3D Human Pose Estimation using WiFi CSI Data

João Pedro Martins Rocha
Master in Network and Information Systems Engineering
Department of Computer Science
Faculty of Sciences of the University of Porto
2025

3D Human Pose Estimation using WiFi CSI Data

João Pedro Martins Rocha

Dissertation carried out as part of the Master in Network and Information Systems Engineering

Department of Computer Science

2025

Supervisor

Prof. Dr. Hélder Filipe Pinto de Oliveira, Professor Auxiliar, Faculty of Sciences of the University of Porto

External Host Supervisor

Mestre Ana Filipa Rodrigues Nogueira, Assistant Researcher, Institute for Systems and Computer Engineering, Technology and Science (INESC TEC)

External Host Supervisor

Mestre Leonardo Gomes Capozzi, Assistant Researcher, Institute for Systems and Computer Engineering, Technology and Science (INESC TEC)

“ I can’t change the direction of the wind, but I can adjust my sails to always reach my destination. ”

Jimmy Dean

Acknowledgements

I would like to thank my supervisor Hélder Oliveira and my co-supervisors Ana Filipa Nogueira and Leonardo Capozzi for all the patience and assistance.

My friends, for all the pressure and jokes, it wouldn’t be possible without them.

My family, for their support, patience, and cheering me up.

Finally, a special thanks to my grandparents, Maria de Lurdes Cerqueira and Félix Duarte, for all the encouragement and for all that they taught me.

Resumo

O principal objetivo da Estimativa da Pose Humana em 3D é prever a posição das articulações do corpo usando dados de sensores, permitindo que os computadores analisem o movimento humano para uso em casos como cuidados de saúde ou vigilância. Tradicionalmente, as câmaras têm sido a escolha preferida para este tipo de tarefa, mas esta abordagem tem os seus próprios problemas, tais como oclusões, má iluminação e questões de privacidade, que dificultam a sua implementação. Com o objetivo de superar esses desafios, abordagens baseadas em Informação do Estado do Canal (IEC), que podem detetar as mudanças no sinal WiFi causadas por movimentos humanos, começaram a ganhar força. Devido ao facto de os routers WiFi serem dispositivos comuns, as soluções baseadas em IEC tornaram-se mais atraentes, pois podem lidar com as questões mencionadas anteriormente e oferecer uma solução económica. No entanto, ainda não é possível obter uma estimativa tão precisa da pose humana com IEC como com dados de imagem. Uma vez que atividades que envolvem múltiplas pessoas ou objetos, bem como a necessidade de lidar com os efeitos do desvanecimento e interferência multipath, são alguns dos problemas que os métodos baseados em IEC ainda enfrentam.

Portanto, o objetivo desta tese é contribuir para a pesquisa atual neste campo e propor uma nova solução para uma estimativa mais precisa da pose humana 3D usando dados IEC como entrada. Assim, vários estudos de ablação foram realizados com diferentes arquiteturas, técnicas de regularização, como batch normalization e dropout, número de camadas e mecanismos de atenção. A partir desses estudos, foi possível alcançar uma solução melhor combinando uma CNN com uma LSTM, aplicando batch normalization, duas camadas de dropout com uma taxa de 0.2, 6 camadas LSTM e um mecanismo de atenção com 4 cabeças de atenção e uma dimensão de modelo de 64.

Em seguida, foi realizado um estudo comparando diferentes modelos de aprendizagem profunda para reconhecimento de imagens, como InceptionV3, ResNet-18, GoogLeNet, utilizando espectrogramas criados com base nos dados IEC. No entanto, o modelo com melhor desempenho com espectrogramas não foi capaz de superar o modelo com melhor desempenho com dados IEC, apresentando um desempenho 0.212 pior para o limiar mais restrito em comparação com o resultado do CNN+LSTM usando dados IEC.

Assim, a melhor configuração de modelo alcançada foi o modelo CNN+LSTM mencionado, utilizando os dados IEC brutos como entrada. A comparação deste modelo com o estado da arte revelou que esta abordagem é melhor para a maioria dos limiares, sendo

apenas ligeiramente inferior para o limiar mais restritivo, com uma diferença de 0.047.

Palavras-chave: Estimativa de pose 3D, Informação de Estado do Canal (IEC), Visão Computacional, Aprendizagem de Máquina

Abstract

3D Human Pose Estimation's main objective is to predict the position of body joints using data from sensors, allowing computers to analyse human movement for use cases like healthcare or surveillance. Traditionally, cameras have been the primary choice for this kind of task, but this kind of approach has its own problems, such as occlusions, poor lighting, and privacy concerns, that hinder their implementation. In order to overcome those challenges, approaches based on Channel State Information (CSI), which can detect the changes in the WiFi signal caused by human movements, started gaining traction. Due to WiFi routers being common devices, CSI-based solutions became more appealing as they can handle the previously mentioned issues and offer a cost-effective solution. However, it is not yet possible to obtain as accurate a human pose estimates with CSI as with image data. Since activities involving several people or objects, as well as handling the effects of multi-path fading and interference, are some of the problems that CSI-based methods still face.

Therefore, the objective of this thesis is to contribute to the current research in this field and propose a new solution for a more accurate estimation of the 3D Human pose using CSI data as input. Hence, several ablation studies were performed with different architectures, regularisation techniques such as batch normalisation and dropout, number of layers and attention mechanisms. Out of these studies, it was possible to achieve a greater solution by combining a CNN with a LSTM, employing batch normalisation, two layers of dropout with a rate of 0.2, 6 LSTM layers, and an attention mechanism with 4 attention heads and a model dimension of 64.

Then, a study comparing different image recognition deep learning models, such as InceptionV3, ResNet-18, GoogLeNet, was performed using spectrograms created based on the CSI data. However, the best performing model using spectrograms, which was InceptionV3, was not able to overcome the best performing model using CSI data, giving a performance 0.212 worse for the stricter threshold compared to the result of the CNN+LSTM using CSI data.

Thus, the best model configuration achieved was the mentioned CNN+LSTM model using the raw CSI data as input. Comparing this model with the state-of-the-art revealed that this approach is better for most thresholds, being only slightly worse for the most restrictive threshold, with a difference of 0.047.

Keywords: 3D Human Pose Estimation, Channel State Information, Computer Vision, Machine Learning

Table of Contents

List of Tables

List of Figures

1. Introduction

This chapter provides the context and explains the motivation that guided this thesis. Following that, the research questions are discussed, and the goals are presented. Finally, this chapter concludes with a summary of the thesis's structure along with a short description of each chapter.

1.1. Context and Motivation

Human pose estimation is a computer vision task focused on detecting and estimating the positions of human body parts, such as arms, knees, and shoulders. The output is usually a set of keypoints that are connected to create a skeleton-like representation of a human. This task is typically handled by camera-based methods. However, these techniques have limitations in certain environmental conditions, such as poor lighting, occlusion, or when privacy is a concern [1, 2].

Researchers have explored multiple solutions, and one of those solutions is using WiFi Channel State Information (CSI). It measures small variations in wireless signals as they propagate between the transmitter and the receiver. When an individual moves through the environment, the WiFi signal is reflected and scattered when in contact with the human body. These variations in the signal carry information about the body position and movement, allowing algorithms to reconstruct the poses without the need for cameras.

Compared to camera-based methods, WiFi CSI offers some benefits. It is not affected by poor lighting and can work through visual obstructions like walls. It also handles privacy concerns, since it does not capture human identifiable information, making it less intrusive in sensitive cases such as healthcare or smart home systems.

1.2. Research questions and Objectives

The main goal of this thesis is to develop an algorithm for 3D Human pose estimation using WiFi CSI data. To accomplish that, the following research questions have to be tackled:

- What are the limitations and challenges of the current methods developed to use WiFi CSI data for estimating the 3D human pose?

- What architectures, regularisation techniques or data preprocessing methods can be applied to develop an accurate solution based on WiFi CSI data for estimating the 3D human pose?

Hence, the next objectives were delineated to properly address these research questions and fulfill the main goal of this thesis:

- Deepen the knowledge on 3D human pose estimation using WiFi CSI data as input by critically reviewing the current research on this field and analysing their strengths and shortcomings.
- Establish a baseline through a comparative evaluation of different architectures for reconstructing the 3D human poses using the WiFi CSI data.
- Evaluate the use of distinct regularisation approaches, data preprocessing methods or other machine learning techniques to enhance the accuracy and efficiency of the baseline.
- Create a new solution for reconstructing the 3D human pose using the WiFi CSI data based on the attained knowledge.
- Compare the newly developed solution with the state-of-the-art methodology.

1.3. Outline of the thesis

The rest of the thesis is structured as follows:

- **Chapter 2**: Presents concepts and contextualization related to 3D Human Pose Estimation with WiFi CSI.
- **Chapter 3**: Provides a state-of-the-art review of the main research works and subjects related to this thesis.
- **Chapter 4**: Presents and explains the experimental setup used in the developed work.
- **Chapter 5**: Describes the experiments and presents the results.
- **Chapter 6**: Presents the conclusions of this thesis and the possible future work directions.

2. Background

In this chapter, some basic concepts related to 3D Human Pose Estimation (HPE) with Wireless Fidelity (WiFi) CSI will be presented, and some contextualization will be provided to help understand the following chapters.

2.1. Wireless Fidelity

WiFi is the technology responsible for wireless connection of devices to the internet. It uses radio waves to send data across short distances in the 2.4 GHz and 5 GHz frequency bands. WiFi has become widely used in homes, businesses, and public spaces, providing a convenient and efficient means of accessing network resources without physical cables [3].

2.1.1 Orthogonal Frequency Division Multiplexing

Orthogonal Frequency Division Multiplexing (OFDM) is a digital modulation technique used in communication systems. It improves data transmission efficiency and robustness, specially in environments with multipath propagation and interference.

It splits high-speed data streams into multiple lower-speed sub-streams, which are transmitted simultaneously over different frequencies known as subcarriers. Each subcarrier is orthogonal to the others, meaning their cross-correlation is zero, minimizing interference and maximizing spectral efficiency.

Its ability to transmit data easily and reliably over various channels makes it a vital component in wireless communication technologies [3].

2.1.2 Channel State Information

WiFi CSI provides detailed information about the state of a wireless channel. It captures the amplitude and phase of the signal across multiple subcarriers, providing a more detailed view of how a signal propagates through a wireless environment.

It's derived from the OFDM technology used in modern CSI standards such as Institute of Electrical and Electronics Engineers (IEEE) 802.11n, 802.11ac, and 802.11ax. CSI data includes the channel properties for each of these subcarriers, offering a comprehensive understanding of the wireless channel.

Its applications span from enhancing communication systems to enabling advanced location-based services and activity recognition, making it a powerful tool in modern wireless research and development [3].

2.2. Artificial Intelligence

Artificial Intelligence (AI) is a multidisciplinary field concerned with the study and development of computer systems capable of acquiring knowledge through learning and observation, reasoning with that knowledge, and making decisions, mimicking human cognitive behaviour [4].

2.2.1 Machine Learning

Machine Learning is a field of artificial intelligence that focuses on the design of algorithms that allow computers to learn from data and make decisions based on it. Unlike traditional programming, where specific instructions are given to the computer, machine learning focuses on developing systems that can automatically improve based on prior experience. To accomplish that, it relies on various learning techniques:

- **Supervised learning**: Algorithms that are trained with the help of labelled data to map the input and output. It is particularly useful for classification and regression purposes. For instance, a supervised learning algorithm might learn to categorise emails as spam or not based on labelled examples [5].
- **Semi-supervised learning**: Algorithms that make use of both labelled and unlabeled data to improve learning accuracy. The main principle is that the structure in the unlabeled data can provide additional information about the class distribution, which can help the model make a better generalisation. This approach is useful when labelled data is limited or difficult to obtain, allowing the model to take advantage of the abundant unlabeled data to improve its performance on tasks like classification and regression [6].
- **Unsupervised learning**: Algorithms with the objective to find hidden patterns or structures in unlabeled data. Clustering and dimensionality reduction are common techniques in unsupervised learning, often used for tasks such as data visualization [5].

- **Reinforcement learning**: Alternative approach where agents are able to learn to make decisions by interacting with an environment. Through a combination of rewards and penalties, the agent improves its performance over time [5].

2.2.1.1 Transfer Learning

Transfer learning is a machine learning technique that uses knowledge acquired from solving one problem and applies it to a different but related problem. It is useful when the new task has limited labelled data, but there is data available for a similar task. In deep learning, models trained on large datasets can be adapted to new applications by reusing the learned representations [5].

2.2.1.2 Deep Learning

A subgroup of machine learning, focuses on the use of artificial neural networks to model and solve complex problems. This approach mimics the human brain's structure and functionality, allowing computers to learn from large amounts of data through multiple layers.

Deep learning relies on the creation of neural networks with multiple hidden layers, where each layer consists of neurons that process input data and pass the results to the next layers. This hierarchical structure enables the network to automatically learn and extract complex patterns and features from the data.

The training process is driven by algorithms such as backpropagation, which adjusts the weights of the network by minimizing a loss function. This optimization is achieved through gradient descent methods, where the model iteratively improves its predictions by reducing the error between the predicted and actual outputs.

Its performance is dependent on two critical factors: data and computational power. The availability of large datasets provides the necessary fuel for training deep networks, while advances in hardware, like Graphic Processing Unit (GPU), have drastically improved the training process [5].

2.2.1.3 Loss Function

Loss function is a function that quantifies the difference between the predicted values and the true values. This difference reflects how far off the model is from the ideal solution.

It is important to fine-tune the model parameters in order to attain the optimal predictions [5].

2.2.1.4 Gradient Descent

Gradient descent is a crucial optimization algorithm in machine learning and deep learning, essential for training models by minimizing the loss function. It operates by iteratively adjusting the model parameters to find the optimal values that reduce the error between the predicted and actual outputs [5].

2.2.1.5 Batch Normalization

Batch normalization is a technique used to improve the training of deep neural networks by normalizing the inputs to each layer. It addresses problems where the distribution of each layer's input changes during training, which can slow down learning and require careful parameter initialization. By normalizing layer inputs, batch normalization allows higher learning rates, reduces sensitivity to initialization, and in some cases, can reduce the need for other regularization techniques like dropout.

Additionally, batch normalization introduces a slight regularizing effect, as the statistics, mean and variance, are computed over each minibatch, adding some noise into the training process, which can help improve generalization [5].

2.2.1.6 Dropout

Dropout is a regularization technique introduced to reduce overfitting in deep neural networks. The key idea is to randomly drop units, called neurons, along with their connections from the neural network during training. This means that, in each training iteration, each neuron is kept with a fixed probability or is temporarily removed from the network, effectively creating a different network in each iteration [5].

2.2.1.7 Neural Network

Neural networks are machine learning models inspired by the human brain's structure and function. They are composed of interconnected layers of artificial neurons, which process input data to extract features and make predictions. Neural networks form the foundation of deep learning, enabling the modelling of complex patterns and relationships in data [5].

A typical neural network consists of three types of layers:

- **Input Layer**: The first layer receives the raw input data. Each neuron in this layer represents a feature of the input.
- **Hidden Layers**: These layers are situated between the input and output layers. Neurons in hidden layers apply transformations to the inputs they receive, capturing complex patterns. A neural network can have one or more hidden layers, and the term *Deep Learning* refers to networks with multiple hidden layers.
- **Output Layer**: The final layer produces the network's predictions. The number of neurons in this layer corresponds to the number of predicted classes or output values.

2.2.1.8 Convolutional Neural Network

A type of deep learning model, Convolutional Neural Network (CNN), see Figure 2.1, works by arranging layers in an organized manner, mainly for the purpose of analyzing input data, especially images. Typically, this type of network is composed of convolutional, pooling and fully-connected layers, which are crucial for feature extraction, dimensionality reduction, and classification [7].

Convolutional Layers: In the initial stage, filters, also known as kernels, are applied to the input data. Each filter traverses the data using a sliding window, doing element-wise multiplication with the local region. With this approach, feature maps are produced to highlight important patterns like edges, textures, and shapes. To enable the network to learn a variety of representations, multiple filters are applied to capture different features that exist in the input data.

Pooling Layers: Following convolutional layers, sometimes pooling operations are employed to reduce the spatial dimensions of the feature maps while retaining essential information. Common pooling techniques include max pooling, which selects the maximum value within each local region, and average pooling, which computes the average value. By downsampling the feature maps, pooling layers help reduce computational complexity and extract robust features.

Fully Connected Layers: After multiple convolutional and pooling layers, the resulting feature maps are flattened into a vector and fed into one or more fully connected layers. These layers serve as the final stages of the network and perform classification based on the learned features. By mapping the input data to output classes through weighted connections, fully connected layers enable the network to make predictions or decisions.

During this phase, CNNs utilize backpropagation, an optimization algorithm, to adjust the weights of the filters and the fully connected layers. This adjustment aims to minimize a defined loss function, reflecting the disparity between predicted and actual outputs. Through iterative optimization, the network learns discriminative features from the input data and refines its parameters, gradually improving its performance over time.

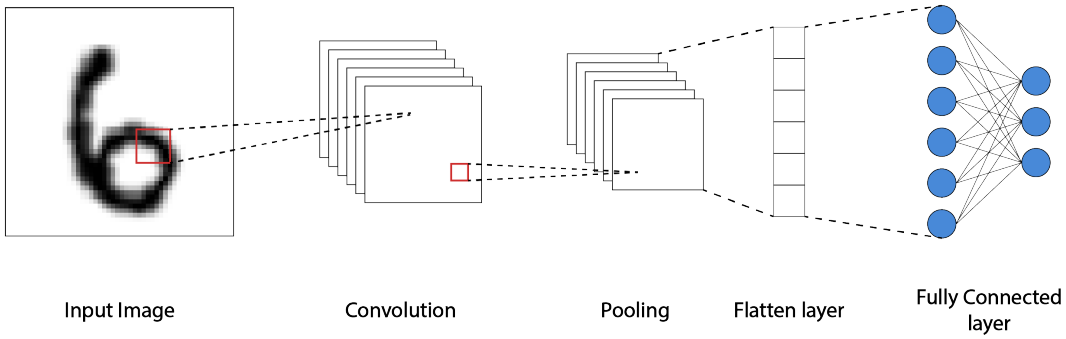

FIGURE 2.1: Convolutional Neural Network Structure

Overall, CNNs leverage the hierarchical arrangement of layers to automatically learn relevant features from raw input data, making them highly effective for tasks such as image recognition, object detection, and image segmentation [5].

2.2.1.9 Recurrent Neural Network

A Recurrent Neural Network (RNN), see Figure 2.2, is a type of neural network designed to process sequential data by maintaining an internal state of memory. RNNs are integral to deep learning and are particularly well-suited for tasks involving sequential data, such as time series prediction, Natural Language Processing (NLP), and speech recognition.

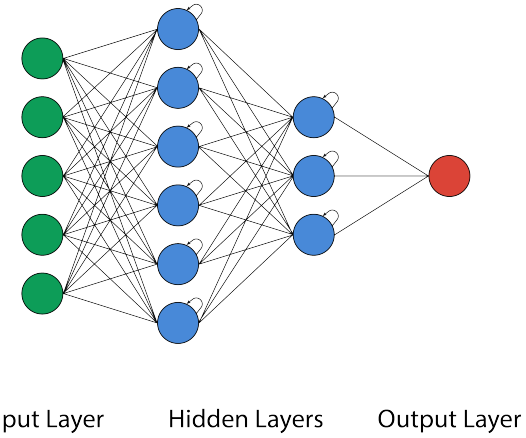

FIGURE 2.2: Recurrent Neural Network Structure

RNNs are a fundamental building block for processing sequential data. They can be stacked to form deeper architectures or combined with other types of neural networks, such as CNNs, or attention mechanisms, to tackle complex tasks. However, traditional RNNs suffer from the vanishing gradient problem, where gradients diminish exponentially over long sequences, limiting their ability to capture long-term dependencies.

To address this limitation, more advanced variants of RNNs have been developed, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. These models incorporate gating mechanisms that regulate the flow of information through the network, allowing them to capture long-range dependencies more effectively. These advanced RNN architectures have become indispensable tools in deep learning, enabling breakthroughs in areas such as natural language understanding, machine translation, and speech synthesis [5].

2.2.1.10 Long Short-Term Memory

In deep learning, LSTM networks, see Figure 2.3, are a type of RNN that specializes in sequential data processing, such as language translation and time series analysis. Their ability to capture long-term dependencies makes them particularly well-suited for tasks requiring context understanding across extended sequences of data.

Memory Cells: LSTMs feature memory cells that can maintain information over extended periods. These cells help overcome the vanishing gradient problem encountered in traditional RNNs, enabling the network to capture long-term dependencies in sequential data.

Gating Mechanisms: Crucial components responsible for regulating the flow of information within the network's memory cells. These mechanisms consist of three types of gates:

- **Forget Gate**: Determines what information from the previous cell state should be discarded based on the current input.
- **Input Gate**: Controls the flow of new information into the cell state, deciding which parts of the current input and candidate values should be added to the cell state.
- **Output Gate**: Determines which parts of the cell state should be output to the next layer or used for the final prediction.

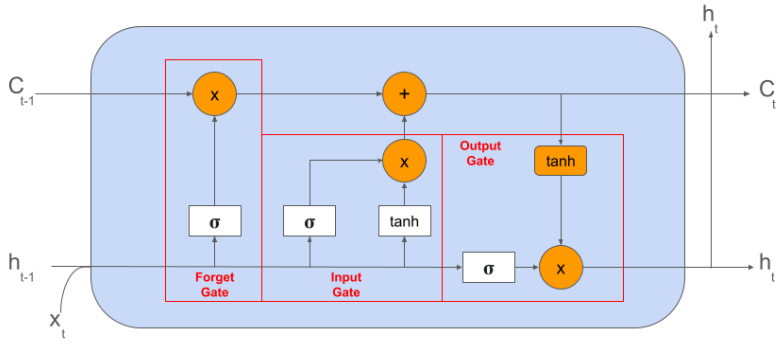

FIGURE 2.3: Long Short-Term Memory Structure

Overall, LSTM networks stand out in deep learning as a specialized form of RNN tailored for sequential data processing. With their capacity to capture long-term dependencies, they excel in tasks like language translation and time series analysis, where understanding context across extended sequences is essential. By incorporating memory cells and gating mechanisms, LSTMs address the limitations of traditional RNNs, enabling more effective learning and inference in complex sequential data scenarios [5].

3. State-of-the-Art

The combination of wireless technology and HPE has been receiving increasing attention in recent years, giving new perspectives on human movement [8]. WiFi CSI is growing as a tool that allows the investigation of the details of various human body postures. This section digs into the present state of research, tracking its recent developments, outlining key methodologies, and revealing the potential for advancing our comprehension of 3D HPE.

3.1. Human Pose Estimation

Human pose estimation went through significant changes over the decades. Initial efforts in the field were focused on the development of methods for tracking and analysing simple movements using elementary image processing techniques [9]. These systems often relied on marker-based approaches, where physical markers were fixed onto the subject's body and subsequently tracked. As technology progressed, more markerless approaches began to emerge, using complex algorithms to identify and contextualize human poses directly from video footage [10, 11].

The integration of depth sensors, like Microsoft Kinect [12], has led to major advances in HPE. These sensors made it feasible to keep track of body movements in real-time without being intrusive, which increased their accessibility and range of use. However, there were still issues to take into account, like occlusion and accuracy [13].

Computer vision algorithms and machine learning approaches started gaining prominence, allowing systems to learn and infer human poses from visual data. Systems based on rules are used to provide a way towards more flexible and data-driven approaches, that have increased robustness and accuracy in estimating different poses in a variety of environments [14].

The introduction of WiFi CSI data has given pose estimation a new dimension. Since WiFi CSI data has distinct characteristics, it has some advantages compared to images for pose estimation. While objects can look distorted in images, as cameras rely on what is in their field-of-view as well as on the lighting and backgrounds; WiFi CSI data uses the electromagnetic spectrum to transmit and receive information in the radio frequency range. This makes it less disturbed by obstacles and changes in light. In addition, it has

the ability to capture even small changes, such as faint motions, that could be missed in images. Also, it ensures confidentiality because it doesn't capture visual details [1].

3.2. Current Research

In the current body of research focused on the estimation human poses using WiFi CSI, numerous models have been put forward and investigated. These models use a range of methods, spanning from advanced deep learning structures to probabilistic approaches [15].

The models presented in Table 3.1 are a selection of researchers' ongoing efforts to investigate and enhance various methodologies, representing a subset of the diverse approaches in current literature for estimating the 3D human pose using WiFi CSI. To fully exploit the spatial and temporal complexity of the data, selecting the appropriate architecture is crucial.

CSI-Former [1] and **Wi-Pose** [16] used CNNs as part of its architecture, bringing its capabilities in feature extraction from complex data, facilitating accurate pose estimation. However, they still had some limitations, preprocessing requirements and computational complexity remain notable considerations in implementing CNN-based approaches.

Person-In-Wifi [17] added U-Net to the CNN architecture, to address challenges like capturing long-term dependencies. This approach demonstrated promising results in extracting spatial features across different scales, but may increase computational costs and require large amounts of labelled data for training.

WiLISensing [18] used a CNN along with Transfer Learning to recognize activities in a position without training or with very few training samples. This method made a good trade-off between performance and quantity of training data.

WiPose [8], **Wi-Mose** [19] and **GoPose** [13] leveraged architectures that combined CNN and LSTM, enabling the CNN to extract spatial features and the LSTM to model temporal dependencies, leading to improved performance. Nevertheless, extensive training durations and meticulous parameter adjustment may be necessary for CNN and LSTM model training.

Danger-Pose [20] used Support Vector Machine (SVM)-based classifier as an anomaly-detection method. This approach was chosen because CSI data in danger situations is difficult to collect, and SVM require a small amount to no data collected in danger situations

for training. However, SVMs may not be well-suited for large datasets with many features, since the model training can be very slow and can consume a large amount of memory when the dataset has many features [20].

Winect [15], **WiLink** [21], **WiSPE** [22] and **Deep Transfer Learning Fall Detection Learning for Actions Recognition** [23] used Residual Neural Network (ResNet) which have advantages in terms of feature learning and capacity. To avoid overfitting, deep ResNet architecture training may need a significant amount of processing power and careful regularization.

TABLE 3.1: Summary of the research works found for HPE using WiFi CSI

System	Neural Network	Input Features	Dataset	Evaluation Metrics
GoPose [13]	CNN+LSTM	Angle-of-Arrival (AoA) spectrums	A total of 10 participants and divided into two non-overlapping datasets: a training set with 8 participants and a test set with the 2 remaining participants	Euclidean distance between the predicted joint location and the ground-truth
CSI-Former [1]	Performer (Transformer-based architecture) + CNN	Amplitude and Phase	Composed of 12 volunteers with different weights and heights. Each volunteer participates in 12 different actions.	Percentage of Correct Keypoints (PCK)
Person-In-WiFi [17]	CNN + U-Net	Amplitude	Data acquired from 6 laboratory scenarios and 10 scenes in the scenario room using 8 volunteers	PCK
Winect: 3D Human Pose Estimation for Free-form Activity Using WiFi Signals [15]	ResNet-18	AoA	Data acquired from 6 volunteers who performed 20 minutes of free-form activity. More than 900 activities total.	Euclidean distance between the predicted joint location and the ground-truth
Danger-Pose Detection System [20]	Support Vector Machine-based classifier	Amplitude and Phase	Data collected in a bathroom of an apartment, six days in total. Five-day data was used to train one-class SVM classifiers and one-day data was used to evaluate their performance.	Precision, Recall and F1-score

Continued on next page

System	Neural Network	Input Features	Dataset	Evaluation Metrics
WiSPE: A COTS WiFi Based 2-D Static Human Pose Estimation [22]	ResNet: One input layer, four residual modules and one linear fully connected layer	AoA images	Data collected from 6 volunteers in 3 scenarios and contains 5 different poses: Standing, Akimbo, Raising arm flat, Raising arm up and Sitting	PCK
WiLISensing [18]	CNN	Amplitude	6 volunteers performed 6 activities in 36 different positions and at least 50 samples in each position for each activity	Precision, Recall and F1-score
Deep Transfer Learning Fall Detection Learning for Actions Recognition [23]	ResNet-18	Amplitude and Phase	7 actions in 3 different environments	Accuracy
WiPose: Towards 3D Human Pose Construction Using WiFi [8]	CNN+LSTM	Amplitude	CSI data was collected to generate the poses of 10 users. For each activity, the first 70% of the data samples were used for training and the remaining 30% were used for testing.	Euclidean distance between the predicted joint location and the ground-truth
Wi-Mose: From Point to Space: 3D Moving HPE Using Commodity WiFi [19]	CNN+LSTM	Amplitude and Phase	10 hours of data acquired from 5 volunteers with different heights, weights, genders and clothing.	Procrustes Analysis-Mean Per Joint Position Error (P-MPJPE)
WiLink: Link Selection-Based 3D Human Pose Estimation Using Commodity WiFi [21]	ResNet-18: 6 Residual blocks and 2 Fully Connected Layers	Amplitude and Phase	Data acquired from 5 volunteers, they walked freely and 10 hours of data were collected.	P-MPJPE
Wi-Pose: From Signal to Image [16]	CNN	Amplitude and Phase	5 hours of data acquired from 5 volunteers with different heights, weights, genders and clothing.	Percentage of Correct Skeletons (PCS)

In summary, each approach has its distinct benefits and drawbacks, and the choice of strategy depends on factors such as the nature of the data, computational resources, and the particular demands of the pose estimation task. Future research may explore hybrid

approaches that combine multiple techniques to utilize their complementary strengths and further enhance pose estimation accuracy and efficiency in real-world applications.

3.3. Improvements and Innovation

Improving human pose estimation with CSI models is a complex attempt that requires the exploration of various methodologies, each offering unique advantages and challenges.

Firstly, Data Augmentation techniques introduce variations in WiFi signals like noise and interference, aiming to facilitate better generalization to real-world scenarios [24]. Additionally, designing specialized neural network architectures, such as hybrid models combining CNN and RNN, can effectively capture both spatial and temporal dependencies inherent in human movements [14]. Leveraging Transfer Learning from pre-trained models on related tasks accelerates training and boosts performance, particularly when labeled WiFi CSI datasets are limited [18]. Integrating information from multiple modalities and robust feature extraction methods tailored for WiFi CSI data also contribute to improved accuracy. Attention mechanisms transform the processing of CSI data by enabling models to selectively focus on salient spatial and temporal features crucial for understanding human movement dynamics [1]. These mechanisms dynamically allocate attention across different regions of the CSI data, allowing models to distinguish subtle nuances in pose variations with unprecedented precision and adaptability.

Each approach presents its own set of challenges and considerations, requiring careful optimization and trade-off assessment. Despite this, the integration of these diverse strategies offers a comprehensive approach to advancing human pose estimation with WiFi CSI models, promising more accurate and reliable results across various applications and scenarios.

3.4. Conclusion

In this chapter, the models and techniques from existing works are analysed in order to find their performance, possible issues, and potential scope for improvement. Not only does this examination help in understanding what these models can do, but it also helps in figuring out how to improve them in the future. This evaluation process involves carefully looking at the pros and cons, which helps make better decisions, improve the models, and think about how to create a model adequate for figuring out how people are positioned using WiFi CSI.

The advancement of modern computing technologies, and, in particular, the application of CNN and LSTM components played a major role in improving the performance of pose estimation applications. CNNs are powerful in spatial feature extraction from the CSI data, while LSTMs are good at temporal feature capture, thus both are appropriate for dynamic pose estimation. Additionally, the use of complex structures like ResNets has made the models deeper and more powerful in performance for effective feature representation, thus resulting in more accurate pose estimation.

However, these developments also have problems: the scarcity of data, the complexity of the human pose and the high processing power were the main challenges found by the researchers.

The objective of this state-of-the-art is to provide an overview of what was and is being done and the main challenges in the field of HPE with WiFi. Generally speaking, the use of WiFi CSI with deep learning techniques provide an innovative, non-invasive and low-cost approach for HPE compared to traditional vision-based methods. Therefore, as the research keeps going, further optimization and real-world application of the models promise to bring greater advantages in this area.

4. Experimental Setup

In this chapter, an overview of the fundamental components employed in this study is presented. Firstly, the background and details of the dataset are explained. Following this, there is a description of the deep learning models used, along with an explanation of their architecture. The chapter also presents the optimization function that guides the learning process. Lastly, the evaluation metrics that were used to measure models' performance during the experiments are presented and described.

4.1. Dataset

The Wi-Pose dataset integrates 5 GHz WiFi CSI with synchronized human pose images and annotated skeleton points. Collected from 12 volunteers performing 12 different actions, the dataset includes approximately 166600 data points, with an 80-20 split for training and testing. Each action is repeated ten times per volunteer, lasts about 5 to 6 seconds. The CSI data, captured by a three-antenna WiFi setup, is aligned with images taken at 20 Hz and annotated to identify 18 key skeleton points. The CSI data has the format of 5×30×3×3, where:

- 5 represents the **5** CSI data corresponding to 1 image
- 30 represents the **30** sub-carriers
- 3×3 represents the **3** WiFi transmitting antennas and receiving antennas

The dataset addresses limitations of traditional image-based pose estimation, especially under poor lighting and privacy concerns, and is publicly available to support further research in WiFi-based pose estimation [1].

4.2. Deep Leaning Models

Deep learning models are the basis of this study, providing a means to study and interpret complex data. In this section, the deep learning models used in this thesis will be described and explained.

4.2.1 Combination of CNN and LSTM networks

CNNs are good at extracting spatial features from data, and also good at finding local patterns and hierarchical features, both of which are important for recognition and localization of body parts.

On the other hand, LSTMs are designed for sequential data, so, they can capture temporal dependencies. In human pose estimation, understanding movement over time is crucial. LSTMs model the sequence of poses and predict future positions of the body joints from the previously learned information. Due to the fact that information is kept for a long period of time in their memory cells, these make them very fit for tracking and predicting human movements that have long-term dependencies and are very complex.

Therefore, the system benefits of bringing in the spatial feature extraction power of CNNs together with the strengths of LSTMs in modeling temporal sequences. This way, the model learns about the static structure of the spatial body and the dynamic behavior in time, giving more accuracy to the pose estimation. While CNNs focus on finding the correct position of each body part in a given frame, LSTMs provide further information to ensure that the poses generated by the network form a coherent sequence over time [8, 13, 19].

Therefore, by using a CNN followed by a LSTM, the CNN extracts higher-order features from the preprocessed data. Then, the obtained features are fed to a LSTM network to model the temporal relationships in the data for pose sequence prediction. Using the strengths of both CNNs and LSTMs, the resulting system performs robust and precise human pose estimation. This ensures that the system is able to run effectively in a dynamic environment.

4.2.2 GoogLeNet

It is a deep learning model developed for image recognition tasks. This type of CNN was designed by Google researchers back in 2014 and, along with its architecture, became an important breakthrough for the entire field. Unlike previous models, GoogLeNet was designed to be used with Inception modules, see Figure 4.1, in its architecture; thus, performing convolution operations at several scales simultaneously, leading to faster performances and more feature-descriptive capabilities.

GoogLeNet's architecture is efficient due to the minimal number of parameters, which is the result of the dimensionality reduction occurring in the Inception modules. This design choice allows it to build up a very deep network with low computational cost, which is practical and important for applications such as image classification or object detection [25].

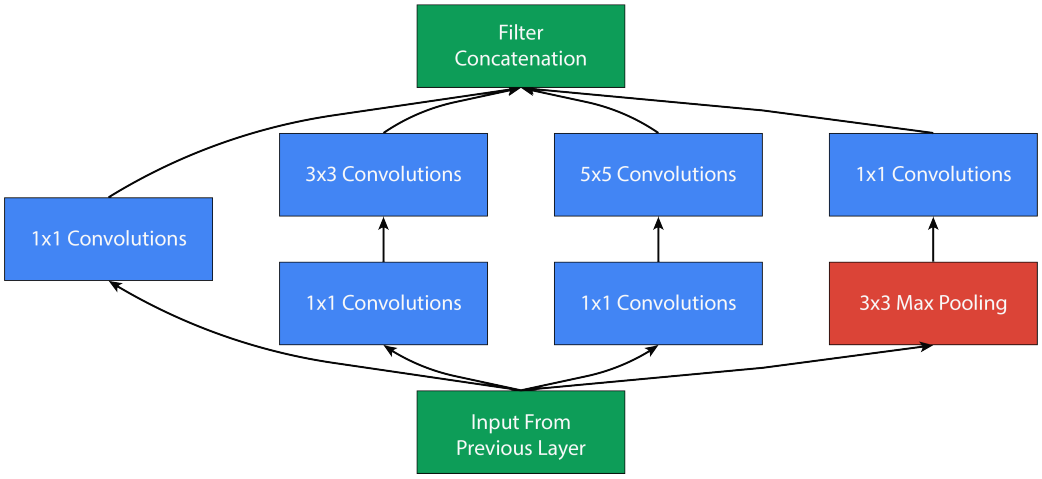

FIGURE 4.1: Example of Inception Module

4.2.3 Inception V3

It is an extension of the GoogLeNet architecture aimed at improving computational performance and model accuracy in computer vision tasks. It uses factorized convolutions, where larger convolutions are decomposed into smaller and easier computations. This reduces the computational cost and the number of parameters significantly, with only a minor degradation in performance.

This model incorporates auxiliary classifiers to improve convergence and act like regularizers in order to improve the overall performance of the network. Label smoothing is also introduced to avoid overfitting and to make the model less confident about its predictions, hence improving generalization. Inception V3 architecture strikes a balance between depth and width and employs various design principles in regard to the optimization of network structure for large-scale image recognition tasks [26].

4.2.4 ResNet

ResNets aim to improve the training of deep neural networks by introducing a residual learning framework. The key idea is to reformulate the layers to learn residual functions

with reference to the layer inputs rather than learning unreferenced functions. This approach helps to solve vanishing gradients and also degradation problems that occur when more layers are added, resulting in higher training errors.

In ResNets, the shortcut connections skip one or more layers, helping to create identity mappings, see Figure 2.2. These shortcuts add no extra parameters or computational complexity, making the network easier to optimize. Additionally, it becomes possible to train deeper models compared to the standar vai dormir d convolutional neural networks [27].

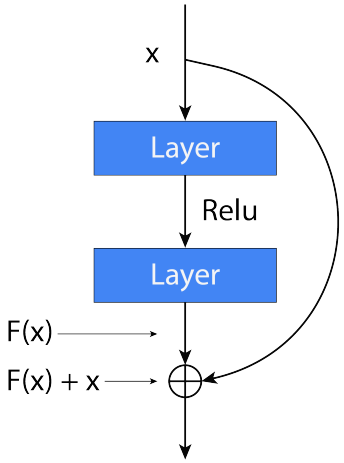

FIGURE 4.2: Residual Block

4.3. Optimization function

Optimizing a neural network involves adjusting its weights to minimize the loss function. Different optimization algorithms can be used to achieve this goal, each with its unique approach and benefits.

4.3.1 Adam

Adam is an adaptive learning rate optimization algorithm. This optimization function computes adaptive learning rates for each parameter, thereby helping to adjust the learning rate during training for better convergence. It maintains exponentially decaying averages of past gradients and squared gradients, smoothing the updates. Additionally, Adam includes bias correction to offset the initialization bias of the momentum estimates, which is very critical at the initial layers of training. This optimizer is computationally efficient and requires little memory, making it suitable for large-scale problems. It also performs well

with sparse gradients and non-stationary objectives, common in machine learning tasks [28].

4.4. Loss function

In machine learning, the loss function is a mathematical function that is used to numerically quantify the difference between the predictions of the model and the actual data. The objective of the model during training is to minimize the value of this function.

In the context of human pose estimation, this loss function calculates the difference between predicted and ground-truth human poses. Specifically, the loss function used in this thesis was a weighted mean squared error between the predicted and actual keypoint coordinates. The weight is applied to each keypoint, taking into account their confidence. In case the ground-truth confidence is high, the keypoint should have a significant impact on the calculation because the model should be able to predict it. Otherwise, it should have a low impact on the calculation, as this keypoint is likely incorrect; therefore, the model should not be penalised for being wrong.

The function can be defined by Eq. 4.1:

$$L = \frac{1}{2 \times N_i \times N_k} \sum_{i=1}^{N_i} \sum_{k=1}^{N_k} \left(G_c'^{i,k} \times \left(S_x'^{i,k} - G_x'^{i,k} \right)^2 + G_c'^{i,k} \times \left(S_y'^{i,k} - G_y'^{i,k} \right)^2 \right) \tag{4.1}$$

where:

- N_i represents the number of keypoints.
- N_k represents the number of samples in the batch.
- $G_c'^{i,k}$ represents the Alphapose confidence c of keypoint i belonging to the sample k.
- $S_x'^{i,k}$ represents the prediction for the *x* coordinate of the keypoint *i* belonging to the sample *k*.
- $G_x'^{i,k}$ represents the ground-truth for the *x* coordinate of the keypoint *i* belonging to sample *k*.
- $S_y'^{i,k}$ represents the prediction for the *y* coordinate of the keypoint *i* belonging to the sample *k*.
- $G_y'^{i,k}$ represents the ground-truth for the *y* coordinate of the keypoint *i* belonging to the sample *k*.

4.4.1 Confidence threshold

It's a value used in machine learning models to determine when a prediction is reliable. This threshold sets the minimum probability score that a prediction must achieve to be considered trustworthy. For example, if the threshold is set to 80%, the model will only consider predictions with a confidence level of 80% or higher as correct. Predictions below this confidence level are deemed invalid. Adjusting the confidence threshold affects the model performance by influencing the balance between accuracy and reliability. Increasing the threshold can reduce incorrect predictions but might miss some true cases, while decreasing it can capture more true cases, but might increase incorrect predictions.

4.5. Evaluation metrics

Evaluating the performance of a pose estimation model requires precise and relevant metrics. These metrics help to understand how well the model performs under different conditions and with various datasets.

4.5.1 Percentage of correct keypoints (PCK)

PCK is a metric for evaluating pose estimation models. It measures the percentage of correctly predicted keypoints within a certain distance from the ground-truth. It is calculated by determining the distance between predicted keypoints and the ground-truth keypoints, normalized by a reference length, often the length of the torso or a key body part. The PCK@a_j metric measures the percentage of predicted keypoints that fall within a certain threshold distance, a_j. The formula for PCK is given by Equation 4.2.

$$\text{PCK@}a_j = \frac{1}{N}\sum_{i=1}^{N}\delta\left(\frac{\|S_i^k - G_i^k\|_2}{\text{TD}_k} \leq a_j\right) \tag{4.2}$$

- $\|S_i^k - G_i^k\|_2$ represents the Euclidean distance between the predicted, S_i^k, and ground-truth, G_i^k, keypoints for the i-th sample of the k-th skeleton.
- TD_k represents the torso diameter for the k-th skeleton.
- a_j represents the threshold within which the keypoints are considered correct.
- N represents the number of keypoints.
- δ represents the indicator function that returns 1 if the condition is true and 0 otherwise.

The set thresholds considered were values between 5 and 50 to demonstrate the performance of the models from different evaluation scales. The higher the threshold, the wider the error margin allowed for skeleton point estimation. For this reason, increasing the threshold may lead to higher PCK. However, lower thresholds represent more strict evaluation criteria, meaning that they can demonstrate models' performance better.

5. Results and Discussion

In this chapter, the experimental work is presented and analysed. The experiments were developed in a Python v3.10 environment along with PyTorch (Compute Unified Device Architecture (CUDA)) v11.8. Each model training was performed on one 8GB GPUs and took less than 72 hours. Experiments were conducted with two types of data: CSI and spectrograms.

Hence, the chapter presents the results of the studies conducted for each model architecture and ends with a discussion of the obtained results.

5.1. Confidence Threshold

This section describes how different confidence thresholds impact the accuracy and reliability of pose estimation. A balance between accuracy and robustness can be found by analysing performance across different threshold levels.

For the selection of which confidence threshold to use, five options were considered: 0%, 60%, 70%, 80% and 90% thresholds. Out of these experiments, it was possible to understand that the confidence threshold depends on specific goals and constraints, see Table 5.1.

- **0% threshold** is impractical for most use cases due to its high false positive rate.
- **60% threshold** is the value that provides a more balanced approach for general use.
- **70%** and **80% thresholds** are better suited for tasks requiring greater precision.
- **90% threshold** is reserved for scenarios where false positives are intolerable, but it comes at the cost of significantly reduced recall.

Table 5.2 compares model performance across two confidence thresholds, 0 and 0.6, for three architectures: MLP, CNN and LSTM, evaluating both precise (PCK@5) and permissive (PCK@50) accuracy. MLP is composed of two linear layers with ReLU as the activation function, CNN is composed of two convolutional layers with ReLU as the activation function and one linear layer, and LSTM is composed of three LSTM layers and one linear layer.

TABLE 5.1: Comparison of Confidence Thresholds

Threshold	Precision	Recall	False Positives
0%	Very Low	100%	Very High
60%	Moderate	High	Moderate
70%	High	Moderate	Low
80%	Very High	Low	Very Low
90%	Extremely High	Very Low	Extremely Low

TABLE 5.2: Results for the MLP, CNN and LSTM models using CSI data as input and considering a confidence threshold of 0 and 0.6

Keypoints	Confidence Threshold 0						Confidence Threshold 0.6					
	PCK@5			PCK@50			PCK@5			PCK@50		
	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM
Nose	0.072	0.090	**0.303**	0.836	0.838	**0.874**	0.073	0.092	**0.308**	0.840	0.841	**0.877**
Neck	0.083	0.094	**0.313**	0.875	0.875	**0.896**	0.089	0.102	**0.355**	0.886	0.883	**0.906**
RShoulder	0.074	0.092	**0.281**	0.864	0.867	**0.888**	0.075	0.094	**0.289**	0.862	0.865	**0.887**
RElbow	0.034	0.040	**0.138**	0.797	0.804	**0.850**	0.035	0.042	**0.148**	0.806	0.814	**0.862**
RWrist	0.019	0.040	**0.100**	0.689	0.804	**0.782**	0.020	0.020	**0.102**	0.700	0.711	**0.793**
LShoulder	0.071	0.077	**0.290**	0.869	0.867	**0.889**	0.086	0.098	**0.408**	0.904	0.892	**0.908**
LElbow	0.063	0.019	**0.268**	0.850	0.698	**0.876**	0.082	0.093	**0.359**	0.885	0.880	**0.905**
LWrist	0.052	0.076	**0.263**	0.836	0.867	**0.870**	0.064	0.095	**0.343**	0.897	0.893	**0.918**
RHip	0.096	0.073	**0.307**	0.908	0.848	**0.920**	0.133	0.158	**0.471**	0.959	0.956	**0.958**
RKnee	0.127	0.121	**0.354**	0.924	0.921	**0.941**	0.140	0.131	**0.395**	0.928	0.928	**0.942**
RAnkle	0.135	0.112	**0.414**	0.930	0.925	**0.938**	0.138	0.116	**0.440**	0.936	0.930	**0.946**
LHip	0.093	0.104	**0.312**	0.917	0.910	**0.929**	0.152	0.166	**0.559**	0.987	0.985	**0.981**
LKnee	0.123	0.120	**0.344**	0.930	0.927	**0.950**	0.140	0.143	**0.415**	0.953	0.950	**0.964**
LAnkle	0.108	0.098	**0.408**	0.921	0.917	**0.929**	0.113	0.106	**0.486**	0.950	0.950	**0.957**
REye	0.071	0.090	**0.295**	0.831	0.834	**0.870**	0.072	0.090	**0.298**	0.833	0.836	**0.872**
LEye	0.111	0.128	**0.403**	0.898	0.893	**0.915**	0.134	0.157	**0.495**	0.938	0.932	**0.943**
REar	0.069	0.090	**0.282**	0.831	0.832	**0.866**	0.067	0.084	**0.259**	0.822	0.821	**0.858**
LEar	0.113	0.137	**0.439**	0.885	**0.891**	0.850	0.128	0.154	**0.492**	0.942	0.948	**0.936**
AVERAGE	0.084	0.098	**0.306**	0.866	0.865	**0.892**	0.097	0.108	**0.368**	0.890	0.890	**0.912**

Based on the results of Table 5.2, at almost all keypoints, the LSTM model consistently achieves higher PCK@5 and PCK@50 scores compared to the MLP and CNN models, although the difference is smaller at the PCK@50 threshold. The LSTM's capacity to identify temporal dependencies in the data may be the factor that makes this model better suited for this task.

Regarding the confidence thresholds, at threshold 0, the models exhibit lower precision, as seen in the lower PCK@5 scores but relatively high PCK@50 scores. In contrast, raising the confidence threshold to 0.6 improves overall precision. The threshold of 0.6

benefits keypoints such as **LHip** and **RHip**, where scores improve significantly, suggesting that filtering low confidence predictions improves accuracy for specific body parts.

Some keypoints are harder to predict accurately than others. **Elbows** and **Wrists** have lower prediction scores overall. This may occur because these keypoints have a higher degree of freedom compared to other keypoints like **Hips**, which exhibit more constrained movement patterns, making it easier to estimate their positions. On the other hand, **Elbows** and **Wrists** have a wider range of motion because they can be flexed and extended, making them harder to predict.

The development started with Multilayer Perceptron (MLP), CNN and LSTM. These models obtained poor results due to their simplicity, but they were useful as building blocks for more complex models. Therefore, in the next sections, the combination of CNN and LSTM, along with confidence threshold of 60%, will be explored to determine whether the advantages of both models can translate into better performance.

5.2. CNN+LSTM

After analysing the results of the MLP, CNN and LSTM models, a study was carried out with a hybrid model, built with the combination of the CNN and LSTM models, to look for possible improvements. A representation of the CNN+LSTM network is shown on Figure 5.4. The experiments are summarized on Table 5.3, which shows the results of the CNN+LSTM model.

The combined performance of the basic models allows for taking advantage of the strengths of both basic architectures, leading to notably higher accuracy across most thresholds. A left-side bias is visible throughout the results, where left-side keypoints consistently outperform their right counterparts. The **LWrist**, achieves 0.371 accuracy at PCK@5 compared to the **RWrist**'s 0.122, a pattern that holds true for **Shoulders**, **Elbows**, and other paired joints. This consistent asymmetry could result from movement patterns in the training data or hardware placement during data collection.

When comparing overall performance with the results from the previous section, the CNN+LSTM model is a direct improvement. It achieves the highest average scores across both strict and permissive thresholds. For PCK@5, the best average from the previous section, LSTM with confidence threshold 0.6, is 0.368, while CNN+LSTM scores 0.394. For PCK@50, the LSTM achieves a score of 0.912, matching the performance of the

TABLE 5.3: CNN+LSTM built with the previous models

Keypoints	PCK@5	PCK@10	PCK@20	PCK@30	PCK@40	PCK@50
Nose	0.332	0.537	0.696	0.783	0.836	0.877
Neck	0.373	0.565	0.720	0.807	0.868	0.905
RShoulder	0.309	0.518	0.695	0.787	0.847	0.885
RElbow	0.162	0.357	0.593	0.723	0.804	0.857
RWrist	0.122	0.273	0.485	0.628	0.725	0.799
LShoulder	0.428	0.615	0.753	0.825	0.876	0.911
LElbow	0.400	0.585	0.735	0.817	0.869	0.906
LWrist	0.371	0.564	0.731	0.814	0.875	0.920
RHip	0.505	0.709	0.846	0.904	0.936	0.957
RKnee	0.422	0.642	0.812	0.882	0.917	0.940
RAnkle	0.459	0.693	0.838	0.891	0.923	0.944
LHip	0.606	0.788	0.899	0.935	0.964	0.981
LKnee	0.440	0.670	0.841	0.912	0.946	0.967
LAnkle	0.504	0.730	0.866	0.912	0.941	0.960
REye	0.327	0.531	0.692	0.778	0.831	0.872
LEye	0.514	0.731	0.851	0.897	0.926	0.945
REar	0.289	0.477	0.660	0.752	0.812	0.857
LEar	0.532	0.734	0.845	0.888	0.914	0.931
AVERAGE	0.394	0.595	0.753	0.830	0.878	0.912

CNN+LSTM model. However, CNN+LSTM demonstrates superior consistency, maintaining this high accuracy with greater consistency across most keypoints and thresholds, see Table A.2.

The results indicate that the combination of CNN with LSTM achieves clear gains over using either approach individually. The hybrid model attains both higher accuracy and greater consistency in recognizing keypoints, especially under stricter evaluation criteria. This highlights the added value of using architectural combinations, CNNs for extracting spatial features and LSTMs for temporal dependencies.

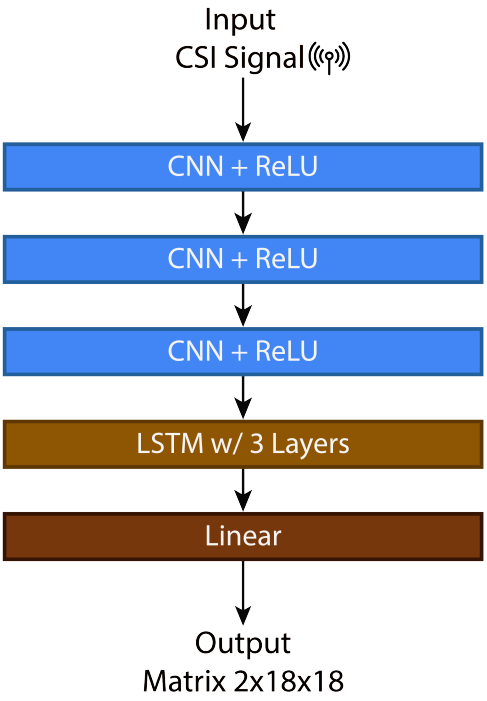

FIGURE 5.1: First iteration of CNN+LSTM network

5.3. Batch normalization

After evaluating the CNN+LSTM approach, an additional set of experiments was done with the introduction of batch normalization to the neural network architecture to test its impact on performance. The results, summarized in Table A.3, detail the configurations and outcomes of models updated with batch normalization.

Although batch normalization offers theoretical benefits for training stability, the actual performance differences between the model with and without batch normalization are small across most keypoints and thresholds. The average scores show nearly identical performance, suggesting batch normalization's main benefit may lie more in training efficiency than in improving accuracy.

Batch normalization tends to have slightly different effects depending on the body part and precision level. For some keypoints like the **LEar** and **LEye**, batch normalization improves the performance across multiple thresholds, with the **LEar** showing the most substantial gain of 0.056 at PCK@5. However, for joints like **RElbow** and **RWrist**, batch normalization does not provide improvement or deterioration in scores. This pattern suggests that batch normalization's effectiveness may depend on the specific characteristics of how different joints move and are represented in the data. The left-side joints show

TABLE 5.4: Comparison of the model's performance without and with batch normalization

Keypoints	PCK@5		PCK@10		PCK@20		PCK@50	
	w/o BN	w/ BN	w/o BN	w/ BN	w/o BN	w/ BN	w/o BN	w/ BN
Nose	**0.332**	0.323	**0.537**	0.526	**0.696**	0.689	**0.877**	0.872
Neck	**0.373**	0.372	**0.565**	0.556	**0.720**	0.718	**0.905**	0.897
RShoulder	**0.309**	0.304	**0.518**	0.512	**0.695**	0.691	**0.885**	0.877
RElbow	**0.162**	0.156	**0.357**	0.357	**0.593**	0.588	**0.857**	0.847
RWrist	**0.122**	0.118	**0.273**	0.266	0.485	**0.487**	**0.799**	0.792
LShoulder	**0.428**	0.426	**0.615**	0.615	**0.753**	0.751	**0.911**	0.899
LElbow	0.400	**0.402**	0.585	**0.596**	0.735	**0.742**	**0.906**	0.894
LWrist	0.371	**0.391**	0.564	**0.578**	0.731	**0.739**	**0.920**	0.917
RHip	0.505	**0.506**	**0.709**	0.706	**0.846**	0.846	**0.957**	0.954
RKnee	0.422	**0.424**	0.642	**0.648**	**0.812**	0.811	**0.940**	0.939
RAnkle	0.459	**0.467**	0.693	**0.700**	0.838	**0.839**	**0.944**	0.941
LHip	**0.606**	0.604	0.787	**0.801**	0.899	**0.903**	**0.981**	0.974
LKnee	0.440	**0.444**	0.670	**0.677**	0.840	**0.843**	**0.967**	0.964
LAnkle	0.504	**0.519**	0.730	**0.736**	**0.866**	0.858	**0.960**	0.951
REye	**0.327**	0.312	**0.531**	0.516	**0.692**	0.684	**0.872**	0.868
LEye	0.514	**0.535**	0.731	**0.732**	**0.851**	0.846	0.945	**0.946**
REar	**0.289**	0.272	**0.477**	0.467	**0.660**	0.651	**0.857**	0.853
LEar	0.532	**0.589**	0.734	**0.769**	0.845	**0.856**	0.930	**0.935**
AVERAGE	0.394	**0.398**	0.595	**0.598**	**0.753**	0.752	**0.912**	0.907

more responsiveness to batch normalization than their right-side counterparts, continuing the left-right asymmetry observed in previous analysis.

These findings suggest that, while batch normalization remains a valuable tool for model training, its inclusion does not drastically alter the final results. The small performance variations between the two versions indicate that other architecture choices or data processing steps likely influence overall accuracy more than batch normalization alone. Although batch normalization does not directly improve results, it contributes to enhancing training efficiency and was therefore kept in the subsequent experiments.

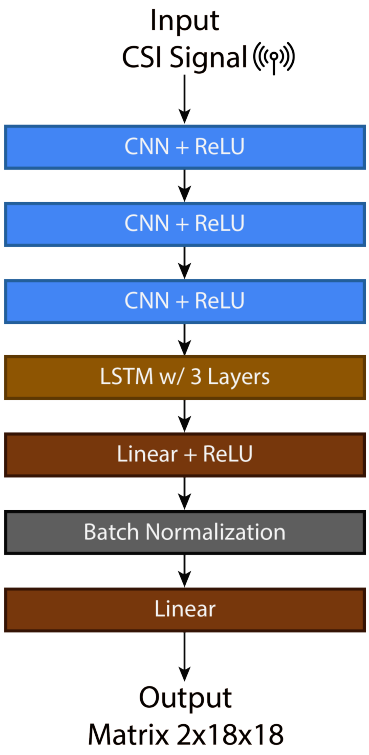

FIGURE 5.2: Second iteration of CNN+LSTM network with Batch Normalization

5.4. Dropout

Additional experiments were made to see how the dropout rate affects the CNN+LSTM model's performance. Table 5.5 shows the results of how different dropout rate levels, between 0 and 0.8, impact the balance between feature learning and generalization.

TABLE 5.5: Dropout rate comparison CNN+LSTM model

Dropout Rate	PCK@5	PCK@10	PCK@20	PCK@30	PCK@40	PCK@50
0	0.394	0.595	0.753	0.830	0.878	0.912
0.1	0.449	0.632	0.776	**0.847**	**0.891**	0.920
0.2	**0.465**	**0.642**	**0.778**	0.846	0.889	0.919
0.3	0.428	0.620	0.768	0.844	**0.891**	**0.922**
0.5	0.325	0.512	0.682	0.774	0.837	0.882
0.7	0.126	0.274	0.476	0.614	0.711	0.777
0.8	0.111	0.241	0.434	0.577	0.687	0.764

Table 5.5 shows that the general trend is that lower dropout rates, 0.1 and 0.2, generally result in better model performance across most PCK thresholds, indicating that low dropout rates help to prevent overfitting and enhance generalization. However, performance drastically deteriorates as the dropout rate rises above 0.2, particularly at more

rigorous thresholds like PCK@5 and PCK@10. This implies that high dropout rate values may impair the model's ability to learn valuable features, resulting in a significant portion of the data being ignored and lowering prediction accuracy.

The optimal dropout rate appears to be around 0.2, because it obtains the best balance between preventing overfitting and maintaining model performance. It also performs well across all thresholds and has the highest PCK@5 score. The performance for the dropout rates of 0.1, 0.3 and 0.5 are also competitive on all thresholds, however, they show a small decline in performance when compared to dropout rate of 0.2.

The performance declines significantly with a dropout rate of 0.8. As it shows poor performance across all thresholds, with the lowest scores, PCK@5 of 0.111 and PCK@10 of 0.241, indicating that a dropout rate of 0.8 significantly restricts the model's learning capacity.

The performance at different PCK thresholds varies based on how strict the thresholds are. Stricter thresholds like PCK@5 or PCK@10 are highly sensitive to dropout rates, with lower dropout rates, 0.1 and 0.2, performing significantly better, while higher rates lead to poorer performance. However, dropout rates have less impact on permissive thresholds such as PCK@40 and PCK@50, with even dropout rates of 0.5 yielding competitive scores. This suggests that even with aggressive dropout rates, the model remains capable of generating reliable predictions.

In summary, a dropout rate of 0.2 achieves the best balance between preventing overfitting and maintaining high performance across all PCK thresholds. Lower dropout rates, 0.1 and 0.2, enhance generalization, while higher dropout rates, like 0.8, result in significant performance degradation, especially for precise predictions.

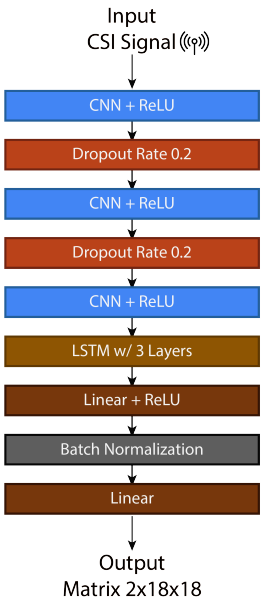

FIGURE 5.3: Third iteration of CNN+LSTM network with Dropout Rate

5.5. LSTM Layers

Following the results of the dropout rate tests, some experiments were performed by adding LSTM layers to the network architecture presented in the previous section, with the objective of better capturing temporal dependencies in the data. The details of these experiments are presented in Table 5.6, which outlines the impact of incorporating more LSTM layers into the model.

TABLE 5.6: Performance comparison of LSTM layers

Nº of layers	PCK@5	PCK@10	PCK@20	PCK@30	PCK@40	PCK@50
3	0.465	0.642	0.778	0.846	0.889	0.919
5	0.484	0.650	0.775	0.840	0.883	0.912
6	**0.504**	**0.664**	0.785	0.846	0.885	0.913
7	0.496	0.655	0.775	0.837	0.877	0.908
9	0.497	0.660	**0.789**	**0.857**	**0.893**	**0.920**

Performance generally improves for stricter thresholds as the layer count increases from 3 to 6, with the 6-layer model achieving peak scores at PCK@5 and PCK@10. However, the pattern becomes less straightforward beyond 6 layers, while the 9-layer configuration shows the best performance from PCK@20 to PCK@50, it underperforms the 6-layer model at PCK@5 and PCK@10. This suggests that additional layers may enable better feature extraction up to a certain point and considering the required prediction accuracy.

Additionally, the models featuring 5 to 7 layers show consistent performance, ranging from 0.908 to 0.913 at PCK@50. This stability suggests that the model becomes relatively insensitive to small depth variations in this range. The performance of the 3-layer model, particularly at PCK@50, may demonstrate that simpler networks can still achieve strong results for less precise estimation, however, the results lag in strict metrics like PCK@5 compared to the rest of the models.

The findings indicate that picking the right number of layers depends on what is more important for a specific problem. The 6-layer model works best for tasks that require exact joint positions, especially at strict thresholds such as PCK@5, however, if the goal is to get the overall body pose correct, the 9-layer model performs slightly better.

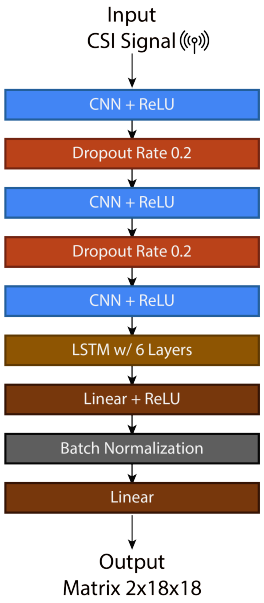

FIGURE 5.4: Fourth iteration of CNN+LSTM network with 6 LSTM Layers

5.6. Attention Mechanism

After analysing the results of previous models with different numbers of LSTM layers, a similar study was performed for the CNN+LSTM model with batch normalization, dropout rate of 0.2, six LSTM layers and an attention mechanism to see if the model's performance improves. The experiments are summarized on Table 5.7, showing different attention mechanism configurations for CNN+LSTM model.

Analysing the results, increasing model dimension tends to improve performance up to MD-64 with NH-4, with MD-64 NH-4 obtaining the best scores for PCK@5 with 0.504 and PCK@10 with 0.663, indicating an optimal trade-off between model capacity and computational efficiency.

TABLE 5.7: Attention mechanism comparison for Model Dimension (MD) and Number of Heads (NH)

Attention Mechanism	PCK@5	PCK@10	PCK@20	PCK@30	PCK@40	PCK@50
MD-16 NH-2	0.457	0.639	0.778	**0.850**	**0.894**	**0.923**
MD-16 NH-4	0.495	0.661	**0.786**	**0.850**	0.891	0.918
MD-32 NH-2	0.498	0.655	0.779	0.841	0.880	0.909
MD-32 NH-4	0.499	0.661	0.783	0.844	0.884	0.913
MD-64 NH-2	0.486	0.647	0.775	0.840	0.880	0.910
MD-64 NH-4	**0.504**	**0.663**	**0.786**	0.847	0.885	0.912
MD-128 NH-2	0.469	0.644	0.776	0.840	0.882	0.912
MD-128 NH-4	0.483	0.648	0.778	0.842	0.882	0.910

Moreover, performance is consistently improved by using more attention heads (NH-2 vs. NH-4). NH-4 configurations outperform their NH-2 counterparts in almost every model dimension value. This shows that multiple attention heads are more effective at capturing unique feature relationships in the data. For example, while MD-16 NH-2 performs best at PCK@40 with 0.894 and PCK@50 with 0.923, the MD-16 NH-4 configuration performs particularly well at PCK@20 with 0.786 and PCK@30 with 0.850, indicating that increasing the number of heads may improve the results on stricter PCKs.

Performance patterns indicate that attention mechanisms are most beneficial for strict precision thresholds, PCK@5 and PCK@10, where MD-64 NH-4 model is superior, while their advantage drops at more permissive thresholds (PCK@30 to PCK@50).

The poor performance of MD-128 models indicates that overfitting may result from large model dimensions. This emphasizes how important it is to choose the right model dimension to balance generalization and complexity.

In summary, the results indicate that the MD-64 NH-4 is the optimal overall configuration for balanced performance across precision requirements, and that careful tuning of both model dimension and number of attention heads is required. Up to a certain point, performance can be enhanced by larger model dimensions and more attention heads, however, overfitting can result from excessive complexity. The configuration that obtained best results with CNN+LSTM model is presented on 5.5.

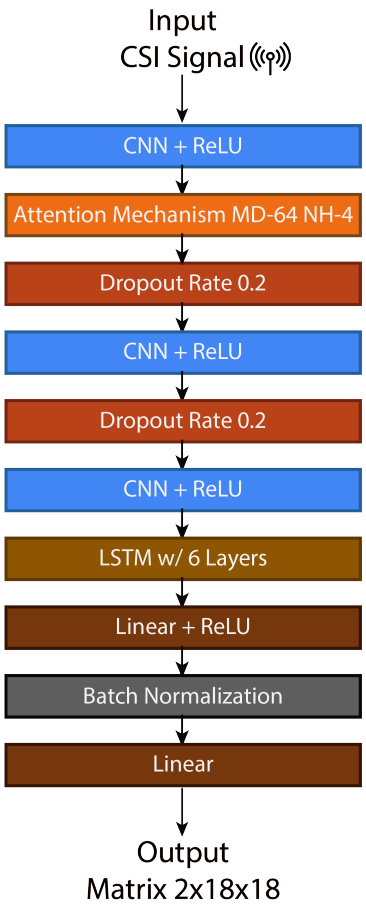

FIGURE 5.5: Final iteration of CNN+LSTM network

5.7. Spectrogram

In this section, the results of spectrogram data with various models is analysed. The goal of this approach was to convert the human pose estimation problem into an image classification and recognition problem by transforming raw CSI data into spectrogram representations. In this process, the raw multi-antenna CSI measurements are converted into a 2D time-subcarrier matrix, creating a spectrogram-like image [16]. It uses spectrograms' ability to visually encode motion patterns and frequency variations that may be less evident in CSI measurements. A representation of the transformation from CSI data to spectrogram is presented on Figure 5.6.

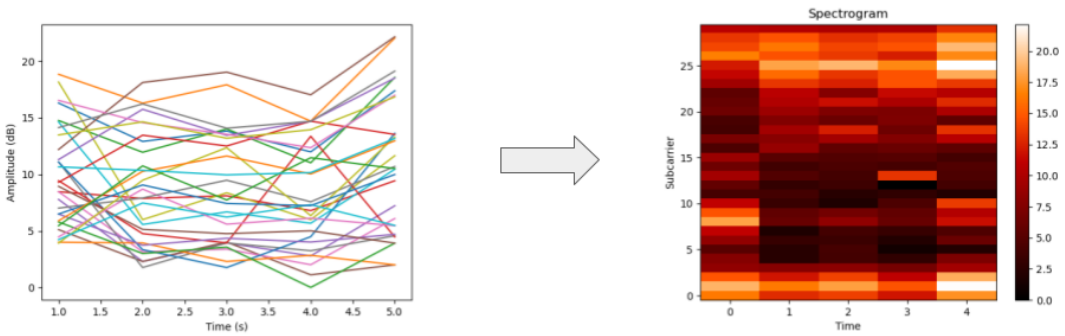

FIGURE 5.6: Transformation from CSI data to spectrogram

To generate these representations, a custom data processing pipeline was implemented. In which the CSI data files from the dataset are processed by iterating through each file and reshaping the data into a matrix that combines information from subcarrier and antenna pair. Then, the data from each frequency subcarrier is separated and the values for each antenna pair accross all frames is extracted. The code then calculates the absolute value of the CSI data, organizing the results into an array where each entry represents the signal strength over time and across subcarriers for that particular antenna pair.

5.7.1 Comparison of the CNN+LSTM with image recognition deep learning models

In this section, a comparative analysis of different neural network architectures for spectrogram data is presented, see Table 5.8.

TABLE 5.8: Comparison of various models using Spectrogram data

Architecture	PCK@5	PCK@10	PCK@20	PCK@30	PCK@40	PCK@50
CNN+LSTM	**0.303**	0.446	0.590	0.685	0.755	0.807
GoogLeNet	0.224	0.393	0.568	0.677	0.754	0.810
InceptionV3	0.292	**0.450**	**0.606**	**0.702**	**0.768**	**0.817**
ResNet-18	0.259	0.412	0.573	0.675	0.748	0.802

Looking at the results, CNN+LSTM model performs best at PCK@5 threshold with 0.303, suggesting a particular strength in precise pose estimation tasks. This is likely because it can capture both spatial features through convolutional layers and temporal dependencies via LSTM, making it well-suited for analysing sequential spectrogram data.

InceptionV3 is the top performer in the range of thresholds from PCK@10 to PCK@50. This consistent performance advantage indicates that its advanced architecture, which

features parallel convolutional paths for extracting multiscale features, enables the network to capture and represent information at various spatial resolutions. By processing the input through multiple convolutional filters of different sizes in parallel and subsequently concatenating their outputs, it excels at identifying both fine-grained and large-scale patterns within the data [26]. When compared to GoogLeNet, InceptionV3 outperforms GoogLeNet across all thresholds. This superiority might be due to InceptionV3's refined architecture, which builds upon the original GoogLeNet design by including factorized convolutions for multiscale feature extraction [25, 26].

The results indicate that the choice of architecture should be based on the specific precision requirements. For tasks that require high precision (PCK@5), the CNN+LSTM hybrid model is the most effective. Despite the weaker performance at PCK@5, InceptionV3 seems to be the best choice because the performance difference is minimal and provides the best overall performance across the rest of the thresholds when working with spectrogram data.

5.7.2 Comparison of the best model using CSI data vs. Spectrogram data

A comparative analysis was conducted between InceptionV3, which displayed the best performance when processing spectrogram data, and CNN+LSTM, which achieved the best performance with CSI data. The results of this comparative analysis are presented in Table 5.9.

The CSI data processing method demonstrates consistent superiority across all keypoints and precision thresholds, accomplishing significantly higher accuracy than the spectrogram conversion approach. This advantage is particularly pronounced for precise pose estimation, as shown by the higher average PCK@5 score for CSI with 0.504 when compared to spectrograms with 0.292. The performance gap persists throughout all evaluation thresholds. However, it becomes less pronounced at more permissive precision requirements, suggesting that while spectrograms can approximate general pose information, they struggle to translate the exact joint localizations.

Taking a closer look at specific extremity keypoints, such as **Wrists** and **Elbows**, suggests that the CSI method offers better improvements in accuracy when compared to the spectrogram approach. For instance, the **RWrist** improves from 0.076 on spectrogram to 0.188 on CSI at PCK@5. A similar trend can also be seen on **Elbows**. The relative gains for extremity parts slightly exceed the performance improvements that are observed on

TABLE 5.9: Comparison of the performance of the best model using Spectrogram data with the best model using CSI data as input

Keypoints	PCK@5		PCK@20		PCK@50	
	Spectrogram	CSI	Spectrogram	CSI	Spectrogram	CSI
Nose	0.238	**0.405**	0.560	**0.734**	0.783	**0.878**
Neck	0.272	**0.463**	0.587	**0.753**	0.815	**0.895**
RShoulder	0.216	**0.383**	0.551	**0.730**	0.791	**0.882**
RElbow	0.113	**0.229**	0.442	**0.630**	0.713	**0.863**
RWrist	0.076	**0.188**	0.349	**0.541**	0.625	**0.813**
LShoulder	0.319	**0.545**	0.610	**0.792**	0.808	**0.903**
LElbow	0.299	**0.535**	0.574	**0.779**	0.772	**0.902**
LWrist	0.294	**0.511**	0.581	**0.780**	0.802	**0.926**
RHip	0.382	**0.606**	0.703	**0.857**	0.901	**0.950**
RKnee	0.305	**0.516**	0.671	**0.834**	0.886	**0.941**
RAnkle	0.314	**0.626**	0.700	**0.860**	0.896	**0.941**
LHip	0.447	**0.722**	0.730	**0.909**	0.900	**0.967**
LKnee	0.324	**0.549**	0.704	**0.857**	0.904	**0.966**
LAnkle	0.367	**0.678**	0.719	**0.892**	0.888	**0.957**
REye	0.232	**0.398**	0.556	**0.727**	0.781	**0.875**
LEye	0.407	**0.640**	0.670	**0.872**	0.869	**0.952**
REar	0.190	**0.354**	0.526	**0.695**	0.761	**0.861**
LEar	0.454	**0.714**	0.675	**0.898**	0.816	**0.951**
AVERAGE	0.292	**0.504**	0.606	**0.786**	0.817	**0.912**

core body keypoints such as **Hips** and **Shoulders**. As the PCK threshold becomes more permissive, the scores for both data types improve and the discrepancies between them narrow.

As mentioned in the previous sections, the data also indicate a left-right asymmetry, with left-side keypoints generally achieving better results than those on the right. This observation might indicate movement pattern biases in the dataset.

The widening performance gap at stricter thresholds, PCK@5, compared to more tolerant ones, PCK@50, further highlights that spectrogram conversion tends to lose precision when exact localization is most critical. These findings imply that while spectrogram conversion provides an alternative method for processing CSI data, it seems to overlook vital information necessary for accurate pose estimation, especially concerning precise joint

localization and tracking of extremities.

5.8. Comparison with the State-of-the-Art

The following analysis makes comparison between CSI-Former, one of the projects mentioned in the state-of-the-art, and the model with the best performance from the previous analysis, the CNN+LSTM architecture with CSI data. The results are displayed on Table 5.10.

TABLE 5.10: Comparison of the CSI-Former model with the CNN+LSTM model

Keypoints	PCK@5		PCK@10		PCK@20		PCK@50	
	CSI-Former	CNN+LSTM	CSI-Former	CNN+LSTM	CSI-Former	CNN+LSTM	CSI-Former	CNN+LSTM
Nose	**0.562**	0.403	**0.654**	0.587	0.786	**0.807**	0.842	**0.879**
Neck	**0.591**	0.471	**0.684**	0.628	0.818	**0.820**	0.871	**0.895**
RShoulder	**0.579**	0.385	**0.677**	0.572	**0.810**	0.807	0.866	**0.887**
RElbow	**0.439**	0.221	**0.558**	0.426	0.735	**0.748**	0.812	**0.857**
RWrist	**0.353**	0.181	**0.471**	0.344	**0.666**	0.664	0.755	**0.800**
LShoulder	**0.543**	0.542	0.653	**0.685**	0.812	**0.845**	0.868	**0.903**
LElbow	0.448	**0.536**	0.561	**0.678**	0.744	**0.832**	0.825	**0.908**
LWrist	0.404	**0.507**	0.515	**0.656**	0.697	**0.849**	0.781	**0.924**
RHip	0.617	**0.624**	0.712	**0.758**	0.845	**0.908**	0.892	**0.957**
RKnee	**0.709**	0.527	**0.791**	0.699	**0.893**	0.889	0.932	**0.941**
RAnkle	**0.742**	0.627	**0.811**	0.767	0.890	**0.898**	0.917	**0.939**
LHip	0.605	**0.723**	0.712	**0.855**	0.840	**0.933**	0.888	**0.978**
LKnee	**0.667**	0.560	**0.764**	0.730	0.871	**0.918**	0.908	**0.969**
LAnkle	**0.690**	0.682	0.758	**0.817**	0.848	**0.922**	0.781	**0.964**
REye	**0.592**	0.396	**0.677**	0.577	**0.802**	**0.802**	0.845	**0.876**
LEye	0.524	**0.621**	0.619	**0.785**	0.769	**0.911**	0.836	**0.948**
REar	**0.603**	0.358	**0.690**	0.533	**0.811**	0.778	**0.864**	0.862
LEar	0.239	**0.705**	0.294	**0.840**	0.461	**0.922**	0.556	**0.950**
AVERAGE	**0.551**	0.504	0.645	**0.663**	0.783	**0.847**	0.842	**0.913**

The data presented in Table 5.10 shows that, at the strictest threshold of PCK@5, CSI-Former outperforms CNN+LSTM, particularly for keypoints such as **RShoulder**, **RElbow**, **RWrist**, and **REar**. However, CNN+LSTM performs better for certain left-side keypoints like **LHip**, **LWrist**, **LElbow**, **LEye**, and **LEar**, suggesting a specialization in detecting these points due to the asymmetry verified in previous analysis.

The performance difference between the two models narrows as the threshold relaxes to PCK@10, leading to the CNN+LSTM outperforming CSI-Former by 0.018. While CSI-Former still has an advantage for some right-side keypoints, CNN+LSTM outperforms it in others, particularly on the left side. This trend continues for the most permissive thresholds. At PCK@50, CNN+LSTM outperforms CSI-Former in most keypoints, achieving an average of 0.913 compared to CSI-Former's 0.842.

Additionally, CNN+LSTM excels in detecting some left-side extremities like **LWrist** and **LElbow**, while CSI-Former shows relative strength in right-side upper-body keypoints. The contrast in performance for, for example, the **LEar** keypoint, where CNN+LSTM outperforms CSI-Former at all thresholds, might suggest a possible limitation in CSI-Former's ability to detect certain keypoints consistently.

Overall, the choice between the two models depends on the precision requirements of the application. CSI-Former is better suited for tasks requiring higher precision, while CNN+LSTM is better suited for applications where moderate to permissive precision mistakes are tolerable. The left-right performance disparity also highlights an area for potential improvement in model training or architecture to ensure balanced detection across all keypoints.

5.9. Conclusion

In this chapter, the performance of multiple deep learning models was evaluated for keypoint detection using a set of metrics and thresholds. By analyzing results across simple models, such as MLP, CNN and LSTM, a hybrid CNN+LSTM architecture and more complex models, such as GoogLeNet, ResNet and InceptionV3, it is evident that model selection and design significantly affect accuracy and reliability in detecting human body keypoints.

While individual architectures such as LSTM obtained solid results, the hybrid CNN+LSTM approach consistently outperformed the standalone models, achieving the highest average accuracy at both strict and permissive confidence thresholds for CSI data.

In order to further improve the performance of the CNN+LSTM model, some data techniques were implemented. Each iteration of experiments aimed to improve a specific part of the model, such as adding batch normalization to improve model efficiency or dropout rate to prevent overfitting and enhance generalization. After these experiments, the performance of CNN+LSTM model improved significantly, increasing the average results on all PCKs.

There was also a set of experiments with spectrograms, converted from the CSI data. In this set of experiments, the CNN+LSTM model performance was compared with more complex models like GoogLeNet, InceptionV3 and ResNet-18. The CNN+LSTM model performed better than GoogLeNet and ResNet-18 but was outperformed by InceptionV3

in most PCKs. Following this, the best performing model using spectrograms, InceptionV3, was compared to the best performing model using CSI data, CNN+LSTM. This comparison showed that the results of the CNN+LSTM were much superior compared to the results of the InceptionV3 for all keypoints, with an average difference of 0.212 for PCK@5.

Finally, comparing the best obtained results with the state-of-the-art, it was possible to achieve a superior average performance for the PCK@10 until PCK@50, with the state-of-the-art only being superior for the PCK@5 by 0.047.

6. Conclusion

The main objective of this dissertation was to develop a method for 3D Human pose estimation using WiFi CSI data. To fulfil that, two research questions emerged: ”What are the limitations and challenges of the current methods developed to use WiFi CSI data for estimating the 3D human pose?” and ”What architectures, regularisation techniques or data preprocessing methods can be applied to develop an accurate solution based on WiFi CSI data for estimating the 3D human pose?”.

Therefore, to begin the development, a search was conducted to find the current research on the field of 3D human pose estimation using WiFi CSI data, in which the latest advancements and contributions within the field were reviewed to further widen my knowledge in this field and understand the advantages and drawbacks of current methods. This allowed to make an informed approach to the planning and development of the model.

The experiments began after analysing the state-of-the-art, starting with simple models which were then combined to form the CNN+LSTM model. Then, to further optimise this model, several experiments were performed by adding batch normalisation, dropout, then changing the number of LSTM layers and adding attention mechanisms. Out of these experiments, it was possible to understand that even though batch normalisation does not directly contribute to more precise results, it can help improve training efficiency. The use of dropout can prevent overfitting and improve the model’s performance until a certain dropout rate, in this case 0.2, after which the performance starts to decline. The increase in the number of LSTM layers also proved to be a good add-on as it allows to get better performance, reaching an optimal result for more restricted thresholds with 6 LSTM layers. Regarding the attention mechanisms, the employment of the Multi-head attention with 4 attention heads and a model dimension of 64 seems to be the most optimal combination, as it gives better performances for stricter thresholds. Hence, the best model configuration is the CNN+LSTM model with 6 LSTM layers, an attention mechanism with 4 attention heads and a model dimension of 64, batch normalisation and two dropout layers with a rate of 0.2.

Then, it was performed a set of experiments with spectrograms, converted from the CSI data, in which the CNN+LSTM model performance was compared with models like ResNet-18 and InceptionV3. The results revealed that InceptionV3 is the most suitable

model to handle spectrogram data, with the CNN+LSTM model being the second-best. However, when comparing the performance obtained with CSI data and with the spectrogram data, the results of using CSI data surpass by a large margin the results of the model using the spectrograms as input.

Finally, a comparison of the best model with the state-of-the-art was performed. The evaluation showed that CNN+LSTM achieves high keypoint detection on more flexible confidence thresholds, outperforming CSI-Former in PCK@10 until PCK@50. The CNN+LSTM model looks reliable, mainly for applications where a medium to large margin of localization error is acceptable.

Looking closely at the objectives of this dissertation, all goals set were successfully accomplished, and therefore, it was possible to create a new method using a combination of CNN and LSTM, which is able to achieve very reliable results across different precision thresholds using CSI data as input.

Future research should focus on the exploration of other architectures or different data preprocessing techniques that might be more suitable to visually represent the WiFi CSI data.

References

[1] Y. Zhou, C. Xu, L. Zhao, A. Zhu, F. Hu, and Y. Li, “Csi-former: Pay more attention to pose estimation with wifi,” *Entropy*, vol. 25, no. 1, p. 20–20, Dec 2022. [Online]. Available: https://www.mdpi.com/1099-4300/25/1/20 [Cited on pages 1, 12, 13, 15, and 17.]

[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,” *arXiv:2012.13392 [cs]*, Jan 2021. [Online]. Available: https://arxiv.org/abs/2012.13392 [Cited on page 1.]

[3] T. S. Rappaport, *Wireless communications*. Upper Saddle River, N.J. ; London: Prentice Hall Ptr, 2001. [Cited on pages 3 and 4.]

[4] V. E. Balas, R. Kumar, and R. Srivastava, *Recent trends and advances in artificial intelligence and internet of things*. Cham: Springer International Publishing, 2020. [Cited on page 4.]

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Nov 2016. [Cited on pages 4, 5, 6, 8, 9, and 10.]

[6] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*. Cambridge, Mass. ; London: Mit, 2010. [Cited on page 4.]

[7] H. H. Aghdam, E. J. Heravi, and S. O. Service, *Guide to Convolutional Neural Networks : A Practical Application to Traffic-Sign Detection and Classification*. Springer International Publishing, 2017. [Cited on page 7.]

[8] W. Jiang, H. Xue, C. Miao, S. Wang, S. Lin, C. Tian, S. Murali, H. Hu, Z. Sun, and L. Su, “Towards 3d human pose construction using wifi,” *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, Apr 2020. [Online]. Available: https://engineering.purdue.edu/~lusu/papers/MobiCom2020.pdf [Cited on pages 11, 12, 14, and 18.]

[9] J. Aggarwal and Q. Cai, “Human motion analysis: A review,” *Computer Vision and Image Understanding*, vol. 73, no. 3, p. 428–440, Mar 1999. [Cited on page 11.]

[10] D. Gavrila, “The visual analysis of human movement: A survey,” *Computer Vision and Image Understanding*, vol. 73, no. 1, p. 82–98, Jan 1999. [Cited on page 11.]

[11] T. B. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *Computer Vision and Image Understanding*, vol. 104, no. 2-3, p. 90–126, Nov 2006. [Cited on page 11.]

[12] Z. Zhang, ““microsoft kinect sensor and its effect”,” *IEEE MultiMedia*, vol. 19, p. 4–10, Jun. 2012. [Cited on page 11.]

[13] Y. Ren, Z. Wang, Y. Wang, S. Tan, Y. Chen, and J. Yang, “Gopose,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, p. 1–25, Jul 2022. [Cited on pages 11, 12, 13, and 18.]

[14] P. F. Moshiri, R. Shahbazian, M. Nabati, and S. A. Ghorashi, “A csi-based human activity recognition using deep learning,” *Sensors*, vol. 21, no. 21, p. 7225–7225, Oct 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/21/7225 [Cited on pages 11 and 15.]

[15] Y. Ren, Y. Chen, and J. Yang, *3D Human Pose Estimation for Free-form Activity Using WiFi Signals*. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2110/2110.08314.pdf [Cited on pages 12 and 13.]

[16] L. Guo, Z. Lu, X. Wen, S. Zhou, and Z. Han, “From signal to image: Capturing fine-grained human poses with commodity wi-fi,” *IEEE Communications Letters*, vol. 24, no. 4, p. 802–806, Apr 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8943379 [Cited on pages 12, 14, and 35.]

[17] F. Wang, S. Zhou, S. Panev, J. Han, and D. Huang, *Person-in-WiFi: Fine-grained Person Perception using WiFi*. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/papers/Wang_Person-in-WiFi_Fine-Grained_Person_Perception_Using_WiFi_ICCV_2019_paper.pdf [Cited on pages 12 and 13.]

[18] X. Ding, T. Jiang, Y. Li, W. Xue, and Y. Zhong, “Device-free location-independent human activity recognition using transfer learning based on cnn,” Jun 2020. [Cited on pages 12, 14, and 15.]

[19] Y. Wang, L. Guo, Z. Lu, X. Wen, S. Zhou, and W. Meng, “From point to space: 3d moving human pose estimation using commodity wifi,” *IEEE Communications Letters*, vol. 25, no. 7, p. 2235–2239, Jul 2021. [Cited on pages 12, 14, and 18.]

[20] Z. Zhang, S. Ishida, S. Tagashira, and A. Fukuda, “Danger-pose detection system using commodity wi-fi for bathroom monitoring,” *Sensors*, vol. 19, no. 4, p. 884–884, Feb 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/4/884 [Cited on pages 12 and 13.]

[21] L. Wang, L. Guo, Z. Lu, X. Wen, and S. Zhou, “Wilink: Link selection-based 3d human pose estimation using commodity wi-fi,” Mar 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10118926 [Cited on pages 13 and 14.]

[22] M. Xu, Z. Guo, L. Gui, B. Sheng, and F. Xiao, “Wispe: A cots wi-fi-based 2-d static human pose estimation,” *IEEE Systems Journal*, vol. 17, no. 3, p. 3560–3571, Sep 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10124461 [Cited on pages 13 and 14.]

[23] M. Ding, Q. Wang, and C. Wang, “Deep transfer learning for actions recognition with wifi signals,” *2022 IEEE 10th International Conference on Information, Communication and Networks (ICICN)*, Aug 2022. [Cited on pages 13 and 14.]

[24] J. Zhang, F. Wu, B. Wei, Q. Zhang, H. Huang, S. W. Shah, and J. Cheng, “Data augmentation and dense-lstm for human activity recognition using wifi signal,” *IEEE internet of things journal (Online)*, vol. 8, no. 6, p. 4628–4641, Mar 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9205901 [Cited on page 15.]

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. [Cited on pages 19 and 37.]

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” *arXiv.org*, 2015. [Online]. Available: https://arxiv.org/abs/1512.00567 [Cited on pages 19 and 37.]

[27] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, Dec 2015. [Online]. Available: https://arxiv.org/pdf/1512.03385 [Cited on page 20.]

[28] D. Kingma and J. Lei Ba, *ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION*, 2015. [Online]. Available: https://arxiv.org/pdf/1412.6980 [Cited on page 21.]

A. Complete Tables of the Models

This appendix presents the complete tables for the experiments performed in this dissertation:

- Table A.1 for the results of all confidence thresholds for the most basic models, MLP, CNN and LSTM.
- Table A.2 for the complete comparison of the various basic models with a confidence threshold of 0.6.
- Table A.3 represents the complete results for the model with and without batch normalisation for all precision thresholds.
- Table A.4 presents the comparison between the best created model with the state-of-the-art model for all precision thresholds.

TABLE A.1: Results for the MLP, CNN and LSTM models using CSI data as input and considering different confidence thresholds.

Keypoints	Confidence Threshold 0						Confidence Threshold 0.6						Confidence Threshold 0.7						Confidence Threshold 0.8						Confidence Threshold 0.9					
	PCK@5			PCK@50			PCK@5			PCK@50			PCK@5			PCK@50			PCK@5			PCK@50			PCK@5			PCK@50		
	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM
Nose	0.072	0.090	**0.303**	0.836	0.838	**0.874**	0.073	0.092	**0.308**	0.840	0.841	**0.877**	0.073	0.093	**0.317**	0.846	0.847	**0.883**	0.077	0.101	**0.351**	0.866	0.864	**0.896**	0.108	0.141	**0.491**	0.927	0.927	**0.949**
Neck	0.083	0.094	**0.313**	0.875	0.875	**0.896**	0.089	0.102	**0.355**	0.886	0.883	**0.906**	0.107	0.128	**0.461**	0.939	0.935	**0.946**	0.143	0.146	**0.531**	0.969	**0.981**	0.973	0.000	0.000	0.000	0.000	0.000	0.000
RShoulder	0.074	0.092	**0.281**	0.864	0.867	**0.888**	0.075	0.094	**0.289**	0.862	0.865	**0.887**	0.077	0.103	**0.312**	0.881	0.888	**0.909**	0.093	0.119	**0.359**	0.897	0.916	**0.937**	0.097	0.024	**0.146**	0.804	0.852	**0.876**
RElbow	0.034	0.040	**0.138**	0.797	0.804	**0.850**	0.035	0.042	**0.148**	0.806	0.814	**0.862**	0.036	0.045	**0.157**	0.812	0.822	**0.869**	0.039	0.052	**0.172**	0.826	0.838	**0.879**	0.049	0.070	**0.253**	0.815	0.828	**0.861**
RWrist	0.019	0.040	**0.100**	0.689	0.804	**0.782**	0.020	0.020	**0.102**	0.700	0.711	**0.793**	0.021	0.021	**0.106**	0.707	0.718	**0.800**	0.023	0.021	**0.118**	0.731	0.739	**0.823**	0.026	0.024	**0.099**	0.763	0.760	**0.854**
LShoulder	0.071	0.077	**0.290**	0.869	0.867	**0.889**	0.086	0.098	**0.408**	0.904	0.892	**0.908**	0.096	0.106	**0.465**	**0.931**	0.924	0.927	0.100	0.106	**0.512**	**0.957**	0.955	0.939	0.000	0.036	**0.500**	**0.964**	**0.964**	**0.964**
LElbow	0.063	0.019	**0.268**	0.850	0.698	**0.876**	0.082	0.093	**0.359**	0.885	0.880	**0.905**	0.085	0.100	**0.389**	0.902	0.895	**0.910**	0.096	0.110	**0.482**	0.930	0.930	**0.933**	0.090	0.076	**0.417**	0.972	**0.986**	0.972
LWrist	0.052	0.076	**0.263**	0.836	0.867	**0.870**	0.064	0.095	**0.343**	0.897	0.893	**0.918**	0.067	0.102	**0.361**	0.906	0.900	**0.928**	0.069	0.114	**0.390**	0.926	0.917	**0.942**	0.075	0.120	**0.469**	0.954	0.962	**0.972**
RHip	0.096	0.073	**0.307**	0.908	0.848	**0.920**	0.133	0.158	**0.471**	0.959	0.956	**0.958**	0.198	0.201	**0.648**	0.982	**0.986**	**0.986**	0.099	0.399	**0.799**	0.966	0.966	**1.000**	0.000	0.000	0.000	0.000	0.000	0.000
RKnee	0.127	0.121	**0.354**	0.924	0.921	**0.941**	0.140	0.131	**0.395**	0.928	0.928	**0.942**	0.151	0.140	**0.453**	0.927	0.926	**0.939**	0.181	0.163	**0.537**	0.906	0.903	**0.923**	0.066	0.041	**0.124**	0.681	0.689	**0.690**
RAnkle	0.135	0.112	**0.414**	0.930	0.925	**0.938**	0.138	0.116	**0.440**	0.936	0.930	**0.946**	0.154	0.127	**0.488**	0.941	0.937	**0.952**	0.162	0.149	**0.551**	0.957	0.960	**0.970**	0.124	0.157	**0.469**	**1.000**	**1.000**	**1.000**
LHip	0.093	0.104	**0.312**	0.917	0.910	**0.929**	0.152	0.166	**0.559**	0.987	0.985	**0.981**	0.173	0.196	**0.649**	0.991	**0.995**	0.989	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
LKnee	0.123	0.120	**0.344**	0.930	0.927	**0.950**	0.140	0.143	**0.415**	0.953	0.950	**0.964**	0.139	0.156	**0.456**	0.964	0.964	**0.973**	0.141	0.176	**0.492**	0.975	0.977	**0.980**	0.101	0.124	**0.324**	0.971	0.970	**0.977**
LAnkle	0.108	0.098	**0.408**	0.921	0.917	**0.929**	0.113	0.106	**0.486**	0.950	0.950	**0.957**	0.115	0.108	**0.515**	0.954	0.954	**0.956**	0.120	0.111	**0.503**	0.946	**0.954**	0.952	0.124	0.140	**0.597**	0.986	**0.990**	**0.990**
REye	0.071	0.090	**0.295**	0.831	0.834	**0.870**	0.072	0.090	**0.298**	0.833	0.836	**0.872**	0.073	0.093	**0.306**	0.840	0.842	**0.878**	0.080	0.099	**0.336**	0.854	0.855	**0.894**	0.094	0.117	**0.388**	0.879	0.881	**0.908**
LEye	0.111	0.128	**0.403**	0.898	0.893	**0.915**	0.134	0.157	**0.495**	0.938	0.932	**0.943**	0.141	0.168	**0.520**	0.951	0.949	**0.954**	0.155	0.183	**0.574**	0.973	0.978	**0.984**	0.157	0.182	**0.564**	0.968	0.991	**0.994**
REar	0.069	0.090	**0.282**	0.831	0.832	**0.866**	0.067	0.084	**0.259**	0.822	0.821	**0.858**	0.063	0.080	**0.246**	0.816	0.816	**0.855**	0.059	0.081	**0.249**	0.817	0.817	**0.857**	0.069	0.103	**0.319**	0.866	0.880	**0.900**
LEar	0.113	0.137	**0.439**	0.885	**0.891**	0.850	0.128	0.154	**0.492**	0.942	0.948	**0.936**	0.126	0.159	**0.498**	0.947	**0.949**	0.939	0.098	0.144	**0.478**	0.917	**0.927**	0.888	0.142	0.000	**0.287**	**0.855**	0.567	0.854
AVERAGE	0.084	0.098	**0.306**	0.866	0.865	**0.892**	0.097	0.108	**0.368**	0.890	0.890	**0.912**	0.105	0.118	**0.408**	0.902	0.903	**0.922**	0.096	0.126	**0.413**	0.856	0.860	**0.876**	0.073	0.075	**0.303**	0.745	0.736	**0.764**

TABLE A.2: Complete table of the results for MLP, CNN and LSTM with confidence threshold 0.6

Keypoints	PCK@5			PCK@10			PCK@20			PCK@30			PCK@40			PCK@50		
	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM	MLP	CNN	LSTM
Nose	0.073	0.092	**0.308**	0.219	0.265	**0.514**	0.479	0.525	**0.687**	0.656	0.679	**0.776**	0.769	0.779	**0.834**	0.840	0.841	**0.877**
Neck	0.089	0.102	**0.355**	0.259	0.286	**0.540**	0.533	0.554	**0.712**	0.715	0.720	**0.803**	0.827	0.818	**0.865**	0.886	0.883	**0.906**
RShoulder	0.075	0.094	**0.289**	0.226	0.263	**0.489**	0.505	0.527	**0.685**	0.692	0.693	**0.782**	0.799	0.799	**0.844**	0.862	0.865	**0.887**
RElbow	0.035	0.042	**0.148**	0.127	0.142	**0.332**	0.358	0.383	**0.576**	0.558	0.578	**0.717**	0.708	0.715	**0.804**	0.806	0.814	**0.862**
RWrist	0.020	0.020	**0.102**	0.073	0.076	**0.251**	0.242	0.247	**0.468**	0.425	0.433	**0.615**	0.577	0.589	**0.716**	0.700	0.711	**0.793**
LShoulder	0.086	0.098	**0.408**	0.263	0.277	**0.590**	0.553	0.558	**0.742**	0.724	0.727	**0.820**	0.838	0.831	**0.875**	0.904	0.892	**0.908**
LElbow	0.082	0.093	**0.359**	0.252	0.265	**0.551**	0.519	0.529	**0.720**	0.702	0.700	**0.812**	0.815	0.812	**0.871**	0.885	0.880	**0.905**
LWrist	0.064	0.095	**0.343**	0.216	0.258	**0.546**	0.519	0.530	**0.732**	0.709	0.716	**0.822**	0.828	0.829	**0.876**	0.897	0.893	**0.918**
RHip	0.133	0.158	**0.471**	0.368	0.404	**0.689**	0.693	0.725	**0.831**	0.859	0.865	**0.899**	0.927	0.928	**0.936**	**0.959**	0.956	0.958
RKnee	0.140	0.131	**0.395**	0.371	0.353	**0.615**	0.673	0.670	**0.796**	0.812	0.807	**0.878**	0.885	0.883	**0.918**	0.928	0.928	**0.942**
RAnkle	0.138	0.116	**0.440**	0.368	0.336	**0.673**	0.673	0.658	**0.828**	0.828	0.808	**0.883**	0.900	0.886	**0.922**	0.936	0.930	**0.946**
LHip	0.152	0.166	**0.559**	0.399	0.426	**0.778**	0.743	0.748	**0.895**	0.902	0.894	**0.939**	0.961	0.965	**0.966**	**0.987**	0.985	0.981
LKnee	0.140	0.143	**0.415**	0.383	0.358	**0.653**	0.693	0.673	**0.831**	0.836	0.830	**0.904**	0.913	0.909	**0.944**	0.953	0.950	**0.964**
LAnkle	0.113	0.106	**0.486**	0.353	0.314	**0.710**	0.678	0.639	**0.852**	0.832	0.808	**0.905**	0.910	0.903	**0.937**	0.950	0.950	**0.957**
REye	0.072	0.091	**0.298**	0.215	0.262	**0.505**	0.471	0.520	**0.683**	0.648	0.673	**0.770**	0.761	0.772	**0.828**	0.833	0.836	**0.872**
LEye	0.134	0.158	**0.495**	0.352	0.424	**0.707**	0.636	0.723	**0.844**	0.812	0.845	**0.898**	0.899	0.902	**0.927**	0.938	0.932	**0.943**
REar	0.067	0.084	**0.259**	0.199	0.237	**0.457**	0.450	0.485	**0.647**	0.628	0.641	**0.742**	0.745	0.749	**0.809**	0.822	0.821	**0.858**
LEar	0.128	0.154	**0.492**	0.344	0.409	**0.705**	0.644	0.708	**0.838**	0.813	0.849	**0.889**	0.896	0.916	**0.920**	0.942	**0.948**	0.936
AVERAGE	0.097	0.108	**0.368**	0.277	0.298	**0.573**	0.559	0.578	**0.743**	0.731	0.737	**0.825**	0.831	0.833	**0.877**	0.890	0.890	**0.912**

TABLE A.3: Comparison of model without and with batch normalization

Keypoints	PCK@5		PCK@10		PCK@20		PCK@30		PCK@40		PCK@50	
	w/o BN	w/ BN	w/o BN	w/ BN	w/o BN	w/ BN	w/o BN	w/ BN	w/o BN	w/ BN	w/o BN	w/ BN
Nose	**0.332**	0.323	**0.537**	0.526	**0.696**	0.689	**0.783**	0.779	**0.836**	0.834	**0.877**	0.872
Neck	**0.373**	0.372	**0.565**	0.556	**0.720**	0.718	**0.807**	**0.807**	**0.868**	0.860	**0.905**	0.897
RShoulder	**0.309**	0.304	**0.518**	0.512	**0.695**	0.691	**0.787**	0.783	**0.847**	0.839	**0.885**	0.877
RElbow	**0.162**	0.156	**0.357**	0.357	**0.593**	0.588	**0.723**	0.712	**0.804**	0.791	**0.857**	0.847
RWrist	**0.122**	0.118	**0.273**	0.266	0.485	**0.487**	**0.628**	0.627	**0.725**	0.723	**0.799**	0.792
LShoulder	**0.428**	0.426	**0.615**	0.615	**0.753**	0.751	**0.825**	0.824	**0.876**	0.867	**0.911**	0.899
LElbow	0.400	**0.402**	0.585	**0.596**	0.735	**0.742**	**0.817**	0.813	**0.869**	0.858	**0.906**	0.894
LWrist	0.371	**0.391**	0.564	**0.578**	0.731	**0.739**	0.814	**0.819**	**0.875**	0.873	**0.920**	0.917
RHip	0.505	**0.506**	**0.709**	0.706	**0.846**	0.846	0.904	**0.905**	**0.936**	0.932	**0.957**	0.954
RKnee	0.422	**0.424**	0.642	**0.648**	**0.812**	0.811	0.882	**0.878**	**0.917**	0.914	**0.940**	0.939
RAnkle	0.459	**0.467**	0.693	**0.700**	0.838	**0.839**	0.891	**0.892**	**0.923**	0.921	**0.944**	0.941
LHip	**0.606**	0.604	0.787	**0.801**	0.899	**0.903**	0.935	**0.938**	**0.964**	0.960	**0.981**	0.974
LKnee	0.440	**0.444**	0.670	**0.677**	0.840	**0.843**	**0.912**	0.907	**0.946**	0.944	**0.967**	0.964
LAnkle	0.504	**0.519**	0.730	**0.736**	**0.866**	0.858	**0.912**	0.906	**0.941**	0.933	**0.960**	0.951
REye	**0.327**	0.312	**0.531**	0.516	**0.692**	0.684	**0.777**	0.775	**0.831**	**0.831**	**0.872**	0.868
LEye	0.514	**0.535**	0.731	**0.732**	**0.851**	0.846	0.897	**0.901**	0.926	**0.930**	0.945	**0.946**
REar	**0.289**	0.272	**0.477**	0.467	**0.660**	0.651	**0.752**	0.751	**0.812**	0.811	**0.857**	0.853
LEar	0.532	**0.589**	0.734	**0.769**	0.845	**0.856**	0.888	**0.895**	0.914	**0.917**	0.930	**0.935**
AVERAGE	0.394	**0.398**	0.595	**0.598**	**0.753**	0.752	**0.830**	0.828	**0.878**	0.874	**0.912**	0.907

TABLE A.4: Comparison of the CSI-Former model with the CNN+LSTM model

Keypoints	PCK@5		PCK@10		PCK@20		PCK@30		PCK@40		PCK@50	
	CSI-Former	CNN+LSTM	CSI-Former	CNN+LSTM	CSI-Former	CNN+LSTM	CSI-Former	CNN+LSTM	CSI-Former	CNN+LSTM	CSI-Former	CNN+LSTM
Nose	**0.562**	0.403	**0.654**	0.587	0.786	**0.807**	**0.740**	0.727	0.819	**0.848**	0.842	**0.879**
Neck	**0.591**	0.471	**0.684**	0.628	0.818	**0.820**	**0.771**	0.748	0.848	**0.863**	0.871	**0.895**
RShoulder	**0.579**	0.385	**0.677**	0.572	**0.810**	0.807	**0.762**	0.727	0.841	**0.851**	0.866	**0.887**
RElbow	**0.439**	0.221	**0.558**	0.426	0.735	**0.748**	**0.673**	0.633	0.779	**0.814**	0.812	**0.857**
RWrist	**0.353**	0.181	**0.471**	0.344	**0.666**	0.664	**0.596**	0.532	0.715	**0.749**	0.755	**0.800**
LShoulder	**0.543**	0.542	0.653	**0.685**	0.812	**0.845**	0.758	**0.788**	0.844	**0.878**	0.868	**0.903**
LElbow	0.448	**0.536**	0.561	**0.678**	0.744	**0.832**	0.676	**0.782**	0.791	**0.873**	0.825	**0.908**
LWrist	0.404	**0.507**	0.515	**0.656**	0.697	**0.849**	0.625	**0.783**	0.746	**0.887**	0.781	**0.924**
RHip	0.617	**0.624**	0.712	**0.758**	0.845	**0.908**	0.800	**0.859**	0.873	**0.932**	0.892	**0.957**
RKnee	**0.709**	0.527	**0.791**	0.699	**0.893**	0.889	**0.861**	0.835	0.916	**0.920**	0.932	**0.941**
RAnkle	**0.742**	0.627	**0.811**	0.767	0.890	**0.898**	**0.861**	0.859	0.906	**0.923**	0.917	**0.939**
LHip	0.605	**0.723**	0.712	**0.855**	0.840	**0.933**	0.794	**0.917**	0.868	**0.953**	0.888	**0.978**
LKnee	**0.667**	0.560	**0.764**	0.730	0.871	**0.918**	0.836	**0.868**	0.893	**0.949**	0.908	**0.969**
LAnkle	**0.690**	0.682	0.758	**0.817**	0.848	**0.922**	0.817	**0.904**	0.869	**0.943**	0.781	**0.964**
REye	**0.592**	0.396	**0.677**	0.577	**0.802**	**0.802**	**0.755**	0.722	0.832	**0.845**	0.845	**0.876**
LEye	0.524	**0.621**	0.619	**0.785**	0.769	**0.911**	0.714	**0.868**	0.809	**0.935**	0.836	**0.948**
REar	**0.603**	0.358	**0.690**	0.533	**0.811**	0.778	**0.767**	0.691	**0.844**	0.827	**0.864**	0.862
LEar	0.239	**0.705**	0.294	**0.840**	0.461	**0.922**	0.387	**0.896**	0.517	**0.939**	0.556	**0.950**
AVERAGE	**0.551**	0.504	0.645	**0.663**	0.783	**0.847**	0.733	**0.785**	0.817	**0.885**	0.842	**0.913**