CYBERNETICS AND PHYSICS, VOL. 13, NO. 2, 2024, 161–167

A DEEP LEARNING FRAMEWORK FOR GYM-GESTURE RECOGNITION USING THE COMBINATION OF TRANSFORMER AND 3D POSE ESTIMATION

Hung Le Viet*
Department of Information Technology
The University of Danang,
University of Science and Technology
Vietnam
hungle.dut@gmail.com

Han Le Hoang Ngoc
Department of Information Technology
The University of Danang,
University of Science and Technology
Vietnam
hanlehngoc080601@gmail.com

Khoa Tran Dinh Minh
Department of Information Technology
The University of Danang,
University of Science and Technology
Vietnam
mkhoatd@gmail.com

Son Than Van Hong
Department of Information Technology
The University of Danang,
University of Science and Technology
Vietnam
sontvh2002@gmail.com

Article history:
Received 25.07.2024, Accepted 16.09.2024

Abstract

In recent years, the gym has become a vital part of social life, with many people hiring personal trainers to monitor their form and enhance their fitness more effectively. To support this trend, we propose an innovative deep learning framework to recognize 22 different types of gym gestures, aimed at developing AI-powered personal trainer applications. Our framework identifies gym gestures based on video in two stages. The first stage is a feature extraction module that employs a 3D pose estimation model to extract skeleton data. The second stage is a fusion module that classifies this skeleton data. Utilizing a Transformer-based fusion module, our framework achieves an impressive 91.07% accuracy and an 88.94% F1-score. This method is both fast and effective, demonstrating significant potential for real-world applications.

Key words

Transformer, Attention mechanism, 3D pose estimation, Deep Learning, Gym-gesture Regconition

1 Introduction

Nowadays, as gym workouts become more popular in everyday life, a rising number of individuals are engaging personal trainers to assist and monitor their activities, which is both cumbersome and expensive. Training a personal trainer to test gym motions is time-consuming and costly. As a result, our goal is to propose an AI model capable of understanding gym gestures in order to create an application that serves as an AI Personal Trainer. In recent years, various deep learning architectures for human action recognition have been developed. In this study, we realize the potential of deep learning for recognizing gym movements. Numerous investigations on gym-gesture recognition have been conducted in the past, including those by Lanza Bernardo et al. [Lanza et al., 2022] and Preetham Ganesh et al. [Ganesh et al., 2020] . Using Convolutional Neural Network architecture to extract features from videos is a prominent strategy in video classification [Ur Rehman et al., 2023]. CNN architecture is used to extract feature vectors from video input in recent studies by Sadia Kiran et al. [Kiran et al., 2021], Zhang et al. [Zhang et al., 2018], Karen Simonyan et al. [Simonyan and Zisserman, 2014], Xiaodong Yang et al. [Yang et al., 2016], Hao Ye et al. [Ye et al., 2015], Shengxin Zha et al. [Zha et al., 2015], and Zuxuan Wu et al. [Wu et al., 2015]. Nevertheless, we observe that these methods are not particularly successful when there are a lot of noisy objects in the input video. Specifically, there is a significant amount of noise in the video from our gym-gesture dataset.

Recently, the posture estimation model has emerged as a potent instrument for skeleton data extraction [Sengupta et al., 2020; Cao et al., 2017a]. During exer-

cise, one can estimate body movements without wearing measuring devices thanks to vision-based deep learning models [Lanza et al., 2022; Cao et al., 2017b]. In this work, we extracted skeleton data from the subject in the movie using the MMPose pretrained model [Sengupta et al., 2020]. We employ this technique because it eliminates noise elements more successfully than the CNN-based approach to feature vector extraction from videos. We treat skeleton data taken from video as time-series data. There are numerous deep learning algorithms typically used in time-series data, including MLP, CNN-1D, RNN, LSTM, and Bi-LSTM. These strategies produce good results with time-series data. Because of its ability to learn from sequence data, the Transformer architecture has recently gained popularity in the domains of NLP and computer vision [Vaswani et al., 2017]. We concluded that the attention mechanism has potential in time-series and sequence data. In the meantime, our study's skeleton data is regarded as time-series data. With this skeleton data, we suggest using a transformer-based design to solve the time-series classification problem. Our proposed transformer-based model performs better than conventional models like MLP, CNN-1D, RNN, LSTM, and Bi-LSTM. The two stages of our suggested design are the Fusion Module and Skeleton Estimation. Skeleton data, which is similar to time-series data, is extracted from videos in the first step. This skeleton data serves as the basis for the categorization function in the second step.

Our research makes the following specific contributions:

Create a separate dataset comprising 22 different gym-gestures with the number of each class ranging from 300 to 1500 samples.

Propose innovative Deep Learning Framework employs a combination of Transformer and 3D Pose Estimation, helping to distinguish gym-gestures in movies. This architecture helps develop an AI Personal Trainer in the future.

Propose Transformer-based model for time-series data with improved accuracy when compared to standard models

2 Related Work

2.1 CNN-based Video Recognition

In recent studies related to video recognition, researchers frequently partition videos into image sequences in chronological order and utilize CNN architectures to extract feature vectors from them. [Ur Rehman et al., 2023]. Sadia Kiran et al [Kiran et al., 2021] classified data using CNN architecture paired with classifiers such as SVM or KNN. Shengxin Zha et al [Zha et al., 2015] used architecture in their research as well. To classify videos, CNN combines with an SVM classifier. Zuxuan Wu et al [Wu et al., 2015] employed CNN architecture and LSTM to classify films. Zhang et al [Zhang et al., 2018], Karen Simonyan et al [Simonyan and Zisserman, 2014], and Hao Ye et al [Ye et al., 2015] have all used two-stream CNN to extract spatial and temporal information from video input. Xiaodong Yang et al. [Yang et al., 2016] suggested 2D and 3D CNN architectures for extracting information from video input. However, we discover that these methods are frequently ineffectual due to the possibility of several distracting items in photos or movies. In our work, we extracted the skeleton data coordinates of a moving object's human pose from a movie using a pretrained model called MMPose [Sengupta et al., 2020], which is a more effective method. When it comes to removing noise components from the technique of extracting feature vectors from videos using CNN. We suggest using 3D Pose Estimation in this study because it extracts characteristics with more information than 2D Estimation [Chen and Ramanan, 2017]. The Pose Estimation approach combines 2D and 3D Estimation.

2.2 Time-series or sequence model

Time-series classification issues can benefit from the effective application of deep learning, according to a recent study [Ismail Fawaz et al., 2019]. It has been demonstrated by recent research that CNN-1D (One Dimensional Convolutional Neural Network), LSTM (Long Short Term Memory), RNN (Recurrent Neural Network), and Bi-LSTM (Bidirectional Long Short Term Memory) perform well on time-series data. In the time-series and sequence classification model problem, models such as MLP, RNN, LSTM, Bi-LSTM, or CNN-1D are frequently utilized. Known as the original neural network model [Ruck et al., 1990], MLP performs best when applied to classification and regression issues; however, it is less effective when applied to time-series or sequence data. Since then, RNN has been able to efficiently evaluate information from sequence or time-series data [Grossberg, 2013], outperforming the MLP model in time-series data analysis. LSTM was developed to solve the problem of short term memory and gradient vanishing. LSTM has overcome the shortcomings of RNN [Cheng et al., 2016]. Recent research have also shown that LSTM outperforms RNN. Bi-LSTM is a sequence processing model made up of two LSTMs: one receives input in the forward direction and one in the reverse direction. Bi-LSTM effectively increases the quantity of information available on the network, which improves the context provided to the algorithm [Shahid et al., 2020]. RNN, LSTM, and Bi-LSTM models are frequently employed in natural language processing (NLP) difficulties. CNN-1D is a version of CNN that uses one-dimensional convolution. This method works well with time-series or sequence data. The attention mechanism has recently found significant success in text and picture classification models [Van Huy L, 2023; Van Hieu et al., 2023], owing to its ability to learn associations between features on a sequence dataset. From there, we understood that the attention mechanism would outperform earlier time-series models, therefore we used

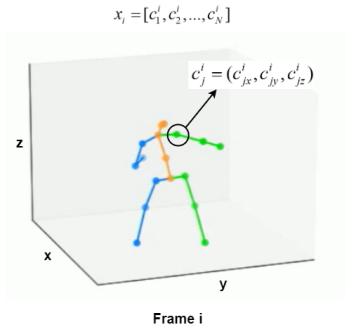

Figure 2. Graphical representation of each coordinate point of skeleton at each frame

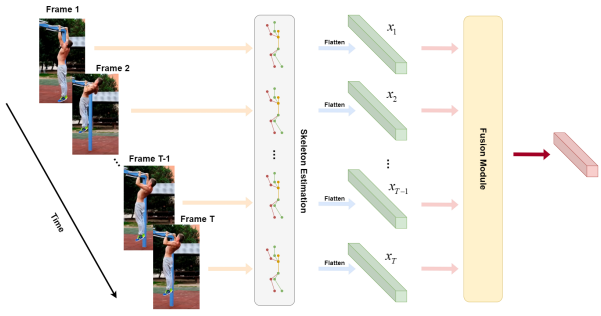

Figure 3. General architecture of our proposed framework

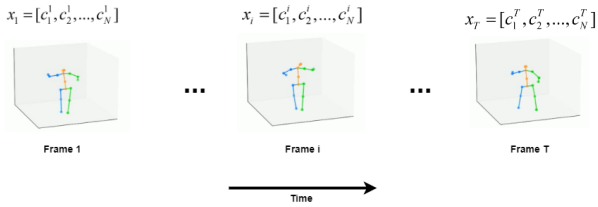

Figure 1. Graphical representation of skeleton at each frame

it in our work. In this study, we employ the MMPose pretrained architecture to extract skeleton data from a human in a movie. This skeleton data is classified as a time-series data.

3 Methodology

3.1 Skeleton-based architecture

In this study, we propose a general architecture for the gym-gesture classification model, including 2 stages: Skeleton Estimation and Fusion Module. The former has the function of extracting skeleton data from the video. This skeleton data is like time-series data. The latter has the function of synthesizing this skeleton data and classifying it. Skeleton Estimation Stage is based on the MMPose pre-trained model to extract the coordinate vectors of a person's skeleton data over time.

We use the pretrained model MMPose to extract a Tensor from the video $X = [x_1, x_2, ..., x_T]$, where $X \in \mathbb{R}^{T\times N\times 3}$ and $x_{i\in\{1,2,...,T\}} \in \mathbb{R}^{N\times 3}$ is the matrix containing the coordinate vectors of the skeleton data at the second frame i, where T is the number of frames that the video is divided into, and is visually represented as in Figure 1. Each $x_i = \left[c_1^i, c_2^i, \ldots, c_N^i\right]$ contains values $C^i_{j\in\{1,2,...,N\}} \in \mathbb{R}^3$ that are the 3D coordinates of each point in the skeleton data at frame i, with N the number of coordinate points of the skeleton. c_j^i is represented by the 3D coordinate formula $c_j^i = \left(c_{jx}^i, c_{jy}^i, c_{jz}^i\right)$, depicted in Figure 2.

We flatten the matrix x_i into a vector with the purpose of reducing the dimensionality of the Tensor X from 3 dimensions to 2 dimensions: Flatten : $\mathbb{R}^{T\times N\times 3} \rightarrow \mathbb{R}^{T\times 3N}$, this helps reduce the amount of calculation for the model. We then obtain the feature matrix $X = [x_1, x_2, \ldots, x_T] \in \mathbb{R}^{T\times 3N}$, which $x_{i\in\{1,2,...,T\}} \in \mathbb{R}^{3N}$ is a vector of length 3 N. This is then $X = [x_1, x_2, \ldots, x_T] \in \mathbb{R}^{T\times 3N}$ included in the Fusion Module. Our proposed framework architecture is depicted in Figure 3, including 2 stages.

3.2 Transformer-based Fusion Module

Once extracted $X = [x_1, x_2, \ldots, x_T] \in \mathbb{R}^{T\times 3N}$, we consider $x_1, x_2, \ldots, x_T$ as a time-series data. In the experimental results section, we apply traditional architectures such as MLP, RNN, LSTM, Bi-LSTM, CNN-1D to this time-series data, but the accuracy is very low. Recently, the Transformer model emerged for its performance in the LLM model. The Transformer architecture is the key to the success of recent advanced LLM architectures, because it works well thanks to the attention mechanism. Attention Mechanism architecture is also strongly applied in text or image classification problems [5,6]. Attention Mechanism works well thanks to its ability to learn on time-series or sequence data sets better than traditional CNN models. The formula for scaled dot-product attention is expressed as follows

$$\text{Attention}(Q, K, V) = \text{soft}\max\left(\frac{Q\cdot K^T}{\sqrt{D}}\right)\cdot V \quad (1)$$

With: $Q = I.W^Q; K = I.W^K; V = I.W^V; I \in \mathbb{R}^{N\times D_1}; W^Q, W^K \in \mathbb{R}^{D_1\times D}; W^V \in \mathbb{R}^{D_1\times D_v}$. In particular, I is the input of that attention class. The multihead attention architecture is built by concatenating the output of multiple attention layers together and then passing them through a linear layer with the weights $W^O \in \mathbb{R}^{hD_v\times D_I}$ illustrated in Figure 4.

Before introducing the Transformer block architecture, we used positional encoding to capture sequence data efficiently [Vaswani et al., 2017]. With input $X \in \mathbb{R}^{T\times 3N}$ is a matrix representing the coordinates of skeleton data, we use $P = [p_1, p_2, \ldots, p_T] \in \mathbb{R}^{T\times 3N}$ the matrix to perform positional encoding, this matrix has the same shape as input X :

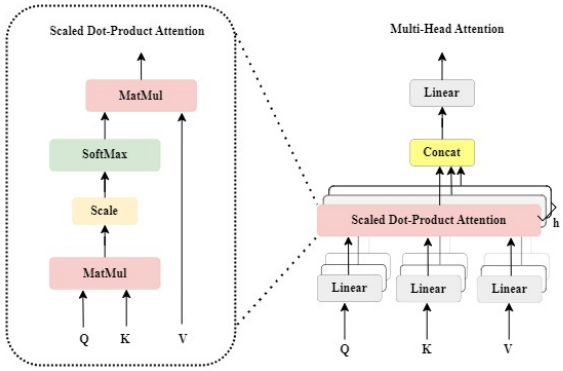

Figure 4. (Left) Attention architecture (Right) Multihead Attention

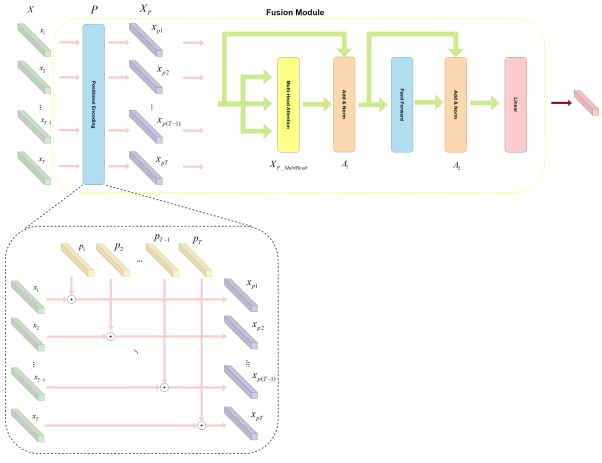

Figure 5. Our proposed transformer-based Fusion Module

$$P_{i,2j} = \sin\left(\frac{i}{10000^{2j/3N}}\right) \tag{2}$$

$$P_{i,2j+1} = \cos\left(\frac{i}{10000^{2j/3N}}\right) \tag{3}$$

With $i = 0, 1, \ldots, l-1$ and $j = 0, 1, \ldots, \lfloor (d-1)/2 \rfloor$

Because video data of gym movements is cyclical over time like time-series data, using positional encoding helps to better represent and capture this series of information. To implement positional encoding, we perform the calculation $X_P = P + X \in \mathbb{R}^{T \times 3N}$, where X_P is the input matrix after having positional encoding applied. From there, it is possible to capture the sequential information effectively, and maintain the position information of the input sequence. Then, X_P is inserted into a Transformer Block with an architecture similar to the architecture of the Transformer Encoder [Vaswani et al., 2017], described as Figure 5.

Once entered X_P into Multi-Head Attention, the output is calculated by the formula:

$$X_{P\_\text{MultiHead}} = \text{MultiHead}\ (X_P)$$

$$= (\ \text{head}_1,\ \text{head}_1, \ldots,\ \text{head}_h) \cdot W^O \tag{4}$$

Where: $\text{head}_i = \text{Attention}_i\left(Q^i, K^i, V^i\right); Q^i = C.W_i^Q, K^i = C.W_i^K, V^i = C.W_i^V$, with $i \in \{1, 2, \ldots, h\}$. Which h is the number of attention layers of the Multi-Head Attention layer. The output $X_{P\_\text{MultiHead}}$ is then passed through the Add&Norm class, Feed-Forward (FFN) and another Add&Norm class:

$$A_1 = \text{Norm}\left(X_p + X_{p\_\text{MultiHead}}\right); \tag{5}$$

$$A_2 = \text{Norm}\left(A_1 + FFN\left(A_1\right)\right); \tag{6}$$

$$\text{Output} = \text{Linear}\left(A_2\right) \tag{7}$$

The final output value is the probability vector with length 22, corresponding to 22 classes (22 different types of gym gestures).

3.3 Undersampling method

In deep learning, the undersampling method is a technique used to address class imbalance in datasets, where some classes have significantly more samples than others. The imbalanced class can negatively impact model performance because the model may become biased toward the majority class, neglecting the minority classes. Undersampling method seeks to deal with this issue by reducing the number of samples from the majority class to better balance the dataset. The undersampling method is depicted in Figure 6

4 Experimental Results

Dataset: The dataset we use includes 22 classes, which are 22 different gym movements, each class includes about 300-1500 samples. We collect data from many sources on the internet and capture videos at gym centers, ensuring a rich diversity of data. With such a large dataset, it helps us evaluate more objectively and put it into practical applications more effectively. The statistical data of the dataset is depicted in Figure 7 and Table 1. A video sample captured by us at the gym center is depicted in Figure 8.

Preprocessing: The raw data contains noise and has very long videos, which negatively impacts training. To address this, we implement the data preprocessing method, depicted in Figure 9. Our preprocessing steps are carried out as follows:

Step 1: We remove noise from the raw video data, specifically the segments that contain activities unrelated to gym gestures.

Gym gesture	Hammer Curl	Bench Press	Pull Up	Tricep Dips	Leg Extension	Leg Pull-down	Incline Bench Press	Leg Raises	Lateral Raise	T Bar Row	Plank
Sample size	642	496	489	352	508	375	499	584	430	555	1441
Gym gesture	Romanian Dead-lift	Shoulder Press	Decline Bench Press	Push Up	Chest Fly Machine	Hip Thrust	Barbell Biceps Curl	Russian Twist	Tricep Push-down	Deadlift	Squat
Sample size	513	348	763	552	588	494	676	472	317	511	753

Table 1. The sample size of our dataset's classes

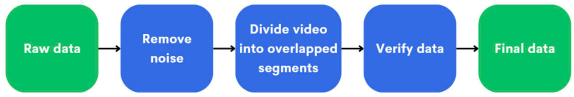

Figure 9. The steps of our preprocessing method

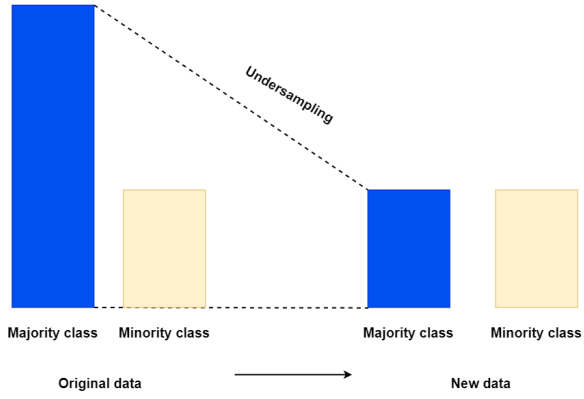

Figure 6. Undersampling method

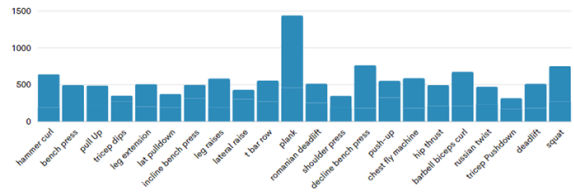

Figure 7. The statistical column chart of our dataset

Figure 8. A video sample is captured by us at the gym center (Shoulder Press)

Step 2: We split the video into segments with a length of around 12 seconds. Each segment can be overlapped to increase the sample size.
Step 3: We verify the data after splitting.

We aim to remove data unrelated to the gym gesture and split the very long video into short videos

Training Strategy: In the experiment, we divided the training and test sets in a ratio of 7:3. The optimizer algorithm we use during training is Adam Optimizer with learning rate 0.001. The number of frames that we use in the experiment are $T = 60$. For the Transformer-based Fusion Module, we utilize the multihead-attention layer with $h = 4$ attention layers. All models are trained with 100 epochs.

Evaluation: The results are described in Table 2, we use 4 metrics: Recall, Precision, F1-score, Accuracy to demonstrate objective comparison between models. The model using MLP for Fusion Module achieved the lowest accuracy at 25.32%. However, the models using CNN-1D, RNN, LSTM, Bi-LSTM are not significantly higher than MLP, only about 1-8%. Meanwhile, our proposed Transformer architecture for Fusion Module outperforms the remaining models, with accuracy up to 91.07%, and F-1 Score 88.94%.

Undersampling method: Our dataset has a slight imbalance, with the sample size of "Plank" class being two to three times larger than the other classes. To save computational resources, we prefer utilizing the Undersampling method which reduce the sample size of the "Plank" class by around 50%. By using the Undersampling method, we notice an improvement of F1-score of all models except for 3D Estimation + RNN, depicted in Table 3.

Table 3. The comparison of our proposed architectures with Undersampling method.

Model	Recall (%)	Precision (%)	F1-Score (%)	Accuracy (%)
3D Estimation + MLP	26.2	26.77	**25.01**	25.2
3D Estimation + CNN-1D	24.89	24.13	**25.79**	26.17
3D Estimation + RNN	31.00	29.89	28.76	30.73
3D Estimation + LSTM	31.14	27.82	**30.88**	31.53
3D Estimation + Bi-LSTM	31.65	33.93	**31.74**	32.09
3D Estimation + Transformer	**91.30**	**90.98**	**90.14**	**90.65**

Table 2. The comparison of our proposed architectures.

Model	Recall (%)	Precision (%)	F1-Score (%)	Accuracy (%)
3D Estimation + MLP	27.12	26.11	24.23	25.32
3D Estimation + CNN-1D	25.11	23.33	25.19	26.89
3D Estimation + RNN	31.12	30.83	28.91	30.23
3D Estimation + LSTM	30.95	26.97	29.83	32.49
3D Estimation + Bi-LSTM	30.87	32.23	29.74	33.22
3D Estimation + Transformer	**91.82**	**90.77**	**88.94**	**91.07**

5 Conclusion

In conclusion, we present a novel deep learning framework in this study that aids in the identification of gym gestures. This architecture facilitates the development of AI Personal Trainer applications. The two stages of our suggested deep learning framework are the Fusion Module and Skeleton Estimation. The purpose of the first step is to extract skeleton data, which is similar to time-series data, from the video. The second stage synthesizes and classifies the skeleton data. In the experimental section, we present a Transformer-based Fusion Module for our deep learning framework. This architecture outperforms deep learning frameworks utilizing MLP, RNN, LSTM, Bi-LSTM, and CNN-1D for Fusion Module, with an accuracy of 91.07%. Experimental results suggest that our strategy is effective in terms of accuracy and training time, indicating that it will be practically applied in the future. Furthermore, we built a dataset of 22 classes matching to 22 gym-gestures from various online and offline sources. In the future, we will create a dataset with more classifications with the goal of using it in actual applications.

References

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

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

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

Cheng, J., Dong, L., and Lapata, M. (2016). Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 551–561.

Ganesh, P., Idgahi, R. E., Venkatesh, C. B., Babu, A. R., and Kyrarini, M. (2020). Personalized system for human gym activity recognition using an rgb camera. In *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 1–7.

Grossberg, S. (2013). Recurrent neural networks. *Scholarpedia*, **8** (2), pp. 1888.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data mining and knowledge discovery*, **33** (4), pp. 917–963.

Kiran, S., Khan, M. A., Javed, M. Y., Alhaisoni, M., Tariq, U., Nam, Y., Damaševičius, R., and Sharif, M. (2021). Multi-layered deep learning features fusion for human action recognition. *Computers, Materials & Continua*, **69** (3).

Lanza, B., Nuzzi, C., Pasinetti, S., Lancini, M., et al. (2022). Deep learning for gesture recognition in gym training performed by a vision-based augmented reality smart mirror. In *ISBS Proceedings Archive*, vol. 40, pp. 363–366.

Ruck, D. W., Rogers, S. K., and Kabrisky, M. (1990). Feature selection using a multilayer perceptron. *Journal of neural network computing*, **2** (2), pp. 40–48.

Sengupta, A., Jin, F., Zhang, R., and Cao, S. (2020). mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns. *IEEE Sensors Journal*, **20** (17), pp. 10032–10044.

Shahid, F., Zameer, A., and Muneeb, M. (2020). Predictions for covid-19 with deep learning models of lstm, gru and bi-lstm. *Chaos, Solitons & Fractals*, **140**, pp. 110212.

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, **27**.

Ur Rehman, A., Belhaouari, S. B., Kabir, M. A., and Khan, A. (2023). On the use of deep learning for video classification. *Applied Sciences*, **13** (3), pp. 2007.

Van Hieu, N., Hien, N. L. H., Van Huy, L., Tuong, N. H., and Thoa, P. T. K. (2023). Plantkvit: A combination model of vision transformer and knn for forest plants classification. *JUCS: Journal of Universal Computer Science*, **29** (9).

Van Huy L, Hien NLH, P. N. H. N. V. (2023). Deep learning model with hierarchical attention mechanism for sentiment classification of vietnamese comments. *Cybernetics and Physics*, **12** (2), pp. 111—20.

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

Wu, Z., Wang, X., Jiang, Y.-G., Ye, H., and Xue, X. (2015). Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 461–470.

Yang, X., Molchanov, P., and Kautz, J. (2016). Multilayer and multimodal fusion of deep neural networks for video classification. In *Proceedings of the 24th ACM international conference on Multimedia*, pp. 978–987.

Ye, H., Wu, Z., Zhao, R.-W., Wang, X., Jiang, Y.-G., and Xue, X. (2015). Evaluating two-stream cnn for video classification. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pp. 435–442.

Zha, S., Luisier, F., Andrews, W., Srivastava, N., and Salakhutdinov, R. (2015). Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv preprint arXiv:1503.04144*.

Zhang, H., Liu, D., and Xiong, Z. (2018). Convolutional neural network-based video super-resolution for action recognition. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, IEEE, pp. 746–750.