THESIS - EE235401

Classification of Elderly Exercise Activity Based on Pose Estimation Using Deep Learning

AMIK RAFLY AZMI ULYA
NRP 6022221057

SUPERVISOR
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

MASTER PROGRAM
MULTIMEDIA INTELLIGENT NETWORK
DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2024

This page is intentionally left blank

THESIS VALIDATION SHEET

The thesis is written to fulfill one of the requirements for obtaining a degree
of
Magister Teknik (MT)
at
Institut Teknologi Sepuluh Nopember

By:
Amik Rafly Azmi Ulya
NRP: 6022221057

Examination Date : July 8, 2024
Graduation Period : September 2024

Approved by:
Supervisor:

1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng. .....
   NIP:195809161986011001

2. Dr. Eko Mulyanto Yuniarno, S.T., M.T. .....
   NIP:196806011995121009

Examiner:

1. Dr. Diah Puspito Wulandari, S.T., M.Sc. .....
   NIP:198012192005012001

2. Dr. Surya Sumpeno, S.T., M.Sc. .....
   NIP:19696120919970310002

3. Dr. Arief Kurniawan, S.T., M.T. .....
   NIP:119740907200212100l

Head of Teknik Elektro
Fakultas Teknologi Elektro Dan Informatika Cerdas

Dedet Candra Riawan, S.T., M.Eng., Ph.D
NIP:197311192000031001

This page is intentionally left blank

STATEMENT OF THE THESIS AUTHENTICY

I hereby declare that the entire content of my thesis with the title of **Classification of Elderly Exercise Activity Based on Pose Estimation Using Deep Learning** is truly the result of an independent intellectual work, completed without the use of unauthorized materials, and is not the work of another party which I acknowledge as my work.

All references cited or referred to have been written in full in the bibliography. If it turns out that this statement is not true, I am willing to accept sanctions under applicable regulations.

Surabaya July 24, 2024

Amik Rafly Azmi Ulya

Nrp: 6022221057

This page is intentionally left blank

PREFACE

Praise and gratitude to the presence of Allah SWT. for all His grace, the author will able to work this research with the title **Classification of Elderly Exercise Activity Based on Pose Estimation Using Deep Learning**. This research was written as a requirement for completing a master's degree at the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember.
The author also expresses great thanks to:

- Parents and family for their encouragement and support during the author's study.
- Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng. and Dr. Eko Mulyanto Yuniarno, ST. MT. as the supervising lecturer who has provided a lot of advice on this research.
- Nabila Shafa Oktavia who helped the author in technical and non-technical support.
- Friends from the B300 laboratory who always accompany and provide support in various forms of support.
- As well as friends from the class of 2022 who have passed the lecture period together with the author.

Surabaya, July 2024

The Author

This page is intentionally left blank

Classification of Elderly Exercise Activity Based on Pose Estimation Using Deep Learning

By	:	Amik Rafly Azmi Ulya
Student Identity Number	:	6022221057
Supervisor	:	1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
		2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

ABSTRACT

Older people's productivity often declines, especially in terms of physical ability. Physical decline can be slowed down with exercise and other physical activities. However, activities such as stretching are often neglected by the elderly. Exercise activities for the elderly are important so that the elderly can maintain their health in old age. Research on elderly activity recognition has been widely developed. Convolutional Neural Network (CNN) is a type of artificial neural network specifically designed for image processing and recognition. On the other hand, Long Time-Term Memory (LSTM) is an efficient method for solving real-time problems. These two methods can be used for labeling and recognizing physical activity movements in the elderly. Physical activities performed by the elderly need to be customized. In this study, we developed a model with elderly pose estimation. One of the frameworks for human pose estimation is Mediapipe Pose Estimation (MPE). Therefore, this research focuses on the recognition and detection of fitness movements in the elderly. The work begins with the acquisition of datasets in the form of exercise activities that have been adapted to the physical issues of the elderly. Data acquisition was carried out since there are few and limited datasets that discuss this training activity. The acquired video data was then subjected to a video frame extraction process. Each frame sequence represents the exercise activity information. Pose estimation has been done using the Mediapipe framework. The extraction results are then trained using CNN, LSTM, CNN-LSTM, and deep CNN-LSTM architectures. The accuracy of each model is 83.68%, 92.89%, 96.05%, and 87.11%. Based on these results, the CNN-LSTM model outperforms the other models with an accuracy rate of 96.05%. The error in recognizing data patterns is shown using the loss metric. The loss value of the CNN-LSTM model is 0.1498, the smallest compared to other models. This value indicates the model's ability to predict data with the lowest error rate. In addition, in other metrics, this model outperforms other models. Precision, recall, and f1-score of the CNN-LSTM model are at 0.96, respectively.

Keyword: Activity, Deep Learning, Elderly, Pose Estimation

This page is intentionally left blank

Klasifikasi Aktivitas Kebugaran Lansia Berdasarkan Estimasi Pose Menggunakan Deep Learning

Nama Mahasiswa	:	Amik Rafly Azmi Ulya
NRP	:	6022221057
Pembimbing	:	1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
		2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

ABSTRAK

Produktivitas orang tua sering kali menurun, terutama dalam hal kemampuan fisik. Penurunan kemampuan fisik dapat diperlambat dengan olahraga dan aktivitas fisik lainnya. Namun, aktivitas seperti peregangan sering diabaikan oleh orang tua. Aktifitas latihan untuk elderly ini menjadi penting agar elderly dapat menjaga kesehatannya di usia lanjut. Riset mengenai pengenalan aktivitas lansia telah banyak dikembangkan. Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan yang dirancang khusus untuk pengolahan dan pengenalan gambar. Di sisi lain, Long Shoert-Term Memory (LSTM) adalah metode yang efisien untuk memecahkan masalah real-time. Kedua metode ini dapat digunakan untuk pelabelan dan pengenalan gerakan aktivitas fisik pada lansia. Aktivitas fisik yang dilakukan oleh lansia perlu disesuaikan. Dalam penelitian ini, kami mengembangkan sebuah model dengan estimasi pose lansia. Salah satu kerangka kerja untuk estimasi pose manusia adalah Mediapipe Pose Estimation (MPE). Oleh karena itu, penelitian ini berfokus pada pengenalan dan deteksi gerakan kebugaran pada lansia. Pekerjaan diawali dengan akuisisi dataset berupa aktifitas latihan yang telah disesuaikan dengan isu-isu fisik para elderly. Akusisi data dilakukan sejak sedikit dan terbatasnya dataset yang membahas aktifitas latihan ini. Data video yang telah diakuisisi kemudian dilakukan proses ekstraksi video frame. Setiap urutan frame mewakili informasi aktifitas latihan. Estimasi pose telah dilakukan menggunakan framework Mediapipe. Hasil ekstraksi ini kemudian dilatih menggunakan arsitektur CNN, LSTM, CNN-LSTM, dan deep CNN-LSTM. Akurasi setiap model sebesar 83.68%, 92.89%, 96.05%, dan 87.11%. Berdasarkan hasil tersebut, model CNN-LSTM mengungguli model-model lainnya dengan tingkat akurasi 96.05%. Kesalahan dalam mengenali pola data ditunjukkan menggunakan metric loss. Nilai loss model CNN-LSTM sebesar 0.1498, paling kecil dibandingkan dengan model-model lainnya. Nilai ini menunjukkan kemampuan model dalam memprediksi data dengan tingkat kesalahan paling rendah. Selain itu, pada metrcis lainnya, model ini mengungguli daripada model lainnya. Precision, recall, dan f1-score model CNN-LSTM berada pada nilai 0.96, masing-masing.

Kata kunci : Aktivitas, *Deep Learning*, Lansia, Estimasi Pose

This page is intentionally left blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

This page is intentionally left blank

NOMENCLATURE

b	LSTM - bias
Conv1D	Convolutional 1 Dimension
CNN	Convolution Neural Network
C_t	LSTM - Cell State
FC	Fully Connected
FN	False Negative
FP	False Positive
f_t	LSTM - Forget Gate
h_t	LSTM - Hidden Outputs
h_{t-1}	LSTM - Hidden State at t - 1 time
i_t	LSTM - Input Gate
LSTM	Long Short Term Memory
NN	Neural Networks
o_t	LSTM - Output Gate
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Networks
σ	LSTM - Sigmoid Function
tanh	LSTM - tanh Function
TN	True Negative
TP	True Positive
V	Mediapipe - Keypoint Vector
W	Weights
x_t	LSTM - Input at t time
z_{max}	Mediapipe - Maximum Depth
z_{min}	Mediapipe - Minimum Depth

This page is intentionally left blank

CHAPTER 1
INTRODUCTION

1.1 Background

The global elderly population is experiencing unprecedented growth, and this trend is expected to continue into the foreseeable future. As of 2019, there were approximately 703 million individuals aged 65 or older worldwide, accounting for 9% of the total global population [5]. Projections indicate that this number will double by the year 2050, highlighting the increasing importance of addressing the needs of this demographic. One significant challenge faced by the elderly is a decline in productivity, particularly in terms of physical capabilities, which can greatly affect their independence and quality of life.

The decline in physical capabilities among the elderly can be mitigated through regular exercise and physical activity. However, such activities are often neglected by the elderly due to various barriers, including lack of motivation, fear of injury and limited access to professional guidance. Exercise for the elderly requires special considerations, as their physical capacities differ significantly from those of younger, more productive individuals. Consequently, specialized exercise programs designed specifically for the elderly are essential. These programs often necessitate the assistance of physiotherapists or other trained professionals to ensure that exercises are performed safely and effectively.

An elderly person can experience a variety of physical issues. Some physical issues that often affect elders are frozen shoulder [6], tennis elbow [7], and knee pain [8]. Adhesive Copsufitis, often known as frozen shoulder, is a painful condition in which the shoulder joint sustains damage without causing damage to the soft tissues. This type of injury affects 2-5% of the population on average, with women accounting for 60% of all injuries. According to recent research, people with diabetes have a five-fold increased risk of developing

frozen shoulder in their 40s to 60s when compared to the general population. The shoulder joint's wide range of motion and heavy movement loads over it make it more prone to injury. This specific joint is dependent on the Deltoid Major, a large muscle, which is supported by the Cuff Rotators, a smaller set of muscles that are also essential for joint stability and muscular activity [9].

Lateral epicondylitis or better known as tennis elbow is a condition that causes pain in the elbow due to inflammation of the tendons in the upper arm. This pain tends to be constant and causes disability in the elbow, especially the radio humeral joint which is known as lateral epicondylitis or lateral epicondylalgia. The disease often occurs due to repetitive activities involving the wrist and arm. The occurrence of this disease is also in line with a person's age. The disease affects the general population at about 1-3% and increases sharply to 19% in subjects aged 30-60 years. This physical issue is more prone to affect women and lasts longer. The average duration of this physical issue is about 6 months to 24 months [10].

Knee pain is a common knee problem experienced by individuals with routine or excessive activity on the knee. It often affects athletes to the elderly. Knee pain is caused by various conditions and affects various structures around the knee, including bones, muscles, tendons and ligaments. The frequency and severity of knee pain increases when the disease affects the elderly aged 50 years and above [11]. Increased severity of knee pain is associated with more serious problems. There is a greater risk of falls and hip fractures [12].

Ariyani et al. [13] developed a heuristic-based pose detection application system for recognizing elderly activities such as standing, sitting, and lying down. This research successfully leveraged human pose estimation to improve the accuracy of elderly activity recognition. Mediapipe Pose Estimation (MPE), an open-source cross-platform framework provided by Google, is one of the frameworks used for human pose estimation. MPE estimates 2D human joint coordinates in each image frame, utilizing the BlazePose architecture [3]. BlazePose, a lightweight convolutional architecture, is designed for real-

time pose estimation, achieving a frame rate of 10 FPS for the BlazePose Full architecture and 31 FPS for the BlazePose Lite architecture on a single mid-tier phone CPU [14].

Convolutional Neural Networks (CNNs) have proven effective for image processing and recognition tasks, offering high accuracy in recognizing activities from images. CNNs are a type of deep learning model specifically designed to process and analyze visual data. They are composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply filters to the input images, creating feature maps that capture various aspects of the images such as edges, textures, and shapes. Pooling layers then reduce the spatial dimensions of the feature maps, which helps to minimize the computational load and reduce the risk of overfitting. Finally, fully connected layers integrate the extracted features to make predictions. CNNs are particularly effective at recognizing patterns and objects within images, making them ideal for tasks such as image classification, object detection, and activity recognition.

Building on this, Ordóñez et al. [15] combined Deep Convolutional Neural Networks with Long Short-Term Memory (LSTM) networks for human activity recognition. This combination resulted in higher accuracy compared to previous methods. LSTM networks are an advanced form of Recurrent Neural Networks (RNNs), designed to overcome the limitations of traditional RNNs, which often struggle with retaining information over long sequences. LSTMs address this issue by incorporating special units called memory cells, which can store information for extended periods. These memory cells are regulated by three types of gates: the input gate, the forget gate, and the output gate. The input gate controls the addition of new information to the memory cell, the forget gate determines which information should be discarded, and the output gate manages the retrieval of information from the memory cell.

Activity recognition research for the elderly has advanced considerably in recent years. One notable study by Gochoo et al. [16] proposed a Deep

Convolutional Neural Network (DCNN) classification approach to detect basic activities such as Bed_to_Toilet, Eating, Meal_Preparation, and Relaxation. This approach demonstrated the potential of using deep learning techniques for elderly activity recognition. Similarly, Xu et al. [17] introduced a two-stage method for recognizing activities in elderly homes based on random forest and activity similarity, successfully identifying basic activities performed by seniors.

The integration of CNNs and LSTMs leverages the strengths of both architectures: CNNs excel at extracting spatial features from images, while LSTMs are adept at capturing temporal dependencies in sequential data. In the context of human activity recognition, CNNs can process the visual information from image sequences to identify relevant features, such as the posture and movement of individuals. These features are then fed into the LSTM network, which analyzes the temporal relationships between the sequences to recognize and classify different activities. This synergistic approach enhances the overall accuracy of the recognition system, making it more robust and reliable for real-time applications.

Adapting physical activities to the needs of the elderly is crucial to ensure they can maintain their movement routines and keep their body parts active. Simple exercises tailored to the elderly can provide a Tailoring physical activity to the needs of older adults is essential to ensure they can maintain their movement routines and keep their body parts active. Safeguarding their health by ensuring they perform exercise activities correctly is a focus of research that needs to be deepened. Simple exercises tailored to older adults can be a solution, allowing them to perform physical activities safely and effectively. Ensuring that these exercises are performed correctly is critical to preventing injury and increasing the health benefits of physical activity. In order to be able to recognize each exercise activity performed by the elderly, a deep learning model is needed that is able to properly classify each activity performed by the elderly. This is to ensure that the exercise activities performed are correct and reduce the risk of injury., allowing them to engage in physical activity

safely and effectively. Ensuring that these exercises are performed correctly is essential to prevent injuries and promote the health benefits of physical activity.

The focus of this research is on the classification of exercise activities in the elderly. The proposed exercises are tailored to address common physical issues experienced by seniors. The dataset used consists of images representing sequences of each exercise type. Exercise classification is performed using the MediaPipe Pose Estimation (MPE) framework, and the sequences of exercise activities are trained using the CNN-LSTM method. This method enables the labeling and classification of different exercise types. The resulting model will be evaluated for its accuracy in recognizing and classifying elderly exercise activities. Additionally, the research introduces PhysioExercise, a dataset that presents elderly exercise activities based on common physical issues experienced by elderly.

1.2 Formulation of the Problems

Based on the background previously described, the problems in this study can be formulated. Currently, there are limitations on exercise activity datasets for the elderly. The exercise must be adapted to their physical problems. There are three physical issues that are often experienced by the elderly, namely frozen shoulder, tennis elbow, and knee pain. The dataset collection contains exercise activities that can prevent these physical issues. In addition, the classification of these exercises needs to be done using deep learning methods. The utilization of deep learning model in classification task is necessary considering its good capability in classification task.

1.3 Objectives

The objective of this research is to develop a The purpose of this research is to create a dataset that contains training activities in solving some physical issues that are often experienced by the elderly. These physical issues are frozen shoulder, tennis elbow, and knee pain. Each proposed exercise activity is a movement that can prevent the appearance of these physical issues. In

addition, this research develops a model for classification of exercise activities for the elderly using deep learning methods based on the dataset that has been created. We implemented several methods to get outstanding results. Each model that has been generated is then analyzed for its performance to get the model with the most optimal elderly exercise activity classification ability. for the classification of exercise activities for the elderly using deep learning methods. We implement several methods to get an outstanding result. In addition, we created exercise activities for the elderly dataset, where the dataset is adaptable to elderly physical issues.

1.4 Scope and Limitations

In order to focus on the research goal, the limitations need to be adapted. The limitations of the research are:

- The data used is a private dataset that was created in a laboratory and home environment.
- Since we only extract the keypoint of pose estimation, the subjects of the dataset include individuals of various ages.
- The exercise activities used are exercises that are adaptable to certain elderly physical issues.
- The exercise activities contain activity that do not require any equipment.
- Pose estimation is performed using MediaPipe Pose Estimation.
- The model is labeled and classified using the CNN, LSTM, CNN-LSTM, and deep CNN-LSTM architectures.
- The testing of the model exercise activities is done by some performance metrics, i.e., confusion matrix, precision, recall, f1-score, accuracy, and loss metric.

1.5 Contribution

With this research, we aim to develop a classification of exercise activities for the elderly. Presenting datasets that are suitable for the physical problems of the elderly is also one of our research focuses. Specific exercises will make it easier for the elderly to maintain their health. We perform classification in several deep learning methods to show how the performance is generated for each method. We hope this work can be one of the references for improving the health of the elderly.

This page is intentionally left blank

CHAPTER 2

LITERATURE REVIEW

2.1 Elderly

Elderly refers to individuals who are around 65 years of age or older. Socially, people categorized as elderly are those who are 65 years or older and may require medicare or home health services. According to the United Nations (UN), an individual is classified as elderly or an older person when they reach the age of 60 or older. The increasing number of elderly individuals is considered one of the four megatrends that can affect sustainable development [5].

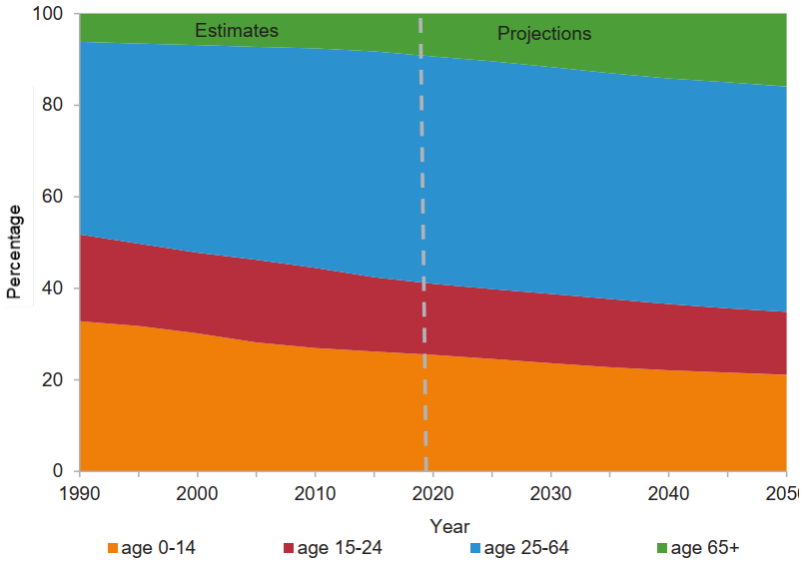

Figure 2.1. Global population by age distribution.

The global population of elderly individuals has been growing rapidly. This trend is predicted to continue growing significantly in the future. According to data from the United Nations (UN) in 2019, the global number of elderly people aged 65 years or older reached 703 million (9% of the total global population). This number is projected to continue rising and double in the next three decades, which means 2050. Fig. 2.1 illustrates the distribution of the global population categorized by age groups. The elderly population

is expected to continue growing as the decline in the working-age population decreases.

As individuals age, they are highly likely to experience various physical and mental changes and conditions, such as decreased mobility, loss of sensory functions, depression, difficulty expressing themselves, and a decline in physical abilities. In some cases, the elderly require assistance in performing daily activities. Accompanying and providing support to the elderly has become a dedicated focus for researchers. Solving the issues related to the safety and daily care of the elderly requires research proposals in activity recognition [17].

2.2 Elderly Physical Issue

The issue of eldelry health has become important since the increase in the number of elderly in the world. Increasing age makes the elderly experience a decrease in organ or tissue function. Diseases that affect the elderly can vary, such as chronic health problems, mobility and balance, decreased sensory function, and physical issues. For physical issues, diseases affect the muscles and bones of the elderly. Strengthening these body parts is important because they are the main support of the body [18]. Some of the physical issues that affect the elderly are explained in this section.

2.2.1 Frozen Shoulder and Exercise Activity

Frozen shoulder, another name for adhesive capsulitis, is a common shoulder condition marked by a progressive rise in pain and stiffness. Women are more likely than males to be impacted, and it usually affects those over 40. The illness develops in three stages: freezing (painful), adhering, and melting. It usually goes away on its own in one to three years [19]. There are two categories of frozen shoulder: primary and secondary. Primary frozen shoulder is caused by the presence of pericapsular attachments, while secondary frozen shoulder is caused by various factors such as sprains, strains, tendinopathy, tendon tears, or bursitis [20].

In the frozen period, strengthening activities such isometric shoulder external rotation, posterior capsule stretch, and scapular retraction might be

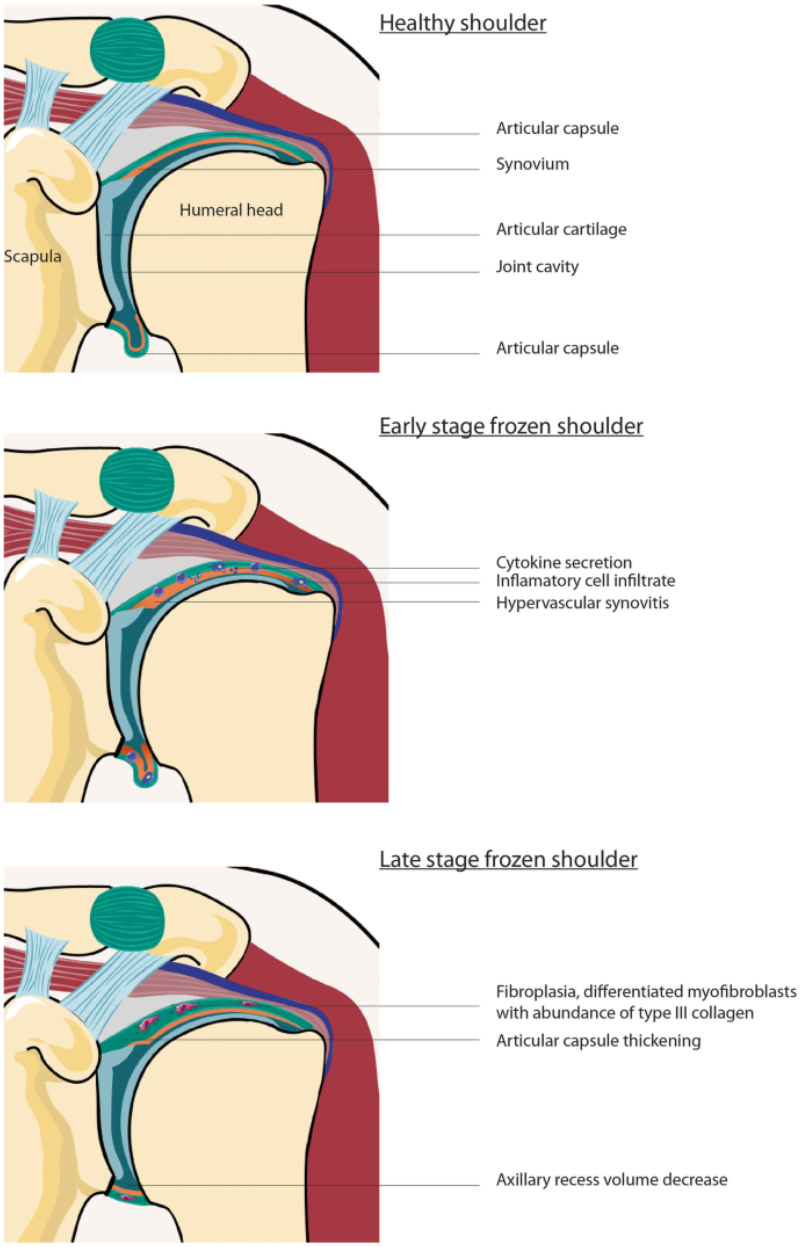

Figure 2.2. Comparison of healthy shoulders with those with frozen shoulder [1].

introduced. Stretching and strengthening exercises can also get more intense during the thawing phase by holding onto the poses for longer periods of time.

Case studies demonstrate how well physical therapy works to improve joint mobility and day-to-day functioning. For instance, a case study published in the Journal of the Formosan Medical Association described how physiotherapy was used to treat a patient with primary frozen shoulder, leading to better joint mobility and ability to do everyday tasks.

These physical issues can be prevented using exercise activities. A wide variety of exercise activities can prevent these physical issues. Exercise activities that move the shoulder are the main focus of the movement. Some of the training activities are shoulder flexion extension [21], adduction abduction [22], and arm circumduction [23].

Shoulder flexion and extension are the two main types of movements performed by the shoulder joint, which involve moving the arm around the body axis. Shoulder flexion is the movement of raising the arm forwards and upwards, away from the body. It involves muscles such as the anterior deltoid, pectoralis major and coracobrachialis. An example of this movement is when an individual lifts the arm straight forward until the arm is above the head. Shoulder extension is the movement of moving the arm behind the body. This movement involves muscles such as the posterior deltoid, latissimus dorsi, and teres major. An example of this movement is when individuals swing their arms backwards while walking or running [24].

Adduction and abduction arm movements are two types of movements that involve the displacement of the arm with respect to the body axis. Adduction is the movement of bringing the arm closer towards the centerline of the body. When the arm moves from a distant position towards the body is called adduction. For example, when an individual brings the arm to the side of the body. While abduction is the movement of moving the arm away from the midline of the body. When the arm moves from a position near the body outwards it is called abduction. For example, when individuals raise their arms to the side away from the body. These two movements in a series are often used in various physical activities and exercises to train certain muscles and increase flexibility and body strength [25].

Arm circumduction is a circular motion of the arm that incorporates several different shoulder joint movements, including flexion, extension, abduction, and adduction. During a circumduction movement, the arm moves in a cone shape, with the apex of the cone at the shoulder joint and the base of

the cone at the hand. Circumduction is often seen in everyday activities such as throwing a ball or swimming freestyle. This movement allows the arm to move in multiple directions and achieve a variety of flexible positions, making it important for many sports and functional activities [23].

2.2.2 Tennis Elbow and Exercise Activity

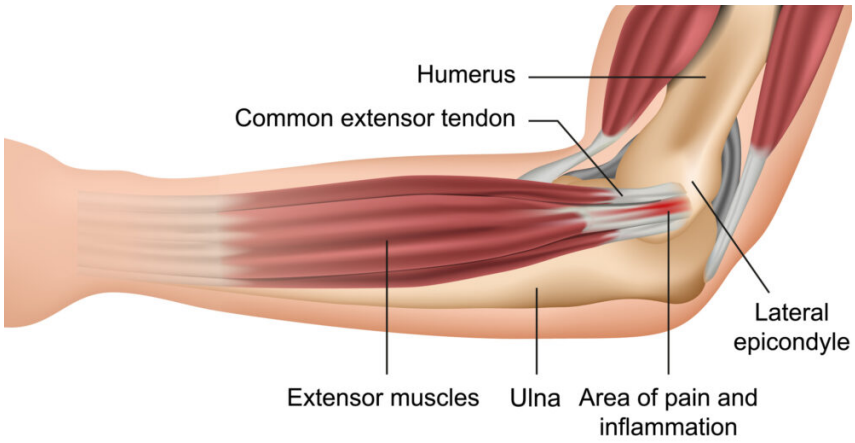

Figure 2.3. Anatomy of tennis elbow. The area of pain and inflammation is shown in the region located around the lateral epicondyle, which is the main characteristic of this condition.

Tennis elbow, often referred to as lateral epicondylitis, is a common musculoskeletal ailment marked by soreness and discomfort in the area of the elbow's common extensor origin. It is more prevalent in the dominant arm and is thought to impact 1% to 3% of adults annually. Though it can also happen as an acute injury (trauma to the lateral elbow), the condition is usually thought to be an overuse ailment requiring repetitive wrist extension against resistance [26].

Tennis elbow is characterized by pain and sensitivity in the area surrounding the lateral epicondyle, pain that gets worse as the middle finger and wrist are resisted, and elbow pain and stiffness. Repetitive forearm and hand motions, as those used in tennis, golf, and other sports, are frequently linked to the syndrome. People who perform repetitive gripping or lifting jobs at work may also experience it.

Case reports highlight the effectiveness of physiotherapy in improving joint

movement and reducing pain. For example, a case series published in the Journal of Orthopaedic and Sports Physical Therapy detailed the use of dry needling as an alternative treatment for tennis elbow, resulting in significant improvements in patient-reported pain and function [27].

One of the training activities that can prevent the physical issue of tennis elbow is elbow flexion extension [28]. The elbow flexion and extension movements are the two main types of movements performed by the elbow joint, which involve the movement of the forearm against the upper arm. Elbow flexion is the movement of bending the elbow so that the forearm comes closer to the upper arm. This movement involves muscles such as the biceps brachii, brachialis, and brachioradialis. Examples of this movement are when individuals bring their hands towards their shoulders or lift weights by bending their elbows. The elbow extension is the movement of straightening the elbow so that the forearm moves away from the upper arm. This movement involves muscles such as the triceps brachii and anconeus. Examples of this movement are when an individual pushes something away from the body or swings the arm into a straight position after bending it.

2.2.3 Knee Pain and Exercise Activity

Knee pain is a common and multifaceted condition that can be caused by various factors, including osteoarthritis, patellofemoral pain, meniscal tears, and other conditions. About 25% of adults exprience knee discomfort, a condition that has become much more common during the previous 20 years [29]. Osteoarthritis, meniscal tears, patellofemoral discomfort, quadriceps or patellar tendinopathy, pes anserine bursitis, and iliotibial band syndrome are among the common causes. Running, crouching, and ascending stairs can all aggravate pain that is localized to particular parts of the knee, such as the anterior, medial, lateral, or posterior regions. A tool for determining the location and pattern of knee pain is the Knee Pain Map, which can be useful in both diagnosing and treating knee pain [30, 31].

There are two approaches in solving knee pain problems, namely ex-

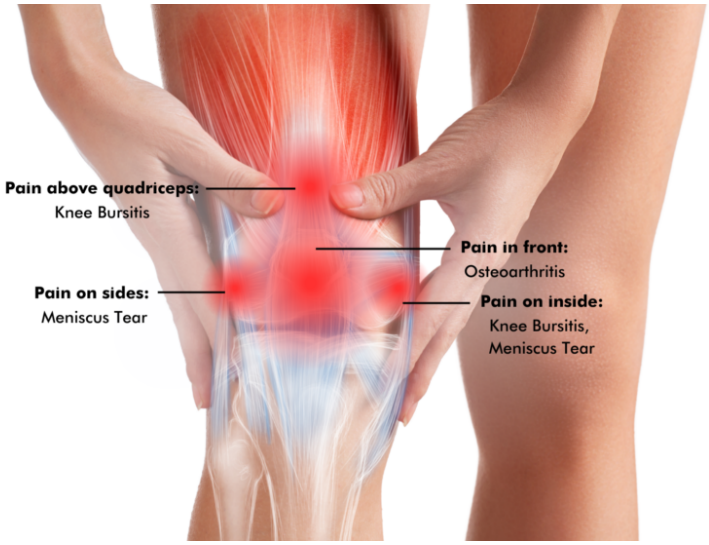

Figure 2.4. Illustration of a set of issues located at the knee.

ercise therapy and radiography. First-line management for osteoarthritis, patellofemoral pain, and meniscal tears often involves exercise therapy, weight loss (if overweight), education, and self-management programs to empower patients to better manage their condition [32]. Radiography may be necessary to further evaluate undifferentiated knee pain, but should be reserved for chronic knee pain of more than six weeks duration or acute traumatic pain in patients who meet specific evidence-based criteria [29].

Knee pain is a frequent issue that has to be evaluated thoroughly. To confirm the diagnosis, a regular history and physical examination as well as imaging and laboratory tests may be necessary. Exercise therapy, weight loss, education, and self-management programs are common forms of treatment. In cases of severe meniscal tears or end-stage osteoarthritis, surgical referral is taken into consideration.

Exercise activities that can prevent this physical issue include leg flexion extension and knee flexion extension. Leg flexion and extension movements are the two main types of movements performed by the knee joint that involve the movement of the leg against the body. In the leg flexion stage, flexion of the knee is the movement of bending the knee so that the heel approaches the buttocks. The muscles involved in this movement include the hamstrings

(biceps femoris, semitendinosus, and semimembranosus). Whereas in leg extension, extension of the knee is the movement of straightening the knee so that the leg returns to a straight position. The muscles involved include the quadriceps (rectus femoris, vastus lateralis, vastus medialis, and vastus intermedius). Leg bending and straightening movements are often performed in walking and running activities [33].

The knee flexion and extension movements are the two main types of movements performed by the knee joint, which involve the movement of the lower leg relative to the upper leg. In the knee flexion stage, the movement involves bending the knee so that the lower leg approaches the buttocks. The main muscles involved in this movement are the hamstrings, which consist of the biceps femoris, semitendinosus and semimembranosus. Movements that are often done daily such as squatting positions. As for knee extension, the movement that occurs is straightening the knee so that the lower leg moves away from the buttocks and returns to a straight position. The main muscles involved in this movement are the quadriceps, which consists of the rectus femoris, vastus lateralis, vastus medialis, and vastus intermedius. Daily activities that involve this movement include standing up from a squatting position [34].

2.3 Human Pose Estimation

Pose estimation is a field of computer vision research that has been widely developed for multiple purposes. Pose estimation involves detecting, associating, and tracking data points on body parts that represent the human body. Pose estimation focuses on estimating the positions of body parts using predefined keypoints. Pose estimation can also be used for tracking human activity. For instance, pose estimation from images or videos in motion is used for healthcare monitoring [35, 36], sports [37, 38, 39], sign language understanding [40, 41, 42], psychology [43, 44, 45], and human gesture control [3, 13]. Generally, human pose estimation consists of two approaches, i.e., the top-down approach and the bottom-up approach.

2.3.1 Top-Down Approach

This method detects the persons first then the landmarks are localized for each person. The localization aims to determine the keypoints. An increase in the number of people indicates a higher level of computational complexity. They perform well on popular benchmarks in terms of accuracy. However, due to the complexity of these models, achieving real-time inference is computationally expensive [46]. The majority of top-down techniques now in use make use of human detector models like High-Resolution Net (HRNet) [47], AlphaPose [48], Feature Pyramid Networks [49], Faster R-CNN [50], and Mask R-CNN [51]. One of the first top-down models to use the Faster R-CNN for the person detection step is proposed by Papandreou et al. [52]. The Cascaded Pyramid Network (CPN), developed by Chen et al. [53], which integrates multi-scale feature maps from various GlobalNet layers with an online hard keypoint mining loss for challenging-to-detect joints.

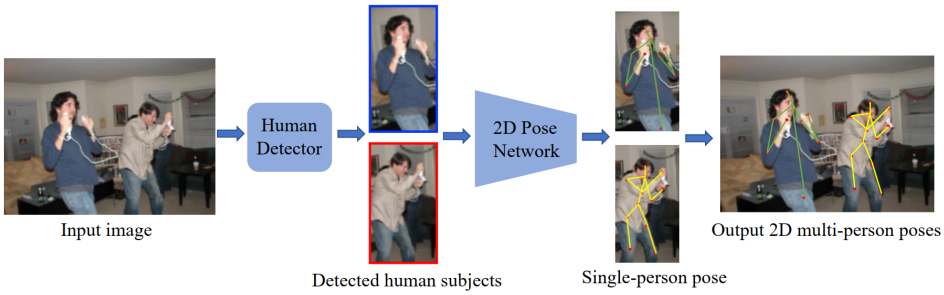

Figure 2.5. Illustration of top-down approach in a 2D human pose estimation task [2].

Figure 2.5 shows an example of a top-down approach in human pose estimation. The process begins with an input image containing one or more people. The input image is then processed by a human detector that detects the presence of human subjects in the image. This human detector marks a bounding box around each detected individual. Once detected, the human subject is clearly identified within a bounding box. Then each individual in the image is processed in a separate state. One example of the model used is the 2D pose network. Each bounding box containing an individual is then fed into a

2D pose network. This network is responsible for estimating human poses so as to produce body keypoint coordinates for each detected subject. The resulting keypoints may vary depending on the deep learning model used. The result of this estimation is the pose of the individual in the form of keypoints connected by lines that indicate the basic skeletal of the body. After each individual has been estimated, the separate images are then reassembled according to the input image. This final image shows the pose estimation results together.

2.3.2 Bottom-Up Approach

In this approach, it finds keypoints of all the persons in an image at once, followed by grouping them into individual persons [54]. A probabilistic map called heatmap is used by these approaches to estimate the probability of every pixel containing a particular landmark (keypoint). With the help of Non-Maximum Suppression, the best landmark is filtered. These are less complex compared to Top-down methods but at the cost of reduced accuracy [46]. The method adopted to group recognized body parts in a picture is where different techniques diverge. Some examples of these approach models are OpenPose [55], PifPaf [56], and single-stage encoder-decoder [57]. To forecast keypoint heatmaps and part affinity fields, which are 2D vectors describing the relationships between joints, OpenPose [55] creates a model with two branches. In the grouping procedure, an affinity field is employed in part. Using the embedding spaces, Pose Partition Networks [58] propose a dense regression method across all the keypoints to construct individual partitions.

Figure 2.6. Illustration of top-down approach in a 2D human pose estimation task [2].

Figure 2.6 shows an example of a bottom-up approach in human pose estimation. For example, the model used in this estimation is a 2D pose

network. The process starts with an input image containing single or multiple individuals. This input image is processed by the 2D pose network. This network is responsible for detecting possible humans in the input image. In the detection process, the network will first detect human body candidates. These points indicate the possible locations of certain body parts depending on the keypoints specified by the model. The next process is human body association. Each of these points will be connected to each other to form the skeleton of a human pose. This process matches the detected body parts into the human skeletal structure of each person in the image. The end of this approach is a multi-person pose. The result is the complete pose of each person in the image, displayed with keypoints connected by lines indicating the skeletal structure.

2.4 MediaPipe Pose Estimation

One of the frameworks for human pose estimation is Mediapipe Pose Estimation (MPE). MPE is an open-source cross-platform framework provided by Google for estimating 2D human joint coordinates in each image frame [59]. MPE builds pipelines that analyzes cognitive data provided as video using machine learning (ML).

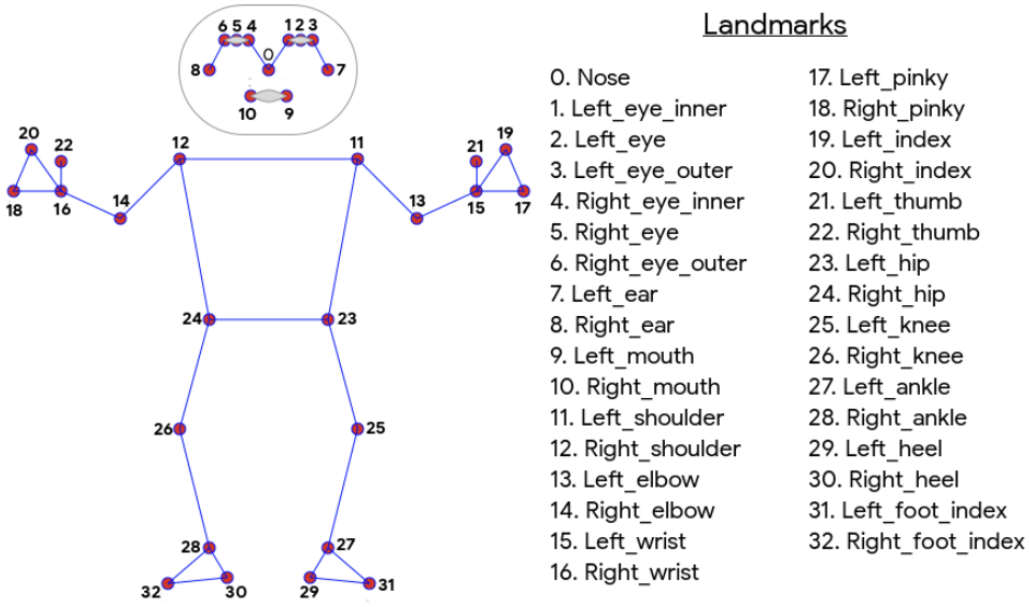

Figure 2.7. BlazePose keypoint topology.

The backbone architecture behind this framework is called BlazePose [3]. Fig.2.8 shows the inference workflow of BlazePose architecture. The pose estimate component of BlazePose architecture predicts the location of all 33 person keypoints. During inference, the architecture adopt a detector-tracker configuration, which displays good real-time performance on a range of applications such as hand landmark prediction [60] and dense face landmark prediction [61]. This pipeline comprises of a lightweight body pose detector followed by a pose tracker network. The tracker predicts keypoint coordinates, the presence of the person on the current frame, and the refined region of interest for the current frame. When the tracker reports that there is no human present, the architecture re-run the detection network on the following frame.

Pose estimation using the BlazePose architecture offers the advantage of producing lightweight models. For low-computational devices, a lightweight model is highly preferred. BlazePose is a lightweight convolutional architecture designed for real-time pose estimation. On a single mid-tier phone CPU, the frame rate for the BlazePose Full architecture is 10 FPS while the BlazePose Lite architecture is 31 FPS [14].

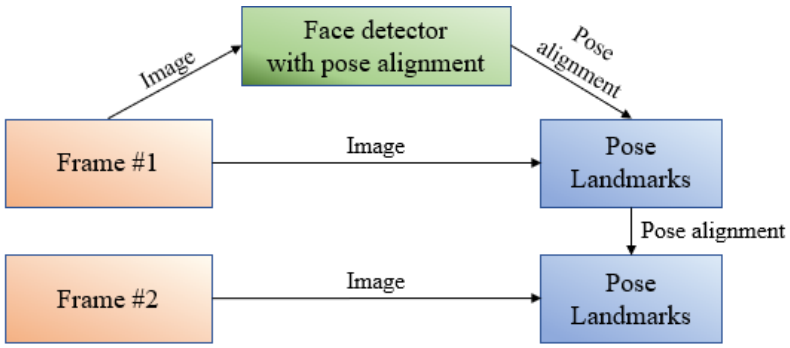

Figure 2.8. Inference workflow of BlazePose architecture [3].

2.5 Deep Learning

Deep learning is a branch of machine learning that uses artificial neural networks to analyze and extract patterns from large and complex data. Artificial neural networks, which are composed of linked layers of artificial

neurons, including input layers, hidden layers, and output layers, are computer models that are modeled after the structure and operation of the human brain. The utilization of numerous hidden layers, which enables the model to learn increasingly complicated data representations, is one of the primary features of deep learning [62].

Deep learning is particularly effective when used on large and diverse datasets, allowing the model to improve accuracy and performance. The learning process in deep learning involves optimizing model parameters using algorithms such as stochastic gradient descent (SGD) and backpropagation techniques to update the weights in the neural network based on the error gradient, allowing the network to learn from errors.

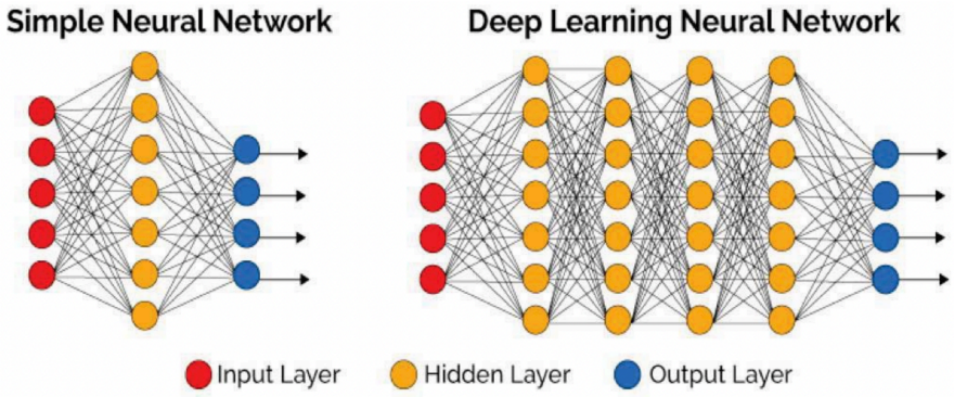

Figure 2.9. Compared between simple neural network architecture and deep learning neural network.

Deep Learning is a type of machine learning that excels in working with unstructured data. It has the capability to process a huge number of characteristics [63]. Deep Learning algorithms process data over numerous levels. Each layer in Deep Learning increasingly extracts features and transmits them to the next layer. The early layers in Deep Learning are responsible for extracting low-level information, while subsequent layers combine various features to build a full representation. Deep Learning is connected to artificial neural networks, which are algorithms based on the structure and function of the brain.

Deep Learning enables computational models with several layers of processing to learn various degrees of abstraction for data representation. The layers in Deep Learning are neural networks with more than three layers of neurons (including input and output layers). More layers and neurons can represent increasingly complicated models, but they also require more time and resources for processing. The depiction of the Deep Learning architecture is presented in Figure 2.9 [64].

Types of deep learning networks include Convolutional Neural Networks (CNN) used in image and video processing, Recurrent Neural Networks (RNN) suitable for sequential data such as text and audio, and Generative Adversarial Networks (GAN) used for data generation. Applications of deep learning include image and speech recognition, natural language processing (NLP), autonomous vehicles, medical data analysis, and recommendation systems.

Deep learning has revolutionized many fields with the ability to learn and generalize from complex and unstructured data, enabling the development of advanced technologies such as virtual assistants and automated medical diagnosis.

2.5.1 Convolutional Neural Network (CNN)

CNN (Convolutional Neural Network) has applications in image and video recognition, recommendation systems, image classification, medical image analysis, natural language processing, and financial time series analysis. Traditional neural network methods do not perform well when it comes to image processing and need breaking down pictures into low-resolution patches. CNN has neurons arranged as portions involved for processing visual input in humans and other. The layers of neurons are designed in such a manner that they span the full visual field to avoid the gradual image processing difficulties of typical neural networks. The sequence of CNN layers includes of input layers, output layers, and hidden layers that contain numerous convolutional layers, pooling layers, and fully connected layers, as depicted in Figure 2.10 [65].

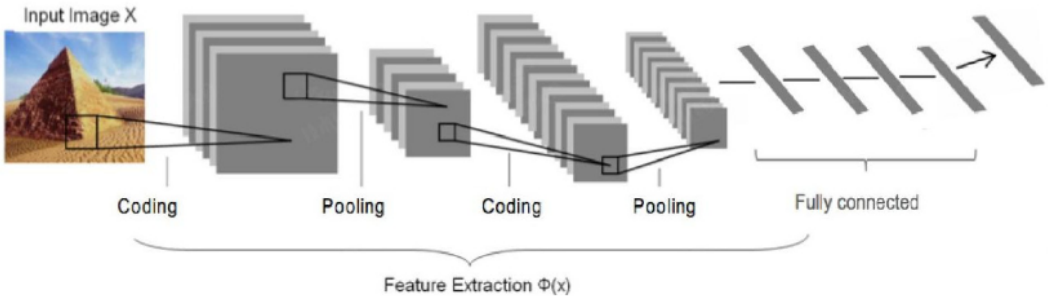

Figure 2.10. An example of CNN architecture for image recognition [4].

Convolution operation is a crucial component of convolutional neural networks. The convolutional layer consists of parameters that comprise a set of learnable filters (kernels). Each filter in this layer is small in width and height but extends through the full depth of the input volume. The commonly used filter sizes are 3x3, 5x5, and 7x7. The third dimension of the filter corresponds to the number of channels in the input. The depth of a grayscale image is 1, while a color image has 3 color channels (RGB).

CNN often employs pooling layers after the convolutional layers. These layers serve to lower the dimensions, often known as subsampling or downsampling. The hyperparameters of the pooling layer are the filter size and stride. The most typically used pooling layer has a filter size of 2 and a stride of 2. There are two main forms of pooling: max pooling, which takes the maximum value, and average pooling, which takes the average value. Max pooling is more often utilized than average pooling.

After several convolutional and pooling layers, CNN typically ends with several fully connected layers. The tensors from the output of these layers are flattened into vectors and then passed through several neural network layers. The dropout regularization technique can be applied to the fully connected layers to prevent overfitting. The final fully connected layer in the architecture contains the same number of output neurons as the number of classes to be recognized in an object detection model.

2.5.2 Long Short-Term Memory (LSTM)

LSTM network is an enhanced special network architecture of Recurrent Neural Network (RNN) [66] which has its major usage in evaluating time series data of many disciplines. LSTM was meant to reduce the drawbacks of RNN such as vanishing gradient because of having short term memory. LSTM is capable of modeling the time series sequential data. A memory unit known as the cell was introduced in the network. As a result, LSTM can overcome the long-range dependency issue of RNN as it can keep the findings for a long-range of time [67].

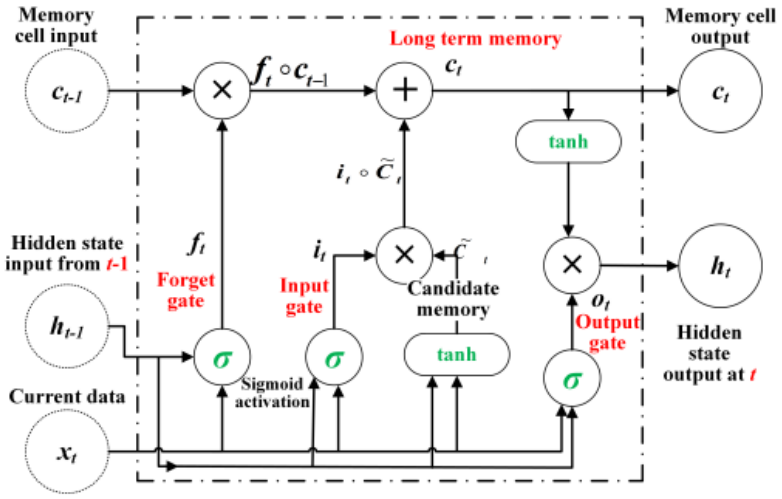

Figure 2.11. Illustration of one cell of the LSTM memory block.

The fundamental LSTM unit is shown in Fig. 2.11, and is made of a cell with an input gate, output gate, and forget gate. LSTMs use the concept of gating to deal with the disappearing or exploding gradient problem [68]. The cell is responsible for remembering values over arbitrary time intervals, and each of the three gates can be thought of as a conventional artificial neuron, computing an activation (using an activation function) of a weighted sum of the current data x_t, a hidden state h_{t-1} from the previous time step, and any bias b. Intuitively, the gates can be considered as regulators of the flow of values through the connections of the LSTM. At each time step they govern which action is done by the cell as stated below. In 2.1 through 2.6, w_i are

the weights associated with each multiplication at gate i, while σ and $tanh$ are possibilities for the activation functions.

Based on Fig. 2.11, the output gate controls the amount to which a new value flows into the cell, known as a write operation:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{2.1}$$

The forget gate performs a similar process, regulating the amount to which the current cell value is preserved, executing a reset operation

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{2.2}$$

The candidate memory cell is updated similarly as:

$$\tilde{C}_t = \tanh(w_c[h_{t-1}, x_t] + b_c) \tag{2.3}$$

and by merging these distinct internal values the internal long-term memory or the next cell memory is formed as

$$c_t = f_t * c_t + i_t * \tilde{C}_t \tag{2.4}$$

From this, the cell output is created by the output gate to regulate the extent to which the value in the cell is utilized to compute the output activation, conducting a read operation:

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{2.5}$$

Lasltly, the cell's hidden output is found as

$$h_t = o_t * \tanh(c_t) \tag{2.6}$$

2.6 Performance Metrics

Performance metrics in deep learning are measures used to evaluate model performance. They offer information on how well the model can forecast new and never-before-seen data. Large datasets are frequently used to train models in deep learning, and performance metrics are used to evaluate the model's performance under various conditions. Here are some commonly used performance metrics in deep learning.

2.6.1 Confusion Matrix

The confusion matrix is a tool used to measure the performance of a classification model by providing a detailed overview of the model's predictions against the test data. It is a box-shaped matrix that shows the number of correct and incorrect predictions made by the model. The number of predictions is based on the actual class and the predicted class. Figure A shows the elements in the confusion matrix, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FP). Through this matrix, various performance metrics can be measured, such as precision, recall, accuracy, and loss.

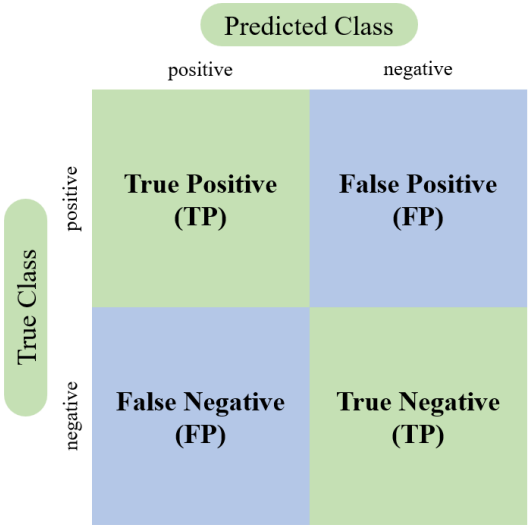

Figure 2.12. A common example of a confusion matrix.

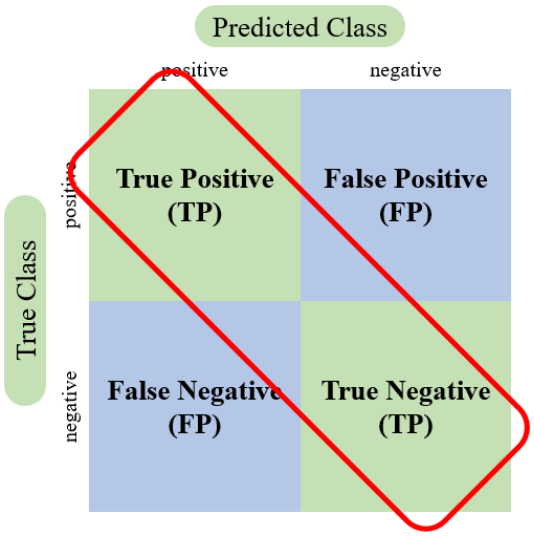

Figure 2.13. Accuracy is calculated as the ratio of the number of correct predictions to the total number of predictions.

2.6.2 Accuracy

Accuracy is an evaluation measure used to assess model performance by calculating the proportion of correct predictions to total predictions. Accuracy shows how well the model performs overall. The accuracy equation is directly proportional to the True Positive (TP) and inversely proportional to the total prediction samples as shown in equation 2.7.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

2.6.3 Loss

Loss is a function used to calculate the error of the deep learning model during the learning process. It is a quantitative measure that describes how well or poorly the model predicts the class of the input data compared to the ground truth. This function gives an indication of how far the model's prediction is from the correct value with the aim of minimizing the loss value during the training process to improve the accuracy and overall performance of the model.

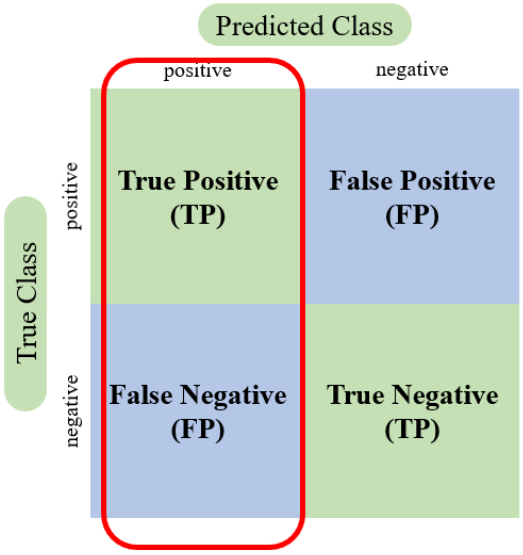

Figure 2.14. Precision is calculated as the ratio of the number of correct positive predictions to the total number of positive predictions made by the model.

2.6.4 Precision

Precision is a metrics that measures the proportion of correct positive predictions out of all predictions. Precision measures how well the model can find True Positive (TP) out of all positive predictions (TP+FP). The higher the precision value, the more positive predictions are predicted to be true. This metric becomes very important when in a situation where the false positive value is high. Precision is formulated in the equation 2.8.

$$precision = \frac{TP}{TP + FP} \tag{2.8}$$

2.6.5 Recall

This metric is used to measure how well the model can detect all positive events among all events that are actually positive. This metric becomes very important when false negatives have a significant probability. Recall is calculated as True Positive (TP) divided by the sum of True Positive and False Negative (TP+FN), according to the equation 2.9.

$$recall = \frac{TP}{TP + FN} \tag{2.9}$$

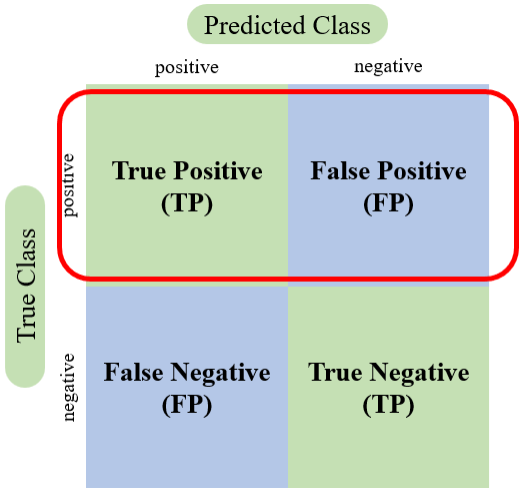

Figure 2.15. Recall is calculated as the ratio of the number of correct positive predictions to the total number of positive examples actually present in the dataset.

2.6.6 F1-Score

Another metric used is F1 score. The F1 score is a metric commonly used to measure the performance of a classification model. It is the harmonic mean of precision and recall. The F1 score combines precision and recall into a single value, providing a balanced measure of the model's performance. It is calculated using the following equation 2.10.

$$F1score = 2 * \frac{precision * recall}{precision + recall} \tag{2.10}$$

This page is intentionally left blank

CHAPTER 3

METHODOLOGY

This section describes our proposed method for pose estimation-based classification of elderly exercise activities using deep learning. We begin our work by determining the types of exercises that the elderly can do to address health issues in old age. One of the challenges of our work is the limited dataset available. So in collecting datasets, we created new datasets according to the predetermined dataset classes. The dataset we have is then subjected to preprocessing and keypoint extraction. The keypoint extraction set is trained using deep learning architecture to produce the desired model. The results of this model are then subjected to evaluation metrics to review our model. Figure 3.1 provides a brief summary of the proposed method.

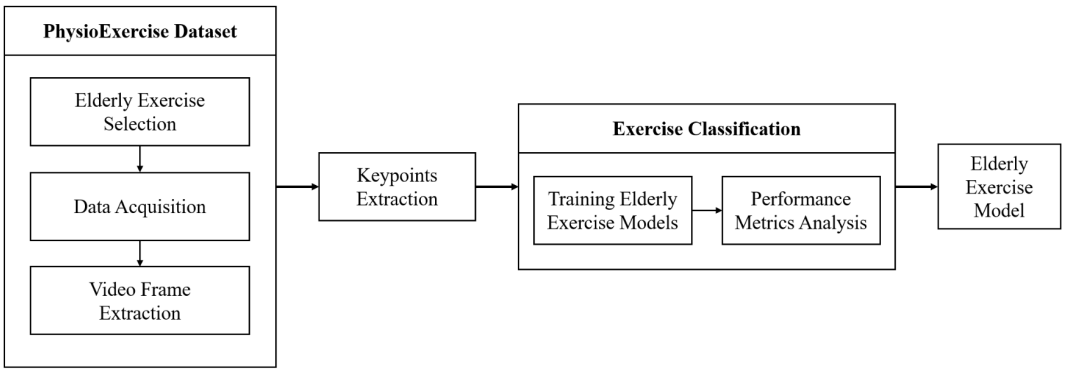

Figure 3.1. Research Block Diagram

3.1 PhysioExercise Dataset

Exercise performed by the elderly is different from exercise performed in general. Since the elderly have special issues regarding their health, the exercises given must be adapted to their conditions. There are not many exercise datasets for the elderly. Therefore, this section explains the process of creating a dataset that is suitable for elderly exercise activities.

3.1.1 Elderly Exercise Selection

As a first step in our work, we had to determine the types of exercises for the elderly. Most human activity recognition, especially for the elderly, revolves around daily activities such as standing, sitting, lying down, eating, opening doors, etc. [16, 17]. Exercises performed for the elderly must be more selective and careful due to their physical condition. Therefore, it must be consulted with physiotheraphists first. In addition, the exercises that will be categorized are the types of exercises that can be done alone and at home. One example of exercises that fit the criteria is in [69]. The work shows physiotherapy activities for the elderly that can be done at home. In addition, the exercises are specialized based on health issues that are common to the elderly physique.

3.1.2 Dataset Acquisition

We made adjustments to the dataset we used. The dataset used is a collection of physiotherapy exercise videos for the elderly based on predetermined classes. The videos were taken in our laboratory and some were taken in the home environment. Since the utilization of physiotherapy exercises is done at home independently, the use of mobile phone cameras is one of the effective solutions. Therefore, we used mobile phone cameras as input for dataset capture. The camera specifications used in this work are 64 MP, 26mm focal length, f/1.8 aperture lens, 0.8µm pixel size, and 1/1.7" sensor size. The dataset was captured in the form of a 1080p resolution video. There is 30 frames per second in the video.

We varied the angle of the video capture. We placed the camera at 3 angles, i.e., straight ahead, 30° angle relative to the left, and 30° angle relative to the right. In addition to the camera angle, we also gave people a variety of angles. We apply various orientations of the person's position to the camera. This aims to enrich our dataset. However, we limit these orientations to the possible positions of the elderly in performing physiotherapy exercises.

3.1.3 Video Frame Extraction

The acquired data is a collection of exercise activity videos. This data is organized in each folder according to the predefined class classification. We perform data preprocessing after we have collected all the required video datasets. We divided each video into 100 images. These images are sequences of physiotherapy exercise activities for each class that will have their keypoint values extracted. We have selected the videos so that they still have a high diversity value for each class.

3.2 Keypoint Extraction

In this stage, we focus on the keypoint feature extraction process which is crucial for human motion analysis in the context of exercise activity classification for the elderly. This keypoint extraction is performed by utilizing the MediaPipe framework, a solution developed by Google to enable real-time and accurate human pose estimation. MediaPipe facilitates the extraction of keypoint coordinates by using machine learning models that have been trained on large and diverse datasets, enabling efficient human body position detection.

The extraction process begins with the receipt of video frames as input, where each frame first undergoes pre-processing to ensure that the incoming data is in optimal condition for analysis. Normalization is performed on the obtained pixel coordinates, converting them to a scale that is consistent and independent of the original dimensions of the image, thus enabling accurate comparisons among different sources and image resolutions. Coordinate normalization is done following equations (3.1) and (3.2).

$$(x', y') = \left(\frac{x}{widht}, \frac{y}{height} \right) \tag{3.1}$$

Width and height refer to image dimensions. Equation (3.2) refers to depth normalization (z-depth). The minimun and maximum depths represented by $z_{\min}$ and $z_{\max}$ are detected within the frame or a preset range.

$$z' = \frac{z - z_{min}}{z_{max} - z_{min}} \tag{3.2}$$

After preprocessing, the frames are processed using MediaPipe's pose estimation model that uses a Convolutional Neural Network (CNN) architecture. This model effectively identifies and tracks 33 keypoints located in key areas of the human body, such as the head, shoulders, elbows, wrists, hips, knees and ankles. Each keypoint is detected with (x, y, z) coordinates and comes with a confidence score that indicates the accuracy of the detection.Since Mediapipe Pose Estimation has 33 keypoints, we utilize all of them in our work. We perform indexing for each keypoint. Indexing keypoints follows the equation 3.3.

$$KP = kp_0, kp_1, ..., kp_{32} \tag{3.3}$$

Where each kp_{i} corresponds to a specific part like the nose, left eye inner, right eye, etc.

The keypoints in this work are represented as vectors. The vector in Mediapipe Pose Estimation consists of keypoint kp_{i} 3-axis coordinates (x, y, z) and confidence value c. Thus, the vector V for a single image could be:

$$V = [(x_0, y_0, z_0, c), (x_1, y_1, z_1, c), ..., (x_{32}, y_{32}, z_{32}, c)] \tag{3.4}$$

Each frame that has gone through pose estimation is then stored in an array file in '.npy' format. In this storage file, there are two lists, namely sequences and labels. The sequences list will store the sequential data, containing the keypoints feature information that will be used for training. The labels list contains the labels associated with the sequences data, which gives the class information of each data. In a data set, there are 100 images that represent the number of windows in the data. Each window has keypoint coordinate data according to equation 3.4. This V vector becomes the feature used for training data. One window will have an array of size 132 because 33 keypoints have 4 vector values. So, the data format used in this study is:

$$X = (nData, nWindow, nFeatures) \tag{3.5}$$

$$y = (nData, nClass) \tag{3.6}$$

where X stores the sequential data of the dataset and y is the label for each data. nData is the total amount of data used in this dataset. It indicates how many sequences we have. nWindow represents the sequence length or number of frames in each data. In this case nWindow is 100. nFeatures is the number of features in a data frame. It indicates how many values are described in each frame. In a pose estimation using the Mediapipe framework, there are 132 feature data extracted. nClass is the number of classes that will divide each data following the training activity label. This number is predefined as 9 exercise activity classes.

In the training process, the data sequences and labels of the entire dataset are combined into one file with the '.npz' format. This file format is a format used to store multiple arrays in one file. An '.npz' file is a zip archive containing multiple '.npy' files where each array is stored as a separate '.npy' file within the archive. This format makes it possible to store arrays in a larger form.

3.3 Training Model

We trained on a laptop with an AMD Ryzen 5900HX with an integrated graphics card, NVIDIA RTX 3050 GPU support, 4 GB of VRAM, and DDR4 16 GB of RAM. The parameters used in this training apply to all models used. Our system is built in a Jupyter Notebook container with Python and the Tensorflow framework. The dataset training is done over 100 epochs. We used 80% data for data training and 20% data for data validation. Details of the parameters used can be seen in Table 3.1.

3.3.1 CNN Architecture

The architecture shown in figure 3.2 is a one-dimensional Convolutional Neural Network (CNN) model (Conv1D). Conv1D is chosen because the data used is sequential data or time-based data. The data form used is $X = (nData, nWindow, nFeatures)$. Each sample is a temporal sequence of features taken from each frame. The features used are keypoints extracted

Table 3.1. Training model hyperparameter configuration.

Epoch	100
Batch Size	16
Learning Rate	0.001
Optimizer	Adam
Loss	Categorical Crossentropy
Activation Layer	ReLU, Softmax
Class	9

from the pose estimation. Conv1D is simpler compared to other number of dimensions because it only works along one dimension, reducing the amount of computation, and parameters to be optimized. Conv1D also has good dimensionality reduction and local pattern detection capabilities. This relates to the way a model reduces data complexity by extracting important features from temporal sequences and finding relationships between keypoints in short time sequences.

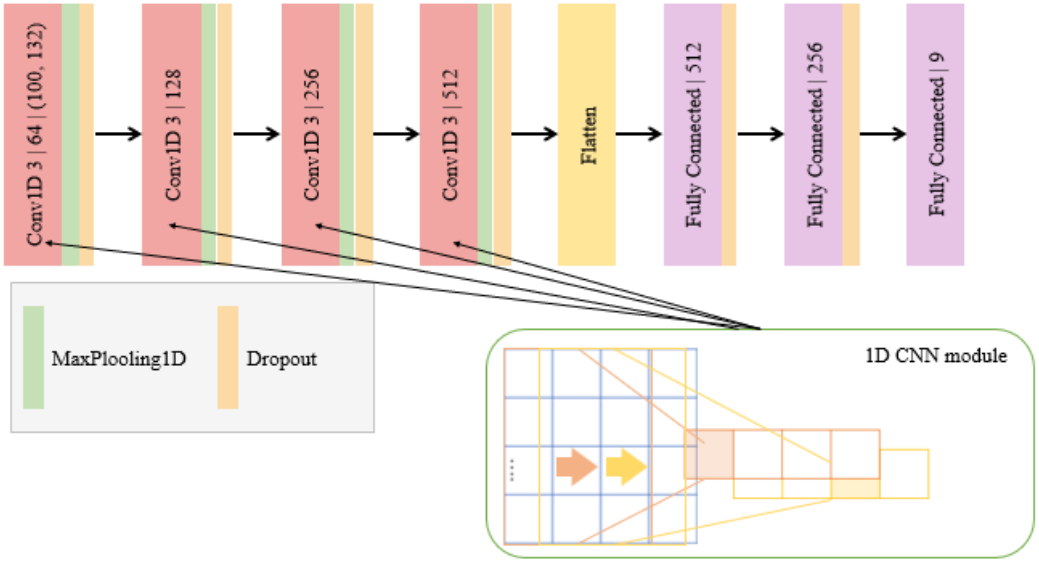

Figure 3.2. Architecture of CNN that used in this research.

The model consists of several main components. First, there are four 1D convolution layers (Conv1D) with a kernel of size 3, which have 64, 128, 256, and twice 512 filters, respectively. These layers are responsible for extracting features from the input data. After each convolution layer, there is a

MaxPooling1D layer that is used to reduce the dimensionality of the data and capture the main features, helping in reducing the data size and computation. In addition, multiple convolution layers are followed by a dropout layer to prevent overfitting by randomly disabling a number of units in the network during training. The output of the convolution and pooling layers is then flattened using the Flatten layer, turning the data into a 1D vector that can be fed to the fully connected layer. There are three fully connected layers in this architecture, with 512, 256, and 9 units respectively, where the last layer is typically used as the output layer for classification into 9 classes.

Overall, this architecture is designed to process sequential data by extracting features through a convolution layer, reducing dimensionality by pooling, preventing overfitting with dropout, and performing classification through a fully connected layer. This combination makes the 1D CNN model very suitable for tasks such as signal analysis, text processing, or other sequential data.

3.3.2 LSTM Architecture

Another architecture used is the LSTM. This architecture is a further development of the traditional RNN. LSTMs address the traditional RNN problems of vanishing gradient and long-term dependencies. LSTMs are able to remember important information for a longer period of time than traditional RNNs. The data used in this research is time series data with long temporal patterns, making LSTM suitable for this dataset.

The architecture shown in figure 3.3 is a Long Short-Term Memory (LSTM) model used for processing sequential or time-based data. This model consists of several main components. First, there are two LSTM layers. The first LSTM layer has 128 units and accepts inputs with dimensions (100, 132), followed by a dropout layer to prevent overfitting by randomly disabling a number of units in the network during training. The second LSTM layer has 64 units and is also followed by a dropout layer. This LSTM layer is used to capture long-term dependencies in sequential data.

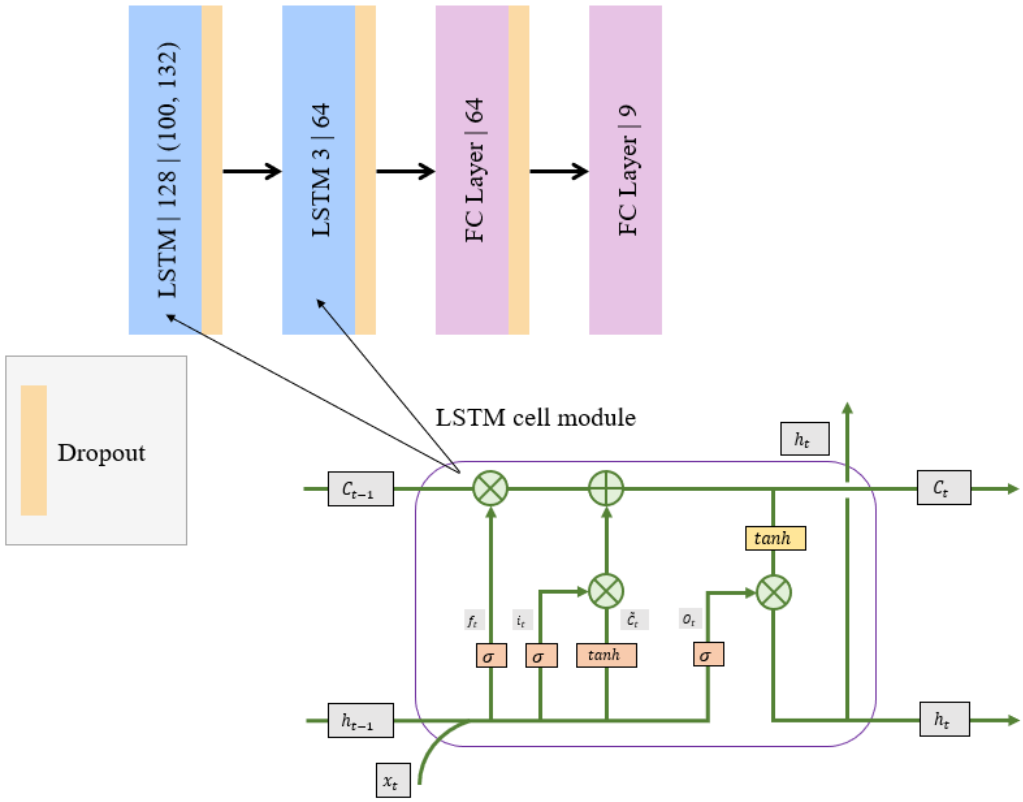

Figure 3.3. Architecture of LSTM that used in this research.

After the LSTM layer, the output is flattened and fed to two fully connected (FC) layers. The first fully connected layer has 64 units, while the second fully connected layer has 9 units, which is usually used as the output layer for classification into 9 classes. The diagram on the bottom right shows the LSTM cell module in detail, which illustrates the internal mechanism of the LSTM cell including the input gate (i_t), forgetting gate (f_t), and output gate (o_t), as well as how the state cells (C_t) are updated through sigmoid operation (σ) and tanh activation function.

The architecture's overall goal is to handle and evaluate sequential data by utilizing the fully connected layer to do classification, the LSTM layer to capture temporal information, and dropout to minimize overfitting. Because of this combination, the LSTM model performs exceptionally well in applications like signal processing, time series prediction, and text analysis.

3.3.3 CNN-LSTM Architecture

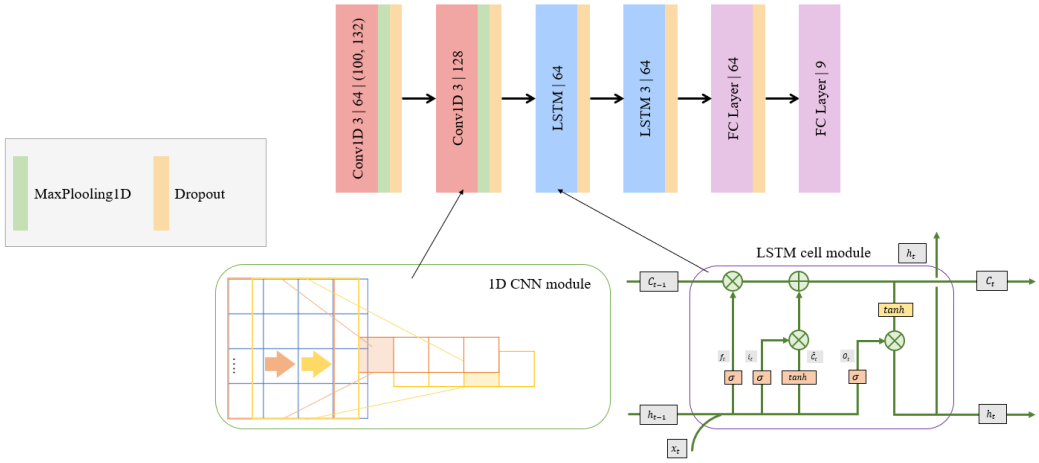

Figure 3.4. Architecture of CNN-LSTM that used in this research.

Our proposed CNN-LSTM architecture starts with two consecutive one-dimensional convolution (Conv1D) layers, each followed by a one-dimensional max pooling (MaxPooling1D) layer. The first Conv1D layer has the objective of capturing local features of the input sequence, with the convolution kernel applying a non-linear filter on a subset of the data. After each convolution layer, the MaxPooling1D layer is used to reduce the output dimension and control overfitting by taking the maximum value of a small subset of the convolution layer output. Next, the architecture continues with two Long Short-Term Memory (LSTM) layers followed by a dropout layer. The first LSTM layer serves to capture temporal dependencies in the data sequence, while the subsequent dropout layer helps reduce overfitting by randomly disabling units during training. The second LSTM layer deepens the model's understanding of complex sequence patterns, followed by another dropout layer for additional regulation. Then, the model ends with two Dense layers, each of which is followed by a dropout layer. The first Dense layer with ReLU activation function serves to combine the learned features and apply them to a lower output dimension. The last dropout layer ensures that the model still generalizes well to data that has never been seen. Finally, a second Dense layer with a softmax activation function is used to generate the final prediction that

represents the probability of each possible class. The output of this model is a model that can classify 9 classes of elderly exercise activities.

3.3.4 Deep CNN-LSTM Architecture

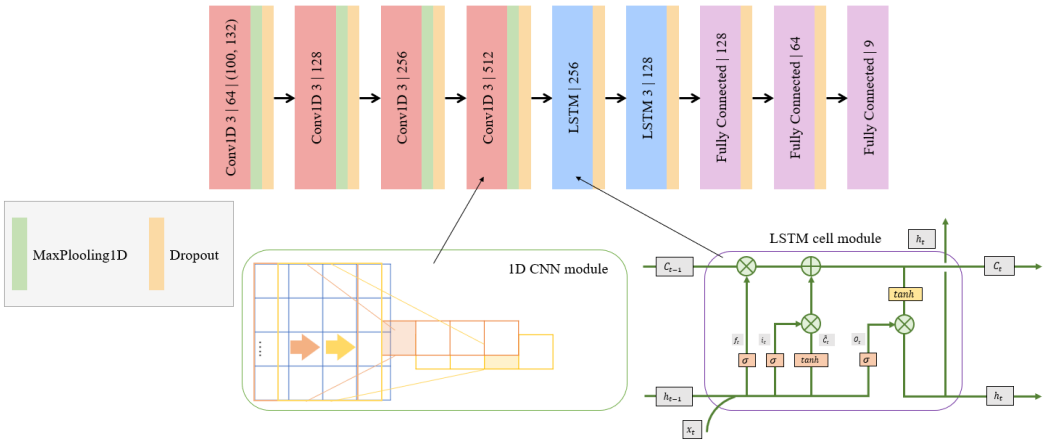

Figure 3.5. Architecture of Deep CNN-LSTM that used in this research.

The architecture shown in figure 3.5 is a combination of a one-dimensional Convolutional Neural Network (CNN) (1D CNN) and Long Short-Term Memory (LSTM) model designed to process sequential or time-based data, such as signals or text data. The model consists of several main components that work sequentially.

The first layer consists of several 1D convolution layers (Conv1D) with a size 3 kernel. The first convolution layer takes input with dimensions (100, 132) and has 64 filters. A Conv1D layer with 128 filters is next, then a layer with 256 filters, and two layers with 512 filters apiece. After each of these convolution layers, there is a dropout layer to avoid overfitting and a MaxPooling1D layer to reduce the dimensionality of the data and capture important characteristics.

The output of the convolution layer is sent to the LSTM layer following the feature extraction procedure using the convolution layer. This architecture consists of two LSTM layers, the first with 256 units and the second with 128 units. The long-term dependencies in sequential data are captured by these LSTM layers.

The output of the LSTM layer is supplied to three fully connected (FC)

layers after being flattened. The output layer for categorization into nine classes is typically the third fully connected layer, which has nine units. The first fully connected layer has 128 units, the second fully connected layer has 64 units, and so on.

This page is intentionally left blank

CHAPTER 4

RESULT AND DISCUSSION

4.1 PhysioExercise Dataset

This section is about the collected dataset and its details. Our dataset consists of 9 classes according to the predefined elderly exercise activities. Within each class, videos were taken as described in Methodology. We take several physical issues that are common in the elderly. There are three categories of physical issues that we take for this dataset, i.e., frozen shoulder, tennis elbow, and knee pain. In brief, frozen shoulder is a common condition affecting the elderly, characterized by pain and limited movement in the shoulder joint. The second physical problem is tennis elbow. Tennis elbow is a condition characterized by pain and tenderness on the outside of the elbow. The knee as the main support of the body often experiences knee pain problems.

Based on the three physical issues of the elderly that have been mentioned, we determine the exercise activities. Table 4.1 shows the exercise activity classes used in this work. There are a total of 9 kinds of activities performed to solve the above physical issues. There are 4 kinds of exercises to prevent frozen shoulder problems. The activities are adduction abduction, left arm circumduction, right arm circumduction, and shoulder flexion tension. These four exercises are movements that use the shoulder as the main focus in doing them. We take one exercise activity to prevent tennis elbow problems, which is elbow flexion tension. This movement uses the arm as the main focus. Finally, there are 4 types of training activities to prevent knee pain problems. The training activities are left knee flexion extension, right knee flexion extension, left leg flexion extension, and right leg flexion extension. Left knee flexion extension and right knee flexion extension are movements performed in a sitting position with the thighs and calves forming a 90°. In this movement an elderly person will try to straighten their calves forward. The left leg flexion extension

Table 4.1. Physiotherapy exercises used as dataset classes and the distributions.

Physical Issue	**Physiotherapy Exercise**	**Total Person**	**Total Video**
Frozen Shoulder	Adduction Abduction	5	206
	Left Arm Circumduction	5	227
	Right Arm Circumduction	6	227
	Shoulder Flexion Tension	7	229
Tennis Elbow	Elbow Flexion Tension	5	219
Knee Pain	Left Knee Flexion Extension	5	195
	Right Knee Flexion Extension	5	195
	Left Leg Flexion Extension	5	206
	Right Leg Flexion Extension	5	195
Total video dataset			1899

and right leg flexion extension are performed in a standing position. An elderly person will try to bend their knees backwards so that the thighs and knees form a 90° angle.

We used multiple subjects to conduct the data acquisition process. There were at least 7 different subjects for this data acquisition process. The age range of the subjects in this work is between 21 to 36 years old. A wide age range was chosen because the feature used for training is pose estimation keypoint extraction. Moreover, these subjects are in good health, which makes it easier to capture the right movements in accordance with the physiotherapist's recommendations. There are a total of 1899 videos in this dataset.

Figure 4.1 shows the diversity and distribution of the dataset for each class. The dataset is organized to have a variety of capture angles, such as front view, left view, and right view. The capture angle for each view

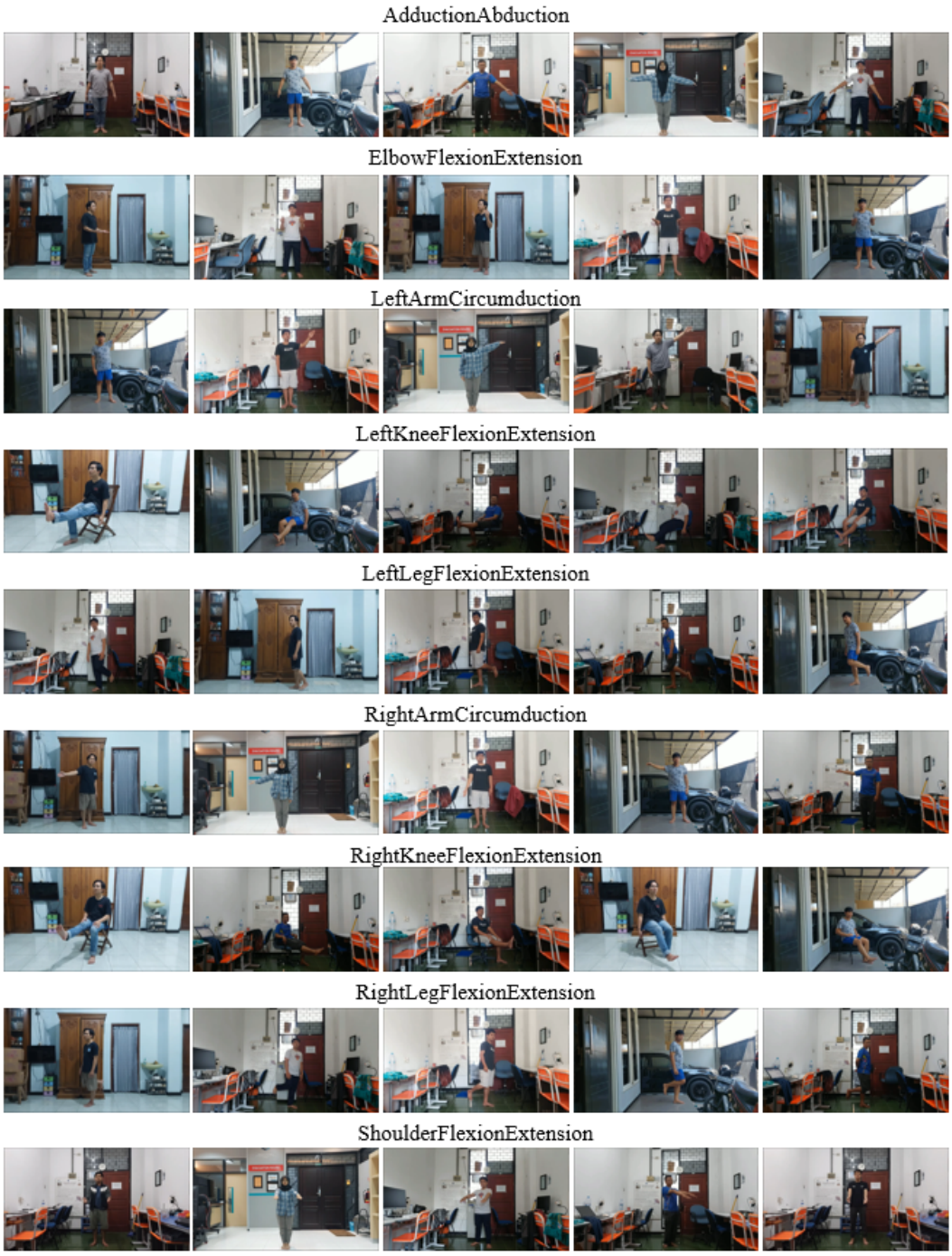

Figure 4.1. Dataset distribution of PhysioExercise.

was varied to close to a 90° relative to the front view. In addition, the video illumination level was also varied. We took the dataset in several time situations, such as morning, afternoon, evening, and night. Some videos have normal light and others have low light. The diverse lighting is intended to make

the model adaptable to various lighting conditions. We took this dataset in several different environments, namely a laboratory environment and a home environment. The home environment was chosen to match the use of the model that will be used by the elderly in their respective homes.

4.2 Video Frame Extraction

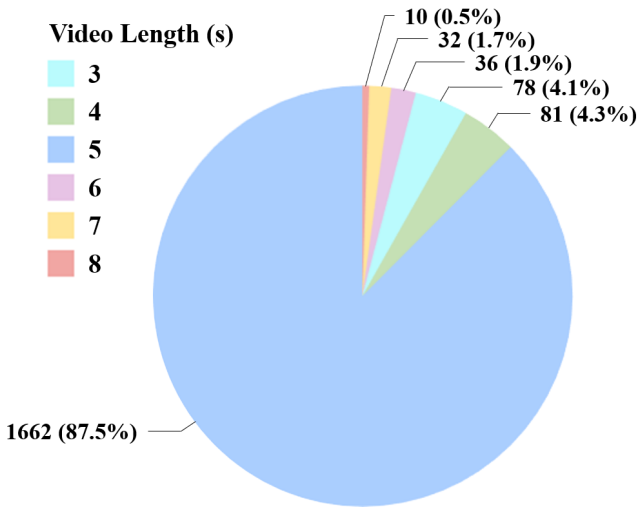

Figure 4.2. Statistics for the length distribution of our video dataset. Videos with a length of 5 seconds are the most recorded.

The acquired dataset is a set of exercise activity videos for the elderly. Since the Mediapipe framework performs frame by frame pose estimation, our data must first go through a video frame extraction process. The video data we have is a short duration video. Figure 4.2 shows the distribution of video length in the dataset. The duration of the videos we collected is in the range of 3 to 8 seconds. The duration with the least amount of data is 3 seconds, with only a total of 10 videos or 0.5% of the overall data. In contrast, the duration with the most data is 5 seconds, with a total of 1662 videos or 87.5% of the overall data.

Video frame extraction is done by dividing each video into 100 frames. This number of frames is determined so that not too many frames are extracted (12 - 30 FPS). We extract the video into fewer frames because we want to avoid frames that have too much motion blur. Motion blur in the context of

pose estimation can blur the information carried, resulting in frames that are not good for deep learning training data. On the other hand, we also keep the overall information in a video intact. Within the 100 frames for a video, we spread them evenly throughout the video from beginning to end. So no information is left out in this video frame extraction.

4.3 Keypoint Ekstraction

This stage is the stage to extract keypoint features from human pose estimation. Keypoint extraction is performed using the Mediapipe framework. The keypoint extraction process that we performed using the MediaPipe framework successfully produced high-quality keypoint data from each processed video frame. Figure 4.3 shows an example of keypoint extraction results from a frame, where keypoints are clearly identified at key positions of the subject's body. This result demonstrates the effectiveness of MediaPipe in extracting pose information with precision, even in activities that involve fast and complex movements.

Each extracted frame provides comprehensive information on the type of exercise activity performed by the subject. By sequentially extracting keypoints for each frame, we were able to build a detailed and dynamic dataset that reflects the entire range of motion in the activity. From this dataset, we were able to see how the position and orientation of the body changed over time, providing deep insight into the biomechanical characteristics of each exercise activity.

From the keypoint extraction process, we generate a data array that describes the keypoint position in three dimensions (x, y, z) and the confidence score. This array shows the (x), (y) and (z) positions for each keypoint with their associated confidence scores, where these scores indicate the accuracy of keypoint detection by the model. The use of three-dimensional data allows us to perform a more detailed analysis of the motion performed, including the depth of motion that cannot be achieved with only two-dimensional data.

Figure 4.3. Examples of pose estimation results using mediapipe for exercise activity types (a) adduction abduction, (b) elbow flexion extension, and (c) right knee flexion extension.

4.4 Data Training Preparation

This section describes the data structure of the dataset that will be used for the training process. The PhysioExercise dataset consists of raw data and data extracted from keypoints using the Mediapipe framework. Figure 4.4 shows the data structure of the dataset. The data used in this training is a collection of arrays containing information from images. To get a collection of arrays, the process goes through several stages. The first stage is to collect all video data into a folder. These video data are the raw data that will be

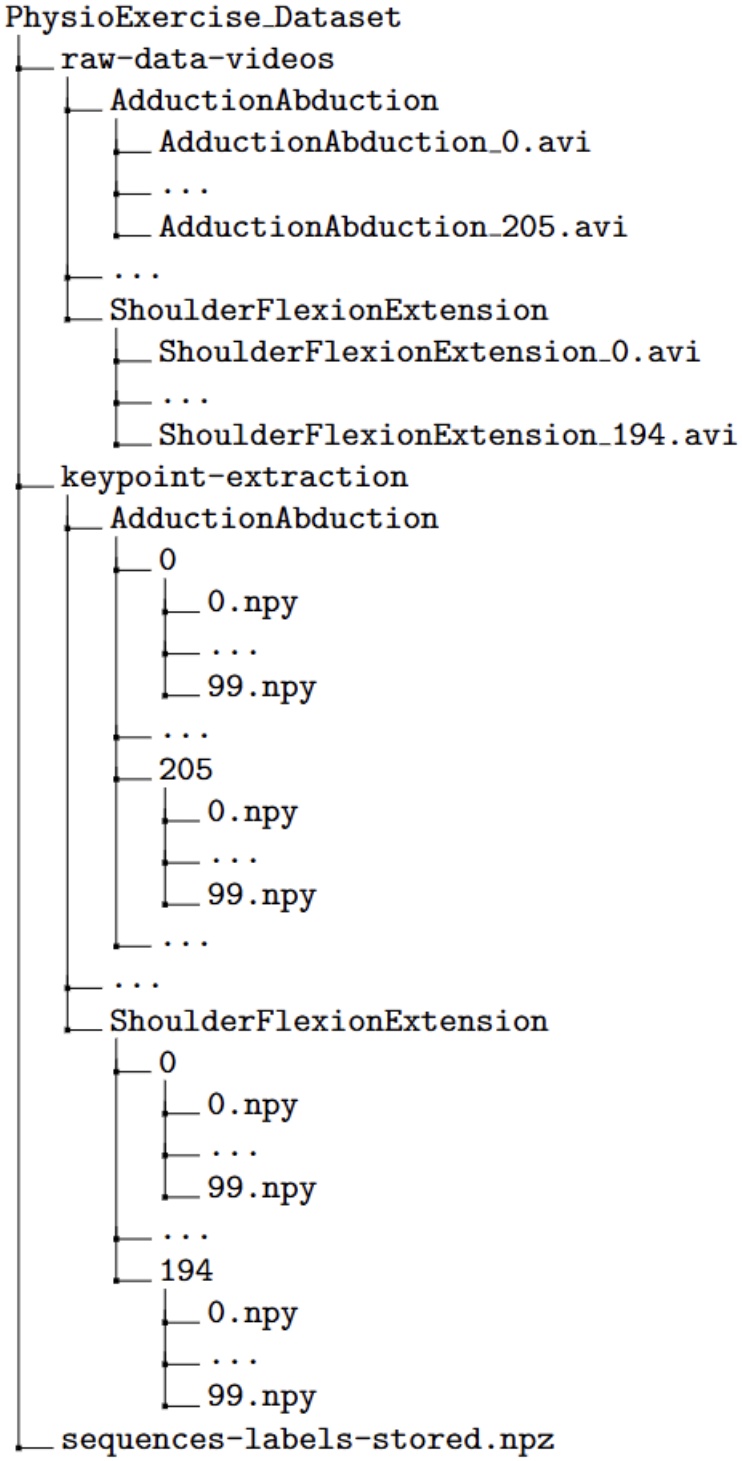

Figure 4.4. Data structure of PhysioExercise dataset.

extracted from the video frames and keypoints. The amount of data in each class is different, as shown in table 4.1.

```
Keypoint 0:
x: 0.423430860042572
y: 0.2585785984992981
z: -0.5113552212715149
c: 0.999998927116394
Keypoint 1:
x: 0.43116316199302673
y: 0.24846068024635315
z: -0.49011099338531494
c: 0.9999969005584717
Keypoint 2:
x: 0.43714720010757446
y: 0.247788667678833
z: -0.49014443159103394
c: 0.9999971389770508
...
Landmark 32:
x: 0.5420525670051575
y: 0.6121591329574585
z: 0.3682002127170563
c: 0.9751439094543457
```

Figure 4.5. Example of keypoint extraction results for a frame.

After being separated into folders according to class, the data was then extracted for video frames and keypoints. Each data is either extracted into 100 frames which are then pose estimated using the Mediapipe framework. The pose estimation takes the coordinates and visibility of each detected keypoint, following the equation 3.4. Figure 4.5 is an example of keypoint extraction results on a frame. This coordinate data belongs to all frames and is combined for each window into one complete data information. The keypoint extraction results of each frame are stored in '.npy' files so that a video data will have 100 '.npy' files that sequentially carry the exercise activity information.

To perform training, each extracted data frame is saved into a single file. Each frame generates a set of keypoints that are stored in an array. Frames from one video are collected in one time window to create a sequence. Each window is labeled according to the training activity class. The data is organized following the form of equations 3.5 and 3.6. This array of data is

```
Arrays in the npz file:
sequences: shape (1899, 100, 132)
labels: shape (1899,)

Contents of 'sequences':
[[[ 4.87661779e-01 3.11535120e-01 -2.34423041e-01 ... 9.52848315e-01
6.51585683e-02 9.85042572e-01]
[ 4.88037497e-01 3.10886085e-01 -2.15023205e-01 ... 9.52799380e-01
3.15906182e-02 9.84754205e-01]
[ 4.88345623e-01 3.10531914e-01 -2.18225241e-01 ... 9.52809811e-01
2.51260139e-02 9.84644115e-01]
...
[ 4.90713596e-01 3.10016543e-01 -2.47297361e-01 ... 9.52778459e-01
1.65367201e-02 9.89496648e-01]
[ 4.90500867e-01 3.10013443e-01 -2.45748788e-01 ... 9.52728510e-01
1.65824648e-02 9.89543736e-01]
[ 4.90239203e-01 3.10015142e-01 -2.45856449e-01 ... 9.52671826e-01
1.69171784e-02 9.89561856e-01]]

[[ 5.22423863e-01 2.83315539e-01 -2.46881694e-01 ... 9.63253617e-01
5.32212071e-02 9.89582300e-01]
[ 5.25366127e-01 2.80411243e-01 -2.36928344e-01 ... 9.66800690e-01
7.00890496e-02 9.89524782e-01]
[ 5.26178718e-01 2.77596295e-01 -2.47585267e-01 ... 9.68481362e-01
6.36128485e-02 9.89068627e-01]
...
-9.72992100e-04 9.92523074e-01]]]

Contents of 'labels':
[0 0 0 ... 8 8 8]
```

Figure 4.6. The contents of the '.npz' file that stores sequences and labels data.

stored in '.npz' format, a format capable of storing multiple arrays. Figure 4.6 shows the contents of the '.npz' file containing the data ready for the training process.

4.5 Performance Metrics Evaluation

We have trained various deep learning architectures on the PhysioExercise dataset. The tested architectures include CNN, LSTM, CNN-LSTM, and deep CNN-LSTM. From each of these architectures, we managed to develop one best model, which we then compared against each other to evaluate their performance.

4.5.1 CNN Model Performance

One of the architectures used in this work is the CNN architecture. This section describes the performance of the CNN model using the PhysioExercise dataset. Figure 4.7 is the accuracy and loss graph for training and validation

data. This graph provides an overview of how the model behaves during the training and validation process using the CNN architecture. The training was done in 100 epochs. In the initial 20 epochs, the model experienced a fairly high increase in accuracy. The increase occurred around 0.2 to 0.6 in these first 20 epochs. After entering the 21st epoch, the training accuracy continued to increase and stabilized at around 0.8 accuracy, while the validation accuracy fluctuated between 0.7 to 0.8. This fluctuation shows that the model experienced some overfitting at some points. This overfitting indicates that the model is too adaptive to the training data and less able to generalize to the validation data.

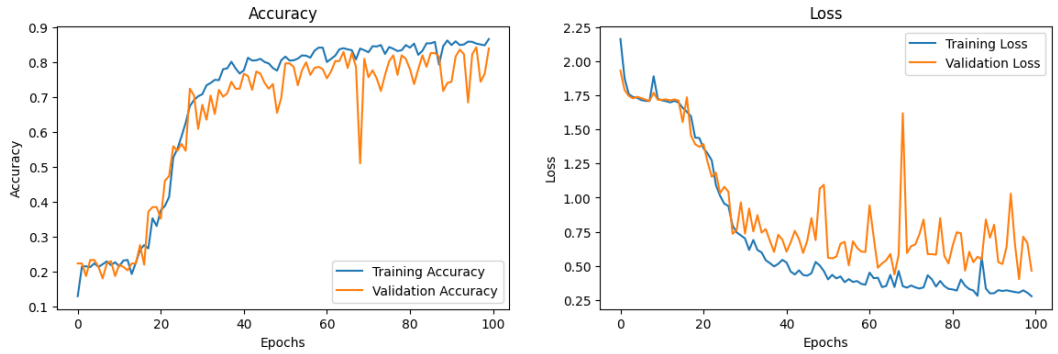

Figure 4.7. Accuracy and loss of training and validation on the CNN model.

In the first 20 epochs of the loss graph, the model loss has decreased dramatically to below 1.0. This value shows that the model is starting to be able to predict accurately. Training and validation loss tend to decrease until the end of training. However, validation loss does not decrease as well as training loss. In addition, the validation loss experiences several peaks and significant fluctuations. This graph once again shows an overfitting model and a lack of generalization to the validation data.

The confusion matrix shown in figure 4.8 provides a visual representation of the performance of the CNN model in classifying exercise activities for the elderly. Each row of the confusion matrix shows the actual number of instances for each class, while each column shows the number of instances predicted by the model for each class. There are 4 classes that show excellent

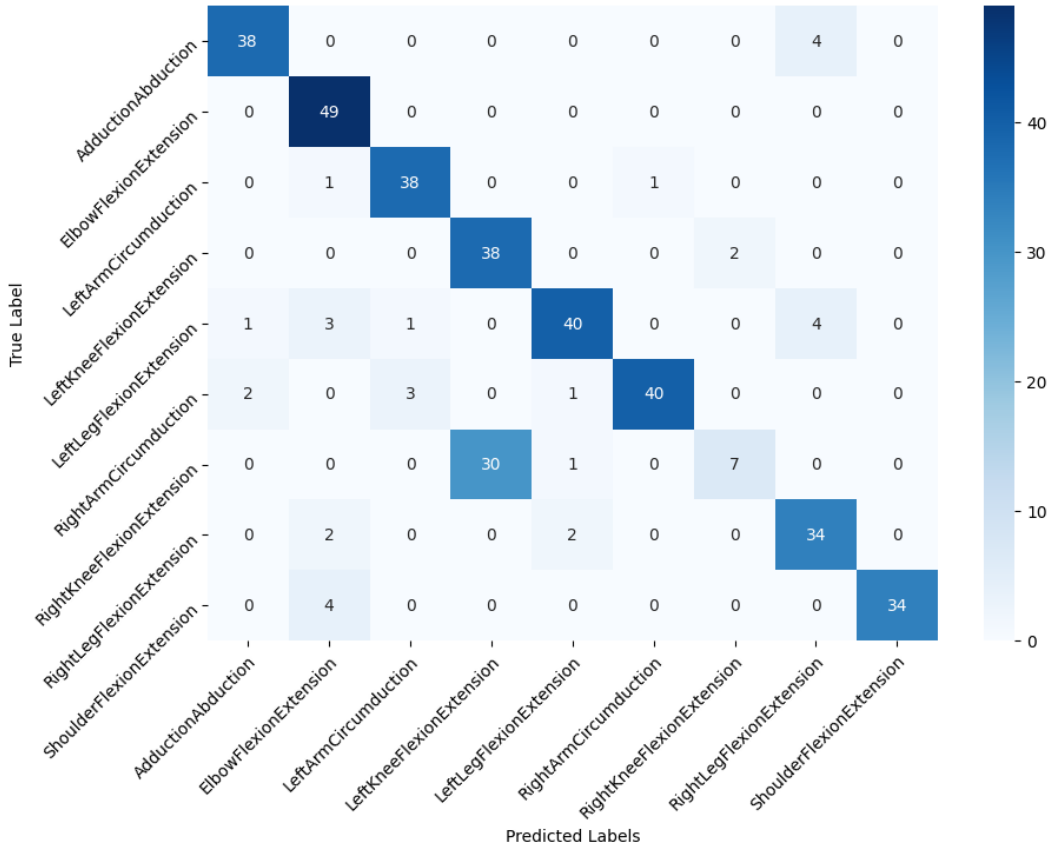

Figure 4.8. Confusion matrix of the CNN model outcomes.

performance, namely in the ElbowFlexionExtension, LeftArmCircumduction, RightArmCircumduction, and ShoulderFlexionExtension classes. The model predictions in these classes closely match the actual labels, with 49, 38, 30, and 34 correctly classified instances, respectively.

However, there are classes that experience significant confusion in making predictions. That class is RightKneeFlexionExtension. Of the 38 instances in the RightKneeFlexionExtension class, only 7 were correctly classified, while 30 instances were classified as LeftKneeFlexionExtension. These errors contributed to the low precision and recall of these classes, as shown in the classification report. This misclassification can occur based on several factors. One of the main factors is the similarity in motion between the two. The model failed to predict the RightKneeFlexionExtension instance instead classifying it as LeftKneeFlexionExtension. The similar features between the

two are the main reason for the model's confusion in classifying correctly. Although the model showed good ability in classifying most of the classes, further improvement is needed to reduce confusion between classes that have movements that may be visually similar.

Table 4.2. Classification report of the CNN model.

	precision	recall	f1-score	support
AdductionAbduction	0.93	0.90	0.92	42
ElbowFlexionExtension	0.83	1.00	0.91	49
LeftArmCircumduction	0.90	0.95	0.93	40
LeftKneeFlexionExtension	0.56	0.95	0.70	40
LeftLegFlexionExtension	0.91	0.82	0.86	49
RightArmCircumduction	0.98	0.87	0.92	46
RightKneeFlexionExtension	0.78	0.18	0.30	38
RightLegFlexionExtension	0.81	0.89	0.85	38
ShoulderFlexionExtension	1.00	0.89	0.94	38
accuracy			0.84	380
macro avg	0.85	0.83	0.81	380
weighted avg	0.86	0.84	0.82	380

Table 4.2 presents the classification report for the model using CNN architecture. Evaluation using the classification report shows the precision, recall, and f1-score metrics for each activity. This classification model has shown quite good performance. Overall, the accuracy of this model is 84%. The overall results for each metric were calculated using macro averaging and weighted averaging methods. Using macro averaging, the average results for precision, recall, and f1-score of this model are 0.85, 0.83, and 0.81 respectively. Unlike the macro averaging method, calculations using the weighted averaging method showed a precision of 0.86, recall of 0.84, and F1-score of 0.82. This method is more representative of the class distribution in a dataset.

The ShoulderFlexionExtension and RightArmCircumduction classes have very high precision, reaching 1.00 and 0.98 respectively. This value indicates that there are very few false positives in the predictions for these activities. This explanation is in line with the confusion matrix results where these

classes can predict all instances well. However, the model showed difficulty with the RightKneeFlexionExtension class with a precision of only 0.78 and recall of 0.18. This indicates a large number of false negatives that affect the model's ability to detect this activity. The LeftKneeFlexionExtension activity also has a relatively low precision of 0.56 although the recall is quite high at 0.95, indicating that the model often incorrectly predicts this class as another activity.

Table 4.3. Accuracy results for training, validation, and test data of the CNN model.

Training accuracy (%)	Validation accuracy (%)	Test accuracy (%)
86.83	83.83	83.68

Table 4.3 is a summary of the accuracy for training, validation, and test data. Of the three, the accuracy value of the CNN model shows a consistent value. The model achieved a training accuracy of 86.83%, which shows that the model is able to learn well from the training data. The validation accuracy of 83.83% shows that the model has a good generalization ability to data not seen during training. Both accuracy results are consistent with the previous accuracy-loss graph where the validation accuracy remains stable around 0.83 after 20 epochs. The test accuracy value of 83.68% confirms that the model performance remains stable when tested on an entirely new dataset. This indicates that the model did not experience significant overfitting. This is also supported by the classification report and confusion matrix which show that most classes are well classified, although there are some classes that show lower performance such as RightKneeFlexionExtension. The consistency between training, validation and test accuracies shows that the model performs reliably in this classification task, although there is still room for improvement especially in classes that show higher misclassification.

4.5.2 LSTM Model Performance

Training of the PhysioExercise dataset was also performed using the LSTM architecture. Figure 4.9 represents the accuracy and loss graphs for

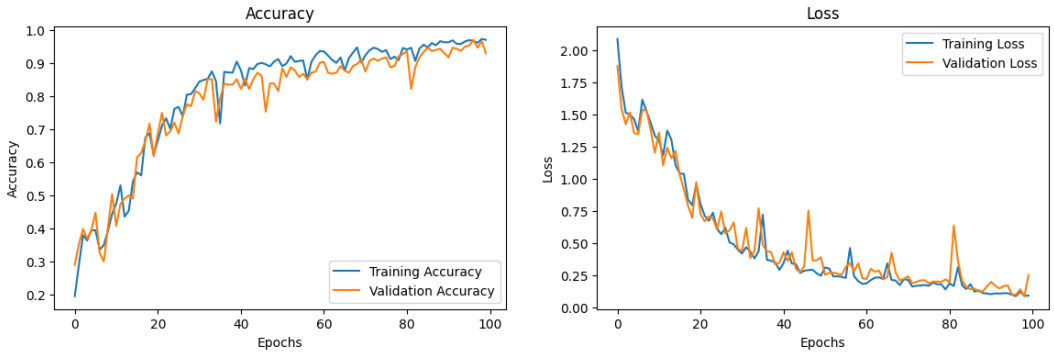

Figure 4.9. Accuracy and loss of training and validation on the LSTM model.

training and validation data for each training epoch. The training has been done in 100 epochs. The accuracy graph on the left depicts the increase in accuracy in both the training data (Training Accuracy) and validation data (Validation Accuracy). At the beginning of training, both accuracies increase rapidly, indicating that the model immediately starts learning patterns from the data. Validation accuracy also shows an increasing trend, although there are some larger fluctuations. These fluctuations indicate a slight overfitting but overall it still generalizes well to the untrained data.

The loss graph on the right illustrates the decrease in loss in both the training data (Training Loss) and the validation data (Validation Loss). At the beginning of training, both losses decrease rapidly. The rapidly decreasing loss values indicate that the model is quickly reducing the prediction error. The training loss continues to decrease and approaches zero, indicating that the model is almost perfect on the training data. Meanwhile, the validation loss also decreases but with larger fluctuations, indicating that there are some epochs where the model does not predict the validation data as accurately as in other epochs. This fluctuation could be an indication of overfitting, where the model learns too specifically on the training data and is less able to generalize to new data. On the other hand, the validation loss value is still above the training loss value by quite a distance.

The confusion matrix of the LSTM model for classification of exercise activities in the elderly is shown in figure 4.10. This confusion matrix describes

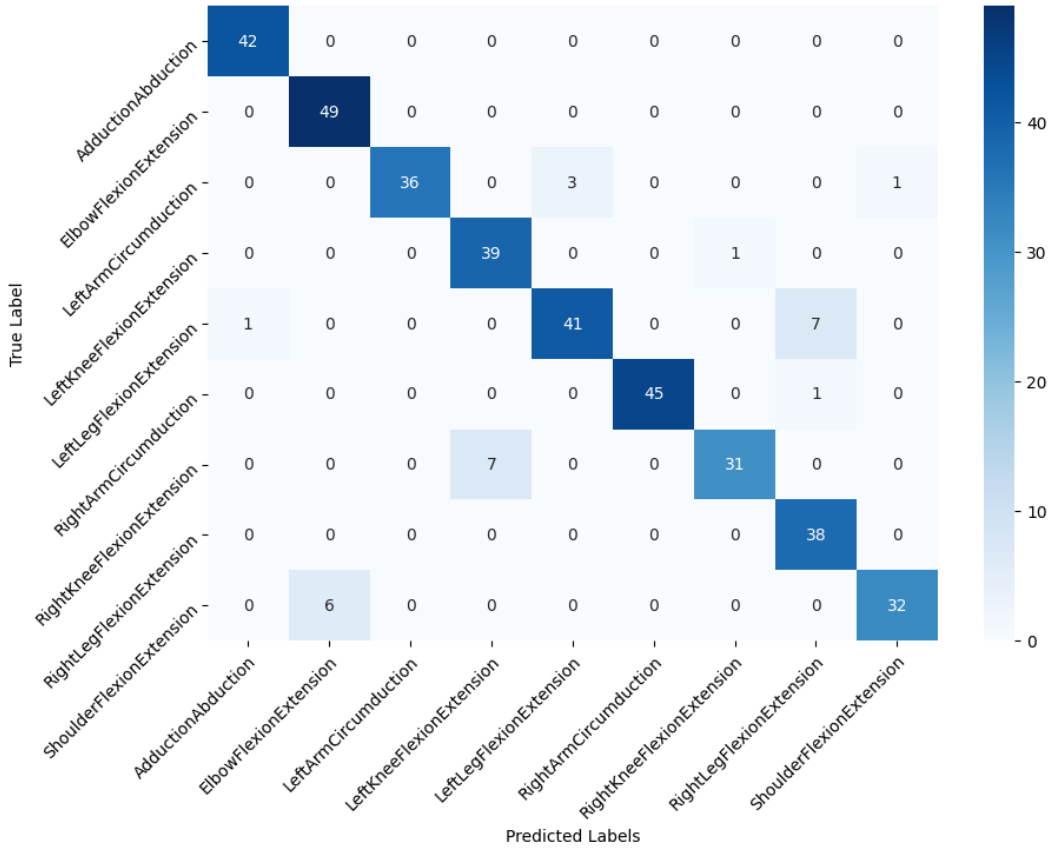

Figure 4.10. Confusion matrix of the LSTM model outcomes.

the number of correct predictions (main diagonal) and misclassifications (off-diagonal) for each class. Looking at the main diagonal, it appears that the model managed to classify most of the data correctly. Some examples are in the AdductionAbduction activity with 42 correctly classified data, ElbowFlexionExtension with 49 correctly classified data, and RightLegFlexionExtension with 38 correctly classified data. However, there are some classes that experience errors in classification. For example, the LeftLegFlexionExtension class predicts 7 data as RightLegFlexionExtension, and the RightKneeFlexionExtension class predicts 6 data as LeftKneeFlexionExtension.

As a further analysis, misclassification often occurs in classes that have similar or adjacent activities in the motion space, such as LeftLegFlexionExtension and RightLegFlexionExtension. However, the model as a whole performed quite well with high accuracy in the majority of the classes, especially Ad-

ductionAbduction, ElbowFlexionExtension, and RightLegFlexionExtension. The misclassifications indicate that although the LSTM model is solid, there is room for further improvement, such as by adding more training data or tweaking the model to improve the classification ability on hard-to-distinguish classes. Thus, this confusion matrix provides deeper insight into the model's performance in classifying exercise activities in the elderly, supports the previously described analysis, and shows areas where the model can still be improved.

Table 4.4. Classification report of the LSTM model.

	precision	recall	f1-score	support
AdductionAbduction	0.98	1.00	0.99	42
ElbowFlexionExtension	0.89	1.00	0.94	49
LeftArmCircumduction	1.00	0.90	0.95	40
LeftKneeFlexionExtension	0.85	0.97	0.91	40
LeftLegFlexionExtension	0.93	0.84	0.88	49
RightArmCircumduction	1.00	0.98	0.99	46
RightKneeFlexionExtension	0.97	0.82	0.89	38
RightLegFlexionExtension	0.83	1.00	0.90	38
ShoulderFlexionExtension	0.97	0.84	0.90	38
accuracy			0.93	380
macro avg	0.93	0.93	0.93	380
weighted avg	0.94	0.93	0.93	380

An explanation of the classification report of the LSTM model is shown in table 4.4. This table reports the precision, recall, f1-score, and support metrics for each class, as well as macro averaging and weighted averaging of the entire model. The precision value measures the correct positive predictions out of all positive predictions made by the model. A higher precision indicates the model makes fewer errors in predicting a particular class. Based on the table 4.4, LeftArmCircumduction and RightArmCircumduction have a precision value of 1.00. This result shows that the model is able to predict this class well, even though they are both activities with adjacent motion spaces. RightLegFlexionExtension is the class with the lowest precision among

the other classes with a precision value of 0.83. Even so, this value is still considered high for prediction work. On the other hand, recall is a metric that measures the proportion of actual instances of a class that are correctly identified by the model. In this LSTM model, there are three classes with perfect recall values, i.e., AdductionAbduction, ElbowFlexionExtension, and RightLegFlexionExtension. This value indicates that the model is able to predict these classes perfectly. The lowest recall value in this LSTM model is in the RightKneeFlexionExtension class with a value of 0.82. This value is still relatively high even though the error rate is the highest compared to other classes.

F1-score gives the average harmony of precision and recall which shows the balance between these two metrics. The classes with the highest f1-score values are AdductionAbduction and RightArmCircumduction with f1-score values of 0.99. The lowest value for this metric is 0.88 for the LeftLegFlexionExtension class. This value corresponds to its lowest recall value compared to the recall of other classes. These metrics are overall modeled using macro averaging and weighted averaging methods. In the macro averaging method, the precision, recall, and f1-score metrics have the same value, which is 0.93. On the other hand, the values of the three using the weigthed averaging method have values of 0.94, 0.93, and 0.93 respectively. Weighted averaging is considered more representative of the class distribution in a dataset.

Table 4.5. Accuracy results for training, validation, and test data of the CNN model.

Training accuracy (%)	Validation accuracy (%)	Test accuracy (%)
94.93	93.07	92.89

We also measured the model accuracy for each training, validation, and test data. Table 4.5 shows the accuracy value of the model for each of these data. The training accuracy reaches 94.93% which means the model is able to learn the patterns from the training data very well. The validation accuracy reached 93.07%. Both values are in line with the accuracy-loss graph

in figure4.9, where the validation accuracy is not higher than the training accuracy. The ability of the model is shown in the test accuracy, where the data given is data that has never been read by the model. The test accuracy of this model is 92.89%. This value indicates that the generalization ability of the model is quite good.

4.5.3 CNN-LSTM Model Performance

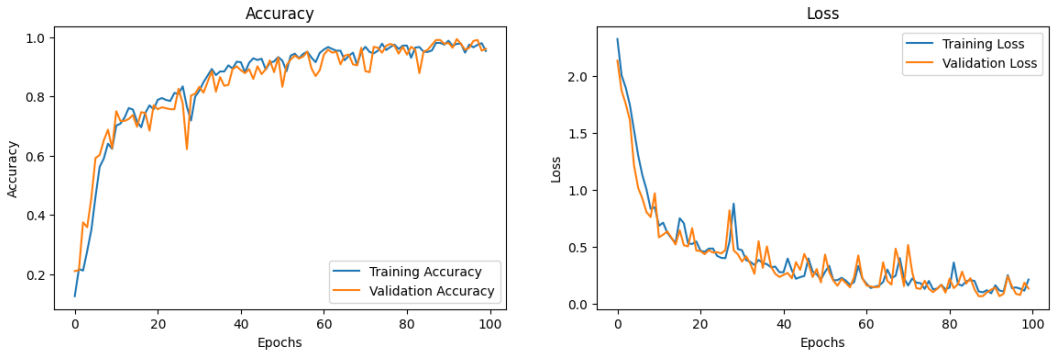

Figure 4.11. Accuracy and loss of training and validation on the CNN-LSTM model.

Another architecture used in this work is the CNN-LSTM architecture. Figure 4.11 provides an overview of the model in learning the PhysioExercise dataset. There are accuracy and loss graphs for training and validation data. The training has been done in 100 epochs. The left graph shows the accuracy of the model for each epoch. A significant improvement occurs in the first 15 epochs for training accuracy. This significant increase indicates that the model is able to learn the data patterns very well. In the following epochs, the training accuracy continues to increase until the end of the training. Although on several occasions the training accuracy had decreased, the trend in training accuracy continued to rise. Similarly, the validation accuracy experienced a significant increase for the first 15 epochs. Then the increase in accuracy occurs until the end of training although the fluctuations that occur are greater than the training accuracy. Fluctuations between training and validation accuracy occurred several times but the trend still showed an increase in accuracy until the training ended.

The loss graph shows the error rate made by the model in learning the data pattern. The loss value decreased significantly in the first 15 epochs. This graph trend applies to both training and validation losses. This decreasing loss trend continues until the end of training although fluctuations in both occur. Similar to accuracy, the loss graph fluctuates throughout the training. This fluctuation shows the ability of the model to learn the given data pattern. The validation graph has larger fluctuations when compared to the training graph. This shows the ability of the model to learn the data several times overfitting. However, the difference between the two values does not show a high number so that the model's ability to generalize data patterns is still considered good.

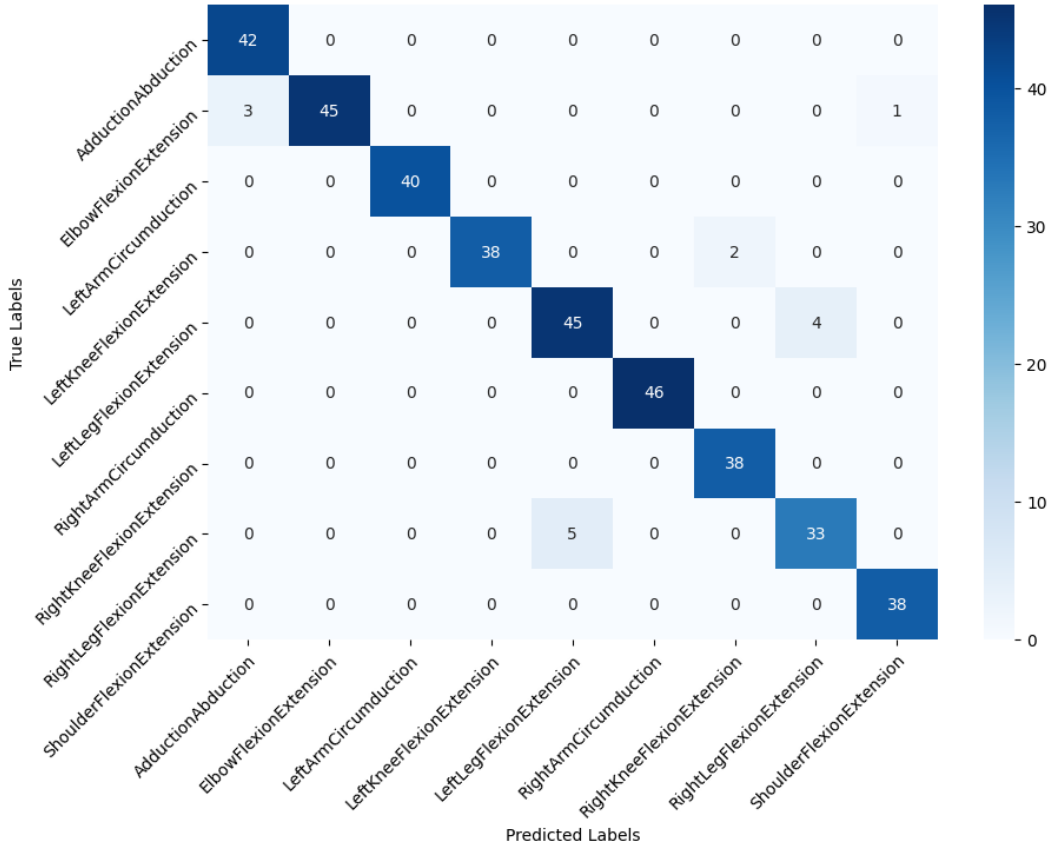

Figure 4.12. Confusion matrix of the CNN-LSTM model outcomes.

Figure 4.12 displays the confusion matrix of the CNN-LSTM model. This confusion matrix illustrates the performance of the model in classifying various exercise activities for the elderly. The main diagonal shows the number of

correct predictions for each class, while values off the main diagonal show the number of incorrect predictions.

Based on this matrix analysis, it appears that the model performs very well in classifying exercise activities for all classes. For example, the AdductionAbduction class, the model successfully predicted correctly 42 times without any errors. In addition to the AdductionAbduction class, there are other classes that were successfully predicted without any errors, i.e., LeftArmCircumduction, RightArmCircumduction, RightKneeFlexionExtension, and ShoulderFlexionExtension. The other class had some errors when the model tried to predict. For instance, 3 instances of the ElbowFlexionExtension class are considered as AdductionAbduction and 1 instance as ShoulderFlexionExtension. The RightLegFlexionExtension class has the highest error in prediction. Five instances are predicted as LeftLegFlexionExtension. Even so, this number of errors is still classified as a very low number of errors. The other class had some errors when the model tried to predict. For instance, 3 instances of the ElbowFlexionExtension class are considered as AdductionAbduction and 1 instance as ShoulderFlexionExtension. The RightLegFlexionExtension class has the highest error in prediction. Five instances are predicted as LeftLegFlexionExtension. Even so, this number of errors is still classified as a very low number of errors.

The classification report of the CNN-LSTM model is shown in table 4.6. This table includes several metrics, i.e., precision, recall, f1-score, and support for each class. In addition, the overall value of the model is also displayed using macro averaging and weighted averaging methods. There are several classes that have perfect precision values (1.00). These classes are ElbowFlexionExtension, LeftArmCircumduction, LeftKneeFlexionExtension, and RightArmCircumduction. This value indicates that the model correctly predicted all instances of these classes. This result is in line with the explanation in the previous confusion matrix. Although not perfect, the other classes have high precision values too. The lowest precision value in

Table 4.6. Classification report of the CNN-LSTM model.

	precision	recall	f1-score	support
AdductionAbduction	0.93	1.00	0.97	42
ElbowFlexionExtension	1.00	0.92	0.96	49
LeftArmCircumduction	1.00	1.00	1.00	40
LeftKneeFlexionExtension	1.00	0.95	0.97	40
LeftLegFlexionExtension	0.90	0.92	0.91	49
RightArmCircumduction	1.00	1.00	1.00	46
RightKneeFlexionExtension	0.95	1.00	0.97	38
RightLegFlexionExtension	0.89	0.87	0.88	38
ShoulderFlexionExtension	0.97	1.00	0.99	38
accuracy			0.96	380
macro avg	0.96	0.96	0.96	380
weighted avg	0.96	0.96	0.96	380

this model is in the RightLegFlexionExtension class of 0.89. This value is high considering that in previous models there were several classes with lower precision values. In another metric, recall, the model is able to predict finding all positive instances very well. There are 4 classes with perfect recall values, namely AdductionAbduction, LeftArmCircumduction, RightArmCircumduction, RightKneeFlexionExtension and ShoulderFlexionExtension. Similar to precision, the lowest recall value in this model is 0.87, in the RightLegFlexionExtension and RightLegFlexionExtension classes.

F1-score shows the average harmony between precision and recall. Similar to the previous two metrics, the f1-score of each class in this model shows satisfactory results. The lowest value in this metric is 0.88 and is only in one class, namely RightLegFlexionExtension. There are 2 classes with perfect f1-score values, namely LeftArmCircumduction, and RightArmCircumduction. This perfect F1-score is obtained from perfect results in the precision and recall of these classes. Both macro averaging and weighted averaging, the average value of all metrics for this model is at a very high value. For all metrics, the score was 0.96 for both methods. These results show excellent performance in all classes whether or not the data distribution is taken into account.

Table 4.7. Accuracy results for training, validation, and test data of the CNN-LSTM model.

Training accuracy (%)	Validation accuracy (%)	Test accuracy (%)
96.71	96.04	96.05

Another measurement calculated is the accuracy value of the model against the training, validation, and test data. Table 4.7 shows the accuracy value of each data. The training accuracy shows a high result of 96.71%. This high accuracy shows that the model is able to learn the training data patterns very well. The learning error is very small. On the validation data, the accuracy was 96.04%. This high value has shown that the model is not just overfitting the training data. The model was able to classify the new data very well. This training and validation accuracy value is consistent with the previous graph which shows an increase in accuracy every epoch. Model testing was performed on completely new data using test data. The test accuracy that emerged in this model was 96.05%. The very high accuracy on all data confirms that the model is reliable and has good generalization.

4.5.4 Deep CNN-LSTM Model Performance

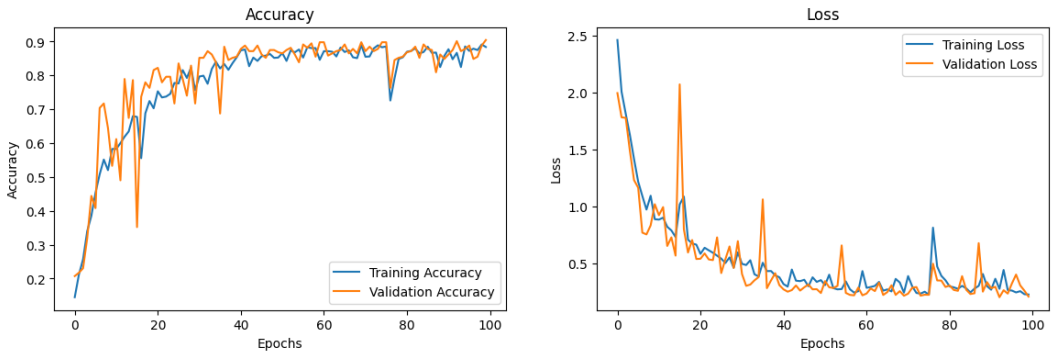

Figure 4.13. Accuracy and loss of training and validation on the Deep CNN-LSTM model.

The next architecture used for training the PhysioExercise dataset is deep CNN-LSTM. This training resulted in a deep CNN-LSTM model. Figure 4.13 shows a detailed overview of the accuracy and loss for each epoch. There are two lines representing the training and validation data. The training took place

in 100 epochs. The training accuracy of each epoch has a good increase. The gradual increase from the beginning to the end of training shows the ability of the model to learn and adapt to the given data pattern. Fluctuations occur in the training data, but the fluctuations are not too large. Unlike the validation accuracy, which experienced high fluctuations on several occasions. Even so, the trend of validation data shows the same trend as training accuracy. Both go hand in hand in improving the model's ability to learn data patterns.

The loss graph illustrates the decrease in loss values on the training and validation datasets. The decreasing value indicates that the model is able to reduce errors in learning data patterns. Entering the 41st epoch, the loss values in both training and validation experience values that tend to stabilize towards lower values. Even so, fluctuations in both losses occurred during the training process. Especially in the validation data, the loss value had a fairly high spike at some points. This spike was quite high at the beginning of the training. The spike indicates that the model had experienced some overfitting because it relied too much on the training data.

Figure 4.14 displays the prediction results of the deep CNN-LSTM model against each class. The main diagonal shows the number of correct predictions for each class. While the values on the off-diagonal show the number of incorrect predictions. Overall, the model has a good ability to predict each instance. For example, the model successfully classified 42 AdductionAbduction samples, 49 ElbowFlexionExtension samples, 38 RightKneeFlexionExtension samples, and 38 RightLegFlexionExtension samples without any error.

There was one class where the model failed to correctly predict all instances, namely the LeftKneeFlexionExtension class. Of all the instances, instead of predicting it as the LeftLegFlexionExtension class, the model predicted it as RightKneeFlexionExtension. This is a very likely error given that these two classes have similar motion spaces. In the remaining instances, the misclassification was minor and insignificant. The features of the two classes are also similar, so the model experiences confusion in classifying

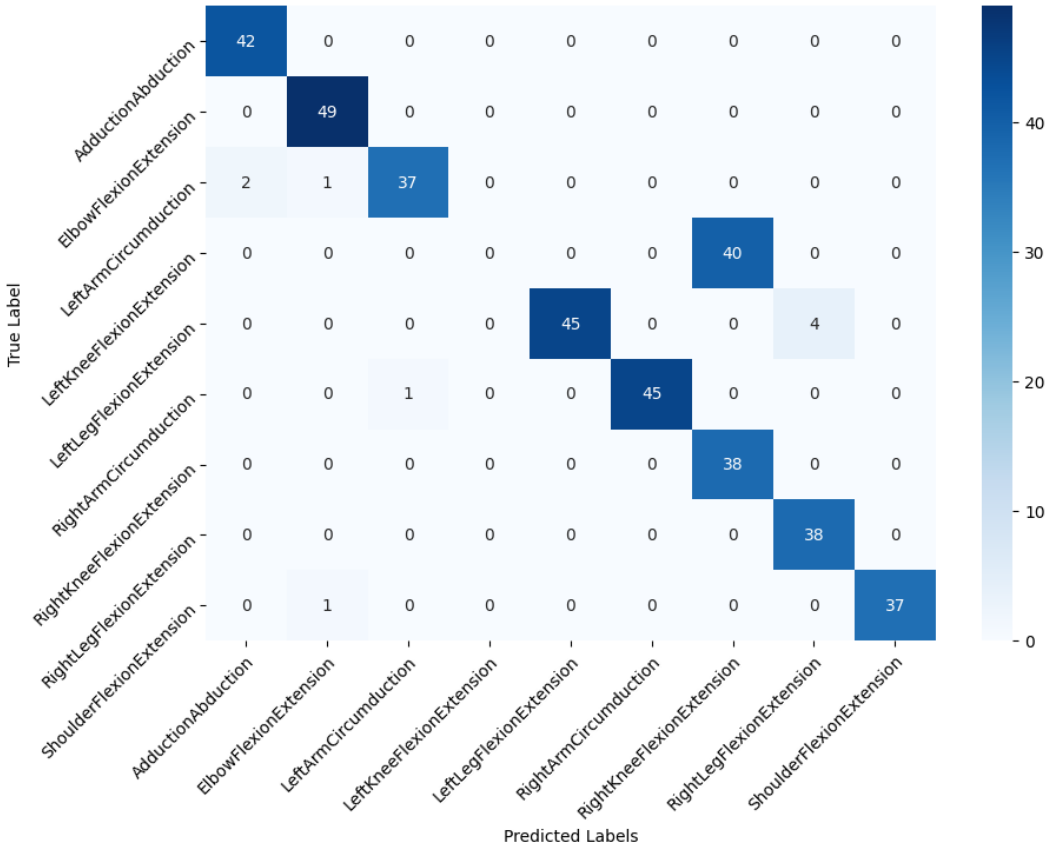

Figure 4.14. Confusion matrix of the Deep CNN-LSTM model outcomes.

correctly. In addition, when looking back at the dataset, the angle of the image capture needs to be paid more attention so that the keypoint extraction process becomes better. The addition of augmentation data is important if the model is not good enough to classify all classes.

The classification report of the deep CNN-LSTM model is presented in table 4.8. Evaluation using the classification report shows the precision, recall, and F1-score metrics for each activity. This classification model has shown quite good performance. Overall, the accuracy of the model was 0.87. The overall results for each metric were calculated using macro averaging and weighted averaging methods. Using macro averaging, the average results for precision, recall, and F1-score of this model are 0.81, 0.87, 0.83 respectively. Unlike the macro averaging method, calculations using the weighted averaging method showed a precision of 0.82, recall of 0.87, and F1-score of 0.84. This

method is more representative of the class distribution in a dataset.

Table 4.8. Classification report of the Deep CNN-LSTM model.

	precision	recall	f1-score	support
AdductionAbduction	0.95	1.00	0.98	42
ElbowFlexionExtension	0.96	1.00	0.98	49
LeftArmCircumduction	0.97	0.93	0.95	40
LeftKneeFlexionExtension	0.00	0.00	0.00	40
LeftLegFlexionExtension	1.00	0.92	0.96	49
RightArmCircumduction	1.00	0.98	0.99	46
RightKneeFlexionExtension	0.49	1.00	0.66	38
RightLegFlexionExtension	0.90	1.00	0.95	38
ShoulderFlexionExtension	1.00	0.97	0.99	38
accuracy			0.87	380
macro avg	0.81	0.87	0.83	380
weighted avg	0.82	0.87	0.84	380

The precision value measures the correct positive predictions out of all positive predictions made by the model. A higher precision indicates the model makes fewer errors in predicting a particular class. Based on the table 4.8, there are several classes with perfect precision values. The classes are LeftLegFlexionExtension, RightArmCircumduction, and ShoulderFlexionExtension. For the other classes, the model has a good ability to predict the given instance. However, there is one class with a very low precision value, namely RightKneeFlexionExtension. Precision in this class is only 0.49. This fairly low value indicates the number of instances that are predicted incorrectly by the model. In addition, there is one class where the model failed to predict correctly. That class is LeftKneeFlexionExtension. The precision value of 0.00 indicates that the model did not predict this class correctly at all. This precision value is in line with the discussion on the confusion matrix for the LeftKneeFlexionExtension class.

Similar to precision, the recall results for some classes have perfect values. These classes are AdductionAbduction, ElbowFlexionExtension, RightKneeFlexionExtension, and RightLegFlexionExtension. This value in-

dicates the model's good ability to find predictions for all positive instances. In the case of the RightKneeFlexionExtension class, the recall value of this class is not as bad as the precision value. Precision is quite low despite perfect recall, indicating that many predictions are wrong for this class. In the other hand, the LeftKneeFlexionExtension class gets no recall value at all, which is 0.00. In the other metric, f1-score, every class is above 0.90 except for two classes. The two classes are RightKneeFlexionExtension and LeftKneeFlexionExtension. The RightKneeFlexionExtension class had an f1-score of 0.66. This is far below the f1-score of the other classes. Even worse, the LeftKneeFlexionExtension class does not get an f1-score at all. This is related to the absence of precision and recall values for this class. The model failed to predict this class. One of the reasons is the complexity of this movement.

The overall results for each metric were calculated using macro averaging and weighted averaging methods. Using macro averaging, the average results for precision, recall, and F1-score of this model are 0.81, 0.87, and 0.83 respectively. Unlike the macro averaging method, calculations using the weighted averaging method showed a precision of 0.82, recall of 0.87, and F1-score of 0.84. The macro average and weighted average of precision, recall, and F1-Score each show good values, indicating a balanced performance in most classes.

Table 4.9. Accuracy results for training, validation, and test data of the deep CNN-LSTM model.

Training accuracy (%)	Validation accuracy (%)	Test accuracy (%)
89.14	90.43	87.11

Figure 4.9 provides information about the accuracy of the deep CNN-LSTM model on training data, validation data, and test data. This is the final part of the evaluation series that has previously been illustrated through the accuracy and loss graphs, confusion matrix, and classification report. The model achieved an accuracy of 89.14% on the training data. This shows that the model is able to learn from the training data quite well, identifying patterns

that match the target. The accuracy on the validation data reached 90.43%. The higher accuracy on the validation data compared to the training data indicates that the model has good generalization ability and is not overfitting on the training data. The accuracy on the test data was 87.11%. This is the most important metric as it describes the performance of the model on data that is completely new and not seen during training and validation. This accuracy is consistent with the validation accuracy and slightly lower, which is still within reasonable limits, indicating that the model is reliable in making predictions on new data.

4.6 Performance Analysis of Models

There are 4 different models that have been generated in this work. These models have different architectures used. An in-depth discussion has been done in the previous section. The results of the four models were then analyzed to compare the performance of each model. Table 4.10 shows an overview of the performance of each model. Several metrics are taken for comparison. These metrics are precision, recall, f1-score, accuracy, and model loss.

Table 4.10. Classification performance of exercise activities for the elderly on models.

Metrics		CNN	LSTM	CNN-LSTM	Deep CNN-LSTM
Macro Average	Precision	0.85	0.93	**0.96**	0.81
	Recall	0.83	0.93	**0.96**	0.87
	F1-Score	0.81	0.93	**0.96**	0.83
Weighted Average	Precision	0.86	0.94	**0.96**	0.82
	Recall	0.84	0.93	**0.96**	0.87
	F1-Score	0.82	0.93	**0.96**	0.84
Accuracy (%)		83.68	92.89	**96.05**	87.11
Loss		0.4513	0.2629	**0.1498**	0.3004

Higher is better. For loss, lower is better.

The model performance comparison table shows that the CNN-LSTM architecture consistently has the best performance in the classification of exercise activities for the elderly compared to the CNN, LSTM, and Deep CNN-LSTM

models. The CNN model is the model with the lowest performance. The accuracy value of this model is only 83.68% and the loss value is quite high, which is 0.4513. The Deep CNN-LSTM model is the next low-performing model. The accuracy of this model is at 87.11% with a loss value of 0.3004. Unlike the two previous models, the LSTM model has a better performance. The accuracy of this model is at a value of 92.89% and the loss is 0.2629. The CNN-LSTM model is the model with the best performance. The accuracy of the CNN-LSTM model is 96.05% and the loss is 0.1498. This figure shows the ability of the CNN-LSTM model in the classification work of the best exercise activity compared to other models.

The CNN model is the model with the worst performance compared to other models. The accuracy of this model is only 83.68%. This figure shows that the accuracy of the model in classifying exercise activities is very low. In the classification process, the CNN model has the highest error rate among other models. This is indicated by the highest loss value of the CNN model compared to other models. CNN model loss is 0.4513. There are many factors that can affect the low performance of the CNN model. CNN is a well-known architecture in the context of image data processing. In time series data processing, the time series data structure will be considered as a one-dimensional image. This data structure will consider time as one dimension and other features as other dimensions. However, CNN's ability to handle this data has many limitations. CNNs tend to be less effective in capturing long-term dependencies in time series data, especially when compared to LSTM. In addition, determining the right CNN architecture for time series is a job in itself because it is more complicated when compared to other architectures such as LSTM. The dataset used in this work is a time series data with a large number of windows and features. With the large amount of temporal and spatial data used, the CNN model did not succeed in classifying the instances properly.

The performance of the LSTM model is high compared to the CNN

and Deep CNN-LSTM models. The LSTM model used in the classification of exercise activities in the elderly showed excellent performance, with high accuracy on training, validation, and test data. The accuracy-loss graph shows a steady and consistent upward trend, while the confusion matrix and classification report provide further evidence of the model's ability to correctly classify most classes. However, the model is not better when compared to the CNN-LSTM model. LSTM models are designed to capture long-term temporal dependencies, but are not effective in extracting locally or spatially significant features in short time windows. In the dataset used, the spatial data in question is the keypoint extracted features from the Mediapipe framework. The LSTM model is not able to handle this keypoint extraction data compared to the CNN-LSTM model. The LSTM model is a complex model with many parameters that can cause overfiting problems. In addition, the LSTM model requires a lot of data for effective training. In contrast to the CNN-LSTM model, where CNN will reduce the dimensions of the input entering the LSTM. In the end, the complexity of the CNN-LSTM model becomes lower and is able to improve generalization.

CNN-LSTM achieved the highest precision, recall, and f1-score values in both macro average and weighted average metrics, with 0.96 each. This shows that the CNN-LSTM model is highly accurate in making positive predictions and is able to detect almost all positive instances correctly, providing an optimal balance between precision and recall. In addition, the CNN-LSTM model also showed the highest accuracy of 96.05% and the lowest loss value of 0.1498, reflecting a very low prediction error rate. The combination of CNN and LSTM shows a good ability to handle spatial data and temporal data in one dataset.

The Deep CNN-LSTM model is a model with poor performance after the CNN model. This model has a similar architecture to the CNN-LSTM model. The difference of this model is the addition of several CNN layers that become deeper in the training process. However, the additional layer architecture does

not show better results. There are several reasons why this might happen. First, too many layers make the model learn too much from the training data and fail to generalize to the test data. This is evidenced in the figure 4.13 where the accuracy of the validation dta experiences many times large fluctuations. Secondly, more complex models require more data to train effectively. If the available data is not large enough or diverse enough, adding layers is ineffective because it does not have enough information to learn the data patterns. This will create a model that needs to be. Thirdly, the addition of these layers causes redundancy in the extracted features. Instead of making the model get new information, the length of the layer adds complexity without increasing the model's ability because the information carried is the same or does not increase. This is evidenced in the figure 4.14 where there is a class that fails to be classified correctly for all its instances.

These results indicate that the combination of CNN and LSTM in the CNN-LSTM architecture provides significant advantages in this classification task. Therefore, CNN-LSTM can be considered as the most efficient and effective model for classifying exercise activities for the elderly, with potential for real applications that require accurate and reliable predictions. Meanwhile, the performance of Deep CNN-LSTM shows potential for further improvement to increase accuracy and reduce loss values.

4.7 Processing Time Speed Result

This section describes the processing time speed of the model in classifying an instance. The experiment is conducted by randomly selecting one Inference on data that the model has never recognized before. The input for this test is video data. This data is then extracted into 100 frames following the trained model. The result of these frames is extracted keypoints and then stored into an array. The result of the data that is ready to be classified by this model is calculated processing time until it comes out the class that has been classified by the model.

The CNN model shows the fastest processing time speed compared to

Table 4.11. Inference time of exercise activities for the elderly on models.

Model	Inference Time (s)	Percentage (%)
CNN	**0.1316**	14.57
LSTM	0.2491	27.58
CNN-LSTM	0.2416	26.75
Deep CNN-LSTM	0.2810	31.11

other models. This is influenced by the simplicity of the model and the CNN model's ability to process local or spatial data. Even so, keep in mind that the CNN model has the lowest accuracy rate compared to other models. The LSTM model is not faster than the CNN-LSTM and CNN models. This can be explained because the parameter complexity of the LSTM model and its ability to handle spatial data is not better than the CNN architecture. Therefore, the CNN-LSTM model is the fastest model after the CNN model. The process in CNN reduces the dimension of the LSTM input, making the classification process slightly faster than the LSTM model alone. The Deep CNN-LSTM model is a fairly long model because of its longer and heavier process.

This page is intentionally left blank

CHAPTER 5
CONCLUSION AND SUGGESTION

After implement the methodology to solve the problem that stated in the 1.2, there are several findings related to the research described as follows:

5.1 Conclusion

In this study, we propose the classification of exercise activities for the elderly using deep learning. Exercise activities for the elderly are important so that the elderly can maintain their health in old age. The work begins with the acquisition of datasets in the form of exercise activities that have been adapted to the physical issues of the elderly. Data acquisition is carried out since there are few and limited datasets that discuss this exercise activity. Video data that has been acquired is then subjected to a video frame extraction process. Each frame sequence represents exercise activity information. Pose estimation has been done using the Mediapipe framework. The extraction results are then trained using CNN, LSTM, CNN-LSTM, and deep CNN-LSTM architectures. The accuracy of each model is 83.68%, 92.89%, 96.05%, and 87.11%. Based on these results, the CNN-LSTM model outperforms the other models with an accuracy rate of 96.05%. The error in recognizing data patterns is shown using the loss metric. The loss value of the CNN-LSTM model is 0.1498, the smallest compared to other models. This value indicates the model's ability to predict data with the lowest error rate. In addition, in other metrics, this model outperforms other models. Precision, recall, and f1-score of the CNN-LSTM model are at 0.96, respectively.

5.2 Suggestion

Although this work shows a good ability to classify exercise activities for the elderly, there are some suggestions that can be implemented. These suggestions are used to optimize the work to get better results.

- Addition of datasets to enrich the trained data.
- Adding subjects to the dataset with a wider age range.
- Addition of activity classes to solve more physical issues.
- Tuning the hyperparamater to get the optimal model.
- Considering other architectures in order to get a model evaluation with more architectural variations.

Bibliography

[1] T. Kraal, J. Lübbers, M. P. J. van den Bekerom, J. Alessie, Y. van Kooyk, D. Eygendaal, and R. C. T. Koorevaa, "The puzzling pathophysiology of frozen shoulders: a scoping review," *Journal of Experimental Orthopaedics*, vol. 11, no. 3, pp. 249–257, Nov 2020.

[2] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *CoRR*, vol. abs/2012.13392, 2020. [Online]. Available: https://arxiv.org/abs/2012.13392

[3] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," *CoRR*, vol. abs/2006.10204, 2020. [Online]. Available: https://arxiv.org/abs/2006.10204

[4] D. Dai, "An introduction of cnn: Models and training on neural network models," in *2021 International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR)*, 2021, pp. 135–138.

[5] U. Nations, *World Population Ageing 2019 Highlights*. United Nations., Dec 2019. [Online]. Available: https://books.google.co.id/books?id=-mz8DwAAQBAJ

[6] A. E.-H. Mostafa Okasha and H. Mohamed Al-Kaliouby, "Physiotherapy and dynamic exercises as interventions for improving the functional efficiency of the frozen shoulder in elderly people with type 2 diabetes," *Journal of Applied Sports Science*, vol. 4, no. 2, pp. 1–11, 2014.

[7] R. P. Nirschl, "Tennis elbow," *Orthopedic Clinics of North America*, vol. 4, no. 3, pp. 787–800, 1973.

[8] M. Porcheret, K. Jordan, C. Jinks, and P. C. in collaboration with the Primary Care Rheumatology Society, "Primary care treatment of knee pain—a survey in older adults," *Rheumatology*, vol. 46, no. 11, pp. 1694–1700, 2007.

[9] S. Green, R. Buchbinder, S. E. Hetrick, and C. M. Group, "Physiotherapy interventions for shoulder pain," *Cochrane database of systematic reviews*, vol. 2013, no. 3, 1996.

[10] H. K. Rashid, D. Samanta, S. S. George, V. J. Cardoza44, and Z. Ali, "Current physical therapies available for the rehabilitation of tennis elbow: A," *International Journal of Physiotherapy Research and Clinical Practice*, vol. 1, pp. 26–32, Aug 2022.

[11] G. Peat, R. McCarney, and P. Croft, "Knee pain and osteoarthritis in older adults: a review of community burden and current use of primary health care," *Annals of the rheumatic diseases*, vol. 60, no. 2, pp. 91–97, 2001.

[12] N. K. Arden, S. Crozier, H. Smith, F. Anderson, C. Edwards, H. Raphael, and C. Cooper, "Knee pain, knee osteoarthritis, and the risk of fracture," *Arthritis Care & Research: Official Journal of the American College of Rheumatology*, vol. 55, no. 4, pp. 610–615, 2006.

[13] S. Ariyani, E. Mulyanto Yuniarno, and M. Hery Purnomo, "Heuristic application system on pose detection of elderly activity using machine learning in real-time," in *2022 IEEE 9th International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 2022, pp. 1–6.

[14] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, "Blazeface: Sub-millisecond neural face detection on mobile gpus," *CoRR*, vol. abs/1907.05047, 2019. [Online]. Available: http://arxiv.org/abs/1907.05047

[15] F. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan 2016. [Online]. Available: http://dx.doi.org/10.3390/s16010115

[16] M. Gochoo, T.-H. Tan, S.-C. Huang, S.-H. Liu, and F. S. Alnajjar, "Dcnn-based elderly activity recognition using binary sensors," in *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 2017, pp. 1–5.

[17] H. Xu, Y. Pan, J. Li, L. Nie, and X. Xu, "Activity recognition method for home-based elderly care service based on random forest and activity similarity," *IEEE Access*, vol. 7, pp. 16 217–16 225, 2019.

[18] D. Sharma and P. D. B. R. Sharma, "Geriatric age – boon or bane – role of yoga for geriatric health and healthy ageing," vol. 44, p. 167–172, Oct. 2023. [Online]. Available: http://jazindia.com/index.php/jaz/article/view/576

[19] H. B. Y. Chan, P. Y. Pua, and C. H. How, "Physical therapy in the management of frozen shoulder," *Singapore Med J*, vol. 58, no. 12, pp. 685–689, Dec 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5917053/

[20] B. Dirgantari, "The physiotherapy treatment of frozen shoulder cases to improve the scope of joint movement and ability for daily activities: Case report," *Jurnal Fisioterapi dan Kesehatan*

Indonesia, vol. 3, no. 2, pp. 222–227, Oct. 2023. [Online]. Available: https://ifi-bekasi.e-journal.id/jfki/article/view/185

[21] D. Wattanaprakornkul, I. Cathers, M. Halaki, and K. A. Ginn, "The rotator cuff muscles have a direction specific recruitment pattern during shoulder flexion and extension exercises," *Journal of science and medicine in sport*, vol. 14, no. 5, pp. 376–382, 2011.

[22] J. C. Politti, C. J. Felice, and M. Valentinuzzi, "Arm emg during abduction and adduction: hysteresis cycle," *Medical engineering & physics*, vol. 25, no. 4, pp. 317–320, 2003.

[23] V. Gracia-Ibáñez, J. L. Sancho-Bru, M. Vergara, A. Roda-Sales, N. J. Jarque-Bou, and V. Bayarri-Porcar, "Biomechanical function requirements of the wrist. circumduction versus flexion/abduction range of motion," *Journal of Biomechanics*, vol. 110, p. 109975, 2020.

[24] D. Wattanaprakornkul, I. Cathers, M. Halaki, and K. A. Ginn, "The rotator cuff muscles have a direction specific recruitment pattern during shoulder flexion and extension exercises," *Journal of science and medicine in sport*, vol. 14, no. 5, pp. 376–382, 2011.

[25] A. Kuppuswamy, M. Catley, N. K. King, P. H. Strutton, N. J. Davey, and P. H. Ellaway, "Cortical control of erector spinae muscles during arm abduction in humans," *Gait & posture*, vol. 27, no. 3, pp. 478–484, 2008.

[26] S. Cutts, S. Gangoo, N. Modi, and C. Pasapulab, "Tennis elbow: A clinical review article," *Journal of Orthopaedic*, vol. 17, pp. 203–207, Aug 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6695331/

[27] D. W. Hadi, H. Sugiharto, and A. Tiksnadi, "Functional and pain improvement in tennis elbow with dry needling as alternative treatment: Case series," *touchREVIEWS in Neurology*, vol. 17, no. 1, pp. 60–63, 2021, publisher Copyright: ©2021, Touch Medical Media. All rights reserved.

[28] T. Kodek and M. Munih, "An analysis of static and dynamic joint torques in elbow flexion-extension movements," *Simulation modelling practice and theory*, vol. 11, no. 3-4, pp. 297–311, 2003.

[29] C. W. Bunt, C. E. Jonas, and J. G. Chang, "Knee pain in adults and adolescents: The initial evaluation," *American Family Physician*, vol. 98, no. 9, pp. 576–585, Nov 2018.

[30] A. J. Thirumaran, L. A. Deveza, I. Atukorala, and D. J. Hunter, "Assessment of pain in osteoarthritis of the knee," *Journal of Personalized Medicine*, vol. 13, no. 7, 2023. [Online]. Available: https://www.mdpi.com/2075-4426/13/7/1139

[31] S. Farrokhi, Y.-F. Chen, S. R. Piva, G. K. Fitzgerald, J.-H. Jeong, and C. K. Kwoh, "The influence of knee pain location on symptoms, functional status and knee-related quality of life in older adults with chronic knee pain: data from the osteoarthritis initiative," *Clin J Pain*, vol. 32, no. 6, pp. 463–470, June 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4892642/

[32] V. Duong, W. M. Oo, C. Ding, A. G. Culvenor, and D. J. Hunter, "Evaluation and treatment of knee pain: A review," *JAMA*, vol. 330, no. 16, pp. 1568–1580, 2023.

[33] T. J. Housh, W. G. Thorland, G. D. Tharp, G. O. Johnson, and C. J. Cisar, "Isokinetic leg flexion and extension strength of elite adolescent female track and field athletes," *Research Quarterly for Exercise and Sport*, vol. 55, no. 4, pp. 347–350, 1984.

[34] G. L. Smidt, "Biomechanical analysis of knee flexion and extension," *Journal of biomechanics*, vol. 6, no. 1, pp. 79–92, 1973.

[35] M. Pal and P. Rubini, "Gesture recognition for autistic children using person pose estimation and supervised learning," in *2021 IEEE 3rd PhD Colloquium on Ethically Driven Innovation and Technology for Society (PhD EDITS)*, 2021, pp. 1–2.

[36] G. Airò Farulla, D. Pianu, M. Cempini, M. Cortese, L. Russo, M. Indaco, R. Nerino, A. Chimienti, C. Oddo, and N. Vitiello, "Vision-based pose estimation for robot-mediated hand telerehabilitation," *Sensors*, vol. 16, no. 2, p. 208, Feb 2016. [Online]. Available: http://dx.doi.org/10.3390/s16020208

[37] A. Jalal, A. Nadeem, and S. Bobasu, "Human body parts estimation and detection for physical sports movements," in *2019 2nd International Conference on Communication, Computing and Digital systems (C-CODE)*, 2019, pp. 104–109.

[38] L. Bridgeman, M. Volino, J.-Y. Guillemaut, and A. Hilton, "Multi-person 3d pose estimation and tracking in sports," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 2487–2496.

[39] R. Blythman, M. Saxena, G. Tierney, C. Richter, A. Smolic, and C. Simms, "Assessment of deep learning pose estimates for sports collision tracking," *Journal of Sports Sciences*, vol. 40, 09 2022.

[40] A. K, P. P, and J. Paulose, "Human body pose estimation and applications," in *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2021, pp. 1–6.

[41] K. Padmanand and P.-C. Lim, "Malaysian sign language recognition using 3d hand pose estimation," in *2022 International Conference on Digital Transformation and Intelligence (ICDI)*, 2022, pp. 214–218.

[42] S. Adhikary, A. K. Talukdar, and K. Kumar Sarma, "A vision-based system for recognition of words used in indian sign language using mediapipe," in *2021 Sixth International Conference on Image Information Processing (ICIIP)*, vol. 6, 2021, pp. 390–394.

[43] E. Marinoiu, M. Zanfir, V. Olaru, and C. Sminchisescu, "3d human sensing, action and emotion recognition in robot assisted therapy of children with autism," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2158–2167.

[44] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social motion capture," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[45] F. Noroozi, C. A. Corneanu, D. Kamińska, T. Sapiński, S. Escalera, and G. Anbarjafari, "Survey on emotional body gesture recognition," *IEEE Transactions on Affective Computing*, vol. 12, no. 2, pp. 505–523, 2021.

[46] D. Maji, S. Nagori, M. Mathew, and D. Poddar, "Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 2636–2645.

[47] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, 2021.

[48] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7157–7173, 2023.

[49] Y. Zhang, J. H. Han, Y. W. Kwon, and Y. S. Moon, "A new architecture of feature pyramid network for object detection," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020, pp. 1224–1228.

[50] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[51] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[52] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3711–3719.

[53] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7103–7112.

[54] M. Kresovic and T. D. Nguyen, "Bottom-up approaches for multi-person pose estimation and it's applications: A brief review," *CoRR*, vol. abs/2112.11834, 2021. [Online]. Available: https://arxiv.org/abs/2112.11834

[55] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021.

[56] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 969–11 978.

[57] A. Amini, H. Farazi, and S. Behnke, "Real-time pose estimation from images for multiple humanoid robots," in *RoboCup 2021: Robot World Cup XXIV*, R. Alami, J. Biswas, M. Cakmak, and O. Obst, Eds. Cham: Springer International Publishing, 2022, pp. 91–102.

[58] X. Nie, J. Feng, J. Xing, and S. Yan, "Pose partition networks for multi-person pose estimation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 705–720.

[59] J.-W. Kim, J.-Y. Choi, E.-J. Ha, and J.-H. Choi, "Human pose estimation using mediapipe pose and optimization method based on a humanoid model," *Applied Sciences*, vol. 13, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/4/2700

[60] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," 2020.

[61] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile

gpus," *CoRR*, vol. abs/1907.06724, 2019. [Online]. Available: http://arxiv.org/abs/1907.06724

[62] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[63] A. Mathew, P. Amudha, and S. Sivakumari, "Deep learning techniques: An overview," in *Advanced Machine Learning Technologies and Applications*, A. E. Hassanien, R. Bhatnagar, and A. Darwish, Eds. Singapore: Springer Singapore, 2021, pp. 599–608.

[64] Y. Tu, X. Chu, and W. Yang, "Computer-aided process planning in virtual one-of-a-kind production," *Computers in Industry*, vol. 41, pp. 99–110, 01 2000.

[65] H. Agrawal, "Comparative analysis of different convolutional neural network algorithm for image classification," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, pp. 1110–1120, 09 2020.

[66] H. Shao, "Delay-dependent stability for recurrent neural networks with time-varying delays," *IEEE Transactions on Neural Networks*, vol. 19, no. 9, pp. 1647–1651, 2008.

[67] A. Samad, Bhagyanidhi, V. Gautam, P. Jain, Sangeeta, and K. Sarkar, "An approach for rainfall prediction using long short term memory neural network," in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, 2020, pp. 190–195.

[68] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[69] K. R. Islam, M. S. Ahmed, A. Ahamad, K. Fatima, T. A. Popy, B. Ryu, and M. T. Bin Iqbal, "A video-based physiotherapy exercise dataset," in *2022 25th International Conference on Computer and Information Technology (ICCIT)*, 2022, pp. 780–784.

This page is intentionally left blank

AUTHOR BIOGRAPHY

Personal Identity

Nama	:	Amik Rafly Azmi Ulya
Tempat Lahir	:	Jepara
Tanggal Lahir	:	14 November 1999
Alamat	:	Mangkuyudan, Kartasura, Sukoharjo, Jawa Tengah

Educational Background

2022-2024	:	Master Degree (S2), Department of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember
2018-2022	:	Bachelor Degree (S1), Departement of Physics, Faculty of Science and Data Analytics, Institut Teknologi Sepuluh Nopember
2015-2018	:	2nd Kudus Islamic State Senior High School (MAN)
2012-2015	:	Unggulan Pondok Modern Selamat Junior High School (SMP)
2006-2012	:	Al-Islam Kartasura Islamic Elementary School (MI)

Publication List

1. Amik Rafly Azmi Ulya et al. "HiroPoseEstimation: A Dataset of Pose Estimation for Kid-Size Humanoid Robot". In: Journal of Information Technology and Computer Science 8.3 (2023), 231–240. DOI: 10.25126/jitecs.202383568. URL: https://jitecs.ub.ac.id/index.php/ jitecs/article/view/568.

This page is intentionally left blank