Human Action Recognition with Skeleton and Infrared Fusion Model

Amine Mansouri [1],*, Toufik Bakir [1], and Smain Femmam [2,3]

[1] Department of Electronics, Faculty of Sciences, University of Burgundy, ImViA Laboratory, Dijon, France
[2] Department of Computer and Networks, Faculty of Sciences, Haute-Alsace University, Mulhouse, France
[3] Department of Computer, Faculty of Sciences and Technology, Polytechnic Engineering School Paris-Cachan, France
*Correspondence: Amine.Mansouri@u-bourgogne.fr (A.M.)

***Abstract*—Skeleton-based human action recognition conveys interesting information about the dynamics of a human body. In this work, we develop a method that uses a multi-stream model with connections between the parallel streams. This work is inspired by a state-of-the-art method called FUSION-CPA that merges different modalities: infrared input and skeleton input. Because we are interested in investigating improvements related to the skeleton-branch backbone, we used the Spatial-Temporal Graph Convolutional Networks (ST-GCN) model and an EfficientGCN attention module. We aim to provide improvements when capturing spatial and temporal features. In addition, we exploited a Graph Convolutional Network (GCN) implemented in the ST-GCN model to capture the graphic connectivity in skeletons. This paper reports interesting accuracy on a large-scale dataset (NTU-RGB+D 60), over 91% and 93% on respectively cross-subject, and cross-view benchmarks. This proposed model is lighter by 9 million training parameters compared with the model FUSION-CPA.**

***Keywords*—deep learning, Human Action Recognition (HAR), convolutional neural networks, Graph Convolutional Networks (GCNs)**

I. Introduction

In the past decade, Human Action Recognition (HAR) has received increasing attention among researchers as it provides an understanding of videos and other types of acquisition devices used in medical applications such as human-computer interaction, as well as video retrieval, autonomous navigation systems and frequently in video surveillance [1–4]. In the early days, researchers focused their work on using Red, Green and Blue (RGB) or grayscale videos to feed the HAR models [5], due to their availability. In recent years, however, new modalities have emerged [6–8] using point cloud, infrared, depth, event stream and other modalities for HAR. These advances may be traced to the development of accurate and affordable sensors. Our work focuses primarily on using a skeleton modality [9] combined with Infrared (IR) videos to achieve HAR, based on the fact that skeleton data contains information about the joints of the human body in space (i.e., information about x, y, z coordinates) during a time series. Skeleton modality is efficient and compact for HAR in the case of a study which does not involve objects other than humans or scene context. From a biological perspective, we can recognize an action simply by observing the motion of the skeleton without the need for the appearance of the entire human body. Infrared technology is usually not used in HAR since RGB offers a better and richer representation of a given scene.

However, IR operates better than RGB in the absence of light and it is especially efficient in capturing motion in dark scenes, thus providing information when that obtained from the skeletons is insufficient. The main problem in HAR is to extract discriminant spatial features and to capture temporal dynamics in order to comprehend human actions [10]. To reach this objective, researchers have thus tended to create sophisticated and over-parameterized networks, leading to complicated training processes and high computational costs, resulting in low inference speeds. For example, DynamicGCN developed in [11] contains around 14 million parameters just to deal with skeleton data and requires several days of model training with GPU on the NTU RGB-D dataset [9]. In addition, fusing different modalities to help a model generalize while training [12] is yet another area calling for further exploration, because fusion allows us to select valuable candidates from each modality and combine the best features to maximize accuracy. Another limitation is explained in [13], which mentions that early methods like [14] for HAR simply utilize joint coordinates at individual time steps, providing more direct information about an individual's movement than raw images do. The feature vectors that are formed in the process are then subjected to temporal analysis. The limitations of these methods lie in the fact that they do not consider the spatial relationships between joints, which are essential to understanding HAR. In this paper, we develop a model that combines advantages from various state-of-the-art methods [10, 12, 13], to create a lightweight architecture (see Fig. 1) compared with the FUSION-CPA model from [12], while maintaining/improving accuracy. The first step in addressing these limitations is to employ graph-based models [15] that have been introduced to encode and model dynamic skeleton sequences, chosen for

Manuscript received March 13, 2023; revised April 18, 2023; accepted May 5, 2023.

doi: 10.18178/joig.11.4.309-320

their robust manner of representing structured data. The first of such models, the Yan *et al.* model [13] named Spatial-Temporal Graph Convolutional Networks (ST-GCN) for skeleton-based action recognition, laid the foundations for subsequent works including [16, 10] that followed. Graph Convolutional Networks (GCNs), considered a variant of Graph Neural Networks (GNNs), generalize the concept of Convolutional Neural Networks (CNNs) to the extent that early CNNs were generally deployed using regular data or Euclidean structured data. An important amount of data in real-world applications are, however, non-Euclidean graph-based structures. Indeed, the recent advancements in GNNs are based on this concept of non-regularity in data. Some examples are semi-supervised learning [15], image classification and traffic prediction. The second step in dealing with limitations is to create a multi-stream architecture with an intermediate fusion strategy [17] in order to capture rich and useful spatiotemporal features extracted from skeleton joint and IR video input.

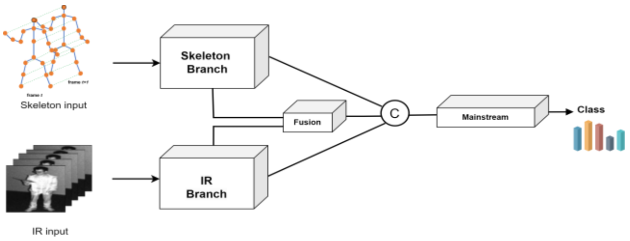

Figure 1. The main view of the proposed model with its different branches. C: for concatenation scheme.

The first branch, which is called the skeleton-branch as explained in [10], is composed of 3D joint positions as the first input, motion velocity as a second input and relative distance (distance between the spine joint and the rest of the joints) as the third input. The second branch, called the IR-branch as explained in [12], is solely composed of an IR sequence of images. Intermediate fusion, also known as feature-level fusion, uses convolutional blocks to transform raw data inputs into a higher-level dimension / representation by mapping them through a stack of layers. After merging the feature representation, we obtain multimodal feature maps used afterward for recognition purposes. The intermediate fusion is equipped with an attention mechanism [10] that identifies the most important joints from a skeleton sequence and helps the network extract features discriminately. The contributions of the study reported here are multiple:

- We changed the skeleton-branch backbone from a ResNet18 that deals generally with images as inputs to the ST-GCN model which has demonstrated its efficiency when dealing specifically with skeleton data. We also added sub-inputs to ST-GCN, the original ST-GCN including joint input only. In what follows, we explore three inputs (joints, velocity, relative distance to spine joint), see Fig. 2.
- The skeleton branch is equipped with Dynamic Representation (DR) from the SGN model.
- Feature fusion is a question of adding lateral connections to the architecture that have proved to be practical and effective in dealing with skeleton data merged with other modalities like IR. These connections allow the features to travel from one stream to another to share useful information while creating characteristic extracted features.
- The model was trained on a large-scale dataset, i.e., NTU RGB+D 60, and provided satisfactory results while retaining good accuracy with fewer trainable parameters compared to [12].

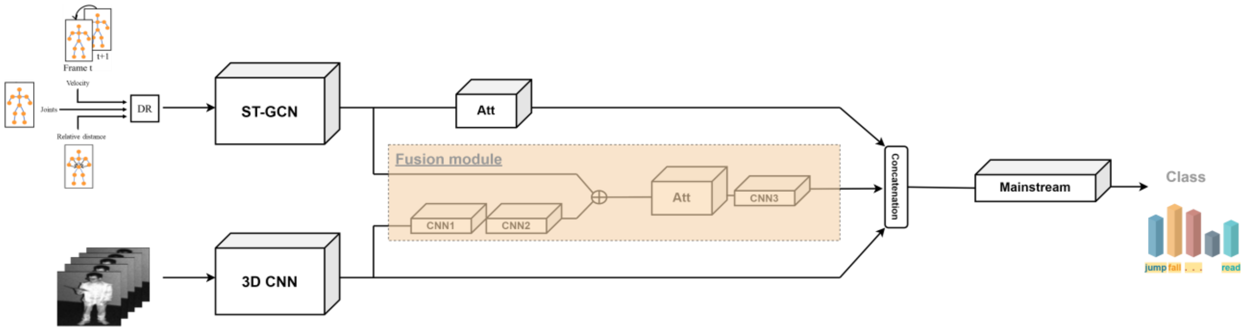

Figure 2. The framework of the proposed end-to-end SIRFusion model. It consists of a skeleton branch, an IR branch and a fusion module. In *DR*, we use embedding to obtain the Dynamic Representation of a joint and we merge information related to the position, velocity and relative distance. *Att* is the attention mechanism used to extract more discriminant features. *CNNs* are Convolutional Neural Networks based on FC convolutional layers.

II. Related Work

In early studies, RGB-based methods were wide-spread among researchers. RGB data usually contains rich appearance information when capturing scene context; the data is easy to collect and RGB sensors (cameras) are everywhere [18, 19]. However, attaining HAR from RGB data often proves challenging on account of variations in background, illumination conditions and viewpoints. An additional challenge involves large-scale data size, resulting in high computational costs when modeling fine-grained spatio-temporal relationships [18], a key concept for HAR. 2D, 3D-CNNs or RNNs-based methods depend heavily on the RGB modality to achieve HAR. They began with handcrafted feature-based approaches, such as the space-time volume-based methods [20, 21], dense-trajectory methods [22, 23] and Space-Time Interest Point (STIP)-based methods [24]. These methods were suited for

RGB video-based HAR. Afterward, Deep Learning techniques showed great progress and started to take center stage. These frameworks were divided into three categories, namely, 2D Convolutional Neural Networks (CNNs) including two-stream architectures [25], Recurrent Neural Networks (RNNs) [26] and finally 3D CNN-based architectures [27, 28].

Nevertheless, when the above-mentioned methods were applied under skeleton data as in earlier works, the CNN or RNN-based models tended to ignore spatial configurations as mentioned in [10]. Consequently, a new alternative was needed to model the spatial features and temporal dynamics of skeletons, leading to the introduction of graph-based methods, in particular Graph Convolutional Networks (GCNs).

Yan *et al.* [13] initially introduced a generic representation applied to recognize actions in skeleton sequences by exploiting graph neural networks to design a spatial-temporal graph-based model. This baseline, named ST-GCN, was a milestone for future works. After this work, Song *et al.* [29, 30] developed a subtle multi-stream GCN with ST-GCN as a baseline to test the effect of the occlusion problem in the HAR task. Li *et al.* [31] proposed an Actional-Structural GCN (AS-GCN) with an encoder-decoder structure to capture action-specific latent dependencies, combined with structural links to represent higher-order dependencies. Peng *et al.* [32] noted that graph structures in GCNs are pre-defined, so they proposed an automatically designed GCN with a Neural Architecture Search (GCN-NAS). Specifically, they explored more implicit correlations between joints, exploiting multiple dynamic sub-structures to build their search space. Shi *et al.* [33] developed a Two-stream Adaptive Graph Convolutional Network (2s-AGCN) in which the topology of the graph may be either uniformly or individually learned using the backpropagation algorithm. The model accepts two inputs, referred to as first-order (skeleton joints) and second-order information (lengths and directions of skeleton bones). Zhang *et al.* [16] added a semantic level to joints, whereby each time a frame is loaded, the joint type (e.g., head, hand, etc.) and the frame index are provided. This semantic level enables the network to enhance the relationship between joints, thus, enhancing the feature representation capability. MS-G3D presented in Liu *et al.* [34] features a multi-scale aggregation scheme to connect joints across space and time. The authors implemented a spatio-temporal operator called G3D to achieve feature extraction.

Despite the fact that such sophisticated methods provide considerable performance, the computational cost is an issue that needs to be resolved in the quest for real-time recognition and hardware implementation (our next step and the subject of upcoming research). In this way, combining the positive specifications of certain models to improve performance is an ongoing challenge.

III. Skeleton and Infrared Fusion

We have built a deep neural network combining skeleton and IR input data, named Skeleton and InfraRed Fusion (SIRFusion). The architecture consists of two branches in parallel with an intermediate fusion between them and an Multi-Layer Perceptron (MLP). The first branch deals with skeleton data, and the second branch interprets IR videos. After each stream fine-grains its data, the resulting features are fused with a concatenation scheme. We also add the extracted features from the fusion module to the concatenation. MLP comes into play after the fusion as the last stage stream, providing a probability density. The whole network undergoes an end-to-end training process to optimize the classification score.

We assume the input set is defined as $\{S = S_{j,t,k}\}$; the indexes are defined as follows: *j*: joint index, *t*: frame index and *k*: coordinate axis (3D coordinates: x, y and z). For an IR sequence set, we note $I = \{I_t\}$ where the index *t* varies between $\{1,...,T\}$, *T* being the maximum number of frames that a sequence can indulge. This number is experimental and $T = 20$ happens to be a compromise between accuracy and the volume of input.

A. Skeleton Branch

This stream is built using the ST-GCN model to exploit the concept of correlation between joints/nodes within the same frame. The model adopts Graph Convolutional Networks (GCNs) to search for correlations in skeleton data. There are two types of GCN-based methods, the first one based on predefined connections of the graph (edges) using a manually designed rule [13]. The second method is an adaptive graph, meaning it learns the topology of the graph adaptively [33]. In other words, the topology of the graph can be learned either informally or individually by the Back Propagation (BP) algorithm.

There are different ways to visualize a skeleton sequence. It can be represented by the 2D or 3D coordinates of the body's joints/nodes in subsequent frames. Prior work [35] combined all joints through their coordinate vectors and outputted a mono-feature vector per frame, this input in turn feeding a model based on temporal convolution equipped with residual connections. However, the ST-GCN model uses a spatial-temporal graph to construct structural and hierarchical forms representing more accurate skeleton data. In other words, the *V* (*V*: Nodes) intra-body joints that form a skeleton over *T* frames (duration) and conceptualize the inter-frame connection, will construct a spatial-temporal graph denoted $G = (V, E)$ {*V*: *joints/nodes*, *E*: *edges*}.

1) Normalization process: To better exploit skeleton data, we definitely need to subject the skeleton sequence to a normalization process that translates the camera coordinate system to the spine joint of the main subject, as per previous studies [10, 12, 13] (see Fig. 3). We calculate the translation vector used on the first frame and then we apply it to the rest of the frames, creating what we can call a sequence-wise normalization. The new local coordinate system is calculated as follows:

$$S' = S_{(:,:,:)} - S_{(1,0,:)} \tag{1}$$

S' denotes the normalized skeleton sequence. The first index $j = 1$ of *S* matches with the middle spine joint/node using the Kinect 2 topology [36], t=0 corresponds to the first frame and. means that all coordinates/dimensions (x, y and z) are taken into consideration.

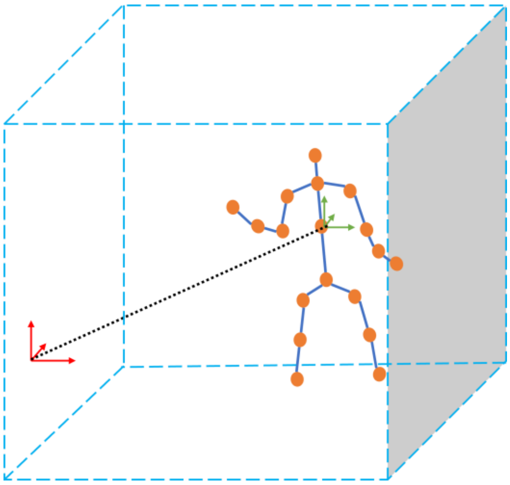

Figure 3. Schema that shows the normalization process applied to the skeleton data before training. The camera coordinate system, shown as the red coordinate system, is translated through the black line to the new coordinate system in green that refers to the location of the spine joint of the main subject in the normalization process. The main subject is represented as the skeleton in orange which is considered as the sequence's first frame.

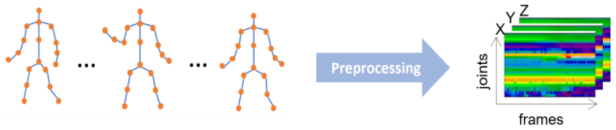

Figure 4. Transforming skeleton sequence to 2D maps (X, Y and Z maps/matrices). For each x, y and z coordinate of a skeleton 25 joints are collected in a column, and the column is considered the height of the maps/matrices. The row of a map is the different values of the same arrangement of joints over time i.e., tracking the joints over time.

2) Skeleton sequence to 2D maps/patch: In order to make skeleton data exploitable by a convolutional layer, we can map a skeleton sequence containing the different positions of the subject performing an action to an image [13] (see Fig. 4). The image has three channels corresponding to X, Y and Z matrices containing the coordinates of all joints during the entire action period. A column is a collection of all V joints $\{V = 25\}$ [9], and a row shows a joint/node coordinate throughout all the frames. Like in [12] a dataset normalization is applied; c_{min} and c_{max} are respectively the minimum and maximum values that a joint coordinate can take throughout the entire dataset. The new mapping is then calculated as follows:

$$\hat{S} = \frac{S' - c_{min}}{c_{max} - c_{min}} \tag{2}$$

The normalized skeleton map $\hat{S} = \hat{S}_{j,t,k}$ is in the range [0, 1] to fully exploit convolutional layers. Unlike Ref. [12], there is no input resizing to fit the network because this is no longer needed.

3) Multi-subject network: This model, specifically the skeleton branch, is flexible and accepts various subjects. Like Ref. [13], the input matrix $IN = [n, c, t, v, m = 2]$, {n: batch size. c: channels x, y and z. t: number of frames. v: number of joints. m: number of subjects} where m denotes the maximum number of subjects the model will process. A simple trick is adopted to obtain the right mapping by reshaping the input matrix to $IN = [n \times m, c, t, v \mid \times$: *multiplication*]. The number of subjects is limited to two, to comply with the NTU RGB+D dataset [9]. However, this model may accept multiple subjects if the need arises. If there is only one subject in the action, the rest of the slot (m = 2) will be filled with zeros to avoid bugs in calculations.

4) Graph Convolutional Network (GCN) implementation: Implementing a GCN (graph-based convolution) requires certain intermediate steps compared with 2D or 3D convolutions. The ST-GCN model acquires an equivalent implementation of GCNs as discussed in Kipf and Welling [15]. The connections of body joints in the same frame are represented by an adjacency matrix A alongside the identity matrix I corresponding to self-connections of the nodes/joints. The following formula in Eq. (3) was developed by Kipf and Welling [15] to define the propagation rule for information across the graph, which is a key component in conceptualizing GCNs. This rule updates the feature representations of the nodes in the graph. By applying this propagation rule iteratively, the GCN can learn representation of a given graph, in our case, the skeleton graph. In the final stage, this graph can be exploited for classification tasks.

$$f_{out} = \sigma(\widehat{D}^{-\frac{1}{2}}.\ \widehat{A}\ .\ \widehat{D}^{-\frac{1}{2}}.f_{in}\ .W) \tag{3}$$

where f_{in} and f_{out} are respectively input and output feature maps. $\widehat{A} = A + I$ is the adjacency matrix with self-connections of the undirected graph $\widehat{D} = \sum(\hat{A}_{ij})$. Here W is the weight matrix containing trainable parameters. σ is the *ReLU* activation function. In practice, the input feature map is represented via a tensor of dimensions *(C, T, V)* under the spatial-temporal case. We perform the graph convolution by using a standard 2D convolution $conv(f_{in}) = f_{in} \cdot W$ and the resulting tensors are multiplied to the normalized adjacency matrix $\widehat{D}^{-\frac{1}{2}}.\ \widehat{A}\ .\ \widehat{D}^{-\frac{1}{2}}$. When a subject performs an action, joints tend to move in groups, as one joint may link to various body parts. These links may be of varying importance in illustrating the dynamic aspect of these parts. Accordingly, ST-GCN added a learnable mask M applied throughout every layer inside the spatial-temporal graph convolutional block. This learnable edge importance weighting [13] enables a joint feature to scale its contribution to the neighboring joints. To implement this learnable mask, the adjacency matrix $\hat{A}$ is accompanied by a learnable parameter matrix M, $\hat{A} \odot M$, where the operation $\odot$ denotes the element-wise product. M initialization is a matrix of ones.

5) Attention mechanism: Attention mechanisms have gained a fair amount of popularity in sequence modeling in the accomplishment of numerous tasks. The concept of attention is when a model (or human perception in general) attends to and focuses on specific joints of the body to extract selective information about these joints during a specific frame. The attention module used in this work is inspired by Song *et al* [10]; the module is shown in Fig. 5.

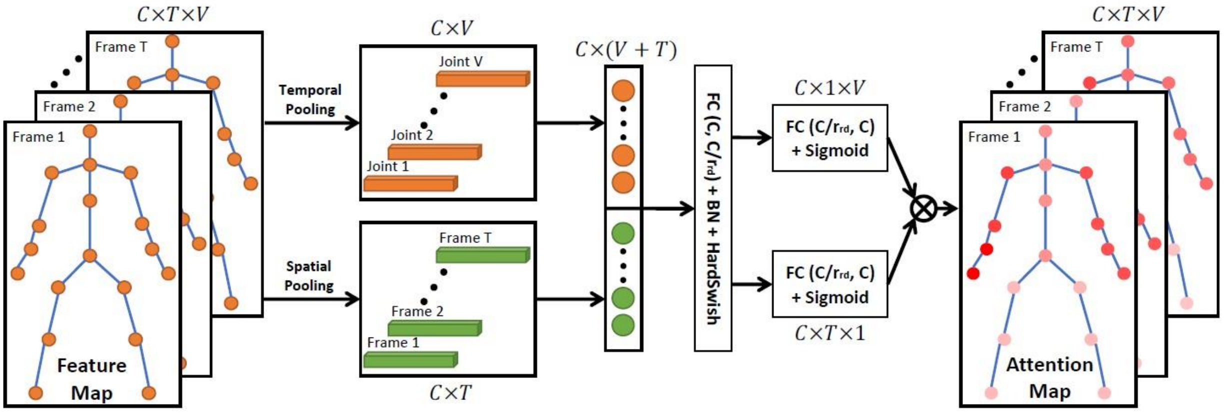

Figure 5. Workflow of ST-JointAtt (attention mechanism) taken from EfficientGCNv1 model [10]. The model extracts selective information about a skeleton movement by focusing on the most relevant joints describing an action. Here C, T and V are respectively input channels, 20 frames, and 25 skeleton joints, ⊗ denotes the outer-product, r_{rd} is the reduction coefficient, BN represents Batch Normalization, Sigmoid and Hardwish are activation functions. (Best viewed in color)

The input features are separated into two categories: averaged-joint (i.e., $pool_v(.)$) and averaged-frame (i.e., $pool_t(.)$). Indeed, intuitively the spatial and temporal information could be relevant to each other, unlike the previous attention modules that tend to be implemented by a Multi-layer Perceptron (MLP) fashion (or a bloc stacking convolutional layer, batch normalization layer and activation function). Next, the resulting pooled vectors are merged together with concatenation (stacking the features). This way, the features are prepared for the following process. After that, the features are fed to an FC layer (the parameter W in Eq. (4) is represented practically by a convolution layer) to obtain more compact information. Furthermore, as we need to calculate the attention scores at the joint-level and frame-level, we use two separate FC layer blocs, i.e., different network is assigned to each category (joint-level FC layer and frame-level FC layer) to allocate leaning weights (Matrices W_v and W_t) to each category distinctively. This way, the attention module will learn efficiently. Finally, we apply an outer-product multiplication to the joint and frame scores previously calculated in order to reconstruct the feature map f_{out} with the same dimensions as the input feature map f_{in}. The formula that governs this module is as follows:

$$\begin{aligned} f_{inner} &= \theta((pool_t(f_{in}) \oplus pool_v(f_{in})) \cdot W) \\ f_{out} &= f_{in} \odot (\sigma(f_{inner} \cdot W_t) \otimes \sigma(f_{inner} \cdot W_v)) \end{aligned} \tag{4}$$

Here, the feature maps are denoted f_{in} and f_{out} for input and output respectively, ⊕ is used for concatenation, ⊗ is the channel-wise outer-product, ⊙ denotes the element-wise product, $pool_v(.)$ is the average pooling applied at joint-level and $pool_t(.)$ is applied at frame-level, $\sigma(.)$ and $\theta(.)$ are the activation functions Sigmoid and Hardwish, and W are learnable weights/parameters.

B. IR Branch

When a subject performs an action, the space that the subject takes up inside a frame is relatively small in comparison to the whole video frame. To capture the region of interest in which the action happens, 2D skeleton data are used to locate this area, even in the case of a multi-subject scene. IR sequence input is processed by a 3D CNN.

1) Cropping and multi-subject localization: To achieve complex video understanding tasks, a 3D CNN typically contains a large number of training parameters. To help the model process more quickly, frame sequence is downscaled to minimize memory usage. We may lose some information during the process, but such losses are not alarming, since the background contribution is minimal to the context of actions and we privilege the model concentrating on the subject. A cropping strategy is adopted to help the network focus on the subject. First, we project 3D skeleton data onto the 2D frames. Then, minimal and maximal pixel coordinates are calculated from the totality of frames and joints. After that, using the pixel positions, a bounding box is used to extract the volume of the action (the subject through spatial-temporal dimensions). This method can be used for multi-subject detection. The bounding box will simply expand to wrap all subjects.

2) Sampling: A 3D CNN requires a fixed number of frames T in each IR sequence. In order to determine T, the first method is a simplistic approach that considers subsequent frames until reaching the number T. This method comes with drawbacks, since the first frames may not capture the totality of meaningful action. The second approach consists in dividing the IR sequence into equidistant segments or windows (e.g., 1 segment = 8 frames) and selecting a random frame from each segment. At the end, a sequence is formed of the size T.

C. Fusion Module

Intermediate fusion has proved its efficiency, as reported in many works [37]. It can effectively exploit the complex correlation between two different modalities. This fusion module is used as a link between the two branches (skeleton and IR). In this case, an intermediate fusion precedes the late fusion. It helps enrich the final fusion with more pertinent feature maps. The formula behind this bloc is as follows:

$$
\begin{gathered}
f_{inner} = f_{skl} + \sigma(\sigma(f_{ir} \cdot W_1 + b_1) \cdot W_2 + b_2)) \\
f_{out} = Att(f_{inner}) \cdot W_3 + b_3
\end{gathered} \tag{5}
$$

where f_{ir} and f_{skl} are feature maps from the IR branch and the skeleton branch (ST-GCN network) respectively, $\sigma(.)$ is the ReLU activation function, $Att(.)$ is the attention function as explained in Section III.A5, $W_{1,2,3}$ are trainable parameters and $b_{1,2,3}$ are biases.

To understand the flow of this module. In the first row of Eq. (5), the feature map coming from the IR branch needs to be upsampled and processed to match the feature map of the skeleton branch, this is where CNN$_1$ and CNN$_2$ introduce the learning parameters W_1, b_1 and W_2, b_2. Once the features have matching dimensions, they are combined. In the second row of the same equation, the attention module $Att(.)$ is used at this stage to extract informative features. CNN$_3$ which introduces W_3 and b_3 is applied at the end of the module to cast the feature map to the proper dimensions. The ReLU activation function is used after each Convolution to introduce the nonlinearity which helps the module to learn efficiently.

D. Mainstream

The previous Sections III.A and III.B have explained the feature extraction step to obtain a feature vector from a skeleton branch and another feature vector with the same dimensions from the IR branch. An MLP is a block of successive layers (batch normalization layer, convolution layer, activation function, etc.) implemented after the fusion scheme (e.g., concatenation) to train the rest of the network on mix modalities and generally finishes with a Softmax activation function to calculate the probability distribution related to each class (action) in a dataset.

IV. Experiments

In this section, we evaluate the performance of our model using the NTU RGB+D large-scale dataset [9]. The final results are reported in Table I. Most displayed models are CNN-based and LSTM-based models, the last ones are GCN-based networks.

A. Dataset

We used the NTU RGB+D 60 indoor large-scale dataset shared by Shahroudy *et al.* [9]. The authors used Microsoft Kinect v2 sensors, which offer four data modalities (RGB, depth maps, IR and skeleton), so we selected the IR and skeleton modalities for the present project. The dataset contains 56,880 sequences. The actions are performed by 40 different subjects providing 60 action classes and the classes are divided into health-related actions, daily and mutual actions. The human skeleton is represented by 25 joints, each joint localized by its 3D coordinates x, y and z. The IR sequences are collected with a size of 512×424; this size is reshaped/cropped in a preprocessing stage before training. The dataset is split into 2 benchmarks:

a) Cross-Subject (CS): Splitting 40 subjects into 2 sets containing 40,320 sequences for training and 16,560 for evaluation.

b) Cross-View (CV): the data collected from cameras 2 and 3 are regarded as the training set (37,920 sequences); the remainder of the acquisitions with camera 1 are included in the evaluation set (18,960 sequences).

TABLE I. COMPARISON BETWEEN OUR WORK AND STATE-OF-THE-ART (SOTA) MODELS ON THE NTU 60 DATASET, ACCURACY IN (%)

Method	Skeleton	RGB	Depth	IR	CS (%)	CV (%)
Lie Group [38]	X	-	-	-	50.1	82.8
HBRNN [39]	X	-	-	-	59.1	64
Deep LSTM [9]	X	-	-	-	60.7	67.3
PA-LSTM [9]	X	-	-	-	62.9	70.3
ST-LSTM [40]	X	-	-	-	69.2	77.7
STA-LSTM [41]	X	-	-	-	73.4	81.2
VA-LSTM [42]	X	-	-	-	79.2	78.8
TCN [35]	X	-	-	-	74.3	83.1
C+CNN+MTLN [43]	X	-	-	-	79.6	84.8
Synthesized CNN [44]	X	-	-	-	80	87.2
3scale ResNet [45]	X	-	-	-	85	92.3
DSSCA-SSLM [46]	-	X	X	-	74.9	-
[47]	X	-	X	-	75.2	83.1
CMSN [48]	X	X	-	-	80.8	-
STA-HANDS [49]	X	X	-	-	84.8	90.6
Coop CNN [50]	-	X	X	-	86.4	89
ST-GCN [13]	X	-	-	-	81.5	88.3
RA-GCNv2 [30]	X	-	-	-	87.3	93.6
2s-AGCN [51]	X	-	-	-	88.5	**95.1**
our proposed methods: **Skeleton network**	X	-	-	-	82.3	89.5
IR network	-	-	-	X	89.8	**93.8**
SIRFusion	X	-	-	X	**91.1**	93.7

B. Implementation Details

1) Network settings: The Dynamic Representation (DR) or embedding is set to 16 layers, keeping in mind that the weights of the layers are not shared for velocity, position or relative distance. The skeleton branch input size is set to {N: batch size, C: 3 input channels, T: 20 frames, V: 25 joints, M: 2 subjects}, input channels are C=3 for x, y and z, the input of ST-GCN blocs is 16 layers according to the encoding of the DR of position, velocity and relative distance. The output of the ST-GCN bloc is 512, the adjacency matrix A is 25×25. The attention mechanism input and output are set to 256 layers with a reduction ratio of 2. The IR branch requires a fixed input size of 112×112. CNN$_1$, CNN$_2$ and CNN$_3$ are used to reshape the feature maps from 512 layers to the same size 512 layers through intermediate sizes 500, 1, and 256 layers respectively. Batch normalization, dropout, Softmax, and ReLU nonlinear activation functions are used.

2) Training settings: All training was conducted on the Pytorch framework with one GPU Tesla V100s Card from the University Cluster (CCUB). We used the Adam optimizer and we set the initial learning rate to 0.0001 with a consistent value during training. The batch size was set to 16 in order to fit in most GPUs. To avoid gradient exploding issues, gradient clipping was applied. Training was finished after 15 epochs. The loss function used to train the model for classification was cross entropy.

3) Results: We studied the branches individually (skeleton network and IR network) and then we compared them to the SIRFusion fused model. The accuracy metric was used to compare the SOTA models on NTU 60 dataset.

The confusion matrix (see Fig. A1) demonstrates the efficiency of the skeleton network classifying action (only the skeleton branch was trained) with dense kinetic movements/motion such as falling, walking towards or away from other subjects, jumping, etc., showing an accuracy of over 96%. On the other hand, actions with less kinetic movement such as reading or touching the head are more challenging to classify, showing less than 53% accuracy due to the high similarity of frames. The issues of frame similarity and object-related actions in a scene are two limitations that affect skeleton data in general. We suspect this problem cannot be solved without the intervention of another modality (IR, RGB, etc.). The experiments also revealed a major contribution of the IR branch. When predicting the action of writing for example (a tricky action with low kinetic movement) using the IR-based network (only the IR branch is trained while the skeleton branch is discarded), the output accuracy was 88% (see Fig. A2), much better than a prediction from the skeleton-based network (only the skeleton branch is trained while the IR branch is discarded) with only 43% accuracy. We found that for certain actions, such as playing with a phone or making/answering a phone call, the skeleton network proved more accurate with a difference of 13% and 3% respectively. This finding reinforces the idea that position information and visual information contribute beneficially and mutually to each other (see Fig. A3). Nonetheless, in exceptional cases associated with object-related actions such as playing with a phone/tablet, the model tended to mismatch the action with writing. There are two possible causes for this issue. First, it is possible that object information was lost during the resizing stage. Second, as IR data is a grayscale image, additive noise might confuse the model prediction.

The performances of SIRFusion shown in Table I almost outperform the mentioned SOTA methods. In the mono-modality methods, three typical methods should be pointed out. The first model is ST-GCN [13], the most renowned baseline for skeleton-based human action recognition. SIRFusion leads over 9.6% on CS benchmark and 5.4% on CV benchmark. The second method 2s-AGCN [51] is another famous backbone for skeleton-based action recognition. SIRFusion baseline outperforms 2s-AGCN in CS benchmark with a lead of 2.6%, even thought, it is slightly less efficient in CV benchmark with a deference score of −1.4%. The third method is STA-LSTM [41] which is also known for being strengthened/enhanced by an attention module, in comparison to STA-LSTM, our model is outperforming with a difference of 17.7% and 12.5% in respectively CS and CV benchmarks. One of the reasons might be in the used attention module, the authors applied the attention module individually for frames and joints. Contrary with the attention module implemented in our model SIRFusion, it works cooperatively on frames and joints as explained in Section III.A5. For multi-modal methods mentioned in Table I, SIRFusion seems to provide fair performances compared with them and proves that the IR modality is also applicable for action recognition. In terms of compromise, FUSION-CPA [12] slightly outperforms SIRFusion in accuracy. Nevertheless, FUSION-CPA [12] has nine million training parameters more than SIRFusion, which implies an increase of model complexity and computational costs.

V. Conclusion

The model developed here is an end-to-end trainable network exploiting 3D skeleton data alongside infrared videos to achieve Human Action Recognition (HAR). The model consists of two branches/streams (the skeleton branch and the IR branch). The skeleton branch is used to extract discriminant features with the GCN-based method, spatial-temporal convolutions and an attention mechanism. The IR branch deals with videos or cropped videos using a 3D CNN. The two branches are fused in several stages to synchronize the branches' feature maps for more accurate results and a classification prediction is returned at the end of the model. Each branch considered individually provides reasonable performance, but when merged together the results are greatly improved. Compared with the model introduced in the original paper (FUSION-CPA), the present model is lighter by 9 million training parameters thanks to the ST-GCN model. Our model illustrates the potential of infrared data, particularly in applications where the RGB modality may not operate due to illumination conditions (night scenes). Given the design of this network, changes and improvements are easy to implement. In future work, we will focus on further reducing training parameters by changing the 3D CNN, and on identifying more robust ways of fusing features.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Author Contributions

T. Bakir and S. Femmam contributed to foundational concepts, while A. Mansouri handled the implementation and experimentation. The analysis phase involved collective efforts from all authors. All authors had approved the final version.

Appendix

In this appendix reside three figures representing the confusion matrices to showcase the results of this paper model.

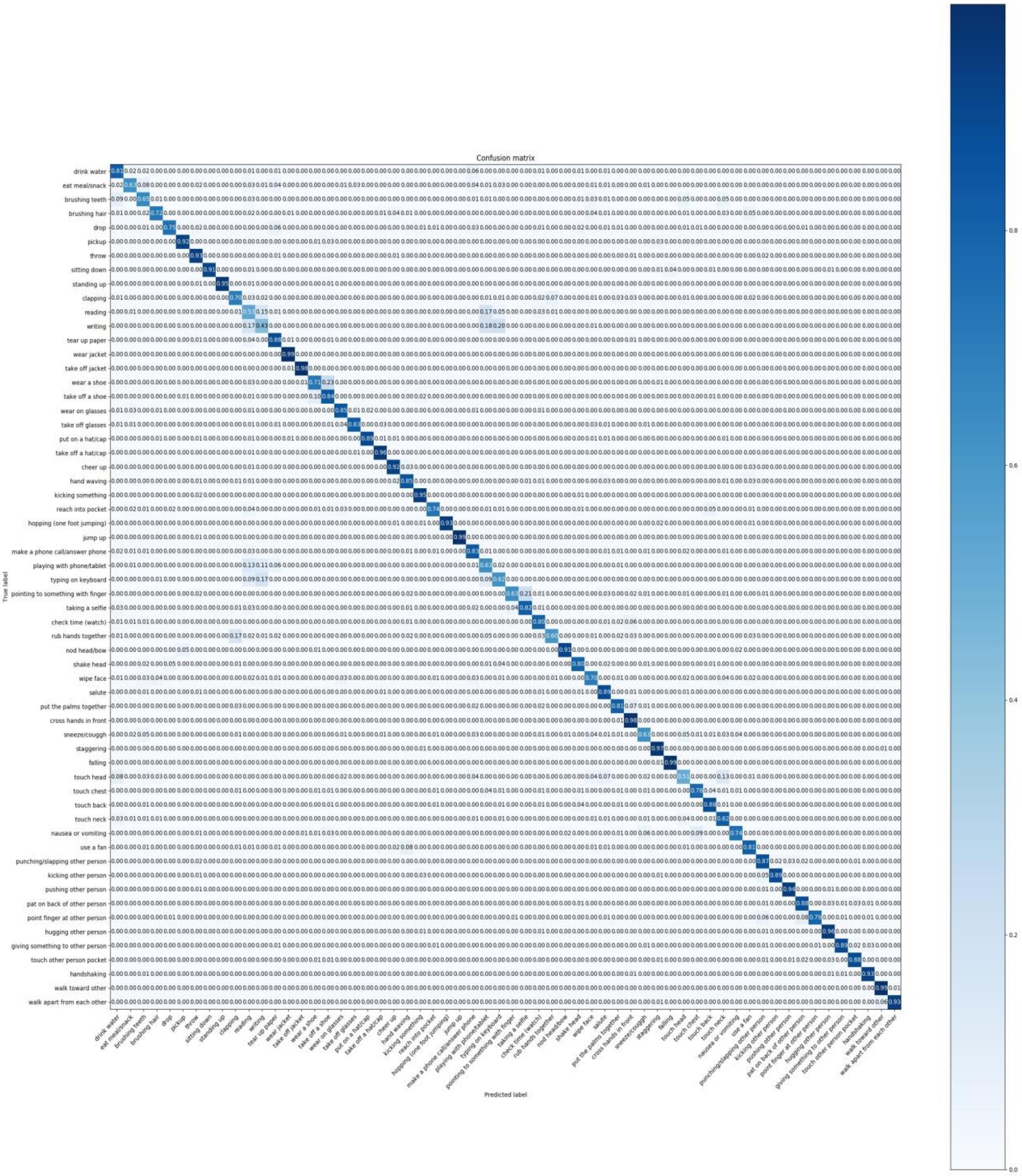

Figure A1. Confusion Matrix for the Skeleton-branch.

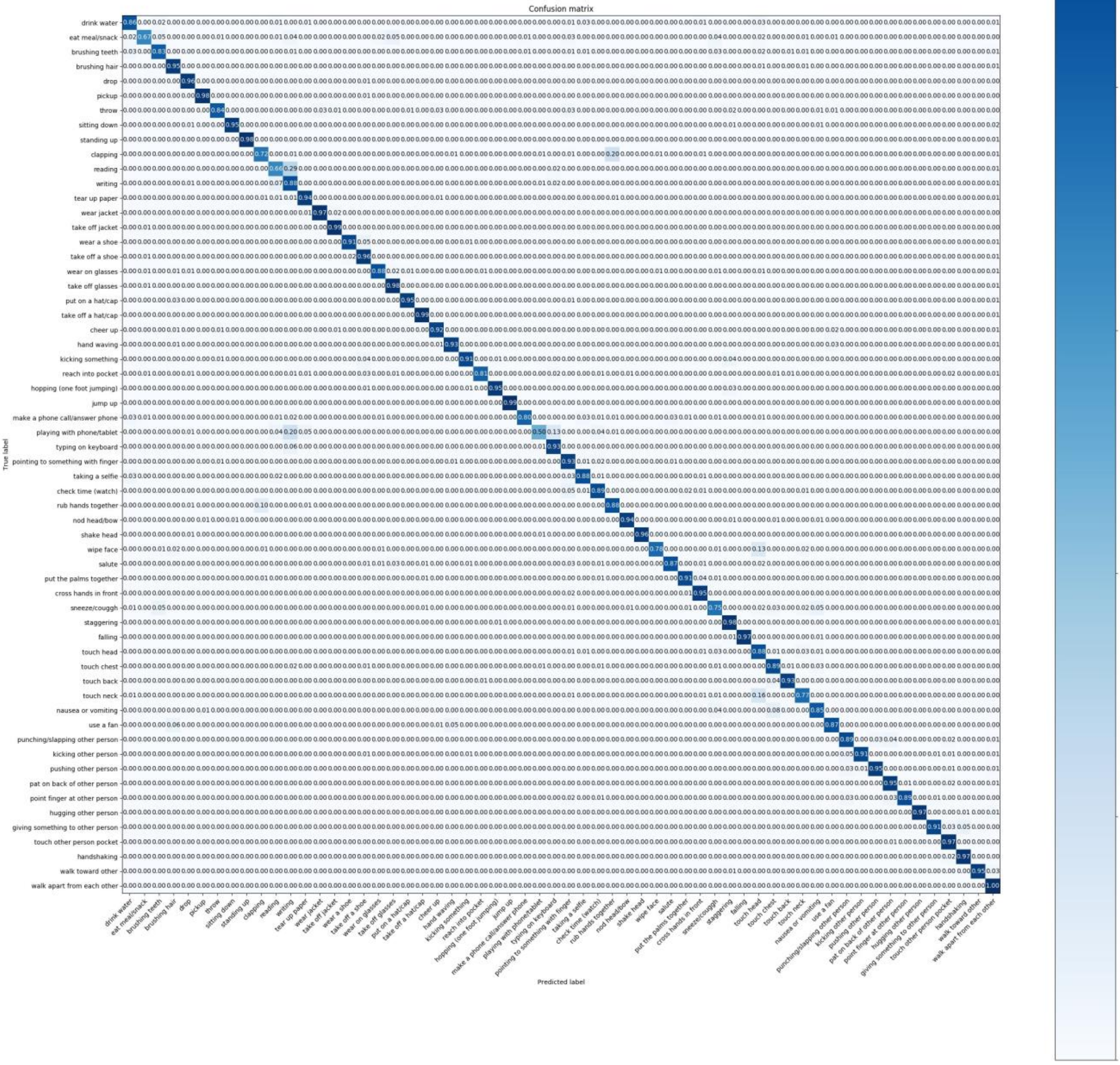

Figure A2. Confusion Matrix for the IR-branch.

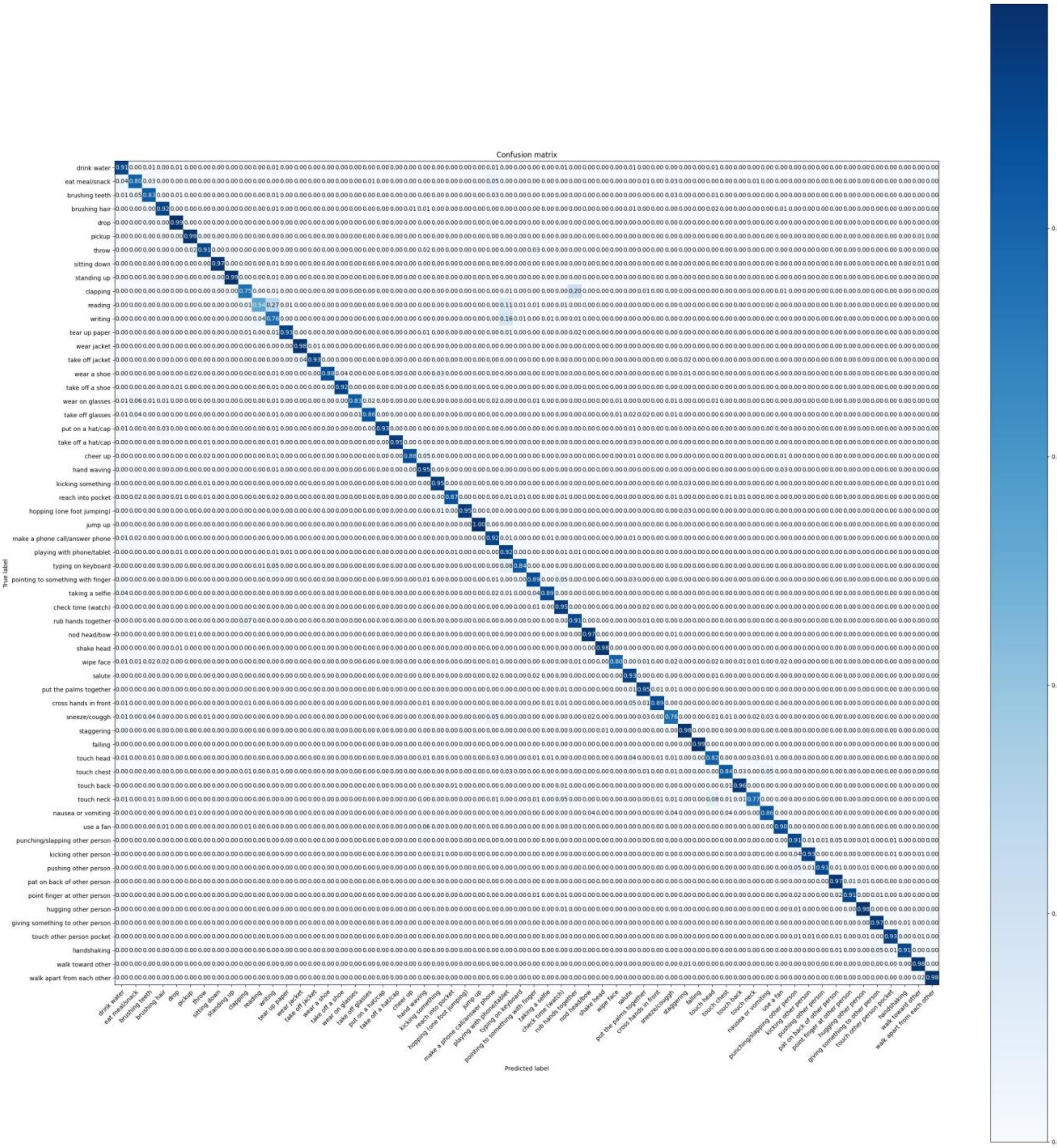

Figure A3. Confusion Matrix for the SIRFusion model.

References

[1] G. Cheng, Y. Wan, A. N. Saudagar, K. Namuduri, and B. P. Buckles, “Advances in human action recognition: A survey,” arXiv preprint, arXiv:1501.05964, 2015.

[2] M. Lu, Y. Hu, and X. Lu, “Driver action recognition using deformable and dilated faster R-CNN with optimized region proposals,” *Applied Intelligence*, vol. 50, no. 4, pp. 1100–1111, 2020.

[3] O. Liouane, S. Femmam, T. Bakir, and A. B. Abdelali, “Novel DVHOP algorithm-based machines learning technics for node localization in rang-free wireless sensor networks,” *International Journal of Informatics and Communication Technology (IJ-ICT)*, 2023.

[4] O. Liouane, S. Femmam, T. Bakir, and A. B. Abdelali, “Improved two hidden layers extreme learning machines for node localization in range free wireless sensor networks,” *J. Commun.*, vol. 16, no. 12, pp. 528–534, 2021.

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

[6] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, “Deep residual BIDIR-LSTM for human activity recognition using wearable sensors,” *Mathematical Problems in Engineering*, vol. 2018, 2018.

[7] B. R. Pradhan, Y. Bethi, S. Narayanan, A. Chakraborty, and C. S. Thakur, “N-har: A neuromorphic event-based human activity recognition system using memory surfaces,” in *Proc. 2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2019, pp. 1–5.

[8] A. D. Singh, S. S. Sandha, L. Garcia, and M. Srivastava, “Radhar: Human activity recognition from point clouds generated through a

millimeter-wave radar," in *Proc. the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*, 2019, pp. 51–56.

[9] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu RGB+ D: A large scale dataset for 3d human activity analysis," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1010–1019.

[10] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Constructing stronger and faster baselines for skeleton-based action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[11] F. Ye, S. Pu, Q. Zhong, C. Li, D. Xie, and H. Tang, "Dynamic GCN: Context-enriched topology learning for skeleton-based action recognition," in *Proc. the 28th ACM International Conference on Multimedia*, 2020, pp. 55–63.

[12] A. M. De Boissiere and R. Noumeir, "Infrared and 3D skeleton feature fusion for RGB-D action recognition," *IEEE Access*, vol. 8, pp. 168297–168308, 2020.

[13] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5378–5387.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint, arXiv:1609.02907, 2016.

[16] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1112–1121.

[17] S. Y. Boulahia, A. Amamra, M. R. Madi, and S. Daikh, "Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition," *Machine Vision and Applications*, vol. 32, no. 6, pp. 1–18, 2021.

[18] 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*, 2022.

[19] M. Hassan, T. Ahmad, N. Liaqat, A. Farooq, S. A. Ali, and S. R. Hassan, "A review on human actions recognition using vision based techniques," *Journal of Image and Graphics*, vol. 2, no. 1, pp. 28–32, 2014.

[20] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[21] T. Ahmad, J. Rafique, H. Muazzam, and T. Rizvi, "Using discrete cosine transform based features for human action recognition," *Journal of Image and Graphics*, vol. 3, no. 2, pp. 96–101, 2015.

[22] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[23] N. Kumar and N. Sukavanam, "Motion trajectory for human action recognition using Fourier temporal features of skeleton joints," *Journal of Image and Graphics*, vol. 6, no. 2, pp. 174–180, 2018.

[24] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107–123, 2005.

[25] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[26] J.-Y. He, X. Wu, Z.-Q. Cheng, Z. Yuan, and Y.-G. Jiang, "DB-LSTM: Densely-connected bi-directional LSTM for human action recognition," *Neurocomputing*, vol. 444, pp. 319–331, 2021.

[27] M. Kalfaoglu, S. Kalkan, and A. A. Alatan, "Late temporal modeling in 3d CNN architectures with BERT for action recognition," in *Proc. European Conference on Computer Vision*, Springer, pp. 731–747, 2020.

[28] S. Yucer and Y. S. Akgul, "3d human action recognition with Siamese-LSTM based deep metric learning," arXiv preprint, arXiv:1807.02131, 2018.

[29] Y.-F. Song, Z. Zhang, and L. Wang, "Richly activated graph convolutional network for action recognition with incomplete skeletons," in *Proc. 2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 1–5.

[30] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Richly activated graph convolutional network for robust skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1915–1925, 2020.

[31] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3595–3603.

[32] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," in *Proc. the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 2669–2676.

[33] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12026–12035.

[34] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 143–152.

[35] T. Soo Kim and A. Reiter, "Interpretable 3d human action analysis with temporal convolutional networks," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 20–28.

[36] Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[37] T. Zhou, H. Fu, G. Chen, Y. Zhou, D.-P. Fan, and L. Shao, "Specificity-preserving RGB-D saliency detection," in *Proc. the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4681–4691.

[38] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 588–595.

[39] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1110–1118.

[40] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "SPATIO-temporal LSTM with trust gates for 3d human action recognition," in *Proc. European Conference on Computer Vision*, Springer, 2016, pp. 816–833.

[41] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatiotemporal attention model for human action recognition from skeleton data," in *Proc. the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[42] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," in *Proc. the IEEE International Conference on Computer Vision*, 2017, pp. 2117–2126.

[43] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3288–3297.

[44] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, 2017.

[45] B. Li, Y. Dai, X. Cheng, H. Chen, Y. Lin, and M. He, "Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN," in *Proc. 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2017, pp. 601–604.

[46] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang, "Deep multimodal feature analysis for action recognition in RGB+ D videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1045–1058, 2017.

[47] H. Rahmani and M. Bennamoun, "Learning action recognition model from depth and skeleton videos," in *Proc. the IEEE International Conference on Computer Vision*, 2017, pp. 5832–5841.

[48] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox, "Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection (supplementary material)," in *Proc. 2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017.

[49] F. Baradel, C. Wolf, and J. Mille, "Pose-conditioned SPATIO-temporal attention for human action recognition," arXiv preprint, arXiv:1703.10106, 2017.

[50] P. Wang, W. Li, J. Wan, P. Ogunbona, and X. Liu, "Cooperative training of deep aggregation networks for RGB-D action recognition," in *Proc. the AAAI Conference on Artificial Intelligence*, 2018, vol. 32.

[51] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12026–12035.