2D skeleton-based Human Action Recognition using Action-Snippet Representation and Deep Sequential Neural Network

by

Aizada Askar

Submitted to the School of Engineering and Digital Sciences
in partial fulfillment of the requirements for the degree of

Master of Science in Data Science

at the

NAZARBAYEV UNIVERSITY

Apr 2022

Author: .....
Aizada Askar
School of Engineering and Digital Sciences
Apr 29, 2022

Certified by .....
Nguyen Anh Tu
Assistant Professor
Thesis Supervisor

Certified by .....
Min-Ho Lee
Assistant Professor
Thesis Co-supervisor

Accepted by .....
Vassilios D. Tourassis
Dean, School of Engineering and Digital Sciences

2D skeleton-based Human Action Recognition using Action-Snippet Representation and Deep Sequential Neural Network

by

Aizada Askar

Submitted to the School of Engineering and Digital Sciences
on Apr 29, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science in Data Science

Abstract

Human action recognition is one of the crucial and important tasks in data science. It aims to understand human behavior and assign a label on performed action and has diverse applications. Domains, where this application is used, includes visual surveillance, human–computer interaction and video retrieval. Hence, discriminating human actions is a challenging problem with a lot of issues like motion performance, occlusions and dynamic background, and different data representations. There are many researches that explore various types of approaches for human action recognition. In this work we propose advanced geometric features and adequate deep sequential neural networks (DSNN) for 2D skeleton-based HAR. The 2D skeleton data used in this project are extracted from RGB video sequences, allowing the use of the proposed model to enrich contextual information. The 2D skeleton joint coordinates of the human are used to capture the spatial and temporal relationship between poses. We employ BiLSTM and Transformer models to classify human actions as they are capable of concurrently modeling spatial relationships between geometric characteristics of different body parts.

Thesis Supervisor: Nguyen Anh Tu
Title: Assistant Professor

Thesis Co-supervisor: Min-Ho Lee
Title: Assistant Professor

Acknowledgments

I am deeply grateful to my primary supervisor, Anh Tu Nguyen, who guided me throughout this project. Also, I wish to acknowledge the help provided by the technical opportunities of Department of Computer science of the Nazarbayev University.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview and motivation

Vision-related tasks have become progressively important for the scientific society. The reason is that they can be useful for multimedia content analysis[1], human behavior understanding and event interpretation[2]. They become a central function of broad spectrum applications, from intelligent surveillance to smart healthcare, which aim to automatically understand the scenes and activities of the subjects within an environment. One of the most challenging problems in the vision field is video-based human action recognition (HAR) whose fundamental goal is to identify the activities taking place in the video. For example, HAR can be used for health monitoring for the elderly or kid who stay alone at home. The application can detect dangerous moments such as falling down or other injuries[3]. Another important application of a HAR system is automatic detection of crimes[4][14].

The HAR is one of the challenging problems in Deep learning. There are a variety of issues in the human action recognition field including anthropometrical variation, view variations, background variations, occlusion, illumination variation and insufficient data[5]. Anthropometrical variation refers to the measurement of the individual. Person movements can be quite complex and present infinite variability depending on the body parts, age or angles of view. Also the background in front of which human actions are performed is an important to accurately recognise actions. Human action

recognition systems give good results when the background is static and uniform. Occlusion issue refers to the vanishing of human body parts by being behind another object. The purpose of the human action recognition model is to classify a type of performed actions effectively. Input data for training can be presented in various formats like infrared data, structured-light-based data, time-of-flight-based data or RGB videos that makes it a challenging topic. There are a lot of approaches for human action recognition to deal with these kind of problems.

1.2 Problem statements

In general, human action recognition can be investigated by non-skeleton based and by skeleton based approaches. The first one refers to the algorithms that directly learn or recognize activities from RGB frames. The works in the first category are often based on handcrafted or deep features[6][25][26], and detect foreground subjects by using techniques based on image analysis which can extract key features such as keypoints, shapes and regions[10]. However, this category is usually sensitive to the quality of foreground detection techniques, background movement, and shadows. Meanwhile, a skeleton based approaches learn human poses and employ skeleton data for HAR, where 2D/3D pose data is either gained by human pose estimation algorithms[23] or directly received from RGB-D cameras. A wide range of features like leg acceleration, body velocity and elbow rotation can be calculated from the human skeleton[11], which are concise, intuitive, and easy for differentiating various human actions and extracting contextual information. A lot of works have been done based on this 3D skeleton data that are acquired from RGB-D cameras[7][8][9]. However, despite the advantages provided by a depth camera, it presents several limitations like poor performance in outdoor environment. Now, a single RGB camera can generate 2D skeletons, solving the limitations of previous 3D skeleton-based approaches[12]. The 2D skeletons give opportunity to investigate new approaches for action recognition by borrowing body processing techniques from 3D skeleton approaches. Recent works have applied RNN and LSTM to 2D skeleton-based HAR data and obtained outstand-

ing results[11][12][13] because they can automatically learn the temporal structure and effectively derive information from the sequential data. Despite this, due to the use of simple framewise representation as an input of RNN/LSTM (e.g., joint coordinates), these approaches have not completely exploited the geometric relationship between body joints or leveraged pose transitions across frames. This may prevent extracting the informative representation leading to the difficulty of discriminating pose-similarity actions (e.g., sitting down vs. standing up).

1.3 Aims and objectives

In this project, to tackle the technical issues of current skeleton-based approaches, we explore more advanced geometric features and employ adequate deep sequential neural networks (DSNN) for 2D skeleton-based HAR. Specifically, we first treat an action video as a sequence of action-snippets, where each is a sub-sequence of 2D consecutive skeletons. Subsequently, we propose a highly discriminative representation of action-snippet based on feature extraction schemes that exploit geometric relationship and body transition (e.g., joint distance, angle, velocity). These schemes enable the skeletal representation to be less sensitive to the inter-class action variability. Then, we use BiLSTM, an advanced version of LSTM, for classifying activity mapped from the sequence of the skeletal representations. As an alternative to BiLSTM, we further employ Transformer, which is a current leading DSNN model for Natural Language Processing (NLP)[27][28] and has recently achieved remarkable results on several recognition tasks in computer vision[29][30]. Both BiLSTM and Transformer are capable of concurrently modeling spatial relationships between geometric characteristics of different body parts and capturing the temporal dependencies in terms of inter-frame correlation.

1.4 Key contributions

In summary, the main contributions of this project are: 1) Different geometric feature extraction schemes are studied to capture the spatial and temporal relationship between poses in an action-snippet. 2) Effective DSNN models (BiLSTM and Transformer) are introduced to thoroughly learn the deep correlations of consecutive action-snippets in a long skeleton sequence.

The rest of the paper is structured as follows: In Chapter 2, related works are presented. Chapter 3 describes the proposed method in detail, including 2D skeleton extraction, feature extraction and DSNN models. In Chapter 4 comparative experiments are demonstrated to evaluate the effectiveness of proposed model. Finally, Chapter 5 concludes the paper.

Chapter 2

Related work

The HAR methods can be categorised as Feature based approaches and Model based approaches. The first one refers to SCAR (subspace clustering based approach), Linear Coding (LLC), Histogram of Oriented Features/Gradients (HOF/HOG) and Spatio-temporal interest points (STIP) techniques[15][16][17]. The LLC is a common and effective technique which can be used for spatio-temporal features. In [10] 2D spatial temporal samples are computed to retrieve SIFT features of multi layer patches. Then these features are encoded by the LLC algorithm. It will capture the correlations between similar descriptors which helps to solve action recognition problem. Scar is a clustering technique whose goal is to find that subspaces of data points and discriminates data based on these subspaces. Paoletti and Beyan in [16] uses SCAR method for action recognition to improve the distinguishability of action and a timestamp clipping method. It gives an opportunity to better manage the temporal information of the data. The HOG/HOF technique counts occasions of gradient orientation in visible parts of an image. In [17] this technique is used to retrieve contextual information from motion image sequences. By using the HOG features recognition rate can be improved.

These methods use human insights to solve action recognition problems and are often restricted to common machine learning tasks. In contrast model based approaches design human model and can be solved with deep learning tasks. In this approach features are extracted from a sequence of frames and a model is built us-

ing human pose. Human poses can be 2D data acquired by RGB cameras and 3D skeletons acquired by RGB-D cameras. The proposed model are strongly related to skeleton based methods. There are a lot of works done using 3D skeleton data that provides significant advantages, for example the motion information of actions can be more discriminative[18].

The earlier works [19] retrieve action templates from 3D data and use them as the training samples of the actions to train several classifiers. The raw 3D coordinates can be preprocessed by using techniques such as rotation, translation and scaling. The normalized coordinates are used to define key poses with pose kinetic energy. After that the atomic action templates are computed using these key poses. The extracted features are then fed into a classification model. Gao et al [20] propose a novel approach based on Graph Convolutional Network with 3D skeleton data. They extract 3D human skeleton data by a multitasking method which combines multi-modal data from depth data streams and color data streams. The Graph Convolutional Network model is applied on 3d data which divides human skeleton into five regions to extract internal and inter-regional features.

However, researchers started studying new methods that can give similar results using just RGB cameras as 3D data has several limitations. The 3D skeletons are often noisy because of the complexity in localizing body parts, sensor range errors and occlusions[21]. Also, sensors have a minimal working range which makes difficult to implement in surveillance conditions. With this motivation there have been developed 2D pose estimators such as OpenPose, PoseNet and DeeperCut. The OpenPose is much more accurate and meant to be run on GPU powered systems.

In [12] Angelini et al. propose a novel 2D skeleton based approach, the ActionX-Pose. This technique retrieves low and high level features from 2D skeleton data. Authors deal with occlusions that can result in constant or short-time missing data. To handle occlusions they use high level features, landmark borrowing and short time interpolation. The first strategy is provided for the problem of persistent occlusions, also the second one improves low level features if there are persistent occlusions. The third one deals with the problem of short time occlusion. In this work, a new dataset,

named ISLD, is proposed by authors which was collected in a private Intelligent Sensing Lab.

Recurrent Neural Network (RNN) is a well-known model for sequential data modeling which is commonly applied to solve the timing issues of a sequence. Moreover, the poor memory problem of RNN can be solved by using LSTM. Thus it became a common approach to deal with the sequence problem, having advantages over CNN models. One of the recent works [22] proposes a two branch stacked RNN-LSTM model for action recognition problems based on 2D skeleton data. In this work skeleton data is segmented into two parts to extract local features. Segmentation reduces the feature space with respect to the entire network. Extracted features are fed into the classifier. Proposed model is composed of two branches. Each branch contains N-stacked LSTM-RNNs that receives an informative context on the input data. This model is applied on KTH and Weizmann datasets. Despite the advantages that RNN-LSTM can provide they need to be further investigated to deal with model size and computation speed issues.

A lot of works for HAR problems mostly based on 3D data rather than 2D skeleton data. In this work we present advanced geometric features and adequate deep sequential neural networks (DSNN) for 2D skeleton-based HAR to tackle the technical issues of current skeleton-based approaches.

Chapter 3

Methodology

3.1 Architecture overview

In this section, a proposed approach for the HAR problem is described. The overall workflow consists of four main steps as shown in Figure 3-1. The system takes video frames as an input. The first step of the algorithm is to detect the human skeleton from each frame of video using OpenPose software. The output of the OpenPose algorithm is joint coordinates of human skeleton[23]. After skeleton detection, a sliding window of size W aggregates the skeletons of the W consecutive frames to form the action-snippet. The skeleton data in each snippet is used to extract joint-based and distance-based features to obtain highly discriminative representation of action snippet. Then the PCA algorithm is adopted to reduce the dimensionality of joint-based and distance-based feature vectors because it reduces the time and storage space required and helps remove redundant features, if any. These reduced feature vectors are concatenated for further classification. Specifically, in the last step, features are fed into BiLSTM and Transformer classifiers to obtain final recognition result. These DSNN models are able to simultaneously capture the temporal dependencies in terms of inter-frame correlation and build spatial relationships between geometric characteristics of different body parts.

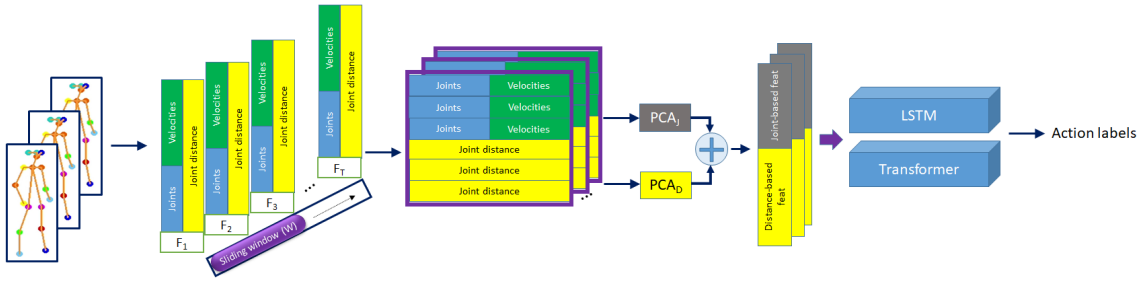

Figure 3-1: The HAR framework. The detected human skeletons(1) are fed into feature extraction algorithm(2) with sliding window of size N frames to extract joint based and distance based features. Extracted features are fed into PCA algorithm(3) to reduce feature vector dimension. Then features are fed into LSTM and Transformer classifiers(4).

3.2 Preprocessing

In general, 2D human skeleton data can be extracted from RGB video frames by using with the algorithm like OpenPose for human pose estimation. The main idea of OpenPose is producing two heapmaps with the Convolutional Neural Network. The first heapmap predicts joint positions, and the second one for associates the joints into human skeletons. Subsequently, the skeleton of each subject can be represented as 18 or 25 joint coordinates, and each joint is defined as a point with coordinate (x,y) in the 2D space.

The 2D skeleton data extracted from video sequences are preprocessed to deal with missing data. In some cases, some joint positions might be missed because of occlusions or OpenPose can fail to extract full skeleton joints from the image, leading to some nulls in the joint positions. These missed joints are replaced with its relative position in the previous frame in relation to the neck.

3.3 Feature extraction

The feature extraction stage aims to process the raw 2D skeleton data to retrieve more salient features within the input video sequence. These features enable the skeletal representation to be less sensitive to the inter-class action variability. We form up action-snippet by aggregating the skeleton data of the W consecutive frames

and extract two kind of features including joint based and distance features. Notably, a full set of skeletal joints in a given sequence of skeletons in W frames are expressed as $\mathcal{S} = \{\mathbf{s}_i^t = (x_i^t, y_i^t) | i \in [1, N], t \in [1, W]\}$, where N is the number of body joints, *i-th* joint $\mathbf{s}_i^t$ is a point with 2D coordinate at frame t.

3.3.1 The joint-based features

- **A concatenated joint positions($\mathbf{s}_{joints}$)** of W frames. This feature is needed for calculating subsequent features.

$$\begin{aligned} \mathbf{s}_i &= [s_i^1, s_i^2, ..., s_i^W] \\ \mathbf{s}_{joint} &= [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N] \end{aligned} \tag{3.1}$$

- **An average height(H)** of skeleton of consecutive W frames of sliding window. It is the length from neck to thigh. The height feature is needed to normalize all features described below, where the dimension of average height should be approximately equal to 1.

- **The moving velocity of the body($\mathbf{v_{body}}$)** feature is computed by dividing velocity of the neck($\mathbf{v_{center}}$) in $\mathbf{s}_{joints}$ by average height H:

$$\mathbf{v_{body}} = \frac{\mathbf{v_{center}}}{H} \tag{3.2}$$

  where $\mathbf{v_{center}}$ is computed as:

$$\mathbf{v_{center}} = [\mathbf{s}_1^1, (\mathbf{s}_1^2 - \mathbf{s}_1^1), ..., (\mathbf{s}_1^W - \mathbf{s}_1^{W-1})] \tag{3.3}$$

  and where $\mathbf{s}_1$ is a position of neck point.

- **The normalized joint positions(s)** is equal to:

$$\mathbf{s}_{norm,i}^t = \frac{\mathbf{s}_i^t - \bar{\mathbf{s}}_{\mathbf{joints}}}{H} \tag{3.4}$$

where $\bar{\mathbf{s}}_{joints}$ is a mean of joint positions in W frames.

- **The velocities of joints(v)** are computed using normalized joint positions in W frames:

$$\mathbf{v} = [\mathbf{s}^1_{norm,i}, (\mathbf{s}^2_{norm,i} - \mathbf{s}^1_{norm,i}), ..., (\mathbf{s}^W_{norm,i} - \mathbf{s}^{W-1}_{norm,i})] \tag{3.5}$$

After computing joint based features, normalized joint positions, velocity of body and velocity of joints are concatenated before the further processing to form up F_{joint} feature vector:

$$\mathbf{F_{joint}} = [\mathbf{x}, \mathbf{v_{body}}, \mathbf{v}] \tag{3.6}$$

2)**The distance feature(d)**. The Euclidean distance of each pairs (i, j) of joint positions at frame t is computed for distance feature using following equation:

$$d^t_{ij} = \sqrt{(x^t_i - x^t_j)^2 + (y^t_i - y^t_j)^2} \tag{3.7}$$

Accordingly, we can retrieve $n(n-1)/2$ distances in the frame t and arrange them into the following frame-wise representation:

$$\mathbf{f}^t_{dist} = [d^t_1, d^t_2, ..., d^t_K] \tag{3.8}$$

where $K = n(n-1)/2$. Then, the distance feature of the given snippet is calculated as below:

$$\mathbf{F}_{dist} = [\mathbf{f}^1_{dist}, \mathbf{s}^2_{dist}, ..., \mathbf{s}^W_{dist}] \tag{3.9}$$

After extracting features we apply PCA algorithm for both joint based (F_{joint}) and distance features (F_{dist}) to reduce feature vector. With PCA algorithm we can select useful patterns based on the correlation between features. Then the reduced two feature vectors are concatenated to form up F feature vector:

$$\mathbf{F} = [\mathbf{F_{joint}}, \mathbf{F_{dist}}] \tag{3.10}$$

3.4 Deep Human Activity Recognition models

The fourth step of analysing the action of person over time and predicting is done using deep sequential neural networks. We develop two deep sequential models including BiLSTM and Transformer to coordinate the sequential information in the extracted features and effectively exploit the correlation of action-snippet sequence. In other words, using these models allow us to thoroughly learn the discriminative representation of spatiotemporal features for HAR tasks.

3.4.1 The BiLSTM model

Our first model is BiLSTM which is based on the LSTM architecture. The LSTM network model was proposed by Hochreiter and Schmidhuber[24]. The basic LSTM cell architecture consist of three gates, namely, forget(f_t), input(i_t) and output(o_t) gates to control cell states. The forget gate(f_t) determines whether to retain or forget the information of previous state(c_{t-1}) by using sigmoid activation function. Similarly, the input gate uses sigmoid function to determine how much information of the input($F = (F^1, F^2...F^T)$) needs to be saved to current cell state(c_t). The third output gate controls how much information of current state is passed to the current hidden state(h_t). The mathematical representation of these gates is as follows:

$$f_t = \sigma(U_f F^T + V_f h_{t-1} + b_f) \tag{3.11}$$

$$i_t = \sigma(U_i F^T + V_i h_{t-1} + b_i) \tag{3.12}$$

$$o_t = \sigma(U_o F^T + V_o h_{t-1} + b_o) \tag{3.13}$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(\sigma(U_o F^T + V_o h_{t-1} + b_o)) \tag{3.14}$$

$$h_t = o_t * \tanh(c_t) \tag{3.15}$$

where σ is a sigmoid activation function, F^T indicates in our case t-th action-snippet feature vector(F) of the video, h_t is a hidden state, U and V terms represent weigth matrices and b term is a bias vector. The single LSTM cell captures information only

in a forward direction which means only previous state context is available for the current state. In sequential data modeling, information from the future state also plays an important role. It could be easier for the network to understand what the next action is by using the information from the future. Thus, in our research work we present Bidirectional Long Short Term Memory network model for human activity recognition problem. It is a sequence processing model that runs input in two ways, one runs the input in a forward direction and the second takes input in backward direction, as shown in Figure 3-2. The Bidirectional LSTM can improve the context available to algorithm by increasing the amount of information preserved from past and future. This model is a good fit for the problem of action recognition as video can be represented as a sequence of action-snippets. The "many-to-one" architecture is used in designing the model, it accepts sequence of feature vectors to convert them to a probability vector at the output for classification. The input sequence vector F_t is given for BiLSTM model, where t is a length of action-snippets of video. The proposed architecture consist of two LSTM cells as hidden layers. The first cell runs from past to future $\overrightarrow{h_t} = o_t * \tanh(c_t)$ and the second cell runs from future to past $\overleftarrow{h_t} = o_t * \tanh(c_t)$ through time steps with M hidden layer units. This will add deepness to the neural network. In the end the outputs of forward and backward layers are concatenated $z_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$.

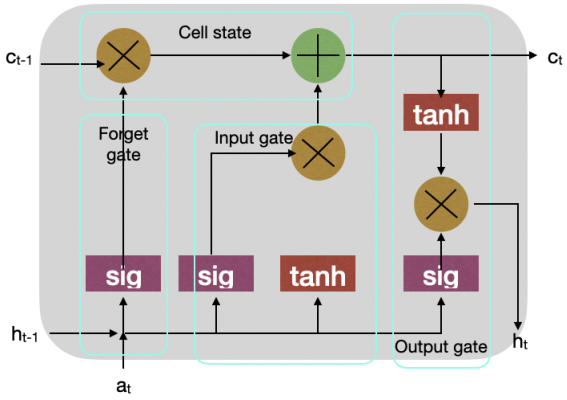

Figure 3-2: The LSTM architecture.

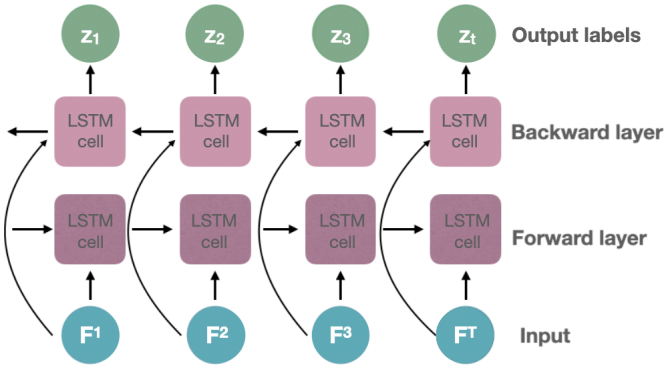

Figure 3-3: The BiLSTM architecture. The first layer runs the input in a forward direction and the second layer takes the input in a backward layer. Then the outputs of forward layer and backward layer are concatenated.

3.4.2 Transformer model

The second model that we use in our work is Transformer model. It is a deep learning model that uses the mechanism of self-attention. The Transformer architecture follows an encoder-decoder structure. For this project, we propose Transformer model with encoder block to classify human actions, as illustrated in Figure 3-4. The attention mechanism of Transformer is represented as follows:

$$Attention = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.16}$$

where Q, K and V matrices are constructed from the input features through trainable linear transformations, as follows:

$$\begin{aligned} Q &= FU^Q \\ K &= FU^K \\ V &= FU^V \end{aligned} \tag{3.17}$$

where Q is a query matrix that is vector representation of one action-snippet in the video sequence, K is a keys matrix that is vector representation of all action-snippets in video sequence and V are values of all action-snippets in sequence given in vector representation and U^Q, U^k and U^v are trainable weight matrices. The softmax

function is applied to calculate attention scores. These scores determine how much attention to pay on other action-snippets in a sequence. The self-attention layers of Transformer are order-agnostic. Transformer model needs to take into account order information since videos after feature extraction represented as an ordered sequences of action-snippets. For this purpose we use positional encoding which simply embeds the positions of the action-snippets through an Embedding layer that turns positive indexes into dense vectors of fixed size. Then the positional embeddings are added to the precomputed feature maps. After that embedding vectors are fed into the encoder block, consisting of the two sublayers multi head attention and fully connected feed forward network layers followed by normalization layers. Finally, the output of the encoder passes through a softmax layer, to generate a prediction.

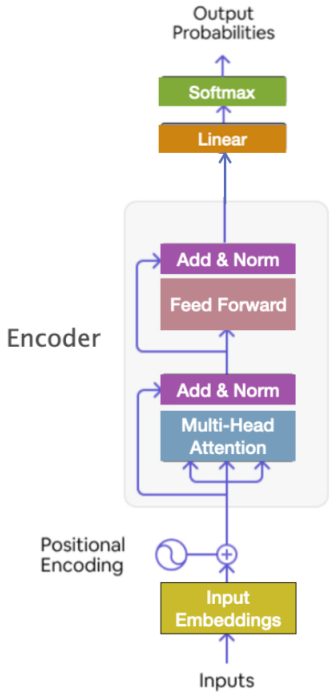

Figure 3-4: The Transformer architecture. Embedding layer takes the input and turns positive indexes of frames into dense vector. Then embedding vectors with feature vector are fed into the encoder block that contains fully connected feed forward network layer and multi-head attention layer followed by normalization layers. The output of the encoder passes through a softmax layer

Chapter 4

Experimental results and discussions

4.1 Experimental setup

4.1.1 Dataset

We experiment with two 2D skeleton-based datasets, as JHMDB and MHAD datasets to evaluate our proposed approach.

- JHMDB dataset contains 928 videos and 21 types of human actions where each video consists of 32 frames. It is a subset of a large HMDB51 dataset collected from Youtube videos and has 3 training and test splits. The 2D skeletons in the dataset are interpreted from RGB videos and consist of 15 joint positions. The data is performed on JHMDB dataset as it has small training set.

- MHAD dataset contains 6 types of actions. In total there are 1438 videos made up 211200 frames. Pose estimations of dataset are made using the OpenPose software. The 2D skeletons consists of 18 joint positions.

4.1.2 Experimental settings

Experiments were conducted on macOS 15.10.7 running on a MacBook Pro 2017 with dual core Intel i5, and with graphics Intel Iris Plus Graphics 640. The feature computations and the classification were performed using the Jupyter notebook interactive

computing platform.

4.1.3 Data Augmentation

Data augmentation was performed to increase three training splits of JHMDB dataset for the classification requirements. A large amount of data is usually required by Deep learning networks to generate good results. In this paper, pose-flipping, rotation and pose-shifting techniques were used to augment training data.

The first technique, pose-flipping, flips the skeleton along the vertical axis. The flipped skeleton looks mirrored, using right and left body symmetry as illustrated in Figure 4-1. The pose-flipping technique doubles the original training dataset.

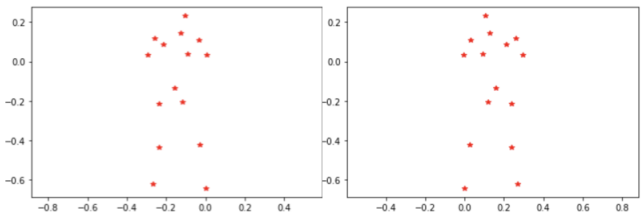

Figure 4-1: The pose-flipping.

The second augmentation method is rotation. It is used to mimic multiple views and helps the model to recognize an action in cross-view. The skeleton is rotated by 15°, 30°, and in opposite side around the origin as shown in Figure 4-2. Let (x, y) be a joint coordinate. The coordinates (x_r, y_r) of a joint position at (x, y) after rotation can be defined as:

$$\begin{aligned} x_r &= x\cos\theta - y\sin\theta \\ y_r &= y\cos\theta + x\sin\theta \end{aligned} \tag{4.1}$$

We augment flipped training data thus increasing original data by ten times rotating them in four direction.

The third technique is pose-shifting which adds Gaussian noise to joint coordinates, specifically $N(0,\ \sigma^2)$ with 0 mean and σ standard deviation. In this project we use $\sigma = 0.002$ and $\sigma = 0.004$ to add some noise. Adding noise means that we

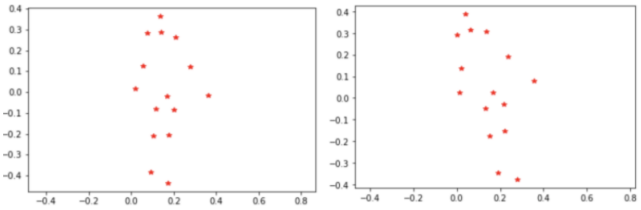

Figure 4-2: The rotation.

shift the skeleton by adding σ noise to each joint position, as shown in Figure 4-3. The pose-shifting method augments the original data with two different standard deviations thus increasing training dataset by two times. Overall, data augmentation method increases training dataset by twelve times thus training sets of three splits contains in average 7942 video samples.

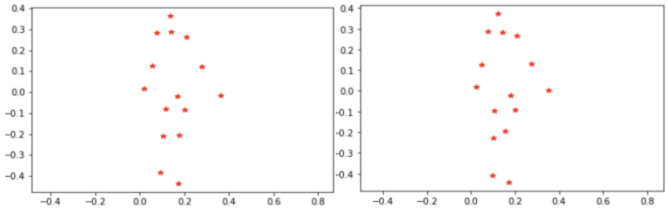

Figure 4-3: The pose-shifting.

4.1.4 Training setup

Initially, we applied preprocessing techniques to each dataset as introduced before. We chose W=5 frames after experiments as window size for feature extraction. Concatenated features of JHMDB and MHAD datasets form a feature vector of dimension 470 and 524 respectively and distance based feature form a vector of dimension 525 and 765 respectively. Then we adopted PCA algorithm to reduce joint-based feature to 100 and distance-based feature to 50 dimensions as in this case we have less variance among data points in feature vector. To sum up resultant concatenated feature vector has a dimension of 150.

For this study, we trained Transformer and Bidirectional LSTM models on two datasets. The different number of encoder layers(1,2,5) and number of heads(1, 6, 8, 16) with Adam optimizer were used in Transformer model. The model was trained

for total of 30 epochs. After each epoch, we evaluated the model with the test data. For BiLSTM model, Adam optimizer with decaying learning rate with 0.05 initial learning rate was used. Training has been run using variety of hidden units(196, 200) per LSTM layer and batch size for total of 350 epochs.

4.1.5 Evaluation metrics

To evaluate our two trained models, we use average accuracy for 10 iterations and average accuracy on three splits of JHMDB dataset. We tune different hyperparameters such as number of encoder layers, number of heads, number of hidden units, decaying rate and batch size to compare model complexity and time complexity and to find best parameters.

4.2 HAR accuracy with different features and their combination

4.2.1 HAR accuracy with different window sizes

It is worth mentioning that with $W = 1$, we obtain a frame-wise representation. However, the static pose in one frame might be insufficient and difficult to discriminate some actions (e.g., standing up and sitting down) having similar poses. Therefore, the use of action-snippet (with a predefined window size $W > 1$) is an effective solution to address this issue since we consider the temporal order of consecutive poses when calculating the features. Moreover, the joint-based feature and distance features are complementary to each other. While the joint-based feature captures the spatial and motion information of different joints, the distance feature represents the geometric relationship between different body joints. Hence, their combination helps to improve the robustness of action-snippet representation. We experimented with different sliding window sizes such as W=1,3,5,8 to form the action snippets, as shown in Table 4.1. After experiments we chose W=5 as an optimal size for sliding window because very short or very long action snippets can not give adequate representation.

Table 4.1: HAR accuracy with different window sizes

Dataset	Model	Window size	Accuracy
JHMDB	Transformer	1	60.7%
	Transformer	3	64.5%
	Transformer	5	**66.8%**
	Transformer	8	64.6%
JHMDB	BiLSTM	1	59.9%
	BiLSTM	3	63.6%
	BiLSTM	5	**70.2%**
	BiLSTM	8	62.7%
MHAD	Transformer	1	99.1%
	Transformer	3	97.3%
	Transformer	5	**99.4%**
	Transformer	8	98.5%
MHAD	BiLSTM	1	96.3%
	BiLSTM	3	98.2%
	BiLSTM	5	**99.3%**
	BiLSTM	8	97.2%

4.2.2 HAR accuracy of Transformer model with different parameters

After finding optimal sliding window size the effectiveness of encoding structure (as illustrated in Fig. 3-4) with different parameters was evaluated on the JHMDB and MHAD dataset, and the classification results are listed in Table 4.2 and Table 4.3, respectively. The Table 4.3 shows that the proposed transformer model can effectively capture spatio-temporal data with 1 encoder layer and the number of heads of 16. Increasing the number of encoding layers negatively effects on the HAR accuracy because the more encoding layers, the more complex the model which can lead to overfitting. However, increasing the number of heads improves the HAR accuracy. This can be explained by that having several heads per layer makes the model capable to try out several pathways at once. So, the model can learn different patterns with each head. Moreover, data augmentation is applied on JHMDB dataset as it has small training dataset. The data augmentation can improve the performance of the Transformer model by around 5%. So, the Transformer has a good scalability.

Table 4.2: Transformer model accuracy on JHMDB with different parameters

Dataset	**Encoder layers**	**Number of heads**	**Accuracy without data augmentation**	**Accuracy with data augmentation**
JHMDB	1	1	60.1%	66.8%
	1	6	61.3%	66.9%
	1	8	**62.4**%	**67.7**%
	1	16	**64.5**%	**70.9**%
	2	1	62.5%	65%
	2	6	60.3%	63%
	2	8	62.5%	64%
	5	1	60.2%	63.6%
	5	6	55.6%	59.7%
	5	8	55.7%	57.9%

Table 4.3: Transformer model accuracy on MHAD with different parameters

Dataset	**Encoder layers**	**Number of heads**	**Accuracy**
MHAD	1	1	97.5%
	1	6	98.2%
	1	8	**99.3**%
	1	16	**99.4**%
	2	1	96.7%
	2	6	97.4%
	2	8	96.3%
	5	1	96.2%
	5	6	95.7%
	5	8	95.6%

After finding the best combination of hyperparameters, we further experiment with different batch sizes. The experiment results with batch sizes is listed in Table 4.4. As shown in the table, the optimal batch size is 64 which reaches to 74.7%.

Table 4.4: Transformer model accuracy on JHMDB with different batch sizes

Dataset	**Batch sizes**	**Accuracy**
JHMDB	16	70.9%
	32	71.5%
	64	**74.7**%
	128	71.8%
	256	70.7%
	512	68.7%

4.2.3 HAR accuracy of BiLSTM model with different parameters

The performance of BiLSTM network (as illustrated in Fig. 3-3) with different hidden units, batch sizes and decay rate parameters was evaluated on the JHMDB and MHAD dataset. Conducted experiments demonstrate that the BiLSTM better performs with 196 hidden units, 2048 batch size and decay rate of 0.96. Increasing batch size and keeping number of hidden units closer to actual number of features improves the accuracy and ability to generalize.

4.2.4 Comparisons with state-of-the-arts

The HAR results of JHMDB dataset are listed in Table4.5 and the results of MHAD dataset are presented in Table 4.6. Overall, Transformer and BiLSTM models with fewer parameters can reach good results on JHMDB and MHAD dataset. We choose the Adam optimizer fuction, with decaying learning rate for Transformer model and BiLSTM model. Initial learning rates are 0.00035 and 0.005 for Transformer model and BiLSTM model respectively. The confusion matrices show that Transformer and BiLSTM models are robust enough to each action class. Despite the fact that video

sequences of JHMDB dataset are acquired from outdoor environment, Transformer model is able to generalize the data. Moreover, our baseline models such as kNN, Random forest and MLP classifiers can reach comparable results on MHAD dataset, reaching highest 97% result.

Table 4.5: Comparison results on JHMDB dataset

Method	**Parameters**	**Speed on GPU**	**Accuracy**
DD-Net(filters=64)[31]	1.82M	2200 FPS	77.2%
DD-Net(filters=16)[31]	0.15M	3618 FPS	65.7%
DD-Net(with data augmentation)	1.82M	2200 FPS	74.4%
KNN	-	3450 FPS	59.1%
Random Forest	-	3745 FPS	63.5%
MLP	-	3750 FPS	61.5%
Transformer(layers=1, heads=16)	1.45M	3696 FPS	**74.7%**
Transformer(layers=1, heads=8)	0.83M	3875 FPS	**73.7%**
Transformer(layers=1, heads=1)	**0.11M**	3893 FPS	**69.7%**
Transformer(without data augmentation)	1.45M	3875 FPS	65.8%
Transformer(without PCA reduction)	49.09M	3472 FPS	63.6%
BiLSTM	0.65M	3540 FPS	70.2%

From overall conducted experiments, we can explore that when actions performed indoor environment(e.g MHAD dataset), Transformer and BiLSTM models can achieve superior results as 99%. When actions performed in outdoor environment(e.g JHMDB dataset), geometric representation and body transition characteristics of skeleton helps to improve performance, but not as considerably as in previous case. The Transformer model can handle its model size by changing the number of heads(1,8,16). The comparison results shows that when the number of heads is 16, Transformer model achieves its best performance, 74.7%, on JHMDB and MHAD datasets, resulting in larger number of parameters. Even so, the largest number of parameters

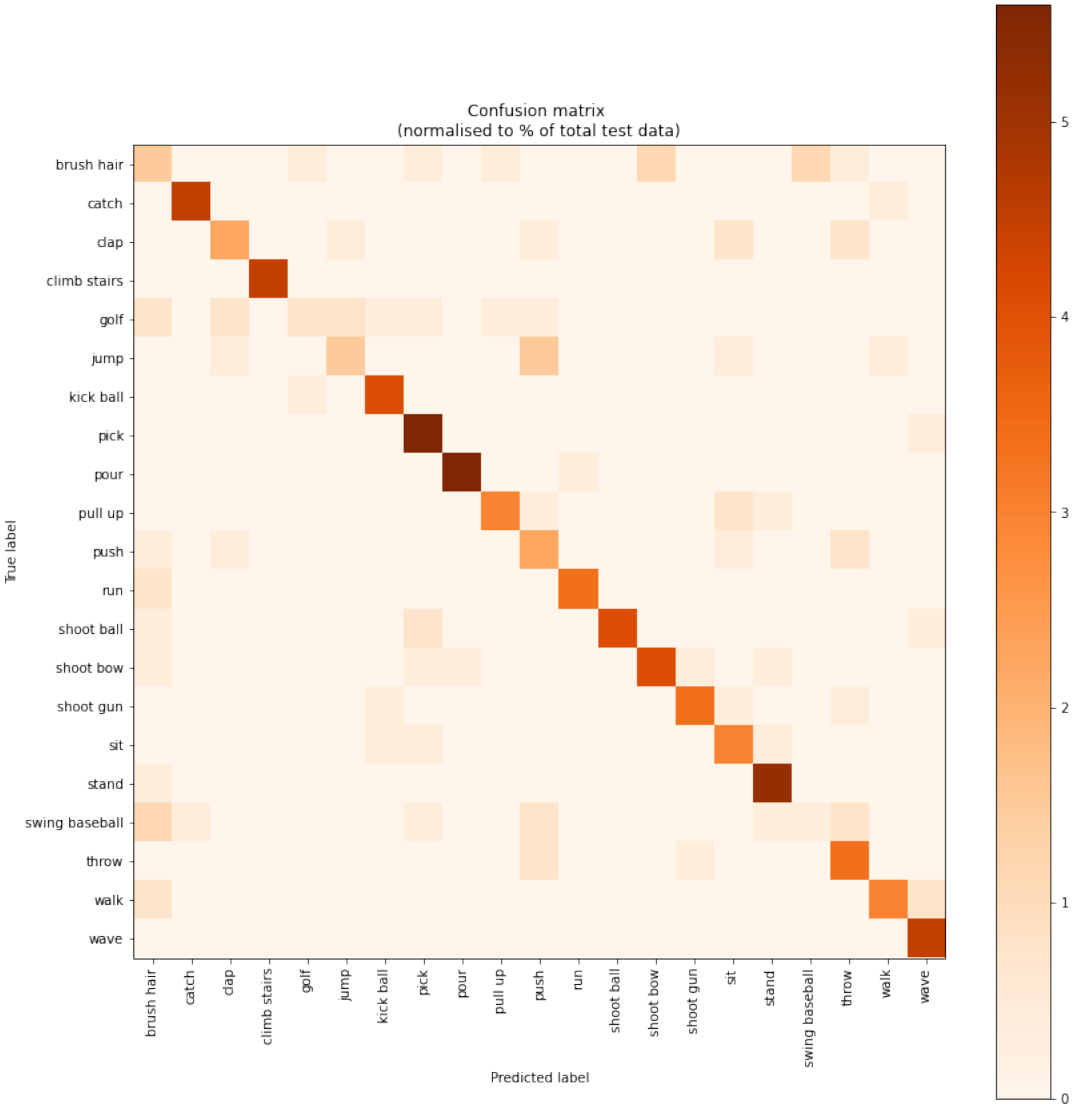

Figure 4-4: Confusion matrix of JHMDB dataset obtained by Transformer model.

of Transformer model is significantly less than the parameters of DD-net model[31]. The Transformer model can reach higher results comparing to the state-of-the-art methods, by using only 0.11 million parameters. Also, even though the accuracy of Transformer on the small dataset like JHMDB is lower than that of DD-net due to its data-hunger property, we can achieve the better performance if we have more data. In other words, Transformer has better scalability than DD-net. Also, according to the results, the number of trainable parameters are in direct ration with the dimension of feature vector, which means model complexity of Transformer model can be regularized with PCA algorithm. Moreover, using original feature vector can lead to overfitting. The conducted experiments show that adopting PCA algorithm significantly improves the Transformer accuracy because PCA algorithm chooses the most important patterns based on the correlation between features by avoiding redundant data. The BiLSTM model generates good results regarding to the speed and number of parameters, it can reach comparable results by using only 0.65 million parameters and classifying 3540 FPS.

Table 4.6: Comparison results on MHAD dataset

Dataset	**Methods**	**Accuracy**
MHAD	LSTM&RNN[32]	97%
	KNN	95.3%
	Random Forest	96.2%
	MLP	97.4%
	BiLSTM	**99.3%**
	Transformer	**99.4%**
	Transformer(without PCA reduction)	97.4%

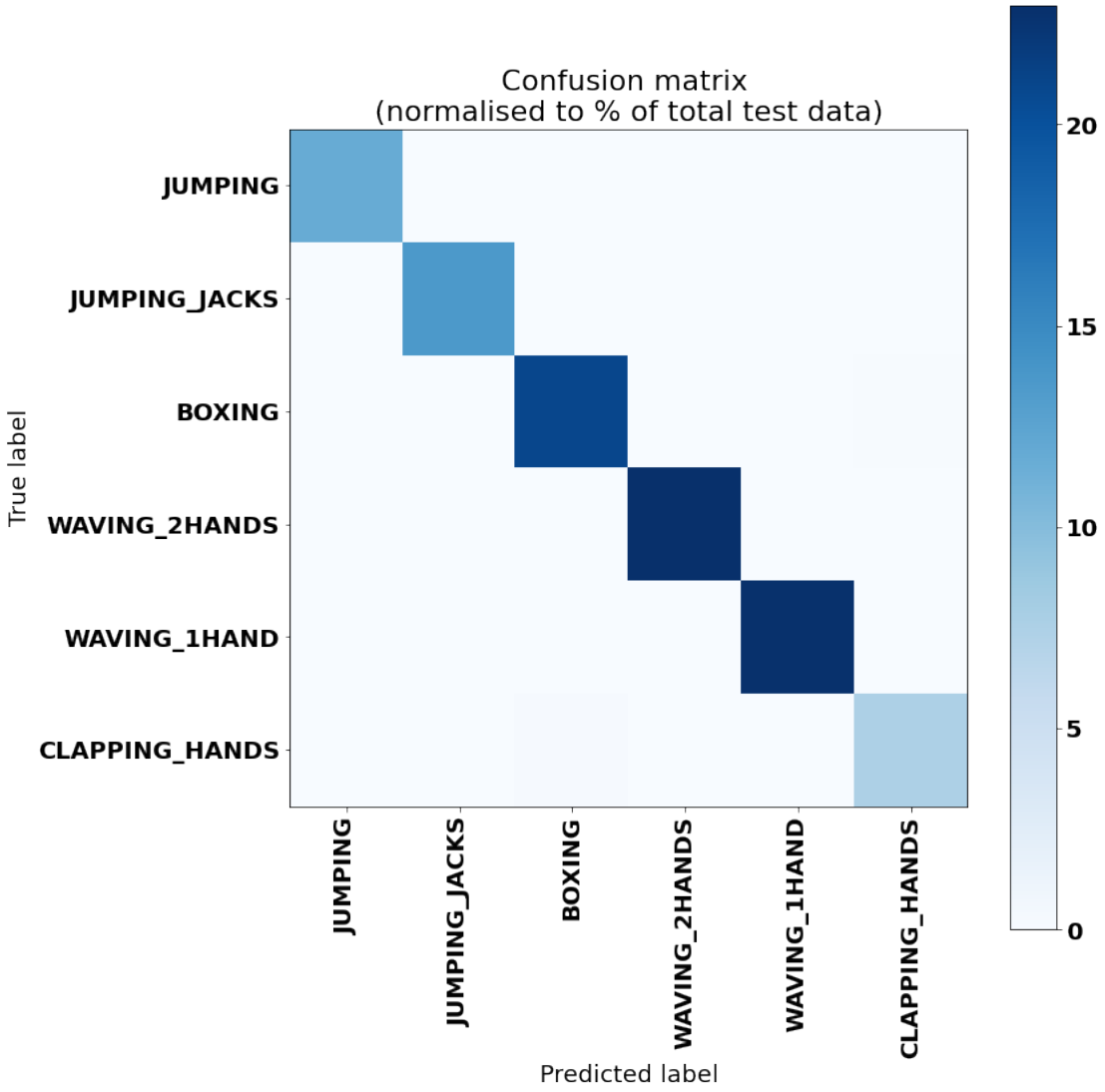

Figure 4-5: Confusion matrix of MHAD dataset obtained by BiLSTM model.

Chapter 5

Conclusion and future directions

By analyzing the geometric representation and body transition properties of skeleton within action snippets, we propose two joint-based and distance-based feature types to capture the spatial and temporal relationship between poses and a DSNN models for efficient 2D skeleton-based HAR. Effective Transformer and BiLSTM architectures are able to acuurately learn the deep correlations of consecutive action-snippets in a long skeleton sequence. Hense, the DSNN models, containing a few parameters, can achieve comparable results, with 74.7% on JHMDB dataset and 99% on MHAD dataset with Transformer model. These datasets were used to test the proposed method under several experimental conditions, including data augmentation techniques.

Future work will focus more on generalization ability of the method that is to make the system to work in complex surveillance scenarios. It should be investigated how further skeleton properties could be used to further improve the performance of action classification. Also, due to the simplicity of Transformer model, it can be approached for online action recognition.

Appendix A

Tables

Appendix B

Figures

Bibliography

[1] C. Yan, L. Li, C. Zhang, B. Liu, Y. Zhang and Q. Dai, "Cross-Modality Bridging and Knowledge Transferring for Image Understanding," in IEEE Transactions on Multimedia, vol. 21, no. 10, pp. 2675-2685, Oct. 2019, doi: 10.1109/TMM.2019.2903448.

[2] A. G. D'Sa and B. G. Prasad, "A Survey on Vision Based Activity Recognition, its Applications and Challenges," 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), 2019, pp. 1-8, doi: 10.1109/ICACCP.2019.8882896.

[3] Gao, Yongbin & Xiang, Xuehao & Xiong, Naixue & Bo, Huang & Lee, Hyo Jong & Alrifai, Rad & Jiang, Xiaoyan & Fang, Zhijun. (2018). Human Action Monitoring for Healthcare Based on Deep Learning. IEEE Access. 6. 1-1. 10.1109/ACCESS.2018.2869790.

[4] D. Roy and C. K. Mohan, "Snatch theft detection in unconstrained surveillance videos using action attribute modelling," Pattern Recognition Letters, vol. 108, pp. 1–9, 2018.

[5] Imen Jegham, Anouar Ben Khalifa, Ihsen Alouani, Mohamed Ali Mahjoub, Vision-based human action recognition: An overview and real world challenges, Forensic Science International: Digital Investigation, Volume 32, 2020, 200901, ISSN 2666-2817, https://doi.org/10.1016/j.fsidi.2019.200901.

[6] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local svm approach,” in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 3. IEEE, 2004, pp. 32–36.

[7] Y. Han, S. -L. Chung, Q. Xiao, W. Y. Lin and S. -F. Su, "Global Spatio-Temporal Attention for Action Recognition Based on 3D Human Skeleton Data," in IEEE Access, vol. 8, pp. 88604-88616, 2020, doi: 10.1109/ACCESS.2020.2992740.

[8] Y. Yang, C. Deng, S. Gao, W. Liu, D. Tao and X. Gao, "Discriminative Multi-instance Multitask Learning for 3D Action Recognition," in IEEE Transactions on Multimedia, vol. 19, no. 3, pp. 519-529, March 2017, doi: 10.1109/TMM.2016.2626959.

[9] X. Gao et al., "3D Skeleton-Based Video Action Recognition by Graph Convolution Network," 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), 2019, pp. 500-501, doi: 10.1109/SmartIoT.2019.00093.

[10] R.Poppe,“A survey on vision-based human action recognition,”Image and Vision Computing, vol. 28, no. 6, pp. 976–990, 2010.

[11] S. Wei, Y. Song and Y. Zhang, "Human skeleton tree recurrent neural network with joint relative motion feature for skeleton based action recognition," 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 91-95, doi: 10.1109/ICIP.2017.8296249.

[12] F. Angelini, Z. Fu, Y. Long, L. Shao and S. M. Naqvi, "2D Pose-Based Real-Time Human Action Recognition With Occlusion-Handling," in IEEE Transactions on Multimedia, vol. 22, no. 6, pp. 1433-1446, June 2020, doi: 10.1109/TMM.2019.2944745.

[13] X. Jiang, K. Xu and T. Sun, "Action Recognition Scheme Based on Skeleton Representation With DS-LSTM Network," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 7, pp. 2129-2140, July 2020, doi: 10.1109/TCSVT.2019.2914137.

[14] Sabokrou M, Fathy M, Hoseini M, Klette R (2015) Real-time anomaly detection and localization in crowded scenes. In: Proceedings of the IEEE CVPR Workshops, pp 56–62

[15] L. Liu, S. Ma and Q. Fu, "Human action recognition based on locality constrained linear coding and two-dimensional spatial-temporal templates," 2017 Chinese Automation Congress (CAC), 2017, pp. 1879-1883, doi: 10.1109/CAC.2017.8243075.

[16] G. Paoletti, J. Cavazza, C. Beyan and A. Del Bue, "Subspace Clustering for Action Recognition with Covariance Representations and Temporal Pruning," 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 6035-6042, doi: 10.1109/ICPR48806.2021.9412060.

[17] C. Huang, C. Hsieh, K. Lai and W. Huang, "Human Action Recognition Using Histogram of Oriented Gradient of Motion History Image," 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control, 2011, pp. 353-356, doi: 10.1109/IMCCC.2011.95.

[18] B. Liang and L. Zheng, "A Survey on Human Action Recognition Using Depth Sensors," 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2015, pp. 1-8, doi: 10.1109/DICTA.2015.7371223.

[19] J. Shan and S. Akella, "3D human action segmentation and recognition using pose kinetic energy," 2014 IEEE International Workshop on Advanced Robotics and its Social Impacts, 2014, pp. 69-75, doi: 10.1109/ARSO.2014.7020983.

[20] X. Gao et al., "3D Skeleton-Based Video Action Recognition by Graph Convolution Network," 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), 2019, pp. 500-501, doi: 10.1109/SmartIoT.2019.00093.

[21] P. Koniusz, A. Cherian, and F. Porikli, “Tensor representations via kernel linearization for action recognition from 3d skeletons,” in European Conference on Computer Vision. Springer, 2016, pp. 37–53.

[22] D. Avola, M. Cascio, L. Cinque, G. L. Foresti, C. Massaroni and E. Rodolà, "2-D Skeleton-Based Action Recognition via Two-Branch Stacked LSTM-RNNs," in IEEE Transactions on Multimedia, vol. 22, no. 10, pp. 2481-2496, Oct. 2020, doi: 10.1109/TMM.2019.2960588.

[23] Z. Cao, G. Hidalgo, T. Simon, S. -E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172-186, 1 Jan. 2021, doi: 10.1109/TPAMI.2019.2929257.

[24] Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. Neural Comput 1997; 9 (8): 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735

[25] F. Husain, B. Dellen and C. Torras, "Action Recognition Based on Efficient Deep Feature Learning in the Spatio-Temporal Domain," in IEEE Robotics and Automation Letters, vol. 1, no. 2, pp. 984-991, July 2016, doi: 10.1109/LRA.2016.2529686.

[26] Khan, M.A., Javed, K., Khan, S.A. et al. Human action recognition using fusion of multiview and deep features: an application to video surveillance. Multimed Tools Appl (2020). https://doi.org/10.1007/s11042-020-08806-9

[27] S. Singh and A. Mahmood, "The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures," in IEEE Access, vol. 9, pp. 68675-68702, 2021, doi: 10.1109/ACCESS.2021.3077350.

[28] B. Myagmar, J. Li and S. Kimura, "Cross-Domain Sentiment Classification With Bidirectional Contextualized Transformer Language Models," in IEEE Access, vol. 7, pp. 163219-163230, 2019, doi: 10.1109/ACCESS.2019.2952360.

[29] K. S. Krishnan and K. S. Krishnan, "Vision Transformer based COVID-19 Detection using Chest X-rays," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 644-648, doi: 10.1109/ISPCC53510.2021.9609375.

[30] L. Bashmal, Y. Bazi and M. A. Rahhal, "Deep Vision Transformers for Remote Sensing Scene Classification," 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, 2021, pp. 2815-2818, doi: 10.1109/IGARSS47720.2021.9553684.

[31] Fan Yang, Yang Wu, Sakriani Sakti, and Satoshi Nakamura. 2019. Make Skeleton-based Action Recognition Model Smaller, Faster and Better. In Proceedings of the ACM Multimedia Asia (MMAsia '19). Association for Computing Machinery, New York, NY, USA, Article 31, 1–6. DOI:https://doi.org/10.1145/3338533.3366569

[32] https://github.com/stuarteiffert/RNN-for-Human-Activity-Recognition-using-2D-Pose-Input