Real-Time Pose Recognition for Billiard Players Using Deep Learning

Zhikang Chen

A project report submitted to the Auckland University of Technology
in partial fulfillment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2023

School of Engineering, Computer & Mathematical Sciences

Abstract

Human Pose Estimation (HPE) has long been an area of interest for computer vision researchers, the development of deep learning, in particular, has led to significant progress in the problem of recognizing dynamic and complex human poses. Transformer structure is very good at handling the problems related to time series. A billiard player's stroke has a beginning and an end, we can consider a complete stroke as a time sequence problem containing multiple video frames. This project combines human posture estimation with the Transformer for recognizing and analyzing billiard players' striking actions. The key point extraction transforms the player's striking action into key point coordinates with time series and inputted into the Transformer for real-time analysis. We optimize the structure of the Transformer model by using ablation experiments and finally achieved 98% accuracy and 0.02 seconds response time. Compared with the existing pose estimation, the proposed Transformer model can capture much complex movement and take the time into account. In addition, this method provides a much objective and accurate analysis tool for billiards game by improving the player's skills.

Keywords: Billiards posture analysis, human skeleton, key point detection, Transformer, deep learning

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature: Zhikang Chen Date: 8 October 2023

Acknowledgment

Fisrt of all, I would like to thank my parents for their finanical support. Owing to the unselfish and generous sponsor from them, I have this invaluable opportunity to complete my Master's study with the Auckland University of Technology (AUT), New Zealand.

I would also like to express my deepest gratitude to my supervisor Wei Qi Yan. In this study, he not only provided me with professional knowledge support and careful guidance, but also helped me enrich my learning experience. I believe I could not complete my study without Dr Yan's supervision and instructions. Meanwhile, I also appreciate the secondary supervisor Parma Nand and the school administrators of AUT for their invaluable guidance.

Zhikang Chen

Auckland, New Zealand

October 2023

Chapter 1
Introduction

This chapter is composed of five parts: The first part introduces the background and motivations, the second part includes the research question, followed by the contributions, objectives, and structure of this report.

1.1 Background and Motivation

In recent years, human pose estimation has been a popular topic in computer vision, especially with the development of deep learning techniques combined with human pose recognition to make it gain a broader application prospect. Deep learning has proven effective in analyzing diverse poses, complex scenes, and occlusion problems (Angelini et al., 2019). Thus, human pose estimation combined with deep learning provides accurate and automated human motion analysis in areas ranging from fitness instruction and medical rehabilitation to training and analysis of advanced motor skills (Illavarason et al., 2019; Cust et al., 2018).

Billiards is a game that requires players to perform high-precision, highly technical and strategic maneuvers, with a variety of complex body movements and slight postural variations designed into a player's stroke. For coaches and players, accurately identifying and analyzing errors in the striking motion is a critical factor in improving player proficiency and performance (Neuman & Gray, 2012). However, traditional manual analysis methods are time-consuming and laborious and may only identify some parts of the error. Then, a computer-aided analysis provides a new perspective to view and improve a player's skill and ability in billiards.

Human Posture Estimation (HPE) aims to identify the critical parts of the human body from an image or video and obtain their spatial arrangement. The primary method is to obtain the joint coordinate points of the human body to draw a skeletal model of the key points describing the human body. Initially, computer arithmetic and training data limited human pose estimation techniques, and early pose recognition methods formed the basis of classical computer vision techniques such as edge detection and geometric models (Ekvall, 2005). With the continuous optimisation of deep learning, especially when applying convolutional neural networks (CNN) and recurrent neural networks (RNN), significant improvements have been achieved in the accuracy of human pose estimation (Dang et al., 2019).

Transformer performs well in time series prediction and classification tasks (Tange & Matteson, 2021). Transformers provide a powerful ability to process time-series data, especially in capturing long-term dependencies. Combining human pose estimation and Transformer allows researchers to analyse the temporal relationships of movements while identifying each movement of a player, and optimised Transformer models have the potential

to process and identify sequential data of billiard players' striking motions, providing more meaningful feedback to coaches and players more accurately.

The hastened growth of deep learning and computer vision provides the opportunities for pose analysis, by using advanced gesture recognition techniques, a much objective and accurate assessment of athleticism is possible. It can effectively avoid the subjective errors of manual analysis. Billiards is a highly skill-demanding sport with unique action and timing factors, making further research and technique optimisation needed. Transformer is particularly well-suitable for capturing long-term dependencies in time series data due to its unique design. It is an ideal tool for analysing time-series data in billiards. While traditional machine learning models encounter difficulties processing time-series data, the optimised Transformer model may make another breakthrough for player stroke recognition due to its structure and features.

1.2 Research Questions

In this report, our main objective is to combine human pose estimation techniques and deep learning techniques to accomplish the recognition and analysis of billiard players' striking movements in real-time scenarios and to improve the effectiveness and accuracy of the recognition and analysis. Therefore, the main research questions of this report are:

- How can we efficiently combine human posture estimation and deep learning in analyzing the hitting posture of a billiard player to achieve accurate real-time action analysis and recognition?

In order to refine this research question, we can split it again:

- How can human pose estimation algorithms be adapted for optimal human pose resolution in billiard scenarios and player strokes?
- How can the deep learning model be optimised for best performance in player stroke action analysis?
- In player batting action analysis, which human joints are particularly critical for action recognition and analysis?
- When considering real-world applications, how can we ensure that the proposed technology framework is highly real-time for real-world training?

- How to evaluate to ensure that the current technology has significant advantages?

The core of this project is real-time pose analysis of billiard players. While choosing a human pose estimation technique, we must ensure that the real-time response speed meets the training requirements. Secondly, the hitting action has a temporal feature, so the deep learning model should be able to process the temporal data. We need to use the dataset for training to get the best result in the pose recognition based on deep learning and to conduct a comprehensive evaluation of the model.

1.3 Contributions

The focus of this research project is to realize the recognition and analysis of billiard players' hitting poses based on human posture estimation and time series deep learning. We propose a method that combines key point recognition and Transformer to realize the recognition and analysis of human movements in videos. We have developed a highly accurate, real-time and efficient system for recognizing and analyzing the hitting poses of billiard players. In order to improve the accuracy and robustness of the Transformer model in recognizing striking poses, we also built a billiard player striking pose dataset for training the model. By the end of this project, we were able to:

- Combine human pose estimation and Transformer for real-time billiard player striking pose recognition;
- Provide real-time multifaceted analysis and suggestions for striking poses using key point coordinates from human pose recognition;
- Optimize Transformer to achieve higher accuracy and efficiency in striking action recognition scenarios;
- Collect and create a billiard player striking pose dataset for training deep learning models.

In addition, the Transformer model will be optimized in this project by using ablation experiments and hyperparameter tuning. We collect model data under various conditions to find the optimal solution for the Transformer model. We also compare and analyze the methods employed in this project with other temporal models, such as GRU, RNN, and LSTM, to

delineate the advantages and disadvantages of each method.

1.4 Objectives of This Report

In this report, we outline a research methodology that integrates human posture estimation techniques with optimized Transformer models, which aims to identify and analyze a billiard player's stance in real time during a stroke. Additionally, we offer a comprehensive review of the contemporary literature surrounding human posture estimation methods and Transformer modelling techniques.

Subsequently, we introduce a novel human posture estimation system. This system discerns human body poses by extracting key points from image data and translating this data into linguistic information suitable for processing by the Transformer model. We map human key point data to a high-dimensional space, thereby creating a lexicon of key points interpretable as “utterances”. Leveraging the potent sequence processing capabilities of the Transformer, we can delve deeply into the analysis of human dynamics and poses.

Furthermore, we employ coordinate data to compute joint angles, facilitating the generation of a skeletal model. This model can be juxtaposed in real-time with a standard batting stance model, furnishing players with intuitive suggestions for movement adjustments. Incorporating deep learning methodologies enhances the system's prowess in performing highly accurate and effective posture recognition and analysis.

Therefore, the specific objectives of this report are twofold: Firstly, to employ human posture estimation techniques to extract key point coordinates, thereby facilitating the creation of a key point corpus, and to utilize these coordinates for computing joint angles, consequently generating a skeletal model available for real-time comparison; Secondly, to integrate an optimized Transformer model endowed with temporal and analytical functionalities to achieve real-time posture estimation. The development environments utilized for this endeavor are Python and PyTorch.

To comprehensively evaluate the merits of the proposed methodology, we undertake a comparative analysis involving the Transformer model and other prevalent temporal analysis models, aiming to discern the respective strengths and weaknesses of each approach.

1.5 Structure of This Report

The description of this report is as follows:

- In Chapter 2, we conduct a literature review, targeting human pose estimation and delving into research related to human pose estimation. Then, we focus on deep learning models with a temporal nature, especially the progress of methods such as Transformer. Finally, we cover the literature combining human pose estimation and deep learning methods.
- In Chapter 3, we present the specific methodology, experimental design, and comparison of expected results.
- In Chapter 4, we conduct experiments according to the designed methodology in Chapter 3 while collecting experimental data for detailed processing and analysis to evaluate the results visually. In addition, we have analyzed and discussed the advantages and disadvantages of different methods.
- In Chapter 5, we analyze the results and summaries the overall experiment.
- In Chapter 6, we look into the future and discuss possible future research directions and improvement strategies.

Chapter 2 Literature Review

The focus of this report is on pose capturing based on dynamic motion for deep learning, this chapter will introduce a plenty of traditional methods and the relevant knowledge of deep learning.

2.1 Introduction

As a cutting-edge research direction in computer vision, human pose estimation has shown great potentiality in multi-use application scenarios (Andriluka et al., 2014), such as medicine (Wu et al., 2020), animation production (Borodulina, 2019), and sports analysis (Badiola-Bengoa & Mendez- Zorrilla, 2021). Especially in the field of motion analysis, recognizing athlete's postures and analyzing them accurately, human posture estimation is not only effective in helping athletes improve their skills (Siddiqui et al., 2023) but also in preventing sports injuries during real-time analysis (Yu & Guo, 2022). Recently, we have witnessed the rapid evolution of deep learning technologies. Human posture estimation combined with deep learning has been the exploration direction. In particular, deep learning models that process temporal information, such as Transformer (Zheng et al., 2021) and LSTM (Lee et al., 2022), can improve the accuracy and effectiveness of human posture estimation and analysis.

In order to gain insight into the latest advances in the field, in this section, we will briefly review the relevant research literature. We will start with human posture recognition (Lin et al., 2023) and explore its technical advantages and limitations. Then, we turn into deep learning models with a temporal nature and explore their application in the motion domain (Rossi et al., 2021). Finally, we offer an overview of an approach that successfully combines human pose-aware estimation with deep learning models and discuss its performance and results.

2.2 Human Pose Estimation (HPE)

Human posture estimation aims to recognize and estimate 3D posture of the human body from digital images or videos (Wang et al., 2021). Before the development of deep learning techniques, human pose recognition mainly relies on traditional computer vision, and the primary approach was to manually extract features from images and adapt these features to design models for the prediction of key points of the human body. The related work (Dalal and Triggs, 2005) was a milestone (Dalal & Triggs, 2005), who proposed a feature descriptor called Histogram of Oriented Gradients (HOG), which captures an image's edge and texture information. HOG features are advantage for pedestrian detection and human gesture

recognition. Ramanan et al. further explored human gesture recognition based on part models further (Ramanan et al., 2007), which recognizes individual parts of the human body, such as arms, legs and knees., and then combines them to estimate the overall pose, which can deal with complex poses. However, the more is needed to solve the problem of multiplayer scenes and occlusion.

With the maturity of deep learning gradually replacing the traditional manual feature approach, it automatically learns from data and extract practical features, especially convolutional neural networks (CNN) that can efficiently process images, human pose recognition has developed rapidly. The relevant work (Toshev and Szegedy, 2014) is representative one (Toshev & Szegedy, 2014), which proposed a method called DeepPose that, for the first time, applied deep neural networks to human pose estimation.

DeepPose makes use of a convolutional neural network to extract key point coordinates from an image, which allows to the generation of a 2D skeletal model of the human pose. After that, Newell et al. proposed an Hourglass network (Newell et al., 2016), which has a deep network structure with multiscale and repeated connections, characterized by its ability to capture information at different scales in the image. Thus, the network extracts much accurate information about the key points of the human body, in the field, which adopt this structure for various human pose estimation. Deep learning methods for human pose estimation significantly improve the accuracy and can handle the challenges of multiple people, complex scenes and occlusions.

Sports analytics aims to assess and improve athletes through scientific methods, meanwhile computer vision can be a powerful tool to help in sports analytics, especially human pose recognition. For example, Parmar et al. investigated combining computer vision and deep learning for a professional assessment of diving, vaulting, and figure skating (Parmar et al., 2017), which takes use of a 3D convolutional neural network (C3D) to learn spatiotemporal features of the movement, followed by regressions of assessment scores through a combination of SVR and LSTM. Cao and Zhang took use of human body recognition technology to develop a system for evaluating poses for weightlifters (Cao & Zhang, 2019), which combines the FCN

network and CNN network, firstly removing the background interference in the weightlifting scene by FCN then completing the recognition and analysis of poses by CNN, which can efficiently recognize the poses and prevent possible injuries caused by movement errors.

In recent years, under the impetus of deep learning, human pose recognition has made significant progress according to the different methods, which can be roughly divided into two categories: Top-down and bottom-up. The top-down approach firstly performs human body detection on the whole image, recognizes the bounding box of each human body in the image, and then performs key point detection on the human body in each bounding box (Xiao et al., 2018). This method requires two steps to realize the result, so the computational complexity of the model increases, and the advantage is that this approach better recognizes individual key points in multi-person scenarios.

He et al. proposed the model Mask R-CNN that can perform accurate human body segmentation (He et al., 2017), which builds on the Faster R-CNN by extending it with an additional Region of Interest (RoI) Align process after generating a binary mask. The model can localize the human body's key points more accurately after accurately segmenting the body.

Unlike the top-down approach, the bottom-up approach firstly detects all possible key points of human body and then composes them into different human skeleton models by using connections between them. The advantage of this method is that it is fast, at the same time, which can be better applied to real-time scenarios. Cao et al. proposed a Part Affinity Field (PAF) for associating detected body parts with individuals in an image (Cao et al., 2017), which encodes the contextual global to realize real-time multi-person pose detection.

In this research project, human pose recognition mainly is use of MediaPipe framework from Google Research, which serves the purpose of multimedia processing, including target detection, image segmentation and human pose estimation, where the primary part for human pose estimation is BlazePose. The main advantage of this software is that it is lightweight and provides real-time and accurate human pose estimation. The model takes use of a top-down strategy to determine the position of the entire body and then determine the key points of human

body, so the model contains a detector to provide a bounding box, and then the area inside the box is further processed which can predict the position of up to 33 key points. Figure 2.1 shows the human skeleton model of the 33 key points predicted by using BlazePose. These key points cover the major areas from the head to the feet, so BlazePose can provide more accurate gesture recognition and analysis in analyzing the striking motion of a billiard player.

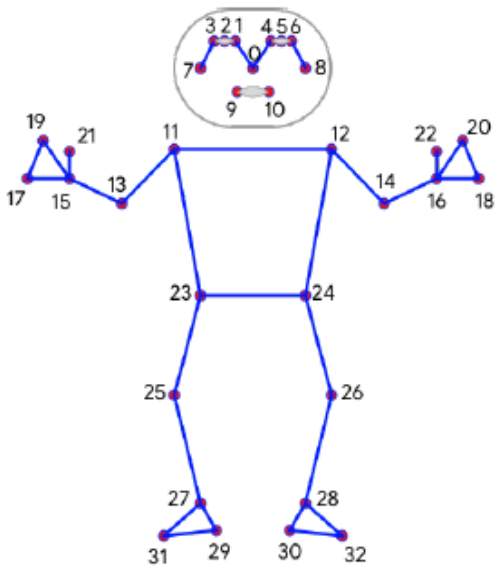


Fig 2.1 The key point skeleton extracted by using BlazePose

BlazeBlock is the core component of this model, a particular type of convolutional block that employs a depth-separable convolution strategy (Bazarevsky et al., 2020), which divides standard convolution operation into deep convolution and pointwise convolution, where deep convolution performs a convolution operation on each of the input channels, while pointwise convolution, a 1×1 convolution operation, combines these channels. Additionally, BlazePose employs convolutional downsampling and max pooling, which allows the model to capture a more extensive range of features and improve robustness (Gholamalinezhad & Khosravi, 2020).

2.3 Deep Learning Model

Unlike traditional feedforward neural networks, Recurrent Neural Networks (RNNs) belong to

a class of neural networks that process sequential data and stand out because they memorize and leverage their internal state to handle time series data (Agrawal & Sharma, 2022). This internal state assists the network to capture temporal dependencies in the data, thus demonstrates superior performance in various time series prediction tasks such as speech recognition, text generation, and action analysis (Hori et al., 2018; Abujar et al., 2019; Cui & Chang, 2020).

Simple recurrent neural networks were firstly introduced in 1990 (Elman, 1990), recurrent neural networks (RNN) provide a new approach for subsequent temporal data processing. In the following years, RNNs achieved great success in speech recognition. Graves et al. demonstrated the effectiveness of Deep Bidirectional RNN (DBRNN) in recognizing different dialects of the TIMIT corpus in 2013 (Graves et al., 2013). The primary outcome was the successful introduction of recognition error rates using DBRNN, combined with Connectionist Temporal Classification (CTC) to deal with variable-length output sequences, allowing the model to output text directly from the raw sound data and providing an essential baseline for subsequent research.

However, though RNNs can handle temporal tasks, they still have inherent limitations concerning the sequence length of the temporal data. RNNs experience difficulties learning tasks with long temporal relationships, mainly due to the gradient vanishing and explosion problems. This problem was initially investigated (Hochreiter, 1991; Bengio et al., 1994). In order to overcome the long-term dependency problem, Hochreiter and Schmidhuber proposed a variant version of RNN-based Long Short-Term Memory Networks (LSTMs) in 1997 for resolving the encountered gradient vanishing and explosion problems (Hochreiter & Schmidhuber, 1997). Initially, LSTMs existed only with the concepts of input and output gates until 2000, before Gers introduced the concept of forgetting gates and integrated them into the overall structure of LSTMs (Gers et al., 2000), an improvement that significantly improved the memory capacity of LSTM units, allowing them to learn and forget information over a more extended period.

LSTM controls the flow of information by introducing the structure of “gates”, which

allows the model to learn and memorize information efficiently over long sequences of time. Figure 2.2 shows the overall inputs and outputs of a standard LSTM module.

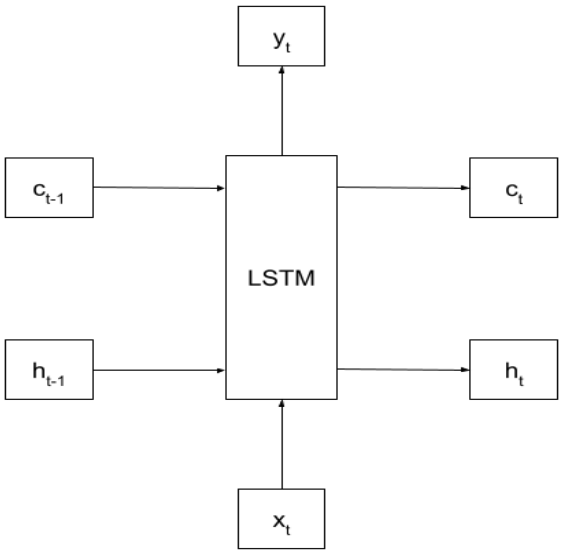


Fig 2.2 External schematic of the LSTM module.

Each LSTM cell receives three inputs at each time step: The input x_t of current time step, the hidden state h_{t-1} of the previous time step, the cell state c_{t-1} of the previous time step, and then the output y_t of the current round, the stateful output h_t of the current round, and the internal structure of the LSTM compute the cell state c_t for the current round. The internal structure of the classical LSTM consists of three gates.

The exisiting architecture of LSTM encompasses three gates: Forget, input, and output. The forget gate determines which information to discard from the cell state. The input gate decides what information is updated. The output gate decides, based on the current cell state and the input values, that the value of the hidden state is sent to the next time step as part of the LSTM inputs (Yu et al., 2019). In addition to the three gates, LSTM has a cell state component as the core structure for storing long-term information, which is updated, forgotten, and controlled by the three gates.

In deep learning, LSTM is employed as a special RNN to solve the long-term dependency

problem in long-sequence learning. To be precise, the core of LSTM is its cell state, the three critical gating mechanisms allow him to tune the information flow finely.

Considered the candidate update of a unit state, given a weight matrix W, the hidden state h_{t-1} of prior time step and the input value x_t of the current time step, we get the candidate unit state z,

$$z = Tanh(W * [h_{t-1}, x_t]) \tag{2.1}$$

Next, we compute the forgetting gate f_t, which determines which information is forgotten or retained from the cellular state.

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

Subsequently, the input gates i_t decide which information to update in the unit state.

$$i_t = Sigmoid(W_i[h_{t-1}, x_t] + b_i) \tag{2.3}$$

Combining the above calculations, we can update the unit status.

$$z_t = f_t * c_{t-1} + i_t * z \tag{2.4}$$

Finally, the output gate is responsible for deciding the value of the next hidden state h_t.

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

$$h_t = o_t * Tanh(z_t) \tag{2.6}$$

The value of each gate is normalized using a sigmoid activation function with its output ranging in [0,1.0]. If the output approaches to 0, the system nearly forgets the information. If the value tends to 1.0, the system retains the information. In the process of computation carried out by an LSTM unit, we can form an LSTM deep learning network by connecting multiple LSTM units in series as shown in Figure 2.3.

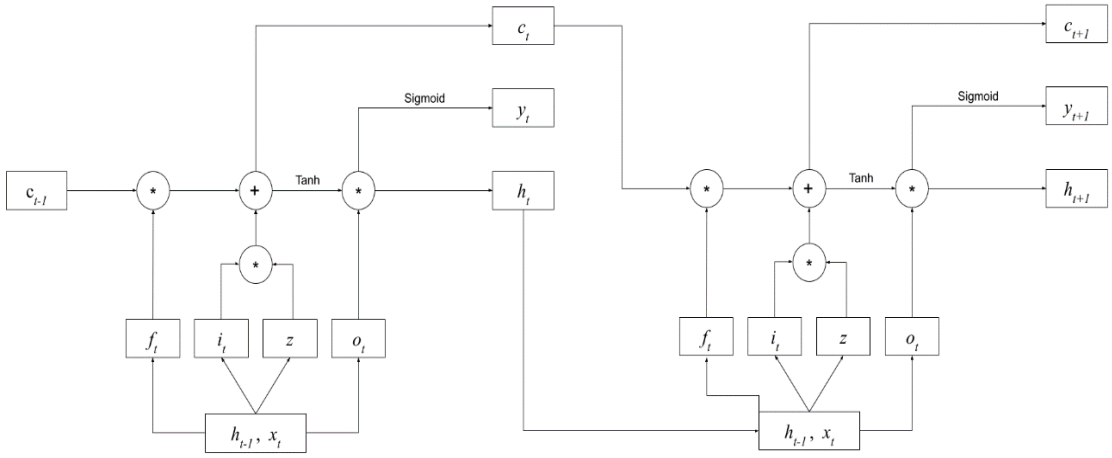


Fig 2.3 The internal structure of LSTM network in deep learning.

LSTM has received much attention since its introduction, the model has been applied to various sequence prediction tasks and produced compelling research results. Sutskever took use of a multilayer LSTM model to map the input sequence to a fixed dimensionality vector (Sutskever et al., 2014). Then, another deep LSTM model is adopted to decode the target sequence from the vector. The method was tested for decoding target sequences on the WMT-14 dataset. Its machine translation got a BLEU score of 34.8, scored higher than traditional translation methods. This demonstrates the strong performance of LSTM on sequence-to-sequence tasks such as machine translation.

Martin et al. created a word-level automated audio-video emotion recognition method based on LSTM (Martin et al., 2013), with the SEMAINE database and compared the emotion recognition scores of Audiovisual Sub-Challenge participants, finally obtained the best average recognition performance of the LSTM-based emotion recognition system, which demonstrated the research value of LSTM in the field of human emotion recognition. Pfeiffer et al. probed a new LSTM-based model whose primary function is to learn human motor behavior from data (Pfeiffer et al., 2018). The model incorporates static obstacles and introduces a method based on 1D grid coding in polar angle space to predict the trajectories of surrounding pedestrians, demonstrating high accuracy in more densely populated scenarios. This also demonstrates the effectiveness of LSTM in pedestrian motion analysis.

As the field of deep learning grows, a slew of new models and architectures were proposed to optimize the structure and performance of LSTM. In 2005, Graves and Schmidhuber firstly proposed a bi-directional LSTM for frame-by-frame phoneme classification (Graves & Schmidhuber, 2005). Compared to traditional structures, bi-directional LSTMs contain a forward layer and an inverse layer, for each time step, there is a combination of forward and inverse outputs.

Owing to this unique design, bi-directional LSTM can tackle both past and future contextual information, and at the same time, it performs better than RNN, MLP, and standard LSTM. Chung et al. introduced the Gated Recurrent Unit (GRU) concept in 2014 (Chung et al., 2014), which is a network design based on the gating mechanism of the LSTM, compared to LSTM GRU has two unique gating structures update gate and reset gate. The update gate memorizes the content, while the reset gate forgets previous memories, so the structure of GRU is simpler than LSTM regarding training speed and more minor data requirements. However, LSTM is much effective in handling complex tasks. In addition, Bahdanau et al. firstly proposed an RNN decoder with an attention mechanism (Bahdanau et al., 2014), which allows the model to focus its attention on a specific part of the target sequence as it generates each element of the target sequence. Many consider this research pioneering work integrating attention mechanisms with deep learning.

Vaswani et al. introduced the Transformer architecture in 2017, marking a departure from previously dominant architectures based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) (Vaswani et al., 2017). Distinctively, the Transformer architecture eschews temporal recursion and convolutions, leveraging instead a self-attention mechanism to enhance performance in sequence-to-sequence learning tasks, as shown in Figure 2.4. This design alleviates the gradient vanishing problem often encountered in RNNs, attributed to the absence of recursive connections in its architecture.

Moreover, it introduces parallel processing for all positions in a sequence during training, thus accelerating the learning process significantly compared to its predecessors. This was a pivotal breakthrough, bypassing the training speed constraints inherent in recurrent

frameworks. The seminal work by Vaswani et al. paved the way for the development of high-impact deep learning approaches, establishing the Transformer as a cornerstone in natural language processing (NLP) and other deep learning endeavors. Subsequent innovations have spawned powerful derivatives such as BERT (Devlin et al., 2018) and GPT (Brown et al., 2020), which have further underscored the potency and efficiency of the Transformer architecture.

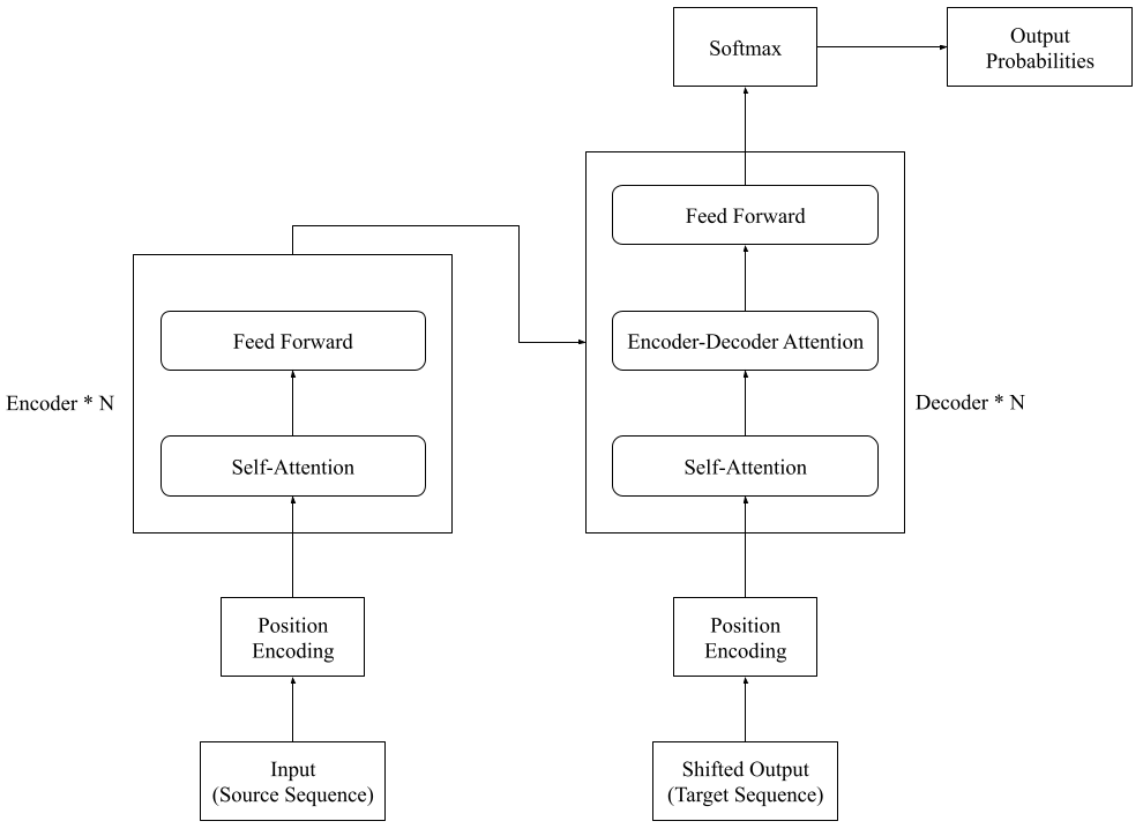


Fig 2.4 The structure of Transformer model

The Transformer employs a structure typical of Seq2Seq tasks, comprising an encoder and a decoder (Liu et al., 2018). The role of encoder is to encapsulate the input sequence information and encode it into a fixed contextual representation. The decoder produces a variable output sequence using the representation from the encoder, taking into account the output from the previous time step. In the Transformer architecture, the encoder typically comprises a self-attention mechanism module and a feedforward neural network, with each module followed by a residual link and a normalization layer. The decoder's structure mirrors the encoder's structure, except it incorporates an encoder-decoder attention module between

the self-attention module and the feedforward neural network. The primary function of this module is to enable the decoder to get the encoder's output, which emphasizes on the pertinent segments of the input sentence. The Transformer also integrates positional encoding, a method enables the self-attention mechanism to assimilate positional information within sequences, which ensures the correct order of this sequence is considered. In Transformer, the encoder and decoder are usually multilayered and ultimately output probabilistic results via Softmax.

The self-attention mechanism, a core component of the Transformer model, enables the assignment of distinct weights to each unit in the input sequence and recombines these inputs to form a new vector. This mechanism enables the Transformer to discern long-term dependencies within the sequence. The initial phase in computing self-attention entails generating three vectors, namely Query, Key, and Value, for each word vector in the input encoder, with each vector corresponding to a weight matrix. The Query, Key, and Value are derived by multiplying their respective word vectors with these matrices.

The second part of the mechanism is to compute the attention score. While computing the attention score for a given word, say A, it necessitates calculating a score for every other word in the sentence relative to word A. These scores determine the attention of each word when encoding word A. The attention score in the self-attention process is computed using the dot product between the Query and Key vectors, indicating the association degree between the Query and each Key. Attention scores undergo division by a factor and normalization through a Softmax layer, stabilizing the gradient.

In the third step, the Softmax output serves as weights, which accentuates positions with higher final scores when applied to and summed with the Values. We assume that the vector Query is Q, the vector Key is K, the vector Value is V, and d_k is the dimension of the key vector. We can get the formula for calculating the result z of self-attention.

$$z = Softmax(\frac{Q * K^T}{\sqrt{d_k}})V \tag{2.7}$$

Transformer has attracted much attention from researchers due to its efficient performance,

and many better Transformer architectures have been proposed based on the original model.

BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model based on Transformer and bi-directional encoder representations. BERT innovatively uses a masked language model for pre-training and in the GLUE and MultiNLI tests (Devlin et al., 2018). GPT (Generative Pre-trained Transformer) uses a different pre-training approach than BERT, which uses an unsupervised approach to train the language model, followed by supervised fine-tuning. The model has 175 billion parameters and performs well in translation, quizzing, and completing the blanks. (Brown et al., 2020). DistilBERT is a streamlined version of BERT that utilizes knowledge distillation in the pre-training phase and decreases the dimensions of the initial model by 40% while retaining 97% of the functionality, in addition to increasing model speed by 60% (Sanh et al., 2019).

2.4 HPE with Deep Learning

As deep learning swiftly advances across various domains, computational methods have begun to continuously combine deep learning with traditional video and image processing, human posture estimation (HPE) is one of them. Transformer, as a deep neural network capable of processing and understanding time sequences, has already demonstrated its superior performance in various fields, the potential benefits of combining deep learning and HPE while processing video and images have been demonstrated. Acquiring features from the contextual information of consecutive frames in processing video or real-time picture tasks is crucial (Donahue et al., 2015). In video analysis, there is rich contextual information between consecutive frames, the Transformer model learns long-term dependencies in the video, so it can effectively understand the context dynamics between front and back frames, thus making HPE prediction more accurate.

The visual Transformer (ViT) guides the Transformer to perform image processing tasks. The main idea of the model is to divide the image into fixed-size blocks and treat these blocks as words in a text sequence. Each image region is linearized into a fixed vector representation, and by position coding, the ViT can recognize the relative position of individual blocks in the

image. Furthermore, a few of experiments have demonstrated that Transformer can outperform traditional CNN models in image classification tasks with sufficient data and resources (Dosovitskiy et al., 2020). Mao et al. proposed a regression human posture estimation framework based on Transformer, which treats human posture estimation as a sequential problem utilizing an attention mechanism to focus the model on the most relevant features at the target key points, which exploits the structured relationships between key points thereby improving performance and avoiding the feature misalignment problem of regression-based methods, which was demonstrated to be effective based on the COCO dataset that Transformer significantly improves the state-of-the-art of regression pose estimation (Mao et al., 2021).

Swin Transformer is designed to be used as a generalized backbone model for computer vision, which is a new visual Transformer. To process visual data more efficiently, Swin Transformer makes use of a hierarchical Transformer structure and a displacement window approach, and the model is also effective in resolving differences in the size of visual entities. Swin Transformer has demonstrated excellent performance in high-resolution tasks like image segmentation and object recognition. In addition, the model outperforms the previous state-of-the-art in the COCO and ADE20K datasets (Liu et al., 2021).

In addition, other deep learning models, such as LSTM, are also broadly employed in human pose estimation. Fragkidaki proposed that LSTM can take historical information into account and assist the model avoiding misjudgments based on a single frame (Fragkiadaki et al., 2015). An Encoder-Recurrent-Decoder (ERD) model is proposed for performing human poses in video real-time analysis, which combines RNN and LSTM by introducing a nonlinear encoder and decoder before and after the loop layer, ultimately, the ERD can predict the displacement of the body joints in a 400-millisecond timeframe.

Furthermore, LSTM can provide the ability to analyse spatial and temporal information for the HPE task. Song, et al. proposed an HPE model constructed based on LSTM and RNN in 2017 (Song et al., 2017), which contains an end-to-end attention mechanism for paying attention to temporal and spatial information, and ultimately, the model can recognise the human being from skeletal data Action. For the HPE task, the fusion of spatial and temporal

attributes plays a critical role.

LSTMs provide strong support for effectively analysing these two types of information in videos. Nie took use of a two-level hierarchical LSTM network combining a 2D pose skeleton and a localized image to predict depth information (Nie et al., 2017). The model contains two LSTMs performing different tasks: Firstly, the skeletal-LSTM learns depth information from global human skeletal features; Secondly, the patch-LSTM learns local image features around critical locations. The main contribution of this work is the proposed two-level LSTM framework to accomplish the monocular 3D human pose estimation task.

In summary, human pose estimation combined with deep learning model provides new perspectives on motion analysis, where modern HPE methods provide accurate and fast global key point data for human movement recognition and analysis. Transformer can capture temporal dependencies in the sequence of key point data and allows for accurately recognising complex movements and variations in high-precision and strategic sports. The two techniques can be highly complementary and fully utilise each other's strengths and improve the effectiveness and performance of existing systems and pointing to a meaningful development direction for sports pose analysis.

Chapter 3 Methodology

The main content of this chapter is to clearly articulate research methods, which satisfy the objectives of this report. The chapter mainly covers the details of research methodology for video dynamic detection using deep learning which will be clearly introduced with the confident and imaginative use of the feature description methods.

3.1 Transformer

In recognizing and analyzing the pose of a billiard player, the hitting pose is a sequence of human movements with temporal information from setting up the stance to complete the striking and enable the computer to learn the complete hitting action of the player. We need the model to have the ability to learn temporal dependencies in action data. Transformer has a significant advantage in handling temporal tasks with its unique self-attention mechanism. Compared to traditional recurrent neural networks, Transformer is very efficient in solving the gradient problem by allowing the model to focus directly on certain parts of the input sequence without having to process the sequences in a strictly sequential manner, which provides a global approach to capturing information that can more easily capture long-distance dependencies.

In our research project, we employ a model inspired by the foundational Transformer architecture. Initially, the model undergoes a convolutional layer, targeting the extraction of local features. Subsequently, max pooling is applied to down-sample and condense these features, enabling the model to assimilate broader contextual nuances. This is followed by a fully connected layer that seamlessly fuses the feature vectors and reshapes them to dimensions appropriate for the Transformer's input.

Before entering the Transformer's encoder-decoder loops, the data is enriched with positional encodings, ensuring the model is attuned to sequential patterns within the data. The final output from the Transformer is passed through another fully connected layer, which generates probabilistic outcomes in conjunction with a softmax activation. The overarching architecture of our Transformer model is detailed in Figure 3.1. We implemented the entire model using the PyTorch framework. The Keras framework's drawing utility was leveraged to represent the model's structure for visualisation purposes.

We preprocessed the input vectors and planned them into a specific format, i.e. (number of samples, number of time steps, feature dimensions). The number of features comes from analyzing the extracted key point information. The output of the BlazePose model contains 33 coordinate points of critical parts of human body, each coordinate point is a (x, y, z) coordinate containing depth information. Then, we can get the number of features needed as 3×33. So, in the input of model structure, we have data in the format of $(None, 75, 99)$, where

none is the number of uncertain data samples. The number of times step 75 is calculated from each billiard player striking video we collected.

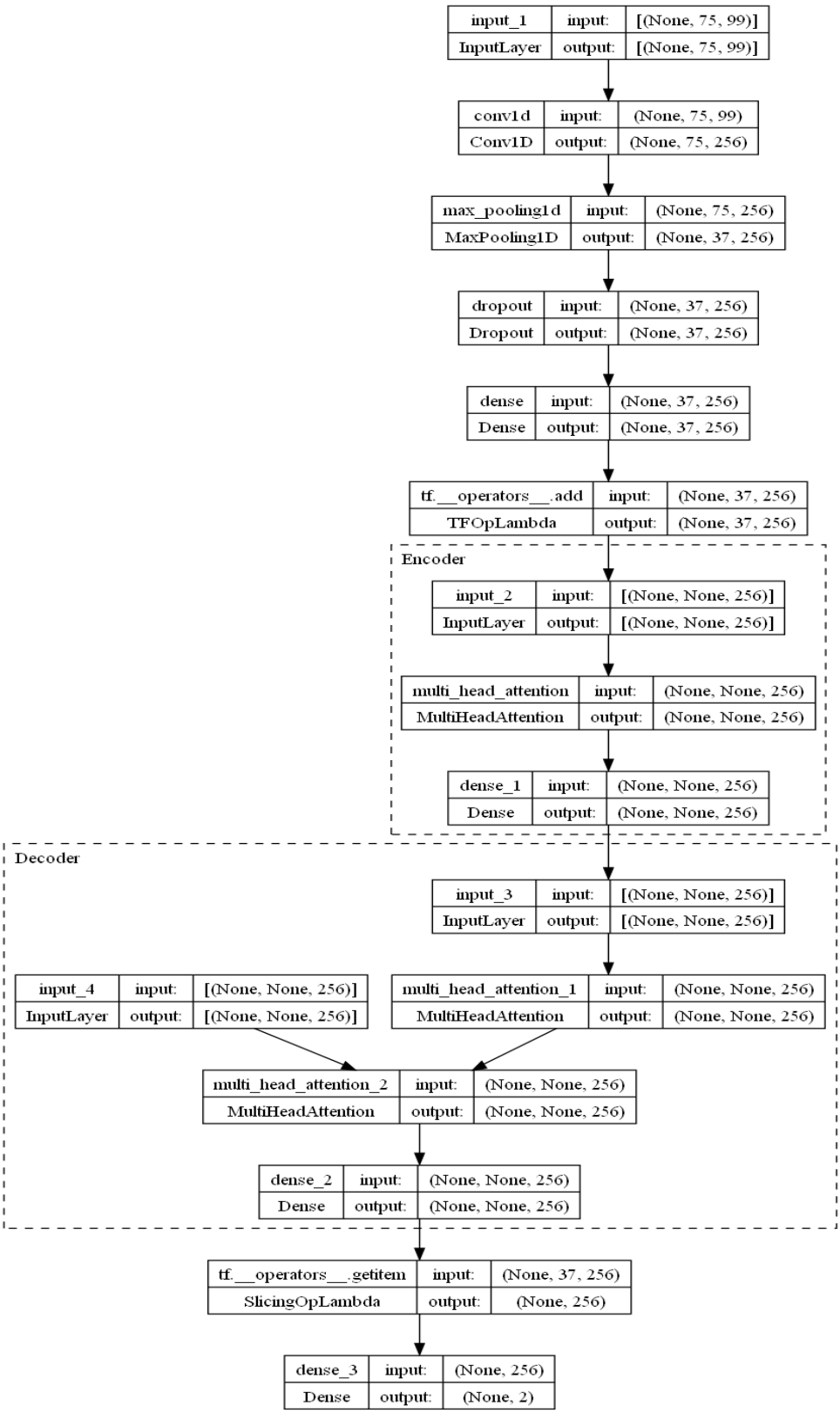


Fig 3.1 Transformer model structure

The data goes through a one-dimensional convolutional layer before feeding the data into the model. This layer primarily functions to learn the local features of the data. Since the convolution kernel slides through the input sequence, this helps to share the parameters throughout the sequence, thus improving the generalisation ability of this proposed model (Bai et al., 2018). Subsequently, a maximum pooling operation is performed on the data to reduce its dimensionality and improve training efficiency (He et al., 2016).

Thus, after maximum pooling, the time step of the data is reduced to general. Since the completion time of the striking pose is very short and the complete striking action can be completed in a shorter time step, this pooling operation can reduce the data dimensionality without too much impact on the model's action learning ability. After the data is pooled, it is processed through the fully connected layer. The processed data is added with positional coding, which helps the model learn the positional features of the data. These features feed the data into encoders and decoders for further processing.

The size of these hidden units defines the dimension of the feature space for each self-attention layer in each Transformer (Vaswani et al., 2017). The self-attention mechanism allows the model to learn long-term dependencies by using different weights at different time steps, which allows the output of each time step to be based on the entire sequence and the Transformer to process the entire sequence in parallel. Stacking multiple self-attention layers allows the model to learn more complex and advanced features. Thus, larger hidden units allow the model to store more contextual information but also lead to overfitting of the model (Jozefowicz et al., 2015). Conversely, smaller hidden units may result in the model needing to learn more contextual information, leading to a lack of accuracy. In the actual model structure, we set the number of hidden units to 256, which was examined to confirm the model's accuracy without triggering overfitting.

In Transformer, the number of layers refers to the number of self-attentive and feedforward neural network layers inside each encoder or decoder. Each layer outputs a sequence that the following layer processes. A multilayer network structure captures data features at various levels. For example, the first layer only captures low-level features, while deeper layers may capture more in-depth feature information (Kaplan et al., 2020). Thus, the benefit of a multi-

layer structure is to increase the ability of the proposed model to process complex data.

However, too many layers can lead to a more complex model structure and consequent overfitting. In addition, an increase in the number of layers also leads to an increase in training and inference time. Therefore, the setting of the model layers can significantly affect the training and inference results of the model. The effect of this proposed model in real-time detection needs to be considered comprehensively in the actual experimental scenarios. The model achieves the best inference speed and accuracy with two layers.

A normalization layer follows each self-attention module and feedforward network in the Transformer (Vaswani et al., 2017). Unlike image data, time series data has various features with different dynamic variations, and normalization stabilizes these dynamic variations. It does not provide a consistent active range of neurons, which makes it easier for the model to capture action patterns in the sequences to alleviate the problem of prolonged dependency. Next, normalizing the size of the input features can speed up model convergence, a significant feature that can reduce the number of iterations for the model to reach peak performance (Cooijmans et al., 2016). For time-series data containing a variety of noise, normalisation enables the model to remain robust to these noise variations.

Overfitting is the term to describe the phenomenon in which a model performs well on training data but performs poorly on never-before-seen data. To improve the generalisation ability of the proposed model and prevent overfitting, we add a dropout layer to the model. *Dropout* is a regularization that randomly discards a portion of neurons during training to generate a slightly different model at each forward propagation (Ozgur & Nar., 2020). This mechanism transferred a previously single network into a collection of different network versions, which makes the model not overly dependent on any particular feature in the training data, thus improving the generalisation ability.

The final layer of this proposed network is a fully connected layer, which is functionally different from the fully connected one described above and whose primary purpose is to integrate all the features of the previous layers and transform them into a specific predictive output. In this report, the task of the Transformer is to predict the classification of a billiard

player's striking posture, so the layer also changes the data dimensions to output a value for each classification, which is turned into more intuitive probabilistic data by a Softmax function.

In summary, our Transformer model adds 1D Convolution, 1D max-pooling regularization to the basic Transformer modules and improve the efficiency and generalization of the model and avoid the overfitting phenomenon. In addition, to evaluate the model in more detail, we will conduct ablation experiments on the proposed Transformer model to determine each module's effect on the model and find the optimal combination and hyperparameters.

3.2 Ablation Experiment

Ablation experiments are a classic experimental method in deep learning that explores the importance of each component in a model by eliminating a component in the complete model and evaluating the change in model performance. The experiment helps optimize the model and identify the structure of the top-performing model. Zeiler and Fergus visualized and understood the features of convolutional neural networks through ablation experiments (Zeiler & Fergus., 2014), inverse convolutional networks are employed to map the activation of the network back to the pixel space in order to find out the role of the features. Typically, R-CNN was improved by using ablation experiments to obtain faster real-time response (Sun et al., 2018).

In this report, we also show the importance of each component by ablating different components in the whole model and finding the best-performing model structure composition by replacing different methods. We split the entire ablation experiment into five steps.

Step 1. Initialization of the evaluation model: Firstly, we establish a baseline and compare the model performance with the baseline model to determine the advantages and disadvantages of each component after the ablation operation. In the baseline model in this ablation experiment, convolutional pooling and regularization are performed, and then the self-attention mechanism is computed. We set the number of hidden units to 256 and the number of layers to 2. In addition, the fully connected layer in the final output of the model needs to ensure that the output result is the correct number of classifications so it is not involved in the ablation experiment.

Step 2. Ablation operation: The model is ablated one component at a time, and the same method is utilized to replace or remove the layer in the ablation of the model structure, e.g., in the ablation experiments, we remove the convolutional layer. Then, we analyze the results to determine the impact of the convolutional layer on the model's performance. Table 3.1 shows all of the combinations in this ablation experiment, which shows the names of the ablation components, the units of the baseline model, and the technique variations used for each ablation.

Step 3. Evaluation Criteria: The task undertaken by the Transformer model in this report is to recognize the striking pose of a billiard player. In real-world scenario, we perform a binary classification task based on the completion of the action, and the model ultimately predicts the correctness and incorrectness of an action. Therefore, for the evaluation criteria, we are use of the loss value, accuracy and F1-Score to evaluate the performance comprehensively.

Step 4. Comparative performance: The optimal combination of structures is identified by comparing the model's loss, accuracy and F1-Score with the baseline model's parameters after each ablation.

Step5. Iterative process: Ablation, evaluation, and comparison are the standard process for conducting an ablation experiment, and we need to iterate on this process to ensure that each component and parameter of our design completes the ablation experiment to optimize the model structure.

Table 3.1 List of the components in ablation experiments

Model Component	Baseline	Ablation 1	Ablation 2
Model Dimension	256	96	512
Num Layers	2	6	10
Model Structure	Conv and Pooling	Only Encoder	Full Transformer
Activation Function	None	Swish	Mish
Multi Attention Heads	8	4	16

In order to better understand the difference of the optimized Transformer model in this billiard player hitting pose recognition, we also conducted a series of comparative experiments by comparing the optimized Transformer with LSTM and RNN, by which we hope to

determine whether the optimized structure is better suited to our task as well as to provide valuable insights and directions for future research.

3.3 Key-Point Skeleton Model

Billiards is a combination of skills that relies not only on a player's intuition and skill but also on his or her posture. Right posture can stabilize the body and ensure the stability of cue stick to hit the expected part of billiard ball to help the player achieve improved performance. Correct posture not only helps improve the performance level but also prevents sports injuries caused by wrong posture.

The purpose of this report is to identify and analyze the striking posture of billiard players in real time. The key point extraction in human posture estimation can accurately obtain the coordinates of the body's main joints in the posture. Through the relationship between the body parts, we connect these key points and draw them into the skeletal model of human posture.

We take advantage of this method to propose a real-time comparison method with standard hitting postures. The method is divided into two parts: One for modelling the standard posture skeleton and the other for modelling the posture skeleton detected in real time. For the modelling of standard posture, we collect posture data from professional billiard coaches beforehand and obtain the joint coordinate points of the professional player's striking posture through the key point extraction technique, then connect and model them. The method is similar to the standard posture model for real-time detection bone modelling. We draw the real-time skeleton from the coordinate point data acquired by the camera so that the user can adjust human movements by placing human posture bones similar to the standard skeleton in actual use.

For beginners or intermediate players, timely detection of errors in their posture is the key to improving their billiard game, and this real-time comparison of skeletal models helps players to correct their mistakes immediately instead of developing bad habits. In addition, this approach allows players to get feedback on each shot to master the correct hitting posture quickly.

In acquiring bone models through key points, we found that the bones detected in real-

time would not match the size and position of the model with the standard bones due to the difference in camera viewpoints. In order to solve this problem, we are use of a series of carefully designed algorithms to make the size and position of the real-time detected skeleton overlap with the standard skeleton to obtain the correct comparison analysis function.

Two steps are required to make the two models the same size and overlap for real-time comparison, namely scaling and alignment. Scaling is a computational process that enlarges the real-time skeleton in proportion to the size of the standard skeleton. After the scaling, the detected skeleton may appear in any part of the video frame, which will significantly affect the real-time comparison, so alignment is a computational way to overlap the two models in the spatial location of the frame to perform real-time pose correction.

We must firstly calculate a factor for scaling the model size to accomplish scaling and alignment. Firstly, we make use of the Euclidean distance to calculate the distance connecting each pair of key points in the standard and detected bones, we can take use of eq. (3.1) to accomplish this.

$$distance = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \tag{3.1}$$

where $(x1, y1)$ and $(x2, y2)$ denote two key point coordinates for each pair of connections, since the comparison of the posture skeleton does not involve the depth of the space, we do not apply the coordinates that include the depth in the calculation of the scaling and alignment method z. Next, based on the average of the Euclidean distances between the key points, we compute the scaling factor as

$$scale_{factor} = \frac{mean(standard_{distance})}{mean(detected_{distance})}. \tag{3.2}$$

With the scaling factor, we get the relationship between each pair of associated coordinate points between the two models. Thus, the scaling factor is harnessed to perform the scaling operation of the detected skeleton. Through the scaling factor, we calculate the coordinate data of each key point after scaling and then connect the lines to generate the scaled detection bone model. The equation for calculating the scaled coordinate points is shown in Eq 3.3 and Eq 3.4.

$$scale_x = x * scale_{factor} \tag{3.3}$$

$$scale_y = y * scale_{factor} \tag{3.4}$$

where x and y are the original key point coordinates, multiply the coordinate data with the

scaling factor to get the scaled coordinate values. At this point, the scaling operation of the model is completed, and the next step is to align the model. In order to make the model overlap correctly, we need to calculate the centre of gravity of the model.

$$centroid_x = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.5}$$

where *n* is the number of key points, and *x* is the coordinates. We must also replace *x* with *y* to get the complete coordinate data. After calculating the centre of gravity of the standard bone and the detected bone separately, we calculate the final alignment vector based on these two values.

$$transfer_{vector} = centroid_{standard} - centroid_{scaled} \tag{3.6}$$

where $centroid_{scaled}$ is the centre of gravity of the scaled bones, the two models will be in a horizontal position after the scaling operation, and by calculating the alignment vectors, we can move the detected bones horizontally to overlap with the standard bones.

The proposed method does not involve complex algorithms or in-depth mathematical modelling, so it guarantees high responsiveness in real-time scene computation. The method is based on the input bone data, which means that it is highly adaptable to different inputs and scenarios, and the centre of gravity calculation reduces the impact of individual anomalies on the overall results and is highly robust in practical applications.

3.4 Posture Evaluation

In order to provide billiard players with more accurate real-time feedback on the completion, we are use of a joint angle-based calculation method to assess the degree of completion. This method also relies on the key point extraction. By drawing the skeleton of the human body posture, we can utilizes the joint angles simulated by using the key point connectors to calculate the angle information of each joint of the player's body after the stroke.

In order to calculate the angle of each joint, we need the key point that represents the current joint and the other two points that are connected to it and form the joint on the image. Assuming that the joint we want to calculate is point **B** and the remaining two points are **A** and **C**, we perform a vector representation.

- The vector from B to A: $\overrightarrow{BA} = \vec{A} - \vec{B}$
- The vector from B to C: $\overrightarrow{BC} = \vec{B} - \vec{C}$

Next, cosine function is employed to calculate the vector pinch angle β.

$$\cos(\beta) = \frac{\overrightarrow{BA} * \overrightarrow{BC}}{\left|\overrightarrow{BA}\right|\left|\overrightarrow{BC}\right|} \tag{3.7}$$

Finally, we make use of the inverse cosine function to get the actual angle value.

$$\beta = \arccos\left(\frac{\overrightarrow{BA} * \overrightarrow{BC}}{\left|\overrightarrow{BA}\right|\left|\overrightarrow{BC}\right|}\right) \tag{3.8}$$

This calculation method takes the positional relationships of key points in 3D space into account, thus, this angle calculation can handle more details in pose recognition.

Eq (3.8) outlines the procedure for calculating joint angles in 3D space, usually, in practice, we wrap the algorithm into a function that ensures that the angle of each joint is calculated correctly by iterating through all the required angle combinations. In order to evaluate the player's striking posture in real time, we extract the body joint parts that have the most influence on the billiard ball striking posture based on the standard posture skeletal model in the real-time skeletal comparison.

The striking posture is a complex and delicate action. By analyzing the postures of professional coaches, a correct striking posture requires that the four joints of the body, namely the shoulder, hip, elbow and knee, are kept at the proper angles, and then the striking action is carried out according to the elbow force of the player's cue hand. Therefore, we firstly extracted the angles of the shoulder, hip, elbow and knee in the standard posture as the standard values. Then, the real-time detected angles of the joints were removed from the standard angles by a difference and saved as the parameter d. Consequently, we compare the real-time angles with the standard angles and convert the results into percentages. Moreover, we obtain the angles.

$$P = \frac{\beta - d}{\beta} \times 100\% \tag{3.9}$$

This approach converts the deviation values of the angles into percentages, as the result gets closer to 100%, the closer the player's posture is to the ideal.

The advantage of this approach is that it is simple to compute. We compute the scores for

each joint angle with minimal delay in a real-time scenario. However, each player's height and body shape differ in a real-time billiards scenario. Each person's correct posture may have a specific error from the standard posture, so in the actual use of this method, we need to set a specific threshold centered on the standard posture to ensure that the actual billiards hitting posture analysis will not be due to the player's height or body shape factors that provide incorrect analysis results on the posture.

3.5 Billiards Player Striking Posture Dataset

A billiard player's striking pose comprises a series of fine-grained body movements. If we want the poses recognized by the model to be as accurate as possible, we need to ensure that each data in the dataset contains the beginning and end of a complete action. Therefore, the existing public datasets are likely inadequate for the project, so we created a dataset containing the striking poses of billiard players. We filmed and recorded each complete stroke from a real billiard scene. In order to make the model have better accuracy and generalization ability, we collect the pose video data from different angles, heights and depths centered on the player, and in order to make the data richer, we collect the data in different scenarios, illumination, and partially occluded situations.

The total number of video footages collected is 668, a small deep learning class. To ensure the accuracy of the recognition and classification, we finally only performed the dataset for the binary classification task, i.e., “standard” and “non-standard poses”. The raw data for the positive category contains 312 entries, while the raw data for the harmful category contains 356 entries. The resolution of each video data is 1920 ×1080 HD video, which means that the video data is richer in human pose details and can capture more accurate human key point information when using key point detection. Moreover, by analyzing and observing the videos, we found that the average number of frames in a completing for a hitting action is around 75 frames, which can further determine the time step of the model in processing the data.

The size of the existing dataset is small, the overfitting phenomenon may occur due to the poor generalization ability of the model because of the insufficient amount of data in the actual deep learning model training. In order to prevent this from happening, we have to use data

enhancement methods to enrich the content before conducting the actual training. The data enhancement methods include:

- **Horizontal flipping.** This method increases the diversity of the data by flipping each frame of video horizontally; thus, the method is relevant in recognizing the hitting posture of a billiard player. In real sports, there will be a significant difference in players' hitting postures depending on their dominant hand. If right-handed players are in most of the dataset, we can increase the model's ability to learn the hitting postures of left-handed players by horizontal flipping.
- **Interpolation or Removal of Frames**. This method is applied to make the video action smoother and normalize the time step of the data by inserting or removing specific frames in the video. In our dataset, the average time step of the video is 75 frames, but there are fluctuations up and down. We control the video frame rate to 75 by interpolation or removal. This method is risky for the time series task, and one needs to ensure that the validity of the data is maintained when performing the differencing or removal. For the billiard ball striking posture, the body posture will not change too drastically after completing a standard. The only force exerted during the striking process is the arms and elbows of the cue holder, so adding or removing frames to eliminate the noise of the data that fluctuates too much will not adversely affect the model's performance for this action.

Furthermore, the paramount advantage of utilizing data augmentation on small datasets is that the approach significantly increases the data diversity and the number of samples in the training data so that the model can learn to pose features of richer scenes, increasing the generalization ability of the model and avoiding model overfitting.

To train the Transformer model, we need to tackle the data further. We are use of the key point extraction method to acquire the key point coordinates of human figure body poses in the video, extract the key points of the poses in each frame, and then combine them according to the order of the frames to obtain the coordinates of the ball player's batting motion with temporal features. The data enhancement methods take into consideration of this aspect by changing only the rate or horizontal direction of the action and not rotating the original image,

thus ensuring that the coordinate point values are reasonable in real space. Next, we saved the extracted key point data in CSV format for training the model. We designated 80% samples for training while reserving the remaining 20% for testing.

Chapter 4 Results

The main content of this chapter is to collect video data and demonstrate the experimental results. In the end, in this chapter, we also discuss the limitations of this project.

4.1 Real-time Pose Comparison

This chapter will examine the real-time skeletal comparison function for billiards players' striking postures and its crucial value in human posture analysis and recognition. Billiards is a sport game that requires precise control and skill in body posture, its striking posture plays a decisive role in the ball's direction, speed, and spin. Thus, constantly correcting the striking posture is essential for players to improve their billiard performance.

Postural skeletal comparison aims to capture a billiard player's striking posture in real-time by using a key point extraction technique to extract the coordinates of human joints and generate a skeletal model of the striking posture. We compare this real-time model to a pre-defined standard or ideal stroke posture skeletal model. This method makes it very obvious to coaches or players if the posture could be better, for example, when the elbows are too high out of place or the knees remain straight. This method provides immediate physical feedback to the player to help them better understand the deficiencies in their senior movements.

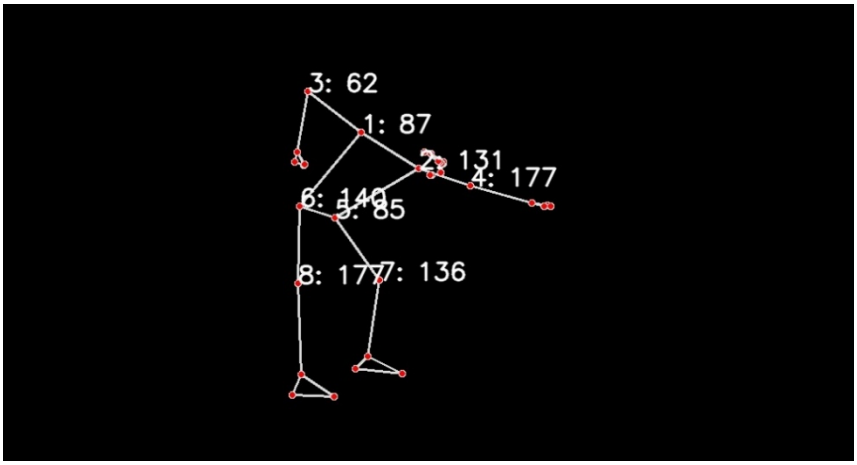


Fig 4.1The standardized bone models and joint angles

The standard skeleton was drawn from the description of a standard billiard player's striking posture by using a professional player on the Internet. We firstly found a picture of a professional billiard coach's striking posture to draw this skeleton model. Then, we extracted the key points of the human body by using human posture recognition to compute the angles of the four significant joints, namely, the shoulder, the hip, the elbow, and the knee. We are use of the MediaPipe framework as the primary source of key point recognition to draw the skeletal model of the standard pose based on the extracted key point coordinates, as shown in Figure

4.1. This standard pose contains 33 key coordinates along with angle data for each of its keys, we extracted angles for only four joints: Shoulder, hip, elbow, and knee, with different angles for the left and right sides of each joint.

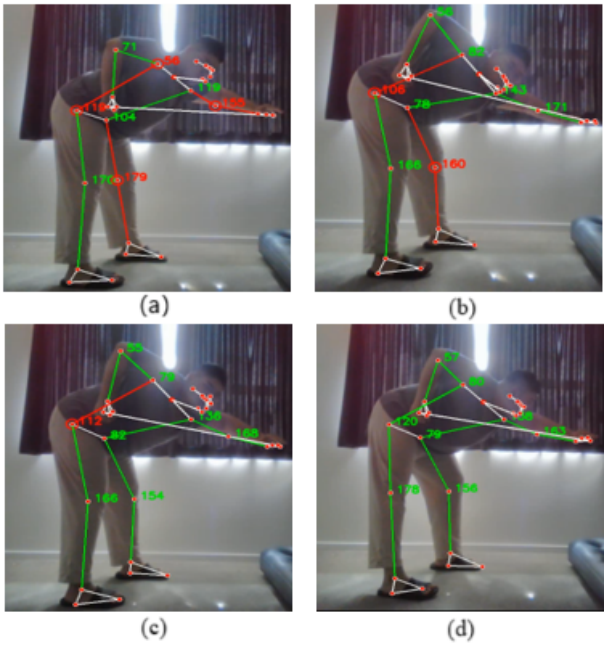


Fig 4.2 Skeletal modeling and angle analysis for real-time detection

The detection and drawing of real-time skeleton are roughly as same as that of the standard skeleton. We firstly extract the real-time coordinates of key points of human posture through key-point extraction method and then draw the human skeleton model. In real-time detection, we combine the cosine and inverse cosine functions to compute the natural joint angles based on the joints and the related two-coordinate points.

In the previous standard posture extraction, we obtained the joint angle data of the standard posture. We are use of this angle as the benchmark and the upper and lower 15 degrees as the threshold for the real-time detection of the skeletal model to provide a simple posture analysis when the player performs the batting posture if the joints comply with the standard, the real-time detection of the joints will appear in green, if they do not comply with the standard, they will change to red. In Figure 4.2, in order to simulate a complete billiard player's hitting action, we added a white line between the two hands to simulate the cue state in the scene.

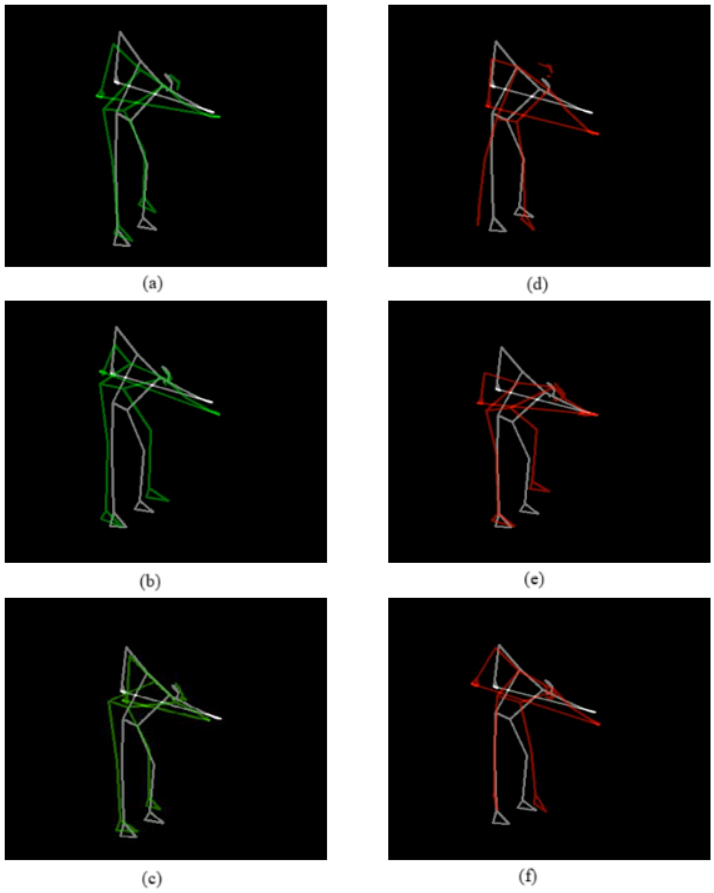


Fig 4.3 Comparison of real-time skeleton with standard skeleton

We have obtained a standard skeletal model and a real-time detected skeletal model. Then, we combine the two to obtain a real-time pose skeletal comparison system. We scale and align the two models so that the two skeletal models overlap in a predefined area. A zooming algorithm scales the skeletal models to make the two models the same size and keep them on the same level, and then the scaled skeletal models are aligned and overlapped using an alignment algorithm.

For the pose comparison, we focus on the pose of human bones, we extract the angle data separately in the comparison model for use in other analysis methods. We have set a threshold for pose completion considering the physical factors of different players. The model will turn green if the pose follows a standardized approach, the player's pose is well completed, while red signals non-standardized poses. Figure 4.3 shows the skeletal model in a real-time detection scenario for comparison experiments. Figure 4.3 (a), (b), and (c) show that when the player's posture is relatively correct, the detected bones will turn green and emit a prompt sound to indicate that the player's posture is good. In contrast, Figure 4.3 (d), (e), and (f) show that when

the player's posture is incorrect, the detected bones will appear red so that the player or the coach can find out the reason for the wrong posture through comparative observation.

4.2 Posture Integrity Evaluation

In the previous step, we took use of the method to extract the key points of human pose and stitch them into a bone model, we calculated the joint angle data for the standard pose. We obtained joint angle data in real-time. In order to provide players and coaches with a more in-depth analysis of the batting posture, we calculated the deviation of the current player's posture from the difference between the real-time angle and the standard angle. Then, we calculated the result as a percentage. To increase the detail, we calculated the angle of each critical joint and ranked them while showing only the four joints with the poorest degree of completion. Figure 4.4 shows the real-time detection of player's joint completion in a billiards scenario.

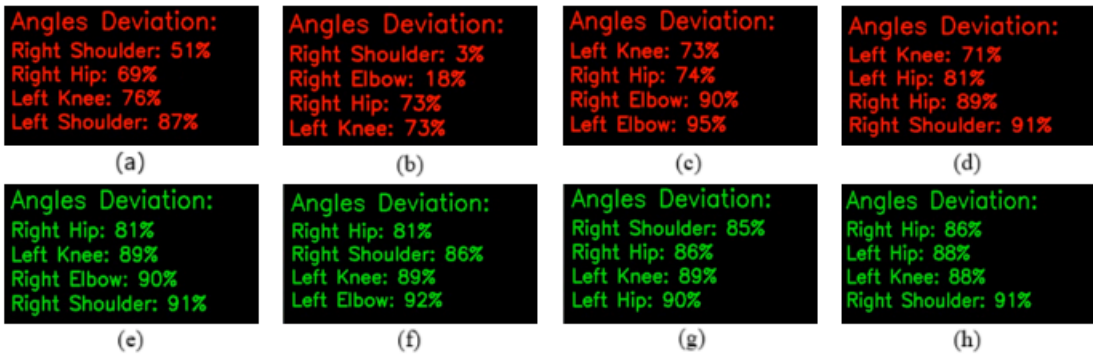


Fig 4.4 Scoring system based on joint angle calculation.

Figure 4.4 shows that each dataset demonstrates the four keys with the most significant deviation values in the current posture, again green when completed correctly and red when incorrect. The evaluation combines the detection results of the whole model, as shown in Figure 4.4 (c) and Figure 4.4 (d). The detection results show that the current player's joint angles are high in completion. However, the detection results are red because the player did not correctly perform the billiard ball striking posture. This approach avoids misrecognition when the detected target does not perform a stroke.

4.3 Training Transformer Model

We have described the structure of Transformer model in the methodology section, next, we will discuss the specific training method of the Transformer model in this subsection.

- **Optimizer.** We are use of Adam optimizer in the training of the model. The benefit of this optimizer is that it can adapt the learning rate, adjusting the learning rate of each parameter by calculating the one-section matrix estimation and the second-order matrix estimation of the gradient. Adam also introduces a bias correction mechanism to avoid underestimating the learning rate at the early stage of training.
- **Loss function.** The cross-entropy loss function enjoys wide recognition as providing meaningful feedback on the correctness of model predictions in classification tasks. The function compresses the gap between the model's predicted probability and the accurate probability distribution, with the loss converging to 0 as the predicted probability approaches the actual probability distribution.
- **Training period.** We set the initial training period to 100 waves. This aims to observe the process of convergence of the model in more extended training periods to find the optimal number of training waves.
- **Learning rate.** We set the initial learning rate at 0.0001, then dynamically adjusted the learning rate using a cosine annealing strategy. This method decreases the learning rate rapidly at the beginning of training. It adjusts the rate of decrease of the learning rate after training, which allows the learning rate to be adjusted accordingly at different training stages.
- **Regularization.** We added regularization to the model to avoid overfitting of the model. The regularization technique is applied by adding a Dropout layer to the model structure and setting the dropout rate to 0.5.

In addition, we initialized the hyperparameters for the model. Among them, the number of hidden units of the model is 256, and the model depth is 2.

After finishing the setup of the model initialization, we start the training of the model. The evaluation criteria of the model are accuracy, loss value and F1score. In addition, the scenario of the model is real time, so we evaluated response speed. We collected the statistics of the

curves for each evaluation criterion after the training of the model lasted for 50 waves. Figure 4.5 illustrates the accuracy, loss value and F1 score curve statistics at the end of the training.

We assessed the model performance by using a blend of the three criteria, so only when each criterion is improved, it indicates that we have enhanced the optimal performance. After 50 training iterations, we obtained the best results for the model with an accuracy of 98.32%, a loss value 0.0957 and an F1 score 0.98.31. We then analyzed the model performance comprehensively using graphs. In 50 training iterations, the model accuracy continues to increase, which indicates that the model gradually learns and adapts to the patterns in the data and classifies them correctly. The loss value shows a decreasing trend, which indicates that the model continues to optimize the prediction ability during training, which is a signal that the model is converging well. The F1 score, which considers precision and recall, shows an increasing trend, further for verifying the model effectiveness in this classification task.

Regarding the test and validation curves, we notice that the loss values of the model on the training and validation data are continuously optimized as the number of waves increases. When we look at the accuracy and F1 scores, the gap between the training and validation curves does not increase. This phenomenon illustrates that the model continuously learns and improves its predictive performance.

The difference in values between training loss and validation loss is slight, and the validation loss converges to the training loss in the later stages of training, suggesting that the model is continuing to improve. We only updated the models when all three metrics improved during training, so we only saved the models that performed relatively well. The absence of overfitting of the model is also shown in the accuracy and F1 score graphs.

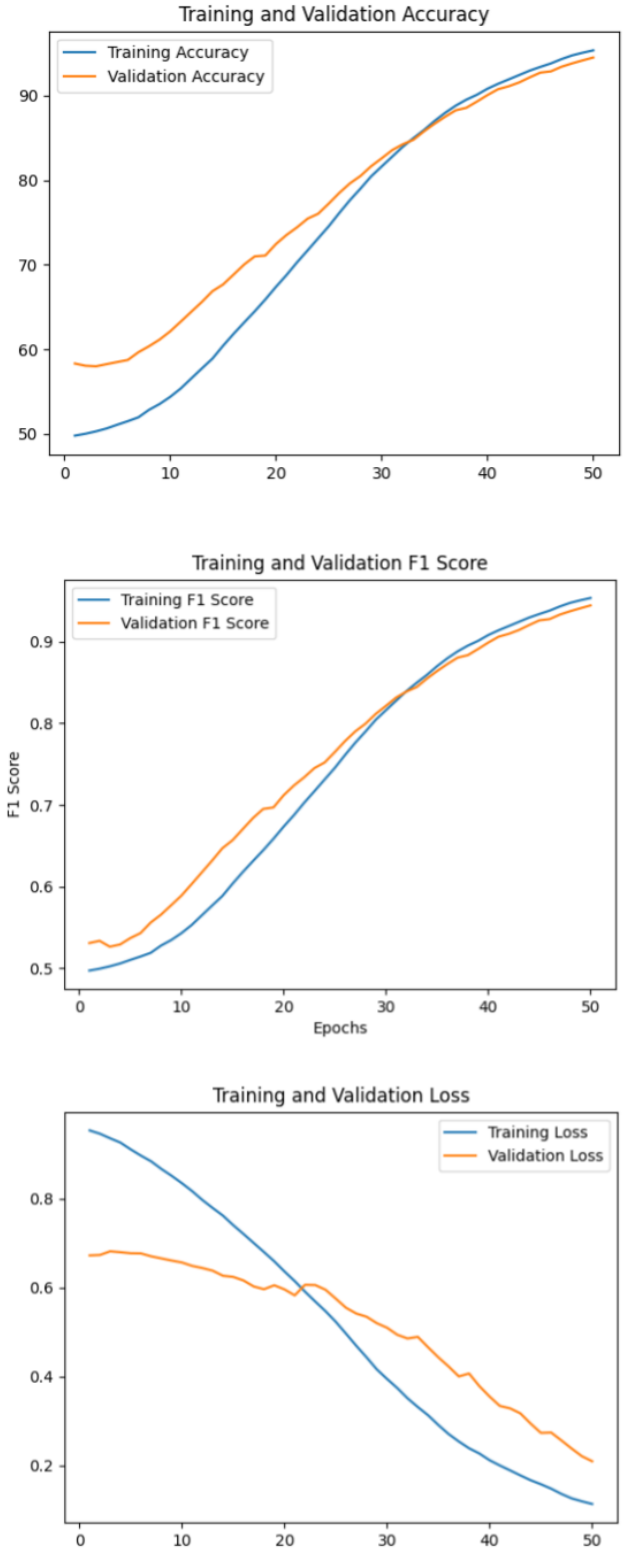


Fig 4.5 Transformer model accuracy, loss, and F1 score plots.

In order to meet the task requirements of real-time human pose detection, we also performed graphical statistics of the model response rate, as shown in Figure 4.6, where the

blue line is the training response rate, which averages around 0.7 seconds, representing the time it takes for the model to receive new data, propagate it, and update the parameters. The red line is the test response speed, which has an average speed 0.02 seconds, ensuring that the model can provide almost real-time feedback when making predictions. In addition, the horizontal trend in the speed profile indicates that model training continues to be stable.

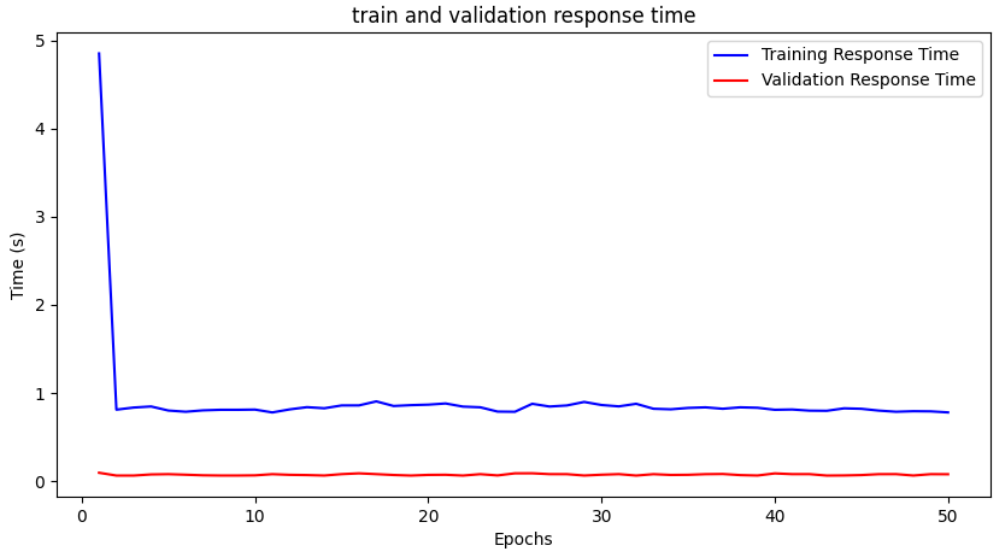


Fig 4.6 Transformer training and validation response speed.

4.4 Ablation Experiment

The initial training parameters of the model were chosen based on theoretical knowledge and experience. In order to provide a more reliable basis for the model parameters, we conducted ablation experiments to ensure that the main components of the model structure play an active role in the model. Further optimization is possible through the ablation experiments, which we conducted in strict accordance with the design of the methodology.

The model components involved in the ablation experiments include convolution and pooling, encoder and decoder, activation function, number of multi-attention heads, number of hidden units and number of self-attention layers. In each ablation experiment, we evaluate the model by combining accuracy, loss value and score. Table 3.1 shows the initial structure of this proposed model.

We firstly perform the ablation experiments with activation functions, Table 4.1 shows the evaluation scores of the best model after 50 waves of training using different activation functions.

Table 4.1 Results of activation function ablation experiments.

Activation Function	Accuracy	Loss	F1-Score
None	98.32%	0.0957	0.9831
Mish	89.72%	0.3588	0.8971
Swish	90.28%	0.4117	0.9028

From the results in Table 4.1, the model overall performance without the activation function is better, with an accuracy 98%, a loss value 0.09, and an F1 score 0.98. The model performance is similar when using $Mish$ and $Swish$. However, the accuracy drops by about 10% compared to the best model performance, and the loss value is much higher than the best performance. Thus, we performed the model structure ablation experiment again with 50 training waves and got a better structure of the model, as shown in Table 4.2.

Table 4.2 Results of model structure ablation experiments.

Model Structure	Accuracy	Loss	F1-Score
Conv and Pooling	98.32%	0.0957	0.9831
Encoder and Decoder	83.18%	0.4541	0.8319
Only Encoder	79.07%	0.4872	0.7903

In the ablation experiments, we explore three model structures: A basic Transformer, a Transformer that integrates convolution and pooling operations, and a Transformer that only contains an encoder. The performance of each structure is compared to clarify the impact of the different structures on model performance. Among them, the Transformer with convolution and pooling operations has the best performance, which improves the model accuracy over the base structure by 15% and 19% over the accuracy of the Encoder-only model, it has an average training time 0.7 seconds and an average validation time of less than 0.1 seconds. The encoder model is less accurate but relatively lightweight, with an average training speed of 0.5 seconds

and a validation speed of 0.02 seconds. In the next step, we explore the effect of the multiple attention heads, Table 4.3 presents the experimental results.

Table 4.3 The results of the multiple attention heads ablation experiment.

Multi Attention Heads	Accuracy	Loss	F1-Score
8	98.32%	0.0957	0.9831
4	90.09%	0.3382	0.9006
16	90.28%	0.4336	0.9041

Multi-head attention allows the Transformer to focus on several locations when processing sequence data, each head can be considered a separate attention mechanism. Increasing the number of heads ramps up the complexity and the capacity of the model, but relatively also boot up the computation time. By varying the number of heads for multi-head attention, we find that the highest performance is obtained with a head count of 8 and that increasing or decreasing the headcount results in a decrease in model performance. Subsequently, we conducted model dimension ablation experiments. Table 4.4 shows the acquired result.

Table 4.4 Model dimension ablation experiment results.

Hidden Size	Accuracy	Loss	F1-Score
256	98.32%	0.0957	0.9831
96	75.51%	0.6363	0.7550
512	97.20%	0.1646	0.9720

The model dimension is critical to the performance of the Transformer, with a 23% decrease in accuracy while reducing the model dimension to 96 compared to a dimension of 256. When the model dimension was increased to 512, the model also achieved extremely high performance, but the training time was significantly increased, as shown in Figure 4.7. At this viewpoint, the model was trained in 1.6 seconds on average, compared to 0.7 seconds on average with 256 model dimensions.

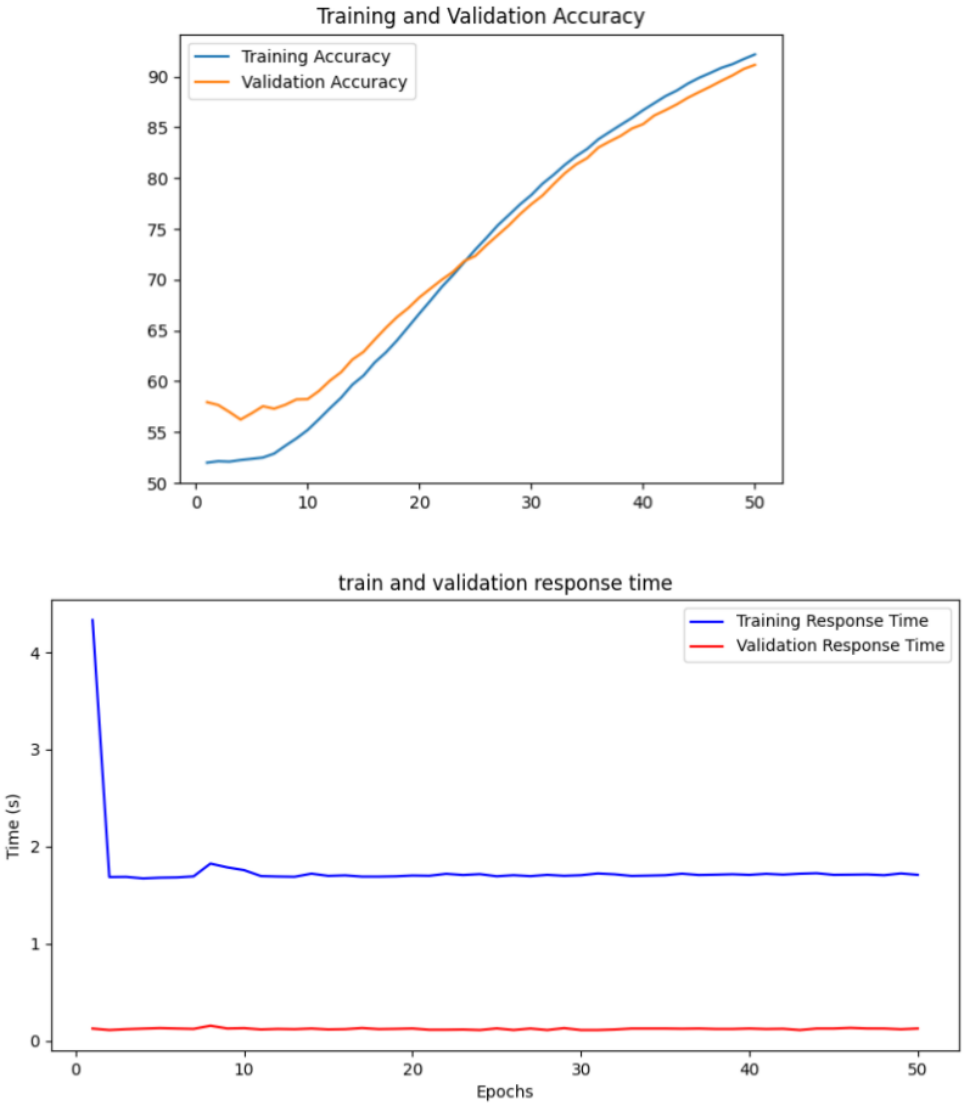


Fig 4.7 512 Model dimension accuracy and speed.

Table 4.5 shows the results of our ablation experiments with a various number of Transformer layers. The number of layers represents the depth of the model that contains the self-attention mechanism. Increasing depth allows the Transformer to capture more complex patterns in the data. However, as the number of layers increases, the model parameters increase, and overfitting may occur.

In addition, an increase in the number of layers increases the training and prediction time of the model. In this experiment, if the number of layers is 6, the model performance decreases by 10%, while its loss curve shows that there is overfitting in the model, which is caused by the complex structure of the model. If the number of layers is 10, the model converges well, but the training time is significantly longer. Although the loss curve shows that the model learns

gradually, the model accuracy only reaches 57% after the 50 iterations. The loss curves for 6 layers and 10 layers are shown in Figure 4.8.

Table 4.5 Results of Num layers ablation experiments.

Layer Size	Accuracy	Loss	F1-Score
2	98.32%	0.0957	0.9831
6	88.79%	0.4355	0.8878
10	57.01%	0.6926	0.5530

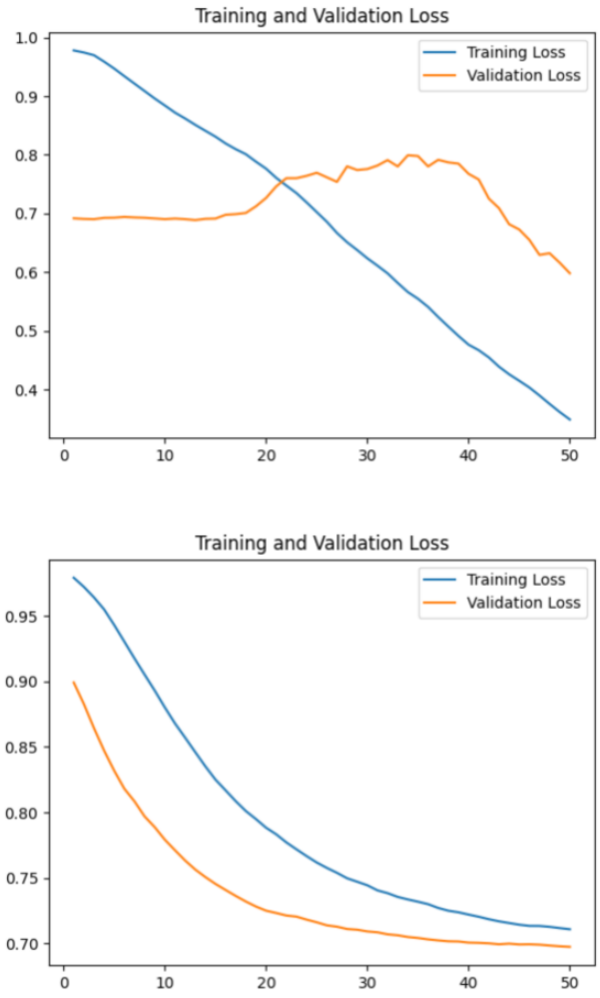


Fig 4.8 The loss curves for 6 layers and 10 layers.

The ablation experiment obtained the best-performing model structure and parameters, as shown in Table 4.6. Through the ablation experiment, we obtained that the value of the ablation

experiment also manifests in the importance of validating the model formation. By removing or replacing components or parameters in the model, we determine which components are critical to the performance of our proposed model. The experiment provides an intuitive way of assessing the validity of the individual components of the model.

Table 4.6 Optimal model structure and parameters.

Activation Function	Model Structure	Multi Attention	Model Dimension	Layers
None	Conv & Pooling	8	256	2

4.5 Comparative analysis of Transformer with RNN and LSTM

We clarify the optimal structure and parameter configurations of Transformer model through ablation experiments, followed by comparisons and analyses to examine the performance of the optimized Transformer model for other standard recurrent neural networks, i.e., the RNN and LSTM. This approach verifies the superiority of Transformer for the present task, it also helps to understand the adaptability of the different models for analyzing the hitting postures of billiard players. To ensure the fairness of the comparison, all the structures in the whole model except the core modules of the different networks are consistent with the Transformer. Meanwhile, each model underwent training through 50 waves. To fully observe the model performance on the dataset, we evaluated the models based on accuracy, loss, F1 score and response speed. After the training, we get the optimal model parameters for LSTM, RNN and GRU as shown in Table 4.7.

Table 4.7 Transformer, RNN, and LSTM optimal model parameters

Model	Accuracy	Loss	F1-Score	Speed (s)
Transformer	98.32%	0.0957	0.9831	0.0263
RNN	76.3582%	0.479	0.804	0.0296
LSTM	94.029%	0.144	0.943	0.0299

We analyze the results comprehensively. Regarding accuracy, Transformer reaches up to 98%, and LSTM reaches 94%. In comparison, RNN only attains 76%, where Transformer has the highest accuracy, which means that the model performs the best in this task. The accuracy of RNN model is significantly lower than Transformer and LSTM, but the value is acceptable in practical application scenarios. Analyzed the loss, Transformer has a lower loss value of 0.09,

which indicates that Transformer fits the data well during training.

In contrast, RNN has a loss 0.47, LSTM has a loss 0.144, both of which are higher than the Transformer model, which indicates that RNN and LSTM do not perform as well as Transformer in fitting the data. Similar to the performance of accuracy, Transformer still has the best performance on F1 score remains the best at 0.98, while the F1 score of RNN and LSTM is only 0.8 and 0.94, which means that they do not perform as well as the Transformer model in terms of precision and recall.

Focusing on the response speed, we find that all three models are very close to each other with a response speed of around 0.0029 seconds, which indicates that all three models demonstrate adaptability to real-time detection scenarios. The model performance tightly correlates with the content of dataset as well as the purpose of the task. In classifying the striking poses of the billiard players, Transformer performs well in terms of accuracy and F1 score that has a low loss value, which means that the Transformer model is the best choice for this particular task and dataset.

Although the Transformer model obtains the highest level in terms of evaluation scores, the combined performance of RNN and LSTM is still up to the task, in order to gain further insight into the behavior and potential differences between the three models during training, we decided to perform a detailed analysis of the training curves. Figure 4.9 illustrates the accuracy and loss graphs for the RNN.

In the previous paragraph we have shown the evaluation graphs of the Transformer model in Figure 4.5. The Transformer model generally shows a stable and expected trend, while there is no severe overfitting. These observations imply that the Transformer model performs well in recognizing the hitting pose of a billiard player while having good generalization ability and robustness.

In summary, for the billiard player striking pose recognition task, the Transformer model performs best overall, especially in the critical metric of accuracy. Theoretically, Transformer, RNN, and LSTM can be analyzed for time-series data; therefore, all three models are suitable for this experiment, while the comparative analysis experiment provides us with the optimal solution for choosing which model is the best one, the comparative experiment can provide us

with precise data to make an informed decision.

In addition, this experiment can prevent over-reliance. No model is optimal in all task scenarios, by comparing methods, we can better understand the limitations of different models. This comparative analysis is crucial for understanding model performance and robustness that provides an essential reference.

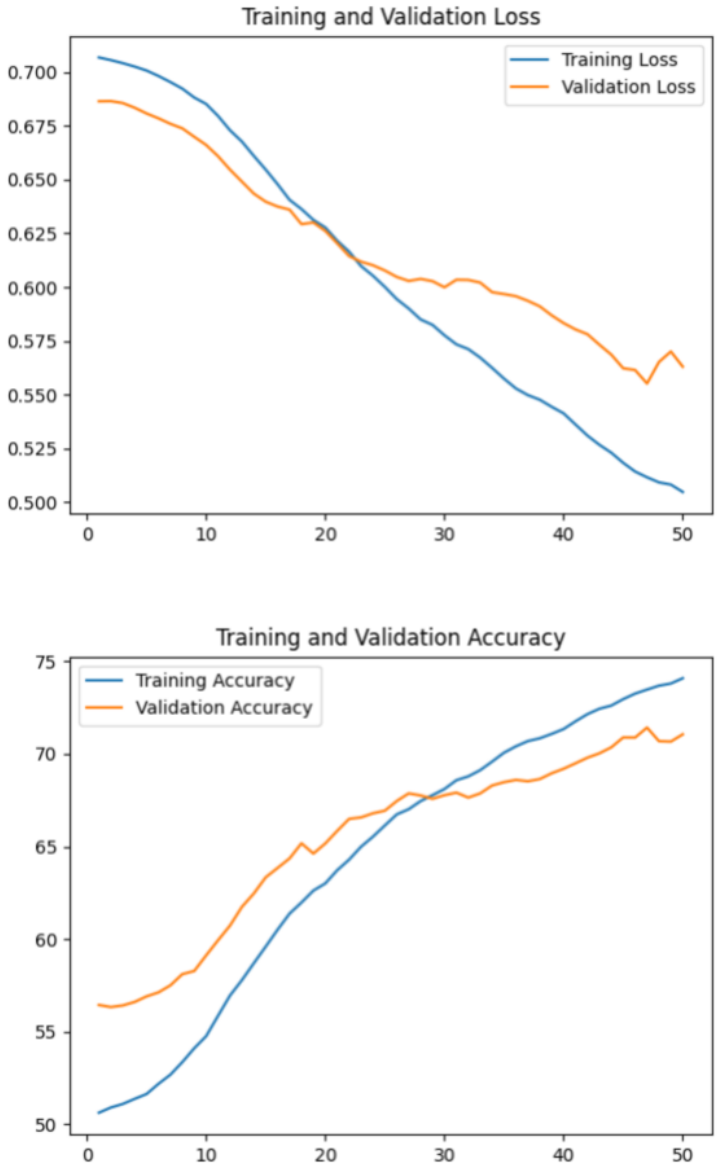


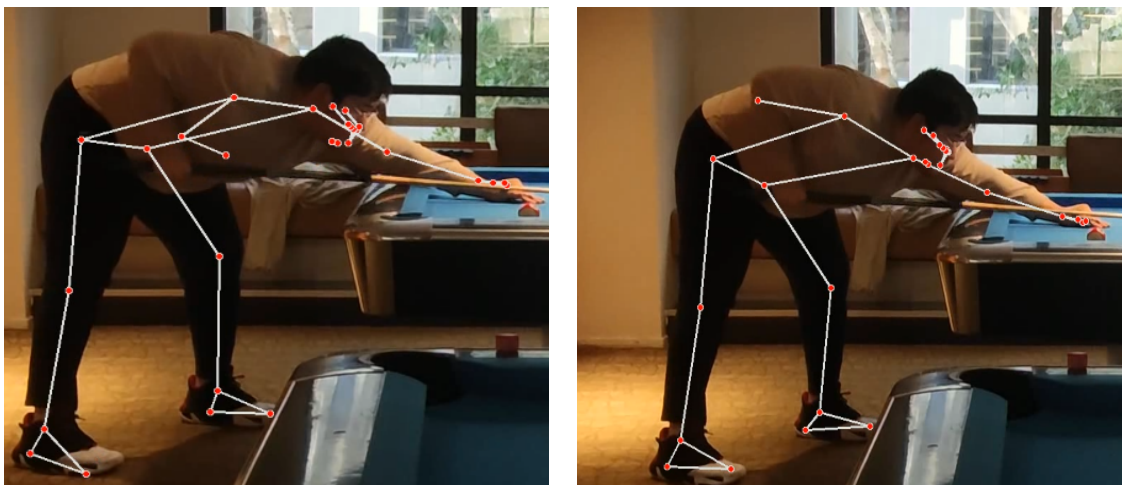
Fig 4.9 Accuracy and loss graphs for the RNN.

4.6 Sliding Window

In the whole course of billiard player posture recognition and analysis, the most critical data source is human body key points. All the sub-systems are realized based on this data. However, in the real-time scenario, based on the complexity of the scene, lighting occlusion and other problems, the extracted key points will be jittery or even chaotic, so in order to make the key point skeletal model more. Therefore, we are use of sliding window method in the global key point extraction to make the key point skeleton model more stable and smoother. The main idea is to move the time series data set to a fixed-size window and process the data in the window to stabilize the data performance.

The main principle of this method is to detect the points in the sliding window through determining whether each of them exceeds the thresholds of the mean and standard deviation to detect to carry out the detection of abnormal key point data. This approach is efficient when dealing with real-time data, the output attitude model can be more stable. Figure 4.10 shows the effect of key point extraction before and after using the sliding window.

In Figure 4.10 (a), before using the sliding window, the key points recognized by the model were jittery, the right-hand key points deviated from the actual position. In Figure 4.10 (b), the key points appear jittery at the same time after using the sliding window. However, the sliding window takes use of the average value as a substitute to the displayed skeleton relatively stable.



(a) (b)

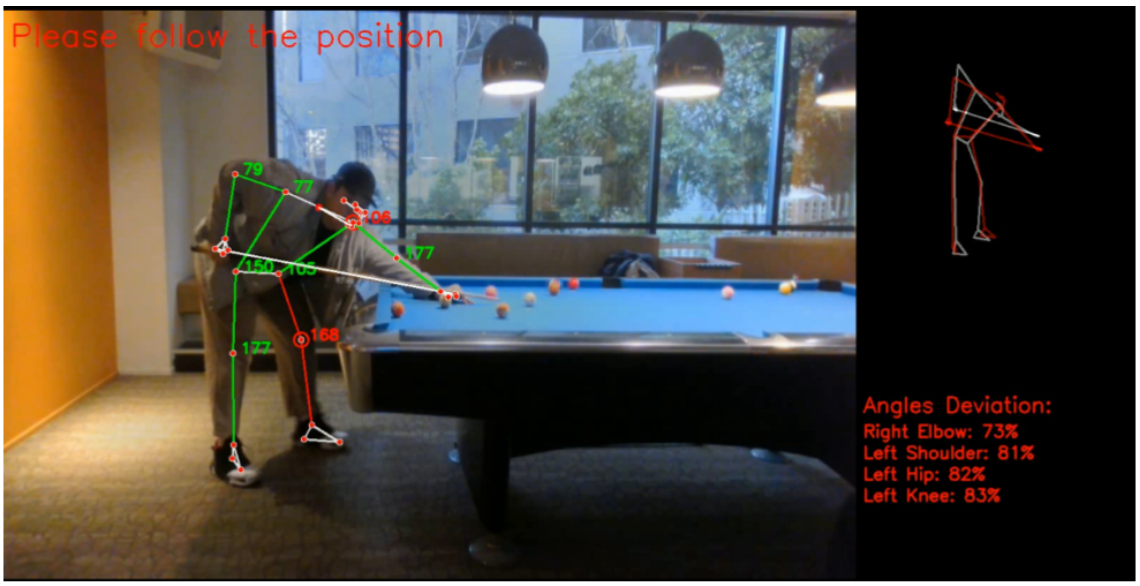
Fig 4.10 Sliding window stabilization pose model

4.7 Real-time Billiard Player Pose Analysis

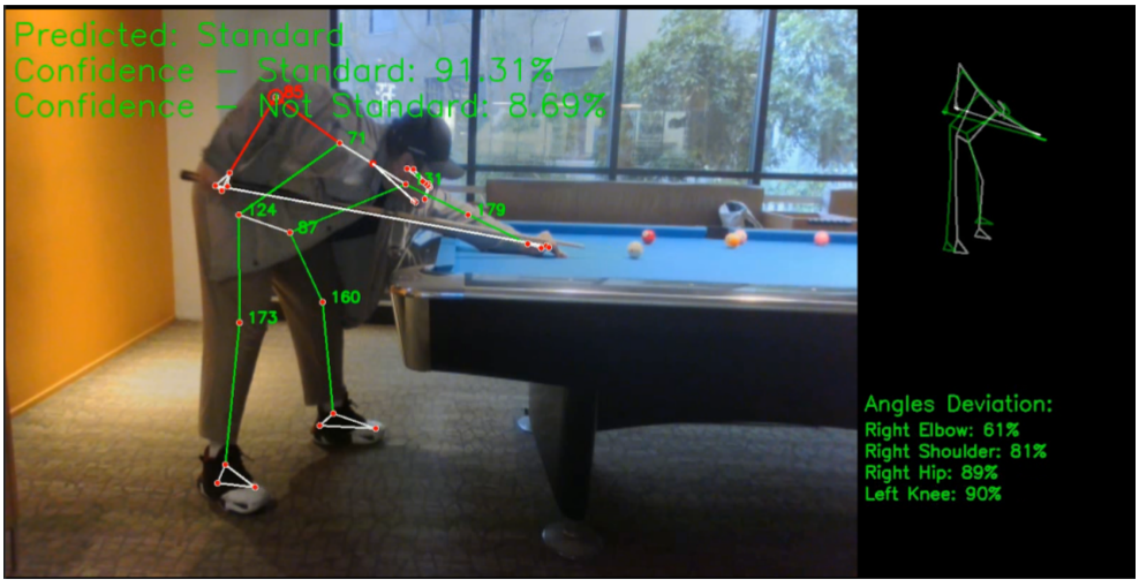
In this report, we describe the various parts of our system, including the real-time skeleton comparison, the pose scoring, the training of the Transformer model, as well as conducting ablation experiments and comparison tests to optimize the Transformer model, comparing the optimized Transformer with other time-series models to demonstrate its excellent performance. In addition, we also used a smoothing window to make the detected critical point skeleton more stable.

For the operation model, we firstly extract real-time pose key points and then use the smoothing windows to deal with abnormal key point noise. These extracted key point data are simultaneously computed and analyzed in three parts. The first part is the real-time comparison of the skeleton of the key points, which draws the real-time skeleton so that the player can compare themselves with the standard pose. Secondly, the system calculates the posture score. This part of algorithm calculates the real-time joint angles from the key point data and then computes the joints in real time based on the standard angles.

Thirdly, we feed the key point data into the trained Transformer model for real-time prediction, the model will display the output score to the player by calculating the confidence level through the Softmax function. This role is to use deep learning methods for judging the player's batting posture of the standard and substandard. Then, we united these three subsystems into one model. Only when the player meets all the evaluation criteria, the system will treat the current action as a standard action, the player can observe through the skeletal comparison of the current action and the standard action that there is still a subtle gap through the posture scores as well as the results of the judgment to provide a more detailed analysis of the defects of the current posture. Figure 4.11 shows the performance of our proposed method.



(a)



(b)

Fig 4.11 A system for analyzing the striking posture of billiard players.

As shown in Figure 4.11, the Transformer scores a percentage for the overall degree of standardization of the pose. At the same time, real-time skeletal and angular calculations can explicitly indicate that part of the player's joints in the current pose is not up to standard. For example, the left knee is marked in red to indicate insufficient flexion and the right hip is marked in red to indicate too much downward pressure on the body.

Chapter 5
Analysis and Discussions

In this chapter, experimental results are analyzed and compared. Comparisons of the results under various conditions will be explored.

5.1 Analysis

In summary, we combined human posture estimation and deep learning to analyze the hitting postures of billiard players. Firstly, the key point extraction method identifies the key point data. Then, the recognition data was stabilized by using a sliding window, followed by drawing a bone model by connecting the key points and comparing the detected bones with the pre-set standard bones in real time. Subsequently, the system calculated the real-time angles of the joints through an algorithm that judge the joint pose completion.

Secondly, we took use of the created billiard player hitting pose dataset to train a Transformer model for player's pose for predicting the player's behaviors. We adopted ablation experiments to seek the optimal model structure, the final model achieved an accuracy 98%, a loss 0.09, an F1 score 0.98, and a response time 0.02 seconds. The favorable evaluation scores attest to the generalization ability and stability of the proposed model, and a response time that satisfies real-time scenarios.

5.2 Discussions

In our experiments, we trained RNN and LSTM models following the same structure as the optimized Transformer model, the training data also was treated as one part of the billiard player's pose dataset. We compare the evaluation results of LSTM, RNN, and Transformer, we find that the RNN has the lowest accuracy 76% with high model loss and overfitting. In comparison, the Transformer has 98% accuracy and no overfitting. These results indicate that the optimized Transformer is more suitable for player pose recognition in the current task scenario.

In human pose estimation scenarios, occlusion and illumination are still main factors affected the recognition accuracy. Although the existing deep learning methods can complement the key points of poorly recognized body parts by using prediction, there are still cases where the recognition results are noisy or even messy, which can be further mitigated by using the sliding window method, which firstly sets a sliding window, then deals with the outliers and complements them with the average value, thus making the detected key point data

more stable.

In this system, the pose comparison of human skeletal model and the evaluation of joint completion play an essential role in assessing the player's posture. The pose comparison provides players with a clear understanding of where their poses differ from the standard pose, the joint completion assessment further enriches the details of such differences. Overall, our study combines human posture estimation techniques with Transformer to realize an efficient real-time recognition and analysis system for billiards players' hitting postures, which provides a powerful solution to such problems. We also provide new perspectives for Transformer in solving visual tasks by converting image data into a coordinate corpus through key point extraction and then using the power of Transformer to learn features between coordinates to solve human pose recognition and analysis.

Chapter 6 Conclusion and Future Work

In this chapter, we will summarize the subject and method of this project and propose new research direction according to the result and insufficiency of the experiment as well as the future work.

6.1 Conclusion

This report aims to combine human pose estimation and realize the striking pose and analysis of a billiard player in a real-time scene. We combine the Transformer with human pose estimation to create a corpus of human poses using key point extraction, which provides a new perspective for the Transformer to recognise and analyse human poses. The extracted key points can also be employed for pose comparison and evaluation. The Transformer model is harnessed to make much accurate and objective judgments on the recognized poses. We also created a dataset for the recognition of striking poses of billiard players, in order to optimize the performance of our model, we performed ablation experiments and compared the results with RNN and LSTM, finally the optimized Transformer achieves an accuracy 98% and a response time 0.02 seconds.

6.2 Future Work

We will increase the size of the dataset to include more video frames of player’s poses in different hitting scenarios through improving the model's ability to generalize different scenarios and its accuracy. We will also focus on tracking methods to enable multitarget player pose recognition by using a top-down approach. In addition, we perform multiclass pose modelling that can enable players to correct their movements across a diversity of hitting angles and postures.

References

Angelini, F., Fu, Z., Long, Y., Shao, L., & Naqvi, S. M. (2019). 2D pose-based real-time human action recognition with occlusion-handling. *IEEE Transactions on Multimedia*, 22(6), 1433-1446.

Abujar, S., Masum, A. K. M., Chowdhury, S. M. H., Hasan, M., & Hossain, S. A. (2019). Bengali text generation using bi-directional RNN. In International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5).

Agrawal, S., & Sharma, D. K. (2022). Feature extraction and selection techniques for time series data classification: A comparative analysis. In International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 860-865).

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

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157-166.

Badiola-Bengoa, A., & Mendez-Zorrilla, A. (2021). A systematic review of the application of camera-based human pose estimation in the field of sport and physical exercise. Sensors, 21(18), 5996.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877-1901.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional

and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). Blazepose: On-device real-time body pose tracking. arXiv preprint arXiv:2006.10204.

Borodulina, A. (2019). Application of 3D Human Pose Estimation for Motion Capture and Character Animation (Master's Thesis) University of Oulu, Finland.

Cust, E. E., Sweeting, A. J., Ball, K., & Robertson, S. (2019). Machine and deep learning for sport-specific movement recognition: A systematic review of model development and performance. Journal of Sports Sciences, 37(5), 568-600.

Cao, X. (2022) Pose Estimation of Swimmers from Digital Images Using Deep Learning. *Master's Thesis, Auckland University of Technology*.

Cao, X., and Yan, W. (2022) Pose estimation for swimmers in video surveillance. *Multimedia Tools and Applications, Springer*.

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. *In IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7291-7299).

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cooijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç., & Courville, A. (2016). Recurrent batch normalization. *arXiv preprint arXiv:1603.09025*.

Dang, Q., Yin, J., Wang, B., & Zheng, W. (2019). Deep learning based 2d human pose estimation: A survey. Tsinghua Science and Technology, 24(6), 663-676.

Dalal, N., & Triggs, B. (2005,). Histograms of oriented gradients for human detection. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 1, pp. 886-893).

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. *In IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2625-2634).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Ekvall, S., Kragic, D., & Hoffmann, F. (2005). Object recognition and pose estimation using color cooccurrence histograms and geometric modeling. Image and Vision Computing, 23(11), 943-955.

Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2), 179-211.

Fragkiadaki, K., Levine, S., Felsen, P., & Malik, J. (2015). Recurrent network models for human dynamics. *In IEEE International Conference on Computer Vision* (pp. 4346-4354).

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural Computation, 12(10), 2451-2471.

Gholamalinezhad, H., & Khosravi, H. (2020). Pooling methods in deep neural networks, a review. arXiv preprint arXiv:2009.07485.

Gowdra, N. (2021) Entropy-Based Optimization Strategies for Convolutional Neural Networks. PhD Thesis, Auckland University of Technology, New Zealand.

Graves, A., & Graves, A. (2012). Long short-term memory. In Supervised Sequence Labelling with Recurrent Neural Networks, 37-45.

Gu, D., Nguyen, M., Yan, W. (2016) Cross models for twin recognition. International Journal of Digital Crime and Forensics 8 (4), 26-36.

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *In IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6645-6649).

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6), 602-610.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. Diploma, Technische Universität München, 91(1), 31.

Herrera, A., Beck, A., Bell, D., Miller, P., Wu, Q., Yan, W. (2008) Behavior analysis and prediction in image sequences using rough sets. International Machine Vision and Image Processing Conference (pp.71-76)

Hori, T., Cho, J., & Watanabe, S. (2018). End-to-end speech recognition with word-based RNN language models. In IEEE Spoken Language Technology Workshop (SLT) (pp. 389-396).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *In IEEE International Conference on Computer Vision* (pp. 2961-2969).

Illavarason, P., Arokia Renjit, J., & Mohan Kumar, P. (2019). Medical diagnosis of cerebral palsy rehabilitation using eye images in machine learning techniques. *Journal of Medical Systems*, *43*(8), 278.

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An empirical exploration of recurrent network architectures. *In International Conference on Machine Learning* (pp. 2342-2350). PMLR.

Jian, S., Kaiming, H., Shaoqing, R., & Xiangyu, Z. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision & Pattern Recognition* (pp. 770-778).

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Liu, T., Wang, K., Sha, L., Chang, B., & Sui, Z. (2018, April). Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 10012-10022).

Lee, K., Kim, W., & Lee, S. (2022). From human pose similarity metric to 3D human pose estimator: Temporal propagating LSTM networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(2), 1781-1797.

Lin, Y., Jiao, X., & Zhao, L. (2023). Detection of 3D human posture based on improved MediaPipe. Journal of Computer and Communications, 11(2), 102-121.

Li, F., Zhang, Y., Yan, W., Klette, R. (2016) Adaptive and compressive target tracking based on feature point matching. International Conference on Pattern Recognition (ICPR), (pp.2734-2739).

Liu, C., Yan, W. (2020) Gait recognition using deep learning. Handbook of Research on Multimedia Cyber Security (pp.214-226)

Liu, Y., Nand, P., Hossain, A., Nguyen, M., Yan, W. (2023) Sign language recognition from digital videos using feature pyramid network with detection Transformer. *Multimedia Tools and Applications*.

Lu, J. (2016) *Empirical Approaches for Human Behavior Analytics*. Master's Thesis. Auckland University of Technology, New Zealand.

Lu, J., Shen, J., Yan, W., Boris, B. (2017) An empirical study for human behaviors analysis. *International Journal of Digital Crime and Forensics*, 9 (3), 11-17.

Lu, J., Nguyen, M., Yan, W. (2018) Pedestrian detection using deep learning. IEEE AVSS.

Lu, J., Nguyen, M., Yan, W. (2020) Human behavior recognition using deep learning. International Conference on Image and Vision Computing New Zealand.

Lu, J., Nguyen, M., Yan, W. (2020) Comparative evaluations of human behavior recognition using deep learning. *Handbook of Research on Multimedia Cyber Security*, 176-189.

Lu, J., Nguyen, M., Yan, W. (2021) Sign language recognition from digital videos using deep learning methods. *International Symposium on Geometry and Vision.*

Lu, J. (2021) *Deep Learning Methods for Human Behavior Recognition.* PhD Thesis. Auckland University of Technology, New Zealand.

Mao, W., Ge, Y., Shen, C., Tian, Z., Wang, X., & Wang, Z. (2021). TFPose: Direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*.

Neuman, B., & Gray, R. (2013). A direct comparison of the effects of imagery and action observation on hitting performance. Movement & Sport Sciences-Science & Motricité, (79), 11-21.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. *In European Conference on Computer Vision* (pp. 483-499). Springer International Publishing.

Nie, B. X., Wei, P., & Zhu, S. C. (2017). Monocular 3d human pose estimation by predicting depth on joints. *In IEEE International Conference on Computer Vision (ICCV)* (pp. 3467-3475).

Özgür, A., & Nar, F. (2020). Effect of dropout layer on classical regression problems. In Signal Processing and Communications Applications Conference (SIU) (pp. 1-4).

Parmar, P., & Tran Morris, B. (2017). Learning to score Olympic events. *In IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 20-28).

Pfeiffer, M., Paolo, G., Sommer, H., Nieto, J., Siegwart, R., & Cadena, C. (2018). A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments. *In IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5921-5928).

Pan, C., Yan, W. (2018) A learning-based positive feedback in salient object detection. International Conference on Image and Vision Computing New Zealand.

Pan, C., Yan, W. (2020) Object detection based on saturation of visual perception. Multimedia Tools and Applications, 79 (27-28), 19925-19944.

Pan, C., Liu, J., Yan, W., Zhou, Y. (2021) Salient object detection based on visual perceptual saturation and two-stream hybrid networks. IEEE Transactions on Image Processing.

Ramanan, D., Forsyth, D. A., & Zisserman, A. (2006). Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1), 65-81.

Rossi, A., Pappalardo, L., & Cintia, P. (2021). A narrative review for a machine learning application in sports: an example based on injury forecasting in soccer. Sports, 10(1), 5.

Siddiqui, H. U. R., Younas, F., Rustam, F., Flores, E. S., Ballester, J. B., Diez, I. D. L. T., ... & Ashraf, I. (2023). Enhancing cricket performance analysis with human pose estimation and machine learning. Sensors, 23(15), 6839.

Sun, X., Wu, P., & Hoi, S. C. (2018). Face detection using deep learning: An improved Faster R-CNN approach. Neurocomputing, 299, 42-50.

Song, S., Lan, C., Xing, J., Zeng, W., & Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. *In AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27.

Shen, J., Yan, W., Miller, P., Zhou, H. (2010) Human localization in a cluttered space using multiple cameras. IEEE International Conference on Advanced Video and Signal Based Surveillance.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Tang, B., & Matteson, D. S. (2021). Probabilistic transformer for time series analysis. Advances in Neural Information Processing Systems, 34, 23592-23608.

Toshev, A., & Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. *In IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1653-1660).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wang, J., Tan, S., Zhen, X., Xu, S., Zheng, F., He, Z., & Shao, L. (2021). Deep 3D human pose estimation: A review. Computer Vision and Image Understanding, 210, 103225.

Wu, Q., Xu, G., Zhang, S., Li, Y., & Wei, F. (2020). Human 3D pose estimation in a lying position by RGB-D images for medical diagnosis and rehabilitation. In Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (pp. 5802-5805).

Wang, X., Yan, W. (2019) Human gait recognition based on self-adaptive hidden Markov model. IEEE/ACM Transactions on Biology and Bioinformatics.

Wang, X., Yan, W. (2019) Cross-view gait recognition through ensemble learning. Neural Computing and Applications.

Wang, X., Yan, W. (2019) Gait recognition using multichannel convolutional neural networks. Neural Computing and Applications.

Wang, X., Yan, W. (2019) Multi-perspective gait recognition based on ensemble learning. Springer Neural Computing and Applications.

Wang, X., Yan, W. (2019) Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory. International Journal of Neural Systems.

Wang, X., Yan, W. (2020) Non-local gait feature extraction and human identification. Springer Multimedia Tools and Applications.

Wang, X., Yan, W. (2020) Cross-view gait recognition through ensemble learning. Neural Computing and Applications 32 (11), 7275-7287.

Wang, X., Yan, W. (2022) Human identification based on gait manifold. Applied Intelligence.

Wöllmer, M., Kaiser, M., Eyben, F., Schuller, B., & Rigoll, G. (2013). LSTM-modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 31(2), 153-163.

Xiao, B., Wu, H., & Wei, Y. (2018). Simple baselines for human pose estimation and tracking. *In European Conference on Computer Vision (ECCV)* (pp. 466-481).

Yan, W. (2023) Computational Methods for Deep Learning. Springer.

Yan, W. (2019) Introduction to Intelligent Surveillance. Springer.

Yu, Q., & Guo, H. (2022). Sports medicine image modeling for injury prevention in basketball training. Contrast Media & Molecular Imaging, 2022.

Yu, Z. (2021) Deep Learning Methods for Human Action Recognition. Master's Thesis, Auckland University of Technology, New Zealand.

Yu, Z., Yan, W. (2020) Human action recognition using deep learning methods. International Conference on Image and Vision Computing New Zealand.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural Computation, 31(7), 1235-1270.

Zhi-chao, C., & Zhang, L. (2019). Key pose recognition toward sports scene using deeply-learned model. *Journal of Visual Communication and Image Representation*, 63, 102571.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *In ECCV* (pp. 818-833).

Zhu, Y., Yan, W. (2022) Ski fall detection from digital images using deep learning. ACM

ICCCV.

Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., & Ding, Z. (2021). 3D human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 11656-11665).