# TRANSFORMER AND POSE GRAMMAR BASED DEEP NEURAL NETWORK FOR 3D HUMAN POSE ESTIMATION

ZINIA SULTANA (SN. 1017140027)

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science and Engineering


DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY
DHAKA, BANGLADESH

MARCH 2024

# TRANSFORMER AND POSE GRAMMAR BASED DEEP NEURAL NETWORK FOR 3D HUMAN POSE ESTIMATION

M.Sc. Engineering Thesis

By

ZINIA SULTANA (SN. 1017140027)

Approved as to style and content by the Board of Examination on 27 March 2024:

| | |
|---|---|
| Dr. Md. Hasanul Kabir<br>Professor of Computer Science and Engineering<br>Islamic University of Technology, Dhaka | Chairman (Supervisor)<br>Board of Examination |
| Dr. Muhammad Nazrul Islam<br>Associate Professor of Computer Science and Engineering<br>MIST, Dhaka | Member (Co-Supervisor)<br>Board of Examination |
| Dr. Mohammad Abu Yousuf<br>Professor of Institute of Information Technology<br>Jahangirnagar University, Dhaka | Member (External)<br>Board of Examination |
| Dr. Nusrat Sharmin<br>Assistant Professor of Computer Science and Engineering<br>MIST, Dhaka | Member (Internal)<br>Board of Examination |
| Brig Gen Mohammad Sajjad Hossain<br>Senior Instructor of Computer Science and Engineering<br>MIST, Dhaka | Head of the Department<br>Member (Ex-officio) |

Department of Computer Science and Engineering, MIST, Dhaka.

# TRANSFORMER AND POSE GRAMMAR BASED DEEP NEURAL NETWORK FOR 3D HUMAN POSE ESTIMATION

## DECLARATION

This is to certify that the work presented in this thesis book, titled, "TRANSFORMER AND POSE GRAMMAR BASED DEEP NEURAL NETWORK FOR 3D HUMAN POSE ESTIMATION", is the outcome of the investigation and research carried out by the following student under the supervision of Md. Hasanul Kabir, Professor, Department of Computer Science and Engineering, Islamic University of Technology. I therefore declare that this thesis is my unique work and authored entirely by myself. I have properly credited all sources of material used in the thesis. This thesis has never been presented for a degree or diploma at any university or institute before (in whole or in part). All sources and support obtained in preparing this thesis have been acknowledged and/or cited in the reference section.

---

Zinia Sultana

Department of Computer Science and Engineering, MIST, Dhaka.

# ABSTRACT

## Transformer and Pose Grammar Based Deep Neural Network for 3D Human Pose Estimation

For computers to understand human activity or behavior in a variety of scenarios, reliable 3D human posture estimation is a prerequisite. A number of difficulties have made such work more complex as it is influenced by various factors, including image quality, background, garment texture and diversity, body shape, and the presence of other objects alongside persons in the image which has depicted the necessity of adopting the technique of computer vision. While much work has been done on 2D human pose estimation, showing state-of-the-art performance, the objective of this research is to estimate 3D human pose from 2D joint positions. We have investigated deep neural networks comprising of linear layers with residual blocks and proposed a hybrid deep learning framework in order to achieve this objective. We experimented the proposed by raising the number of residual blocks to anlaysis the performance. The final proposed architecture (HEpose) comprises of three parallel models, one model is base one only the linear layers concept, second one is based on the residul connection without normalization, and third model gathers the information of connection among the joints. We combined ouputs of the three model and finally used a fully connected linear layer to estimate 3D pose. We also showed comparative training results. Finally, the proposed architecture was evaluated on H3WB dataset and presented the evaluation results considering the evaluation metrics of the mean per joint position error (MPJPE) and the percentage of correct keypoints (PCK). The proposed architecture performed about 50% better in terms of MPJPE and PCK@150mm for three residual block. We had also compared the performance of HEpose with other state-of-the-art methods of 3D pose estimator and achieved inevitable performance.

সারসংক্ষেপ

**Transformer and Pose Grammar Based Deep Neural Network for 3D Human Pose Estimation**

বিভিন্ন পরিস্থিতিতে মানুষের কার্যকলাপ এবং আচরণ বোঝার জন্য, মানুষের ত্রি-মাত্রিক অঙ্গভঙ্গি নির্ভরযোগ্যভাবে অনুমান করতে পারা কম্পিউটারের জন্য পূর্বশর্ত। স্বভাবগত ভাবেই এই ত্রিমাত্রিক অবস্থা কম্পিউটারকে বুঝানো বেশ জটিল, এছাড়া আরো কিছু ফ্যাক্টর এই কাজকে আরও জটিল করে তুলেছে। কম্পিউটারকে ক্যামেরার সাহায্যে ধারণকৃত ছবি থেকে মানুষের ত্রি-মাত্রিক অঙ্গভঙ্গি বুঝতে হবে। এখন, একটি ক্যামেরায় ধারনকৃত চিত্রের যে ফ্যাক্টর গুলোর কারনে কাজটির জটিলতা বৃদ্ধি পায় সেগুলো হলঃ চিত্রের গুণমান, পটভূমি, পোশাকের টেক্সচার এবং বৈচিত্র, শরীরের আকৃতি এবং ছবিতে উপস্থিত কোনো ব্যক্তির বা বস্তু দ্বারা আংশিক আড়াল হয়ে যাওয়া ইত্যাদি। যার ফলে, কম্পিউটার ভিশনের কৌশল অবলম্বন করে উক্ত সমস্যার সমাধান করা প্রয়োজন। ইতিমধ্যে দ্বি-মাত্রিক মানব অঙ্গভঙ্গি অনুমানের উপর অনেক কাজ হয়েছে এবং যথেষ্ট সফলতা অর্জিত হয়েছে। মানুষের শরীরের নির্দিষ্ট জয়েন্ট-পয়েন্ট এর দ্বি-মাত্রিক অবস্থান থেকে উক্ত জয়েন্ট-পয়েন্টের ত্রি-মাত্রিক অবস্থান নির্ণয় করে মানব অঙ্গভঙ্গি অনুমান করাই হল এই গবেষণার মূল উদ্দেশ্য। আমরা রেসিডুয়াল ব্লক এর সাথে লিনিয়ার লেয়ার সমন্বয়ে গঠিত Deep নিউরাল নেটওয়ার্ক প্রয়োগ করে গভীরভাবে পর্যালোচনা করেছি। এবং অবশেষে গবেষণার উদ্দেশ্য অর্জনের লক্ষ্যে হাইব্রিড কাঠামোর প্রস্তাব করেছি। কার্যক্ষমতা বিশ্লেষণ করার জন্য রেসিডুয়াল ব্লকের সংখ্যা বাড়িয়ে এবং কমিয়ে প্রস্তাবিত অবকাঠামোর পরীক্ষা করা হয়েছে। চূড়ান্ত প্রস্তাবিত আর্কিটেকচার (HEpose) তিনটি সমান্তরাল মডেলের সমন্বয়ে গঠিত। প্রথম মডেলের ভিত্তি হিসেবে লিনিয়ার লেয়ার এর সাথে দুইটি স্কিপ কনেকশন নেওয়া হয়েছে, দ্বিতীয়টি রেসিডুয়াল সংযোগের উপর ভিত্তি করে গঠন করা হয়েছে যেখানে কোনো ধরনের নরমালাইজেশন করা হয়নি এবং তৃতীয় মডেলটি বানানো হয়েছে গ্রাফ কনভোলুশন দিয়ে যা জয়েন্টগুলির মধ্যে সংযোগের তথ্য সংগ্রহ করে। উক্ত তিনটি মডেলের আউটপুট একত্রিত করে একটি সম্পূর্ণ সংযুক্ত লিনিয়ার লেয়ার (FC Layer) ব্যবহারের মাধ্যমে চূড়ান্ত ত্রি-মাত্রিক অঙ্গভঙ্গি অনুমান করা হয়েছে। অতঃপর, এই প্রস্তাবিত কাঠামোর তুলনামূলক প্রশিক্ষণ পর্বের ফলাফল বিশ্লেষন করা হয়েছে। অবশেষে, প্রস্তাবিত আর্কিটেকচারটি H3WB ডেটাসেট দ্বারা মূল্যায়ন করা হয়েছে এবং ফলাফল Mean Per Joint Position Error (MPJPE) এবং Percentage of Correct Keypoint (PCK) এর মাধ্যমে বিশ্লেষন করা হয়েছে। প্রস্তাবিত আর্কিটেকচারটি তিনটি রেসিডুয়াল ব্লকের জন্য MPJPE এবং PCK@150mm এর পরিপ্রেক্ষিতে প্রায় ৫০% ভালো করেছে। পরিশেষে, প্রস্তাবিত HEpose প্রচলিত অন্যান্য ত্রি-মাত্রিক মানব অঙ্গভঙ্গি অনুমানকারী পদ্ধতির চেয়ে তুলনামূলক (কর্মক্ষমতা) ভাল ফলাফল অর্জন করেছে।

# ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and appreciation to the individuals who have been instrumental in the completion of this thesis book. Their support, encouragement, and guidance have been invaluable throughout this academic journey.

First and foremost, I would like to state unequivocally that ALLAH is the source of all evaluations. He has given me the ability to carry on with and finish this research work. I thank Allah profusely for all of His blessings.

My heartiest gratitude, profound indebtedness and deep respect go to my supervisor, Professor Dr. Md. Hasanul Kabir, Professor, CSE Department, Islamic University of Technology (IUT), for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advice throughout the study was of great help in completing this research smoothly. I am also grateful to my co-supervisor Muhammad Nazrul Islam, CSE Dept, MIST for his constant support to carry on the work and to give me the scope to meet the deadlines. Special thanks go to my previous head of the department Brig Gen Md Abdur Razzak, Brig Gen Md Mahfuz Karim Mazumder, and the current head of the department Brig Gen Mohammad Sajjad Hossain and the Computer Science and Engineering (CSE) department at Military Institute of Science and Technology (MIST) for providing the necessary resources and a conducive academic environment for the completion of this thesis. I would also like to give special vote thanks to Assistant Professor Sharifa Rania Mahmud and Lecturer Tasmiah Tamzid Anannya for their insightful inspiration. Finally, I would like to thank my beloved parents for their support, motivation, patience and suggestions during the course of my thesis work.

This thesis book stands as a testament to the collaborative efforts of many, and I am truly grateful for the contributions of each individual mentioned and those who may not be explicitly named but have played a part in this academic endeavor. Thank you all for being an integral part of this journey.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ALGORITHMS

# CHAPTER 1
# INTRODUCTION

Human pose refers to the way a person's body posture or attitude is presented in different situations or action that is perceived by human eye. That is how different body parts of a person is arranged in any moment. The task which deals with identifying this human pose is known as human pose estimation. Human pose can estimated in different dimension like estimating in two-dimension (2D) or in three-dimension (3D). This chapter will represent overview of the pose estimation in 3D, applications, motivations, research objectives and book organization.

## 1.1 Thesis Background

Human 2D pose estimation means estimating the body pose in a two-dimensional coordinating system. The task of human 2D pose estimation is done by identifying the (x, y) coordinate position of different bone joint positions based on which body parts can be moved. Number of joints is subject to the available dataset. To represent a human skeleton in some datasets 14, 16, 17, 133, or more joint positions can be used. With the progress of 2D pose estimation, later on, this task has been extended to the real-world dimension which is known as 3D human pose estimation. In 3D human pose estimation each joint position is represented with the depth information i.e. (x, y, z) values.

The task of human 3d pose can be estimated from image or video. Image is a two dimensional array of pixel, where each pixel carries the intensity level of red, green and blue channel. In that image, a persons joint locations are estimated in 3D coordinate system. For video, frame is taken to estimate 3D pose.

Image or video is captured using camera, this cameras' different parameter like extrinsic, intrinsic parameters can be used to reconstruct the human posture in 3D. To enhance the

efficiency and precision of the estimated 3D pose, images of the same persons posture can be taken from different angle which is also known as multi-view. 3D pose can be estimated from single-view i.e. single camera's image or multi-view i.e. multiple camera's image from different side or angle.

In computer vision 3D human pose facilitate to understand the human behaviour, activity or action more accurately. Thus improvement in this research domain is highly required. There are challenges due to the image quality as the input image from which 3D pose will be estimated can be of low resolution, or high resolution or in bright sunlight or in low sunlight or in the night light. Moreover, person in the image may have variety of cloths with variety of texture in it which also create the task more challenging as the location of the joint got hidden by the cloths. Again, due to occlusion of body parts in the image that is some body parts got hidden somehow which makes the task further difficult. Occlusion can be due to some object or due to another person or by the persons self occlusion (e.g. side view). Finally the background of the image or video also needs to be considered for the better generalization of the estimation model. The more accurate human 3D pose can be estimated, the more precisely the action can be identified as well as analysed. This estimation can also be used in surveillance, augmented reality, virtual reality, in vehicle automation, scene analysis, sports analysis, robotics and in many more. Getting inspired from the vast area of application, we have stated our journey to do research on human 3d pose estimation. In this work, we have tried out several concept as well as architecture of deep neural network and finally able to identify an architecture which performed better in terms of MPJPE and PCK.

## 1.2 Motivation and Problem Statement

The motivation behind undertaking this thesis stems from a profound curiosity and passion for understanding the intricacies of Human 3D Pose Estimation. In an era marked by rapid technological advancements and evolving societal challenges, the need for comprehensive insights into human 3D posture has been more crucial. From a personal perspective, my

fascination with human 3D pose estimation developed during the exploration of the recent computer vision enhancement. As the domain of computer vision research empowering many applications to automate the AI based systems. Moreover, with the intuition of human 3D pose, in many segments where people are living alone like old age care home, it can facilitate understanding the scene and act accordingly for the faster response. Moreover, the academic importance of this research lies in its potential to contribute to a theoretical framework, address a gap in the literature, or challenge existing paradigms. By exploring deep neural networks within the context of human 3D pose, I aim to provide a nuanced understanding that can inform future research endeavors and guide practical applications. Furthermore, the societal implications of this research are significant. This can be used in vehicle automation, for the monitoring of senior citizens, on street violence detection and many more. The potential practical applications of this research extend beyond academia, with implications for industry, security sectors as well as government. Through this research journey, I aspire to not only deepen my own understanding of human 3D pose but also to contribute valuable knowledge that may shape future research directions and have a positive impact on our broader community.

Human pose estimation involves finding body posture from images or videos in 2D or 3D space. It has a wide range of real world applications such as activity recognition, augmented reality, sports analysis, violence recognition and so on. The challenge is to estimate accurate localization of body parts in world coordinates. Several methods exist for 2D and 3D pose estimation considering different types of input and number of cameras (e.g. single image, video, monocular, multi-view camera). However, in recent days, researchers have paid attention to deep learning based methods for estimating 3D pose Chen et al. (2020); Sarafianos et al. (2016). For example, Martinez et al. Martinez et al. (2017) divided the process into two parts: firstly RGB image to 2D pose detection using stacked-hourglass, then 2D to 3D mapping using a simple linear model. In another study Hossain and Little (2018), the temporal information across a sequence of 2D joint locations has been utilized to estimate 3D pose using a sequence-to-sequence network composed of layer-normalized Long Short-Term Memory units. Again, a deep convolutional neural networks with synthe-

sized training data used for boosting the performance in [5]Chen et al. (2016). Kocabas et al. Kocabas et al. (2020) have used a Skinned Multi-Person Linear body model integrating with a self-attention mechanism to explore plausible motion sequences, while Xu et al. Xu et al. (2021) have included pose grammar with the Bidirectional recurrent neural networks for achieving robustness against appearance variations and cross-view generalization.

The increasing demand of computer vision applications of human 3D pose estimation poses a significant role to the automation systems. Despite the implementation of various measures, there is still scope for the betterment of the task. This research seeks to investigate the specific techniques to estimate human pose in three-dimension to facilitate computers understanding more accurately the scene and to help people to give better intuition in any analysis where human posture is highly required.

## 1.3 Thesis Objectives

The objectives of this thesis are:

1. To explore transformer based DNN architectures for estimating 3D human pose.
2. To develop a DNN architecture with pose grammar for estimating 3D human pose
3. To evaluate the developed architecture in terms of different evaluation metrics for estimating 3D human pose

## 1.4 Methodological Overview

In this section, overview of the research is presented to achieve the objectives. The key steps for the research outline is as follows:

At first, existing deep neural networks have been explored briefly. For this exploration, 2D to 3D pose estimation architectures were mostly focused. As an extension of this exploration, a new DNN architecture is proposed for 133 keypoints from 2D to 3D pose esti-

mation which is described in detail in chapter 4. At first, the architecture were trained on publicly available dataset considering the validation of the model. Then, the performance of the model is measured on the test dataset using the evaluation matrics MPJPE and PCK.

Secondly, the concept of transformer is incorporated in the model considering the RGB image as input and finding the corresponding human's 3D pose estimation. For this, the dataset is processed accordingly from the video files of the dataset. This model is also trained on the dataset considering image as input to output 3D positions of 133 keypoints. Later on, 3D poses are predicted for the test and and stored in a JSON file for further reporting the MPJPE and PCK to represent the model's performance.

Thirdly, the concept of pose grammar is incorporated in the model, which takes the correlation among the joints rather than taking the correlation of the body parts in terms of the kinematics, symmetry and motor-coordinate. This correlation is incorporated in the estimated 2D pose to fine tune the final prediction.

Finally, all the concept of DNN and pose grammar were integrated to depict the final architecture. This architecture constructed considering three module in parallel. That's why we named the architecture as HEpose architecture. Module 1 and Module 2 which are DNN with residual connection based architecture, takes 2d human poses from RGB image as input and generate 3D poses, Module 3 which is a graph convolution based architecture, takes body joint correlations among the joints and 2D poses of the corresponding images as input and generate 3D poses. Output of Module-1, Module-2 and Module-3 is appended together and final 3D poses were generated. Similarly as before, this model was trained and performance were measured.

## 1.5 Organization of the Chapters

The structure of this thesis book is thoughtfully organized to provide a comprehensive exploration of the chosen topic. Chapter 2 delves into the specifics of the human pose, its application, presenting a detailed explanation of existing approach, and representation methods. Chapter 3 serves as the foundation with a thorough literature review, offering insights

into existing research and laying the groundwork for the subsequent chapters. Methodology and implementation are thoroughly detailed in Chapter 4, offering a clear view of the research process and the proposed architecture. The outcomes of the study come to light in Chapter 5, where evaluation results are presented and analyzed. Finally, Chapter 6 encapsulates the entire research journey, summarizing key findings and presenting conclusions drawn from the study, thereby offering a holistic perspective on the subject matter. This sequential organization allows readers to follow a logical progression from background exploration to methodology, results, and ultimately, a comprehensive conclusion.

# CHAPTER 2
# THEORETICAL BACKGROUND

In this chapter, we will discuss on the human pose, and its application. We will also explain the possible way of representing human poses for the computers.

## 2.1 Definition of Human Pose

Pose is a short form of posture. We often say while taking pictures to give any specific pose. Human pose refers to the way of representing human body parts in a particular manner. The human pose estimation problem involves a computational task where the objective is to analyze and interpret visual data, typically in the form of images or videos, with the specific aim of identifying and describing the poses of individuals present in the data. In more detail, the term "pose," refers to the spatial arrangement and orientation of a person's body parts. This includes the positions of key joints such as shoulders, elbows, hips, knees, and ankles. Human pose estimation seeks to locate and characterize these joints in the provided images or video frames, effectively creating a representation of the person's body posture.

## 2.2 Application of Human Pose

In the domain of computer vision, the human pose has a significant role in understanding the real-world scenario more appropriately which will result in better scope for the automation. The applications of human poses are described briefly as follows:

Figure 2.1: Application of human pose estimation in human activity recognition, Image Source

### 2.2.1 Human Activity Recognition

Human activity recognition is a domain of computer vision that helps to automatically determine human actions mostly based on the camera view. Human pose will provide the pose information of any image i.e. spatial information or sequence of images i.e. temporal information. This combination of spatial and temporal information can enrich the activity recognition systems to achieve higher accuracy. This will facilitate monitoring in a wide range. For example, in security sectors, healthcare centers, sports etc. Figure-2.1 pictures effect of pose estimation in human activity in sports sector.

### 2.2.2 Violence Detection

Violence detection refers to the process of automatically identifying instances of aggressive or harmful behavior in various settings, such as surveillance videos, public spaces, or online platforms. This capability is essential for ensuring public safety, and security, and for preventing potential harm. It's important to note that while violence detection technologies hold promise for enhancing public safety, ethical considerations, and potential biases must be carefully addressed to avoid unintended consequences or discriminatory outcomes.

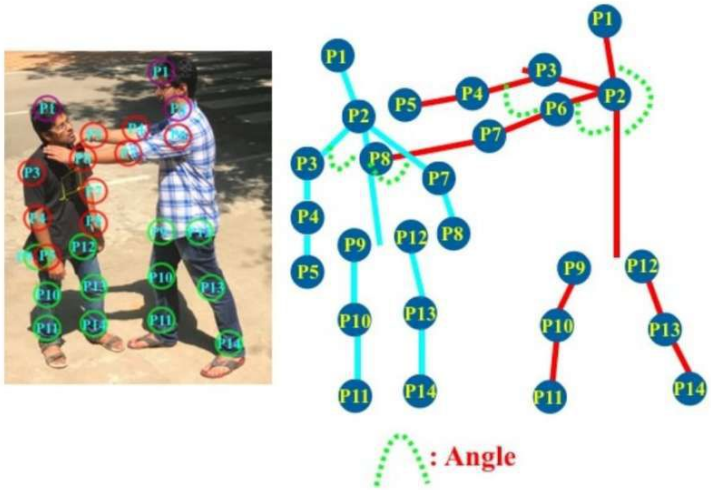


Figure 2.2: Application of human pose estimation for detecting violence in public, Image Source

Additionally, ongoing research and development are essential to improve the accuracy and reliability of violence detection systems. To improve the accuracy, human pose estimation will play a vital role. Figure-2.2 gives a pictorial view of violence detection.

### 2.2.3 Animation

Animation is a visual art form that involves creating the illusion of motion through a series of still images, known as frames. These frames are displayed rapidly in succession, creating the perception of movement. Animation can take various forms, from traditional hand-drawn animation to computer-generated imagery (CGI). For generating motion-captured animation, real-world movement Data needs to be gathered. Humans' movements are recorded using sensors, and this data is then used to animate digital characters. This Human movement data can be generated from images using the human pose estimation technique.

### 2.2.4 Augmented Reality and Virtual Reality

AR and VR technologies continue to advance, and their applications span various industries, including entertainment, education, healthcare, and enterprise. AR overlays digital information (images, text, 3D models) onto the real-world environment, enhancing the user's perception of reality. Users can see and interact with the real world while digital

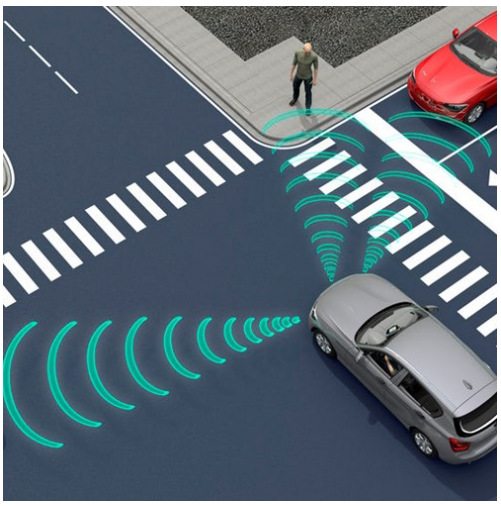

Figure 2.3: Application of human pose estimation in vehicle automation, Image Source

content is superimposed on it. If this augmented reality needs to project human actors, then human pose estimation techniques will be helpful. As AR and VR technologies continue to advance, and their applications span various industries, including entertainment, education, healthcare, and enterprise. Thus, the AR and VR applications involving humans will need the assistance of human pose estimation.

### 2.2.5 Vehicle Automation

Vehicle automation and human pose estimation can be related in the context of advanced driver-assistance systems (ADAS) and autonomous vehicles. Human pose estimation, in this context, refers to the ability to detect and understand the body posture and movements of occupants inside a vehicle. Integrating human pose estimation with vehicle automation systems can facilitate driver monitoring, occupant safety, autonomous vehicle interaction with drivers, occupants as well as pedestrians.

There are many other applications in this domain. For example, in medical assistance, human computer interaction, inference of pedestrians for car automation, sports analysis,

personal fitness care applications, etc.

## 2.3 Representation of Human Pose

The representation of human pose refers to the depiction of the spatial configuration of a person's body, typically in the form of key points or joints, to capture the pose or posture accurately. The choice of representation depends on the specific application and the requirements of the task at hand. Advances in deep learning and computer vision have led to the development of sophisticated pose estimation models that can effectively capture human pose in various contexts, contributing to applications such as activity recognition, virtual reality, and human-computer interaction. The pose of a human can be represented in different forms. They are described below:

### 2.3.1 Skeleton Representation

Skeleton-based representation adopts the structure of human structure as shown in Figure-2.4 to construct the skeleton-based representation as shown in Figure-2.5.

Figure 2.4: Human Skeleton

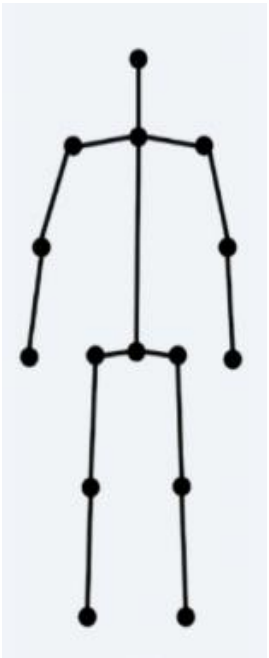

Figure 2.5: Skeleton Based Representation

It can be shown using joint coordinates representing the human pose using the 2D or 3D coordinates of specific joints. Mostly used joints include the head, shoulders, elbows, wrists, hips, knees, and ankles. For the 2D pose, X and Y coordinates are used for the joints, and for the 3D pose, an additional Z coordinate is added to represent the depth. Sometimes, joint angles are also used for expressing the pose through the angles between adjacent body parts or limbs. It captures the relative orientation of body parts, providing information about the posture and movement. Finally, the skeleton Structure is used to represent the human body as a graph or skeleton, where joints are nodes and limbs are edges. Describes how joints are connected, providing a structural understanding of the body pose. Creating a heatmap for each joint, where the intensity of each pixel represents the likelihood of the joint's presence can also be used to represent joint positions.

### 2.3.2 Contour Representation

A contour can be expressed either as a structured sequence of edges or through a mathematical curve. A curve serves as a mathematical representation of a contour, with instances such as line segments and cubic splines. contour representation of human is shown if Figure-2.6.

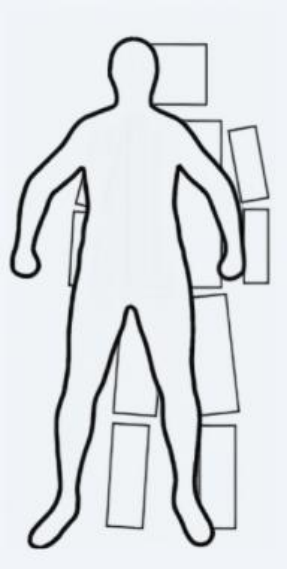

Figure 2.6: Contour Representation

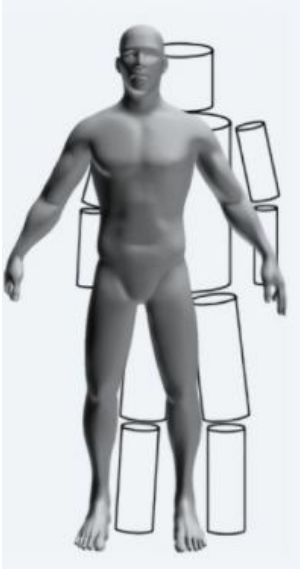

Figure 2.7: Volumetric Representation

### 2.3.3 Volumetric Representation

Representing the human pose in as a volumetric manner is known as volumetric representation. For this kind of representation, a human-like body model is used. Sometimes it is known as mesh representation. In mesh, thousands of points are connected in a specific manner to construct the human structure. Figure-2.7 depicts the volumetric representation. Authors have also focused on the volumetric representation which requires a previously established human body model.

Figure 2.8: Scape Human learnable body model

In Loper et al. (2023), the authors have given SMPL model and in Anguelov et al. (2005) SCAPE body models are introduced for the reconstruction of human pose as volumetric irrespective of pose and shape. Figure-2.8 is the human model of SCAPE: Shape Completion and Animation of People and Figure-2.9 is the human model of SMPL: A Skinned Multi-Person Linear Model.

## 2.4 Approaches of Estimating Human Pose

To estimate human pose different approaches can be incorporated. We can divide this approach into two broad criteria. They are as follows:

Figure 2.9: SMPL Human learnable body model

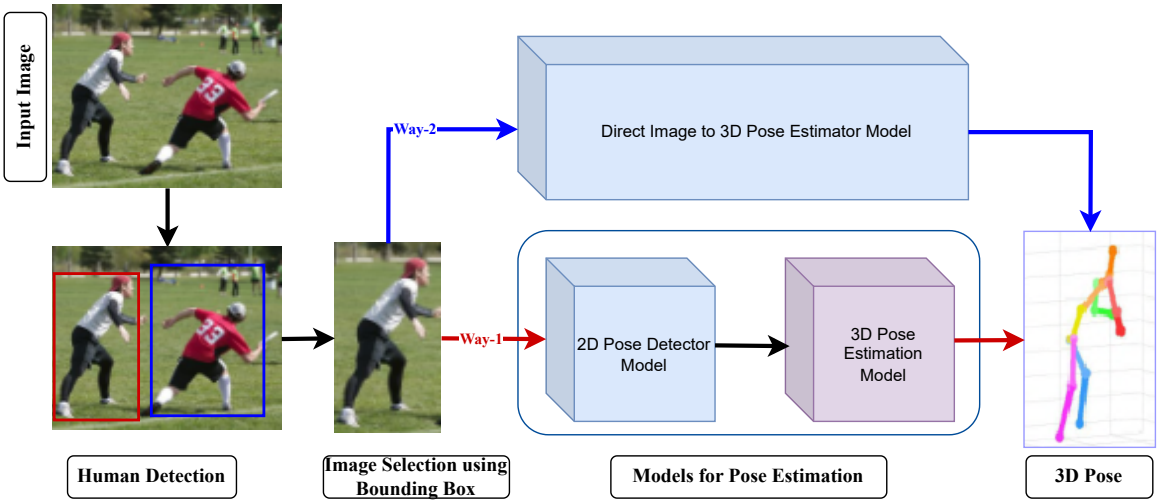


Figure 2.10: Top Down Approach for Estimation Human 3D Pose

1. Top-down Approach

2. Bottom-up Approach

### 2.4.1 Top-down Approach

The process can be broken down into several steps. The input to the pose estimation system is an image or a series of frames from a video, capturing one or more individuals. The approach is shown in Figure-2.10. Initially, the Detection of humans phase is carried out. The system must identify and locate individuals within the input image. This step is often

carried out through object detection algorithms or methods that focus on recognizing human shapes. Then a bounding box around the human will be provided. Based on the estimated bounding box, a cropped image of the one individual human will be created.

Once individuals are detected, the next step is to determine the positions of key body joints for each person. This involves identifying key points such as the head, shoulders, elbows, wrists, hips, knees, and ankles. Different 3D pose estimation models can be used here for the final estimation of the 3D Pose. This 3D estimation can be done by first detecting 2D pose, 2D to 3D pose as shown in way-1 of Figure-2.10. Or it can be done by direct image to 3D pose estimation as shown in Figure-2.10 way-2.

The detected poses are then typically represented in a structured format, such as a set of coordinates for each joint, or a graphical overlay indicating the positions of body parts, or as contour or as mesh/ volumetric representation. The final output of the human pose estimation system is a description or visualization of the detected poses for each person in the input image or video. Many researchers have focused on the uplifting the 3D poses from 2D poses as that the task of estimating human 2D pose has reached to a standard position and there are many extablished state of the art (SOTA) methods.

### 2.4.2 Bottom-up Approach

A bottom-up approach to human pose estimation locates and labels individual body parts or joints first, then combines these localizations to generate the entire human stance. This is in contrast to a top-down method, in which the model initially detects an individual's overall pose before concentrating on fine-tuning the locations of individual joints.

Figure-2.11 depicts the process of bottom-up approach. The Figure shows that 2D keypoint is identified with the help of heatmap or any other method. After that, the key joint points are identified. Therefore, each keypoint is associated with a specific human which is very helpful if the input image contains more than one human. From this, the 3D poses are estimated from the localized 2D positions, and the 3D pose is reconstructed.

There are also some approaches which adopts mixture of both of the approches. This

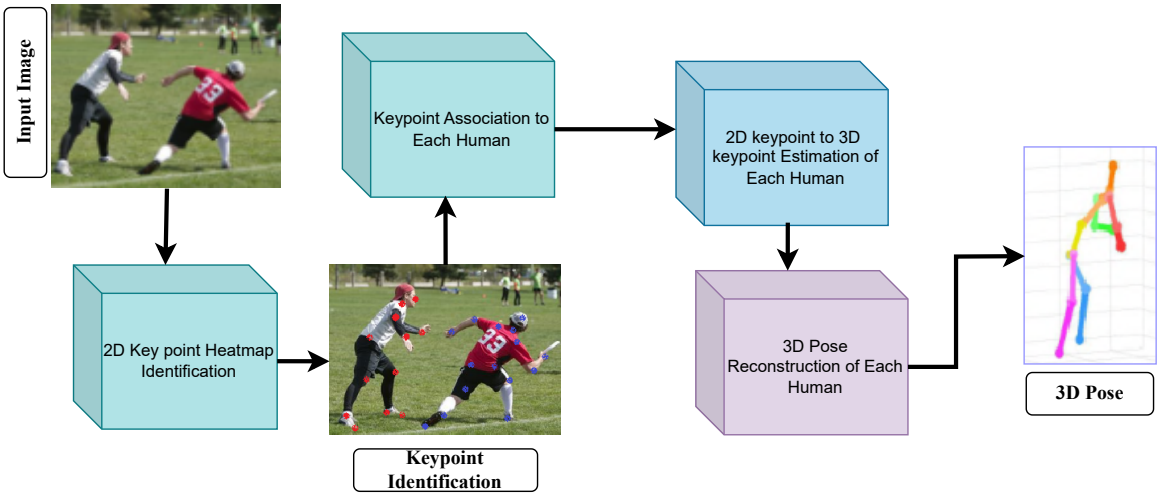


Figure 2.11: Bottom-up Approach for Estimation Human 3D Pose

technology finds applications in various fields, including computer vision, robotics, sports analysis, healthcare, and human-computer interaction. It enables machines to understand and interpret human movements, contributing to the development of systems that can interact with humans more intuitively and effectively.

## 2.5 Concepts and Deep Neural Networks

In this section we will discuss about the two concepts that inspired us a lot to estimate human pose. At first we will discuss on the transformer, then we will discuss on the pose grammar.

### 2.5.1 Transformer

In the context of Deep Neural Networks (DNNs) and Natural Language Processing (NLP), the term ”transformer” typically refers to a specific type of neural network architecture introduced in the paper titled ”Attention is All You Need” by Vaswani et al. (2017). This transformer architecture has since become a fundamental building block for various NLP tasks and has been widely adopted due to its effectiveness. Recently this also drew the attention of the researchers of computer vision. For the task of classification, recently many research work are working on vision transformer which involves only the encoder as

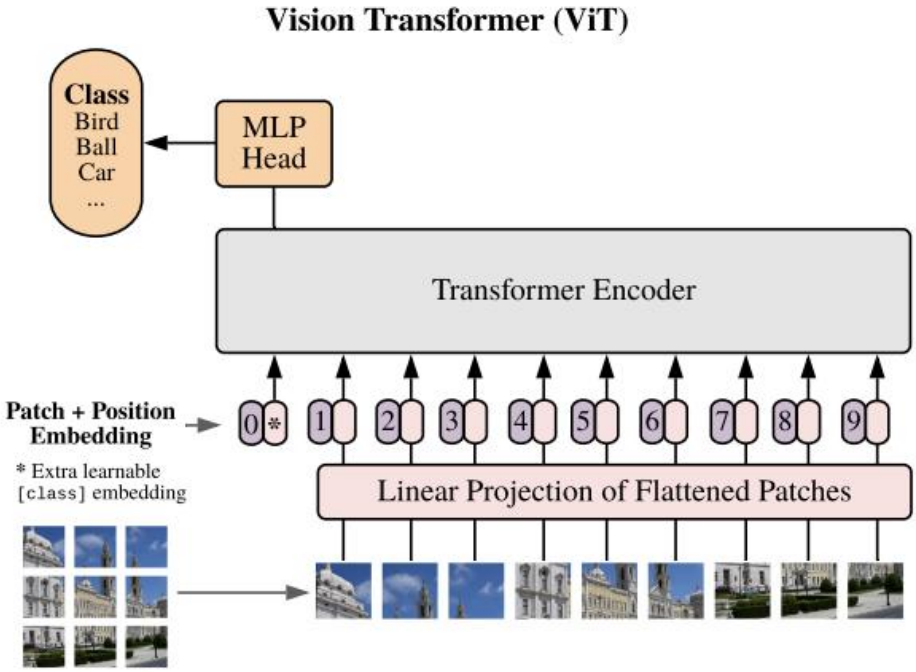


Figure 2.12: Vision Transformer Architecture, Image Source

mentioned by Dosovitskiy et al. (2020). The transformer architecture is based on the self-attention mechanism, which allows the model to weigh different parts of the input sequence differently during processing. This is particularly powerful for tasks involving sequential data, such as language processing.

### 2.5.2 Pose Grammar

Pose grammar is the concept of grammar how the body parts are connected with each other and how they can move arround. Authors of Fang et al. (2018); Xu et al. (2021) have given the idea of human pose grammar. They had given three grammar; one is kinametics, second one is the symmetricity, third one is motor co-ordinate. Figure-2.13 gives a clear idea about the pose grammar of human body.

Kinematic grammar denotes human body motions independently of forces. These frameworks encapsulate the movement characteristics of humans in a topological fashion. Kinematic grammar emphasizes the inter-connectedness of body parts and operates bidirection-

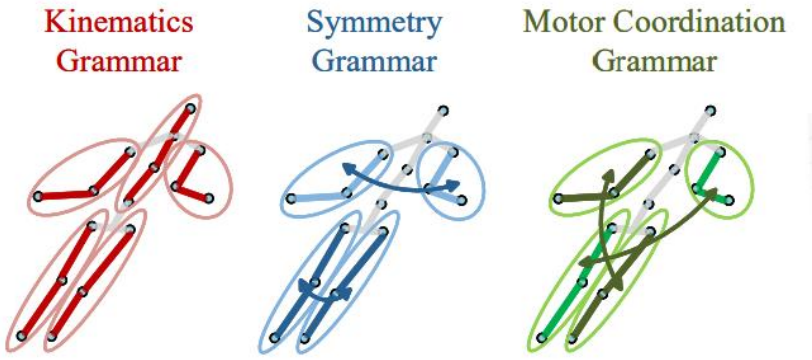


Figure 2.13: Human Pose Grammar of Kinematics, Symmetry and Moro Coordination

ally. Forward kinematics considers the final joint in a kinematic chain, whereas backward kinematics affects a joint in the kinematic chain in reverse from the subsequent joint.

Symmetry grammar assesses the mirror symmetry present in the human body, where a line drawn down the center divides it into matching halves, with the left and right sides exhibiting approximate mirror images of each other. This mirror symmetry, also known as bilateral symmetry, is a fundamental principle in nature, observed in the vast majority of animals including human.

The motor coordination grammar depicts the coordinated movements of multiple limbs in a specific arrangement. Specifically, motor coordination entails the physiological processes incorporating kinematic and kinetic parameters necessary to execute intended actions. Its ubiquity is evident, from simple tasks like moving a limb to more complex actions such as picking up and shooting a ball etc.

# CHAPTER 3
# LITERATURE REVIEW

Human 3D pose can be estimated using different approach. In this chapter different approach and considerations for estimation Human 2D Pose or keypoint of joint location and 3D pose estiomations will be elaborately discussed.

## 3.1 Image to Pose Estimation

Author Xu et al. (2021) have combined two approach for estimating human 3D pose, 3D pose grammar network and a base 3d pose network to integrate human body movement relationships for better 3d pose estimation. To do this, they have used 2d joint positions and image patches centered around the joints as input and predicted 3D pose. They have used Human3.6m, HumanEva and HNOI dataset for evaluating the proposed approach. In addition to these, authors have done a qualitative experiment on MPII dataset.

In Kocabas et al. (2020), an adversarial learning framework is designed using AMASS Mahmood et al. (2019) dataset as well as in the wild 2d pose for finding accurate pose and shape for video sequence. They have used a convolutional network to extract the features of each frame which is sent through a recurrent neural network, Gated Recurrent Unit (GRU), which gives another feature vector based on the previous frame. Their evaluations showed that "Video Inference for Body Pose and Shape Estimation"- VIBE outperforms different existing state-of-the-art models. They have used MPI-INF-3DHP Mehta et al. (2017) and Human3.6M Ionescu et al. (2014) dataset with 17 keypoints.

In another study, a pose grammar is proposed for estimating 3D human pose from a single RGB image in Xu et al. (2021). As input 2D poses of 17 keypoint is provided with image patch centering each keypoint, are fed to a base network which learns the model for 3D poses. Finally these estimated 3D poses are fed to a 3D pose grammar network as input

to interpret the human structure using recurrent neural networks which will again produce 3D poses. These 3D poses are compared with the prior 3D pose found from the base network which defines the loss function i.e. 2D to 3D loss. Further they have provided a data augmentation algorithm so that the model does not over fit the appearance and camera views. After that they have also defined experimental protocol on the Human3.6m Ionescu et al. (2014) dataset for cross-view. Finally they have shown that their proposed model, data augmentation algorithm and experimental protocol for camera view variance can decrease the 3D pose estimation errors significantly.

The author of Mehta et al. (2017) suggested combining a CNN-based 2D pose estimator with kinematics of skeleton, which determines the locations of 2D to 3D joints in real time to create a 3D posture from an RGB camera. They reported the evaluation findings using the MPI-INF-3DHP Mehta et al. (2017) and human3.6m Ionescu et al. (2014) dataset with respect to MPJPE and PCK.

As cross-view corresponding is challenging, the authors of article Lin and Lee (2021), used plane sweep stereo to address both cross-view fusion and reconstruction of multi-view multi-person 3D pose. To do this, they have estimated the 2D pose of each person for each camera with the help of HRNet Sun et al. (2019). Then, depth of each joint position of each person of a specific camera was estimated based on the framework which was built up on the article Collins (1996). In this part, two stages of depth were regressed. First, a person-level depth regressor to identify the depth of each person. And thereafter, joint-level relative depth Regressor is used to regress the depth of each joint position of the specific person. They have also reported their model performance in terms of Percentage of Correctly estimated Parts (PCP) on Campus Berclaz et al. (2011) and Shelf Belagiannis et al. (2014) dataset and Average Precision (AP) and Mean Per Joint Position Error (MPJPE) on CMU Panoptic Joo et al. (2017) dataset.

## 3.2 2D-3D Pose Estimation

Author Martinez et. al. in Martinez et al. (2017), proposed a DNN network to predict 3d positions from 2d ground truth joint locations. The authors have considered 14 to 17 joints to represent human body of Human3.6m Ionescu et al. (2014) to compare the evaluation results with other work. In Ramakrishna et al. (2012), the authors designed a activity independent method ensuring anthropometric regularity to construct a 3d pose from anatomical landmarks of a 2d image. They have used 23 landmarks or positions for each sample. To do this they used CMU Motion Capture Dataset and concatenated PCA components. E. Simo-Serra et. al. in Simo-Serra et al. (2013), proprosed an approach which simultaneously detects 2D parts as well as 3D pose. And finalize the 3D pose combining the estimated 2D pose from image. They have considered 14 joints to do the task. To represent the qualitative evaluation results, the authors have used Humaneva Sigal et al. (2010) dataset. In Schwarcz and Pollard (2018), 3D pose is estimated from 2D pose which is adopted existing state-of-the-art 2D pose detectors from sequence images of multi-view from multiple cameras. They have used 14 joints of a human body and used Campus Berclaz et al. (2011) and Shelf Belagiannis et al. (2014) dataset for this purpose. They have reported percentage of correctly estimated parts (PCP) to evaluate their approach. In paper Zhao et al. (2022), authors stack or two layer named ChevGConv and GraAttention block which is the core part of the graph oriented framework named GraFormer which boosted the performance of the model. They have used three hand dataset which are ObMan Hasson et al. (2019), FHAD Garcia-Hernando et al. (2018), GHD Mueller et al. (2018), and Human3.6m Ionescu et al. (2014) dataset for the evaluation of their proposed model.

## 3.3 Multi-person 3D Pose Estimation

In paper Chu et al. (2021), the authors have worked on multiple human 3D poses which have been estimated from multiple camera views. The authors have proposed two important modules named part-aware measurement for 2D-3D association and a joint filter. The purpose of the joint filter is to mitigate the outlier in 2D poses. The authors have completed

the task by four steps. At first, they detected humans in the given image using some state of the art method which gives a bounding box around a human. Then, they have adopted YoloV3 Redmon and Farhadi (2017) for the estimation of 2D human pose. Later on, 2D poses from different camera views are utilized to estimate 3D human pose which is 2D-3D association. Finally they have reconstructed the 3D human pose by correcting errors using joint filters. The authors have reported the percentage of correct parts (PCP) scores on the benchmark dataset for multi-view human 3d pose estimation named as Campus Berclaz et al. (2011) and Shelf Belagiannis et al. (2014) dataset.

## 3.4 Synthesized and Augmentation Data to Pose Estimation

In paper Chen et al. (2016), the authors have noted that a large number of 3D annotated ground truth dataset is hard to generate, thus near to unavailable. For this, they have presented an approach by which a synthesis dataset will be produced which can be used for human 3D pose estimation. SCAPE Human model is clothing with different texture, background and lightning to produce the annotated database. Finally they have evaluated the 3d human pose estimation task with a very classical CNNs that is AlexNet and VGG to output 3d coordinates and outperforms with other datasets.

In paper Gong et al. (2021), Gong et.al. focused on the performance generalization for new datasets as the train data has a significant role for estimating 2D-3D pose. They have proposed an augmentation technique named as PoseAug which augments the existing training data for incorporating diversity in the dataset. PoseAug adjusts posture, body size, view point and position though an operator. With the help of this PoseAug, the authors have also proposed a part-aware 3D discriminator. The augmentation was done by changing Bone Angle (BA), Bone Length (BL), and Rigid Transformation (RT) with the help of rotation and translation parameters. They have reported their proposed PoseAug frameworks performance on both intra-dataset (i.e. trained and tested on the same dataset) as well as cross-dataset (i.e. trained on a specific dataset and tested on different dataset). To perform these experiments, the authors have chosen Human3.6m Ionescu et al. (2014), 3DHP Mehta

et al. (2017), and 3DPW Von Marcard et al. (2018) publicly available dataset and four estimators known as SemGCN Zhao et al. (2019), SimpleBaseline Martinez et al. (2017), ST-GCN Cai et al. (2019) and VPose Pavllo et al. (2019). As evaluation metrics they have projected Mean Per Joint Position Error (MPJPE) in millimeters and MPJPE over aligned predictions with GT 3D poses by a rigid transformation (PA-MPJPE).

In paper Liu et al. (2022), authors have proposed a Adapted Human Pose (AHuP) framework to enhance the estimation of 3D pose trained on synthetic dataset. They have showed that model trained with fully synthetic data also gives comparable performance with state-of-the-art methods which trained with real 3D data. The authors have worked on five publicly available dataset known as Human3.6m, MuPoTs Mehta et al. (2018), MSCOCO Lin et al. (2014), MPII Andriluka et al. (2014) and SURREAL Varol et al. (2017). Human3.6m and MuPoTs used for real 3D human pose, MSCOCO and MPII used for real 2D pose and SURREAL is used for synthetic human pose. As evaluation result they have reported MPJPE, PA MPJPE, 3D PCK@15cm.

## 3.5 Chapter Summary

In sum the literature review offers a comprehensive overview of various approaches and methodologies for estimating human 3D pose from 2D data. A wide range of methods, from techniques for computer vision like CNN-based estimators to more recent advancements involving adversarial learning and graph-oriented frameworks have been adopted in different studies. It's commendable that studies have been carried out on different datasets for evaluation, including both synthetic and real-world datasets. Each dataset has different characteristics such as the number of samples, diversity in poses, camera viewpoints, having multiple person etc. It is also noteworthy that, for the comparison and bench-marking most of the researchers mentioned evaluation metrics like MPJPE, PA MPJPE, and 3D PCK@15cm. from this literature review, it is observed that the authors have applied different concept to estimate 3D pose of human with better performance. And there is still option to improve the performance. Moreover, very few works has been conduction considering

the concept of giving attention to the joint relation. Thus, the objective of this research is to propose a deep neural architecture which is incorporated with the joint relation to estimate 3D human pose.

# CHAPTER 4
# METHODOLOGY AND PROPOSED ARCHITECTURE

This chapter will discuss on the elaborated methodology that is followed to carry out the research work. Thereafter the explored model's details will be described and the proposed architecture will be explained.

## 4.1 Methodology

Methodology refers to the systematic, theoretical analysis of the methods applied in a particular field of study. It involves the principles, procedures, and rules that guide research or other types of work. Methodology is crucial in ensuring the validity and reliability of results. Figure-4.1 represents the workflow of the methodology we have followed. At first, we have thoroughly explored different research domain and identified the interested one. After the selection of the research domain, we have explored recent work on computer vision and its application. From this, we came to know about the motion-captured sculpture of human from one of the reputed research labs. Thereafter, we looked for the relevant articles and explored its content. The first thing that we pay heed to, is the Dataset, and how it can be used. As a part of this we explored available datasets for example Human3.6m Ionescu et al. (2014), 3DPW Von Marcard et al. (2018), SURREAL Varol et al. (2017), H3WB Zhu et al. (2023), etc. At the same time, we explored approaches that followed Deep learning-based methods. Among a lots of concepts, Transformer, concept of pose grammar, and architectures of deep neural network drew our attention. From this, we selected to work on H3WB dataset and explored DNN architectures. We had built hybrid architectures and trained these using H3WB dataset. Then performance of the models have been analyzed, compared, and reported on the selected test dataset as well as the provided test dataset by Zhu et al. (2023) accordingly.

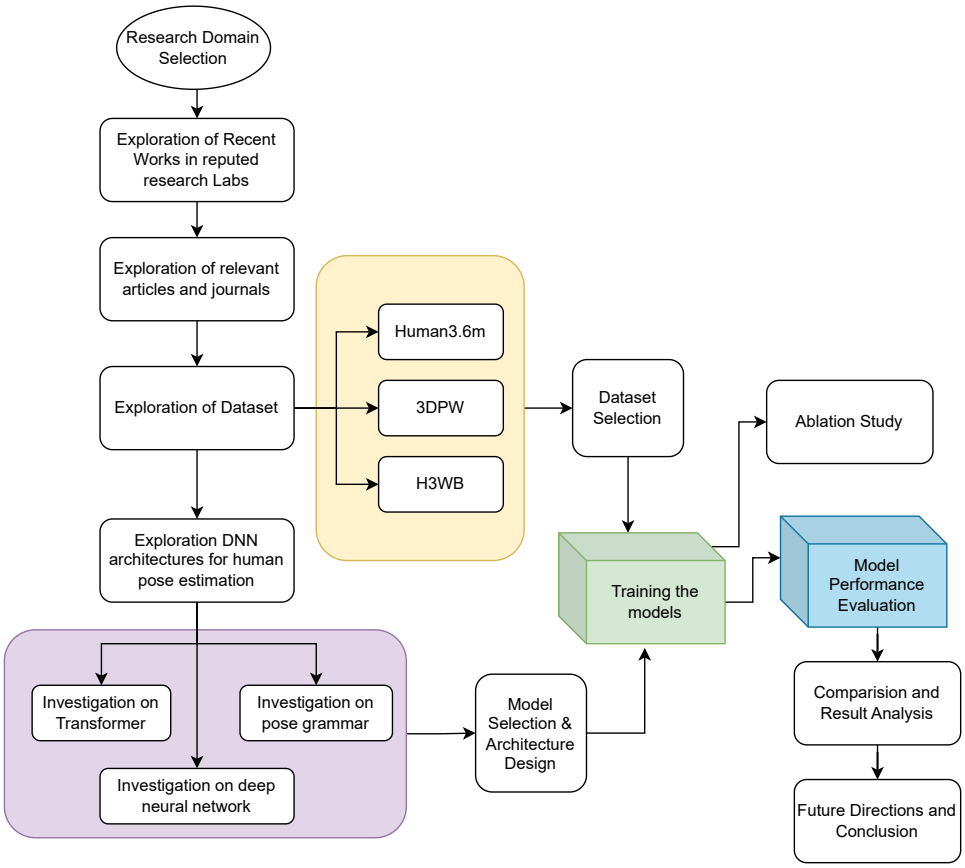


Figure 4.1: Workflow of the research methodology to estimate human 3D poses

## 4.2 Building the Model Architectures

This section elaborately describe the model architectures, that is followed to carry out the intended task. We have considered several DNN architectures adopted from previous works. The initial concept of the research is shown in figure-4.2. This concept diagram consists of two part, one is Transformer and another one is Pose Grammar for Human Body. Recently, the transformer framework has shown remarkable improvement in large language models and vision transformers are also improving the task of image classifications. Thus, the initial conceptual framework of this research was to feed the 2D pose or image in the transformer model and gather information of it. At the same time, as human pose grammar i.e. symmetry, kinematics and motor coordinate has an impact on estimating human pose, we would like to feed 2D poses along with the connections of joints to the pose grammar framework. Output of this two framework would be combined and feed through linear layer for the final 3D pose estimation. Keeping the concept diagram in mind, first we had

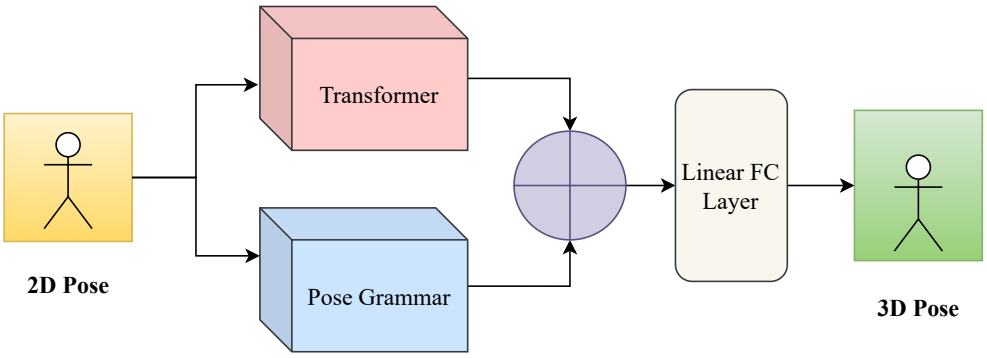


Figure 4.2: Conceptual diagram for the 3D Pose estimation using Transformer and Pose grammar

examined on transformer based architecture, other DNN based architectures and a variant of pose grammar based architecture. The subsections pertaining to articles 4.2.1 through article-4.2.4 present the research we conducted on the transformer models, DNN models, and pose grammar network models.

### 4.2.1 Building Transformer Based Architecture

With the alignment of the concept diagram of figure-4.2, we started to explore transformer architecture. We experimented with the transformer model for a sample of data (20 images) with 100 epochs which gave train vs validation curve as shown in figure-4.3 which indicated a good sign. As the train-validation curve was parallel, we considered that this will perform better if we trained on the full dataset.

Thereafter we trained the model for all 64k samples for 100 epochs but unfortunately the train vs validation curve was not satisfactory which is shown in figure-4.4. Though the train curve was improving, the validation curve crossing over the train curve. Then we started working on why the model was not performing well and stabilizing. After a through check, we found that, the dataset we were using for the purpose was not sequential. And unfortunately, we had no way to get sequential data from the H3WB dataset. Because the H3WB dataset created from Human3.6m dataset and shuffled, so there were no way to get the original sequence of the data.

Therefore we started to look for a different idea instead of using transformer which is de-

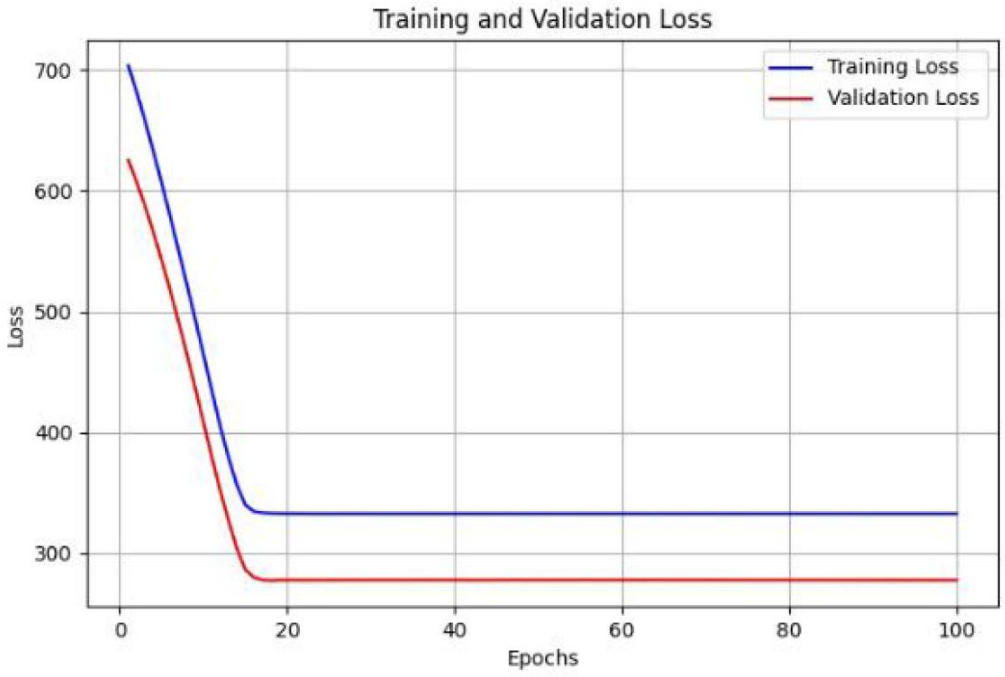


Figure 4.3: Train vs Validation on Transformer framework for 100 epoch on 20 samples

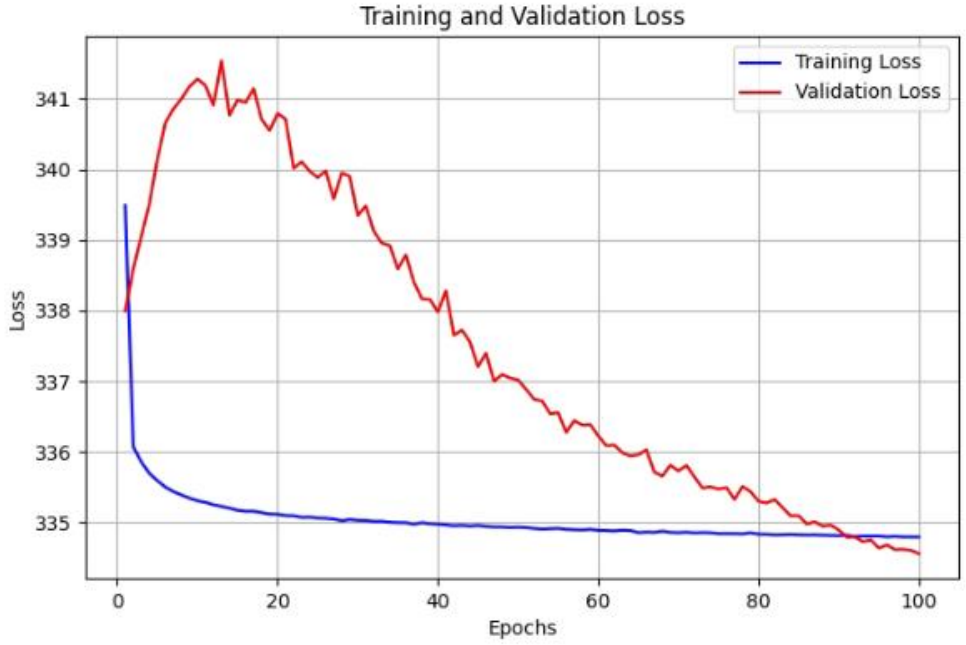


Figure 4.4: Train vs Validation on Transformer framework for 100 epoch on 64K image samples

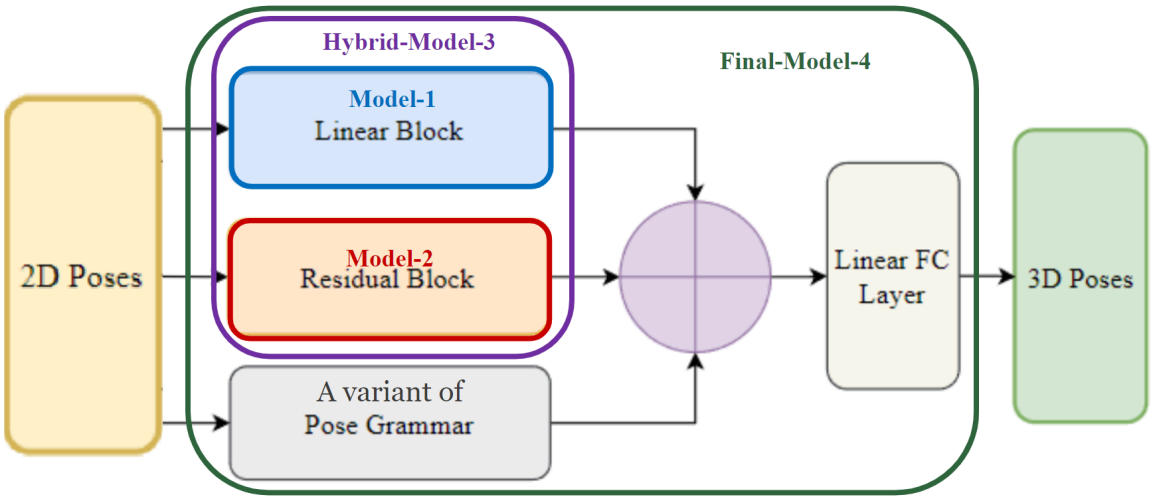


Figure 4.5: Final Conceptual Framework for estimating 3D human pose from 2D joint positions of 133 keypoints using H3WB dataset.

picted in figure-4.5. The figure itself is self-describing. We took 2D poses and fed them to three models parallelly: first model comprises of linear block, second one prioritized the residual block-based model and the third one is a variation of pose grammar based model where we considered only the relations among the joints. Then we combined the results from the three models and fed them to a fully connected layer which outputs the 3D poses of the given 2D pose. This figure-4.5 is the final concept diagram of this research work.

### 4.2.2 Building Linear Layered Based Architecture: Model-1

Model-1 is adopted from network architecture given by Martinez et al. (2017) which considered a number of feed forward linear layer followed by batch normalization, ReLu and Dropout. Figure-4.6 is the architecture of model-1. Each linear layer comprises of 1024 neuron and outputs 1024 neuron except the first and last layer. This 1024 output is normalized and regularized. The model adopts 50% dropout for generalizing the models predictions. The architecture of model-1 architecture was implemented with the help of pytorch framework of python library (See Appendix-A). Model-1 is trained using on the H3WB dataset for task-1 which defines 2D pose to 3D pose estimation of complete body.

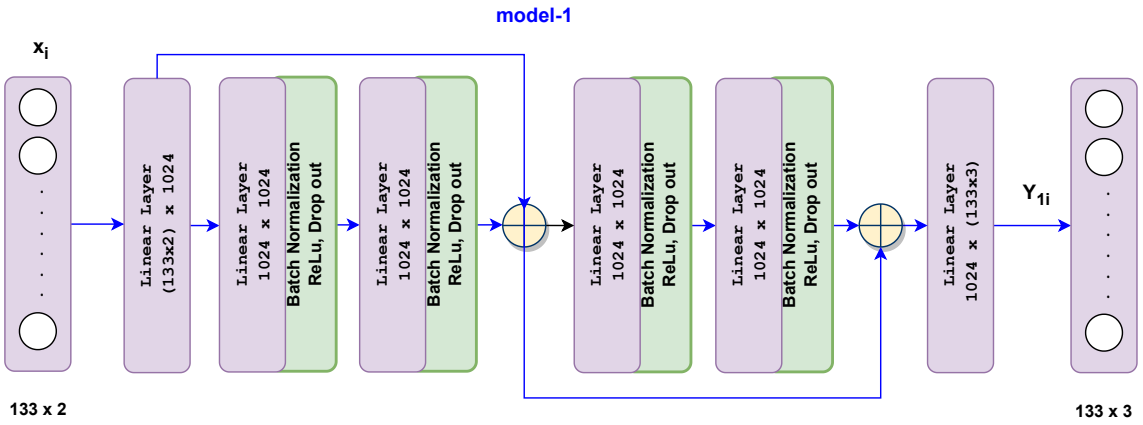


Figure 4.6: DNN architecture based on linear layer (model-1)

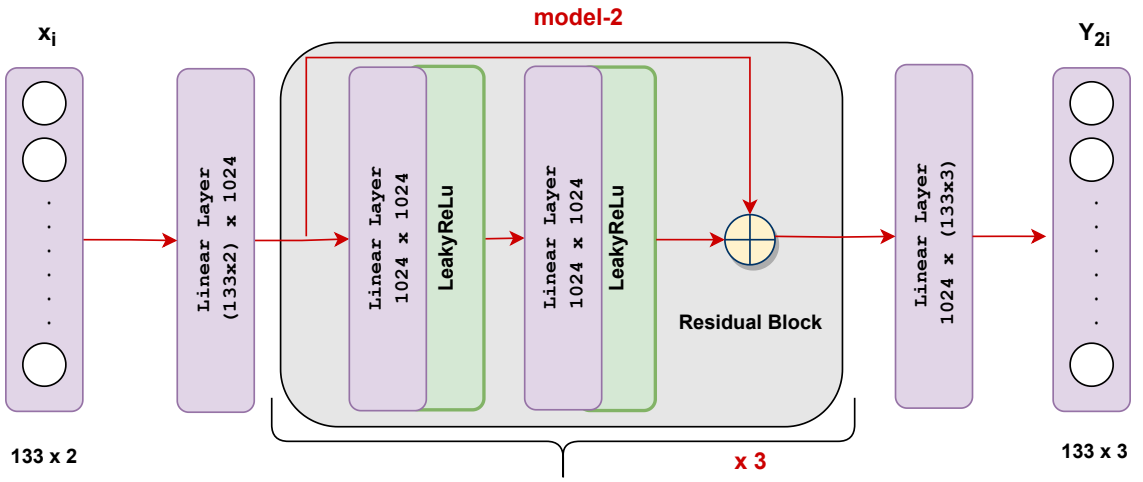


Figure 4.7: DNN architecture with residual bock (model-2)

### 4.2.3 Building Residual Layered Based Architecture: Model-2

This model is the extended version of the model-1. In this architecture, several residual blocks are used without normalization. Several residual block is handled by the value of N as shown in Figure-4.7. In this architecture, instead of using ReLu activation, Leaky ReLu activation function is adopted so that the model can work well with the noise or outlier in the dataset. The architecture of model-2 architecture was implemented with the help of pytorch framework of python library (See Appendix-A).

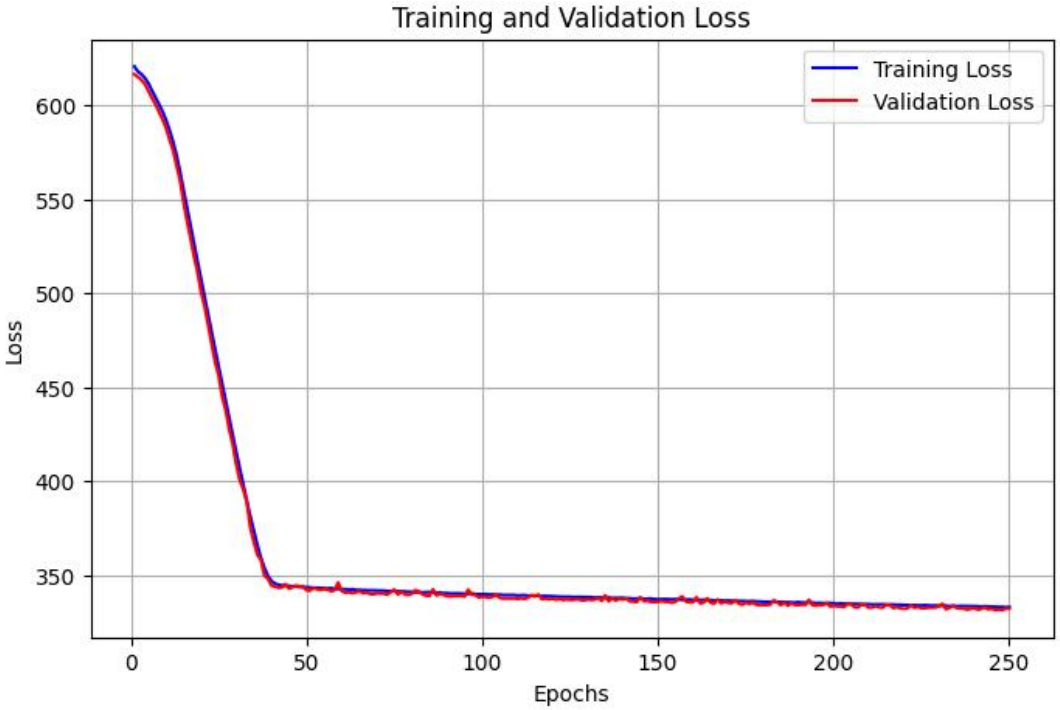


Figure 4.8: Train and Validation curve of pose grammar

### 4.2.4 Building Pose Grammar Based Architecture

In this architecture, we have considered incorporating the pose grammar of human body model as described in Chapter-2. Instead of using the symmetry, motor coordinate or kinematics grammar, a variant of pose grammar is used. That is the relation among the joint e.g. preparing the neighbors of a joint position using adjacency matrix. To implement the this network, I had considered using Graph Convolution Neural (GCN) network as this network performs well with the graph like structure data reported by Kipf and Welling (2016), Chen et al. (2020) and other authors. For giving the relation or connection of the joints an adjacency matrix is created which is graph in nature, so GCN is used for the building the pose grammar based architecture. Example of the training vs validation curve is as represented in figure-4.8. This curve give an intuition to incorporate the information with final architecture for estimating human 3D pose which is described in later section. Implementation of pose grammar is given in Appendix-A.4.

From all of the above exploration of different models, we finalized the followings:

- As we can not get the sequence data from H3WB, we dropped the idea of using Transformer.
- Instead of using Transformer, we used other DNN based architecture with residual blocks.
- As joint connection graph given a better intuition (Figure-4.8), we incorporated this in the final proposed architecture.

## 4.3 Proposed Architectures

In this section, proposed architectures will be described. From the exploration of the different architectures as described in previous section, we have considered proposing two architecture. One is without the pose grammar and another one is with pose grammar. The first one is referred as Hybrid Model-3 and the second one is referred as Final-Model-4 named as HEpose.

### 4.3.1 Proposed Architecture-1: Hybrid Model-3

The proposed architecture (hybrid-model-3) is the combination of both model-1 and model-2 as shown in figure 4.9. Considering the framework of Martinez et al. (2017), We have chosen two architecture. One is simple linear layer based which we referred as model-1 is showed in Fig. 4.6 and another one with a increased linear layer and having residual block (referred as model-2) is showed in Fig. 4.7. Later, combining these two architectures i.e. model-1 and model-2, another architecture (referred as hybrid-model-3) have been proposed which is showed in figure 4.9. It takes the 2D joint positions of 133 joints of each sample as input. This joint positions i.e. $(133 \times 2)$ values are given as input to both model-1 and model-2 with N=3 simultaneously. Both model-1 and model-2 produces 3D pose of 133 keypoints which got concatenated just before the last layer of hybrid-model-3. In the concatenation part the estimated poses of model-1 and model-2 are stacked together and feed to a fully connected linear layer.

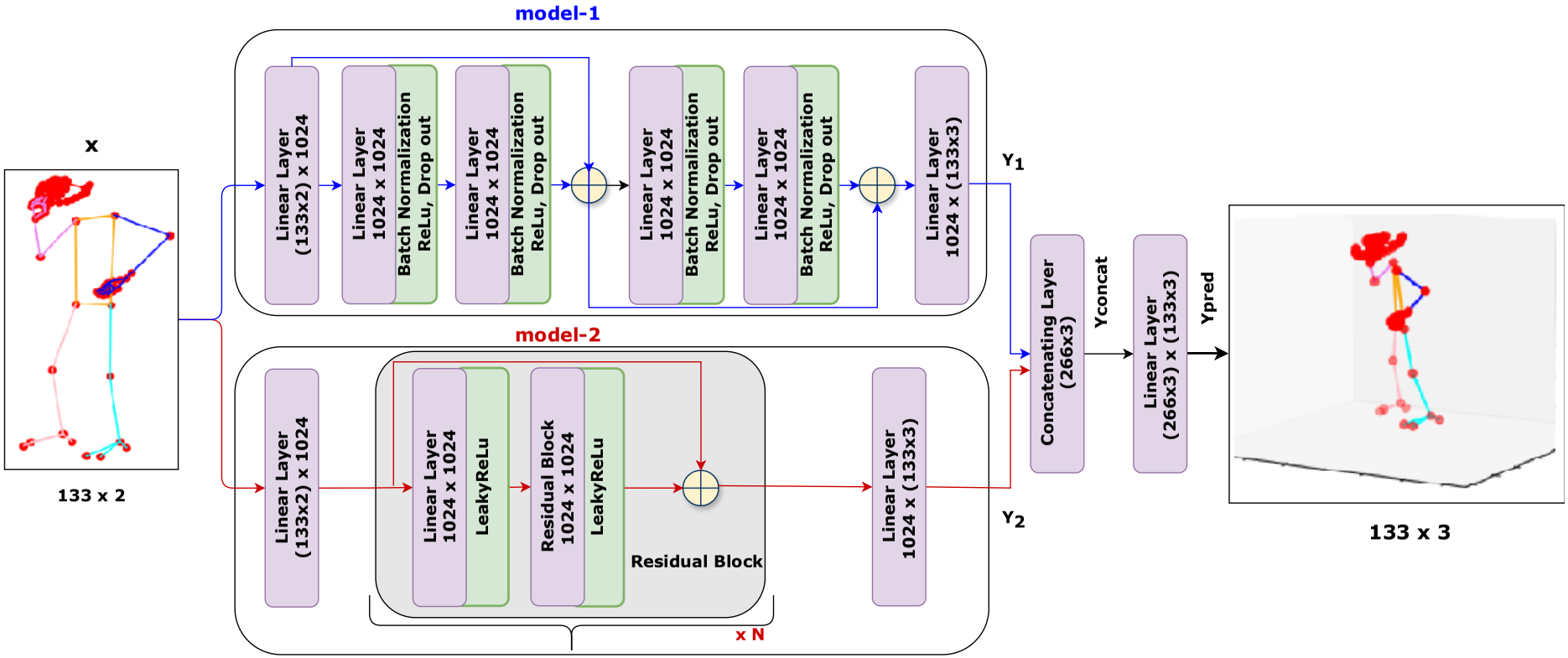


Figure 4.9: Proposed hybrid deep learning framework (Hybrid-Model-3): Combining model-1 and model-2 with two additional layer

This fully connected linear layer produces the final estimated 3D pose of a particular sample. In this hybrid-model-3, output of model-1 and model-2 is concatenated to generate 3D keypoints which consists of three residual block (i.e. N=3) initially and it had been increased to explore how the performance are being varying. Mathematical representation of working process of the hybrid-model-3 of a sample $i$ is as follows:

- Let, $x_i \rightarrow$ each sample having 2D positions i.e. $(x, y)$ values of 133 joint positions. so the size of $x_i$ is $(133, 2)$
- $Y_{1_i} \rightarrow$ 3D positions estimated through model-1 for the sample $x_i$, output size is $(133, 3)$
- $Y_{2_i} \rightarrow$ 3D positions estimated through model-2 for the sample $x_i$, output size is $(133, 3)$
- $Y_{concat} \rightarrow Y_{1_i}$ concatenate $Y_{2_i}$, the output of model-1 and model-2 are appended together. So, size of the $Y_{concat}$ is $(133 + 133, 3) = (266, 3)$
- $Y_{pred_i}$ is obtained from the last fully connected layer which takes input from $Y_{concat}$ and outputs the final pose of sample $x_i$ in 3D.

### 4.3.2 Proposed Architecture-2: Final Model-4 (HEpose)

In this section, the final proposed architecture will be described. It is comprised of Hybrid Model-3 with a variant Pose Grammar. The grammar we have incorporated with the hybrid-model-3 is the connections among the joint positions. The major connections of head, left-arm, right-arm, body, right-foot, left-foot are considered. Joint position numbers are marked in the figure-5.1. According to the marking following connections are considered. We have named this final model-4 as **HEpose**.

- branch_head = [(0,1), (1,3), (0,2), (2,4), (59,64), (65,70),(71, 82), (71,83), (77,87), (77,88),(88,89), (89,90), (71,90)]

- branch_left_arm = [(5,7), (7,9), (9,91), (91,92), (93,96), (96,100), (100,104), (104,108), (91,108)]
- branch_right_arm = [(6,8), (8,10), (10,112), (112,113), (114,117), (117,121), (121,125), (125,129), (112,129)]
- branch_body = [(5,6), (6,12), (11,12), (5,11)]
- branch_right_foot = [(12,14), (14,16), (16,20), (16,21), (16,22)]
- branch_left_foot = [(11,13), (13,15), (15,17), (15,18), (15,19)]

Here, branch_head is the joint positions considered for representing head, branch_left_arm and branch_right_arm represents the edges to construct the right and left hand respectively. branch_body considers the connections among the main body joint points. branch_right_foot and branch_left_foot represents the edges to construct right and left leg respectively.

Figure-4.10 represents the full architecture of proposed final-model-4: HEpose with the module pose grammar incorporated. This joint connection information along with the 2D poses are feed in the graph convolution network from lower dimension to higher dimension that increased from 64 units to 256 units and followed by a linear layer. The linear layer outputs 133x3 values. Output of this linear layer is concatenated with the output of hybrid-model-3 by appending all the estimated value by three parallel path for all joints to get the final estimated 3D poses.

Sample output of the pose grammar model is shown in figure-4.11 without incorporation of the hybrid-model-3's output and 4.12 gives the output after incorporating with the hybrid-model-3.

To analyse the performance of the architectures, Human3.6m 3D Whole Body dataset (H3WB) Zhu et al. (2023) is used which comprise of 133 keypoints of full human body with 2D and 3D annotation for each person. We have used 80k samples of complete 2D to 3D ground truth keypoints. In each samples, body structure, face, both hands are represented with 23, 68, and 42 keypoints respectively which is created from the Human3.6m Ionescu et al. (2014) dataset (main dataset). These are discribed elaborately in Chapter-5.

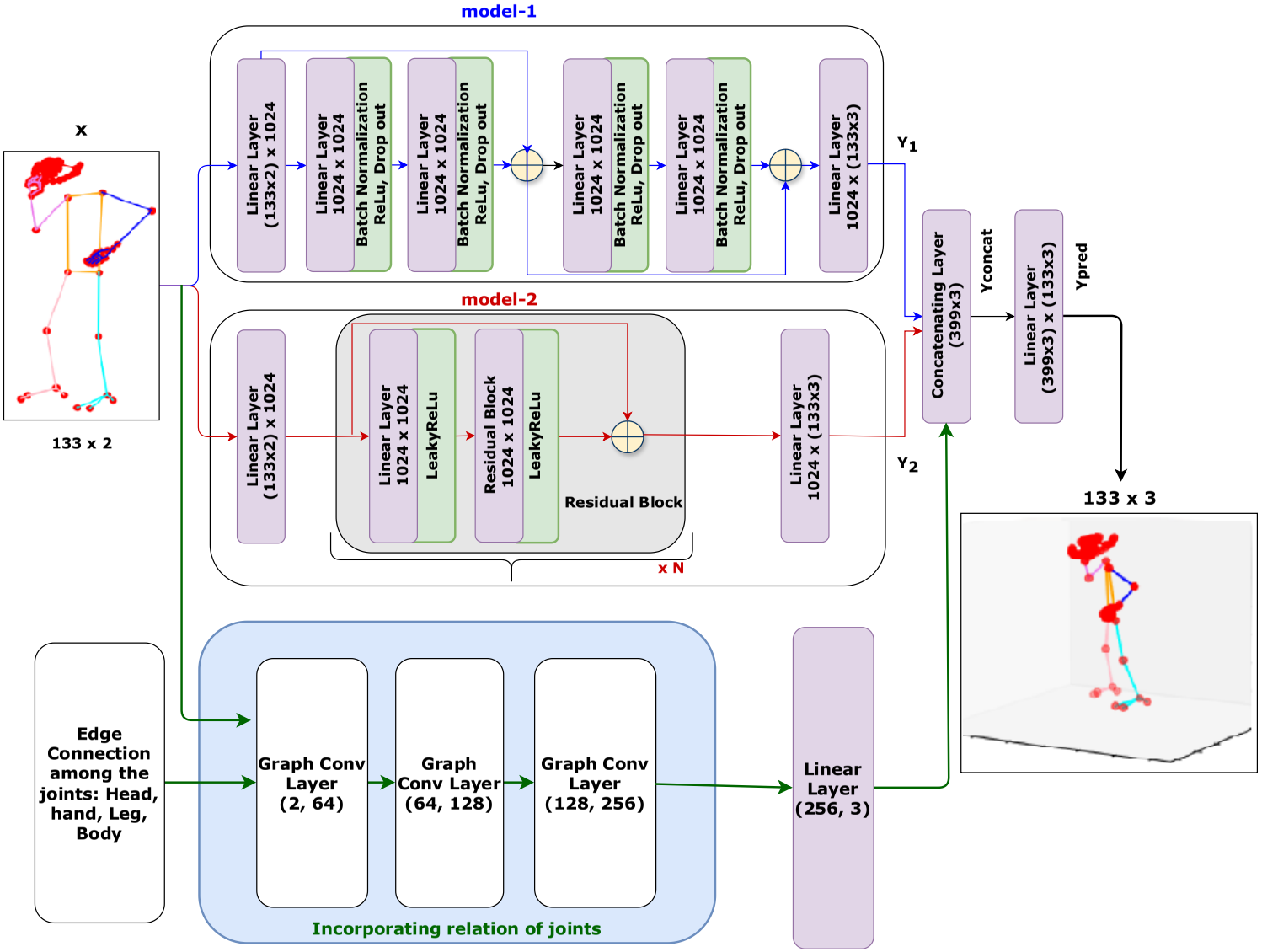


Figure 4.10: Final Architecture (HEpose): Final-Model-4 built with the hybrid-model-3 with joints relations i.e. a variant of pose grammar

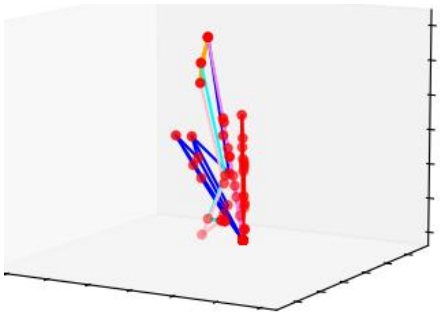

Figure 4.11: Output of Standalone Pose Grammar

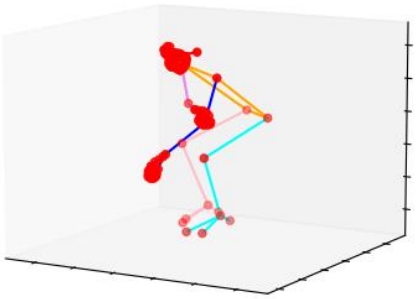

Figure 4.12: HEpose: Output of Pose Grammar while incorporating it with the proposed hybrid-model-3

The architectures model-1, model-2, hybrid-model-3, final-model-4 aka HEpose had been trained on the train dataset from the H3WB dataset and validated during the training. Later on the performance is evaluated on the portion of the dataset which was kept for testing the architecture. Finally, We have simulated the model-1, model-2, hybrid-model-3, final-model-4 aka HEpose one by one and evaluated using the evaluation metrics MPJPE and PCK and reported the results.

# CHAPTER 5
# IMPLEMENTATION AND RESULT

The specifics of implementing the proposed architectures will be covered in this chapter along with the details of the used dataset. Sequential presentations of the training details of the architectures, their performance on the test data, and the analysis carried out on it will also be presented.

## 5.1 Dataset

We have considered Human3.6m and the extended dataset of Human3.6m for training the proposed architectures. Evaluation results are also based on the same dataset. The properties of the datasets are described in the following sub-sections.

### 5.1.1 Human3.6m Dataset

Human3.6m Dataset by Ionescu et al. (2014) is one of the largest dataset containing 3.6 million samples with several annotations like 3D poses with the corresponding images as well as videos. It contains in total 11 professional actors or subjects among them 6 subjects are of male and 5 are of female, who performed poses in 17 scenarios. The scenarios are directions, discussion, eating, activities while seated, greeting, taking photo, posing, making purchases, smoking, waiting, walking, sitting on chair, talking on the phone, walking dog and walking together. All the poses or actions are taken in the lab environment with four camera's mounted in four side of the lab. Ground truth positions of the joints are taken from sensors attached to the joints.

### 5.1.2 Human3.6m 3D Whole Body Dataset

Human3.6m 3D Whole Body (H3WB) dataset by Zhu et al. (2023) is the extension of Human3.6m dataset. This dataset provided three versions for three task. They are: complete 2D pose to 3D human pose estimation, incomplete 2D pose to 3D pose estimation and image to 3D pose estimation. It represents 133 keypoints for full human body with 2D and 3D annotation for each person. In each samples, body structure, face, both hands are represented with 23, 68, and 42 keypoints respectively. Training dataset comprises of 80k samples for 2D to 3D pose estimation data with ground truth keypoints, and testing dataset having 10k samples for both of complete and incomplete 2D-3D task. And for image to direct 3D pose estimation task, the train dataset having 80k training samples containing image path, bounding box around the person, 3D poses and 20k testing samples. This dataset adopts the coco layout of human model which is shown in the Figure-5.1.

For training the proposed architectures and evaluating the models performance, complete 2D pose to 3D pose estimation dataset of H3WB dataset was considered. This dataset has a total of Eighty Thousand samples with ground truth 3D poses. That's why, the dataset is divided into three portion. First portion contains sixty four thousand (64k) samples which was used for training all the models. Second portion contains eight thousand (8k) samples which was used for validating the model during the training phase to understand how the models were performing with the unforeseen data. The last portion was which contains last eight thousand (8k) samples which were used for evaluating the trained model's performance.

## 5.2 Implementation of the Architectures

The proposed architectures were implemented using pytorch framework of python. Sample Python code segments of the implementation is given at the Appendix-A. In Appendix-A.1 sample steps for loading the training dataset, validation and testing dataset is shown. Appendix-A.2 gives the python code of training functions for different architectures. Implementation of model-1, model-2 and hybrid-model-3 is given in Appendix-A.3. Finally,

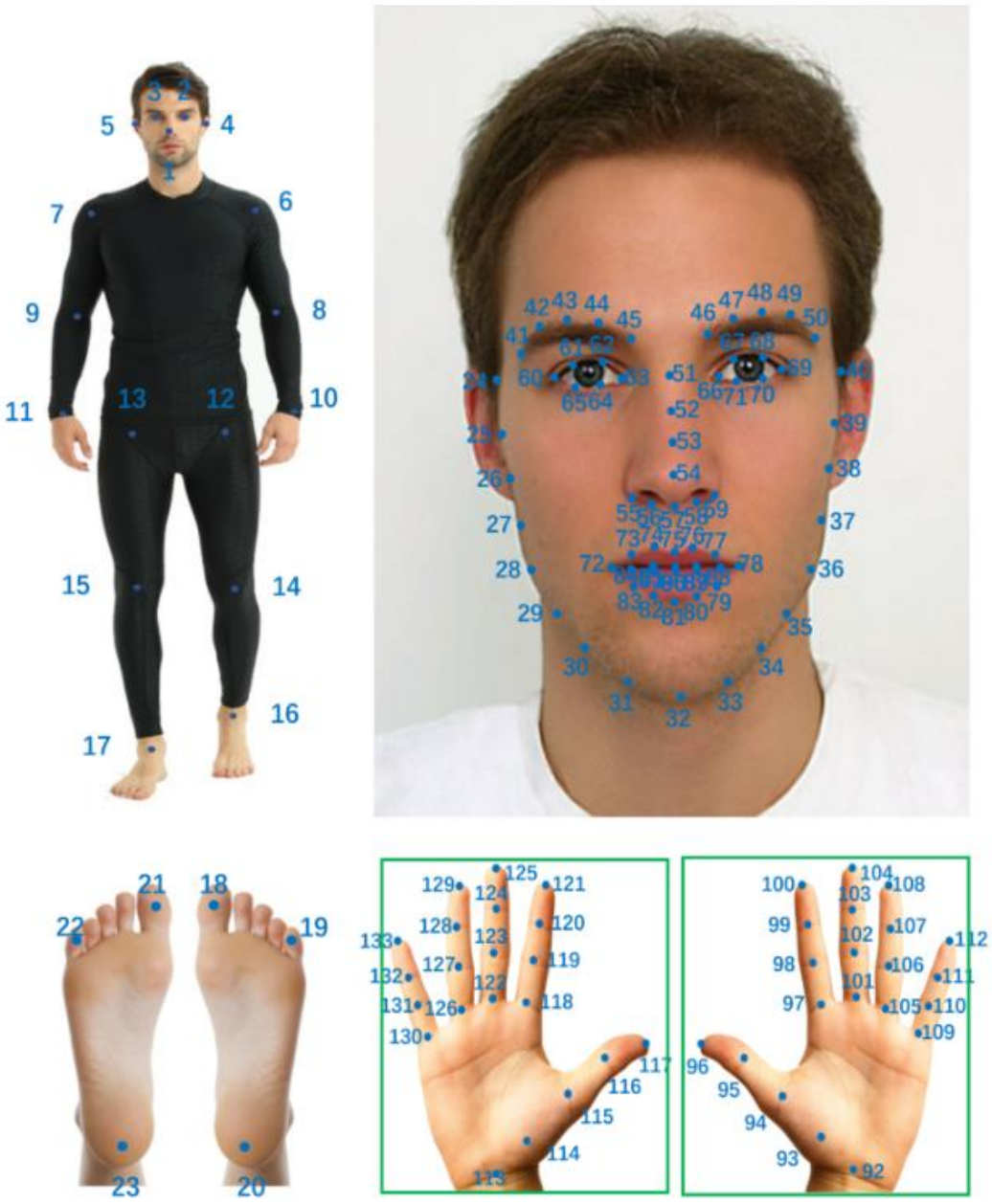


Figure 5.1: COCO Body Layout, Image Source

the Appendix-A.4 gives the final-model-4 architecture i.e. HEpose along with its joint relation selection through adjacency joint creation and corresponding training code.

## 5.3 Training Results of the Architectures

In this section, the results of training phase for all the models will be discussed. Before projecting the proposed architecture, training result of model-1 and model-2 will also be presented here. Following is the experimental setup which was considered during the models training and evaluating phases.

**Experimental Setup:**

Here, considerations to experiment on the implemented models taken, will be described for model-1, model-2, hybrid-model-3 and final-model-4 i.e. HEpose.

We implemented our architecture using python torch framework. As shown in the Fig. 4.6, 4.7, 4.9 and 4.10 the architecture is implemented using pytorch in google colab using TPU-4 GPU runtime environment. The author provided training data of H3WB Zhu et al. (2023) dataset for complete 2D to 3D pose estimation task which indicated all values of all 133 keypoints are present, was divided in to three parts. One part for training, one part for validating during the training and the last part for testing the model. Train, validation and test dataset split ratio was taken 80:10:10. The train data had 80K samples. We have kept 64K samples for training, 8K samples for validating the model during the training and 8K sample for testing the performance of the model. Each networks has been trained for 250 epoch. During the training, the batch size was 64 and learning rate was set to 0.0001. We adopted Adam optimizer during the training of the learning curve for model-1, model-2, hybrid-model-3 and final-model-4 i.e. HEpose. Performance of the model was also evaluated on the author provided 10k test dataset which do not had ground truth 3D poses. To evaluate this, the estimated 3D poses were sent the authors. The authors were very cooperative regarding this matter.

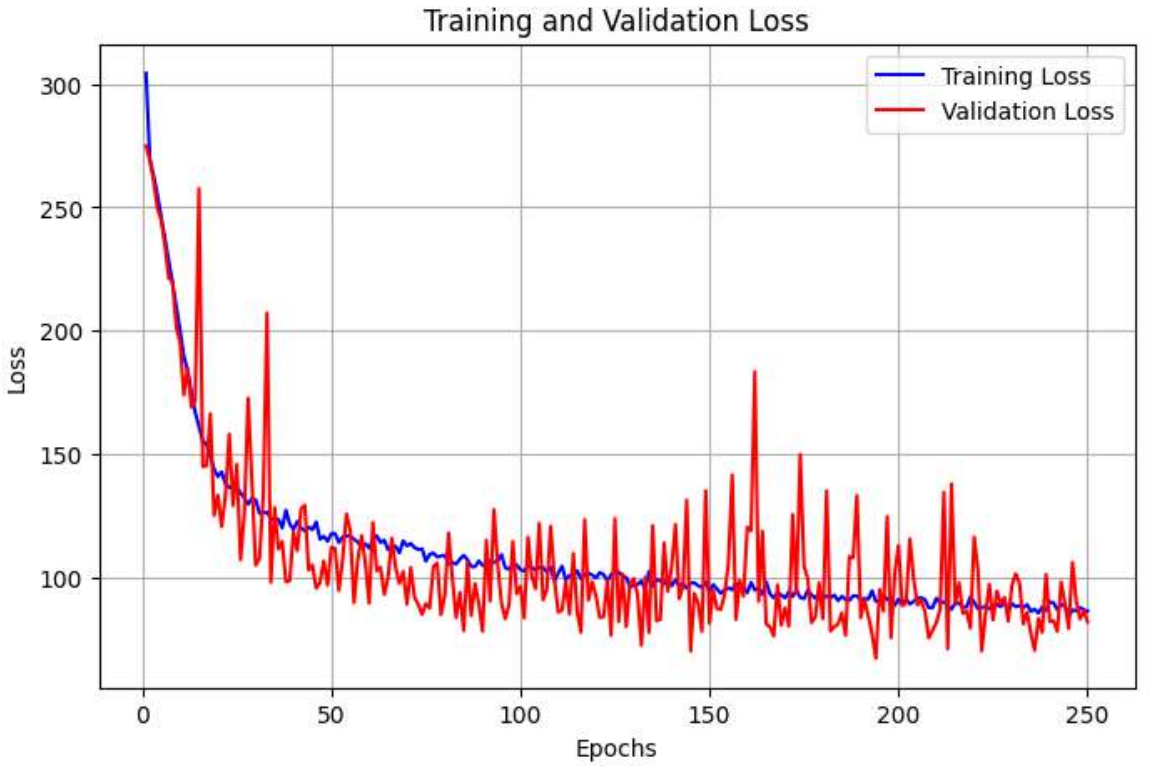


Figure 5.2: Train-Validation Loss of model-1

### 5.3.1 Training Result of Model-1

Training vs validation loss of model-1 is shown in Figure-5.2 for 250 epoch using Adam optimizer. This figure showed that the validation dataset curve fluctuates a lot with respect to training curve. Deviation of the fluctuation of validation curve is about 50 mm and the figure indicated that minimum training loss is around 100.

### 5.3.2 Training Result of Model-2

Training vs validation loss of model-2 is shown in Figure-5.3 for 250 epoch using Adam optimizer. This figure showed that the validation dataset curve became smoother with respect to training curve. It also depicted that the training loss got reduced which is below 50 after 100 epoch.

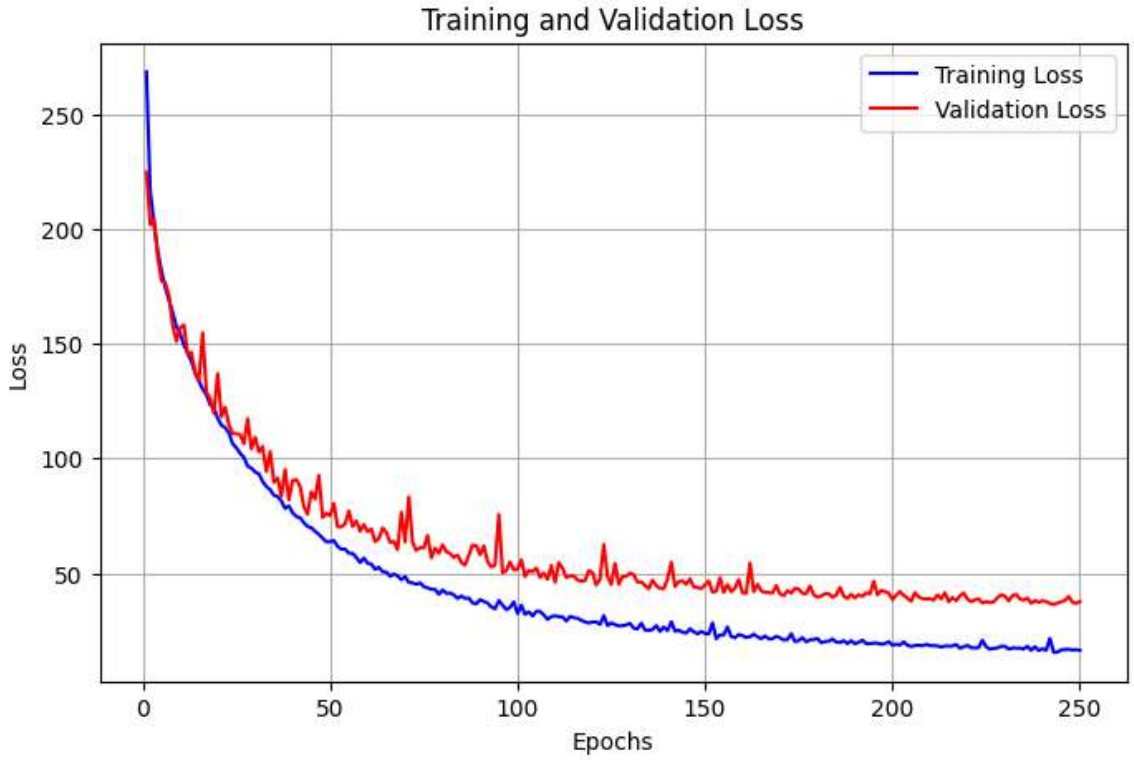


Figure 5.3: Train-Validation Loss of model-2 with 3 residual block

### 5.3.3 Training Result of Hybrid-model-3

Training vs validation loss of hybrid-model-3 is shown in Figure-5.4 for 250 epoch using Adam optimizer with N=3 which means three residual block got cascaded. In this figure, training loss and validation both got reduced to below 50 after 100 epoch.

### 5.3.4 Training Result of Final-model-4: HEpose

Training vs validation loss of the final-model-4 i.e. the HEpose is depicted in Figure-5.5 for 250 epoch using Adam optimizer with N=3. The train-validation curves got stabilized around 200 epoch.

## 5.4 Evaluation Process

In this section, at first we will discuss on the process we followed to evaluation each of the model. The loss function during the training is the mean absolute error which is given as

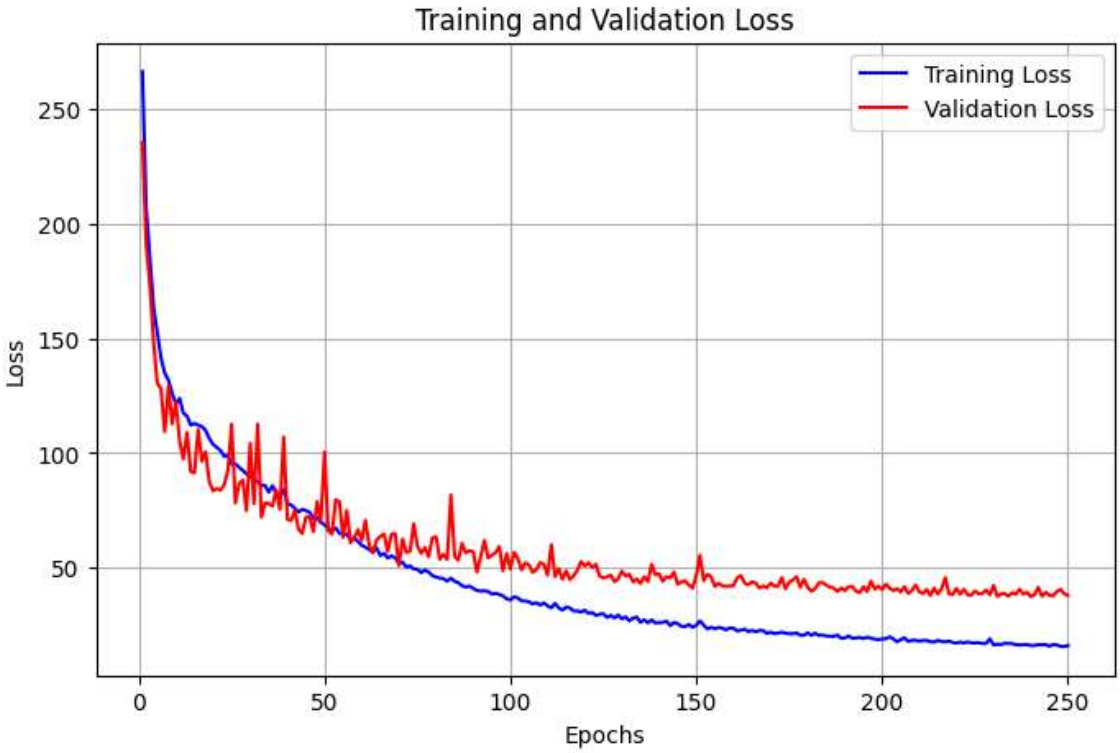


Figure 5.4: Train-Validation Loss of hybrid-model-3 with N=3

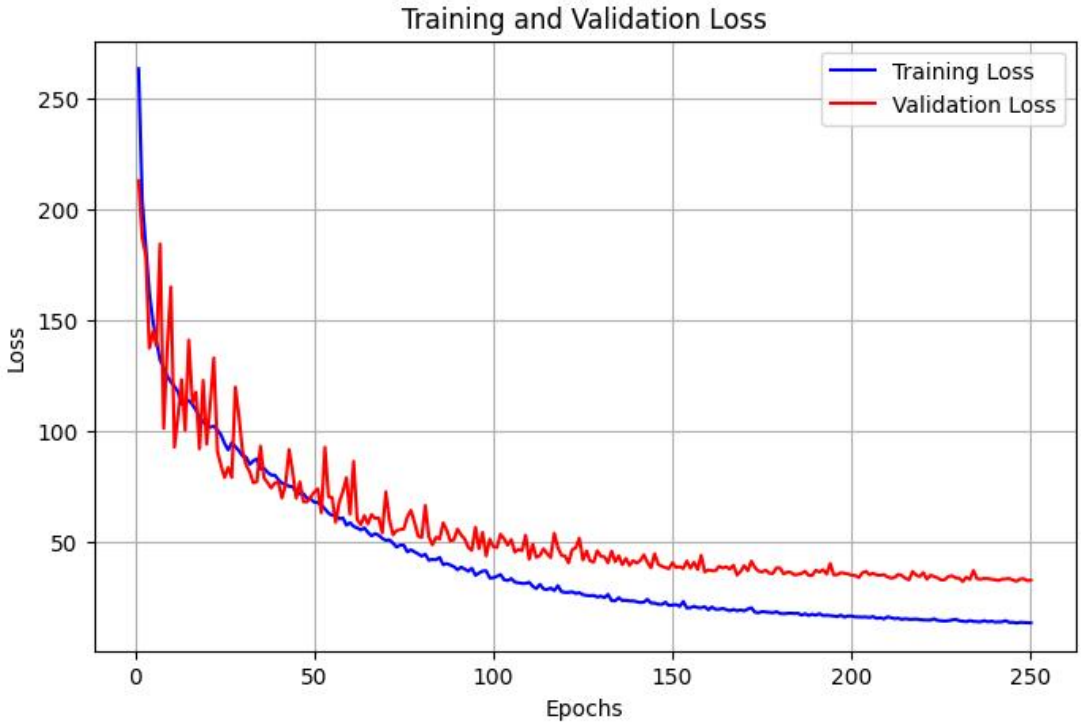


Figure 5.5: Train-Validation Loss of final-model-4 with N=3 (HEpose)

below, where $\hat{y}$ is the estimated 3D pose and $y$ is the ground truth 3D pose for a sample.

$$Loss = avg\left(\sum_{i=1}^{N}\sum_{j=1}^{K}|\hat{y_{ij}} - y_{ij}|\right) \tag{5.1}$$

Here, $K$ is the number of keypoints in a sample and $N$ is the total number of samples are being used for calculating the loss. After the completion of the training, using the test dataset (8k) kept for analysing the model's performance was evaluated with respect to MPJPE and PCK. Thereafter, the original test dataset by Zhu et al. (2023) was also used for analysing the model's performance.

### 5.4.1 Mean Per Joint Position Error (MPJPE)

The MPJPE is calculated by taking the mean of the Euclidean distance between the predicted 3D joint positions and the corresponding ground truth joint positions. Lower the value of this metric is, better the model is. MPJPE of a sample is given by the following equation:

$$MPJPE = \frac{1}{N}(\sum_{i=1}^{N} distance(y\_pred_i, y\_gt_i)) \tag{5.2}$$

Here, $N$ is the number of keypoints presented in each sample, $y\_pred_i$ is the estimated, and $y\_gt_i$ is the ground truth 3D pose for joint $i$ respectively. The $distance()$ function in equation-5.2, calculates the Euclid's distance in 3D. MPJPE is calculated based on six criteria. Four of the error values were measured by aligning each joint position along pelvis: full body considering 133 keypoints, only the body skeleton considering 23 keypoints (17 keypoints for body and 6 keypoints for leg), 68 keypoints of face and 42 keypionts of hand. Rest two error values were measured aligning along nose and wrist which is face and hand respectively. For left and right hand, the alignment was done along left wrist and right wrist respectively.

### 5.4.2 Percentage of Correct Keypoints (PCK)

The PCK is measured considering a threshold value, if the distance between predicted position in 3D and the ground truth 3D position is less than the threshold value then the keypoint is treated as correctly estimated. Thus correctly identified keypoints of $N$ samples are counted and reported as percentage. Higher value of this metrics represents the betterment of the model. In Algorithm-5.1, the process of calculating PCK is given.

**Algorithm 5.1** Pesudo algorithm for calculating percentage of correct keypoints (PCK) for a specific Threshold value

**Require:** $num\_joints, Threshold, y\_gt, y\_pred$

$x \leftarrow Threshold$
$N \leftarrow length(y\_pred)$
$total\_count \leftarrow 0$
**for** $i \in N$ **do**
    $count \leftarrow 0$
    **for** $j \in num\_joints$ **do**
        **if** $distance(y\_pred[i,j], y\_gt[i,j]) \leq x$ **then**
            $count \leftarrow count + 1$
        **end if**
    **end for**
    $total\_count \leftarrow total\_count + count/num\_joints$
**end for**
$PCK = \dfrac{total\_count}{N} \times 100\%$

We have considered the threshold value in two criterion to check the performance. One is fixed distance in millimeter which varied from 50 mm to 500 mm and another one is considered as the 50% of the torso length written as PCK@0.5T. Torso length is the distance between top of the shoulder to the top of hip bone.

## 5.5 Evaluation Result and Analysis

We will report the results obtained from each of the model. Results of the model's performance will be analysed. We have analysed the model-1, model-2, hybrid-model-3 and final-model-4. We have analysed the result using two test dataset. Eight thousand samples which were kept from the training dataset and the ten thousand samples which were

provided by H3WB dataset. Moreover we have also analysed the result of hybrid-model-3 with the increment of the value of N and changing of optimizer with the final-model-4 i.e. HEpose. All the analysis is done based on two evaluation metrics: MPJPE and PCK.

### 5.5.1 Test Result Analysis Based on MPJPE

To compare and analyse the results obtained from the networks are compared and analysed. MPJPE is shown in Table-5.1 for model-1, model-2, hybrid-model-3 with N=3 and the HEpose using Adam optimizer which represents MPJPE in six criteria. Table-5.1 showed that the MPJPE error got reduced by almost 50% in hybrid-model-3 from model-1. Also, it is noticed that the MPJPE got reduces if the joint connection information is incorporated. All the errors for all six criteria is reduced by the HEpose model except the pelvis aligned MPJPE on hand. The results of Table-5.1 were measured using the 8K samples of train dataset which were segregated from the training.

Table 5.1: Evaluation result of MPJPE (in milimeter) of model-1, model-2 with 3 Residual Block, hybrid-model-3 with N=3 and the proposed final-model-4 (HEpose)

| | **MPJPE (mm)** | **model-1** | **model-2, Residual=3** | **hybrid-model-3, N=3** | **HEpose** |
|---|---|---|---|---|---|
| 1 | Pelvis aligned MPJPE (133) | 88.9742 | 51.7882 | 43.8661 | **42.2171** |
| 2 | Pelvis aligned MPJPE on body (23) | 86.0861 | 52.9780 | 44.7417 | **41.5647** |
| 3 | Pelvis aligned MPJPE on face (68) | 68.7416 | 44.2199 | 36.3528 | **34.1577** |
| 4 | Nose aligned MPJPE on face (68) | 19.4384 | 14.7677 | 13.8340 | **12.2497** |
| 5 | Pelvis aligned MPJPE on hands (42) | 123.3132 | 63.3903 | 55.5509 | 55.6231 |
| 6 | Wrist aligned MPJPE on hands (42) | 45.4498 | 34.1689 | 28.5183 | **27.1080** |

Value inside the first bracket represents the number of keypoints considered for evaluating MPJPE on the segregated 8k data samples.

For further analysis, the value of N was increased from 3 to 6, 9 and 12. Figure-5.6 showed that the MPJPE errors were increased for N=12 with Adam optimizer, which got reduced significantly when Adamax optimizer was incorporated. While incorporating the network of pose grammar in final-model-4 (HEpose) out-performed all the models. It is also noticeable that for N=12 hybrid-model-3 deteriorate from its regular behavior. It happened due

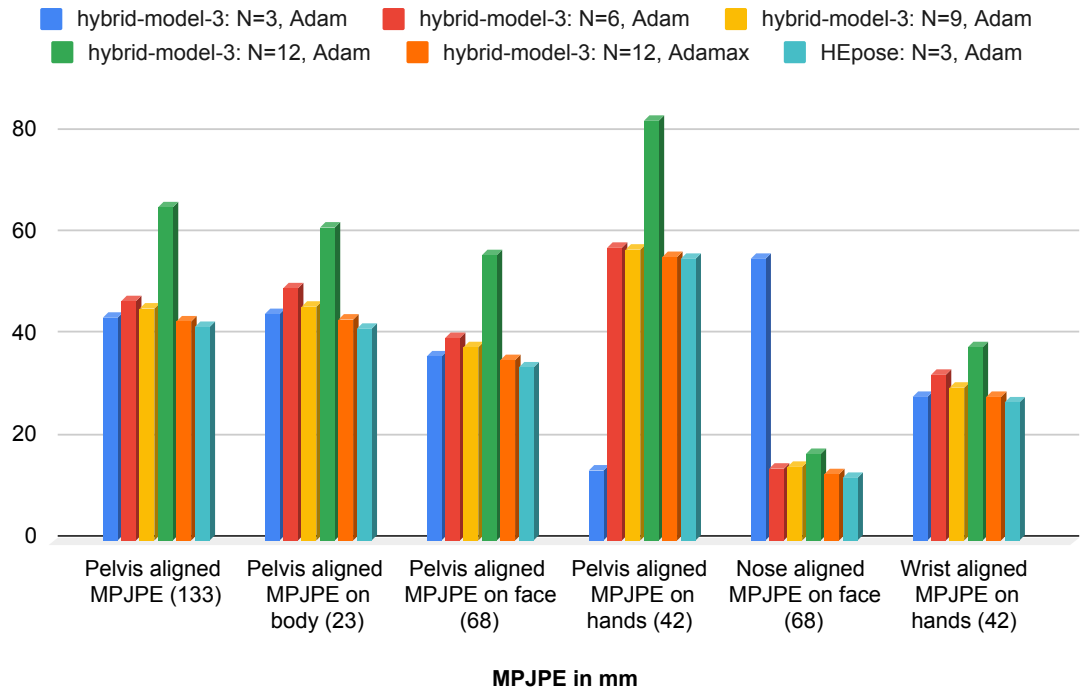


Figure 5.6: Comparative result of MPJPE while increasing the value of residual block i.e N=6 to N=12 with Adam and only N=12 with Adamax optimizer of hybrid-model-3, and HEpose.

the increased residual blocks the model was being overfitted with the training dataset.

### 5.5.2 Test Result Analysis Based on PCK

The PCK of model-1, model-2, hybrid-model-3 and final-model-4 is shown in Table-5.2. Here, all the models were built considering N=3 for the residual blocks. It represent PCK with fixed value threshold as well as the 50% of the torso length. At threshold PCK@0.5T, HEpose outperformed model-1, model-2 and hybrid-model-3, and also behaved similarly at other fixed threshold levels. PCK@150mm and PCK@0.5Torso is highlighted in the table, because PCK at these two values are considered as the standard threshold by most of the researchers in this domain. And we can see that, PCK@150mm and PCK@0.5T is in acceptable region which is 89.97% and 93.93% respectively.

Figure-5.7 depicts the comparison among the models while value of N is being varying in hybrid-model-3 and final-model-4. The hybrid-model-3's estimation of 3D human pose improved noticeably as N was increased. It is also noteworthy that, as the residual block

Table 5.2: Evaluation result of Percentage of Correct Keypoint (PCK) of model-1, model-2, hybrid-model-3 and final-model-4 i.e HEpose

| **PCK (mm)** | **Model-1** | **Model-2** | **Hybrid-model-3: N=3** | **Final-model-4: HEpose** |
|---|---|---|---|---|
| PCK@50 mm | 11.1055 | 62.3753 | 64.0079 | **66.6884** |
| PCK@100 mm | 43.9461 | 82.7347 | 83.1899 | **84.1211** |
| **PCK@150 mm** | 70.0602 | 89.372 | 89.5001 | **89.9760** |
| PCK@250 mm | 90.8028 | 94.0414 | 94.3095 | **94.5006** |
| PCK@300 mm | 94.5209 | 95.182 | 95.4108 | **95.6835** |
| PCK@350 mm | 96.5704 | 96.0956 | 96.1553 | **96.4479** |
| PCK@400 mm | 97.7262 | 96.6873 | 96.6986 | **97.0543** |
| PCK@450 mm | 98.4982 | 97.202 | 97.1691 | **97.4853** |
| PCK@500 mm | 98.9521 | 97.6234 | 97.6658 | **97.9317** |
| **PCK@0.5Torso mm** | 88.5626 | 93.5306 | 93.759 | **93.9325** |

increases, though the rate of performance improved in terms of PCK, but the improvement rate was relatively low. Furthermore, the Figure-5.7 depicts that for N=3 the model has performed comparatively better at PCK@50 and so on. Which means that for N=3 the model is generalizing and stabilizing as well as using less trainable parameters as residual blocks were less. So, the best performing model with respect to N is having the value of N=3. Thus, we selected that the value of N to be set at 3 for the proposed architecture. Moreover, while the pose grammar is being incorporated (HEpose), then the performance also increases compared to all other models.

## 5.6 Ablation Study of HEpose

In this section, behaviour of final-model-4 (HEpose) architecture will be discussed, if different module of it got removed. Table-5.3 and 5.4 gives the summary result of the ablation study.

In Table-5.3, we had analysed the HEpose considering MPJPE. At first we had removed the model-1 module from the HEpose. Then we had trained the model with 64k samples using our selected train dataset. Then, evaluation result is generated using the 10k test dataset provided by Zhu et al. (2023). The evaluation result showed that, without model-1 the performance of HEpose architecture fall down which means model-1 is playing very important

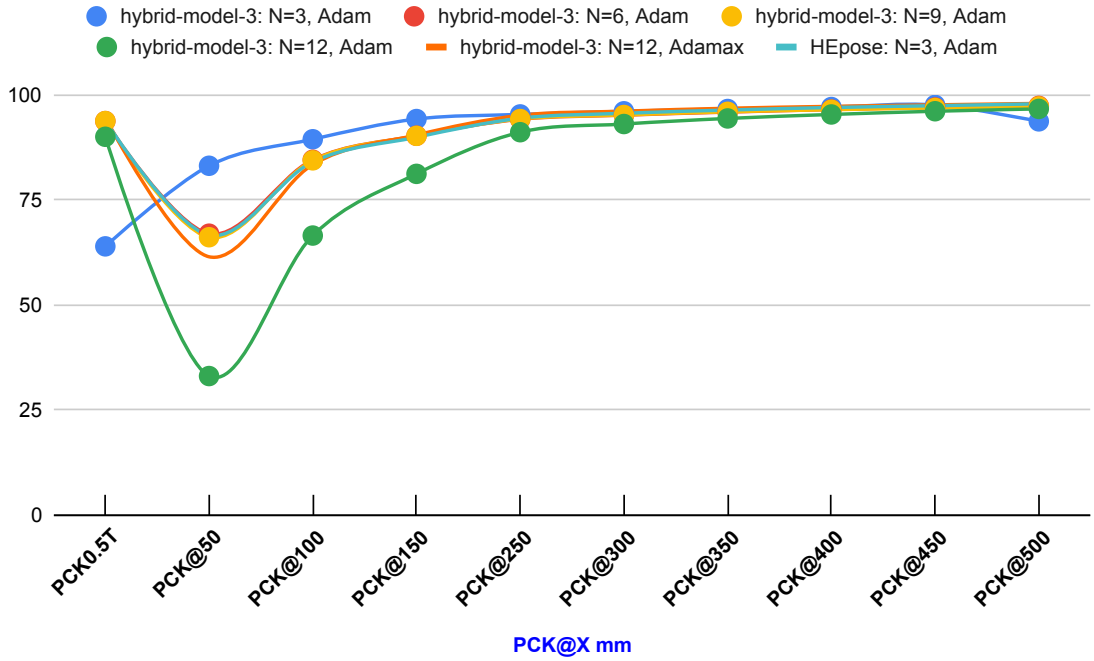


Figure 5.7: Comparative results of Percentage of Correct Keypoint (PCK) among the variations of Hybrid model-3 and Final model-4. Value of N is varied from 3 to 12 in hybrid-model-3.

role in this architecture. The results are projected in the first column of the Table-5.3. Next we had removed the model-2 portion and trained with 64k samples and evaluated using the same dataset. From these evaluation result we found that, for three criterion of MPJPE showed improved performance with respect to HEpose while other three criterion was not improved. Without model-2 compared to the full HEpose which has three module or part: model-1, model-2 and pose network model, HEpose is giving better performance on the pelvis aligned 23 joints of body, and nose aligned 68 joints of face. The reason for this is, the variant of pose grammar we had consider comprises most of the joint relations of the body and face. So, this ablation study gave us the intuition to focus on building the relation among the joints more precisely which we would like to carry on as our future work.

In Table-5.4, Procrustes aligned on MPJPE in millimeter is compared. The meaning of procrustes aligned is to super-impose the a predicted shape on the original shape by rotating, translating or shifting and after that calculating the deviation. In our case, the predicted 3D poses were super-imposed considering the original ground truth 3D poses and thereafter the MPJPE were calculated and reported. To do this task, author Zhu et al. (2023) helped as the

Table 5.3: Performance Evaluation of HEpose with and without incorporating Model-1 and Model-2 in terms of MPJPE

| **MPJPE (mm)** | **w/o Model-1** | **w/o Model-2** | **HEpose** |
|---|---|---|---|
| Pelvis aligned MPJPE (133) | 624.0769 | **88.4820** | 95.1904 |
| Pelvis aligned MPJPE on body (23) | 613.2315 | 86.2650 | **81.9182** |
| Pelvis aligned MPJPE on face (68) | 701.5047 | **67.9163** | 74.3716 |
| Nose aligned MPJPE on face (68) | 128.5384 | 17.6050 | **15.3120** |
| Pelvis aligned MPJPE on hands (42) | 504.6563 | **122.9929** | 136.1652 |
| This result is reported using the author provided test data i.e 10k samples for MPJPE | | | |

evaluation result given on the 10k author provided dataset. From Table-5.4, we had reported the performance of the model on 10k test dataset, if the model- 1 or model-2 is removed respectively. This also gave similar intuition like before that model-1 has significant role in the architecture as without model-1, the performance is deteriorating. Comparing with HEpose and without model-2 column in the table, we found that pose network improved the estimation of face and hand. Though other two i.e. body and all joints results did not improved, the difference is low. So we were again directed to find a more intuitive joint relation among the joints.

Table 5.4: Model-1 and Model-2 is excluded separately to analyse the performance of the final-model-4: HEpose considering the Procrustes aligned MPJPE on the 10K Test dataset of H3WB dataset. Zhu et al. (2023)

| **PA MPJPE (mm)** | **w/o Model-1** | **w/o Model-2** | **Final Model-4: HEpose** |
|---|---|---|---|
| PA MPJPE | 401.3654 | **54.0126** | 57.4395 |
| PA MPJPE on body | 586.5677 | **60.2012** | 63.9371 |
| PA MPJPE on face | 106.9215 | 11.7615 | **10.1550** |
| PA MPJPE on hand | 87.1282 | 14.6668 | **12.4413** |
| This result is reported using the author provided test data i.e 10k samples for PA MPJPE, Lower value indicates better results | | | |

## 5.7 Comparison with SOTA methods

In this section, comparative result of the proposed final-model-4 (HEpose) is shown with respect to the other state-of-the-art methods. Here we had compared our proposed model with SimpleBaseLine by Martinez et al. (2017), SMPL-X by Pavlakos et al. (2019), CanonPose by Wandt et al. (2021), and Jointformer by Lutz et al. (2022). SimpleBaseLine considers

linear layered base architecture like model-1. unified model, called SMPL-X is a unified model for SMPL eXpressive, with shape parameters trained jointly for the face, hands and body. CannonPose comprises of 2D detectors as it takes image as input, a lifting network using linear layer and residual block, and can support multiple view. Jointformer comprise of Transformer based architecture for joints prediction and another refinement transformer for boosting the predictions.

Table 5.5: Comparative analysis of the proposed final model (HEpose) with existing SOTA methods in terms of MPJPE

| **MPJPE (mm)** | **Martinez et al. (2017)** | **Pavlakos et al. (2019)** | **Wandt et al. (2021)** | **Lutz et al. (2022)** | **Ours: HEpose** |
|---|---|---|---|---|---|
| Pelvis aligned MPJPE (133) | 125.4 | 188.9 | 186.7 | **88.3** | 95.2 |
| Pelvis aligned MPJPE on body (23) | 125.7 | 166.0 | 193.7 | 84.9 | **81.9** |
| Pelvis aligned MPJPE on face (68) | 115.9 | 208.3 | 188.4 | **66.5** | 74.4 |
| Nose aligned MPJPE on face (68) | 24.6 | 23.7 | 24.6 | 17.8 | **15.3** |
| Pelvis aligned MPJPE on hands (42) | 140.7 | 170.2 | 180.2 | **125.3** | 136.2 |
| Wrist aligned MPJPE on hands (42) | 42.5 | 44.4 | 48.9 | 43.7 | **34.7** |
| Value inside the first bracket represents the number of keypoints considered for evaluating MPJPE on 10k samples of Test Dataset provided by Zhu et al. (2023). | | | | | |

We found that, our proposed architecture out-performed SimpleBaseLine, SMPL-X and CanonPose in terms of MPJPE for all the six criteria. HEpose also out-perfomed in three criteria compared with Jointformer architecture. And it is also noticeable that, though HEpose lags behind in other three criteria of MPJPE, the difference with Jointformer is very less.

# CHAPTER 6
# CONCLUSION

In this chapter, our focus is on reflecting upon the significant milestones we've attained and articulating our vision for the future. We explored the accomplishments that shaped our journey thus far and elaborated on the strategies we intended to pursue moving forward.

## 6.1 Thesis Outcomes

The thesis investigated deep neural network for finding human 3D pose from 2D pose of an image. As a outcome of the study, two architecture were proposed combining linear layer, residual connection as well as network for handling the pose grammar-based approach. The research achieved the following outcomes at the end of the research.

1. Efficient deep learning architecture was designed for human pose estimation considering skeleton based body representation.

2. A comprehensive and in-depth analysis was carried out and the final architecture (HEpose) outperformed state-of-the-art methods on the skeleton based body model representation for human pose estimation.

## 6.2 Thesis Contribution and Implications

Estimating Human 3D Pose from 2D Keypoints is one of the fundamental problem in computer vision in areas like motion capture, gesture recognition, human-computer-interaction etc. Deep Neural Network (DNN) architectures play a crucial role in estimating human 3D pose from 2D keypoints. By exploring various architectures, the most suitable architecture for estimating human 3D pose was figured out investigating different configurations of the architectures, layer types, and connectivity patters etc. A parallel network architecture

is designed that allows for parallel processing of information and potentially improving performance with SOTA methods. The architectures' parallel branches helps to operate concurrently on different aspects of the input data. A specific dataset was explored in depth and used for training and evaluating the model's performance. The choice of dataset is crucial as it determines the diversity and complexity of the data the model learns from, thereby influencing its generalization performance. In this study, H3WB dataset is thoroughly explored and examined on the designed architecture. For measuring performance of the designed architecture, we found Mean Per Joint Position Error (MPJPE) and Percentage of Correct Keypoints (PCK) is essential for quantitatively assessing its effectiveness. These metrics provide insights into how accurately the model estimated 3D poses compared to ground truth data. We explored and showed through the result analysis that combining multiple models lead to improved performance by leveraging the strengths of each individual model. We had also focused the effect of residual blocks. Increasing the number of residual blocks enhanced the model's capacity to capture complex relationships in the data, potentially leading to better performance. But we also showed that it leads to overfit the model with the training dataset, that's why it is important to determine the optimal number of residual blocks for this task. From our study we identified that three residual block performed best for generalizing the model's performance. Later on, joint connection information were incorporation which improved its ability to capture the spatial relationships between different body joints. This contextual information helped the model produce more accurate and coherent 3D pose estimations. Thus this research, not only explored novel approaches to find human pose estimation but also demonstrated improvements in performance through careful architectural design and model combination strategies. This study will have an impact on the society to accurately take decisions where human 3D pose is required.

## 6.3 Limitation and Future work

In this section, we will describe on the limitations and future directions on this research domain. Though our proposed hybrid model and final model (HEpose) resulted better performance, there were some limitations. We have only used one dataset H3WB which is provided by Zhu et al. (2023) for the task 2D to 3D pose estimation and the dataset had complete body in the 2D pose which means there were no occlusion. And we had worked on from the 2D pose assuming that 2D pose will be estimated using state-of-the-art 2D pose detectors. We did not work on estimating 3D poses from image or video. Moreover, we did not considered the presence of multiple human in the image. So, to use this proposed architecture, one need to detect multiple person using some state-of-the-art person detector, then use state-of-the-art 2D pose estimator to generate 2D pose of 133 keypoints of each person and then the proposed architecture can estimate 3D pose of each of the person.

The future directions of the study is to identify the real-time performance of the architecture so that it can be integrated in different action recognition or in augmented and virtual reality. Researchers can also focus on the self-supervising or reinforcement learning for estimation 3D pose in real-life scenario. More analysis can be conducted while directly estimating 3D pose from image or video frames. As the image quality and angle of the image have effect on the performance in terms of precision in identifying posture in three dimension. Researcher can also work on the real-time applications using 3D pose estimator to evaluate the performance of different action. For example, in sign language recognition, the application needs to understand the hand posture as well as movement to realise the meaning of the hand gesture. So, researching on the full body's each parts 3D pose estimation is crucial for improve the power of the computer vision and understanding scene.

# REFERENCES


Andriluka, M., L. Pishchulin, P. Gehler, and B. Schiele (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pp. 3686–3693.

Anguelov, D., P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis (2005). Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pp. 408–416.

Belagiannis, V., S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic (2014). 3d pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1669–1676.

Berclaz, J., F. Fleuret, E. Turetken, and P. Fua (2011). Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence 33*(9), 1806–1819.

Cai, Y., L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann (2019). Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2272–2281.

Chen, M., Z. Wei, Z. Huang, B. Ding, and Y. Li (2020). Simple and deep graph convolutional networks. In *International conference on machine learning*, pp. 1725–1735. PMLR.

Chen, W., H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen (2016). Synthesizing training images for boosting human 3d pose estimation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 479–488. IEEE.

Chen, Y., Y. Tian, and M. He (2020). Monocular human pose estimation: A survey of deep learning-based methods. *Computer vision and image understanding 192*, 102897.

Chu, H., J.-H. Lee, Y.-C. Lee, C.-H. Hsu, J.-D. Li, and C.-S. Chen (2021). Part-aware measurement for robust multi-view multi-human 3d pose estimation and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1472–1481.

Collins, R. T. (1996). A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE computer society conference on computer vision and pattern recognition*, pp. 358–363. Ieee.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fang, H.-S., Y. Xu, W. Wang, X. Liu, and S.-C. Zhu (2018). Learning pose grammar to encode human body configuration for 3d pose estimation. In *Proceedings of the AAAI conference on artificial intelligence*, Volume 32.

Garcia-Hernando, G., S. Yuan, S. Baek, and T.-K. Kim (2018). First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 409–419.

Gong, K., J. Zhang, and J. Feng (2021). Poseaug: A differentiable pose augmentation framework for 3d human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8575–8584.

Hasson, Y., G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid (2019). Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11807–11816.

Hossain, M. R. I. and J. J. Little (2018). Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 68–84.

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

Joo, H., T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, et al. (2017). Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence 41*(1), 190–204.

Kipf, T. N. and M. Welling (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kocabas, M., N. Athanasiou, and M. J. Black (2020). Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5253–5263.

Lin, J. and G. H. Lee (2021). Multi-view multi-person 3d pose estimation with plane sweep stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11886–11895.

Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer.

Liu, S., N. Sehgal, and S. Ostadabbas (2022). Adapted human pose: monocular 3d human pose estimation with zero real 3d pose data. *Applied Intelligence 52*(12), 14491–14506.

Loper, M., N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black (2023). Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 851–866.

Lutz, S., R. Blythman, K. Ghosal, M. Moynihan, C. Simms, and A. Smolic (2022). Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 1156–1163. IEEE.

Mahmood, N., N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black (2019, October). AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pp. 5442–5451.

Martinez, J., R. Hossain, J. Romero, and J. J. Little (2017). A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pp. 2640–2649.

Mehta, D., H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt (2017). Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 international conference on 3D vision (3DV)*, pp. 506–516. IEEE.

Mehta, D., O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt (2018). Single-shot multi-person 3d pose estimation from monocular rgb. In *2018 International Conference on 3D Vision (3DV)*, pp. 120–130. IEEE.

Mehta, D., S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt (2017). Vnect: Real-time 3d human pose estimation with a single rgb camera. *Acm transactions on graphics (tog) 36*(4), 1–14.

Mueller, F., F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt (2018). Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 49–59.

Pavlakos, G., V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black (2019). Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10975–10985.

Pavllo, D., C. Feichtenhofer, D. Grangier, and M. Auli (2019). 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7753–7762.

Ramakrishna, V., T. Kanade, and Y. Sheikh (2012). Reconstructing 3d human pose from 2d image landmarks. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*, pp. 573–586. Springer.

Redmon, J. and A. Farhadi (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271.

Sarafianos, N., B. Boteanu, B. Ionescu, and I. A. Kakadiaris (2016). 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding 152*, 1–20.

Schwarcz, S. and T. Pollard (2018). 3d human pose estimation from deep multi-view 2d pose. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2326–2331. IEEE.

Sigal, L., A. O. Balan, and M. J. Black (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision 87*(1-2), 4–27.

Simo-Serra, E., A. Quattoni, C. Torras, and F. Moreno-Noguer (2013). A joint model for 2d and 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3634–3641.

Sun, K., B. Xiao, D. Liu, and J. Wang (2019). Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5693–5703.

Varol, G., J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid (2017). Learning from synthetic humans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 109–117.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. *Advances in neural information processing systems 30*.

Von Marcard, T., R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll (2018, sep). Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*.

Wandt, B., M. Rudolph, P. Zell, H. Rhodin, and B. Rosenhahn (2021). Canonpose: Self-supervised monocular 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13294–13304.

Xu, Y., W. Wang, T. Liu, X. Liu, J. Xie, and S.-C. Zhu (2021). Monocular 3d pose estimation via pose grammar and data augmentation. *IEEE transactions on pattern analysis and machine intelligence 44*(10), 6327–6344.

Zhao, L., X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas (2019). Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3425–3435.

Zhao, W., W. Wang, and Y. Tian (2022). Graformer: Graph-oriented transformer for 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20438–20447.

Zhu, Y., N. Samet, and D. Picard (2023). H3wb: Human3. 6m 3d wholebody dataset and benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20166–20177.

# APPENDIX A

# IMPLEMENTATION

```
import torch
import json
import matplotlib.pyplot as plt
import numpy as np
import datetime
import os
import torch.nn as nn
from torch.utils.data import TensorDataset, DataLoader
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
KEYPOINTS = 133

# ================================================
# Following Codes for Loading Dataset
# ================================================
# Original Datapath
# path = 'C:\\Users\Zinia\PycharmProjects\wholebody3d\datasets\json'
input_list, target_list = json_loader(data_path, task=1, type='train')
X = torch.stack(input_list)
y = torch.stack(target_list)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_test, y_test,
   test_size=0.5, random_state=42)

# PyTorch DataLoader for easier iteration during training
train_dataset = TensorDataset(X_train, y_train)
val_dataset = TensorDataset(X_val, y_val)
test_dataset = TensorDataset(X_test, y_test)
```

```

train_loader = DataLoader(train_dataset, batch_size=64, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=64, shuffle=False)
test_loader = DataLoader(test_dataset, batch_size=64, shuffle=False)
```

Listing A.1: Python code for Loading the dataset

```
# Training for a single epoch
def train_one_epoch(epoch_index, training_loader, model, optimizer,
    loss_fn):
    running_loss = 0.
    last_loss = 0.
    number_of_batch = 0
    for i, data in enumerate(training_loader):
        # Every data instance is an input + label pair
        inputs, labels = data
        inputs, labels = inputs.to(device), labels.to(device)
        # Zeroing gradients for every batch!
        optimizer.zero_grad()
        # Make predictions for this batch
        outputs = model(inputs)
        # Compute the loss and its gradients
        outputs = outputs.view(-1, 1, KEYPOINTS, 3)
        loss = loss_fn(outputs, labels)
        loss.backward()
        # Adjust learning weights
        optimizer.step()
        # Gather data and report
        running_loss += loss.item()
        number_of_batch += 1
    last_loss = running_loss / number_of_batch
    return last_loss

def Model_Train(EPOCH, train_loader, val_loader, model, optimizer,
    loss_function, path_of_checkpoint, ck_name):
    training_losses = []
```

```
    validation_losses = []

    # Training loop
    num_epochs = EPOCH
    for epoch in range(num_epochs):
        model.train()
        train_loss = train_one_epoch(epoch, train_loader, model, optimizer, criterion)
        training_losses.append(train_loss)

        model.eval()
        running_val_loss = 0.0
        # using validation set during train
        with torch.no_grad():
            for i, data in enumerate(val_loader, 0):
                inputs, labels = data
                inputs, labels = inputs.to(device), labels.to(device)
                labels = labels.view(-1, 1 * KEYPOINTS * 3)
                outputs = model(inputs)
                loss = criterion(outputs, labels)
                running_val_loss += loss.item()
        val_loss = running_val_loss / len(val_loader)
        validation_losses.append(val_loss)
        print(f"Epoch [{epoch + 1}/{num_epochs}] - Train Loss: {train_loss:.3f}, Val Loss: {val_loss:.3f}")

    torch.save(model.state_dict(), f'{path_of_checkpoint}/{ck_name}_{epoch+1}_final.pth')
    torch.save(model, f'{path_of_checkpoint}/full_{ck_name}_{epoch+1}_final.pth')
    return training_losses, validation_losses
```

Listing A.2: Python code for Training the Model

```
import datetime
import os
```

```python
import torch
import torch.nn as nn
from torch.utils.data import TensorDataset, DataLoader
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt


KEYPOINTS = 133
#-------------------------------------
# Model-1
#-------------------------------------
class Model_Arch_simpleBaseline(nn.Module):
    def __init__(self):
        super(Model_Arch_simpleBaseline, self).__init__()
        self.upscale = nn.Linear(KEYPOINTS * 2, 1024)
        self.fc1 = nn.Linear(1024, 1024)
        self.bn1 = nn.BatchNorm1d(1024)
        self.fc2 = nn.Linear(1024, 1024)
        self.bn2 = nn.BatchNorm1d(1024)
        self.fc3 = nn.Linear(1024, 1024)
        self.bn3 = nn.BatchNorm1d(1024)
        self.fc4 = nn.Linear(1024, 1024)
        self.bn4 = nn.BatchNorm1d(1024)
        self.outputlayer = nn.Linear(1024, KEYPOINTS * 3)

    def forward(self, x):
        x = x.view(-1, 1 * KEYPOINTS * 2)
        x = self.upscale(x)
        x1 = nn.Dropout(p=0.5)(nn.ReLU()(self.bn1(self.fc1(x))))
        x1 = nn.Dropout(p=0.5)(nn.ReLU()(self.bn2(self.fc2(x1))))
        x = x + x1
        x1 = nn.Dropout(p=0.5)(nn.ReLU()(self.bn3(self.fc3(x))))
        x1 = nn.Dropout(p=0.5)(nn.ReLU()(self.bn4(self.fc4(x1))))
        x = x + x1
        x = self.outputlayer(x)
```

```
        x = x.view(-1, 1 * KEYPOINTS * 3)
        return x

#-----------------------------------
# Model-2
#-----------------------------------
# Model-2 (Archi-2)
class lifter_res_block(nn.Module):
    def __init__(self, hidden=1024):
        super(lifter_res_block, self).__init__()
        self.l1 = nn.Linear(hidden, hidden)
        self.l2 = nn.Linear(hidden, hidden)

    def forward(self, x):
        inp = x
        x = nn.LeakyReLU()(self.l1(x))
        x = nn.LeakyReLU()(self.l2(x))
        x += inp
        return x

class model_LargeSimpleBaseline(nn.Module):
    def __init__(self, input_fz=133*2, output_fz=133*3):
        super(model_LargeSimpleBaseline, self).__init__()
        self.upscale = nn.Linear(input_fz, 1024)
        self.res_1 = lifter_res_block()
        self.res_2 = lifter_res_block()
        self.res_3 = lifter_res_block()
        self.pose3d = nn.Linear(1024, output_fz)

    def forward(self, p2d):
        x = p2d.view(-1, 1 * 133 * 2)
        x = self.upscale(x)
        x = nn.LeakyReLU()(self.res_1(x))
        x = nn.LeakyReLU()(self.res_2(x))
        x = nn.LeakyReLU()(self.res_3(x))
```

```
        x = self.pose3d(x)
        x = x.view(-1, 1 * 133 * 3)
        return x


#-------------------------------------
# Hybride-Model-3
#-------------------------------------
class ModelCombined(nn.Module):
    def __init__(self):
        super(ModelCombined, self).__init__()
        self.model_1 = Model_Arch_simpleBaseline()
        self.model_2 = model_LargeSimpleBaseline()
        self.fc_last = nn.Linear(2*133*3, 133*3)

    def forward(self, x):
        output1 = self.model_1(x)
        output2 = self.model_2(x)
        concat = torch.cat((output1, output2), dim=1)
        # print(x.shape)
        x = self.fc_last(concat)
        return x
```

Listing A.3: Implementation code of Model-1, Model-2 and Hyrid-Model-3 architecture

```
#----------------------------------------
# Building Model of HEpose (Final-Model-4)
#----------------------------------------

import torch
import torch.nn as nn
import torch.nn.functional as F


def create_adjacency_matrix(num_keypoints, connections):
    adjacency_matrix = torch.zeros((1, num_keypoints, num_keypoints),
```

```
    dtype=torch.float32)
    for connection in connections:
        keypoint1, keypoint2 = connection
        adjacency_matrix[:, keypoint1, keypoint2] = 1
        adjacency_matrix[:, keypoint2, keypoint1] = 1  # undirected
    connections
    return adjacency_matrix


KEYPOINTS = 133


class GraphConvolution(nn.Module):
    def __init__(self, in_channels, out_channels):
        super(GraphConvolution, self).__init__()
        self.linear = nn.Linear(in_channels, out_channels)

    def forward(self, x, adjacency_matrix):
        x = x.view(-1, 133,  x.shape[-1])
        x = torch.matmul(adjacency_matrix, x)
        x = x.view(-1, x.shape[-1])
        x = self.linear(x)
        return x


class PoseGrammarNetwork(nn.Module):
    def __init__(self, num_keypoints):
        super(PoseGrammarNetwork, self).__init__()
        self.gc1 = GraphConvolution(2, 64)
        self.gc2 = GraphConvolution(64, 128)
        self.gc3 = GraphConvolution(128, 256)
        self.fc_keypoints = nn.Linear(256, 3)

    def forward(self, x, adjacency_matrix):
        x = F.relu(self.gc1(x, adjacency_matrix))
```

```
        x = F.relu(self.gc2(x, adjacency_matrix))
        x = F.relu(self.gc3(x, adjacency_matrix))
        x = x.view(-1, 256)
        keypoints = self.fc_keypoints(x)
        keypoints = keypoints.view(-1, 399)
        return keypoints


class Model1_Model2_PoseGrammar(nn.Module):
    def __init__(self, keypoints):
        super(Model1_Model2_PoseGrammar, self).__init__()
        self.model_pose = PoseGrammarNetwork(keypoints)
        self.model_1 = Model_Arch_simpleBaseline()
        self.model_2 = model_LargeSimpleBaseline()
        self.fc_last = nn.Linear(3*133*3, 133*3)

    def forward(self, x, connection):
        output1 = self.model_1(x)
        output2 = self.model_2(x)
        output3 = self.model_pose(x, connection)
        concat = torch.cat((output1, output2, output3), dim=1)
        x = self.fc_last(concat)
        return x

#-----------------------------------------------------------------
# Joint Connection Buiding for the variation of the pose grammar
#-----------------------------------------------------------------
num_keypoints = 133
list_branch_head = [(0,1),(1,3),(0,2),(2,4), (59,64), (65,70),(71, 82),
    (71,83),(77,87),(77,88),(88,89),(89,90),(71,90)]
list_branch_left_arm = [(5,7),(7,9),(9,91),(91,92),(93,96),(96,100)
    ,(100,104),(104,108),(91,108)]
list_branch_right_arm = [(6,8),(8,10),(10,112),(112,113),(114,117)
    ,(117,121),(121,125),(125,129),(112,129)]
list_branch_body = [(5,6),(6,12),(11,12),(5,11)]
```

```
77 list_branch_right_foot = [(12,14),(14,16),(16,20),(16,21),(16,22)]
78 list_branch_left_foot = [(11,13),(13,15),(15,17),(15,18),(15,19)]
79
80 connections = list_branch_head + list_branch_left_arm +
       list_branch_right_arm + list_branch_body + list_branch_right_foot +
       list_branch_left_foot
81
82 adjacency_matrix = create_adjacency_matrix(num_keypoints, connections).
       to(device)
83
84 #----------------------------------------------------------------
85 # HEpose (Final-Model-4): Model1_Model2_PoseGrammar
86 # Traing the HEpose
87 #----------------------------------------------------------------
88 model_pose_hybrid = Model1_Model2_PoseGrammar(num_keypoints).to(device)
89 criterion = torch.nn.L1Loss()
90 optimizer = torch.optim.Adam(model_pose_hybrid.parameters(), lr=0.0001)
91 epoch=250
92 training_losses, validation_losses = Model_Train(epoch, train_loader,
       val_loader, model_pose_hybrid, adjacency_matrix, optimizer, criterion
       , path_of_checkpoint, 'model_pose_hybrid')
```

Listing A.4: Implementation of the Final-Model-4 i.e. HEpose