*Journal of Computer Science and Cybernetics, V.39, N.4 (2023), 343–368*
DOI no. 10.15625/1813-9663/18043

# TAEKWONDO POSE ESTIMATION WITH DEEP LEARNING ARCHITECTURES ON ONE-DIMENSIONAL AND TWO-DIMENSIONAL DATA

DAT TIEN NGUYEN, CHAU NGOC HA, HA THANH THI HOANG,
TRUONG NHAT NGUYEN, TUYET NGOC HUYNH, HAI THANH NGUYEN*

*College of Information and Communication Technology, Can Tho University, 3/2 Street, Ninh Kieu District, Can Tho City, Viet Nam*



**Abstract.** Practicing sports is an activity that helps people maintain and improve their health, enhance memory and concentration, reduce anxiety and stress, and train teamwork and leadership ability. With the development of science and technology, artificial intelligence in sports has become increasingly popular with the public and brings many benefits. In particular, many applications help people track and evaluate athletes' achievements in competitions. This study extracts images from Taekwondo videos and generates skeleton data from frames using the Fast Forward Moving Picture Experts Group (FFMPEG) technique using MoveNet. After that, we use deep learning architectures such as Long Short-Term Memory Networks, Convolutional Long Short-Term Memory, and Long-term Recurrent Convolutional Networks to perform the poses classification tasks in Taegeuk in Jang lessons. This work presents two approaches. The first approach uses a sequence skeleton extracted from the image by Movenet. Second, we use sequence images to train using video classification architecture. Finally, we recognize poses in sports lessons using skeleton data to remove noise in the image, such as background and extraneous objects behind the exerciser. As a result, our proposed method has achieved promising performance in pose classification tasks in an introductory Taekwondo lesson.

**Keywords.** Pose classification, skeleton, sports lessons, Taekwondo.

## 1. INTRODUCTION

The emergence of information technology has brought many positive effects on education development. The development of information technology, especially the Internet, has opened up an incredibly diverse and rich knowledge base for learners and teachers, making knowledge acquisition easier and higher in quality teaching and learning. Information technology enables learners to learn and acquire knowledge flexibly and conveniently. People can teach themselves anytime, anywhere. For example, the Covid-19 pandemic has created a new urgency for education, forcing a shift toward facilitating learning and improving learn-

*Corresponding author.
*E-mail addresses*: ntiendat.it@gmail.com (D.T. Nguyen); hnchau.it@gmail.com (C.N. Ha); htth2514@gmail.com (H.T.T. Hoang); itnntruong@gmail.com (T.N. Nguyen); hngoctuyet031001@gmail.com (T.N. Huynh); nthai.cit@ctu.edu.vn (H.T. Nguyen).

ing performance by creating, using, and managing appropriate technological processes and resources.

Taekwondo is a traditional Korean martial art. It is a martial art and a cultural heritage that introduces Korean culture to the world. It has become a popular sport worldwide as it is one of the few martial arts that has become an official competition of the world's largest sporting event, the Olympic Games. They are martial arts in which one attacks or defends with hands and feet anytime or anywhere, with the occasional weapon. There are many benefits to using Taekwondo in education. The first benefit is that practicing Taekwondo has a good effect on improving the practitioner's health. Exercise suits the cardiovascular system, increasing heart health and stimulating blood circulation. The next benefit is that we can protect ourselves, especially girls. Although it brings many benefits, if we practice the wrong technique, it will cause joint injuries when practicing. In addition, there is a disadvantage related to the close contact between teachers and learners. This will cause embarrassment and be challenging to achieve.

In recent years, human action recognition (HAR) has been given particular attention by the computer vision community. For example, Vishwakarma et al. [1] provided a comprehensive survey of HAR methods developed from 2008 to 2012. Those HAR methods were divided into three different levels: human detection (low-level vision), human tracking (intermediate-level vision), and behavior understanding methods (high-level vision). HAR covers many research topics in computer vision, including detecting people in videos, estimating human poses, tracking people, and analyzing and understanding time-series data. HAR has applications in different fields, such as surveillance, [2, 3] security, human-computer interaction, automatic video annotation, health monitoring [4], education, smart home, etc. HAR aims to understand human behavior and assign a label to each action. Various data methods for representing human behavior, such as RGB, skeleton, depth, infrared, point cloud, and WiFi signals, encode useful but different information from different sources and have different advantages depending on the application scenario mentioned in [5].

In this study, we propose deep learning architectures and compare their accuracy and efficiency. Our approach is inspired by the desire of practitioners to receive their assessment results after each Taekwondo exercise. We collected videos of different people practicing, then used FFMPEG to cut them into frames to get images of Taekwondo moves. Then, we proceed to group the same poses. From the layered images, we use MoveNet to extract the skeleton features of the movements and save it as a file (.csv) from which we divide and combine the data. Machine learning can learn those features through deep learning methods. Choosing the appropriate deep-learning method for the data set is essential to identify the movements correctly. The following are the main contributions of the research:

- We collected and analyzed videos of a fundamental Taekwondo lesson to provide a workflow for a 16-pose recognition task based on skeleton extraction from collected videos using some deep learning architectures. In addition, we provided a public dataset at [1] for further analyzing poses in a Taekwondo lesson.

- We compared two approaches for human behaviors in the video (with a specific application in pose classification task) using deep learning architectures on 1D (skeleton) and 2D (image) data. We used MoveNet to extract a skeleton from 2-dimensional (2D)

[1] https://github.com/thnguyencit/pose-classification

images into 1D data. On 1D data, the Taekwondo pose classification task was realized faster than image classification on 2D images. Nevertheless, it achieved a rather good average accuracy on the test set. In addition, skeleton-based action data may focus on pose motion and orientation. At the same time, images may contain noises such as gender, background, skin color, etc., that can cause interference in image classification tasks.

- From the experiments, we also found the influence of the number of consecutive skeleton sequences on the pose classification tasks.
- When evaluated on the KTH set, our approach achieved outstanding results compared to Local features classified by SVM. In addition, our method obtained better performance than another study using Gated Recurrent Neural Networks. In addition, our results were approximate with another work with the Gaussian mixture model (GMM) and Kalman filter (KF) on a public human-action recognition dataset.

The rest of this paper is organized as follows. First, we will discuss several related studies in Section 2. After that, we will explain our workflow for pose classification tasks in Section 3. Then, our experiment is introduced in Section 4. Finally, we will summarize our research's main features and directions in Section 5.

## 2. RELATED WORK

Video-based action recognition technique is quite popular. It is applied in many fields: in security and surveillance, the security camera will issue a warning about the danger or suspicious behavior of the object it has recorded; in education, pose classification in a Taekwondo lesson will support people to learn Taekwondo easily [6]; in the field of smart utilities, smart television (smart TV) is a typical example, it can recognize human face interactions to make suitable program recommendations, even recognize whether people are using or not to implement power saving mode, etc.

### 2.1. Deep learning human action recognition

HAR tries to understand human actions and assign labels to each action. Deep learning HAR can be considered the king of video analysis problems due to its wide applications and analytical challenges, such as background complexity, variation in the camera viewpoint, execution rate, movement of humans, etc. It has a wide range of applications, thus attracting more and more attention in computer vision. Researchers have proposed many architectures to recognize human actions. Convolutional Neural Networks (CNN) are becoming popular in the community. It does not require domain-specific expertise, as it can extract features automatically, as mentioned in [7–9]. Despite their robustness and efficiency, CNN-based methods can only be used for stationary and short-sequence classification problems and are not recommended for long and complex time-series data problems.

The fourth industrial revolution has generated a large amount of data, so many benchmark datasets have been formed, such as UCF Sports [10], HMDB51 [11], JHMDB, Weizmann [12], MPII [13], etc. In the study [14], authors proposed two efficient methods for action recognition based on the two-stream convolutional network by reducing the computational

cost of the temporal stream and providing techniques for assembly in action recognition tasks. The proposed methods have achieved performance on par with the state-of-the-art ones on the datasets of HMDB51 (70.9%) and UCF101 (95.4%). The work [15] proposed a new set of second-order motion representations capable of capturing both the motion's geometrical and kinematic properties. The proposal reduced training times without sacrificing the performance and achieved desirable results on a demanding dataset, namely, the UCF101 recording accuracy rate is 98.45%, and the HMDB51 recording accuracy rate is 80.19%. The highlight of the method is that the preprocessing time is reduced to one-sixth, and the training time is reduced to one-third of the time it would normally take. For action content to be presented more accurately, the authors in [16] divided using the symmetrical properties often in various video scenes to filter out redundant (or background) features on two datasets, HMDB51 and UCF101. In another study [17], scientists proposed a human-related multi-stream CNN (HR-MSCNN) architecture, combining traditional streams with novel human-related streams. HR-MSCNN encodes appearance, motion, and the captured tubes of the human-related regions on the JHMDB, HMDB51, UCF Sports, and UCF101 datasets and achieves desirable results on these datasets. The work [18] proposed a novel approach to human action recognition based on a pre-trained deep CNN model for feature extraction and representation, followed by a hybrid Support Vector Machine (SVM) and K-Nearest Neighbors classifier (KNN) classifier for action recognition on UCF Sports and KTH datasets. In [19], researchers proposed recognizing human actions based on motion sequence information in RGB-D video using deep learning on four datasets: NATOPS gesture, SBU Kinect interaction, MIVIA action, and Weizmann. The proposed method has achieved an average recognition rate of 72.58% on NATOPS gesture, 90.98% on SBU Kinect interaction, 93.37% on MIVIA action, and 100% on the Weizmann dataset. The authors realized that the optical flow and motion history image in the human action recognition problem is expensive. The work [20] proposed a new motion estimation technique named image Weber motion history image (WMHI). This technique is tested on five benchmark datasets: JHMDB, MPII, Sub-JHMDB, HMDB51, and UCF101. WMHI's algorithm saves more than 50% disk space and executes 99 times faster than other CNN algorithms. In the study [9], authors used dense binary SIFT flow-based two-stream CNN over the UCF101 dataset instead of the optical flow, which limits optical flow constraints, such as brightness, constancy, and piecewise smoothness. In another study [21], scientists proposed a novel region sequence-based six-stream CNN feature for human action recognition in videos, combining different scales of image appearance information and motion information for human pose estimation in video sequences. This approach uses a human body part position as prior knowledge and better uses spatial image information, distinguishing fine-grained activities. The work [22] proposed a novel action by processing video data using Convolutional Neural Networks (CNN) and Deep Bidirectional LSTM (DB-LSTM) networks on three benchmark datasets, including UCF-101, YouTube 11 Actions, and HMDB51. First, in-depth features are extracted from every six video frames, which helps reduce redundancy and complexity. Next, a DB-LSTM network is used to learn the order information between frame features, stacking multiple layers together in the forward and reverse passes of the DB-LSTM to increase its depth. The study [23] leveraged a Long-Term Short-Term Memory (LSTM) network with a Biologically Inspired Algorithm (BIA) framework to model monitoring effects with discriminant temporal signals, which recognizes the athlete's actions and motivates one to improve athletic skills.

Through the investigation of previous studies, we found similarities between our study and the study of Shuiwang Ji et al. [24]. Although, in their study, they suggested using a contiguous frame sequence. They did not recommend a feature extractor; instead, they extracted features from contiguous frames next to each other to obtain features close to each other. 3D CNN model is built including three convolution layers, two subsampling layers, and one complete connection layer, and this model has made a significant contribution to behavior classification. 3DCNN extracts features from spatial and temporal dimensions by performing 3D textures, thus obtaining motion information encoded in multiple contiguous frames. In addition, it has also evolved to generate multiple channels of information from input frames and represent the final feature that combines information from all channels. This CNN architecture generates multiple channels of information from adjacent video frames and performs (convolution and subsampling separately) in each channel. The final feature is obtained by combining all the features of all layers. The 3D CNN models were proposed by augmenting the model with output computed as high-level motion features. Moreover, 3D CNN models are limited to a small number of contiguous video frames. Input model Normalization makes the number of trainable parameters increase as the size of the input window increases, proposing to compute motion features from a large number of frames and normalize 3D CNN models using these motion features as ancillary outputs. This model has contributed an imposing result when evaluated on the KTH dataset with more than 0.90 accuracy.

Meanwhile, the authors in [25] proposed the Independent Subspace Analysis algorithm for learning invariant spatiotemporal features from unlabeled data. This method performed surprisingly well with deep learning techniques like stacking and convolution to learn hierarchical representations. Using the basic Independent Subspace Analysis algorithm, they will extract and learn features from the still image. Independent subspace analysis (ISA) was an unsupervised learning algorithm that took care of learning features from unlabeled images. However, their proposal took time to train for big data, making the model less efficient. To extend the training of models with extensive input data, they designed the PCA and ISA network architecture as subunits for the unsupervised learning task. Through the above proposal, they have achieved remarkable results on the KTH dataset, falling in the range of 91.4%. By performing feature extraction on spatial-temporal features, Piotr Dollar [26] proposed a spatiotemporal feature detector for our behavior recognition framework directly on the image to obtain different features on the so-called Cuboids. After the Cuboids are extracted, the original clip will be discarded, and each cuboid will be mapped to the nearest vector. With this method, Piotr Dollar and colleagues achieved a relatively good accuracy of 0.812 on the KTH dataset. Juan Carlos Niebles et al. [27] proposed unsupervised associative learning for human postural recognition. Their algorithm automatically learned data recorded from human actions, which they do through the probabilistic Latent Semantic Analysis (pLSA) model. Through learning features on unlabeled data, there was a specific effect. Using a feature extractor (space-time interest points detector), they will extract local space-time regions. These regions were then clustered into a video set called (codebooks). This approach can deal with noisy feature points arising from dynamic backgrounds and moving cameras by applying probabilistic models. They achieved a particular effect with an accuracy of 81.5% on the KTH dataset.

Our proposed approach is similar to that of Shuiwang Ji [24] in the evaluation and classi-

fication of adjacent frames. However, our method proposes more skeleton extraction on the image adjacent to each other to obtain close features. Furthermore, Juan Carlos Niebles et al. [27] used a space-time interest points detector to extract features from the data, compare the obtained results, and propose a Movenet skeleton extractor as the Black feature extraction is more efficient than that proposed by Juan Carlos Niebles et al. [27]. Therefore, we have contributed significantly to proposing a new feature extraction method and model evaluation on the proposed deep learning network. The authors achieved amazing performance on KTH datasets by applying 3DCNN network architectures and performance evaluation on consecutive frame sequences with more than 90%. Meanwhile, Quoc V. Le and colleagues proposed the Independent Subspace Analysis algorithm for learning invariant spatiotemporal features from unlabeled data and achieved unexpected results on the accuracy obtained at the level 91.4%. By using the probabilistic Latent Semantic Analysis (pLSA) model and learning the features of unlabeled data with certain eciency, Juan Carlos Niebles and colleagues showed the eectiveness of their model with an accuracy of about 81.5%. Piotr Dollar also achieved an accuracy of about 81.2% when they used a space-time interest points detector to extract features from the data. Overall, the above results proved the effectiveness of our proposed LSTM model when combining the evaluation on the adjacent skeleton chain and the MoveNet Skeleton extractor, which is considered an extractor featured. Furthermore, this proposal has achieved a particular effect when giving relatively good results, with Accuracy falling at 85.4% and AUC contributing at 92.9%.

### 2.2. Skeleton-based action recognition

There have been many studies on skeleton-based action recognition with different approaches, such as [28–39], etc. For example, in the study [29], authors proposed an end-to-end hierarchical recurrent neural network (RNN) for skeleton-based action recognition. The skeleton is divided into five parts according to the human physical structure, and they are fed five subnets separately. With the increment of layers, the representations extracted by the subnets have been hierarchically merged to become the input of higher layers.

The authors compared five other deep RNN architectures derived from their model and several other methods on three publicly available datasets: MSR Action3D Dataset, Berkeley MHAD, and Motion Capture Dataset HDM05. The best experimental results with the HBRNN-L network achieved 100% accuracy on the Berkeley MHAD dataset. It also proved that the proposed model achieves state-of-the-art performance. Recent methods based on 3D skeleton data have achieved outstanding performance due to their conciseness, robustness, and view-independent representation. For example, in [31], authors proposed LSTM and CNN, and experimental results showed that the score fusion between CNN and LSTM performs better than that between LSTM and LSTM for the same feature. The proposed dataset is NTU RGB+D, the largest dataset for HAR, and consists of 56578 action samples of 60 different classes, including front view, two side views, and left and right 45-degree views performed by 40 subjects aged between 10 and 35. The proposed method achieved state-of-the-art results on NTU RGB+D datasets for 3D human action analysis with an accuracy of 87.40. Furthermore, they asserted that LSTM is good at exploiting complete temporal information while CNN is biased toward mining spatial solid information. The work [33] exploited three-dimensional (3D) depth sensors and proposed a new skeleton-based approach using the Minkowski and cosine distances between the 3D joints. Using the Extremely Randomised

Trees algorithm, they trained and validated their approach on the Microsoft MSR 3D Action and MSR Daily Activity 3D datasets. In the context that skeletal representation of the human body extracted in real-time from depth maps brings high efficiency in action recognition, the authors in the study [34] proposed a method for extracting sets of spatial and temporal local features from subgroups of joints, which are aggregated by a robust method based on the VLAD algorithm and a pool of clusters. Several feature vectors are combined by a metric learning method inspired by the LMNN algorithm to improve the classification accuracy using the nonparametric k-NN classifier. Three public datasets were used: the MSR-Action3D, the UTKinect-Action3D, and the Florence 3D Actions dataset. View variations and noisy data challenge HAR based on skeletons, and how to represent spatiotemporal skeleton sequences effectively are problems to solve. The work [35] presented an enhanced skeleton visualization method for view-invariant human action recognition. The method includes three stages: first, developing a sequence-based view-invariant transform to eliminate the effect of view variations on spatiotemporal locations of skeleton joints, second, visualizing transformed skeletons as a series of color images, implicitly encoding the spatiotemporal information of skeleton joints, third, applying a convolutional neural networks-based model to extract robust and discriminative features from color images. The method experimented on four datasets, including the largest and most challenging NTU RGB+D dataset for skeleton-based action recognition.

The authors in [38] proposed a method to extract joints and skeleton information using a generative adversarial network (GAN). First, an original 1-channel thermal image was converted into a 3-channel thermal image, and then the images were combined to improve the extraction performance. The proposed human action recognition is performed by combining CNN and LSTM. They use self-collected and open datasets: DTh-DB, DI&V-DB, and CASIA C open database. They confirmed that the proposed method could run fast enough to perform both skeleton generation and action recognition, with the total frame rate of the proposed method being 9.38 frames per second (1000/106.64).

Jun Liu et al. [28] proposed SpatioTemporal LSTM as a base model from which they proposed Global Context-Aware Attention LSTM. This model can selectively focus on information matches in each frame of the frame sequence using global context information by combining the skeleton, the GCA-LSTM architecture, and the training model (1:Directly Train of the Whole Network and 2: Stepwise Training) for the network to get more efficient and optimal results. The results obtained on this model and the UT-Kinect dataset are awe-inspiring, with the near-maximum accuracy falling at 99%. Besides, a new method to extract joints and skeletons was proposed by Ganbayar Batchuluun et al. [38]. One-channel thermal images were initially converted to 3-channel thermal images, which were combined to improve extraction efficiency. A network (GAN) was used in the proposed method to extract information about joints and skeletons. In addition, research to recognize different human actions has been conducted using joint and bone information extracted by this method. The proposed human action recognition combines a convolutional neural network (CNN) and long-term, short-term memory (LSTM). With this method, it has proven its effectiveness and power when it is effective when achieving results with an accuracy of more than 99%.

## 3. METHODS

This study uses two approaches for action identification in videos. The first includes extracting and classifying skeletons with deep learning architectures on 1D data. The second approach receives a sequence of images and classifies them with deep learning architecture on 2D data. For example, Figure 1 exhibits a movement recognition system for Taekwondo 1 (TAEGEUK IN JANG) that is based on extracting the skeleton with the following steps:

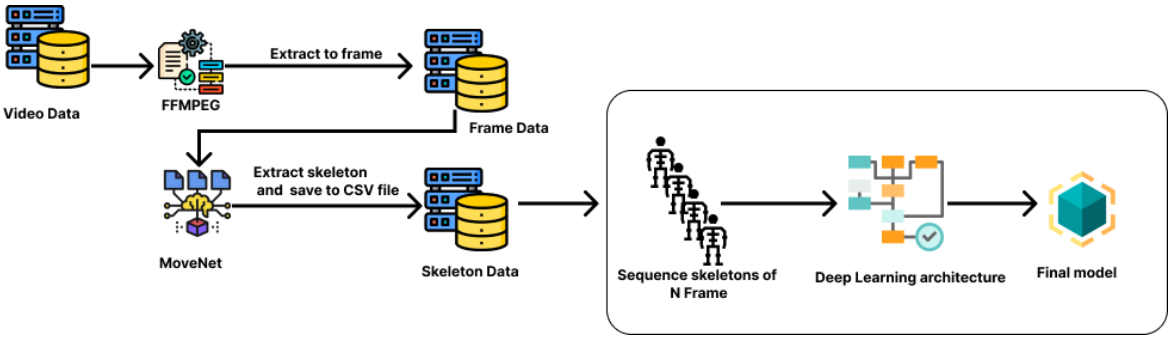



Figure 1: A workflow for Taekwondo action recognition using extracted skeleton

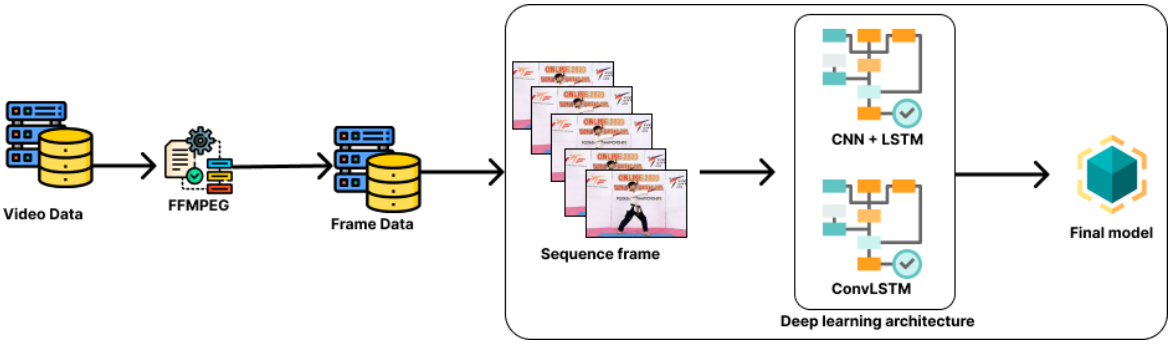



Figure 2: A workflow for Taekwondo action recognition using image frames extracted from video.

- Step 1: From the original video dataset, convert to an image with N frames per second with FFMPEG.
- Step 2: After obtaining a sequence of images, extract the feature skeleton with Movenet. Here, the feature is extracted from a successive frame sequence, so the extracted skeleton is also a successive skeleton sequence, and we save them to a CSV file.
- Step 3: Include the extracted skeleton dataset into model training.

Figure 2 illustrates the approach based on considering a sequence of images (consensus frames in the video) using deep learning architecture:

- Step 1: From the original video dataset, convert them to N frames (images) per second with FFMPEG.

- Step 2: After obtaining a series of images in Step 1, we save them to a CSV file.
- Step 3: Insert the extracted frame from the CSV file into the training model.

### 3.1. Data description

The dataset was developed specifically with data on movements performed by Taekwondo athletes. To achieve this task, we collect video Taekwondo 1 data of students at Can Tho University and other resources such as the Internet, YouTube, etc. From the Taekwondo datasets we collected, we will extract the skeleton and save the result to a CSV file, and then we will use that to train the model. The obtained dataset consists of sixteen classes, each representing the athletes' different technique movements. The primary purpose of this dataset is to collect information on the movements of Taekwondo athletes for use in training deep learning classification methods and to support university Taekwondo teaching. The total number of videos we collect is 35 videos with different resolutions and lengths. They were mostly recorded with the phone's camera. Figure 3 shows the length distribution of the videos. This chart shows that videos are mainly 30 to 60 seconds(27 videos). Videos over 60 seconds only have 1 video. Table 1 summarizes the resolutions of the collected videos. The videos are mainly 1280×720 pixels (with 15 videos) and 1920×1080 pixels (with 12 videos).

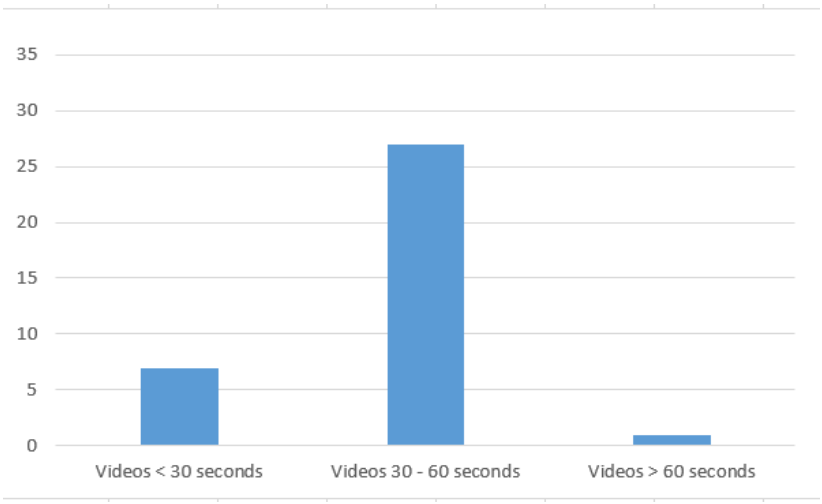



Figure 3: Duration distribution of videos

### 3.2. Data pre-processing

Fast Forward Moving Picture Experts Group (FFmpeg) is an open-source framework that handles powerful multimedia. It allows users to handle multimedia tasks such as decoding, encoding, transcoding, filtering, information extraction, and split video. MoveNet is a high-speed and accurate model to detect 17 points of the body according to the COCO standard. The model is offered on TF.Hub with two variants, Lightning and Thunder. Lightning is intended for latency-critical applications, and Thunder is for applications that require high accuracy. Both models run faster than real-time (more than 30 frames per second) on most

Table 1: Summary table of the resolutions of the collected videos

| No. | Resolution | Number of video |
|---|---|---|
| 1 | 1024 × 576 | 1 |
| 2 | 1272 × 720 | 1 |
| 3 | 1280 × 720 | 15 |
| 4 | 1920 × 1080 | 12 |
| 5 | 256 × 192 | 1 |
| 6 | 320 × 240 | 1 |
| 7 | 608 × 1080 | 1 |
| 8 | 640 × 288 | 1 |
| 9 | 640 × 360 | 3 |
| 10 | 854 × 474 | 1 |
| | Total | **37** |

desktops and laptops, which proves critical for fitness and wellness applications. MoveNet uses heat maps to circumscribe key human points precisely. It is a bottom-up estimation model, which first detects each person's human joints and then assembles them into each person's pose. The feature extractor in MoveNet is MobileNetV2 with an additional Feature Pyramid Network (FPN), capable of outputting semantically rich feature maps at high resolution (output stride 4). Four predictor heads are connected to the feature extractor responsible for prediction. Firstly, the Person center heatmap predicts the geometric center of person instances. Next, the key point regression field aims to predict a complete set of key points for a person, used for grouping key points into instances. In contrast, the Person keypoint heatmap analyzes the location of all key points independent of person instances. Finally, the 2D per-keypoint offset field explored local offsets from each output feature map pixel to the precise sub-pixel location of each keypoint.

### 3.3. Proposed learning architectures

Long Short-Term Memory Networks (LSTM) are an improved form of RNN. It works well with many problems, so it has gradually become popular. However, today, LSTM is designed to avoid the problem of long-term dependence. Keeping this information in mind for a long time is the default asset that LSTM can have. In this study, we build an LSTM network architecture with four layers (Figure 4). The first two layers are the overlapping LSTM layers, followed by 2 Dense layers. For each layer, we intermingle with a Dropout of 0.2 to avoid overfitting when training the model.

Convolutional Long Short-Term Memory (ConvLSTM) is a variant of the LSTM network that contains the convolution operations embedded in the architecture, making it possible to determine the spatial features of the data while still considering the temporal relationship. For video classification, this approach effectively captures the spatial relationship in individual frames and the temporal relationship on different frames, from which ConvLSTM can receive 3-dimensional input (width, height, channel number). In contrast, an LSTM receives 1-dimensional input, so the LSTM is incompatible with spatial-temporal data modeling. The ConvLSTM2D layer takes in the number of filters and kernel size required for applying the

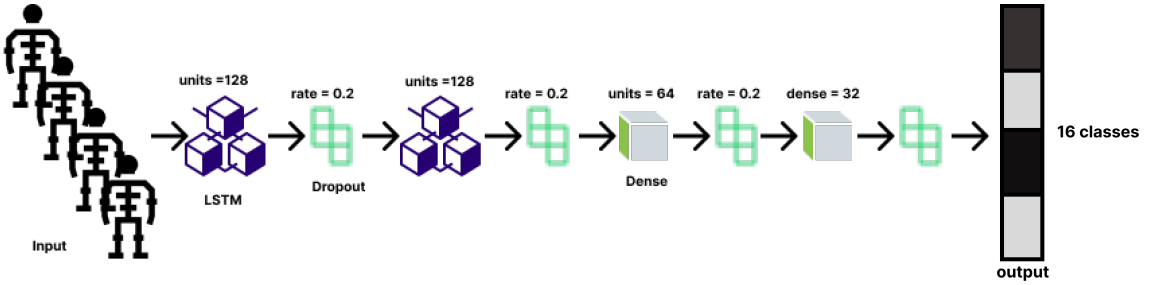



Figure 4: The proposed LSTM architecture for action recognition

convolutional operations. Finally, the output of the layers is flattened in the end and is fed to the Dense layer with softmax activation, which outputs the probability of each action category. We also use MaxPooling3D layers to reduce the dimensions of the frames and avoid unnecessary computations and Dropout layers to prevent overfitting the model on the data. The architecture is simple and has a small number of trainable parameters. This is because we are only dealing with a small subset of the dataset, which does not require a large-scale model (the architecture is described in Figure 5).

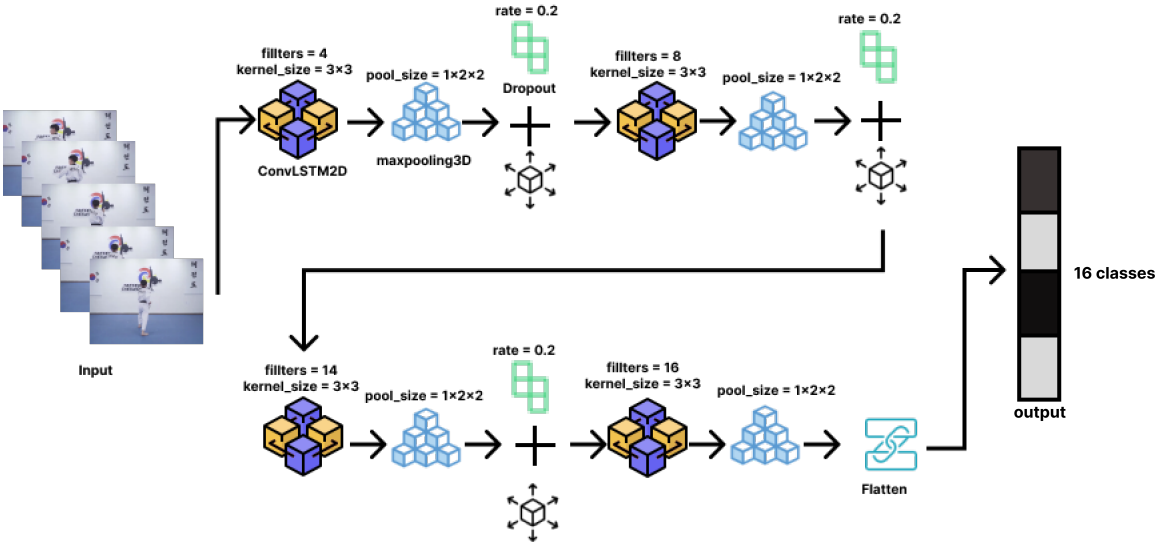



Figure 5: The proposed ConvLSTM architecture for action recognition

Long-Term Recurrent Convolutional Network (LRCN) is a way to incorporate CNN and LSTM layers into a model. Using the CNN model with its own trained LSTM model is also viable. It is possible to take advantage of a pre-trained model to extract spatial data from video frames using CNN, and it is possible to transform this model for the application. Therefore, the LSTM prototype can use the information gathered from CNN's video to predict the activity being performed in the video. On the other hand, at each time step, the collected spatial features will be provided to an LSTM layer to create a temporal sequence model, which starts with the Convolutions layer. A robust model is produced due to the network learning spatiotemporal properties in this way during an end-to-end training session.

Another similar approach can be used to build a CNN and separately trained LSTM models. CNN models can be used to extract spatial features from frames in the video, and for this purpose, a pre-trained model can be used, which can be appropriately refined for each problem. Moreover, the LSTM model can then use features extracted by CNN to predict what action is being taken in the video. This study will implement LRCN modeling by combining CNN and LSTM layers in a single model (Figure 6). The Convolutions classes are used to extract spatial features from the frames, and the extracted spatial features are fed into the LSTM layer(s) at each time step, thereby modeling the time sequence. This way, the network learns spatial features directly during end-to-end training, resulting in a robust model. To implement our LRCN architecture, we will use time-distributed Conv2D layers, followed by MaxPooling2D and Dropout layers. The feature extracted from the Conv2D layers will then be flattened using the Flatten layer and fed to an LSTM layer. Finally, the Dense layer with softmax activation will then use the output from the LSTM layer to predict the action being performed.

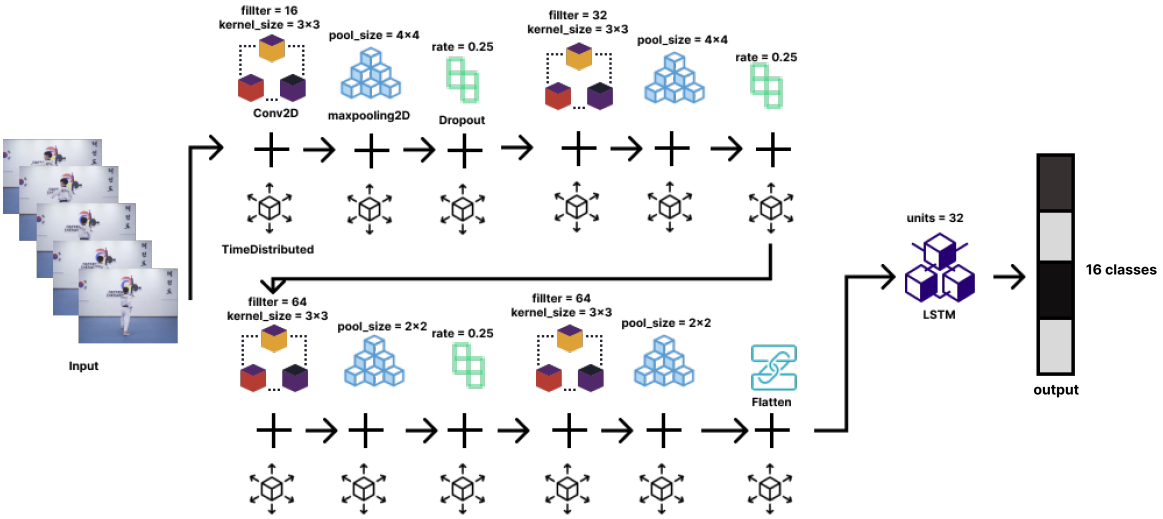



Figure 6: The proposed LRCN architecture for action recognition

## 4. EXPERIMENTAL RESULTS

From methods described in previous sections, we experiment with architectures with parameters including 1000 epochs, learning rate $10^{-4}$, etc., as detailed in Section 4.1. Then, our result was presented and depicted in detail in Sections 4.2 and 4.3. In addition, we present comparisons with other approaches in Section 4.4.

### 4.1. Environment settings

The entire runtime runs on Colab equipped with an NVIDIA Geforce MX150 2GB connected to a personal Core-i5 with 8GB RAM. The data include 35 videos covering 20 actions in a Taekwondo lesson. 80% of the video sequences are used for training and validation purposes, where we use 68% of the total data and 12% for validation, while the remaining

includes 20% for testing. All considered models were configured with a batch size of 64, the Adam optimizer with a default learning rate of 0.001, and a maximum run of 1000 epochs detailed in Table 2.

Table 2: Hyperparameters for our proposed architecture

| | batch size | epochs | learning rate | optimizer | early stopping |
|---|---|---|---|---|---|
| Value | 64 | 1000 | 0.001 | adam | epoch patience: 20 |

### 4.2. Experimental results with various architectures

We evaluate the efficiency by including various values of consecutive skeleton sequences and image sequences. We take the number of frames and consecutive frames summarized from 2, 3, 4, and 5 to evaluate the influence of the hyper-parameters on the performance. The results of training the model from the dataset of various values of consecutive frames obtained are clearly shown in Table 3 with LSTM, ConvLSTM, and LRCN as described in Figures 4, 5, and 6, respectively. We see that the accuracy of the overall training set is 99%, but the accuracy on the testing set is LSTM 76%, ConvLSTM 9%, and LRCN 9%. The loss on the testing set in LSTM architecture reached the lowest value at 58%, then ConvLSTM, and finally LRCN. Furthermore, the training model results from the dataset of 3 consecutive frames obtained are clear. We see that the accuracy on the overall training set is higher than 87%. However, accuracies on the testing set achieved are 0.76, 0.10, and 0.15 on LSTM, ConvLSTM, and LRCN, respectively, and training results show that in architecture ConvLSTM and LRCN, the accuracy is improved compared to training results on data 2 consecutive frames. The loss on the testing set in LSTM architecture reached the lowest value at 57%, followed by ConvLSTM and LRCN. Besides, we see that the accuracy in all three architectures is improved when we increase the number of consecutive sequences, with LSTM achieving 80% ConvLSTM achieving 0.21, and LRCN achieving 0.23.

In Figure 7, we compare the considered method in Taekwondo pose classification tasks in training and test sets. We can see that on the training set, the performance of all three methods is very high and approximately the same (more than 96%). However, a significant difference can be seen in the test set when the ConvLSTM and LRCN models have deficient performance, while the LSTM has an accuracy of up to 83% on the test set. In terms of training and test sets, the LRCN model and the ConvLSTM model have a huge difference in the accuracy between the two sets, while the LSTM shows an advantage when the difference is relatively small. Looking at the overview through the table and the figure, we see that LSTM is superior to ConvLSTM and LRCN architecture. Unfortunately, the image is not suitable due to the background noise on the image. Nevertheless, the overall accuracy of the LSTM is 83%, and the AUC measurement achieved by the LSTM is 99%. From the obtained results, LTSM can be selected for the next experiments.

Figure 8 illustrates training accuracy per epoch in 3 different learning rates of LSTM architecture. It can be seen that most line graphs have a similar form; the accuracy significantly improves in the first period, and it becomes stable at the end of the graph. The line graph in (c) has the training and validation accuracy superior to the rest of the learning rates, with training accuracy reaching more than 85% and validation accuracy at around 83%.

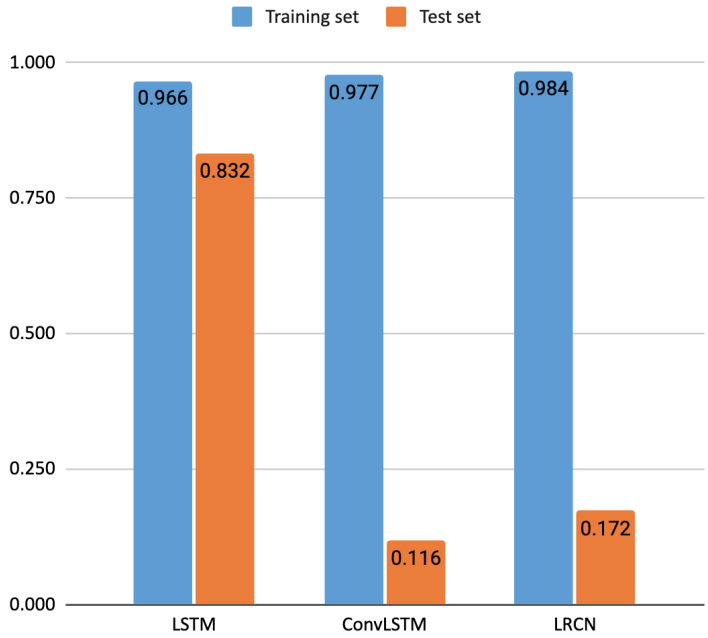
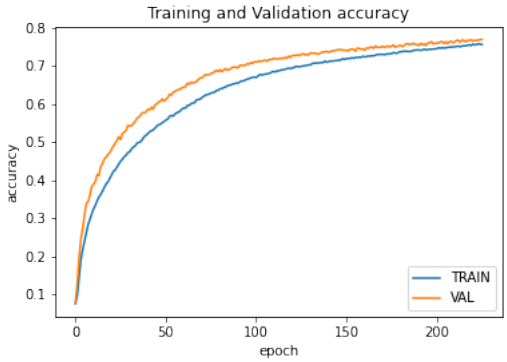



Figure 7: The performance comparison in accuracy for action recognition with various algorithms on the self-collected dataset.

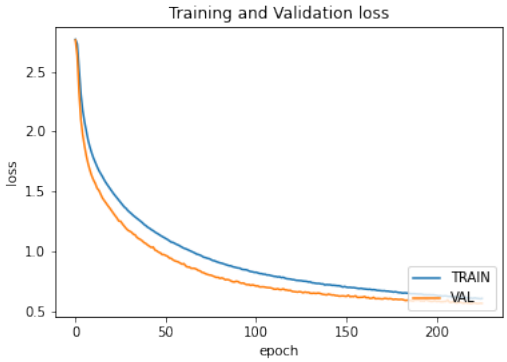
Table 3: Training result of 2, 3, 4, 5 sequence frames on five frames per second dataset

| Architecture | Sequence frames | Training set | | Test set | |
|---|---|---|---|---|---|
| | | Accuracy | Loss | Accuracy | Loss |
| LSTM | 2-frame | 0.8707 | 0.3003 | 0.7616 | 0.5815 |
| LSTM | 3-frame | 0.8793 | 0.2828 | 0.7629 | 0.5695 |
| LSTM | 4-frame | 0.9395 | 0.1524 | 0.8035 | 0.5160 |
| LSTM | 5-frame | 0.9664 | 0.0850 | 0.8323 | 0.4591 |
| ConvLSTM | 2-frame | 0.8177 | 0.5516 | 0.0930 | 7.3081 |
| ConvLSTM | 3-frame | 0.9284 | 0.2500 | 0.1029 | 5.5018 |
| ConvLSTM | 4-frame | 0.9335 | 0.2275 | 0.2154 | 8.8848 |
| ConvLSTM | 5-frame | 0.9769 | 0.0850 | 0.1163 | 6.3838 |
| LRCN | 2-frame | 0.9099 | 0.2771 | 0.0945 | 9.4033 |
| LRCN | 3-frame | 0.9745 | 0.0994 | 0.1493 | 6.3580 |
| LRCN | 4-frame | 0.9684 | 0.1180 | 0.2290 | 6.8483 |
| LRCN | 5-frame | 0.9971 | 0.0236 | 0.3978 | 2.7707 |

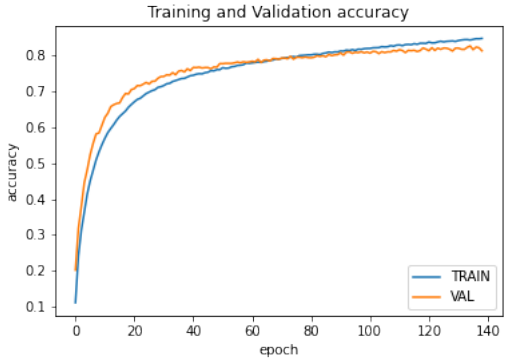
As investigated in the above Fig, the training result declines, followed by the learning rate. In particular, line graph c gives the training result better than the lower learning rate line graph. The chart results show that the training and validation accuracy is relatively low, only about 80%. Figures 8b, 8d, and 8f revealed the training loss per epoch in three different learning rates of LSTM architecture. It can be similar to the accuracy line graphs in Figure 8. Most training loss results drop sharply in the first 50 epochs and stabilize afterward. The



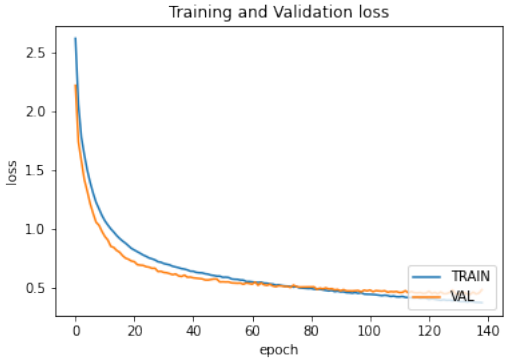

(a) learning rate=0.0001 (in Accuracy)



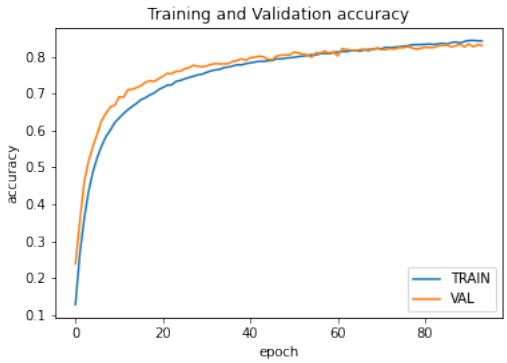

(b) learning rate=0.0001 (in Loss)



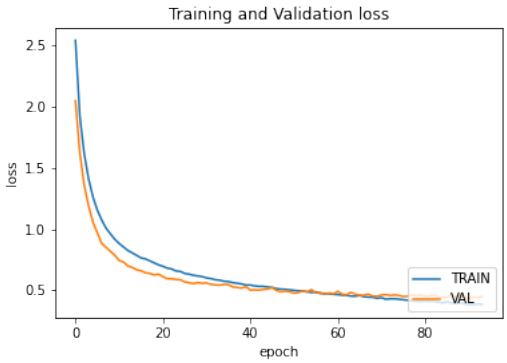

(c) learning rate=0.0005 (in Accuracy)




(d) learning rate=0.0005 (in Loss)




(e) learning rate=0.0010 (in Accuracy)




(f) learning rate=0.0010 (in Loss)

Figure 8: Some performance illustrations of LSTM architecture during epochs with various learning rates.

results of the models are very good and give results 30% lower on training and 50% lower on validation. As investigated in the above figure, the Loss value on the small learning rate architecture is higher than the significant learning rate. This demonstrates that the skeleton dataset is unsuitable for our proposal architecture's small learning rates.

### 4.3. Pose classification performance with various consecutive skeleton sequences

From the results of videos with a speed of five frames per second, we see that the architecture for action-recognized video classification is unsuitable for the dataset, so we continue to increase the number of frames per second in the LSTM architecture. Symbols 2f, 3f, 4f, and 5f are the number of consecutive skeletons to include in training the model; for example, 5f, i.e., five consecutive skeletons, are included in the training. With two consecutive skeletons, we see that in the data sets of 10, 20, and 30 frames per second, the results are nearly as accurate as the results, and the testing set results at 20 frames per second are low. The highest was with 77.5% accuracy, and the highest was 30 frames per second with 79.7%. The AUC on the testing set result, 30 frames per second, also achieved the highest result with 98%. Regarding loss, all three datasets give good results. Overall, 30 frames per second is the best result for two consecutive frames.

The data is three frames in a row. We see that the accuracy on three datasets shows that the accuracy result of 30 frames per second has the highest accuracy with 80% accuracy, and the lowest is the result of the set. Twenty frames per second data reach 78% in accuracy on the testing set. AUC measurement over ten frames per second reached the highest with 98%. Overall, the 30 frames per second dataset is superior to the other two datasets because it excels in parameters such as Accuracy on the training set, Accuracy on the testing set, Loss on the training set, and AUC on the training set. For four consecutive skeletons, the highest accuracy is ten frames per second, with 82% on the test set. The AUC measurement also achieved the highest result, with 98%. The loss on the testing set results on the 30 frames per second data set was best, but vice versa in the training set. Accuracy on the training set also reached the highest value for the 30 frames per second data set. For the remaining five frames in a row, all datasets generally achieve 79% or more, but ten frames per second surpass all with 84% on the training set. The AUC measurement on the ten frames per second data set also reached the highest value. In general, through all data sets, we see the results of taking five consecutive skeletons to train the model to get the best results.

Columes 2f, 3f, 4f, and 5f in Table 4 and Table 5 denote the number of frames included in the training model. For example, 2f, the number of consecutive frames to be input, will be 2f, 3f; the number of consecutive frames to be input will be 3f; and 4f, 5f will do the same. On a data set of 10 frames per second, the accuracy reached the highest value of 98.8% in Pose 10 with four consecutive frames, while Pose 2 reached the lowest value of 53.3% in Pose 2 with three consecutive frames. As for the f1_score measure, the highest result is 97% in Pose 12 with five consecutive frames, and the lowest result is 54.6% in Pose 9 with two consecutive frames.

Transitioning to a dataset with 20 frames per second, the F1 score measure peaked at 94.6% for Pose 12 with four consecutive frames, and Pose 9 exhibited its lowest F1 score of 0.536. Regarding the accuracy measure, the zenith of 0.975 emerged for Pose 5 with five consecutive frames, while the nadir rested at 0.511 for Pose 14 with two consecutive frames. Progressing to the dataset encompassing 30 frames per second, the pinnacle F1 score of 93.4% was ascribed to Pose 12 with two consecutive frames, whereas the suboptimal F1 score of 55% characterized Pose 11 with three consecutive frames. Regarding accuracy, the acme of 98.2% was realized for Pose 10 with four consecutive frames, and the floor value of 51% manifested for Pose 2 with three consecutive frames.

Table 4: The performance in F1 score with 2, 3, 4, and 5 consecutive frames sequence for the input of the architectures in videos with the rate of 10 and 20 frames per second.

| | | 10 frames per second | | | | 20 frames per second | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 2f | 3f | 4f | 5f | 2f | 3f | 4f | 5f |
| **Pose 1** | F1 score | 0.880 | 0.884 | 0.896 | 0.916 | 0.890 | 0.878 | 0.890 | 0.882 |
| | Accuracy | 0.859 | 0.855 | 0.852 | 0.894 | 0.872 | 0.852 | 0.866 | 0.847 |
| **Pose 2** | F1 score | 0.884 | 0.900 | 0.948 | 0.950 | 0.892 | 0.854 | 0.862 | 0.874 |
| | Accuracy | 0.586 | 0.533 | 0.666 | 0.649 | 0.549 | 0.505 | 0.488 | 0.567 |
| **Pose 3** | F1 score | 0.786 | 0.820 | 0.798 | 0.832 | 0.762 | 0.796 | 0.798 | 0.782 |
| | Accuracy | 0.754 | 0.797 | 0.782 | 0.857 | 0.772 | 0.832 | 0.857 | 0.892 |
| **Pose 4** | F1 score | 0.652 | 0.642 | 0.666 | 0.702 | 0.586 | 0.666 | 0.668 | 0.590 |
| | Accuracy | 0.626 | 0.821 | 0.726 | 0.754 | 0.638 | 0.558 | 0.633 | 0.569 |
| **Pose 5** | F1 score | 0.780 | 0.688 | 0.820 | 0.850 | 0.744 | 0.782 | 0.726 | 0.714 |
| | Accuracy | 0.950 | 0.960 | 0.978 | 0.979 | 0.935 | 0.897 | 0.964 | 0.975 |
| **Pose 6** | F1 score | 0.676 | 0.626 | 0.728 | 0.732 | 0.604 | 0.670 | 0.656 | 0.696 |
| | Accuracy | 0.792 | 0.814 | 0.793 | 0.785 | 0.862 | 0.863 | 0.808 | 0.876 |
| **Pose 7** | F1 score | 0.786 | 0.798 | 0.796 | 0.774 | 0.786 | 0.760 | 0.756 | 0.774 |
| | Accuracy | 0.857 | 0.894 | 0.894 | 0.899 | 0.839 | 0.830 | 0.859 | 0.863 |
| **Pose 8** | F1 score | 0.878 | 0.850 | 0.852 | 0.852 | 0.862 | 0.874 | 0.868 | 0.872 |
| | Accuracy | 0.840 | 0.892 | 0.907 | 0.938 | 0.857 | 0.865 | 0.890 | 0.896 |
| **Pose 9** | F1 score | 0.546 | 0.566 | 0.586 | 0.630 | 0.538 | 0.536 | 0.554 | 0.590 |
| | Accuracy | 0.860 | 0.788 | 0.909 | 0.945 | 0.838 | 0.612 | 0.681 | 0.694 |
| **Pose 10** | F1 score | 0.764 | 0.738 | 0.806 | 0.812 | 0.756 | 0.778 | 0.816 | 0.792 |
| | Accuracy | 0.894 | 0.948 | 0.988 | 0.978 | 0.891 | 0.930 | 0.902 | 0.928 |
| **Pose 11** | F1 score | 0.588 | 0.666 | 0.660 | 0.686 | 0.562 | 0.600 | 0.624 | 0.628 |
| | Accuracy | 0.824 | 0.847 | 0.748 | 0.813 | 0.769 | 0.796 | 0.812 | 0.786 |
| **Pose 12** | F1 score | 0.906 | 0.948 | 0.940 | 0.970 | 0.926 | 0.864 | 0.946 | 0.932 |
| | Accuracy | 0.609 | 0.563 | 0.642 | 0.668 | 0.528 | 0.646 | 0.651 | 0.490 |
| **Pose 13** | F1 score | 0.856 | 0.878 | 0.860 | 0.850 | 0.894 | 0.856 | 0.866 | 0.912 |
| | Accuracy | 0.818 | 0.673 | 0.770 | 0.786 | 0.745 | 0.813 | 0.767 | 0.835 |
| **Pose 14** | F1 score | 0.766 | 0.800 | 0.798 | 0.808 | 0.758 | 0.770 | 0.778 | 0.802 |
| | Accuracy | 0.587 | 0.564 | 0.658 | 0.688 | 0.511 | 0.584 | 0.591 | 0.642 |
| **Pose 15** | F1 score | 0.848 | 0.880 | 0.904 | 0.932 | 0.830 | 0.848 | 0.870 | 0.886 |
| | Accuracy | 0.784 | 0.854 | 0.895 | 0.812 | 0.869 | 0.842 | 0.901 | 0.798 |
| **Pose 16** | F1 score | 0.882 | 0.814 | 0.914 | 0.916 | 0.806 | 0.706 | 0.738 | 0.738 |
| | Accuracy | 0.893 | 0.812 | 0.811 | 0.863 | 0.825 | 0.866 | 0.821 | 0.910 |

### 4.4. Comparative analysis

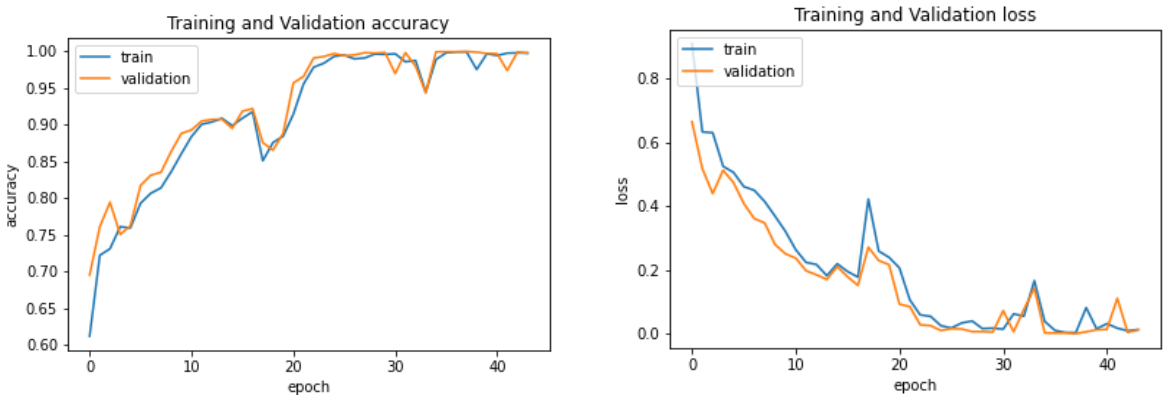
Evaluation of large data sets from previous studies is essential. Datasets, such as UCF101, Hollywood dataset, Olympic dataset, etc., are widely used and popular in behavior recognition or motion recognition tasks. We propose to test on the KTH (Human Action Dataset) set, which is collected consisting of 600 different videos of all videos stored using AVI file format, video files for each combination of 25 objects, six actions, and four scenarios. The

Table 5: F1 score experiment on 2, 3, 4, and 5 frame sequence of 30 frames per second

| | | 30 frames per second | | | |
|---|---|---|---|---|---|
| | | 2f | 3f | 4f | 5f |
| **Pose 1** | F1 score | 0.912 | 0.898 | 0.904 | 0.880 |
| | Accuracy | 0.931 | 0.866 | 0.891 | 0.834 |
| **Pose 2** | F1 score | 0.904 | 0.894 | 0.908 | 0.862 |
| | Accuracy | 0.582 | 0.510 | 0.654 | 0.520 |
| **Pose 3** | F1 score | 0.828 | 0.816 | 0.838 | 0.836 |
| | Accuracy | 0.775 | 0.798 | 0.865 | 0.880 |
| **Pose 4** | F1 score | 0.612 | 0.668 | 0.656 | 0.684 |
| | Accuracy | 0.573 | 0.514 | 0.653 | 0.656 |
| **Pose 5** | F1 score | 0.832 | 0.818 | 0.770 | 0.780 |
| | Accuracy | 0.941 | 0.939 | 0.943 | 0.964 |
| **Pose 6** | F1 score | 0.648 | 0.646 | 0.664 | 0.666 |
| | Accuracy | 0.886 | 0.899 | 0.935 | 0.918 |
| **Pose 7** | F1 score | 0.744 | 0.808 | 0.818 | 0.848 |
| | Accuracy | 0.804 | 0.809 | 0.823 | 0.802 |
| **Pose 8** | F1 score | 0.868 | 0.882 | 0.886 | 0.896 |
| | Accuracy | 0.878 | 0.896 | 0.887 | 0.918 |
| **Pose 9** | F1 score | 0.564 | 0.560 | 0.560 | 0.580 |
| | Accuracy | 0.792 | 0.747 | 0.867 | 0.580 |
| **Pose 10** | F1 score | 0.778 | 0.790 | 0.804 | 0.836 |
| | Accuracy | 0.940 | 0.950 | 0.982 | 0.956 |
| **Pose 11** | F1 score | 0.568 | 0.550 | 0.572 | 0.606 |
| | Accuracy | 0.838 | 0.809 | 0.831 | 0.826 |
| **Pose 12** | F1 score | 0.934 | 0.914 | 0.892 | 0.914 |
| | Accuracy | 0.541 | 0.678 | 0.520 | 0.658 |
| **Pose 13** | F1 score | 0.908 | 0.910 | 0.916 | 0.916 |
| | Accuracy | 0.841 | 0.851 | 0.862 | 0.846 |
| **Pose 14** | F1 score | 0.778 | 0.750 | 0.740 | 0.814 |
| | Accuracy | 0.558 | 0.563 | 0.554 | 0.578 |
| **Pose 15** | F1 score | 0.856 | 0.860 | 0.856 | 0.906 |
| | Accuracy | 0.902 | 0.898 | 0.865 | 0.898 |
| **Pose 16** | F1 score | 0.778 | 0.774 | 0.792 | 0.658 |
| | Accuracy | 0.818 | 0.829 | 0.856 | 0.862 |

dataset is divided into three parts to support model evaluation more intuitively. The training set is divided into 408 videos used to train the model, 72 videos for the validation Set, and 120 videos for the Testing Set. The proposed architecture is used as shown in Figure 4, but we increase the number of frames to 70 per Sample to better fit the data due to the hitting. In addition, the KTH dataset consists of 6 action classes, so we adjust the number of outputs of the proposed LTSM to 6. Moreover, because the Taekwondo data we collected and divided into 16 videos corresponding to 16 poses, each pose has a thickness of about 2 seconds. In addition, each video in the KTH dataset is, on average, 30 seconds for 1 video.

After getting the video, use the FFMPEG tool to extract frames in a data set of 10 frames per second. This cropping is done by dividing per second on video into ten images. This video has high-speed motion, so frames close to each other will have similar features, so using those frames is unnecessary. We obtain the following result by training the above model on the KTH dataset in 1000 epochs and incorporating an early stopping technique with epoch patience of 20. The model can be used early if the accuracy no longer increases.




(a) Accuracy (b) Loss

Figure 9: The performance of training on the KTH dataset using sequentially skeleton was extracted by Movenet to training model.

Figure 9a depicts the resulting Loss during the 40 epochs of the obtained model. Loss results in a sharp decrease in the first 25 epochs and slight fluctuations in the 25th to last epochs. The results show that the difference between Training and Validation is not too much, in the range of only 2-3%. This proves that the model is relatively suitable for this data and the task of classifying human behavioral actions. In addition, Figure 9b depicts the accuracy results of the evaluation model on the KTH dataset during 40 epochs. Accuracy training and Validation results increased sharply in the first 25 epochs and began to peak from epochs 25. Although there were strong fluctuations in the model between epochs 15 to epochs 20 and the beginning of epoch 30, the model's accuracy is still growing.

Table 6: The hyperparameters of architectures were used in our method, Ref. [40] and Ref. [24]. "NA" denotes "Not Available".

| | Ref. [40] | Ref. [24] | Our LSTM |
|---|---|---|---|
| Input size | 224×224×3 | 9×80×60 | 70×34 |
| Components in architecture | Gated Recurrent Unit | 3D convolution | 1D convolution |
| Learning rate | 0.1 | NA | 0.001 |
| Batch size | 128 | NA | 64 |
| Maximum epochs | 30 | NA | 1000 |

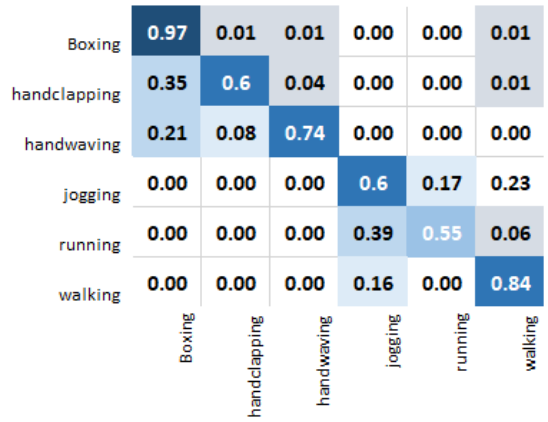
Table 6 compares several hyperparameters of architectures used in the experiments between our method with the methods of [24,40] on the KTH dataset. Each input image of [40] has a dimension (input_shape) of 224 × 224 × 3, and the work in [24] used 9×80×60, while

the dimension of input in the proposed LSTM is 34×70 where 34 denotes the skeleton coordinate points (17 pairs of the coordinate (x,y)) and 70 is the number of consecutive skeletons. To limit the loss of model accuracy, we use a meager learning rate, 0.001, compared to [40], having a learning rate of 0.1. The number of epochs that we use is 1000 in order to achieve optimal training efficiency. This number of epochs is many times higher than [40] and [24]. The method in [40] passed a certain number of frames through the network model to learn contiguous features. In this way, N. Jaouedi et al. [40] implemented Motion Tracking to capture the action of objects in images/videos. They proposed a network model called the GRNN model and trained on the above model to obtain considerable accuracy. Similarly, S. Ji et [24] also performed their model evaluation on consecutive frame sequences and used Motion Cuboid to extract the subject's movements, and then they evaluated their method through a 3DCNN network model they proposed with 3D convolutions. The complexity of their model on The TRECVID 2008 dataset was 289,536 trainable parameters [41]. However, the number of parameters of the model trained on the KTH dataset was not mentioned, while our model has 225,574 parameters. In particular, the method we proposed on the feature extractor called MoveNet to extract the object's posture. After obtaining the set of coordinate points, we collected them into a series of consecutive coordinate points and put them into the proposed LSTM model with 225,574 parameters for evaluation. The results have shown the efficiency of our proposed method with other modern methods, and our model has relatively good results. Besides, other than the study [42], they revealed their Input size as 160x120x100; other studies did not mention the experimental settings in their published study.

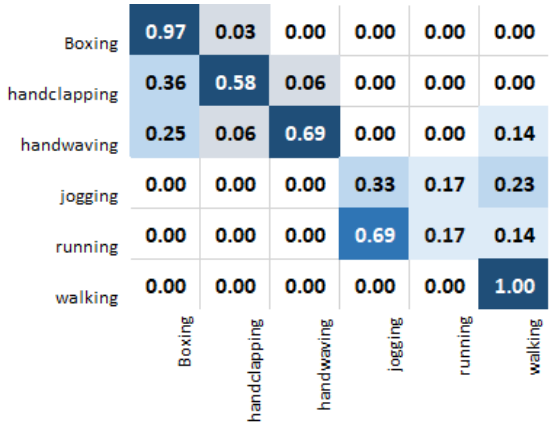
Table 7: Accuracy comparison with some previous approaches

| | Techniques/Models | Dataset | Accuracy |
|---|---|---|---|
| Our method | MoveNet + LSTM | KTH | 0.854 |
| Ref. [26] | Sparse Spatio-temporal feature | KTH | 0.812 |
| Ref. [40] | GMM + KF | KTH | 0.711 |
| Ref. [40] | GRNN | KTH | 0.860 |
| Ref. [24] | 3DCNN | KTH | 0.902 |
| Ref. [25] | ISA + Dense sampling | KTH | 0.914 |
| Ref. [27] | pLSA | KTH | 0.815 |
| Ref. [42] | SVMs | KTH | 0.717 |

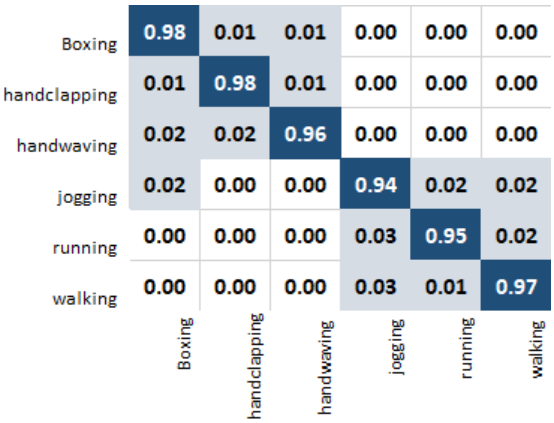
Table 7 exhibits the performance comparison between our method and some previous studies. In the study proposed by Neziha Jaouedi et al. [40], their approach is based on video content analysis and extraction features. First, motion features are represented by tracking human movement using Gaussian mixture model (GMM) and Kalman filter (KF) methods. Then, other features are based on all visual characteristics of each frame on the video sequence using a Recurrent Neural Network model with Gated Recurrence Units. The main advantage of this novel approach is to analyze and extract all the features in each moment and frame by frame of the video. Their results achieved 0.711 with GMM and GMM+KF and 0.86 with the Gated Recurrent Neural Networks (GRNN). The fusion of GMM and Kalman Filter with deep learning models creates a hybrid method capitalizing on classical statistical strengths and modern deep learning techniques. This amalgamation holds promise for



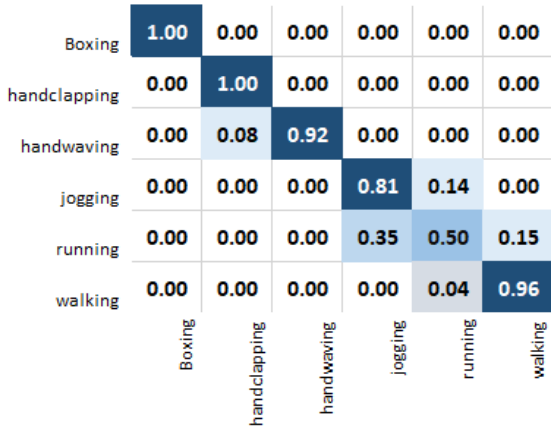

(a) Local Features with SVM with four different scenarios proposed in [42]




(b) Local Features with SVM with scale variations in [42].




(c) GMM + KF + GRNN [40]




(d) Our method

Figure 10: The classification accuracy comparison of six actions on the KTH dataset in confusion matrices between our method and some previous approaches.

improved recognition accuracy. However, incorporating these approaches introduces intricacies, necessitating meticulous parameter tuning and optimization. This complexity might extend training time and computational demands. Concerning our employed methodology, the achieved accuracy stands at 85.4%, a notably favorable outcome. The recorded accuracy of 0.854 signifies the reliability of our approach. With the proposed method of evaluating consecutive skeletal sequences extracted from videos, our model achieved relatively good results, with accuracy achieving approximately 0.86, so it has been proved that the LSTM model does an excellent job of classifying human postures in the KTH dataset.

Figure 10 illustrates the accuracy comparison of six actions in KTH datasets between our method and several methods in [42] and [40] in confusion matrices. The authors in [42] proposed two scenarios for the Local Feature model by combining local features with SVM. They found that Local Features with SVM with four different proposed scenarios bring excellent motion recognition efficiency. It is easy to see that this result is entirely superior to the method proposed by Schuldt et al. in [42]. Looking at each label, we see that

almost all labels give superior and more accurate results. In addition, the results achieved on the Boxing and handicapping labels reached the maximum, and the walking label achieved 96%. In addition, our proposed method gives relatively low results on the running label, so the method also needs to improve on this situation. Almost all methods are identified for confusing results between Running and Jogging labels. Compared with Schuldt's proposed method, our method also solved the confusion between Handwaving, Handclapping, and Boxing labels, which proved the strength of this method on the KTH dataset. Compared to recent work in [40], we achieved better performance in detecting Boxing and hand clapping actions, but the proposed work is still worse for running action.

## 5. CONCLUSION

This study presented an approach using deep learning architectures exploiting 1D data on skeletons extracted from images in videos to perform the pose classification tasks. This transformation from video frames in two-dimensional to one-dimensional data can reduce processing time and computational resources compared to performing video classification tasks. In addition, pose classification tasks use skeleton data to eliminate and skip image noise, such as background behind practitioners, brightness, and unrelated objects appearing in images/videos. The results show that LSTM obtains the best performance while ConvLSTM exhibits the lowest. Further research can work on positioning segmentation and embedding skeletons in videos to support martial arts practitioners visually. The work is expected to enhance the effectiveness of student learning and teachers' assessment in Taekwondo lessons. We will develop the accuracy score by improving LSTM architecture and expanding the number of datasets in the future. In addition, the system supports students' self-learning and assessment in their homes quickly. It also contributes to AI application in education in the fourth industrial revolution.

The datasets used for experiments are publicly available for download from[2],[3].

## ACKNOWLEDGEMENT

This study is partly funded by Can Tho University, Code: TSV2022-33.

## REFERENCES

[1] S. Vishwakarma and A. Agrawal, "A survey on activity recognition and behavior understanding in video surveillance," *The Visual Computer*, vol. 29, no. 10, pp. 983–1009, sep 2012. [Online]. Available: https://doi.org/10.1007/s00371-012-0752-6

[2] M. A. Khan, K. Javed, S. A. Khan, T. Saba, U. Habib, J. A. Khan, and A. A. Abbasi, "Human action recognition using fusion of multiview and deep features: an application to video surveillance," *Multimedia Tools and Applications*, mar 2020. [Online]. Available: https://doi.org/10.1007/s11042-020-08806-9

[3] J. Arunnehru, G. Chamundeeswari, and S. P. Bharathi, "Human action recognition using 3d convolutional neural networks with 3d motion cuboids in surveillance videos,"

[2]https://github.com/thnguyencit/pose-classification
[3]https://www.csc.kth.se/cvap/actions/

*Procedia Computer Science*, vol. 133, pp. 471–477, 2018. [Online]. Available: https://doi.org/10.1016/j.procs.2018.07.059

[4] Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," *Expert Systems with Applications*, vol. 137, pp. 167–190, dec 2019. [Online]. Available: https://doi.org/10.1016/j.eswa.2019.04.057

[5] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022. [Online]. Available: https://doi.org/10.1109/tpami.2022.3183112

[6] H. T. T. Hoang, C. N. Ha, D. T. Nguyen, T. N. Nguyen, T. N. Huynh, T. T. Phan, and H. T. Nguyen, "Poses classification in a taekwondo lesson using skeleton data extracted from videos with shallow and deep learning architectures," in *Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications*. Springer Nature Singapore, 2022, pp. 447–461. [Online]. Available: https://doi.org/10.1007/978-981-19-8069-5_30

[7] F. Cruciani, A. Vafeiadis, C. Nugent, I. Cleland, P. McCullagh, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, and R. Hamzaoui, "Feature learning for human activity recognition using convolutional neural networks," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 1, pp. 18–32, Jan. 2020. [Online]. Available: https://doi.org/10.1007/s42486-020-00026-2

[8] M. H. Javed, Z. Yu, T. Li, T. M. Rajeh, F. Rafique, and S. Waqar, "Hybrid two-stream dynamic CNN for view adaptive human action recognition using ensemble learning," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 4, pp. 1157–1166, Nov. 2021. [Online]. Available: https://doi.org/10.1007/s13042-021-01441-2

[9] S. K. Park, J. H. Chung, T. K. Kang, and M. T. Lim, "Binary dense sift flow based two stream CNN for human action recognition," *Multimedia Tools and Applications*, vol. 80, no. 28-29, pp. 35 697–35 720, Jun. 2021. [Online]. Available: https://doi.org/10.1007/s11042-021-10795-2

[10] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2008. [Online]. Available: https://doi.org/10.1109/cvpr.2008.4587727

[11] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *2011 International Conference on Computer Vision*. IEEE, Nov. 2011. [Online]. Available: https://doi.org/10.1109/iccv.2011.6126543

[12] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. IEEE, 2005. [Online]. Available: https://doi.org/10.1109/iccv.2005.28

[13] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2012. [Online]. Available: https://doi.org/10.1109/cvpr.2012.6247801

[14] Z. Liu, X. Zhang, L. Song, Z. Ding, and H. Duan, "More efficient and effective tricks for deep action recognition," *Cluster Computing*, vol. 22, no. S1, pp. 819–826, Nov. 2017. [Online]. Available: https://doi.org/10.1007/s10586-017-1309-2

[15] R. O. García, E. F. Morales, and L. E. Sucar, "Second-order motion descriptors for efficient action recognition," *Pattern Analysis and Applications*, vol. 24, no. 2, pp. 473–482, Oct. 2020. [Online]. Available: https://doi.org/10.1007/s10044-020-00924-2

[16] S. Alghyaline, J.-W. Hsieh, and C.-H. Chuang, "Video action classification using symmelets and deep learning," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, Oct. 2017. [Online]. Available: https://doi.org/10.1109/smc.2017.8122640

[17] Z. Tu, W. Xie, Q. Qin, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "Multi-stream CNN: Learning representations based on human-related regions for action recognition," *Pattern Recognition*, vol. 79, pp. 32–43, Jul. 2018. [Online]. Available: https://doi.org/10.1016/j.patcog.2018.01.020

[18] A. B. Sargano, X. Wang, P. Angelov, and Z. Habib, "Human action recognition using transfer learning with deep representations," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, May 2017. [Online]. Available: https://doi.org/10.1109/ijcnn.2017.7965890

[19] E. P. Ijjina and K. M. Chalavadi, "Human action recognition in RGB-d videos using motion sequence information and deep learning," *Pattern Recognition*, vol. 72, pp. 504–516, Dec. 2017. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.07.013

[20] S. Chaudhary and S. Murala, "Deep network for human action recognition using weber motion," *Neurocomputing*, vol. 367, pp. 207–216, Nov. 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.08.031

[21] M. Ma, N. Marturi, Y. Li, A. Leonardis, and R. Stolkin, "Region-sequence based six-stream CNN features for general and fine-grained human action recognition in videos," *Pattern Recognition*, vol. 76, pp. 506–521, Apr. 2018. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.11.026

[22] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018. [Online]. Available: https://doi.org/10.1109/access.2017.2778011

[23] J. Chen, R. D. J. Samuel, and P. Poovendran, "LSTM with bio inspired algorithm for action recognition in sports videos," *Image and Vision Computing*, vol. 112, p. 104214, Aug. 2021. [Online]. Available: https://doi.org/10.1016/j.imavis.2021.104214

[24] J. Shuiwang, X. Wei, Y. Ming, and Y. Kai, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan. 2013. [Online]. Available: https://doi.org/10.1109/tpami.2012.59

[25] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *CVPR 2011*. IEEE, Jun. 2011. [Online]. Available: https://doi.org/10.1109/cvpr.2011.5995496

[26] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. IEEE, 2005. [Online]. Available: https://doi.org/10.1109/vspets.2005.1570899

[27] J. C. Niebles, H. Wang, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," in *Procedings of the British Machine Vision Conference 2006*. British Machine Vision Association, 2006.

[28] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, apr 2018. [Online]. Available: https://doi.org/10.1109/tip.2017.2785279

[29] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015. [Online]. Available: https://doi.org/10.1109/cvpr.2015.7298714

[30] J. Tu, M. Liu, and H. Liu, "Skeleton-based human action recognition using spatial temporal 3d convolutional neural networks," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, jul 2018. [Online]. Available: https://doi.org/10.1109/icme.2018.8486566

[31] C. Li, P. Wang, S. Wang, Y. Hou, and W. Li, "Skeleton-based action recognition using LSTM and CNN," in *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*. IEEE, jul 2017. [Online]. Available: https://doi.org/10.1109/icmew.2017.8026287

[32] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3d action recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2842–2855, jun 2018. [Online]. Available: https://doi.org/10.1109/tip.2018.2812099

[33] Y. Hbali, S. Hbali, L. Ballihi, and M. Sadgal, "Skeleton-based human activity recognition for elderly monitoring systems," *IET Computer Vision*, vol. 12, no. 1, pp. 16–26, nov 2017. [Online]. Available: https://doi.org/10.1049/iet-cvi.2017.0062

[34] D. C. Luvizon, H. Tabia, and D. Picard, "Learning features combination for human action recognition from skeleton sequences," *Pattern Recognition Letters*, vol. 99, pp. 13–20, nov 2017. [Online]. Available: https://doi.org/10.1016/j.patrec.2017.02.001

[35] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, aug 2017. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.02.030

[36] Q. Nie, J. Wang, X. Wang, and Y. Liu, "View-invariant human action recognition based on a 3d bio-constrained skeleton model," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3959–3972, aug 2019. [Online]. Available: https://doi.org/10.1109/tip.2019.2907048

[37] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1963–1978, aug 2019. [Online]. Available: https://doi.org/10.1109/tpami.2019.2896631

[38] G. Batchuluun, J. K. Kang, D. T. Nguyen, T. D. Pham, M. Arsalan, and K. R. Park, "Action recognition from thermal videos using joint and skeleton information," *IEEE Access*, vol. 9, pp. 11 716–11 733, 2021. [Online]. Available: https://doi.org/10.1109/access.2021.3051375

[39] M.-F. Tsai and S.-H. Huang, "Enhancing accuracy of human action recognition system using skeleton point correction method," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 7439–7459, jan 2022. [Online]. Available: https://doi.org/10.1007/s11042-022-12000-4

[40] N. Jaouedi, N. Boujnah, and M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 4, pp. 447–453, May 2020. [Online]. Available: https://doi.org/10.1016/j.jksuci.2019.09.004

[41] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 495–502.

[42] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* IEEE, 2004. [Online]. Available: https://doi.org/10.1109/icpr.2004.1334462

*Received on January 21, 2023*
*Accepted on September 19, 2023*