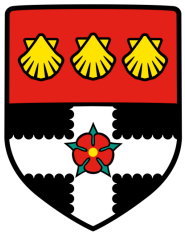


University of Reading

Department of Computer Science

Enhanced Sports Coaching with Deep Learning based 3D Human Pose Estimation and Time Series Comparison Techniques

Raghhuveer Jaikanth

Supervisor: Dr. Luis Patino

A report submitted in partial fulfilment of the requirements of
the University of Reading for the degree of
Bachelor of Science in *Computer Science*

September 22, 2023

Declaration

I, Raghhuveer Jaikanth, of the Department of Computer Science, University of Reading, confirm that this is my own work, and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Raghhuveer Jaikanth
September 22, 2023

Abstract

The recent Coronavirus pandemic and consequent lock downs across the world has disrupted teaching methods, with classes shifting from classrooms to online classes using calling services. This project presents an approach to utilise Human Pose Estimation (HPE), a classical branch of Computer Vision, to aid and enhance sports coaching. Advancements in highly performant hardware systems and the adjoining software have led to faster and more accurate algorithms for Human Pose Estimation. These advancements have facilitated researchers and engineers to employ Human Pose Estimation in various different scenarios, including Sports Coaching. This project explores the potential application of HPE for the training of soccer deadball techniques - Free kicks and penalty kicks. The project leverages two large datasets of sports-specific movements in the form of video data. We fine-tuned existing architectures on these datasets to cater to the dynamic and fast-paced nature of athletics. The evaluation on the dataset demonstrates that fine-tuning models provides precise and reliable body joint localisation even in fast-paced and cluttered environments. Additionally, two novel architectures are proposed and trained to elevate the 2D pose keypoints to the 3D space to remove dependency on the viewpoint of the camera. These keypoints are compared with a baseline video of the technique to score the users' technique. This is done by treating the samples as a time series and using methods for comparing time series data such as Dynamic Time Warping. Further, the potential challenges and limitations of HPE in sports coaching is discussed, offering insights into areas for further improvement and research.

Keywords: computer vision, human pose estimation, deep learning, soccer, sports coaching

Report's total word count: 12200 approx.

Acknowledgements

I would like to thank my supervisor Dr. Luis Patino for providing dedicated support and guidance. Dr. Patino encouraged me and was always willing and enthusiastic to assist in any way possible throughout the research project. I would also like to thank Dr. Lily Sun for her guidance and role in instilling a research oriented approach to projects in me. Furthermore, I would also thank the whole Computer Department for input and guidance throughout this MSc programmes. Finally, many thanks to my peers, especially Himanshi Virak, for pushing me to be the best, and my family who have always encouraged and supported me my entire life.

Contents

List of Figures

List of Tables

List of Abbreviations

SMPCS	School of Mathematical, Physical and Computational Sciences
HPE	Human Pose Estimation
CV	Computer Vision
AI	Artificial Intelligence
YOLO	You Only Look Once
IoU	Intersection over Union
CIoU	Comlete Intersection over Union
DFL	Dual Focal Loss
ViT	Visual Transformers
SGD	Stochastic Gradient Descent
ReLU	Rectified Linear Unit
SiLU	Sigmoid Linear Unit
SPP	Spatial Pyramid Pooling
CNN	Convolutional Neural Network
FFN	Feed Forward Network

Chapter 1

Introduction

Sports coaching involves a careful assessment of the pose and movements of the human body. Sports coaching is a huge market in today's world, poised to grow to around 111824 million USD [1] by 2027 making it a very lucrative research market. Correctly executing movements is crucial in sports to increase performance, minimise risk of injury and improve results.

Human Pose Estimation (HPE) is the classical Computer Vision (CV) task of detecting, classifying and, locating the joints of the human body from images. The purpose of Human Pose Estimation is to identify coordinates in an image known as key points or joints. A pair or joint is then formed by connecting two key points. Deep neural networks have led to a series of breakthroughs in the classical computer vision task of Human Pose Estimation. Open-source datasets and challenges have massively increased the availability of data required for training deep learning architectures, resulting in fast moving research in the field of Human Pose Estimation.

HPE can be used to aid and enhance sports coaching by outsourcing technique correction to AI models. HPE can be used to capture a sequence of poses representing the motion of a player executing a technique. This sequence can be compared with a referral sequence to calculate the correctness of the technique executed by the player.

1.1 Background

Historically, sports training and coaching has been done in person with the coach and the student being present in person. However, in the past 3 years, in-person training has been reduced due to the pandemic and global lockdown. This time period also saw a huge surge in the market of online education and teaching for theoretical fields like science and maths. However, there has not been many advancements in teaching physical activities such as sports online. This is mainly due to the human resource and financial constraint faced by schools and colleges for sports training and coaching. Educational institutions such as schools and colleges are not up to the task of providing quality sports coaching to each student due to human resource constraints especially in the aftermath of the global pandemic.

Traditionally, usage of computers in sports coaching using computers employ the use of body sensors to estimate the pose and movement of the human body. These sensors not only restrict user movement but are also not easily available and expensive for educational houses without financial backing. Recent advances in the field of deep learning and computer vision, particularly Human Pose Estimation using deep learning methodologies, can help solve this

shortage. Human Pose Estimation can be used to analyse body movements by identifying and tracking key points on the human body, generally joints. These advancements can be applied to the domain of sports coaching to aid and enhance the process using Artificial Intelligence (AI).

There is abundant research in the application of Human Pose Estimation using deep learning; however, most of this research is not domain specific and focused on generalising models over a wide range of human body movements. More specifically, there is very less research done on Human Pose Estimation application in the sports domain. This is further reflected in the amount of datasets for general human body movement versus human body movements in sports context. One of the major reasons for this gap in research is due to the difficulty of applying Human Pose Estimation in a sports context. This is because of difficulty in dataset collection, fast movements leading to motion blur and, frequent occlusions.

While there have been many studies to "lift" 2D key points to 3D key points using monocular images, a novel method is adopted for the same task in this project. Two neural network architectures are experimented with - a simple feed forward network and a 1D convolutional neural arranged in residual blocks. 2D poses are lifted to 3D space to ensure view-point invariance. Further, a scoring module to evaluate the correctness of a technique is implemented to aid the work of sports coaches. To evaluate the correctness of a sports technique, the key points for each frame are compared using a distance metric; however this assumes that the the two videos are of the same length and the frames are aligned in both the videos. By treating the sequences of key points as time series, conventional methods for comparing non aligned time series data can be applied.

1.2 Problem Statement

Current computer vision systems in sports coaching focuses on tactics and referee assistance for eg, Virtual Assistant Referee (VAR) in soccer. There is a lack of research in technique correction using vision systems. There is even less research in such systems without the use of body sensors to capture the movement of the player. This project aims to reduce this gap by implementing a system to score the correctness of soccer deadball techniques.

1.3 Aims and objectives

Aim

The key aim of the research is to investigate Human Pose Estimation methods, train them on sports datasets, and use them in real time. The project will attempt to create an application that can estimate the pose of the user using deep learning and comparing it with a reference video for the techniques of penalty kicks and free kicks in soccer.

Research Questions

As part of this dissertation, three Research Questions have been constructed and framed that will be answered:

- **RQ1:** What are the current state-of-the-art models available for Human Pose Estimation and what is their performance with respect to both runtime and accuracy?

- **RQ2:** How to lift 2D key points to 3D using monocular images?
- **RQ3:** What are the methodologies to compare two persons executing a technique in real time and provide feedback?

Objectives

As part of achieving the aim, several objectives have been set that were completed in a chronological order. These objectives are:

- Get a reference video with the correct technique through a domain expert.
- Compare and identify models that are accurate and lightweight.
- Train and develop a model on the chosen datasets and assess the models on run-time metrics and accuracy.
- Identify and implement a pose comparison technique for comparing the referral and user video.
- Develop a prototype application to predict the correctness of the user technique.
- Evaluate the strengths and weaknesses of the developed model, identifying areas of improvement.

1.4 Solution approach

The proposed solution is based on the YOLOv8 [2] pose detection algorithm. The model is fine tuned using the Australian Sports Pose dataset (ASPSet) [3] and the SportsPose dataset [4]. Two novel methods are proposed for lifting the key points from 2D space to 3D space joints, which are then compared using Dynamic Time Warping (DTW) method [5].

1.4.1 Dataset

The ASPset and SportsPose datasets are combined and used to fine tune the YOLOv8 model. 30 frames are extracted from every single video. For the ASPset, the intrinsic and extrinsic camera matrices are provided, which can be used to project 3D key points to the image space. Each frame is then cropped around the human body and resized to (640x640) which is the resolution suggested by YOLOv8 authors.

1.4.2 2D Pose Estimation

Two variants of the You Only Look Once Version 8 (YOLOv8) architecture, (A) YOLOv8n-pose (nano) and (B) YOLOv8x-pose (extra large), are chosen for predicting the 2D key points. The model is fine tuned with the combined dataset specified in Section 1.4.1. The Stochastic Gradient Descent (SGD) [6] and AdamW [7] algorithm is used for YOLOv8n and YOLOv8x respectively to optimise a combination of the Complete Intersection Over Union (CIoU) [8] and Dual Focus Loss (DFL) [9] functions for bounding box loss and, a novel v8PoseLoss [2] which uses Euclidean distances and BCEWithLogitsLoss to calculate the pose loss.

1.4.3 2D to 3D Pose Lifting

Two novel neural network architectures are proposed for lifting key points from 2D to 3D space. The first network uses a simple feedforward network with batch normalisation and Rectified Linear Units (ReLU) [10]. The second network is designed using 1D convolutional layers with batch normalisation and ReLU structured in a residual block. Both the networks are trained using keypoints from the SportsPose dataset. The SGD optimiser is used to minimise the L1 or the Mean Absolute Error (MAE) loss function. Finally, cosine similarity is used to measure the correctness of the predicted movements of the user.

1.4.4 Comparison and Scoring

There has been abundant research on comparing non aligned time series data; however these methods are primarily used in non-sport domains such as market research to identify similarities in stock trends. Inspired by these methods, Dynamic Time Warping (DTW) is used to score the similarity between twosequences and cosine similarity is used as the distance metric between two vectors/joints.

1.5 Summary of contributions and achievements

This project has achieved the following results -

- New combined dataset using ASPset and SportsPose created,
- YOLOv8N 2D Human Pose Estimation model performed with mAP@50 with OKS of 0.995,
- CNN 3D Pose Lifting performed with L1 Loss of 0.1099,
- Application with real-time performance of video comparison created enabling online sports coaching.

1.6 Organisation of the report

This reports is organised into seven chapters as described below.

Chapter 1: This chapter is partitioned into 6 sections. Section 1.1 outlines the background and motivation for completing this project. Section 1.2 and Section 1.3 describes the problem statement and the subsequent aim of the project respectively. Section 1.4 provides a brief overview of the implemented solution. Section 1.5 and Section 1.6 outline the significant results and the organisation of the report respectively.

Chapter 2: This chapter is divided into 6 sections. Section 2.1 dives into the existing datasets for 2D/3D human pose estimation datatsets. Section 2.2 describes in detail relevant existing methods for 2D human pose estimation. Section 2.3 outlines the existing methodologies for 3D pose lifting and Section 2.4 outlines time series comparison methods for comparing sequences of poses. Section 2.5 and 2.6 critiques and summaries the conducted literature review respectively.

Chapter 3: This chapter outlines the methodologies used as part of the project. Section 3.1 reiterates the problem statement while Section 3.2 outlines the processing techniques for

creating the datasets for 2D human pose estimation and 3D pose lifting. Section 3.3 outlines the architecture and training strategy for YOLOv8 2D human pose estimation model. Section 3.4 describes the architecture and training strategy for the novel proposed methods for 3D pose lifting. Section 3.5 outlines the methodology used for comparing two sequences of poses. Finally, Section 3.6 summarises the methodologies used.

Chapter 4: This chapter outlines the results of the project. This chapter is further divided in to 4 sections. Section 4.1 and Section 4.2 outline the results for the YOLOv8N and YOLOv8X models respectively. Section 4.3 and Section 4.4 outlines the results for the proposed CNN and FFN architectures for 3D pose lifting. Finally, the chapter is summarised in Section 4.5.

Chapter 5: This chapter discusses the significance of the findings in Section 5.1 and the potential limitations of the solution in Section 5.2.

Chapter 6: This chapter outlines the final conclusions of the project in Section 6.1 and the future work to be done in Section 6.2.

Chapter 7: This chapter reflects on the work conducted and describes the experiences faced during the course of this project.

Chapter 2

Literature Review

The literature review process was conducted in 4 steps. First, popular datasets for human pose estimation in a 2D, 3D and, sports context are reviewed to understand the current benchmarks in human pose estimation. Additionally, a brief review of the You Only Look Once (YOLO) framework is done to understand the improvements to the latest model based on the framework. Second, the current state of the art methods for deep learning based 2D pose estimation are reviewed to identify potential algorithms that can be used. Third, existing techniques for lifting pose from 2D to 3D space are assessed. Finally, methods to compare two sequences of poses are investigated.

2.1 HPE Datasets

2.1.1 2D HPE Datasets

Image Based Datasets

The Frames Labelled in Cinema (FLIC) dataset [11] is one of the earliest datasets for image based 2D human pose estimation. The data was collected by extracting the 10th frame of 30 popular movies totalling 5000 images. Amazon Mechanical Turk was used for crowd sourcing the annotation. Further, manual rejection of images based on pose orientation was carried out. The FLIC-Full dataset was released later containing the rejected images as well. The Max Planck Institut Informatik (MPII) Human Pose Dataset [12] is a comprehensive dataset containing images of people performing different activities including cycling, jumping etc. The images were collected from YouTube videos whose description matched the specific activity. The dataset contains a total of 24920 frames from all videos and 40522 different people. The frames are annotated manually in a person centric manner, including location of eyes and nose, head bounding box, visibility, 3D view points of head and torso and, occluded body parts.

The Microsoft Common Objects in Context (COCO) Dataset [13] is one of the most commonly used benchmarks in computer vision tasks such as object detection and human pose estimation. The benchmark contains 200,000 labelled images with 250,000 pose annotations. Numerous pose estimation publications use the COCO dataset as a benchmark for their algorithms. Additionally, the COCO pose format with 17 annotations is widely used among researchers as the standard for body keypoints. The CrowdedPose Dataset [14] was designed to benchmark HPE models in crowded scenarios. On average each image contains 4 persons with 80,000 pose annotations in 20,000 images. Data was collected by randomly sampling 20,000 images from three public benchmarks, the MPII, MSCOCO and AIChallenger [15], based on the crowd index of the image. The authors defined the crowd index based on the

Name	Image/Video	Year	Joints	Num Data samples
LSP[20]	Image	2010	14	2000 images
FLIC[11]	Image	2013	10	5000 images
MPII[12]	Image	2014	16	24920 images
MSCOCO[13]	Image	2017	17	200,000 images
CrowdedPose[14]	Image	2019	14	24920 images
Penn Action[16]	Video	2013	13	2326 videos
JHMDB[17]	Video	2013	15	600 videos
PoseTrack2018[18]	Video	2018	15	1138 videos/153,615 frames
HiEve[19]	Video	2020	14	31 videos/1M annotations

Table 2.1: 2D Human Pose Estimation datasets

number of joints not belonging to a bounding box present inside the bounding box.

Video Based Datasets

The Penn Action Dataset [16] covers 15 types of actions across 2326 video clips derived from YouTube. It is an unconstrained human action dataset annotated with 13 keypoints along with visibility score. The dataset is annotated using VATIC, an online video annotation tool for computer vision research that crowdsources work to Amazon's Mechanical Turk platform. A similar piece of work, the Joint Annotated Human Motion Database (JHMDB) Dataset [17] contains 21 action categories across 31838 frames. The JHMDB dataset is fully annotated with 15 keypoints per person and their respective visibilities.

PoseTrack Dataset [18] is a large public benchmark for human pose estimation and articulated keypoint tracking including challenging crowded environments with complicated movements with large number of occlussions. There are two variants of the PoseTrack benchmark, (A) PoseTrack2017 and, (B) PoseTrack2018. The former dataset is smaller with 514 videos and 16,219 pose annotations, while the latter dataset is significantly larger with 1,138 video clops and 153,615 pose annotations. The poses are annotated with 15 keypoints along with joint visibility. Human-Centric Video Analysis in Complex Events (HiEve) Dataset [19] is currently the largest video-based dataset for 2D human pose estimation containing 31 videos with 1,099,357 pose annotations. The dataset includes multiple tasks, including human pose estimation, pose tracking and, action recognition. Table 2.1 lists the datasets covered as part of this study.

2.1.2 3D HPE Datasets

Image Based Datasets

The MPI-INF-3DHP [21] dataset uses a commercial marker-less multi-view 14 camera system to capture 3D poses. This system allowed the use of everyday clothes, including loose clothes, thus enabling the capture of heavily occluded (from clothes) human joints. The dataset 1.3M frames covering 8 activity sets. The Total Capture [22] dataset provides both multi-viewpoint video, inertial measurement unit (IMU), and skeleton annotations obtained using VICON. Similar to MPI-INF-3DHP, the Total Capture dataset used a marker-less system to increase variability of the dataset. The dataset covers five actions over 1,892,176 frames captured by 8 different cameras.

Dataset	Environment	Year	Size
MPI-INF-3DHPE[21]	Indoor and Outdoor	2017	8 subjects, 8 actions, 1.3M frames
Total Capture[22]	Indoor	2017	5 subjects, 5 actions, 1.9M frames
HumanEva[23]	Indoor	2010	6 subjects, 7 actions, 40k frames
3DPW[26]	Outdoor	2010	60 video sequences
Human3.6M[24]	Indoor	2014	11 subjects, 17 actions, 3.6M frames
CMU-PanOptic[25]	Indoor	2016	8 subjects, 1.5M frames
ASPSet[3]	Outdoor	2021	17 subjects, 30 sports actions, 330K frames
SportsPose[4]	Indoor and Outdoor	2023	24 subjects, 5 sports actions, 1.5M frames

Table 2.2: 3D HPE datasets

Video Based Datasets

The HumanEva-I dataset [23] is a multi-view video dataset and uses a marker-less motion capture system to 3D poses. It covers 4 subjects performing 6 day-to-day actions over 40,000 frames. Human3.6M dataset [24] is one of the largest motion capture benchmarks consisting of 3.6M pose annotations and corresponding frames. The dataset contains 17 activities performed by 11 professional actors from 4 different camera views. The CMU Panoptic dataset [25] is one of the most accurate datasets captured using 480 VGA cameras, 31 synchronised HD cameras, and 10 RGB-D sensors for motion capture. The dataset contains 65 video sequences with 1.5 million 3D pose annotations capture large social interactions between the subjects. The 3D Poses in the Wild (3DPW) dataset [26] consists of 60 video sequences in the wild obtained using phone camera and information from IMUs. The dataset covers 18 actions in the wild such as walking in cities and, having coffee, and clothing styles. Table 2.2 lists the datasets covered as part of this study.

2.1.3 Sports HPE datasets

The Leeds Sports Pose (LSP) dataset [20] consists of 2,000 images of full body poses in a sports context. The images are obtained from flickr using respective keywords. The LSP dataset is extended as the LSP-Extended dataset which contains 10,000 images in a sports context. The images are annotated using 14 human body keypoints. The Australian Sports Pose Dataset (ASPset) employs three cameras and manual time synchronisation to capture and label 3D poses in a outdoor sports context [3]. The poses were captured by triangulating 2D keypoints from each one of the 3 cameras. The subjects in ASPset move with an average speed of 1.21 m/s making it suitable for fine-tuning human pose estimation algorithms in the sports domain. The SportsPose dataset is video-based, marker-less dataset captured using seven cameras with 24 subjects all wearing natural clothing in both indoor and outdoor settings. The dataset contains 1.5M frames captured using seven cameras covering five sports activities [4]. 3D poses are obtained by triangulating seven sets of 2D keypoints for each frame along with temporal information from the previous and succeeding frames.

2.2 2D Pose Estimation

2.2.1 Top Down Approach

Top Down Approach in Human Pose Estimation involves two steps - (1) Identifying and cropping human bounding boxes in the image and, (2) Using a Single Person Pose Estimator (SPPE) with the crop as input. SPPE(s) can be further parameterised as Regression based

or Heatmap based architectures.

Keypoint regression based Methods

Deep Pose [27] sets the precedent for human pose estimation with deep learning analagous to LeNet [28] for convolution based deep learning models. Deep Pose introduces an iterative architecture to extract features with cascaded convolutional neural networks. The joint coordinates are regressed directly using fully connected layers taking the features from the previous stage as input. Iterative Error Feedback (IEF) [29] suggests a self-correcting model, progressively changing the initial keypoints instead of directly predicting keypoints. By incorporating a top-down feedback structure, IEF is able to encompass rich structure in both the input and output spaces. Recent success of the transformer architecture in computer vision has led to the development of ViTPose [30], which uses a transformer network backbone to extract features. Using the learned features as input, simple decoders are used to predict the joint keypoints. While keypoint based architectures are highly efficient, they fail to consider the area of the body part around the keypoint. This issue can be solved by regressing heatmaps around the keypoint instead of deterministic coordinates.

Iterative Heatmap Based Methods

Pose Machines [31] use an iterative architecture gradually inferring the locations of joints in multiple stages. Convolutional Pose Machines [32] further extend Pose Machines with a sequential prediction framework employing sequential convolutions to model long-range dependencies between human body parts. Stacked Hourglass Networks [33] features a multi-stage architecture with repeated bottom-up, top-down processing and skip layer feature connections to predict heatmaps. The repeated pooling and upsampling ensures features across different scales are incorporated to help capture spatial relationships.[34] proposes a simple deconvolution network on top of a ResNet backbone to estimate the keypoint heatmaps. By using ResNet as a backbone, [34] is able to estimate heatmaps from both deep and low resolution feature maps. HRNet [35] is a representative network able to maintain high resolution representations. Pyramid Gating Networks [36] takes the HRNet as backbone network, and additionally employs gating mechanism and feature attention module for enhanced features. Heatmap based methods are popular due to their exceptional performance; however, the extra computational cost of heatmap computation introduces new challenges such as high computational overhead and inevitable quantisation error.

Separate human box and pose detection

The Top-Down approach can be further parameterised into algorithms performing proposal detection and pose detection jointly and, algorithms performing the tasks separately. RMPE (AlphaPose) [37] utilises the SSD-512 model as human detector and stacked hourglass as SPPE. RMPE identifies the effect of correct region proposals and propose the Symmetric Spatial Transformer Network (SSTN) that reduces the effect of incorrect region proposals. CrowdPose [14] aims to tackle the issue of people bounding boxes containing body parts from multiple people in crowded scenarios deteriorating the performance of the pose detector. CrowdPose propose a joint-candidate pose detector that predicts heatmaps with multiple peaks, and uses a graph network to perform global joints association. [38] further work on human pose estimation in crowded and occluded scenes and employ a Faster R-CNN based person detector to predict bounding boxes for human body candidates. Additionally, a novel keypoint estimator predicting keypoints using heatmap-offset aggregation is used.

Combined human box and pose detection

While separate models for person detection and keypoint regression works exceptionally in various scenarios, they have high computational overhead and scale poorly as the number of persons in the image increase. An alternate to separate models for person and keypoint detection is to perform bounding box detection and pose detection jointly. MultiPoseNet [39] first detects the bounding box and keypoints jointly; then a Pose Residual Network (residual multilayer perceptron) is used to assign the detected keypoints to the respective bounding box. FCPose [40] implements a pose estimation framework that uses dynamic instance-aware convolutions. This framework does not require post processing in the form of cropping bounding boxes and keypoint grouping.

2.2.2 Bottom Up Approach

The top down framework uses a object detector and pose detector to predict the human body keypoints. It is highly scalable and is constantly improved with continued research in object detection and single person pose detectors; however, there is a computational overhead from using two separate algorithms. In contrast, the bottom-up framework does not rely on human detection and directly regresses the human keypoints, thus improving runtime metrics. This introduces a new issue of keypoint assignment. Based on keypoint assignment, bottom up frameworks can be parameterised as, (A) Human-Center Regression Based, (B) Associate Embedding Based and, (C) Part Field Based approaches.

Human Center Regression

In the human center regression based approach, a center keypoint that can represent the human body is used. Single stage multi-pose machines [41] utilizse root joints, i.e. center biased points, to denote the person instances. The body joint locations are encoded as displacement from the root joints. Dis-Entangled Keypoint Regression (DEKR) [42] adopts adaptive convolutions through pixel wise spatital transformer to predict a heatmap that identifies the person instance, and densely estimates candidate pose at each pixel q within the center map. DEKR uses a multi-branch architecture, where each branch learns a representation with dedicated adaptive convolutions and regresses one keypoint, thus "disentangling" keypoints.

Associate Embedding

The associative embedding method works by assigning a embedding to each keypoint that associates it with a human instance. [43] pioneered this approach by teaching the network to simultaneously predict keypoints and group associations. [43] assign an embedding vector to each keypoint that can be used to identify the human instance the keypoint belongs to. HigherHRNet builds on [43] and uses high resolution feature pyramids to learn scale aware representations solving the scale variance challenge of human pose estimation. HigherHRNet uses feature maps from HRNet, upsampled using transpose convolutions [44]. [45] proposes a framework with two components, (A) SpatialNet for body part detection and part-level data association in a single frame, (B) TemporalNet to group and track human instances across consecutive frames in a trajectory. The part-level data association is represented by human body keypoint embedding.

Part affinity Fields

OpenPose [46] introduces a non-parametric representation of human body keypoints referred to as Part Affinity Fields (PAFs) that help learn body part association with individual human body instances in the image. OpenPose encodes global context used by a greedy algorithm

to group keypoints using connective intensity between joints allowing for an invariant runtime irrespective of the number of human body instances. PifPaf [47] builds on OpenPose and introduces Part Intensity Field (PIF) to localise body parts and, Part Association Field (PAF) to associate body parts to form full human poses. SimplePose [48] rethinks OpenPose and introduce (A) a new representation of body parts to encode information between keypoints, (B) An improved stacked hourglass network with attention, (C) a novel focal L2 loss for keypoint and keypoint association mining and, (D) Robust greedy keypoint assignment algorithm.

Bottom up methods improve the runtime efficiency of pose detection by removing the need for a separate human detection step. This efficiency has led to widespread adoption of bottom up methods, especially OpenPose, in realtime applications. However, the cost of grouping keypoints to human instances is still high and requires complex, extensive algorithm design.

2.2.3 You Only Look Once (YOLO)

You Only Look Once (YOLO) is a popular framework for various computer vision tasks initially designed for object detection tasks. YOLOv1 pioneered the single shot detector framework allowing for identifying region of interest and bounding boxes in a single pass. The input image is divided in to $S \times S$ grids for B boxes. Each grid predicted B bounding box along with the confidence score, ensuring a $S \times S \times B \times (5 + C)$ where C is the number of classes that an object can belong to. The model was trained using a combination of Classification Loss, Confidence Loss for objectness and, Localisation loss for bounding box coordinates. Post processing included Non-Maximum Suppression (NMS) to remove redundant predictions. The proposed YOLO model became popular due to its fast inference time, allowing for real-world applications; however, the model had a high error rate, mainly arising due to dependency on aspect ratio of the bounding box.

YOLOv2 [49] improved the YOLOv1 algorithm by utilizing a higher resolution input, adding batch normalisation and, making the network fully convolutional. Additionally, the YOLOv2 introduced multiple *priors*, i.e, potential bounding boxes for each grid cell. The priors are determined by clustering the ground truth bounding boxes to identify ideal size and aspect ratio. The model was trained using multiple scales, i.e, changing the size of grid cells, and predicts the bounding box relative the position of the grid cell. Similar to YOLOv1, each bounding box was defined by the confidence, coordinates and, object class. Each grid cell was represented by 5 priors, thus giving an 125 dimensional output for each grid cell.

YOLOv3 [50] further improved the framework by utlizing a new Darknet-53 backbone for richer and deeper features. The authors also introduced an objectness score for each bounding box which is defined as 1 for the anchor with the highest IoU with ground truth and 0 for all the other anchors. The framework also uses Binary Cross Entropy to allow for multi-label classification. Further, the Spatial Pyramid Pooling (SPP) block was added to the network to increase the receptive field of the model across multiple scales. The number of priors are reduced but scaled at 3 different scales to ensure better performance for small objects in the scene. Finally, a multi-scale prediction functionality is also added to the network inspired by a Feature Pyramid Network for better results. This architecture followed the design of an object detector having a backbone (feature collector), neck (feature aggregator) and, head (detection module).

YOLOv4 [51] introduces the concept of Bag-of-Freebies (BoF) and Bag-of-Specials (BoS). BoF methods change the training strategy and increases the training cost without affecting the inference runtime, for example - data augmentation. BoS methods increase the accuracy of the model, but increase the inference runtime as well at time same time. YOLOv4 uses data augmentation, focal loss, label smoothing, Complete Intersection over Union (CIoU) loss, Cross mini-Batch Normalisation (CmBF), Drop Block and Mosaic augmentations as BoF methods to improve performance without affecting runtime. Focal loss addresses the foreground and background imbalance by modifying the cross entropy loss. CIoU loss incorporates distance between the centers and aspect ratios of the bounding boxes to calulate a complete localisation error. Additionally, the model was trained with self-adversarial training method to ensure invariance to adversarial attacks. The authors also used the Mish activation and, Spatial Attention Module (SAM) as BoS methods to improve accuracy of the model. SAM uses a sigmoid function to highlight important features to ensure "more attention" on these features. YOLOv4 also introduced architectural changes to the backbone by introducing Cross-Stage Partial connections to the orginial DarkNet53 and named it CSPDarknet53. In the neck stage, a Path Aggregation Network (PAN) is added to increase feature richness and improve gradient propagation through the layers. The SAM module is added to the Head of the network for better accuracy and results. YOLOv5 [52] build on the YOLOv4 model and ports it to PyTorch library from the Darknet library. YOLOv5 also introduces a stem to the backbone to ensure fast computation, and a new SPP-Fast (SPPF) layer for faster runtimes. Finally, YOLOv5 also introduced several scaled models of the architecture from nano to extra-large.

YOLOv6 [53] modifies the backbone using RepVGG for smaller architectures and, a novel CSPStackRep layer for the larger architectures. The neck follows a PAN topology similar to YOLOv5 and additionally modified with RepVGG or CSPStackRep depending on the model scale. The head for YOLOv6 is modified to be anchor-free using the anchor point-based approach. The detection heads are further decoupled for tasks ensuring faster and more accurate performance of the model. The authors also adopted the Task Alignment Learning (TAL) method for label assignment which combines classification score and predicted box quality for a unified metric. Further, the authors also used a self-distillation strategy for training the models on both classification and regression tasks.

YOLOv7 [54] further builds on YOLOv6 and introduces new BoF methods for improving accuracy while maintaining inference runtime along with architectural changes to the network. Based on the Efficient Layer Aggregation Network (ELAN) computational block, YOLOv7 introduces an Extended ELAN (E-ELAN) architecture to enhance learning ability of the network using expand, shuffle and, merger cardinality without destroying the original gradient path. The YOLOv7 is concatenation-based in which traditional scaling methods, such as depth scaling do not work. The authors propose a novel strategy for model scaling in which the depth and width of the a block are scaled with the same factor to maintain the ratio between the input and output channel [53].

2.3 2D to 3D Pose Lifting

2D to 3D lifting approaches infer 3D pose from intermediately estimated 2D human pose using a single monocular image. Such methods employ a two stage framework where the 2D pose is estimated first followed by the 2D to 3D lifting approach. Benefitting by excellent research in for 2D pose estimation, such methods perform well compared to direct 3D coordinate es-

timation. [55] proposed a two stage solution, (1) Using an off-the shelf 2D pose estimation and, (2) Regressing 3D joints from 2D keypoints using simple machine learning methods. [56] follow the same approach as [55] but instead use a fully connected residual network to regress 3D joints from 2D joint locations. [57] adopted the 2D heatmaps instead of 2D pose locations as intermediate pose representations for estimating 3D pose. DRPose3D [58] use a pairwise ranking Convolutional Neural Network to predict the depth ranking of pairwise joints, followed by a coarse-to-fine pose estimator to regress 3D joint keypoints. Normalised 3D poses bring the additional value of being viewpoint and scale invariant which is extremely useful for the next step of pose comaprison.

2.4 Pose Comparison

A sequence of 3D poses extracted from a video can be treated as a time series data. Thus, similarity measures for comparing two non-aligned time series data were explored. Time series similarity measures can be parameterised as (1) Lock-step measure, (2) Elastic measures, (3) Threshold based measure and, (4) Pattern based measures [59]. For this research, we will look into the Elastic similarity measures as it allows for simliarity calculation between non-aligned temporal data. Dynamic Time Warping (DTW) [5] has been used as a similarity measure in finance applications [60]. DTW has also been employed to construct a pattern representation of stock time series for identifying similarities in trends in prices.

2.5 Critique of the review

This piece of work focuses on 3 main objectives. First, an existing 2D human pose estimation method is fine-tuned on a dataset with sports actions. Secondly, two novel models are implemented for lifting pose to 3D. Finally, a time-series comparison method is chosen to calculate action similarity between two videos containing sports actions. Subsequently, the research done as part of this work focuses on the 3 main objectives as mentioned above.

There has been abundant research in the fied of 2D human pose estimation. Recent research has not only improved the accuracy and dependability of such algorithms; however, there is limited research on the application of these topics to aid and enhance sports coaching. This is seen in the scarcity of datasets for 2D/3D pose estimation in a sports context. Only very recently, there are open-sourced datasets for pose estimation in a sports context available for public use. Finally, most research in this field prioritises accuracy over runtime thus, reducing the real-world applicability of such algorithms.

In the case of 3D pose estimation, the focus is more on predicting human joint keypoints based on multi-camera dataset. Research on predicting 3D pose keypoints using monocular RGB images need to catch up with its multi-view counterpart. While there has been some research in lifting 3D pose keypoints using monocular RGB images, none of these works hvae used the 1.5M frames large SportsPose dataset for training their models. This piece of work suggests a simple baseline for such methods using a sports-actions based 3D dataset.

2.6 Summary

This chapter focuses on the advancements in research on the topics of 2D pose estimation, 2D to 3D pose lifting and, time series comparison. Additionally, the literature review focuses extensively on the YOLO architectures and improvements over the years to understand the latest YOLOv8 model implemented. Research in 2D pose estimation and time series comparison is highly advanced in the number of publications as well the quality of work; however, there is scarce research in the estimation of 3D pose keypoints from monocular images using intermediate 2D keypoints from an off the shelf 2D pose estimator.

Chapter 3

Methodology

This section briefly describes the tasks completed and outlines the steps undertaken in implementing the research objectives. It will describe the reasoning behind the proposed algorithms, the training settings and, the evaluation metrics used for assessing performance. Further, the experiments conducted as part of the project are also outlined in this section.

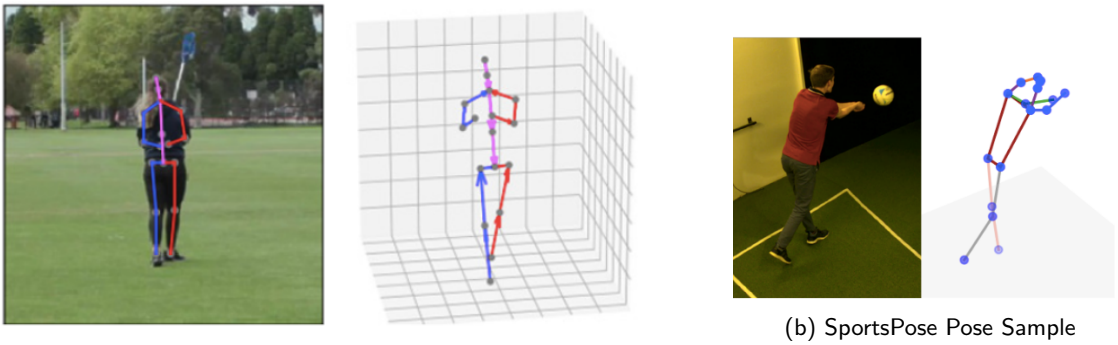
3.1 Task Description

As part of completing the aim described in Section 1.3, the following tasks are implemented:

- **2D Human Pose Estimation:** Human Pose Estimation is the taks of regressing human body keypoints such as joints given an image. For this project, an existing human pose estimation algorithm is fine-tuned on a combination of 2 sports motion based human keypoint datasets.
- **2D to 3D Pose Lifting:** To ensure view point invariance, the 2D human body keypoints are lifted to the 3D dimension. A novel approach for estimating normalised 3D human keypoints from 2D keypoints is proposed and implemented.
- **Pose Comparison:** The location of human body keypoints are compared over a sequence of frames. A module for scoring the user's soccer deadball technique is implemented using time series comparison methods.

3.2 Datasets

The ASPSet and SportsPose datasets were used as a starting point in this project. The ASPSet uses a flexible, low-budget vision pipeline for capturing 3D keypoints. The pipeline uses 3 cameras to capture video from different viewpoints. The 2D pose for each of these viewpoints is extracted. Further, for each frame, the 3D keypoints are calculated by triangulating the 2D locations from each viewpoint. The dataset contains different kinds of fast movements, including throwing, running, jumping etc., which is favourable for training deep learning models in a sports context. Additionally, the ASPSet consists of footage recorded on a football field with natural lighting, allowing models to generalise to in-the-wild video footage.



(a) ASPSet Pose Sample

(b) SportsPose Pose Sample

Figure 3.1: Sample Poses

SportsPose uses a complex setup of 7 calibrated and hardware synchronised colour camera to record video sequences at 90Hz. HRNet was used to predict the 2D pose for each frame captured by each camera. Using subsets of viewpoints for triangulation of 2D keypoints, a range of 3D point candidates were calculated. Further, a graph based approach is used to improve temporal continuity, followed by Butterworth smoothing, which reduced the number of 3D candidates. SportsPose contains video sequences for 5 activities, (1) Soccer, (2) Volleyball, (3) Baseball Pitch, (4) Jump and, (5)Tennis, in both indoor and outdoor settings. These activities cover a wide range of fast human body movements allowing for better perfoming models in a sports context. 2 new datasets, one for 2D human pose estimation and one for lifting 2D keypoints to 3D were created as part of this project. Figure 3.1 shows visualisations of 2D and 3D poses from the ASPset and the SportsPose dataset.

3.2.1 2D Human Pose Estimation Dataset

This section outlines the steps taken to preprocess the videos from the SportsPose and ASPSet. The preprocessing steps including frame sampling, resizing and normalising keypoints and are done before feeding the data for finetuning the 2D HPE model.

ASPSet

For each sample, the ASPSet provides three video files for the three viewpoints, a c3d file for the 3D keypoints, a csv file for the bounding box coordinates and, three JSON files for the camera matrices. First, the sampling frequency for extracting frames from video is set based on the frame rate of the video file. The sampling frequency is set such that 30 frames are extracted from each video. The 2D keypoints for each of the viewpoints are calculated by multiplying the 3D keypoints with the camera projection matrix for the specific viewpoint. Following this, a region of interest is determined from the frame by adding an offset to the bounding box keypoints. The albumentations library is used to crop and resize the image. Each image is resized to 640×640 resolution, adding reflect padding where necessary. Albumentations was specifically chosen because of its ability to manipulate keypoints and bounding box coordinates based on the transformations performed on an image. Finally, the keypoints and bounding box coordinates are normalised using the frame size, arranged in the format expected by the YOLOv8 model and, saved as a text file. The data training and validation splits provided by the authors of ASPSet has been used in this project as well. For parallel processing, all of these steps are wrapped inside a PyTorch DataLoader object.

SportsPose

The SportsPose dataset consists of 554 indoor videos and 100 outdoor videos. These videos

are collected in two lists for indoor and outdoor respectively and shuffled. The shuffled lists are then split into 443/110 and 80/20 training/validation split for indoor and outdoor respectively. The authors of SportsPose have released an open source library called sportspose containing a torch Dataset object that is used to sample data at a frame level. The video sequences have a constant frame rate of 270, hence every 9th frame is sampled to ensure 30 frames are extracted per video. Using the human joint keypoint data provided, a tight bounding box is regressed. The region of interest from each frame is determined by extending the tight bounding box. Albumentations is used again to crop and resize the image and keypoint and bounding box coordinates simultaneously. The annotations are created in the format expected by the YOLOv8 model and saved to the text file. To ensure fast, parallel processing, the steps are wrapped inside a torch DataLoader object.

Configuration File
The frames and annotations are represented using a YAML configuration file as expected by the YOLOv8 training function. This YAML file contains the root, training and validation directories. Further, the metadata for the keypoints including the keypoint shape, flip index and class is also provided.

3.2.2 Pose Lifting

The SportsPose dataset is used for creating the dataset for pose lifting as the 3D joints provided are normalised. All of the 176580 frames provided are used to create an extensive dataset for pose lifting. The 2D points provided are first normalised with respect to the frame size which is constant throughout the dataset. The 2D and 3D joints are binarized and saved into the same file for fast I/O operations using the Numpy library. The dataset and the configuration file is uploaded to a private Kaggle dataset.

3.3 You Only Look Once (YOLO) v8

Over the years, there has been abundant research to make improvements and changes to the architecture ([61], [49], [50], [51], [52], [53], [54]). YOLOv8 builds on top of YOLOv5 and is designed to be an anchor free model; indicating that the center of an object is directly predicted instead of the offset from a known anchor box. A new C2F layer is used as the main building block for the backbone of YOLOv8. The prediction heads for YOLOv8 are decoupled for different computer vision tasks using the same backbone architecture. A new Pose Head is introduced to extend the model's capabilities to human pose estimation as well. This section covers the various building blocks and architecture of YOLOv8 along with training and evaluation details.

3.3.1 YOLOv8 Building Blocks

Conv Block
This is the main building block of the whole network. It consists of a 2D convolutional layer, a batch normalisation layer and a Sigmoid Linear Unit (SiLU) activation layer. The convolution layer does not contain bias and padding is set to same padding to ensure high dimensional feature maps. Additionally, a 'forward_fuse' method is also implemented to improve computational efficiency when required. Figure 3.2 represents the layers in a Conv block and shows the flow of data in a single forward pass. Algorithm 1 shows the forward and forward_fuse

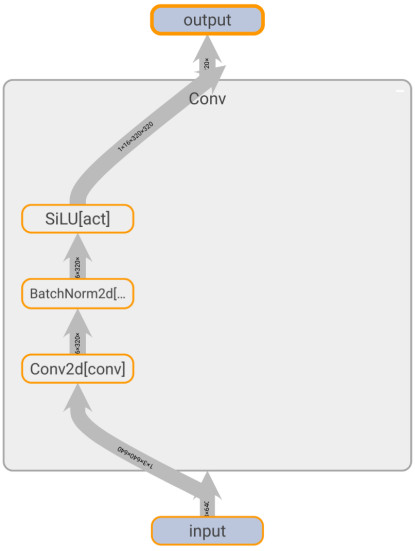


Figure 3.2: Conv Block Architecture Diagram

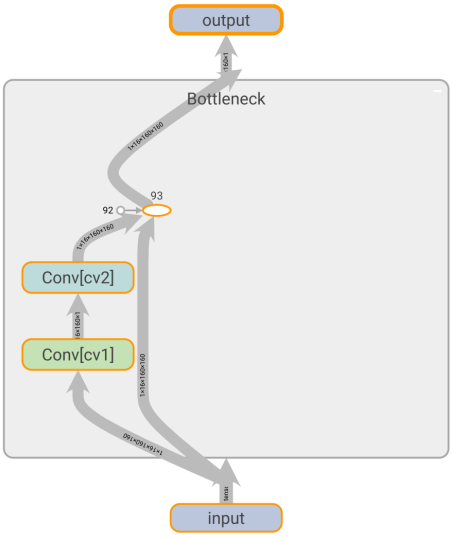


Figure 3.3: Bottleneck Architecture Diagram

methods of the Conv Block.

Algorithm 1 Conv Block Algorithm

Input: x ▷ tensor (batch, c, w, h)
Input: c1 ▷ Input channels
Input: c2 ▷ Output channels
Input: k=1 ▷ Kernel size
Input: s=1 ▷ Stride size
Input: p=None ▷ Padding size
Output: out ▷ tensor (batch, c, $\hat{w}$, $\hat{h}$)

```
struct Conv {
c = nn.Conv2d(c1, c2, k, s, autopad(k, p))
b = nn.BatchNorm2d(c2)
s = nn.SiLU()
};
function FORWARD(x, conv)
    out = conv.s(conv.b(conv.c(x)))
    return out
end function
function FUSEFORWARD(x, conv)
    out = conv.s(conv.c(x))
    return out
end function
```

BottleNeck Block

The Bottleneck Block is used to increase feature representation and reducing the number of parameters at the same time. The Bottleneck block consists of 2 convolution layers, each with 3×3-sized kernel and the same number of channels as the input. Inspired by the ResNet

bottleneck architecture, the inputs are added to the output of the 2 convolutional layers and outputted. Additionally, the YOLOv8 library also provides an parameter to skip the shortcut connection. Figure 3.3 represents the architecture and data flow through a single bottleneck layer. Algorithm 2 outlines the pseudocode for the BottleNeck component of the YOLOv8 algorithm.

Algorithm 2 Bottleneck Block Algorithm

Input: x ▷ tensor (batch, c, w, h)
Input: c1 ▷ Input channels
Input: c2 ▷ Output channels
Input: shortcut=True ▷ Shortcut
Input: g=1 ▷ Groups
Input: k=(3,3) ▷ Kernel Size
Input: e=0.5 ▷ Expansion Factor
Output: out ▷ tensor (batch, c, $\hat{w}$, $\hat{h}$)

```
struct Bottleneck {
c_ = int(c2 * e)
cv1 = Conv(c1, c_, k[0], 1)
cv2 = Conv(c_, c2, k[1], 1, g=g)
sc = shortcut and c1 == c2
};
function FORWARD(x, bottleneck)
    out = bottleneck.cv2(bottleneck.cv1(x))
    if sc then
        out = x+ out
    end if
    return out
end function
```

C2F Block

The C2F module (Cross-stage partial bottleneck with 2 convolutions) is a modified version of the CSP layer introduced in the YOLOv5 architecture where; the kernel size is increased to 3 from 1×1. The C2F implementation contains a parameter, *n*, to determine the number of Bottleneck Layers. The C2F block can be summarised in the architecture diagram in Figure 3.4. After the first convolution operation (Figure 3.4 Node Conv[cv1]), the data is split along the channel dimension. One half of the split is passed to the *n* Bottleneck Layers. The output from each of the bottleneck layer is concatenated with the output from the first convolutional layer. This penultimate output is then passed on to the final convolution operation and the output is obtained. Algorithm 3 outlines the pseudocode for the C2F block.

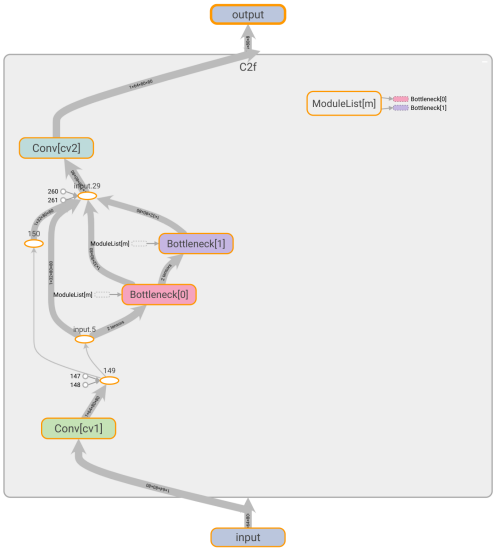


Figure 3.4: C2f Block Architecture Diagram

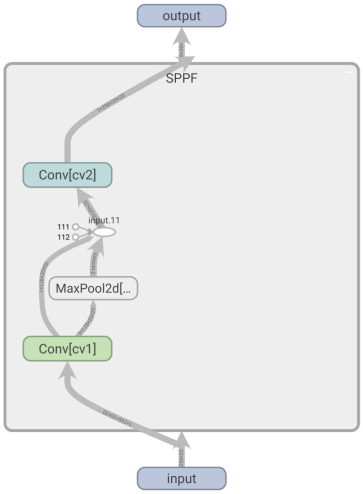


Figure 3.5: SPPF Architecture Diagram

Algorithm 3 C2F Block Algorithm

Input: x ▷ tensor (batch, c, w, h)
Input: c1 ▷ Input channels
Input: c2 ▷ Output channels
Input: n=1 ▷ Number of Bottlenecks
Input: shortcut=True ▷ Shortcut
Input: g=1 ▷ Groups
Input: e=0.5 ▷ Expansion Factor
Output: out ▷ tensor (batch, c, $\hat{w}$, $\hat{h}$)

struct C2F {
c_ = int(c2 * e)
cv1 = Conv(c1, 2 * c_, k[0], 1)
cv2 = Conv((2 + n)* c_, c2, k[1], 1)
ml = []
for i = 1 to n **do**
    ml.append(Bottleneck(c_, c_, shortcut, g, k=((3, 3), (3, 3)), e=1.0
end for
};
function FORWARD(x, c2f)
    out = c2f.cv1(x)
    out = list(out.split((c2f.c_, c2f.c_), 1))
    for m in c2f.ml **do**
        out.append(m(out[-1]))
    end for
    return c2f.cv2(concat(out, 1))
end function

SPPF Block

YOLOv8 introduces a new Spatial Pyramidal Pooling Fast (SPPF) Layer which improves the efficiency of the original Spatial Pyramidal Pooling (SPP) Layer in previous YOLO architectures. The SPPF block consists of 2 1×1 Conv Blocks and a MaxPool operation. After the first convolution operation, the data is max pooled thrice using the previous output as input. The outputs from the first Conv Block and the max pool outputs are concatenated before applying the second and final Conv Block operation. This is evident in Algorithm 4 which outlines the pseudocode for the SPPF block.. Figure 3.5 shows a block diagram of the SPPF block architecture.

Algorithm 4 SPPF Block Algorithm

Input: x ▷ tensor (batch, c, w, h)
Input: c1 ▷ Input channels
Input: c2 ▷ Output channels
Input: k=5 ▷ MaxPool Kernel Size
Output: out ▷ tensor (batch, c, $\hat{w}$, $\hat{h}$)

```
struct SPPF {
c_ = int(c1 / 2)
cv1 = Conv(c1, c_, 1, 1)
cv2 = Conv(c_ * 4, c2, 1, 1)
mp = nn.MaxPool2d(kernel_size=k, stride=1, padding=int(k / 2)
};
function FORWARD(x, sppf)
    out = []
    out.append(sppf.cv1(x))
    for i in 0 to 3 do
        out.append(sppf.mp(out[-1]))
    end for
    out = concat(out, 1)
    return sppf.cv2(out)
end function
```

Concat Block

The concat block takes dimension as input and concatenates a list of tensors along the specified dimension. Rather than using inbuilt PyTorch functions, This block is wrapped with 'torch.nn.Module' to ensure reproducability and reusability of the block.

Pose Head

The Pose Head Block is made of 3 Module lists each containing 3 Sequential blocks. Each sequential block consists of 2 Conv Blocks and 1 nn.Conv2d Layer. Each module list takes input from 3 previous C2F blocks ensuring multi-scale prediction. While the third module list is used for predicting human pose keypoints, the output from the first two modules are concatenated and used to predict the bounding box of the human body. Figure 3.6 represents the architecture diagram for the pose head.

3.3.2 Architecture

The YOLO architecture configuration can be summarised using a YAML file. This configuration is shown in code listing 3.1. The models are scaled according to the depth and width

expansion factors and the maximum number of channels. The configuration file defines the following parameters:

- **Number of Classes (nc)**
  This defines the number of classes that the YOLO model will expect. This also controls the initialisation of loss modules.
- **Keypoint Shape (kpt_shape)**
  This defines the keypoint shape that the model expects. This can be further customised during training as we will see in 3.3.5. YOLOv8 can accept either [17, 3] or [17, 2] shape with 2 dimensions for keypoints, or 3 dimensions for keypoints and visibility.
- **Model Scale (scales)**
  Defines the model scaling constants. This includes the depth expansion factor, width expansion factor and, the maximum number of channels that the network can expand on.
- **Backbone and Head**
  Defines the network architecture for all the scales of the model. The file follows the *From, Repeats, Module, Args* format. *From* indicates the input connections for the network, where -1 implies input is from the previous layer. *Repeats* indicates the number of times a layer is repeated. This value is affected by the depth expansion parameter. *Module* parameter defines the module that needs to be added to the network and *Args* define the arguments for the module. The arguments to the module is affected by the width expansion parameter.

```
# Ultralytics YOLO, AGPL-3.0 license
# YOLOv8-pose keypoints/pose estimation model. For Usage examples see https://docs.ultralytics.com/tasks/pose

# Parameters
nc: 1  # number of classes
kpt_shape: [17, 3]  # number of keypoints, number of dims (2 for x,y or 3 for x,y,visible)
scales: # model compound scaling constants, i.e. 'model=yolov8n-pose.yaml' will call yolov8-pose.yaml with scale 'n'
  # [depth, width, max_channels]
  n: [0.33, 0.25, 1024]
  s: [0.33, 0.50, 1024]
  m: [0.67, 0.75, 768]
  l: [1.00, 1.00, 512]
  x: [1.00, 1.25, 512]

# YOLOv8.0n backbone
backbone:
  # [from, repeats, module, args]
  - [-1, 1, Conv, [64, 3, 2]]  # 0-P1/2
  - [-1, 1, Conv, [128, 3, 2]]  # 1-P2/4
  - [-1, 3, C2f, [128, True]]
  - [-1, 1, Conv, [256, 3, 2]]  # 3-P3/8
  - [-1, 6, C2f, [256, True]]
  - [-1, 1, Conv, [512, 3, 2]]  # 5-P4/16
  - [-1, 6, C2f, [512, True]]
  - [-1, 1, Conv, [1024, 3, 2]]  # 7-P5/32
  - [-1, 3, C2f, [1024, True]]
  - [-1, 1, SPPF, [1024, 5]]  # 9

```

```
# YOLOv8.0n head
head:
  - [-1, 1, nn.Upsample, [None, 2, 'nearest']]
  - [[-1, 6], 1, Concat, [1]]  # cat backbone P4
  - [-1, 3, C2f, [512]]  # 12

  - [-1, 1, nn.Upsample, [None, 2, 'nearest']]
  - [[-1, 4], 1, Concat, [1]]  # cat backbone P3
  - [-1, 3, C2f, [256]]  # 15 (P3/8-small)

  - [-1, 1, Conv, [256, 3, 2]]
  - [[-1, 12], 1, Concat, [1]]  # cat head P4
  - [-1, 3, C2f, [512]]  # 18 (P4/16-medium)

  - [-1, 1, Conv, [512, 3, 2]]
  - [[-1, 9], 1, Concat, [1]]  # cat head P5
  - [-1, 3, C2f, [1024]]  # 21 (P5/32-large)

  - [[15, 18, 21], 1, Pose, [nc, kpt_shape]]  # Pose(P3, P4, P5)
```

Listing 3.1: YOLO yaml configuration file

3.3.3 Loss Functions

This section outlines the various loss functions used for training the YOLO models.

Classification Loss
For the classification loss, BCEWithLogitsLoss is used. BCEWithLogitsLoss combines a sigmoid layer and Binary Cross Entropy Loss (BCELoss) in one class. This implementation is more stable due to the log-sum-exp trick for numerical stability. Equation 3.1 represents the mathematical formulation of the BCEWithLogitsLoss function.

$$\ell(x, y) = \{l_1, \ldots, l_N\}^\top, \quad l_n = -w_n \left[y_n \cdot \log \sigma\left(x_n\right) + \left(1 - y_n\right) \cdot \log\left(1 - \sigma\left(x_n\right)\right) \right] \tag{3.1}$$

Bounding Box Loss
The Bounding Box Loss is calculated using a combination of Complete Intersection Over Union (CIoU) Loss and the Distribution Focal Loss (DFL). As mentioned in Section 2.2.3, CIoU loss incorporates distance between the centers and aspect ratios of the bounding boxes along with the euclidean distance between keypoints (Equation 3.3). Distribution Focal loss directly optimises a distribution of bounding box boundaries. DFL is represented mathematically as shown in Equation 3.2

$$\mathbf{DFL}\left(\mathcal{S}_i, \mathcal{S}_{i+1}\right) = -\left(\left(y_{i+1} - y\right) \log\left(\mathcal{S}_i\right) + \left(y - y_i\right) \log\left(\mathcal{S}_{i+1}\right)\right) \tag{3.2}$$

$$L_{CIoU} = 1 - IoU + \frac{d^2}{C^2} + \alpha v \tag{3.3}$$

where,
$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2$ and $\alpha = \frac{v}{(1-IoU)+v}$ is a trade-off parameter.

Keypoint Loss
The keypoint loss uses keypoint weights provided by the Microsoft COCO paper to calculate a weighted Euclidean distance loss. Additionally, a Binary Cross Entropy loss is used to classify object visibility, in this case keypoint visibility if provided with the ground truth data. In this

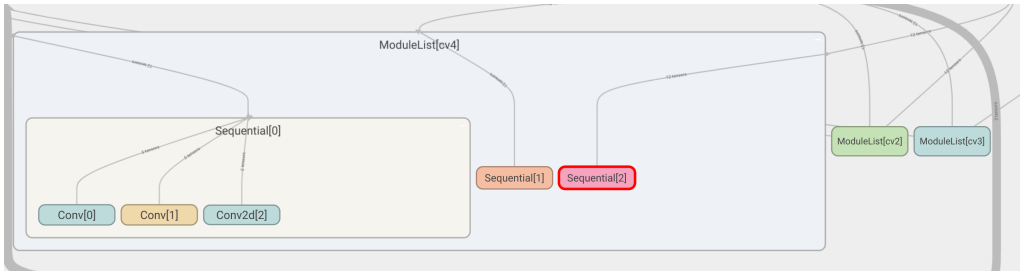


Figure 3.6: Pose Head Architecture

project, the visibility is not provided. Equation 3.4 shows a mathematical representation of the weighted Euclidean distance.

$$\mathbf{L} = \text{mean}\left(\frac{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}{2\sigma^2(\text{area})}\right) \tag{3.4}$$

3.3.4 Evaluation Metrics

Any keypoint detection, including bounding box and human body joints, can be parameterised into the following:

- **True Positives (TP):** When the prediction and ground truth are both positive given a confidence score.
- **True Negatives (TN):** When the prediction and ground truth are both positive given a confidence score.
- **False Positive (FP):** When the prediction is negative and ground truth is positive given a confidence score.
- **False Negative (FN):** When the prediction is positive and ground truth is negative given a confidence score.

The confidence scores are calculated using the Intersection-Over-Union (IoU) metric for bounding boxes and Object Keypoint Similarity (OKS) metric for pose estimation. If the metric is more than a specified threshold, the detection is considered to be positive. A brief description of OKS and IoU is given below.

Intersection Over Union (IoU)

Intersection over union is an evaluation metric for bounding box prediction. Given the ground truth box and prediction box, IoU is defined as the ratio of *area of overlap* between the predicted and ground truth bounding box over *area of union* between the predicted and ground truth bounding box. This metric can take values from 0 to 1, with 1 being a perfect match. Figure 3.7 shows examples of good and bad prediction based on the IoU metric.

Object Keypoint Similarity (OKS)

Object Keypoint similarity [13] for human pose estimation is analogous to IoU for object detection. Equation 3.5 shows the mathematical formula for calculating the OKS of a single human object in an image. To compute OKS, d_i is passed through an un-normalised Gaussian with standard deviation sk_i which yields a keypoint similarity between 0 and 1.

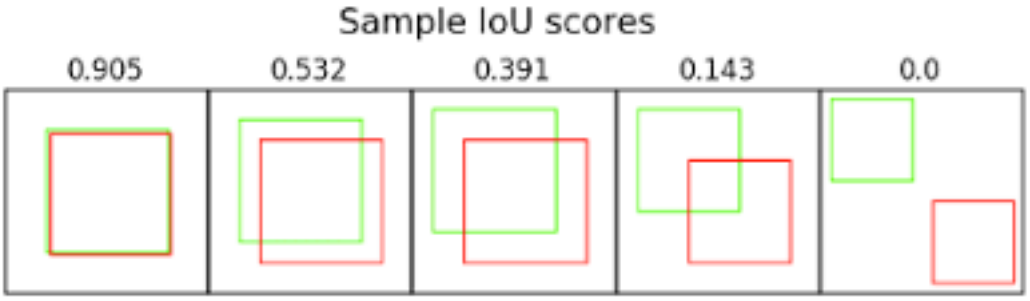


Figure 3.7: Sample IoU scores

$$OKS = \frac{\sum_{i \in K} \exp -\frac{d_i^2}{2s^2k_i^2}}{|K|} \tag{3.5}$$

where,

- d_i is the euclidean distance between ground truth and predicted keypoints.
- s is the scale of the bounding box and effectively the person
- k_i is the per-keypoint scale to equalise importance of keypoitns.
- K is the set of keypoints available in the dataset.

Using these metrics, the precision and recall for the human body keypoints and bounding boxes can be calculated at different confidence thresholds using the formula in Equation 3.6. Precision represents the number of TPs given all the positive predictions of a detector. Recall is a measure of the detector correctly identifying TPs given the ground truth keypoints.

$$\text{Precision} = \frac{TP}{FP + TP} \tag{3.6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.7}$$

mean Average Precision (mAP)
The average precision is defined as the weighted mean of precisions at different specified thresholds for a single class. The weights are defined using the increase in recall from the previous threshold. Algorithm 5 outlines the steps for calculating the Average Precision over multiple confidence thresholds. The mean Average Precision is calculated by averaging the AP for each class category. The mAP also follows the same precision level as AP.

Algorithm 5 Average Precision Threshold

Input: $\hat{y}$ ▷ Predicted keypoints
Input: y ▷ Ground truth keypoints
Input: C ▷ List of confidence scores
Output: out ▷ Average Precision

```
function AveragePrecision(ŷ, y)
    pr = []
    for c in C do
        Calculate IoU or OKS
        Calculate the TP, TN, FP, FN
        Calculate the Precision Recall curve
        pr.append(area under PR curve)
    end for
     return sum(pr)
end function
```

3.3.5 Training and Evaluation

As part of this project, YOLOv8-n(Nano) and YOLOv8-x(Xtra Large) were trained. Both the models were trained to optimise the losses defined in Section 3.3.3. The training details and settings are detailed further in this section.

YOLOv8-n
The YOLOv8-n model was fine-tuned using Automatic Mixed Precision (AMP) for optimising training speed and performance with 16-bit Floating Point (FP16) numbers reducing memory requirements. The dataset consists of 51084 images for training and 9384 images for validation. SGD was used to fine-tune the model with 0.002 learning rate, 0.9 momentum and, 0.0005 weight decay. The image sizes were fixed at 640×640 for both training and validation. The albumentations library was used for augmenting training images using Blur and Contrast Limited Adaptive Histogram Equalisation (CLAHE) to ensure fast movements are detected during inference runtimes. For training the batch size was set to 16, while for validation the batch size was set to 32. The model was fine-tuned using a Nvidia Tesla T4 GPu for a total of 50 epochs.

YOLOv8-x
The YOLOv8-x model was fine-tuned using Automatic Mixed Precision (AMP) for optimising training speed and performance with 16-bit Floating Point (FP16) numbers reducing memory requirements. The dataset consists of 51084 images for training and 9384 images for validation. AdamW was used to fine-tune the model with 0.002 learning rate, 0.9 momentum and, 0.0005 weight decay. The image sizes were fixed at 640×640 for both training and validation. The albumentations library was used for augmenting training images using Blur and Contrast Limited Adaptive Histogram Equalisation (CLAHE) to ensure fast movements are detected during inference runtimes. For training the batch size was set to 16, while for validation the batch size was set to 32. The model was fine-tuned using a Nvidia Tesla T4 GPU for a total of 20 epochs.

3.4 3D Pose Lifting

Lifting keypoints to the 3D space from 2D is necessary to ensure viewpoint invariance. This project proposes 2 novel methods for lifting pose. The first method uses a simple feed forward network for predicting, while the second network using 1D convolutions in residual blocks for predicting 3D pose. The details of training and network design are elaborated below.

3.4.1 Loss Function

The L1 Loss function is used for training the model for lifting the pose to 3D. The L1 loss calculates the Mean Absolute Error between each element in the predicted and ground truth keypoints. As mentioned in Section 3.2.2, the dataset for training the proposed models is normalised, and takes on values around 0. The L1 loss is a good candidate in such situations as it will highlight smaller errors better than Euclidean distance loss or L2 Loss. Equation 3.8 shows the mathematical formula for calculating L1 loss.

$$\text{L1} = \sum_{i=1}^{N} \quad \text{mod } y - \hat{y} \quad \text{mod} \tag{3.8}$$

3.4.2 Feed Forward Network

Architecture

In this project, a simple feed forward network is proposed for lifting 2D keypoints to 3D. The network consists of 4 linear layers. The first 3 linear layers are followed by a Batch Normalisation Layer and ReLU activation. The final layer is the output layer that predicts the 3D keypoints. The weights for the linear layers are initialised using He initialisation for ReLU activations, whereas the biases are initialised as 0. To ensure reproducability, the seed is set to 42 before initialising the weights. Equation 3.9 shows the formula for the distribution from which weights are sampled. Figure 3.8 represents the architecture of the network used in this project.

$$\mathcal{N}(0, \text{std}^2), \quad \text{std} = \frac{\text{gain}}{\sqrt{\text{fan\_mode}}} \tag{3.9}$$

Training

The model is trained using the dataset specified in 3.2.2. The training batch size is set to 64 with shuffling enabled. The validation batch size is set to 128 without shuffling. The Stochastic Gradient Descent (SGD) optimiser with 0.001 learning rate and 0.00005 weight decay is used for training the model. Additionally, the learning rate is scheduled to be reduced by a factor of 0.1 when the validation loss stops improving. The model is trained using a Nvidia Tesla T4 GPU using full floating point precision.

3.4.3 Convolutional Neural Network

Architecture

The second model proposed for the task of lifting pose from 2D to 3D is a 1D convolutional network with ResNet shortcut blocks. A (1×1) convolution layer is initially as a stem, followed by a Batch Normalisation and ReLU activation layer. This has been followed by six residual blocks. The residual blocks consists of two $(3 \times 3$ convolutional layers with Batch Normalisation and ReLU activation. Another 1×1 convolutional layer is used to convert the input to the number of output channels for the shortcut connection. Figure 3.10 shows the

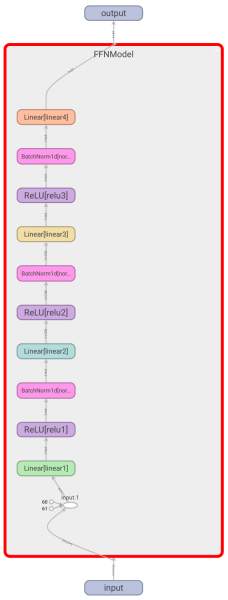


Figure 3.8: Feed Forward Network

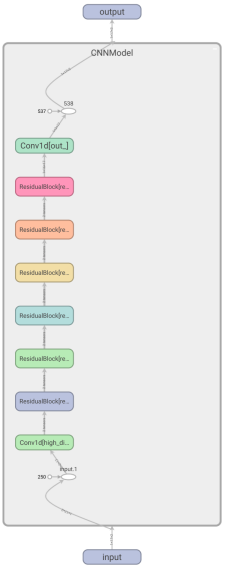


Figure 3.9: Convolutional Neural Network with Residual Blocks

design for a residual block. The residual blocks are followed by a 1×1 convolutional layer that outputs a (17, 3) shaped tensor representing the 3D pose.

Training
The model is trained using the dataset specified in 3.2.2. The training batch size is set to 64 with shuffling enabled. The validation batch size is set to 128 without shuffling. The Stochastic Gradient Descent (SGD) optimiser with 0.001 learning rate and 0.00005 weight decay is used for training the model. Additionally, the learning rate is scheduled to be reduced by a factor of 0.1 when the validation loss stops improving. The model is trained using a Nvidia Tesla T4 GPU using full floating point precision.

3.5 Pose Comparison

The sequence of pose exracted from a video is treated as time-series data enabling the usage of classical time-series comparison methods. A straightforward method would be to calculate the euclidean distances between the keypoints for every time step. However; this method assumes perfectly aligned video sequences with equal length which may not always be the case. Dynamic Time Warping is a similarity measure between time series that seeks for the temporal alignment of data that minimised Euclidean distance. This method is invariant to the size of the time-series data being compared as well as the alignment of the data. Equation 3.10 represents the mathematical formula for calculating DTW metric. In our project, the DTW metric is modified to use the cosine similarity for calculating the distance between the keypoints to ensure that both direction and magnitude are the same. The reference video is taken from the SoccerKicks dataset which contains videos of players executing Freekicks and penalty kicks.

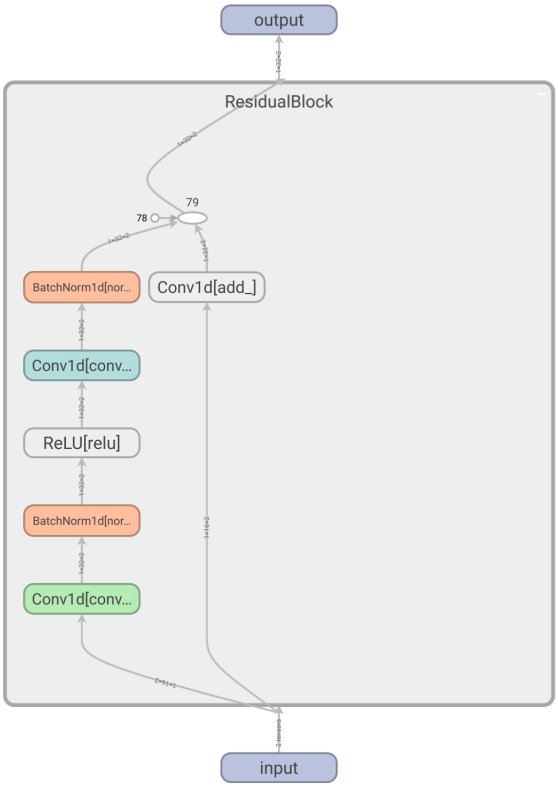


Figure 3.10: Residual Block

$$DTW_q(x, x\prime) = \min_{\pi \in A(x,x\prime)} \left(\sum_{(i,j)\in\pi} d(x_i, x_j\prime)^q \right)^{\frac{1}{q}} \tag{3.10}$$

where, π is the alignment path of length K consisting of K index pairs and $A(x, x\prime)$ is set of all admissible paths.

3.6 Summary

This chapter starts by describing the tasks implemented as part of the project in Section 3.1. Section 3.2 provides a description of the datasets used as part of the dissertation project, expanding on the data collection methods and the range of movements covered. Subsection 3.2.1 expands on the pre-processing techniques used on the ASPset and SportsPose datasets to make them ready for training including frame sampling, resizing and keypoint transformations. Section 3.2.2 outlines the methodologies used to create a pose lifting dataset from the existing SportsPose and ASPSet datasets. Section 3.3 outlines a description of the YOLO algorithm used for 2D human pose estimation. Section 3.3 is further divided into subsections that cover Building Blocks (Section 3.3.1), Architecture and Network Design (Section 3.3.2), Loss functions (Section 3.3.3), Evaluation metrics (Section 3.3.4) and finally the Training and Validation strategies (Section 3.3.5). Further, Section 3.4 provides a description of the novel methods implemented to lift 2D pose keypoints to 3D. Section 3.4 is fragmented into 3 sub sections that cover the Loss Function (Section 3.4.1, the proposed Feed Forward Network (Section 3.4.2) and, the proposed Convolutional Neural Network (Section 3.4.3). Finally,

Section 3.5 outlines the methodology used for implementing a scoring module that compares 2 temporal sequences of human pose keypoints.

Chapter 4

Results

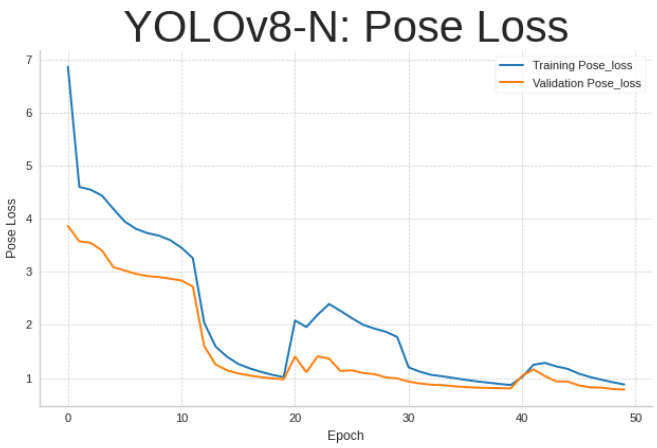
This chapter expands on the results and findings of the experiments conducted as part of the dissertation project. The experiments include training 2 versions of the YOLOv8 Pose model; YOLOv8-Nano and YOLOv8-XtraLarge for 2D Human Pose Estimation. Next, the performance of the novel neural networks proposed; a feed forward neural network and a convolutional neural network is analysed. The analysis of the 4 different models include both quantitative metrics and a visual inspection of the ground truth and the predicted keypoints.

4.1 YOLOv8-N 2D Human Pose Estimation

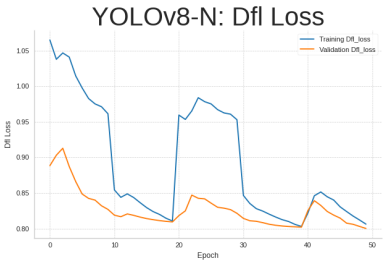
4.1.1 Quantitative Analysis

The YOLOv8N model is trained on the dataset using the methodologies as outlined in Section 3.2.1 and the loss function outlined in Section 3.3.3. The model was trained for 50 epochs using the SGD optimiser. After 50 epochs, YOLOv8N performed with a Pose Loss of 0.868 on the training dataset and 0.783 on the validation dataset. Further, the model benchmarked 0.242 and 0.233 Bounding Box loss on the training and validation dataset respectively. Finally, the model performed with a Classification Loss of 0.137 and 0.120 on the training and validation datasets respectively. Figure 4.1 shows the improvement in loss metrics with respect to the number of epochs trained.

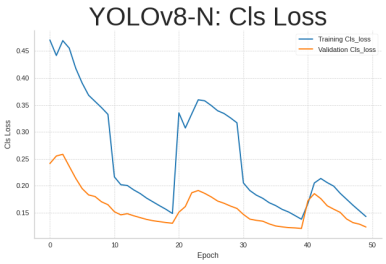
The YOLOv8N model is further evaluated with the metrics described in Section 3.3.4. After 50 epochs, the model predicted human body bounding boxes with 0.99854 precision, 0.99712 recall, 0.983 mAP@50-95 and, 0.995 mAP@50. Figure 4.2 shows the improvement in evaluation metrics for human body bounding box over the training period. Further, the model predicted human body keypoints with 0.99766 precision, 0.99766 recall, 0.88619 mAP@50-95 and, 0.99415 mAP@50. Figure 4.3 outlines the performance improvement in human body keypoint prediction over 50 epochs.



(a) Pose Loss

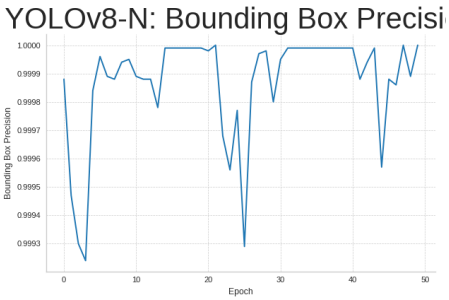


(b) DFL Loss

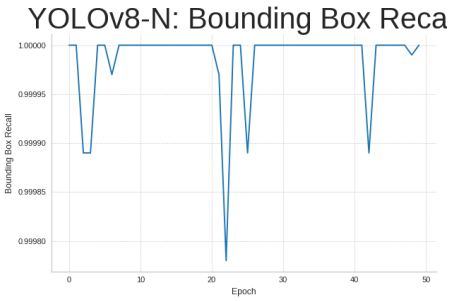


(c) Classification Loss

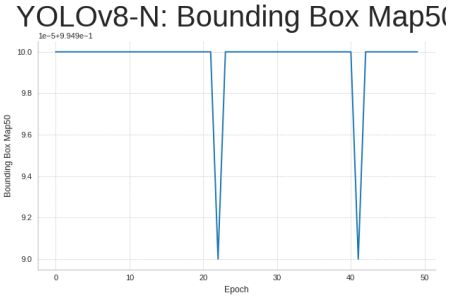
Figure 4.1: Loss Metrics for YOLOv8N



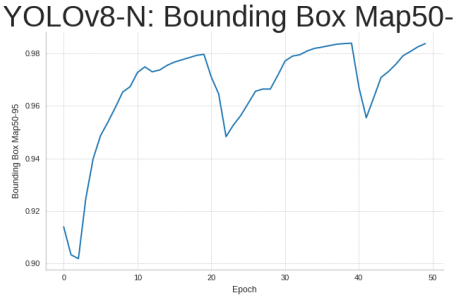
(a) YOLOv8N Bounding Box Precision



(b) YOLOv8N Bounding Box Recall

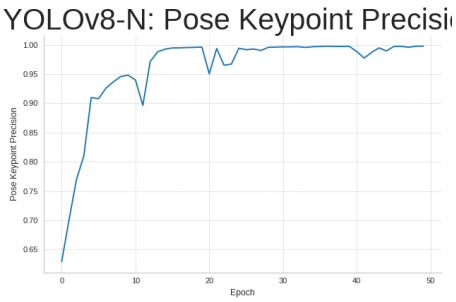


(c) YOLOv8N Bounding Box mAP@50

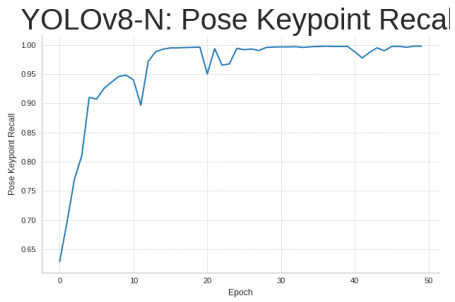


(d) YOLOv8N Bounding Box mAP@50-95

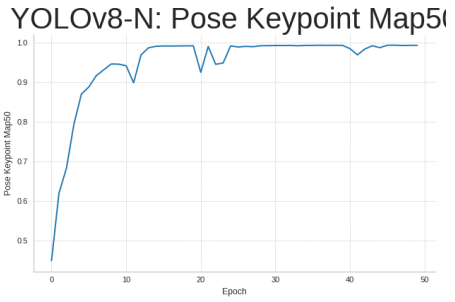
Figure 4.2: Bounding Box Evaluation Metrics for YOLOv8N



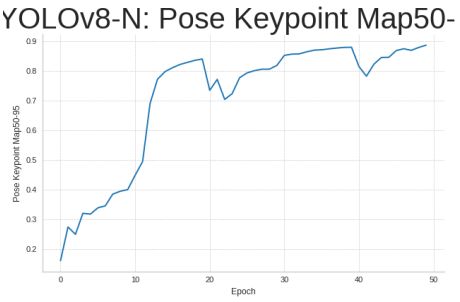
(a) YOLOv8N Pose Keypoint Precision



(b) YOLOv8N Pose Keypoint Recall



(c) YOLOv8N Pose Keypoint mAP@50



(d) YOLOv8N Pose Keypoint mAP@50-95

Figure 4.3: Pose Keypoint Evaluation Metrics for YOLOv8N

4.1.2 Visual Inspection

A visual inspection of the model performance was done by plotting the ground truth and model predictions on the input image. Figure 4.4 contains the model predictions of the YOLOv8N model on various activities included in the SportsPose Dataset; red indicates the predicted values and green indicates the ground truth. As seen in Figure 4.4, the fine-tuned YOLOv8N model works well on various view points and body orientation.

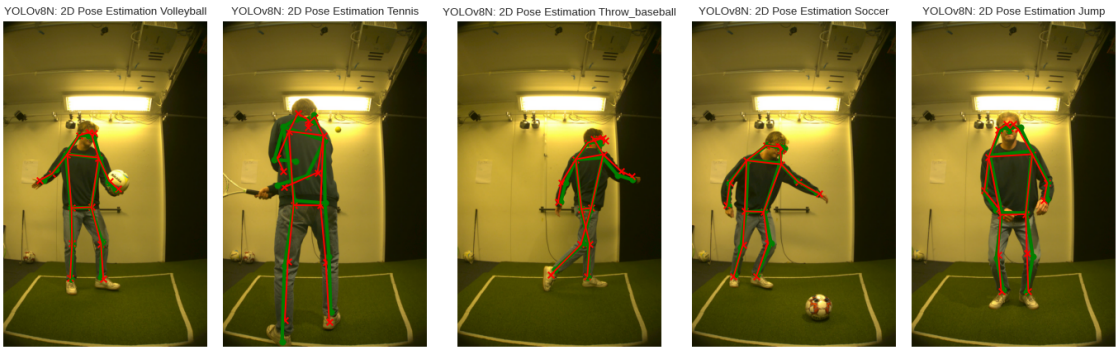


Figure 4.4: YOLOv8N Predictions (Red) vs Ground Truth (Green)

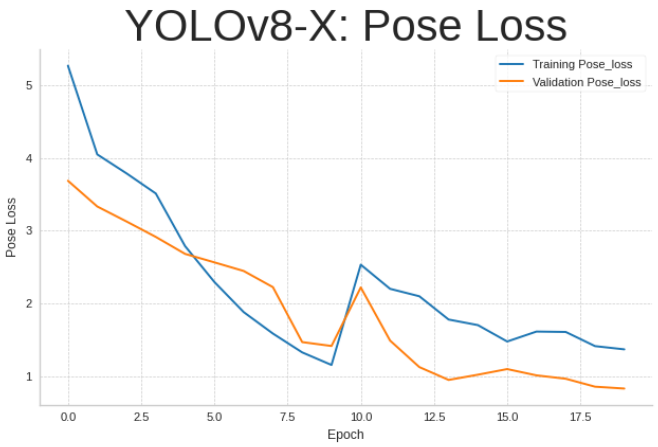
4.2 YOLOv8-X 2D Human Pose Estimation

4.2.1 Quantitative Analysis

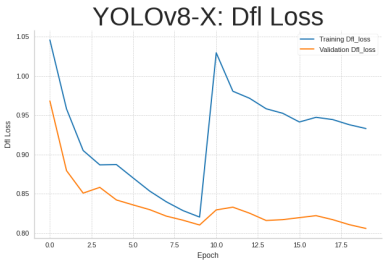
The YOLOv8X model is trained on the dataset using the methodologies as outlined in Section 3.2.1 and the loss function outlined in Section 3.3.3. The model was trained for 50 epochs

using the AdamW optimiser. After 20 epochs, YOLOv8X performed with a Pose Loss of 1.1526 on the training dataset and 0.82792 on the validation dataset. Further, the model benchmarked 0.82024 and 0.80574 Bounding Box loss on the training and validation dataset respectively. Finally, the model performed with a Classification Loss of 0.15865 and 0.12855 on the training and validation datasets respectively. Figure 4.5 shows the improvement in loss metrics with respect to the number of epochs trained.

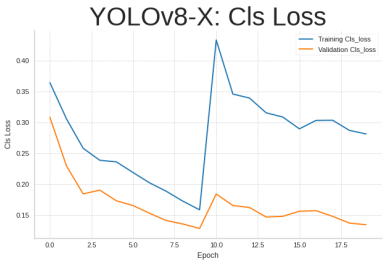
The YOLOv8X model is further evaluated with the metrics described in Section 3.3.4. After 20 epochs, the model predicted human body bounding boxes with 0.99614 precision, 0.99838 recall, 0.97725 mAP@50-95 and, 0.995 mAP@50. Figure 4.2 shows the improvement in evaluation metrics for human body bounding box over the training period. Further, the model predicted human body keypoints with 0.99619 precision, 0.99616 recall, 0.86299 mAP@50-95 and, 0.99383 mAP@50. Figure 4.3 outlines the performance improvement in human body keypoint prediction over 20 epochs.



(a) Pose Loss

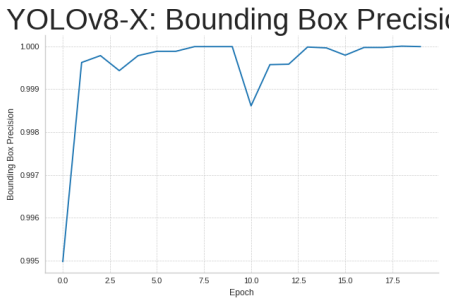


(b) DFL Loss

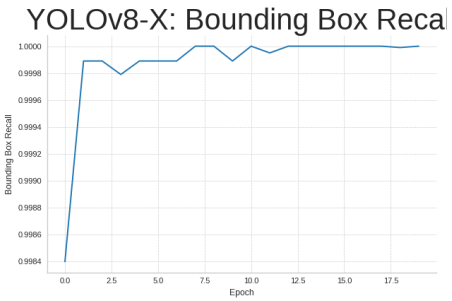


(c) Classification Loss

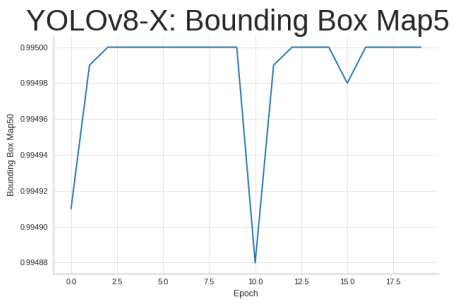
Figure 4.5: Loss Metrics for YOLOv8X



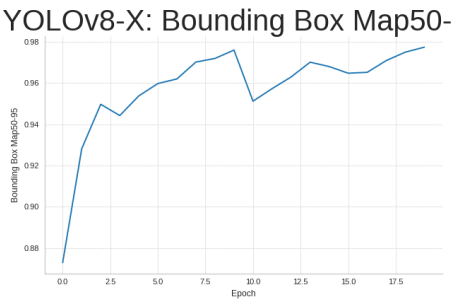
(a) YOLOv8X Bounding Box Precision



(b) YOLOv8X Bounding Box Recall

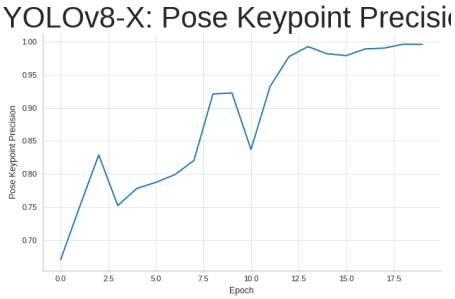


(c) YOLOv8X Bounding Box mAP@50

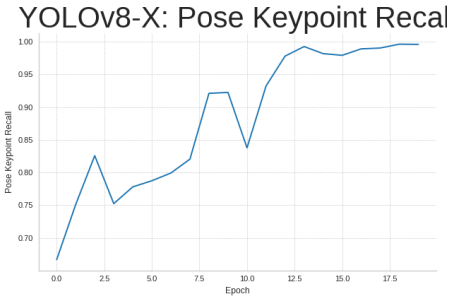


(d) YOLOv8X Bounding Box mAP@50-95

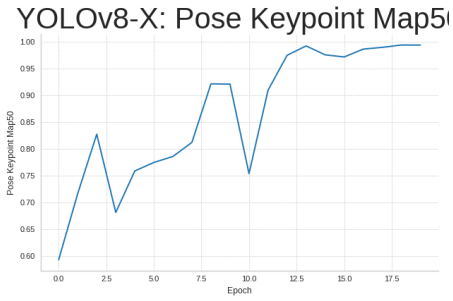
Figure 4.6: Bounding Box Evaluation Metrics for YOLOv8X



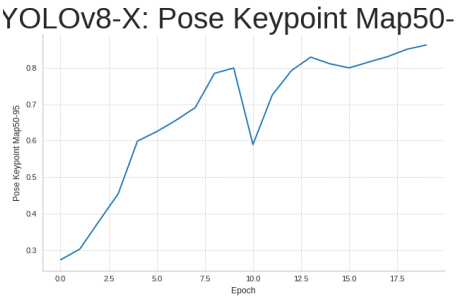
(a) YOLOv8X Pose Keypoint Precision



(b) YOLOv8X Pose Keypoint Recall



(c) YOLOv8X Pose Keypoint mAP@50



(d) YOLOv8X Pose Keypoint mAP@50-95

Figure 4.7: Pose Keypoint Evaluation Metrics for YOLOv8X

4.2.2 Visual Inspection

A visual inspection of the model performance was done by plotting the ground truth and model predictions on the input image. Figure 4.8 contains the model predictions of the YOLOv8X model on various activities included in the SportsPose Dataset; red indicates the predicted values and green indicates the ground truth. As seen in Figure 4.8, the fine-tuned YOLOv8X model works well on various view points and body orientation.

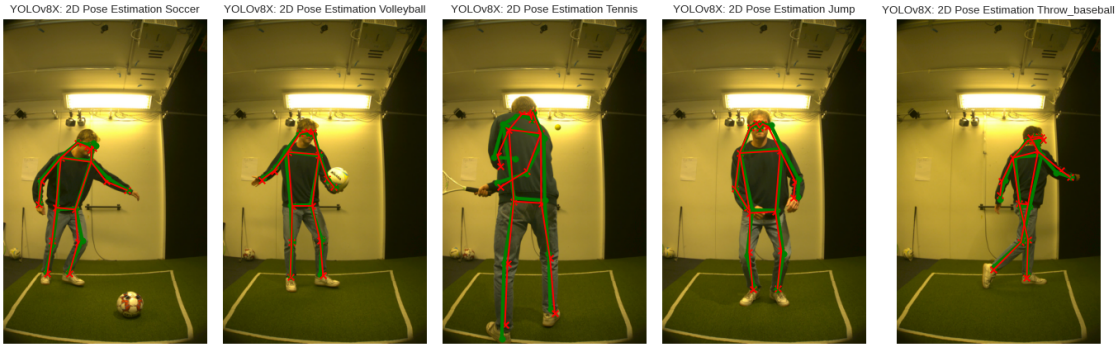


Figure 4.8: YOLOv8X Predictions (Red) vs Ground Truth (Green)

4.3 CNN based Pose Lifting

4.3.1 Quantitative Analysis

The CNN model described in Section 3.4.3 is trained using the dataset created using the methodology described in Section 3.2.2. The trained model performed with a 0.1099 L1 Loss on the training dataset and 0.09906 L1 loss on the validation dataset. Figure 4.9 represents the improvement in L1 Loss of the CNN model over 100 epochs.

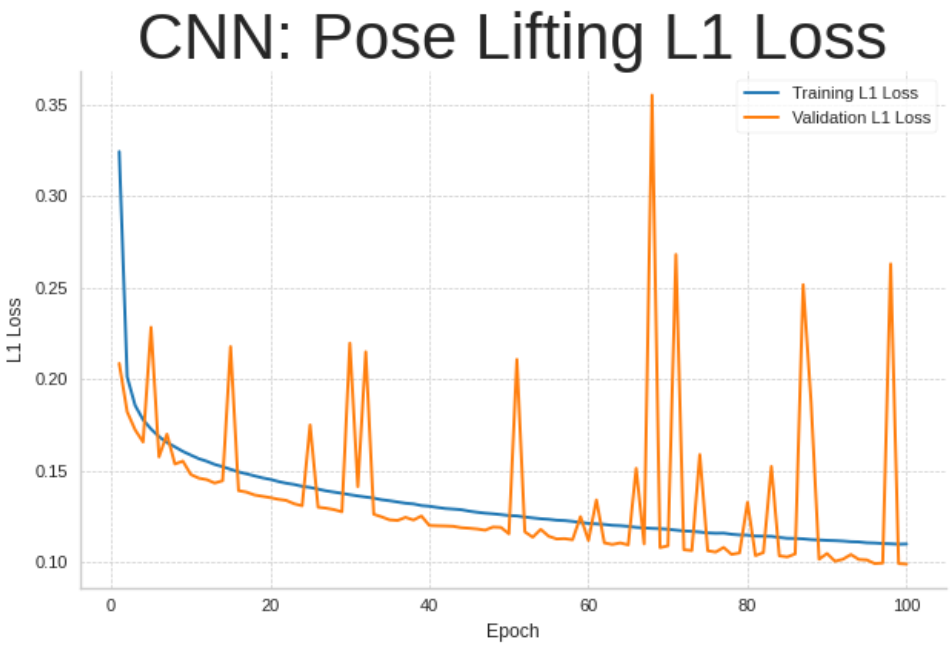


Figure 4.9: CNN L1 Loss

4.3.2 Visual Inspection

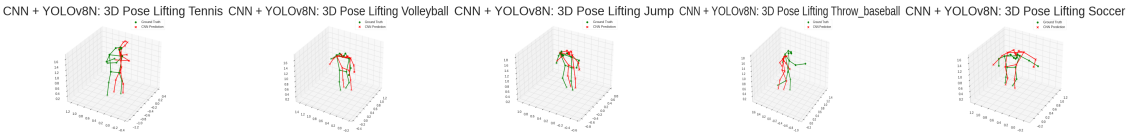


Figure 4.10: 3D pose reconstructed using CNN with keypoints obtained by YOLOv8N

A visual inspection of the results was conducted by plotting the 3D ground truth and 3D predictions outputted by the CNN model. This inspection was done using the 2D keypoint outputs from both YOLOv8N and YOLOv8X models as input for the CNN model. Figure 4.10 shows the comparision of the ground truth keypoints (green) and predicted keypoints (red) for the 3D keypoints predicted using YOLOv8N and CNN models. Similarly, Figure 4.10 shows the comparision of the ground truth keypoints (green) and predicted keypoints (red) for the 3D keypoints predicted using YOLOv8X and CNN models.

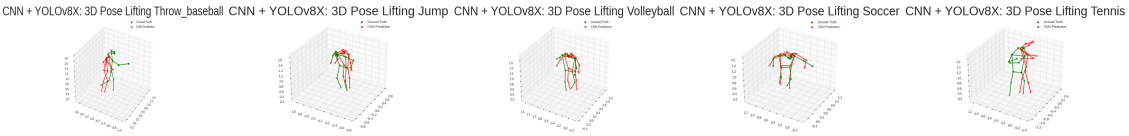


Figure 4.11: 3D pose reconstructed using CNN with keypoints obtained by YOLOv8X

4.4 FFN based Pose Lifting

4.4.1 Quantitative Analysis

The FFN model described in Section 3.4.2 is trained using the dataset created using the methodology described in Section 3.2.2. The trained model performed with a 0.14942 L1 Loss on the training dataset and 0.14819 L1 loss on the validation dataset. Figure 4.12 represents the improvement in L1 Loss of the FFN model over 100 epochs.

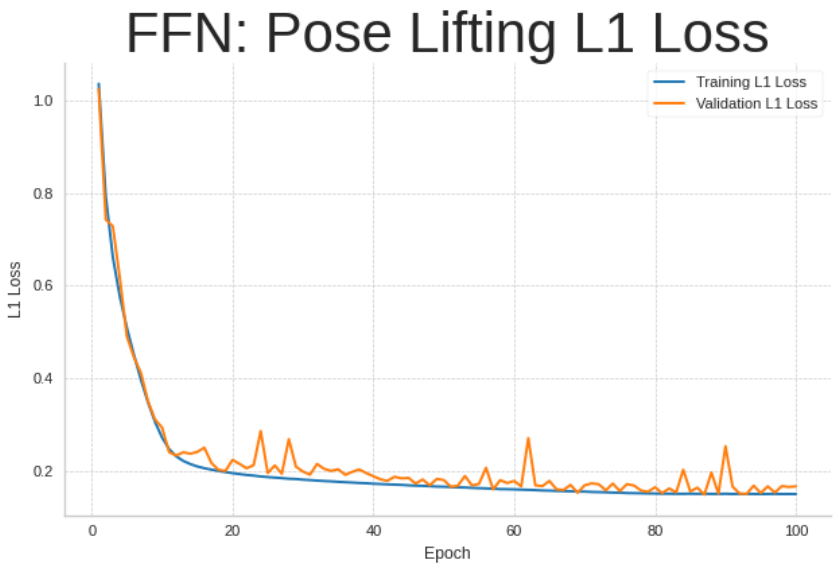


Figure 4.12: FFN L1 Loss

4.4.2 Visual Inspection

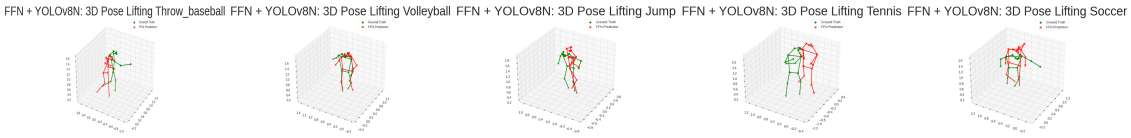


Figure 4.13: 3D pose reconstructed using FFN with keypoints obtained by YOLOv8N

A visual inspection of the results was conducted by plotting the 3D ground truth and 3D predictions outputted by the FFN model. This inspection was done using the 2D keypoint outputs from both YOLOv8N and YOLOv8X models as input for the FFN model. Figure 4.10 shows the comparision of the ground truth keypoints (green) and predicted keypoints (red) for the 3D keypoints predicted using YOLOv8N and FFN models. Similarly, Figure 4.10 shows the comparision of the ground truth keypoints (green) and predicted keypoints (red) for the 3D keypoints predicted using YOLOv8X and FFN models.

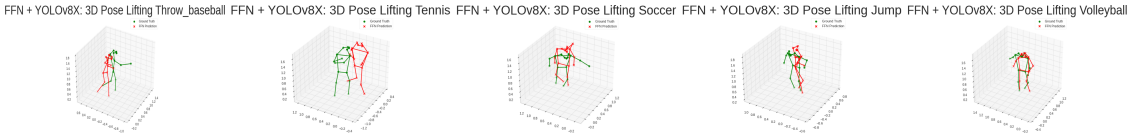


Figure 4.14: 3D pose reconstructed using FFN with keypoints obtained by YOLOv8X

4.4.3 Pose Comparison

The chosen DTW method for comparing pose sequences work well in this project. Figure 4.15 shows a comaprison between the reference (Green) and the query (Red). This particular set of reference of query and reference videos outputted a DTW distance of 2.827. This is evident by a visual inspection which shows a slight disparity in the query and reference poses.

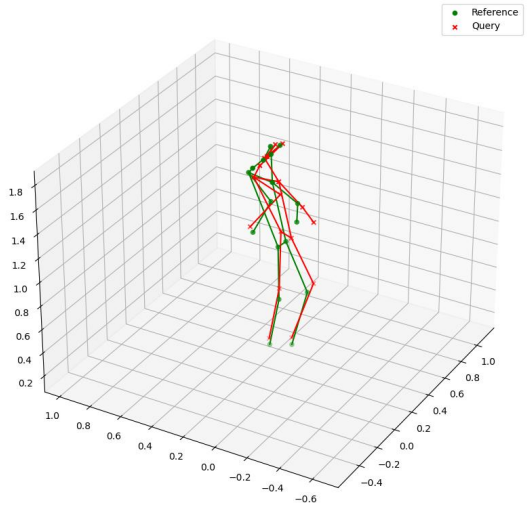


Figure 4.15: Visual Inspection of Reference (Green) and Query (Red) pose

4.5 Summary

The models were evaluated using both quantitative metrics and visual inspection. Section 4.1 outlines the results obtained using the YOLOv8N model for 2D Human Pose Estimation. Further, Section 4.1.1 expands on the quantitative metrics logged during training and finally, Section 4.1.2 outlines the visual inspection analysis of YOLOv8N model and the plots used. Section 4.2 outlines the results obtained using the YOLOv8X model for 2D Human Pose Estimation. Further, Section 4.2.1 expands on the quantitative metrics logged during training and finally, Section 4.2.2 outlines the visual inspection analysis of YOLOv8N model and the plots used. The proposed pose lifting models were evaluated using inputs from the 2 2D pose estimation methods. Section 4.3 outlines the results for 3D pose estimation obtained using the proposed CNN pose lifter. Additionally, Section 4.3.1 expands on the quantitative metrics logged during training and finally, Section 4.3.2 outlines the visual inspection analysis of the CNN model with 2D keypoints estimated using both YOLOv8N and YOLOv8X. Section 4.4 outlines the results for 3D pose estimation obtained using the proposed FFN pose lifter. Further, Section 4.4.1 expands on the quantitative metrics logged during training and finally, Section 4.4.2 outlines the visual inspection analysis of the FFN model with 2D keypoints estimated using both YOLOv8N and YOLOv8X.

Chapter 5

Discussion and Analysis

5.1 Significance of Findings

The 2 YOLOv8 architectures for Human Pose Estimation demonstrated promising performance in estimating 2D human pose keypoints with quantifiable improvements during training. The models for 2D human pose estimation also demonstrate good performance and high evaluation metrics on low resolution images. The YOLOv8N model is light-weight, fast, efficient and easy to train on limited resources. This project establishes a precedent for estimating 3D pose by lifting 2D keypoints in a fast-moving, sports context.

The proposed 3D pose estimation pipeline works for a wide range of environments including both indoor and outdoor environments. The 3D pose estimation pipeline demonstrates good performance on varying human body scale in images and videos. The proposed models perform well on a wide range of fast movements. Finally, the proposed novel methods for 3D pose lifting perform well over a wide range of 2D viewpoints and is able to generalise well.

5.2 Limitations

The accuracy of the dataset is not evaluated potentially leading to the model learning on wrong data points. The dataset only covers movements of 5 sports activities and the model might not generalise on other movements. For 2D human pose estimation, only YOLOv8 was trained and evaluated. Usage of other State-Of-The-Art methodologies could provide comparisons on the results.

Due to lack of datasets for 3D pose estimation in a sports context, the trained models could not evaluated on other datasets. The trained models are not trained on in-the-wild data potentially leading to bad performance in such data. Finally, The models were trained for a small number of epochs due to resource constraints, potentially limiting the capabilities of the model.

5.3 Summary

This chapter discusses the significance and limitations of the project. Section 5.1 outlines the significant achievements of the projects. Section 5.2 describes the potential limitations and difficulties faced during the implementation of the project.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Human Pose Estimation is a classical computer vision task that deals with predicting human body keypoints from images. However, human pose estimation is complicated due to high variance in image and blur due to fast movements especially in sports. This problem has been targeted by numerous deep learning-based models with bigger datasets to tackle the issue of high variance. Recently published sports based datasets such as SportsPose allow for further research in tackling the issue of blur in human pose estimation. The sequence of poses can be treated as temporal data and classical time series comparison methods can be used for scoring the technique. This project aims to employ human pose estimation in a sports context to aid and enhance sports coaching by outsourcing technique correction to AI models.

For the 2D human pose estimation task, a combined dataset created from ASPset and SportsPose datasets was used to train the models. The dataset samples 30 frames from each video. Each frame is resized to 640×640 with reflective padding. For pose lifting, each frame from the SportsPose dataset is used. The 2D and 3D keypoints are normalised to ensure view point and scale invariance of the results.

Two scales of the YOLOv8 model were trained and used to extract 2D poses from video frames. The lighter YOLOv8-N model was trained for 50 epochs with the SGD optimiser and blur augmentation to ensure generalisation for fast movements. YOLOv8-N performed with a Pose Loss of 0.868 and 0.995 MAP@50 on the custom dataset. The bulkier YOLOv-8X model was trained for 20 epochs with the AdamW optimiser. YOLOv8-X performed with a Pose Loss of 1.152 and 0.995 mAP@50 on the custom dataset. Further gains may be realised by further training the model, higher frequency frame sampling and better augmentation techniques. The proposed CNN model was trained for 100 epochs with SGD optimiser and performs with L1 Loss of 0.1099 on the pose lifting dataset. The FFN model performs with a L1 loss of 0.1492 on the dataset after being trained for 100 epochs using SGD optimiser. This project sets the precedent for the application of deep learning based 3D human pose estimation in the field of technique correction in sports. Thus it is concluded that the objectives set as part of this project has been fulfilled.

6.2 Future work

Datasets
The datasets used in this project covered a range of sports movements however; there are many more movements possible in the sports domain. Inclusion of these movements would help improve the quality of the dataset as well as the model. Additionally, the data was collected in controlled situations in both indoor and outdoor settings. A dataset with frames or images in the wild would increase the variance in the images and improve the generalisation of human pose estimation model. Finally, since these datasets were collected without the use of body markers or sensors, there is no way to quantify the accuracy of the labels.

2D Human Pose Estimation
The models used for 2D Human Pose Estimation works well the dataset used. The results can be further improved with continued training of the model for more number of epochs. More augmentation techniques such as flipping and rotation can be used to improve the performance of them models. Finally, additional models with varying scales could be trained to compare the performance of different models on the combined dataset. Better and bigger datasets can be used for further improving the performance and learning capabilities of the model.

3D HPE
The proposed models for lifting the keypoints to 3D work well with the given dataset. The performance can be further improved by adopting a heatmap based input for the models, where a Gaussian distribution is used to create a heatmap around keypoints of the human body. This ensures that the spatial and geometric relationships between joints are captured and used in the task of lifting the keypoints. Further, information from nearby frames can be used to encapsulate more information during 3D reconstruction of the human body skeleton.

Times series comparison
Dynamic Time Warping is a reliable and tested statistical method for comparing time series methods. However, recent advances in RNNs can be used to capture both low-level and high-level information from the sequence of poses for vector comparison. It can be concluded that while the developed project works well with the aims defined, there is still big scope for improving datasets and 3D human pose estimation.

Chapter 7

Reflection

Research
This project provides valuable insights into leveraging deep learning based 3D human pose estimation in the sports coaching domain. Methodologies for conducting research and building foundation for a research project is learned as part of completing this project. Different data collection techniques for image and video based datasets was learned while researching datasets. Additionally, the parameterisation of 2D human pose estimation and the challenges faced were studied. Further, existing neural network architectures for lifting pose to 3D coordinates is explored and the challenges for such architectures are explored. Finally, various techniques for time-series similarity comparison were learned. These learnings contributed to building a strong research foundation for the project.

Dataset Preparation
Various preprocessing skills for human pose estimation such as resizing and normalising keypoints were obtained as part of combining the ASPset and SportsPose datasets for the project. Further different representation of bounding boxes and pose keypoints are learned to easily convert keypoints as required.

2D Human Pose Estimation
The valuable skill of applying existing implementations of algorithms for fine-tuning on customised dataset was obtained during this project. Finally, data caching and augmentation techniques for optimising training are explored and implemented.

3D Pose Lifting
Network architecture design skills are assimilated as part of designing 2 novel architectures for 3D pose lifting. Further training strategies for training models from scratch are examined including appropriate weight initialisation for different activation functions and scheduling learning rate based on validation results.

Pose Comparison
The incredibly valuable skill of adapting a solution from a different domain is imbibed during the development of this module.

Challenges Faced
Sports Coaching is a niche domain in the context of human pose estimation. We overcame this challenge by adapting existing solutions with customised datasets and designing novel architectures for the same. The skills learned as part of completing this project would be

incredibly helpful for ensuring industry readiness during the next phase of my career.

References

[1] Ltd, Research and Markets. (2022) Global sports coaching market 2023-2027. [Online]. Available: https://www.researchandmarkets.com/report/sport-coaching#:~:text=The%20sports%20coaching%20market%20is,increasing%20demand%20for%20sports%20coaches.

[2] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[3] A. Nibali, J. Millward, Z. He, and S. Morgan, "Aspset: An outdoor sports pose video dataset with 3d keypoint annotations," *Image and Vision Computing*, vol. 111, p. 104196, 2021.

[4] C. K. Ingwersen, C. M. Mikkelstrup, J. N. Jensen, M. R. Hannemose, and A. B. Dahl, "Sportspose-a dynamic 3d sports pose dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5218–5227.

[5] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[6] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade: Second Edition*. Springer, 2012, pp. 421–436.

[7] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[8] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 993–13 000.

[9] M. S. Hossain, J. M. Betts, and A. P. Paplinski, "Dual focal loss to address class imbalance in semantic segmentation," *Neurocomputing*, vol. 462, pp. 69–87, 2021.

[10] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[11] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation," in *In Proc. CVPR*, 2013.

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

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

[14] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "Crowdpose: Efficient crowded scenes pose estimation and a new benchmark," *arXiv preprint arXiv:1812.00324*, 2018.

[15] J. Wu, H. Zheng, B. Zhao, Y. Li, B. Yan, R. Liang, W. Wang, S. Zhou, G. Lin, Y. Fu *et al.*, "Ai challenger: A large-scale dataset for going deeper in image understanding," *arXiv preprint arXiv:1711.06475*, 2017.

[16] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2248–2255.

[17] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *International Conf. on Computer Vision (ICCV)*, Dec. 2013, pp. 3192–3199.

[18] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "Posetrack: A benchmark for human pose estimation and tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5167–5176.

[19] W. Lin, H. Liu, S. Liu, Y. Li, R. Qian, T. Wang, N. Xu, H. Xiong, G.-J. Qi, and N. Sebe, "Human in events: A large-scale benchmark for human-centric video analysis in complex events," *arXiv preprint arXiv:2005.04490*, 2020.

[20] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation." in *bmvc*, vol. 2, no. 4. Aberystwyth, UK, 2010, p. 5.

[21] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," in *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. [Online]. Available: http://gvv.mpi-inf.mpg.de/3dhp_dataset

[22] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse, "Total capture: 3d human pose estimation fusing video and inertial sensors," in *2017 British Machine Vision Conference (BMVC)*, 2017.

[23] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International journal of computer vision*, vol. 87, no. 1-2, pp. 4–27, 2010.

[24] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.

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

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

[27] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[29] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4733–4742.

[30] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "Vitpose: Simple vision transformer baselines for human pose estimation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 571–38 584, 2022.

[31] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh, "Pose machines: Articulated pose estimation via inference machines," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*. Springer, 2014, pp. 33–47.

[32] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[33] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 2016, pp. 483–499.

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

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

[36] C. Jiang, K. Huang, S. Zhang, X. Wang, J. Xiao, and Y. Goulermas, "Aggregated pyramid gating network for human pose estimation without pretraining," *Pattern Recognition*, vol. 138, p. 109429, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320323001309

[37] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[38] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4903–4911.

[39] M. Kocabas, S. Karagoz, and E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 417–433.

[40] W. Mao, Z. Tian, X. Wang, and C. Shen, "Fcpose: Fully convolutional multi-person pose estimation with dynamic instance-aware convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9034–9043.

[41] X. Nie, J. Feng, J. Zhang, and S. Yan, "Single-stage multi-person pose machines," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6951–6960.

[42] Z. Geng, K. Sun, B. Xiao, Z. Zhang, and J. Wang, "Bottom-up human pose estimation via disentangled keypoint regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 676–14 686.

[43] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," *Advances in neural information processing systems*, vol. 30, 2017.

[44] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5386–5395.

[45] S. Jin, W. Liu, W. Ouyang, and C. Qian, "Multi-person articulated tracking with spatial and temporal embeddings," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5664–5673.

[46] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.

[47] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 977–11 986.

[48] J. Li, W. Su, and Z. Wang, "Simple pose: Rethinking and improving a bottom-up approach for multi-person pose estimation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 354–11 361.

[49] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[50] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[51] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[52] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Yifu), C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7347926

[53] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[54] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[55] C.-H. Chen and D. Ramanan, "3d human pose estimation= 2d pose estimation+ matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7035–7043.

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

[57] B. Tekin, P. Márquez-Neila, M. Salzmann, and P. Fua, "Learning to fuse 2d and 3d image cues for monocular body pose estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3941–3950.

[58] M. Wang, X. Chen, W. Liu, C. Qian, L. Lin, and L. Ma, "Drpose3d: Depth ranking in 3d human pose estimation," *arXiv preprint arXiv:1805.08973*, 2018.

[59] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, pp. 275–309, 2013.

[60] P. Tsinaslanidis, A. Alexandridis, A. Zapranis, and E. Livanis, "Dynamic time warping as a similarity measure: applications in finance," 2014.

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

Appendix A

Project Specification Form

The codebase for the project has been uploaded to `https://github.com/RJaikanth/dissertation`

MSc Project Specification Form

Version 03

Please complete this form in printed or typed text (Times New Roman size 11). A copy of the approved document will have to be included as appendix in the actual dissertation to help establish to which extend the project has been successful.

Section A

1. Project identification	Proposed dissertation title (maximum 15 words)	
	Enhanced Sports Training with Human Pose Estimation	
2. Student details	Name (*in full*)	
	Raghhuveer Jaikanth	
	e-mail address	
	raghhuveerj97@gmail.com	
	Term time address	
	ih818010@student.reading.ac.uk	
	Contact telephone number	
	7767976256	
3. Supervisor *Name and contact details of the staff member*	Name: Dr. Luis Patino	
	e-mail address	
	j.l.patinovilchis@reading.ac.uk	

4. The supervisor *The person identified in Section A4 hereby approves the dissertation specification*

Name	Dr. Luis Patino	
Signature		Date: 23-03-2023

Fill in section A5 and A6, if the project has industrial input.

5. Company Partner *Name of organisation involved in the project*			
6. *Details of contact person in the organisation involved in the project*	Title *(e.g. Mr/Mrs/Dr)*	Name	
	Address		
	Tel. No.	Fax No.	e-mail address

MSc Project Specification Form

Section B – Overall Programme

1. Background and Literature review

Please describe in the space provided the background to the project and write a short literature review highlighting relevant developments on the topic of the dissertation

Research Background:

Sports coaching plays a vital role in the development and success of athletes. Coaching is crucial for enhancing athletes' abilities, preventing injuries, and preparing them holistically for competitive sports. Without the guidance and support of coaches, athletes may face significant challenges in achieving their goals and maximizing their potential. Historically, coaching has been done in person with the student and coach being present for the session.

However, in the past 3 years, in-person training has been reduced due to pandemic and global lockdown. This period also saw a huge surge in the market of online education and teaching for theoretical fields like science and maths. However, there has not been many advancements in teaching physical activities such as sports online. This is due to the human resource constraint faced by schools and colleges for sports training and coaching. Educational institutions such as schools and colleges are not up to the task of providing quality sports coaching to each student due to human resource constraints.

Traditional methods in sports coaching using computers employ the use of body sensors to estimate the pose and movement of the human body. These sensors not only restrict user movement but are also not easily available and expensive for educational houses without financial backing. Recent advances in the field of deep learning and computer vision, particularly Human Pose Estimation using deep learning methodologies, can help solve this shortage. Human Pose Estimation can be used to analyse body movements by identifying and tracking key points on the human body, generally joints. This work proposes the development of an application to do Human Pose Estimation (HPE) using video input and predict correctness of technique in real time by comparing the body movements of the user with a reference video for soccer dead-ball techniques (freekicks and penalties).

Literature Review:

HPE is an active area of research in the fields of AI and Computer Vision, and there has been a considerable amount of work in improving the same. AlphaPose [1] is a deep learning-based pose estimation algorithm that utilizes Convolutional Neural Networks (CNNs) complemented by Pose Affinity Fields (PAFs) to accurately estimate human poses. YOLOPose [2] combines You Only Look Once (YOLO) with pose estimation to leverage a single neural network to simultaneously detect humans and their poses in a single pass. OpenPose [3] is a widely adopted open-source HPE framework that employs a multi-stage CNN in a bottom-up approach. BlazePose [4] is a lightweight and efficient HPE algorithm developed by Google. It employs a single-stage pipeline using a combination of regression and classification models to estimate human poses accurately.

Current research in the application of HPE in sports coaching focus more on indoor activities. Guo et al. [5] utilize a neural network to generate heat maps for pose estimation and dance capture, followed by post-processing and similarity measures, and implemented an interactive visualization tool. Choo et al created a web app that estimates human pose via video inputs, provides visual feedback for activities like Tai Chi [6]. Zou et al implemented an intelligent fitness trainer system based on human pose estimation, showing training courses and providing motion correction advice [7]. While majority of the work in using HPE for physical training has been done for indoor activities, there has also been some work on applying HPE to outdoor sports. Wang et al. proposed an AI coach system for personalized athletic training experiences with trajectory extraction, human pose estimation, and pose correction using abnormal detection and visual suggestions for sports videos with fast movement and complex actions [8]. Liu et al. use a novel motion analysis approach for assessing golf swing quality by comparing a user's swing to an ideal reference and provide real-time feedback to improve player performance using 17 skeleton points and a score from 0 to 10 [9].

SportsPose [10] is a new dataset that provides 3D key points from monocular images for soccer, volleyball, jump, baseball pitch, and tennis. ASPSet [11] is dataset comprising of over 100,000 dynamic, sports-related 3D poses. SoccerKicks [12] provides 3D reference movements of humans performing dead ball kicks (penalty and foul) obtained from

	reference videos suitable for use in the soccer domain.

MSc Project Specification Form

Section B – Overall Programme (continued)

2. Research question, justification and objectives *Please describe in the space provided the research question to be answered, justify why the topic is important at the present time, and describe the specific objectives of research against which your achievements will be measured.*	**Aim:** The key aim of the research is to investigate human pose estimation methods and tune them on the SoccerKicks data set. The project will attempt to create an application that can estimate the pose of the user using deep learning and comparing it with a reference video for the techniques of penalty kicks and free kicks in soccer. **Research Questions:** To achieve this, the following research questions have been created that will be answered as part of the dissertation – 1. What are the necessary techniques and body movements required to execute a free kick and a penalty kick? This includes input from a domain expert. 2. What are the current state-of-the-art models available for human pose estimation and what is their performance with respect to both runtime and accuracy for the sport application researched in this project? 3. What are the methodologies to compare two persons executing a sports sequence in real time and provide feedback? **Justification:** Online sports coaching presents a significant challenge in accurately assessing an athlete's progress due to limited access to non-verbal cues and data analytics. Coaches may rely on self-reported data, which can be inaccurate and unreliable, affecting the quality of feedback and guidance. To overcome these challenges, online sports coaching can incorporate HPE to enhance the assessment of progress and provide effective feedback. Usage of HPE can be used to automate the assessment of sports techniques by comparing with a reference video, thus reducing the load of the coach as well. **Objectives:** 1. Get a reference video with the correct technique through a domain expert. 2. Compare and identify models that are accurate and lightweight. 3. Train and develop a model on the chosen datasets and assess the models on run-time metrics and accuracy. 4. Identify and implement a pose comparison technique for comparing the referral and user video. 5. Develop a prototype application to predict the correctness of the user technique. 6. Evaluate the strengths and weaknesses of the developed model, identifying areas of improvement.

MSc Project Specification Form

Section B – Overall Programme (continued)

3. Methodology

Please describe the methodology that you will use to achieve the objectives stated in Section B2.

Data Splitting

The dataset identified as a part of literature review is open-sourced and available online. The authors of these datasets have already split the dataset into training, evaluation, and testing splits. For this project, the same splits are used for training, evaluating, and testing the data.

Data Pre-processing and Augmentation

As part of data pre-processing the following steps will be done –

1. Each image frame will be resized to 224x224 pixels.
2. We will normalise each image frame for faster training.
3. Add random jitter to frame to emulate real-life sensors.

Pose Estimation Algorithms

As part of the dissertation, the following algorithms will be tested and evaluated –

1. OpenPose
2. AlphaPose
3. YoloPose

These algorithms have been chosen for their fast inference times and high accuracy. We also transform the 2D key points to 3D using a simple single layer encoder-decoder feed forward network.

Loss Function and Evaluation Metric

// TODO

We would use the Object Key point Similarity to evaluate our 2D pose estimation model as proposed by [7]. To complement this, OKS similarity loss as proposed by [8] will be used. This way we can ensure that our evaluation metric is optimised.

Pose Similarity Methodologies

1. *Angle Comparison*
   This method compares 2 videos by comparing angles between different joints of the human body. Using input from a domain expert, we could identify the progression of angle between joints over time to create a reference video. However, this method requires the speed and length of the video be same which is not always possible.
2. *Dynamic Time Warping*
   Dynamic Time Warping method can compare two time series data with different time lengths. It can be used to calculate the similarity of 2 temporal sequences which may vary in speed. This is especially helpful when the reference and user video do not match in speed or length.

Tech Stack

The following technologies will be used –

1. Python
2. Bash shell
3. PyTorch
4. OpenCV

MSc Project Specification Form

Section B – Overall Programme (continued)

4. References

Please provide a list of references made in Sections B1, B2 and B3. The formatting of the references must comply with the Style Guide for Technical Reports and Academic Papers.

[1] "Fang, H.S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., Li, Y.L. and Lu, C., 2022. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. IEEE Transactions on Pattern Analysis and Machine Intelligence.," [Online].

[2] "Maji, D., Nagori, S., Mathew, M. and Poddar, D., 2022. YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2637-2646).," [Online].

[3] "Cao, Z., Simon, T., Wei, S.E. and Sheikh, Y., 2017. Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7291-7299).," [Online].

[4] "Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F. and Grundmann, M., 2020. Blazepose: On-device real-time body pose tracking. arXiv preprint arXiv:2006.10204.," [Online].

[5] "H. Guo, S. Zou, C. Lai, and H. Zhang, “Phycovis: A visual analytic tool of physical coordination for cheer and dance training,” Computer Animation and Virtual Worlds, vol. 32, no. 1, p. e1975, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi," [Online].

[6] "A. Tharatipyakul, K. T. W. Choo, and S. T. Perrault, “Pose estimation for facilitating movement learning from online videos,” in Proceedings of the International Conference on Advanced Visual Interfaces, ser. AVI ’20. New York, NY, USA: Association for Co," [Online].

[7] “J. Zou, B. Li, L. Wang, Y. Li, X. Li, R. Lei, and S. Sun, “Intelligent fitness trainer system based on human pose estimation,” in Signal and Information Processing, Networking and Computers, S. Sun, M. Fu, and L. Xu, Eds. Singapore: Springer Singapore, 20," [Online].

[8] "J. Wang, K. Qiu, H. Peng, J. Fu, and J. Zhu, “Ai coach: Deep human pose estimation andanalysis for personalized athletic training assistance,” in Proceedings of the 27th ACM International Conference on Multimedia, ser. MM ’19. New York, NY, USA: Associati," [Online].

[9] "J. J. Liu, J. Newman, and D.-J. Lee, “Body motion analysis for golf swing evaluation,” in Advances in Visual Computing, G. Bebis, Z. Yin, E. Kim, J. Bender, K. Subr, B. C. Kwon, J. Zhao, D. Kalkofen, and G. Baciu, Eds. Cham: Springer International Publish," [Online].

[10] "Ingwersen, C.K., Mikkelstrup, C., Jensen, J.N., Hannemose, M.R. and Dahl, A.B., 2023. SportsPose--A Dynamic 3D sports pose dataset. arXiv preprint arXiv:2304.01865.," [Online].

[11] "Nibali, A., Millward, J., He, Z. and Morgan, S., 2021. ASPset: An outdoor sports pose video dataset with 3D keypoint annotations. Image and Vision Computing, 111, p.104196.," [Online].

[12] "Lessa, N.M., Colombini, E.L. and Simões, A.D.S., 2021, October. SoccerKicks: a Dataset of 3D dead ball kicks reference movements for humanoid robots. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3472-3478). IEEE.," [Online].

MSc Project Specification Form

Section C – Social, legal and ethical issues

Describe Social, legal and ethical issues that apply to your project (If your project requires a questionnaire/interview for conducting research and/or collecting data, you will need to apply for an ethical approval).

Does your project require ethical approval?

Ethical Considerations

A public dataset of the specific domain of the research will be employed. Given the nature of the problem, the dataset contains faces of various people. However, there is no data that identifies a person using personal information. For example, no addresses or full names were included in the dataset.

Regarding legal issues, as this data is open-sourced and publicly available, there is no risk of copyright issues or licensing required.

Risk Considerations

Deep learning projects is very GPU intensive. If GPU is not available, training will take more than double the time. Poor code management can lead to messy, unreadable code reducing the reproducibility of the project. It is also necessary that the training pipeline is optimised and fast to ensure training is done on time.

Lack of research can lead to duplication of work. It can also lead to sub-par methodology and code being used for this project.

MSc Project Specification Form

Section D – Work plan

Task No	Task Describe each task to be undertaken and, where appropriate, indicate how it will be carried out. Please consider tasks between the submission of this specification until the submission of the dissertation in August.	Effort *(person weeks)*	Indicate deliverables and decision points (*e.g. data analysis completed, designs completed, software implemented, hardware procured, system configured,..*)
1.	**Establishing Research Background** • 2D HPE in a sports context • Pose Comparison Methodologies • Soccer dataset for HPE	**2**	• Establish background and scope of project • Document scope of project in proposal • Identify and download datasets for further use.
2.	**Understanding mechanics of freekick and penalty kick** • Identify a domain expert and conduct an interview to understand the mechanics of the techniques	**1**	• Understand movement of human body during execution of technique • Represent technique using pose features such as angle between limbs • Have a reference video ready for comparison
3.	**Literature Review** • Write a literature review that provides an overview of existing research into HPE for sports • Focus on recent papers using SOTA for HPE	**3**	• Understand and establish familiarity with current research • Development of literature review for the project • Identify the pose similarity methodology to use.
4.	**Implement a training pipeline** • Implement a generalised training pipeline for training and evaluating multiple models.	**1.5**	• Prototype of the generalised training pipeline using relevant tools including pre-processing and augmentation techniques • Publish work to Git repository
5.	**Training models** • Train models using GPUs for faster computing	**1.5**	• Trained models should be available for comparison • Metrics calculated and saved on the training dataset
6.	**Model evaluation** • Evaluate the performance of the trained models to find the best one.	**1**	• Evaluate models using the OKS metric for different body parts • Best model for each body part.
7.	**Implement pose similarity** • Create framework to score pose similarity based on the identified methodology	**0.5**	• Implement pose similarity method identified. • Best model for each body part

MSc Project Specification Form

Section D – Work plan (continued)

Task No	Task Describe each task to be undertaken and, where appropriate, indicate how it will be carried out. Please consider tasks between the submission of this specification until the submission of the dissertation in August.	Effort *(person weeks)*	Indicate deliverables and decision points (*e.g. data analysis completed, designs completed, software implemented, hardware procured, system configured,..*)
8.	**Evaluate pose similarity** • Evaluate the performance of the pose similarity method	**0.5**	• Evaluate pose similarity with help of domain expert to understand what can be improved
9.	**Compile results and Final Report** • Compile findings and author final report in given format	**1**	• Final Report in given format

MSc Project Specification Form

Section E - Time Plan

For each task identified in Section D, shade the months during which you will be engaged and mark deliverables and decision points.

Task No	February	March	April	May	June	July	August	September	Project Months
Task1									
Task 2									
Task 3									
Task 4									
Task 5									
Task 6									
Task 7									
Task 8									
Task 9									

MSc Project Specification Form

Section F – Costing

For efficient and fast training and evaluation of deep learning models, GPU compute resources would be required.

Item description	Cost (£) *incl. VAT*
Temporary lease of compute resources from AWS, approximately 300 hours of GPU compute time	400 GBP
Total	