 5.25, the histogram illustrates that most temporal offsets cluster near zero, signifying that many detections closely align with

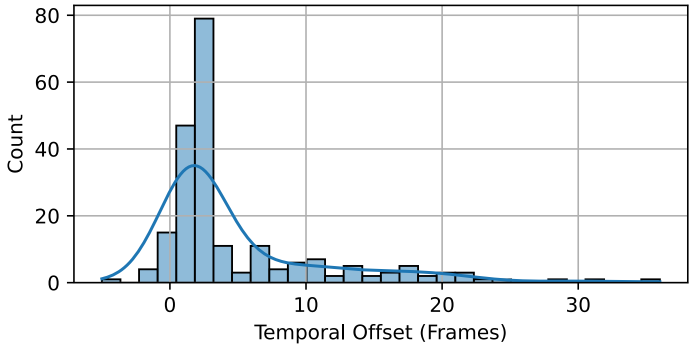


Figure 5.25: Histogram of Temporal Offsets of State Transitions

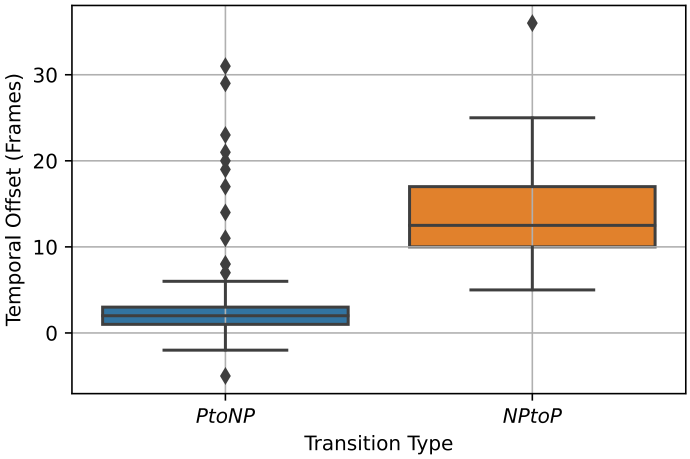


Figure 5.26: Box Plot of Temporal Offsets of State Transitions

the actual transition frames. However, the data spread to the right highlights some delayed detections.

Figure 5.26 further dissects the model's performance by separating the $PtoNP$ and $NPtoP$ transitions. It reveals that $PtoNP$ transitions exhibit a narrow interquartile range (IQR) and a strong central tendency around the median, suggesting that detections for these transitions are generally close to the actual frames. Conversely, the $NPtoP$ transitions display a broader distribution, indicating greater variability in offset.

EXAMINER demonstrates an effective method to detect the presence or absence of an object under ROI using computer vision techniques, primarily through image thresholding. With a 97% success rate in detecting transitions, the model reliably identifies transitional frames, with most detections

Table 5.5: Accuracy of step recognition.

Step	Total Step	Recognized	Accuracy
Pick	11	10	90.91%
Assembly	88	80	91.00%
Tool Use	18	13	72.22%
Submit	13	13	100.00%

occurring proximate to the actual transition moments. This performance suggests that the model can effectively delineate activity boundaries, making it a viable tool for specifying activity transitions in various applications. After obtaining the MA activity and context, the cognitive skill digitization continues with step recognition, showing the error's effects that propagate from the MA activity recognition and assembly space context capture.

Step recognition The process of assembling robot track wheels involves four main steps: pick, assembly, tool use, and submission. These steps are further broken down into primitive steps, as per the EXAMINER data model. Step recognition is an information processing point for the MA activity recognition and context recognition. It transforms them into primitive steps and matches the list of primitive steps to a step.

For the recognition to be deemed accurate, every element in the primitive step list must be correctly identified. Out of 130 ground truth steps, the step recognition system successfully identified 116, yielding an accuracy rate of 89.23%. Table 5.5 outlines a detailed report on the model's performance. Notably, the model exhibits precise recognition of the "Submit" step, and both "Pick" and "Assembly" steps also demonstrate high accuracy levels. However, the accuracy for the "Tool Use" step is lower, primarily due to errors in MA activity recognition that cannot be eliminated.

In the skill digitization evaluation, the EXAMINER system demonstrates strong performance in recognizing left-hand activities and distinguishing complex tasks involving both hands. Although it accurately detects right-hand movements, the precision for "Assembly" tasks could be enhanced. Context capture excels with a 97% accuracy rate in identifying object transitions within the ROI, though some temporal variability is noted. These results affirm the system's robust digitization capabilities with opportunities for further precision improvements. The performance evaluation of components continues to evaluate the skill comparison component.

5.4 Summary of Digitization of Operator Skills

The chapter on ”Digitization of Operator Skills” comprehensively evaluates the EXAMINER system’s ability to convert physical actions and contexts into digital formats for assessment. It covers the implementation and performance of several key components, including MA activity recognition, context capture, and step recognition.

MA Activity Recognition The LSTM-TUNED model is identified as the most effective for recognizing manual assembly activities. It outperforms other models like CNN-TUNED, CNN-LSTM, and LSTM-CNN, achieving the highest F-measures, accuracy, and macro/weighted averages. This model demonstrates excellent precision and low misclassification rates, especially in critical classes like Reach and Retract, making it a robust tool for activity recognition.

Context Capture The context capture component successfully identifies frame transitions within the ROI with a 97% success rate. Temporal offset analysis shows that most detections are closely aligned with actual transitions, though some variability exists. This reliability in detecting activity boundaries and transitions underlines the model’s effectiveness in recognizing the presence and absence of objects within the ROI using computer vision techniques.

Step Recognition The step recognition process is vital for translating MA activity and context data into meaningful steps. The system accurately identifies 89.23% of the steps involved in assembling robot track wheels, showing high precision in recognizing steps like ”Submit,” ”Pick,” and ”Assembly.” Challenges remain with the ”Tool Use” step due to errors in MA activity recognition, indicating areas for improvement.

Feature Pre-Processing To address jitter in skeletal-2d-HPE outputs, the Savitzky-Golay filter is employed to smooth noisy inference results. This post-processing technique reduces oscillations in the inferred joint positions, enhancing the accuracy of the data used for training the recognition models. The effectiveness of the filter in mitigating false-positive data points is crucial for maintaining the integrity of the digitized activity data.

Data Labeling Due to the lack of a publicly available MA dataset, data labeling is performed manually. This involves annotating the data with precise time stamps for each primitive action and identifying any null-class activities. The labeled data serves as the foundation for training and evaluating the recognition models, ensuring that the system can accurately differentiate between various MA activities.

Evaluation Metrics The chapter explains the use of evaluation metrics such as the confusion matrix, F-score, accuracy, precision, recall, and specificity. These metrics provide a comprehensive assessment of the models' performance, particularly in handling imbalanced data. The detailed analysis of these metrics ensures a thorough understanding of the models' capabilities and limitations.

In conclusion, the digitization of operator skills in EXAMINER shows strong potential, with the LSTM-TUNED model leading in activity recognition and the context capture component demonstrating high accuracy. Step recognition and feature pre-processing contribute to the system's overall robustness, making EXAMINER a reliable tool for digitizing and evaluating manual assembly skills. Continuous improvement and fine-tuning of these components will enhance the system's precision and effectiveness in various industrial training applications.

Chapter 6

Comparison of Digitized Skill

The digitized skills comparison aims to objectively identify differences in dexterity by comparing cognitive and motor skills between the trainee and the defined expert's template. These skill comparisons are conducted separately, utilizing the previously digitized MA_{α} as input to the skill comparison component. To differentiate the digitized MA between the expert's temple and the trainee, EXAMINER introduces an additional operator identification parameter, denoted as OID, resulting in the modified representation $MA_{\alpha,OID}$.

As $MA_{\alpha,OID}$ consists of hierarchized digitized objects at two levels, with ST_{β} representing the highest level and PR_{γ} representing the lowest level, cognitive ability comparison focuses on ST_{β}. This choice is driven by the need to evaluate the trainee's ability to perform MA in a strict step-by-step sequence. On the other hand, for motor skill comparison, PR_{γ} is chosen as it emphasizes the importance of similarity in performing the movement time and trajectory of each assembly action. The following section comprehensively explains the methodology employed for comparing cognitive skills.

6.1 Cognitive Skill Comparison

For the comparison of cognitive skill, the focus is directed towards the assembly steps represented by ST_{β}. A stakeholder can evaluate how trainees comprehend and execute the assembly process by examining the list of $ST_{\beta} \in MA_{\alpha,OID}$. Differences in the arrangement of ST_{β} can indicate variations in trainees' cognitive abilities, problem-solving abilities, and decision-making processes during the assembly operation. In the case of EXAMINER, any dissimilarity in $MA_{\alpha,OID}$ between the trainee ($OID =$ trainee) and the template ($OID =$ template) is considered an error in the MA operation sequence.

The cognitive skill comparison sub-component not only determines a binary evaluation of pass or fail but also identifies the specific type of error that occurred. EXAMINER aims to report two errors: missing and executing an unexpected step.

Consequently, an element-wise comparison of $ST_\beta \in MA_{\alpha,OID}$ where $OID \in \{\text{trainee}, \text{template}\}$ is inadequate as it only reports either match or unmatched of an input sequence. An edit distance algorithm or Levenshtein distance is chosen as it can fulfill the requirement to report the type of error efficiently [1]. The algorithm measures the minimum number of single-character edits, either insertions, deletions, or substitutions required to transform one string into another. It quantifies the dissimilarity between two character strings by counting the minimum number of operations needed to convert one string into another. The algorithm considers all possible operations and determines the optimal sequence of edits that yields the smallest total cost.

Under the context of EXAMINER, the algorithm is utilized to determine the minimum number of edits required to convert a trainee's MA sequence ($MA_{\alpha,\text{trainee}}$) into an expert's template sequence ($MA_{\alpha,\text{template}}$). Before determining the optimal cost, the algorithm encodes each sequence's $MA_{\alpha,OID}$ as a string so that each ST_β is treated as a single character. Figure 6.1 depicts every kind of edit for each ST_β. After determining the resulting optimal cost, the subcomponent directly stores the distance result by introducing an additional member called sequence edit distance (EDT) to $MA_{\alpha,OID=\text{trainee}}$. As EXAMINER assumes that the trainee must perform each step precisely and in strict order, the resulting edit distance of any trainee with perfect cognitive skill evaluation must be zero. This indicates that no changes were made to transform the trainee sequence to the template sequence. Otherwise, the algorithm enters the second stage of backtracking for the optimal edits path and stores the types of edits $TE_{\alpha,\beta,\gamma}$ including $insertion(ST_\beta)$, $deletion(ST_\beta)$, $substitution(ST_\beta, ST_{\beta'})$, and $matched$ as an additional member for each $ST_{\beta,OID=\text{trainee}}$. The type of edit will later be converted to augmented feedback.

This section compares the trainee's operation step sequences and the template in the context of EXAMINER. Any difference between them is considered an MA operation sequence error. In addition to a pass or fail evaluation, the comparison identifies the type of operation sequence edit, including insertion, deletion, and substitution. The edit distance algorithm (Levenshtein distance) is utilized to identify all possible edits. A resultant edit distance of zero indicates a trainee with perfect cognitive abilities, while other values indicate specific operation errors. However, the presence of ϵ in insertion error can be relaxed with the τ boundary as the operator may

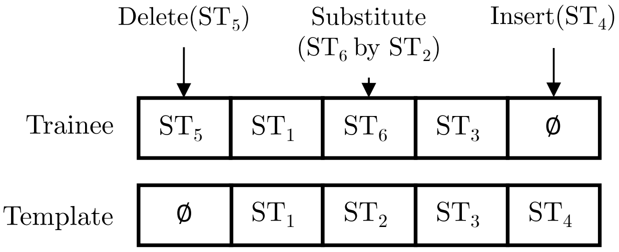


Figure 6.1: The trainee's sequence is optimally edited.

perform the following activity, which is not predefined and labeled including,

1. *Pause motion* - as the operator has to cognitively process the perceived information, environment context, and assembly state. The lesser the pause in time and quantity, the better cognitive performance.

2. *Random motion* - that does not belong in ACT, for instance, scratching or flexing limbs to reduce any discomfort as self-relaxation. The relaxation, however, will introduce the difference in motor skill performance, which the paper will emphasize in the next section.

EXAMINER compares each operator's operation time and motion trajectory in the following section to continue comparing their motor skills.

6.2 Motor Skill Comparison

Motor skill comparison received attention among the research community, especially in sports, dance, rehabilitation, and medical training. In addition to cognitive skills, mastering the motor skill reflects the performance outcomes directly. The introduced papers mainly compared the recorded expert and the trainee at the precise time trajectory.

EXAMINER proposed categorizing MA motor skill assessment into various levels based on the motion precision required to achieve the task. It introduces the different levels of analysis for comparing the digitized motor skill from coarse up to fine motor skills based on the application while maintaining precise operation time measurement. However, a precise time-trajectory comparison is not always required. For instance, an operation of a heavy machine usually requires only the operator's judgment and response to the situation. Hence, measuring the action correctness and response time is needed.

In contrast, a precise task such as a printed circuit board(PCB) manual assembly requires an operator to master soldering skills. As it is a dexterity task, EXAMINER requires precise articulated hand motion and later time measurement. The analysis categorizes a task including snap-fit assembly and hand tool-assisted as the middle of the two. It requires precise time and trajectory similarity measurement as it may explain the difference in operation time.

6.2.1 Operation time comparison

An operation time comparison aims to analyze the differences in operation time for each primitive step (PR_γ) in every individual step (ST_β) in the MA sequence ($MA_{\alpha,OID}$) between the trainee and the template. The operation time of each PR_γ is denoted as $TOPT_{\alpha,\beta,\gamma,OID}$.

EXAMINER determines how the operation time $TOPT$ for each PR_γ in the assembly step ST_β of the trainee differs from that of the template by individually performing a subtraction on each $TOPT$. The difference $\delta_{\alpha,\beta,\gamma}(OID, OID')$ is obtained as follows:

$$\begin{aligned}\delta_{\alpha,\beta,\gamma}(\text{trainee}, \text{template}) = TOPT&_{\alpha,\beta,\gamma,\text{trainee}} \\ - TOPT&_{\alpha,\beta,\gamma,\text{template}}\end{aligned} \tag{6.1}$$

In this context:

- α represents an index to access different sets of assembly steps MA_α.
- β represents the β-th assembly step ST_β.
- γ represents the γ-th primitive step PR_γ within the assembly step ST_β.
- OID and OID' represent unique operator identifications (IDs) for the trainee and the template, respectively.

After determining the difference using Equation (6.1), the subcomponent directly stores the comparison result by introducing an additional member $\delta_{\alpha,\beta,\gamma}(\text{trainee}, \text{template})$ to $PR_{\alpha,\beta,\gamma,OID=\text{trainee}}$. As depicted in Figure 6.2, this operation time comparison is essential for identifying potential performance differences between the trainee and the template, assessing trainee proficiency, and identifying areas where additional training or improvement may be required. The following subsection addresses the motion similarity between the trainee and the template.

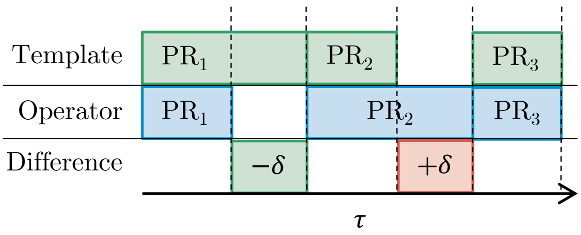


Figure 6.2: In comparison with the template, an operator is either perform the PR faster, slower, or exact(PR_3)

6.2.2 Trajectory comparison

The trajectory comparison in EXAMINER aims to evaluate the motion dissimilarity of individual primitive steps (PR_γ) between the trainee and the template. Given the context of our case study, which involves snap-fit and hand tool-assisted assembly, EXAMINER uses a distance-based comparison method for a demonstration instead of applying motion quantification [114]. This decision is based on the understanding that the trainee's motion trajectory does not need to be a perfect duplicate of the template's but should demonstrate a certain level of motion similarity. The distance-based comparison method provides a practical and effective means to evaluate the alignment of motion trajectories while accounting for variations in human anthropometry and individual motor skills.

Given that α, β, γ is fixed while $OID \in \{\text{trainee}, \text{template}\}$, a selection of distance-based comparison method should consider the following multivariate $TRAJ_{\alpha,\beta,\gamma,OID}$ input characteristics.

1. Variable length of trajectories, the trajectories $|TRAJ_{OID}|$ often differ in length.

2. Differences in human anthropometry and measurement, since different MA operators may have varying physical characteristics and motion patterns due to differences in human anthropometry, resulting in different raw motions trajectory in $\mathbb{R}^3$.

Dynamic Time Warping (DTW) in the trajectory comparison lies in its ability to handle the complexity of the input characteristics $TRAJ_{\alpha,\beta,\gamma,OID}$ when comparing the trainee's trajectory with the expert template. It aligns the trajectories, allowing for comparisons with different time durations. In addition, it accounts for anthropometry variations by warping the trajectories and finding the optimal alignment that minimizes the distance. This ensures that variations in motion due to human differences are considered. DTW

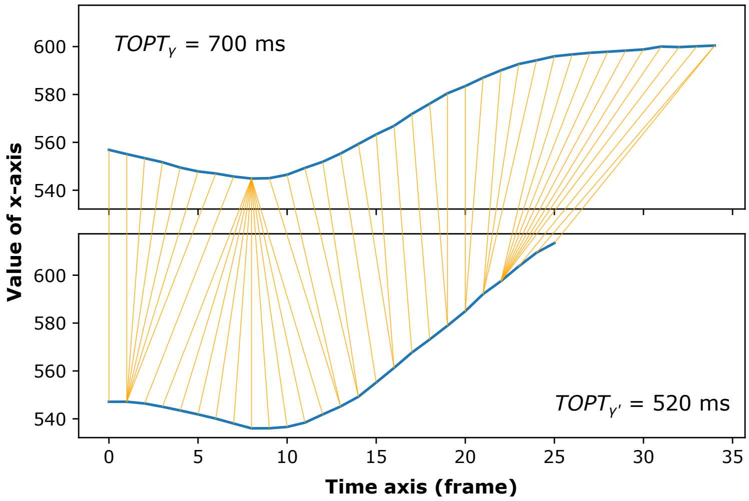


Figure 6.3: DTW of $TRAJ'$ and $TRAJ$.

is a dynamic programming algorithm that can perform optimal matching of two different length time series sequences $x_{1:N}$ and $y_{1:M}$ as visualized in Figure 6.3 [3]. The figures show $TRAJ_{\alpha,\beta,\gamma,OID}$ of wrist movement in the x-axis under $\mathbb{R}^2$ of the fixed $\alpha, \beta, \gamma, OID$ at different primitive operation iterations. Here, the $TRAJ'$ is shorter than $TRAJ$. DTW warps and aligns sequence by finding the best point-to-point match by constructing a cost matrix $D \in \mathbb{R}^{(N+1)\times(M+1)}$. The cost matrix D can be set to $D_{0,0} = 0$, $D_{1:N,0} = \infty$ and $D_{0,1:M} = \infty$. After initialization, the element in D can be populated as follows.

$$D_{i,j} = d(x_i, y_i) + min \begin{cases} D_{i-1,j-1} & \text{(match)} \\ D_{i-1,j} & \text{(insertion)} \\ D_{i,j-1} & \text{(deletion)} \end{cases} \tag{6.2}$$

where $d(x_i, y_i)$ is the distance between points x_i and y_i. The distance can be calculated in its most basic form by taking the difference as $|x_i - y_i|$. The final alignment cost is the sum of the costs along the optimal warping path in D. The cost is typically reported as the distance between two aligned sequences.

Consider an MA motion $TRAJ$ created by moving various limbs, including the wrists, elbows, and other relevant body parts in an assembly space. The comparison of $TRAJ$ involves analyzing the aligned sequences of subjects using specific limbs (e.g., wrists and elbows) in $\mathbb{R}^2$ as constrained by 2D-HPE data. The comparison aims to assess the similarity or dissimilarity between the motion trajectories performed by different subjects during the MA task. The comparison is made independently by dimension and in separate limbs, but it can also be done dimension-dependent [45]. The independent DTW, denoted as DTW_i, calculates the alignment cost separately and then sums it. In contrast, dependent DTW denoted as DTW_d converts $d(x_i, y_i)$ to cumulative squared Euclidean distances to treat all dimensions. As a result, the data from different dimensions must be scaled using a z-score to center the data, making it scale and offset invariant. The usage application determines whether to use DTW_d or DTW_i. For example, multi-dimensional time series pattern matching may benefit from DTW_d. EXAMINER also used this strategy of DTW_d to report the distance as $TRAJ$ is in the form of multi-dimensional time series. The dissimilarity $dist_{\alpha,\beta,\gamma}(OID, OID')$ is obtained as follows:

$$\begin{aligned} dist_{\alpha,\beta,\gamma}(OID, OID') = DTW_d(&TRAJ_{\alpha,\beta,\gamma,\text{trainee}}, \\ &TRAJ_{\alpha,\beta,\gamma,\text{template}}) \end{aligned} \tag{6.3}$$

where (OID = trainee) and (OID' = template). After determining the dissimilarity between the trainee and the template, the subcomponent directly stores the distance result by introducing additional member $dist_{\alpha,\beta,\gamma}(\text{trainee, template})$ to $PR_{\alpha,\beta,\gamma,OID=\text{trainee}}$. By utilizing DTW to address these challenges, the trajectory comparison method can better capture the essence of motion dissimilarity, considering individual differences between the trainee and the template, without requiring strict matching of raw trajectories. This justifies the selection of DTW to compare and evaluate the trainee's performance against the template.

In summary, by utilizing distinct comparison levels for cognitive and motor skills, our methodology ensures a comprehensive and well-justified evaluation of the trainee's cognitive and motor skills. This approach provides a deeper understanding of the underlying factors contributing to skill differences. It enables specific strategies and training programs to improve the overall skills of MA operators. After comparing the digitized trainee skills to the template, the following section of this document provides the trainee and interested parties with the human-interpretable comparison result.

6.3 Evaluation

Evaluating the skill comparison component demonstrates the effectiveness of the proposed comparison method. However, the comparison directly utilizes the information from the digitization component; if the digitization is incorrect, the effect will also be present in the comparison's performance. The subsection addresses the difference in operation sequence.

As the difference in operation sequence utilizes the edit distance to identify the difference, the evaluation is performed as follows. First, the evaluation identifies the ground truth of edit distance. Second, it uses the sequence of activities recognized previously by the skill digitization component as input for the edit distance algorithm. Finally, the resulting edit lists are compared between the ground truth and the inferred result. The result shows a significant discrepancy in the total number of edits, 26 in the ground truth, against 41 in the model. The ground truth consists of 24 inserts and three deletes. In contrast, the model provides 45 inserts, one for each delete and substitution. The model identified all actual "inserts" with over-estimation. It failed to recognize the correct number of "delete" edits and incorrectly introduced a "substitution" operation. The model will likely identify any missing step in the operator's sequence. However, it will also overestimate the missing step and fail to identify some steps that do not belong to the sequence. As a result, one of the noticeable effects of overestimating "inserts" will be exhibited in the feedback-providing component as it is likely to report some of the missing assembly steps even though the trainee may have already performed them.

Evaluating the skill comparison component demonstrates the effectiveness of the proposed comparison method. However, inaccuracies in the digitization component can affect the comparison's performance. The evaluation uses edit distance to identify differences in operation sequences, comparing the ground truth with the model's inferred results. The analysis reveals significant discrepancies in the number of edits, with the model overestimating "inserts" and failing to recognize "deletes" accurately. This overestimation can lead to feedback inaccuracies, potentially reporting missing steps that the trainee has already performed.

6.4 Summary

The chapter on digitized skill comparison aims to objectively identify differences in dexterity by comparing cognitive and motor skills between trainees and expert templates using EXAMINER. The chapter discusses methodologies for comparing cognitive skills through operation sequences and motor skills through motion trajectories. It employs techniques like edit distance for cognitive skill comparison and Dynamic Time Warping (DTW) for motor skill comparison. The evaluation section assesses the performance of these comparison methods, highlighting the effectiveness of the proposed approach while acknowledging the impact of digitization inaccuracies on the results.

6.4.1 Cognitive Skill Comparison

The cognitive skill comparison focuses on evaluating the trainee's ability to follow the exact sequence of assembly steps (ST_{β}). EXAMINER uses the Levenshtein distance algorithm to quantify the dissimilarity between the trainee's sequence and the expert's template. The algorithm reports errors such as insertions, deletions, and substitutions, providing a detailed understanding of cognitive skill discrepancies. A perfect cognitive skill evaluation is indicated by an edit distance of zero.

6.4.2 Motor Skill Comparison

Motor skill comparison assesses the similarity in movement time and trajectory of assembly actions between the trainee and the expert template. EXAMINER categorizes motor skill tasks into various precision levels and employs DTW for trajectory comparison. This method accommodates variations in human anthropometry and motion patterns, ensuring a comprehensive evaluation of motor skills. The comparison considers both coarse and fine motor skills, with tasks like PCB manual assembly requiring precise time and trajectory measurements.

6.4.3 Evaluation

The evaluation section demonstrates the effectiveness of the proposed comparison methods but highlights the dependency on accurate digitization. The analysis reveals discrepancies in edit distance calculations, with the model overestimating insertions and failing to recognize deletions accurately. These inaccuracies can affect feedback, potentially reporting missing steps that the trainee has already performed. Overall, the chapter emphasizes the importance of refining digitization techniques to improve comparison accuracy and enhance training outcomes.

Chapter 7

Automatically Generated Augmented Feedback for Reporting a Training Outcomes

Dexterity plays a significant role in MA's cognitive and motor aspects. To master the MA process, a trainee must recall assembly steps, materials, locations, and tool usage. Trainees rely on their memory to complete the assembly task. In contrast, motor skills emphasize the timing and trajectory of assembly-related actions. Earlier, skill comparisons were conducted, and the trainee's comparison result was stored. However, these results frequently required interpretation by an expert for non-specialists to comprehend them fully.

To address this limitation, our proposed component for providing feedback aims to translate the initial comparison results for cognitive and motor skills. The objective is to present the results in a manner that non-specialists can easily interpret and comprehend, thereby eliminating the need for expert intervention. The feedback component will enable trainees to improve their MA skills independently, fostering continuous skill development and performance improvement by providing a more straightforward and understandable presentation of the comparison results. The feedback differentiates between cognitive and motor abilities. The section first proposes a method to provide augmented feedback for cognitive skills.

7.1 Cognitive Feedback

Augmented feedback plays a crucial role in enhancing cognitive skills during training. The evaluation of cognitive skills focuses primarily on the trainee's ability to accurately recall and reproduce the assembly sequence, including the materials, tools, and methods for each step. Fundamental to cognitive skill evaluation in the context of EXAMINER is the comparison between the trainee's captured and digitized $MA_{\alpha,OID=\text{trainee}}$ and an expert's template. This comparison may result in a perfect match or reveal cognitive errors within ST_β, such as omitting a step, performing an unexpected step, or substituting a step.

EXAMINER uses $TE_{\alpha,\beta}$ stored within ST_β iteratively from $\beta = 1$ to $\beta = N$, with direct mapping through an edit-error mapping process. The edit-error mapping categorizes errors into three classes:

1. $insertion(ST_\beta)$: This edit involves the addition of a missing ST_β to the $MA_{\alpha,\text{trainee}}$.

2. $deletion(ST_\beta)$: This edit removes an unexpected ST_β from the $MA_{\alpha,\text{trainee}}$.

3. $substitution(ST_\beta, ST_{\beta'})$: This edit substitutes an incorrect step with the correct one in $MA_{\alpha,\text{trainee}}$.

The mapping outcome as semantic feedback is stored as an ordered list and will be directly reported to the trainee. EXAMINER only reports the first cognitive error detected during the comparison, as the subsequent cognitive error might propagate from the first error. In addition, the system acknowledges that it can occasionally misinterpret activities, allowing the trainee to mark the incorrect cognitive error determined by the system as correct and dismiss it. This prompts the system to re-evaluate the sequence's edit distance by marking the corresponding $TE_{\alpha,\beta}$ as *matched* and, if available, presenting the first cognitive error available after the re-evaluation.

The recent edit-error mapping is an example of semantic feedback being provided as terminal feedback at the end of the training iteration. It is an augmented feedback technique that generates human-readable text for the trainee. The system continues to report the trainee's motor performance after the cognitive evaluation.

7.2 Motor Feedback

In contrast to cognitive skills, motor skills focus on the elapsed time and correctness of motion to perform the assembly task. Two types of motor skill comparisons, operation time and trajectory comparison, require different methods to provide augmented feedback to the trainee. The subsection begins with feedback on operation time.

7.2.1 Operation time feedback

Operation time feedback aims to provide feedback from a recently compared primitive step operation times, denoted as $\delta_{\alpha,\beta,\gamma}(\text{trainee, template})$. Each record yields one of three possible outcomes, as illustrated in Figure 6.2: "faster" ($-\delta$), "exact (0)," or "slower" ($+\delta$). These outcomes categorize the deviations in operation times of individual primitive steps from the template. For example, the performance of the trainee for $PR_{\alpha,\beta,1}$ was significantly faster than the template. In contrast, the trainee's performance lagged behind the template for $PR_{\alpha,\beta,2}$, indicating room for improvement in executing this particular step. Lastly, the trainee's execution of $PR_{\alpha,\beta,2}$ matched the template, demonstrating outstanding precision and accuracy in the operation time for this primitive step.

The module further converts $\delta_{\alpha,\beta,\gamma}(\text{trainee, template})$ by making it relative to the template $TOPT$. The relative primitive step operating time difference $RTOPT_{\alpha,\beta,\gamma}$ is as follows.

$$RTOPT_{\alpha,\beta,\gamma} = \frac{\delta_{\alpha,\beta,\gamma}(\text{trainee, template})}{TOPT_{\alpha,\beta,\gamma,OID=\text{template}}} \tag{7.1}$$

The calculation is performed iteratively for all β and γ. The expert can then introduce the grading scheme by utilizing a relative difference recently calculated. Table 7.1 provides an example of a grading scheme based on the relative difference.

Table 7.1: Grading scheme based on Relative Difference in Operation Time ($RTOPT_{\alpha,\beta,\gamma}$)

Relative Difference	**Letter Grade**
$RTOPT_{\alpha,\beta,\gamma} < 0$	A+
$RTOPT_{\alpha,\beta,\gamma} = 0$	A
$0 < RTOPT_{\alpha,\beta,\gamma} \leq 0.1$	B
$0.1 < RTOPT_{\alpha,\beta,\gamma} \leq 0.3$	C
$0.3 < RTOPT_{\alpha,\beta,\gamma} \leq 0.5$	D
$RTOPT_{\alpha,\beta,\gamma} > 0.5$	F

The table presents an example of a grading scheme for evaluating the relative difference in operation time ($RTOPT_{\alpha,\beta,\gamma}$) between an expert and a trainee when performing manual assembly tasks. The scheme assigns letter grades to distinguishing characteristics to quantify performance. The letter grades range from A+, awarded to the trainee who demonstrates a faster operation time, to an F for significant deviations from the expert. The grade mapping scheme considers various ranges of relative differences, enabling an objective evaluation of trainee performance relative to the expert standard.

A trainee achieving $RTOPT_{\alpha,\beta,\gamma} < 0$ or A+, indicating a faster completion time of a primitive step than the expert. Even though this $-\delta_{\alpha,\beta,\gamma}(\text{trainee, template})$ performance may appear exceptional, it is essential to consider the potential consequence. Rapid execution may introduce fatigue risk due to increased exertion, compromising precision, and increasing the probability of errors [43]. To ensure that speed does not come at the expense of accuracy and overall task quality, it is necessary to evaluate a trainee's ability to maintain such high efficiency consistently. However, it is essential to note that the article's primary focus is on the training aspect. As such, the article does not go further into the specific implications of prolonged execution at faster rates. The question of sustained performance under such conditions is thus outside the scope of this paper. However, if the proposed system is to be used to track operator performance beyond training, implementers should consider allowance factors such as fatigue and examine deviations from established standards.

The operation time feedback component provides the trainee valuable insight into their performance. It is determined by comparing the operation times for each primitive step to the template. First, the comparison yields three results: "faster," "exact," and "slower," which indicate variances in execution time. These outcomes aid in identifying possibilities for improvement. Later, feedback is introduced so that the comparison results can be further refined by calculating the relative primitive step operating time and assigning it a letter grade. This relative measurement and letter grade mapping provides a more standard and non-expert-friendly interpretation of assessment results. The subsection continues with feedback regarding trajectory similarity.

Trajectory similarity feedback Trajectory similarity feedback aims to interpret the resulting primitive step trajectory dissimilarity calculated by Equation (6.3) from a numerical result and provides a comprehensible evaluation to the trainee. As mentioned earlier, this feedback is additional information to the operation time feedback and is not subject to an assessment.

Previously $dist_{\alpha,\beta,\gamma}(OID, OID')$ is calculated using DTW_d with Eucledian distance. The resulting numerical result is under $\mathbb{R}_{\geq 0}$ and cannot be normalized because the maximum possible distance cannot be reasonably estimated. The number represents the total distance in pixels that the performed sequence deviated from the template. This number alone is not useful for reporting directly to the trainee, requiring an additional step. EXAM-

INER proposes two additional modules to transform the distance data, including the grading scheme and the relative similarity improvement between training iterations.

The grading scheme for trajectory similarity feedback follows a similar implementation idea of operation time feedback; instead of using the relative values, the scheme directly utilizes the output from the Equation (6.3) and matches them with the predefined matching scheme set by the expert. The result from the equation is in the form of $\mathbb{R}_{\geq 0}$, hence requiring the expert intervention of the grading range through their intuition and experiment on a value range for all PR_γ.

The relative similarity improvement between training iterations is introduced based on one of the augmented feedback strategies. Given that OID'_φ identifies the trainee's operation ID, where $\varphi \in \{1, 2, 3, .., n\}$ specifies the training iteration number. Hence, $dist_{\alpha,\beta,\gamma}(OID, OID'_\varphi)$ shorten as $\mathbb{D}_\varphi$ is the dissimilarity between the expert template of the training iteration φ. For each consecutive training iteration, the calculation of relative distance improvement can be performed as follows:

$$RDIMP_{\alpha,\beta,\gamma,OID,\varphi+1} = \frac{\mathbb{D}_{\varphi+1} - \mathbb{D}_\varphi}{\mathbb{D}_\varphi} \tag{7.2}$$

The result of $RDIMP_{\alpha,\beta,\gamma,OID,\varphi+1} < 0$ means that the trainee performs the consecutive $PR_{\alpha,\beta,\gamma}$ more similar to the expert's template than the previous iteration. Otherwise, the trainee does not have an improvement in terms of motion similarity.

The trajectory similarity feedback component provides the trainee additional insight into their motor performance. It is determined by comparing the trajectory for each primitive step to the template. First, the comparison yields numerical distance from the template (dissimilarity), which indicates variances in motion trajectory. Later, it reports the dissimilarity of the trajectory as a grade, and once the trainee performs an additional iteration, the feedback component reports the improvement or deterioration of training outcomes. Besides, the methodology can also be applied to the operation time feedback.

7.3 Summary

This chapter presents a comprehensive methodology for providing automatically generated augmented feedback to report training outcomes. The feedback component translates comparison results of cognitive and motor skills into an easily understandable format for non-specialists, enabling trainees to improve their manual assembly (MA) skills independently. The cognitive feedback component uses an edit-error mapping process to provide human-readable text detailing errors such as insertions, deletions, and substitutions in assembly steps. The motor feedback component offers insights into operation time and trajectory similarity, utilizing grading schemes and relative improvement metrics to standardize and simplify performance evaluations. These feedback mechanisms enhance trainee comprehension and support continuous skill development, contributing to improved proficiency in industrial manual assembly tasks. The chapter concludes with an evaluation of each feedback component, highlighting their effectiveness and areas for further refinement.

Chapter 8

Conclusion, Discussion, Limitation, and Future Work

This chapter provides a comprehensive overview of the conclusions drawn from this research, discusses the implications and contributions of the EXAMINER framework, outlines the limitations encountered during the study, and proposes directions for future research. The aim is to encapsulate the findings and their significance while identifying areas for further exploration and improvement.

8.1 Conclusion and Discussion

By introducing a comprehensive framework and data model, EXAMINER aims to bridge the gap in adopting I-VTS for MA tasks. This model incorporates existing proposals from both industrial and other virtual training systems. Central to the proposed framework is an elementary data structure that governs the entire process, from capturing raw video feeds to providing feedback on training outcomes. Unlike many related studies that offer broader frameworks for realizing an entire factory training environment, our framework specifically addresses industrial manual assembly training [68].

The resulting framework consists of the following components: skill digitization, skill comparison, feedback provider, and multimedia training material. The implementation focuses on the first three components, ensuring their seamless integration. The framework implementation utilized methodologies for skill digitization using a video camera, employing standard and contemporary techniques such as deep learning in computer vision for human pose estimation, recurrent neural networks for activity recognition, and computer vision for contextual sensing. Each underlying subcomponent shows promising performance.

For instance, this paper utilizes a stacked LSTM on an in-house dataset of industrial manual assembly activities, processed into pose coordinates by human pose estimation, to recognize MA activities. The performance is promising, as indicated by an F1 score of over 0.9 in most MA activity classes, comparable to public datasets using equivalent LSTM architecture [107], [117]. Additionally, our model introduces an activity recognition method using human pose estimation data, offering less intrusiveness than methods relying on wearable sensors despite marginally lower performance [92], [100], [116].

Recognizing an MA operation involves more than just identifying activities; it requires integrating contextual information, including the object interaction and the operation's elapsed time. This article introduces a context recognition subcomponent using computer vision techniques, accurately recognizing item state changes within 20-40 milliseconds. The proposed model presents a viable non-deep learning alternative for motion-time studies and activity segmentation [93], [103]. By combining context and activity data, the digitization component can accurately recognize steps, encompassing activities, operation time, and related objects.

The step recognition implementation opted for a predefined algorithm-based matching, achieving over 90% accuracy in assembly steps like 'Pick,' 'Assemble,' and 'Submit.' However, the 'Tool Use' steps lag at 72% accuracy due to activity and context recognition performance limitations.

This study also introduces interpretable augmented feedback for training outcomes. While this concept is heavily utilized in other industries, it is rarely addressed in MA. Some related studies briefly use it in virtual reality training environments, but they often overlook manual assembly's touch and feel aspect [70]. This study thoroughly explores augmented feedback implementation in a physical training setting. Despite errors in the digitization process leading to overestimations in cognitive error feedback, the system reliably reports unperformed steps, demonstrating perfect recall performance.

Moreover, EXAMINER demonstrates that a simple hardware setup comprising a standard video camera and personal computer can effectively digitize MA tasks. This approach simplifies the implementation and makes it economically viable and accessible for small and medium-sized enterprises, promoting broader adoption.

While results are promising, opting for contemporary methods and a transparent data model allows future adaptations and improvements with more advanced models. However, the current framework has limitations. It is evaluated solely on hand-only, non-precise assembly in a single-station setup. Extensions are required to accommodate additional limbs, precise finger motions, and multi-station assembly tasks.

8.2 Contribution

This study introduces a framework in the Industrial Manual Assembly Virtual Training Systems domain, significantly reducing the need for expert involvement and paving the way for autonomous skill development in MA training. Distinctive in its approach, this research offers a comprehensive solution that encompasses the entire spectrum of manual assembly operation digitization and the development of an effective, user-centric training feedback system.

Central to our framework is the innovative digitization of MA operations for experts and trainees. This digitization process is critical because it is the foundation for subsequent skill analysis and comparison between trainees and experts. In analyzing these operations, the study takes a novel approach, employing algorithms such as edit distance and dynamic time warping to identify and quantify skill differences. This methodology enables a more in-depth understanding of cognitive and motor skill differences.

Another contribution of this research is the introduction of MA task data model. This model enhances the framework's adaptability across diverse training scenarios and revolutionizes how information is systematically organized and utilized within I-VTS. The modular design of the framework, emphasizing interconnected yet distinct components, significantly enhances system flexibility and scalability, catering to a wide range of training needs and environments.

The incorporation of augmented feedback is another key aspect of our research. This study bridges the gap between complicated data analysis and actionable training insights, transforming complex skill comparisons into intuitive, easily comprehensible feedback for trainees. This feedback mechanism fosters self-learning and reduces dependency on expert intervention, thereby facilitating a shift towards more independent skill development.

Economically, EXAMINER offers significant benefits by utilizing off-the-shelf components such as a video camera and personal computers, which drastically reduces setup costs. This cost-effectiveness makes the system accessible to small and medium-sized enterprises, promoting wider adoption. Additionally, the flexibility of EXAMINER to operate in various training environments without the need for specialized hardware minimizes financial barriers, further enhancing its economic impact.

In terms of broader societal impact, our framework aligns seamlessly with several Sustainable Development Goals (SDGs), including Quality Education (SDG 4), Economic Growth (SDG 8), and Reduced Inequalities (SDG 10). By improving training methodologies and accessibility, the study contributes significantly to advancing professional skills and educational methodologies while also addressing environmental concerns by minimizing the need for travel in traditional training setups.

In summary, this research offers a comprehensive, flexible, and efficient I-VTS framework, representing a significant leap forward in virtual training systems. The framework utilizes advanced digitization techniques, detailed skill analysis, and user-friendly augmented feedback to address current gaps in MA training and establish a new standard for future developments in the field. The next section addresses the limitations of our current methodology and anticipates investigating these aspects in future research.

8.3 Limitation

While promising, the proposed framework has been primarily evaluated in simulated environments. Real-world testing with diverse user groups and varying contexts is necessary to validate its effectiveness across different scenarios. The current implementation might also face challenges when applied to highly complex assembly environments. The elaboration of each limitation is as follows:

1. **Simulated Environment**: The implementation of the proposed framework has been primarily tested in simulated environments. Real-world validation across diverse scenarios and user groups is essential to ascertain its effectiveness in practical applications. Challenges specific to complex assembly environments need to be explored further.

2. **Limited Dataset**: The in-house dataset was created with a single participant at a location, primarily due to restrictions during the COVID-19 outbreaks. This limitation results in reduced variability, potentially leading to overfitting. Although the study does not aim to develop a new activity recognition model, it demonstrates the feasibility of using the available model with the in-house dataset.

3. **Lack of Real-User Validation**: The absence of real-user validation poses a limitation, impacting the comprehensive evaluation of each component and the overall implementation under real-world conditions. User experience and usability across diverse user demographics have not been fully explored, including adjustments based on user feedback.

8.4 Future Work

The research faces several limitations. Firstly, evaluations primarily in simulated environments necessitate real-world validation for diverse scenarios, assembly environments, and user groups. The dataset's limitation, generated from a single participant due to pandemic constraints, poses challenges related to variability and overfitting. Lack of real-user validation impacts the comprehensive assessment of system components and usability across different users. As of that, the future work of this study is as follows.

- **User-Centric Evaluation**: It is a real-world test involving users from various demographics to assess user experience, system usability, and the effectiveness of the proposed framework in a practical training environment.

- **Long-term User Studies**: It is a measurement of the framework's effectiveness over extended periods. This could reveal insights into the framework's long-term impact on users' skills and performance.

- **Commercial Viability**: The study should explore its commercial viability with the industry partner. For instance, consider scalability, cost-effectiveness, return on investment, and ease of integration into existing infrastructure.

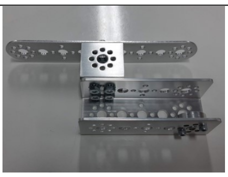
In closing, this study introduces the EXAMINER framework, a comprehensive solution for Industrial Manual Assembly Virtual Training Systems, filling significant gaps in previous research. By automating the digitization of expert and trainee MA operations and providing automated augmented feedback on training outcomes, EXAMINER significantly reduces expert intervention in MA training. While the contributions are significant, several limitations are recognized. Real-world testing is required to validate the effectiveness of the primary evaluation in simulated environments. The limited dataset and lack of real-user validation hamper the system's comprehensive evaluation and usability across various users. To address these limitations, future research will concentrate on User-Centric Evaluation, which will involve a diverse range of users in assessing user experience and system effectiveness in real-world training environments. Long-term User Studies are proposed to determine the framework's efficacy over time, providing insights into the framework's long-term impact on users' skills. Additionally, Commercial Viability exploration with industry partners is advised, with scalability, cost-effectiveness, and ease of integration into existing infrastructure being considered. These initiatives seek to improve the framework's practical applicability and effectiveness.

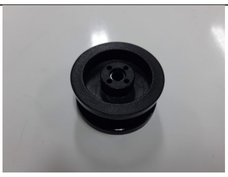
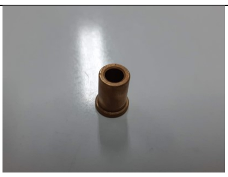
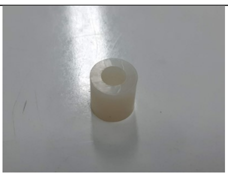
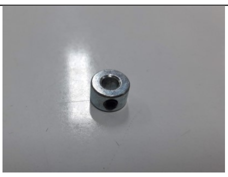
Appendix

Manual Assembly Activity Dataset

The dataset consists of the forward-facing video record of the manual activity performance and the annotation of the activity at the primitive activity level.

Manual activity performance Performers were asked to assemble the TETRIX®MAX Expansion TrackBot step 2.7 and 2.9 [121]. It is an incorporation of Tank Tread Idler Wheel to one of the chassis at the assembly station. The assembly step consists of one partial assembly (chassis), five distinct parts, and two hand tools as in table 8.1. The table shows TrackBot part name, image, amount per assembly, and total amount for five assemblies. The partial assembly consists of 11 steps. However, each step consists of multiple primitive actions and areas of interaction as comprehensively listed in table 8.2. Primitive actions are motor activities related to the manual assembly, including reach and pick, retract, assembly, tool use, reach and pick, and reach and place. Areas can be either parts area, tools area, assembly area, or submission area. An area interaction is when a primitive action is associated in the form of the beginning to the ending location of the primitive action.

Previously assembled parts				
	Name	Image	Amount	Total
1	Partial assembled chassis		1	5

Parts				
	Name	Image	Amount	Total
2	Tank Tread Idler Wheel		1	5
3	11 mm Bronze Bushing		2	10
4	100 mm Axle		1	5
5	Axle Spacer 3/8”		2	10
6	Axle Set Collar		1	5

Tools		
	Name	Image
7	Hex Key Pack	

Table 8.1: The parts and tools for TrackBot Assembly partial assembly of step 2.7 and 2.9

Step	Hand	Primitive action	From area	To area
1: Pick the Partial assembled chassis				
1	left	reach and pick	assembly or submission area	Partial assembled chassis
2	left	retract	Partial assembled chassis	assembly area
2: Incorporate the Tank Tread Idler Wheel to Partial assembled chassis				
3	right	reach and pick	assembly area	Tank Tread Idler Wheel
4	right	retract	Tank Tread Idler Wheel	assembly area
5	any	assembly		
3: Insert the first 11 mm Bronze Bushing to the Partial assembled chassis				
6	right	reach and pick	assembly area	11 mm Bronze Bushing
7	right	retract	11 mm Bronze Bushing	assembly area
8	any	assembly		
4: Insert the second 11 mm Bronze Bushing to the Partial assembled chassis				
9	right	reach and pick	assembly area	11 mm Bronze Bushing
10	right	retract	11 mm Bronze Bushing	assembly area
11	any	assembly		
5: Insert the 100 mm Axle to the Tank Tread Idler Wheel				
12	right	reach and pick	assembly area	100 mm Axle
13	right	retract	100 mm Axle	assembly area
14	any	assembly		
6: Insert the first Axle Spacer 3/8” to the 100 mm Axle				
15	right	reach and pick	assembly area	Axle Spacer 3/8”
16	right	retract	Axle Spacer 3/8”	assembly area
17	any	assembly		
7: Insert the second Axle Spacer 3/8” to the 100 mm Axle				
18	right	reach and pick	assembly area	Axle Spacer 3/8”
19	right	retract	Axle Spacer 3/8”	assembly area
20	any	assembly		
8: Insert the Axle Set Collar to the 100 mm Axle				
21	right	reach and pick	assembly area	Axle Set Collar
22	right	retract	Axle Set Collar	assembly area
23	any	assembly		
9: Tighten Axle Hub				
24	right	reach and pick	assembly area	Hex Key Pack (large)
25	right	retract	Hex Key Pack (large)	assembly area
26	any	use tool		
27	right	reach and return	assembly area	Hex Key Pack (large)
10: Tighten Axle Set Collar				
28	right	reach and pick	Hex Key Pack (large)	Hex Key Pack (small)
29	right	retract	Hex Key Pack (small)	assembly area
30	any	use tool		
31	right	reach and return	assembly area	Hex Key Pack (small)
11: Submit the final assembly				
32	left	reach and place	assembly area	submission area

Table 8.2: The list of eleven sequential assembly step for the assemble of Expansion TrackBot

Publications

- Singhaphandu R, Huynh VN, Pannakkong W, Analysis and Feedback of Movement in Manual Assembly Process, International Conference on Applied Human Factors and Ergonomics (AHFE), Status: in press, pp. 265- 270, 2020 Jul 16, Virtual Conference.

- Singhaphandu R, Huynh VN, Pannakkong W, A Manual Assembly Pro- cess Virtual Training System with Automatically Generated Augmented Feedback [Abstract presentation], The Ninth International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making (IUKM), Abstract book pp. 7, 2022 Mar 18, Online Conference.

- Singhaphandu R, Huynh VN, Pannakkong W, BOONKWAN P, A Manual Assembly Virtual Training System with Comparison of Digitized Operator's Skill for Generating Augmented Feedback, submitted to IEEE Access.

Bibliography

[1] V. I. Levenshtein *et al.*, “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet physics doklady*, Soviet Union, vol. 10, 1966, pp. 707–710.

[2] S. Suzuki *et al.*, “Topological structural analysis of digitized binary images by border following,” *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.

[3] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series.,” in *KDD workshop*, Seattle, WA, USA: vol. 10, 1994, pp. 359–370.

[4] D. Batra and G. M. Marakas, “Conceptual data modelling in theory and practice,” *European Journal of Information Systems*, vol. 4, no. 3, pp. 185–193, 1995.

[5] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, IEEE, 1998, pp. 839–846.

[6] M. Wiesendanger and D. J. Serrien, “Toward a physiological understanding of human dexterity,” *Physiology*, vol. 16, no. 5, pp. 228–233, 2001.

[7] S. Levitch and W. D. Milheim, “Transitioning instructor skills to the virtual classroom,” *Educational Technology*, vol. 43, no. 2, pp. 42–46, 2003.

[8] M. Fowler, *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional, 2004.

[9] S. B. Gokturk, H. Yalcin, and C. Bamji, “A time-of-flight depth sensor-system description, issues and solutions,” in *2004 conference on computer vision and pattern recognition workshop*, IEEE, 2004, pp. 35–35.

[10] D. Minnen, T. Westeyn, T. Starner, J. A. Ward, and P. Lukowicz, "Performance metrics and evaluation issues for continuous activity recognition," *Performance Metrics for Intelligent Systems*, vol. 4, pp. 141–148, 2006.

[11] J.-Y. Chang, G.-L. Chang, C.-J. C. Chien, K.-C. Chung, and A.-T. Hsu, "Effectiveness of two forms of feedback on training of a joint mobilization skill by using a joint translation simulator," *Physical therapy*, vol. 87, no. 4, pp. 418–430, 2007.

[12] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008. DOI: 10.1109/JSSC.2007.914337.

[13] B. Bruegge and A. H. Dutoit, "Object–oriented software engineering. using uml, patterns, and java," *Learning*, vol. 5, no. 6, p. 7, 2009.

[14] C. M. Walsh, S. C. Ling, C. S. Wang, and H. Carnahan, "Concurrent versus terminal feedback: it may be better to wait," *Academic Medicine*, vol. 84, no. 10, S54–S57, 2009.

[15] J. C. Chan, H. Leung, J. K. Tang, and T. Komura, "A virtual reality dance training system using motion capture technology," *IEEE transactions on learning technologies*, vol. 4, no. 2, pp. 187–195, 2010.

[16] H. Ghasemzadeh and R. Jafari, "Coordination analysis of human movements with body sensor networks: A signal processing model to evaluate baseball swings," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 603–610, 2010.

[17] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[18] T. Gutierrez, J. Rodriguez, Y. Velaz, S. Casado, A. Suescun, and E. J. Sanchez, "IMA-VR: a multimodal virtual training system for skills transfer in industrial maintenance and assembly tasks," in *19th International Symposium in Robot and Human Interactive Communication*, IEEE, 2010, pp. 428–433.

[19] A. M. Khan, Y.-K. Lee, S.-Y. Lee, and T.-S. Kim, "Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis," in *2010 5th international conference on future information technology*, IEEE, 2010, pp. 1–6.

[20] A. J. Sarkar, Y.-K. Lee, and S. Lee, "A smoothed naive bayes-based classifier for activity recognition," *IETE Technical Review*, vol. 27, no. 2, pp. 107–119, 2010.

[21] S. Stork and A. Schubö, "Human cognition in manual assembly: Theories and applications," *Advanced Engineering Informatics*, vol. 24, no. 3, pp. 320–328, 2010.

[22] H. Ghasemzadeh and R. Jafari, "Coordination Analysis of Human Movements With Body Sensor Networks: A Signal Processing Model to Evaluate Baseball Swings," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 603–610, 2011.

[23] D. Gorecky, S. F. Worgan, and G. Meixner, "COGNITO: a cognitive assistance and training system for manual tasks in industry," in *Proceedings of the 29th annual European conference on cognitive ergonomics*, 2011, pp. 53–56.

[24] Y.-S. Lee and S.-B. Cho, "Activity recognition using hierarchical hidden markov models on a smartphone with 3D accelerometer," in *International conference on hybrid artificial intelligence systems*, Springer, 2011, pp. 460–467.

[25] S. Kaghyan and H. Sarukhanyan, "Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer," *International Journal of Informatics Models and Analysis (IJIMA), ITHEA International Scientific Society, Bulgaria*, vol. 1, pp. 146–156, 2012.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12, Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105.

[27] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, 2013, pp. 437–442.

[28] R. Sigrist, G. Rauter, R. Riener, and P. Wolf, "Terminal Feedback Outperforms Concurrent Visual, Auditory, and Haptic Feedback in Learning a Complex Rowing-Type Task," *Journal of Motor Behavior*, vol. 45, no. 6, pp. 455–472, 2013.

[29] R. Sigrist, G. Rauter, R. Riener, and P. Wolf, "Augmented visual, auditory, haptic, and multimodal feedback in motor learning: A review," *Psychonomic Bulletin and Review*, vol. 20, pp. 21–53, 1 2013.

[30] K. Swift and J. Booker, *Manufacturing process selection handbook*. Butterworth-Heinemann, 2013.

[31] S. Webel, U. Bockholt, T. Engelke, N. Gavish, M. Olbrich, and C. Preusche, "An augmented reality training platform for assembly and maintenance skills," *Robotics and autonomous systems*, vol. 61, no. 4, pp. 398–403, 2013.

[32] T. Jia, Z. Zhou, and H. Gao, "Depth measurement based on infrared coded structured light," *Journal of Sensors*, vol. 2014, 2014.

[33] C. N. K. Nam, H. J. Kang, and Y. S. Suh, "Golf Swing Motion Tracking Using Inertial Sensors and a Stereo Camera," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 4, pp. 943–952, 2014.

[34] M. Such, C. Ward, W. Hutabarat, and A. Tiwari, "Intelligent composite layup by the application of low cost tracking and projection technologies," *Procedia CIRP*, vol. 25, pp. 122–131, 2014.

[35] Y. Wei, H. Yan, R. Bie, S. Wang, and L. Sun, "Performance monitoring and evaluation in dance teaching with mobile sensing technology," *Personal and ubiquitous computing*, vol. 18, no. 8, pp. 1929–1939, 2014.

[36] M. Zeng, L. T. Nguyen, B. Yu, *et al.*, "Convolutional Neural Networks for human activity recognition using mobile sensors," in *6th International Conference on Mobile Computing, Applications and Services*, 2014, pp. 197–205.

[37] Z. Zhu, V. Branzoi, M. Wolverton, *et al.*, "AR-mentor: Augmented reality based mentoring system," in *2014 IEEE international symposium on mixed and augmented reality (ISMAR)*, IEEE, 2014, pp. 17–22.

[38] G. Bleser, D. Damen, A. Behera, *et al.*, "Cognitive learning, monitoring and assistance of industrial workflows using egocentric sensor networks," *PloS one*, vol. 10, no. 6, e0127769, 2015.

[39] N. Gavish, T. Gutiérrez, S. Webel, *et al.*, "Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks," *Interactive Learning Environments*, vol. 23, no. 6, pp. 778–798, 2015.

[40] P. Hořejšı, "Augmented reality system for virtual training of parts assembly," *Procedia Engineering*, vol. 100, pp. 699–706, 2015.

[41] W. Jiang and Z. Yin, "Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15, Brisbane, Australia: Association for Computing Machinery, 2015, pp. 1307–1310, ISBN: 9781450334594.

[42] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, *Microsoft COCO: Common Objects in Context*, 2015. arXiv: `1405.0312 [cs.CV]`.

[43] N. M. Nur, S. Z. M. Dawal, M. Dahari, and J. Sanusi, "Muscle activity, time to fatigue, and maximum task duration at different levels of production standard time," *Journal of physical therapy science*, vol. 27, no. 7, pp. 2323–2326, 2015.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[45] M. Shokoohi-Yekta, J. Wang, and E. Keogh, "On the non-trivial generalization of dynamic time warping to the multi-dimensional case," in *Proceedings of the 2015 SIAM international conference on data mining*, SIAM, 2015, pp. 289–297.

[46] D. K. Vishwakarma and R. Kapoor, "Hybrid classifier based human activity recognition using the silhouette and cells," *Expert Systems with Applications*, vol. 42, no. 20, pp. 6957–6965, 2015.

[47] G. Westerfield, A. Mitrovic, and M. Billinghurst, "Intelligent augmented reality training for motherboard assembly," *International Journal of Artificial Intelligence in Education*, vol. 25, no. 1, pp. 157–172, 2015.

[48] M. Ziaeefard and R. Bergevin, "Semantic human activity recognition: A literature review," *Pattern Recognition*, vol. 48, no. 8, pp. 2329–2345, 2015, ISSN: 0031-3203.

[49] M. Dalle Mura, G. Dini, and F. Failli, "An integrated environment based on augmented reality and sensing device for manual assembly workstations," *Procedia Cirp*, vol. 41, pp. 340–345, 2016.

[50] M. Funk, J. Heusler, E. Akcay, K. Weiland, and A. Schmidt, "Haptic, auditory, or visual? towards optimal error feedback at manual assembly workplaces," in *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2016, pp. 1–6.

[51] N. Y. Hammerla, S. Halloran, and T. Ploetz, *Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables*, 2016. arXiv: `1604.08880 [cs.LG]`.

[52] A. Langley, G. Lawson, S. Hermawati, *et al.*, "Establishing the usability of a virtual training system for assembly operations within the automotive industry," *Human Factors and Ergonomics in Manufacturing & Service Industries*, vol. 26, no. 6, pp. 667–679, 2016.

[53] W. Liu, D. Anguelov, D. Erhan, *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, Springer, 2016, pp. 21–37.

[54] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*, Springer, 2016, pp. 483–499.

[55] F. J. Ordóñez and D. Roggen, "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition," *Sensors*, vol. 16, no. 1, 2016, ISSN: 1424-8220.

[56] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[57] J.-L. Reyes-Ortiz, L. Oneto, A. Sama, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[58] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235–244, 2016, ISSN: 0957-4174.

[59] Y. Tong, Y. Wang, J. Chen, and C. Chen, "A small scene assistant maintenance system based on optical see-through augmented reality," in *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*, 2016, pp. 155–158.

[60] L. Wang, "Recognition of human activities using continuous autoencoders with wearable sensors," *Sensors*, vol. 16, no. 2, p. 189, 2016.

[61] X. Wang, S. Ong, and A. Y.-C. Nee, "Multi-modal augmented-reality assembly guidance based on bare-hand interface," *Advanced Engineering Informatics*, vol. 30, no. 3, pp. 406–421, 2016.

[62] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[63] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in *CVPR*, 2017.

[64] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *Ieee Access*, vol. 6, pp. 6505–6519, 2017.

[65] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional Multi-person Pose Estimation," in *ICCV*, 2017.

[66] M. Funk, A. Bächler, L. Bächler, T. Kosch, T. Heidenreich, and A. Schmidt, "Working with Augmented Reality? A Long-Term Analysis of In-Situ Instructions at the Assembly Workplace," *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '17*, vol. Part F1285, pp. 222–229, 2017.

[67] M. Gonzalez-Franco, R. Pizarro, J. Cermeron, *et al.*, "Immersive mixed reality for manufacturing training," *Frontiers in Robotics and AI*, vol. 4, p. 3, 2017.

[68] D. Gorecky, M. Khamis, and K. Mura, "Introduction and establishment of virtual training in the factory of the future," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 1, pp. 182–190, 2017.

[69] Y. Guan and T. Plötz, "Ensembles of Deep LSTM Learners for Activity Recognition Using Wearables," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 2, Jun. 2017.

[70] S. Hoedt, A. Claeys, H. Van Landeghem, and J. Cottyn, "The evaluation of an elementary virtual training system for manual assembly," *International Journal of Production Research*, vol. 55, no. 24, pp. 7496–7508, 2017.

[71] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.

[72] J. Liu, J. Mei, X. Zhang, X. Lu, and J. Huang, "Augmented reality-based training system for hand rehabilitation," *Multimedia Tools and Applications*, vol. 76, pp. 1–21, Jul. 2017.

[73] E. Matsas and G.-C. Vosniakos, “Design of a virtual reality training system for human–robot collaboration in manufacturing tasks,” *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 11, no. 2, pp. 139–153, 2017.

[74] K. Nurhanim, I. Elamvazuthi, L. Izhar, and T. Ganesan, “Classification of human activity based on smartphone inertial sensor using support vector machine,” in *2017 ieee 3rd international symposium in robotics and manufacturing automation (roma)*, IEEE, 2017, pp. 1–5.

[75] J. Oyekan, V. Prabhu, A. Tiwari, V. Baskaran, M. Burgess, and R. Mcnally, “Remote real-time collaboration through synchronous exchange of digitised human–workpiece interactions,” *Future Generation Computer Systems*, vol. 67, pp. 83–93, 2017.

[76] V. A. Prabhu, M. Elkington, D. Crowley, A. Tiwari, and C. Ward, “Digitisation of manual composite layup task knowledge using gaming technology,” *Composites Part B: Engineering*, vol. 112, pp. 314–326, 2017.

[77] R. Radkowski and J. Ingebrand, “HoloLens for Assembly Assistance-A Focus Group Report,” in *International Conference on Virtual, Augmented and Mixed Reality*, Springer, 2017, pp. 274–282.

[78] P. Rivera, E. Valarezo, M.-T. Choi, and T.-S. Kim, “Recognition of human hand activities based on a single wrist imu using recurrent neural networks,” *Int. J. Pharma Med. Biol. Sci*, vol. 6, no. 4, pp. 114–118, 2017.

[79] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, “Review of deep convolution neural network in image classification,” in *2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2017, pp. 26–31.

[80] K. Schwab, *The fourth industrial revolution.* Currency, 2017.

[81] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, “Hand Keypoint Detection in Single Images using Multiview Bootstrapping,” in *CVPR*, 2017.

[82] A. Chandran, E. Rahul, and R. R. Bhavani, “Significance of haptic feedback in learning lathe operating skills,” in *2018 IEEE Tenth International Conference on Technology for Education (T4E)*, IEEE, 2018, pp. 97–101.

[83] H. Cho and S. M. Yoon, “Divide and Conquer-Based 1D CNN Human Activity Recognition Using Test Data Sharpening,” *Sensors*, vol. 18, no. 4, 2018, ISSN: 1424-8220.

[84] P. Khaire, P. Kumar, and J. Imran, “Combining CNN streams of RGB-D and skeletal data for human activity recognition,” *Pattern Recognition Letters*, vol. 115, pp. 107–116, 2018, Multimodal Fusion for Pattern Recognition, ISSN: 0167-8655.

[85] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, “CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark,” *arXiv preprint arXiv:1812.00324*, 2018.

[86] M. Murcia-Lopez and A. Steed, “A comparison of virtual and physical training transfer of bimanual assembly tasks,” *IEEE transactions on visualization and computer graphics*, vol. 24, no. 4, pp. 1574–1583, 2018.

[87] S. Werrlich, A. Daniel, A. Ginger, P.-A. Nguyen, and G. Notni, “Comparing HMD-based and paper-based training,” in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2018, pp. 134–142.

[88] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, “Pose Flow: Efficient Online Pose Tracking,” in *BMVC*, 2018.

[89] A. M. Abubakar and İ. Adeshola, “Digital exam and assessments: A riposte to industry 4.0,” in *Handbook of research on faculty development for digital teaching and learning*, IGI Global, 2019, pp. 245–263.

[90] Z. Cao, G. H. Martinez, T. Simon, S. Wei, and Y. A. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[91] Q. Dang, J. Yin, B. Wang, and W. Zheng, “Deep learning based 2d human pose estimation: A survey,” *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 663–676, 2019.

[92] M. Ehatisham-Ul-Haq, A. Javed, M. A. Azam, *et al.*, “Robust human activity recognition using multimodal feature-level fusion,” *IEEE Access*, vol. 7, pp. 60 736–60 751, 2019. DOI: `10.1109/ACCESS.2019.2913393`.

[93] J.-H. Li, L. Tian, H. Wang, Y. An, K. Wang, and L. Yu, “Segmentation and recognition of basic and transitional activities for continuous physical human activity,” *IEEE Access*, vol. 7, pp. 42 565–42 576, 2019. DOI: `10.1109/ACCESS.2019.2905575`.

[94] F. Loch, U. Ziegler, and B. Vogel-Heuser, “Using Real-time Feedback in a Training System for Manual Procedures,” *IFAC-PapersOnLine*, vol. 52, no. 19, pp. 241–246, 2019.

[95] J. J. Roldán, E. Crespo, A. Mart, E. Peña-Tapia, and A. Barrientos, "A training system for Industry 4.0 operators in complex assemblies based on virtual reality and process mining," *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 305–316, May Oct. 2019, ISSN: 07365845.

[96] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM–a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.

[97] A. Subasi, A. Fllatah, K. Alzobidi, T. Brahimi, and A. Sarirete, "Smartphone-based human activity recognition using bagging and boosting," *Procedia Computer Science*, vol. 163, pp. 54–61, 2019.

[98] X. Wan, "Influence of feature scaling on convergence of gradient iterative algorithm," in *Journal of physics: Conference series*, IOP Publishing, vol. 1213, 2019, p. 032 021.

[99] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors," *IEEE Sensors Journal*, vol. 19, no. 17, pp. 7598–7604, 2019.

[100] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "Innohar: A deep neural network for complex human activity recognition," *Ieee Access*, vol. 7, pp. 9893–9902, 2019.

[101] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," *arXiv preprint arXiv:2009.07485*, 2020.

[102] P. Hořejšı, K. Novikov, and M. Šimon, "A smart factory in a Smart City: virtual and augmented reality in a Smart assembly line," *IEEE Access*, vol. 8, pp. 94 330–94 340, 2020.

[103] J. Ji, W. Pannakkong, P. D. Tai, C. Jeenanunta, and J. Buddhakulsomsiri, "Motion time study with convolutional neural network," in *International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making*, Springer, 2020, pp. 249–258.

[104] M. Kitagawa and B. Windsor, *MoCap for artists: workflow and techniques for motion capture.* Routledge, 2020.

[105] S. Mekruksavanich and A. Jitpattanakul, "Smartwatch-based human activity recognition using hybrid lstm network," in *2020 IEEE Sensors*, IEEE, 2020, pp. 1–4.

[106] R. Mutegeki and D. S. Han, "A CNN-LSTM approach to human activity recognition," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, IEEE, 2020, pp. 362–366.

[107] B. N. Raj, A. Subramanian, K. Ravichandran, and D. N. Venkateswaran, "Exploring techniques to improve activity recognition using human pose skeletons," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 2020, pp. 165–172.

[108] V. Sima, I. G. Gheorghe, J. Subić, and D. Nancu, "Influences of the industry 4.0 revolution on the human capital development and consumer behavior: A systematic review," *Sustainability*, vol. 12, no. 10, p. 4035, 2020.

[109] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107 299, 2020.

[110] T. Wang, M. Jin, J. Wang, Y. Wang, and M. Li, "Towards a data-driven method for RGB video-based hand action quality assessment in real time," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 2117–2120.

[111] Z. Wang, X. Bai, S. Zhang, *et al.*, "Information-level real-time AR instruction: a novel dynamic assembly guidance information representation assisting human cognition," *The International Journal of Advanced Manufacturing Technology*, vol. 107, no. 3, pp. 1463–1481, 2020.

[112] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56 855–56 866, 2020.

[113] P. Zawadzki, K. Zywicki, and F. Buń Pawełand Górski, "Employee training in an intelligent factory using virtual reality," *IEEE Access*, vol. 8, pp. 135 110–135 117, 2020.

[114] K. Aouaidjia, B. Sheng, P. Li, J. Kim, and D. D. Feng, "Efficient body motion quantification and similarity evaluation using 3-d joints skeleton coordinates," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 5, pp. 2774–2788, 2021. DOI: `10.1109/TSMC.2019.2916896`.

[115] B. Geslain, G. Bailly, S. D. Haliyo, and C. Duboc, "Visuo-haptic Illusions for Motor Skill Acquisition in Virtual Reality," in *Symposium on Spatial User Interaction*, 2021, pp. 1–9.

[116] J. Male and U. Martinez-Hernandez, “Recognition of human activity and the state of an assembly task using vision and inertial sensor fusion methods,” in *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, IEEE, vol. 1, 2021, pp. 919–924.

[117] H. Ramirez, S. A. Velastin, I. Meza, E. Fabregas, D. Makris, and G. Farias, “Fall detection and activity recognition using human skeleton features,” *IEEE Access*, vol. 9, pp. 33 532–33 542, 2021.

[118] S. Shen, H. Chen, W. Raffe, and T. Leong, “Effects of Level of Immersion on Virtual Training Transfer of Bimanual Assembly Tasks. Front,” *Virtual Real. 2: 597487. doi: 10.3389/frvir*, 2021.

[119] S. Thamm, L. Huebser, T. Adam, *et al.*, “Concept for an augmented intelligence-based quality assurance of assembly tasks in global value networks,” *Procedia CIRP*, vol. 97, pp. 423–428, 2021.

[120] N. M. Tuah, F. Ahmedy, A. Gani, and L. N. Yong, “A survey on gamification for health rehabilitation care: Applications, opportunities, and open challenges,” *Information*, vol. 12, no. 2, p. 91, 2021.

[121] P. Uttley, T. Lankford, B. Eckelberry, and T. McGeorge, *Expansion Set Builder’s Guide*. Pitsco Education, 2021.

[122] J. Peltokorpi, S. Hoedt, T. Colman, K. Rutten, E.-H. Aghezzaf, and J. Cottyn, “Manual assembly learning, disability, and instructions: An industrial experiment,” *International Journal of Production Research*, vol. 61, no. 22, pp. 7903–7921, 2023.