Department of Computer Science
University POLITEHNICA of Bucharest
Faculty of Automatic Control and Computers

Human Action Recognition for Social Robots

PHD THESIS – Summary

Author:
Mihai Nan

Scientific Adviser:
Prof. Adina-Magda Florea

Ph.D. Thesis Committee

President	Prof.Dr.Eng. Florin Pop	University POLITEHNICA of Bucharest
Thesis Advisor	Prof.Dr.Eng. Adina Magda Florea	University POLITEHNICA of Bucharest
Examiner	Prof.Dr.Eng. Sergiu Nedevschi	Technical University of Cluj-Napoca
Examiner	Prof.Dr.Eng. Vasile Manta	Technical University of Iași
Examiner	Prof.Dr.Eng. Irina Georgiana Mocanu	University POLITEHNICA of Bucharest

University POLITEHNICA of Bucharest
Bucharest, 2022

Author:
Mihai Nan

Scientific Adviser::
Prof. Adina-Magda Florea

Institute:
University POLITEHNICA of Bucharest, Romania

ABSTRACT

The ability to recognize the actions performed by people is fundamental for many practical applications in various domains. Many of these applications need real-time recognition, and this aspect imposes special constraints on the proposed solutions. We considered that in the current context social robots represent one of the fields in which the Human Action Recognition (HAR) problem demonstrates its usefulness. A robot cannot be considered social if it is not pro-active, and to fulfil this property it must be able to understand the actions performed by the humans with whom it interacts. Starting from this motivation, researchers have already proposed many solutions for the HAR problem, but because the problem is a complex one that depends on many factors, this direction of research remains open and represents a hot topic in Computer Vision.

Our goal is to specify what a human action represents in our conception and to propose solutions for this type of problem. Moreover, we want the proposed solutions to achieve a trade-off between good accuracy and high inference speed. We analyzed the available data representation modalities and concluded that the most suitable is one based on the skeleton detected from the video sequence because it allows us to achieve the proposed goal and ensures user privacy. To achieve our goal, we designed a series of approaches based on small-sized deep neural networks which obtain performances comparable to the rest of the existing solutions for one of the most complex benchmarks, NTU RGB+D.

To validate the possibility of using the proposed solutions for a robotic platform, we developed a general pipeline that we integrated into the AMIRO framework and tested using the Pepper humanoid robot. With the help of transformations introduced in the integration pipeline, we showed that our neural model, trained using a dataset collected with fixed cameras, can recognize human actions in a real scenario starting from data provided by the robot's cameras.

We proposed and implemented multiple solutions for the HAR problem using original neural architectures. We built these models considering the existing state-of-the-art techniques in the specialized literature. The main requirements we had in mind when we developed these approaches are high inference speed and good generalization capacity.

There are situations in which the correctness of the prediction offered by the neural model is fundamental. This aspect applies especially to applications in the medical field. For this reason, in this thesis we also addressed the problem of explainability for one of the proposed approaches. The proposed neural model returns, in addition to the probabilities for each class, a tensor with features based on which we can determine the most important joints for each frame.

We performed an extensive analysis of the results for the best proposed approaches. This analysis highlights the classes for which the neural models achieve the best performance, but also those for which they fail to correctly identify the actions. Following this analysis, we can conclude that neural models have problems in correctly identifying those actions for which even humans fail when they rely only on skeletal data.

CONTENTS

1 INTRODUCTION

The HAR problem represents the task in which we analyzed a temporal sequence describing an action performed by a human to label it with one of the possible actions. This problem is a hot research topic in the field of Computer Vision, because it has great practical applicability. There are two relevant categories of practical problems that require solutions that integrate a HAR module: human-robot interaction problems and video monitoring problems. Each category presents a series of particularities, and, in this thesis, we will focus on some approaches that fulfil the necessary conditions for application to human-robot interaction problems.

1.1 MOTIVATION

The inspiration for the problem of recognizing human actions comes from the human perception of the visual information that the eye transmits to the brain. We cannot identify a standard algorithm that the human brain applies to analyze the video stream to determine the relevant information. Therefore, even the applications that could solve by using a HAR module cannot depend on rules and strict definitions of the gestures, the particularities of the people acting, the environment, etc. Moreover, the action recognition is sometimes a difficult task even for humans. The difficulty comes from the fact that many aspects must be considered when analyzing a sequence that describes an action. Each person may have a distinctive way of acting. Thus, the same action can differ from one person to another or from one context to another.

1.2 OBJECTIVES

The main objective of this thesis is to design a module capable of solving the problem of HAR that we can easily integrate into frameworks used to solve practical problems. To achieve this goal, our research has identified the challenges that exist and has been guided by some key questions that arise when we propose to evaluate such a solution:

- *What are the main real-world applications that require the usage of a module capable of classifying human actions?*
- *What data sets exist and how do they relate to the approaches proposed up to this point?*

- *What is the appropriate representation to ensure invariance to the environment and the person acting?*
- *What are the most suitable types of neural layers that we can use in the design of the neural network that performs the classification?*
- *How can we integrate a HAR module into a robotic platform and what performance does it achieve in a real-time scenario?*
- *How can we calculate additional features starting from the coordinates of the joints in 3D space and using mathematical formulas?*
- *How can a neural model explain or motivate the prediction provided?*
- *What augmentation operations can we use for skeletal data and how do they influence the performance achieved by neural models?*

We addressed each of these questions in the chapters of this thesis and provided answers for them motivated by the results obtained or the contributions introduced.

2

SPATIO-TEMPORAL HUMAN ACTION RECOGNITION WITH RNN AND TCN

2.1 DATA PROCESSING

An $\mathbf{X}$ vector is read from the dataset for each sample, where $\mathbf{X} \in \mathbb{R}^{C \times T \times V \times M}$ ($C = 3$—number of coordinates, T—number of frames, $V = 25$—number of joints, $M \in \{1, 2\}$—number of people). For the joint-branch, for each joint, 3 values are added, determined based on the difference between the coordinates of the joint j_i and those of the joint considered center of gravity j_c:

$$joint\_features_{j_i} = (x_{j_i}, y_{j_i}, z_{j_i}, x_{j_i} - x_{j_c}, y_{j_i} - y_{j_c}, z_{j_i} - z_{j_c})$$

(the center of gravity was considered the joint with the index 1 – *base of the spine*). For the velocity-branch, the differences between the coordinates of the joint at frame $t + 2$ and those at frame t were determined, as well as the differences between the coordinates of the joint at frame $t + 1$ and those at frame t:

$$velocity\_features^t_{j_i} = (x^{t+2}_{j_i} - x^t_{j_i}, y^{t+2}_{j_i} - y^t_{j_i}, z^{t+2}_{j_i} - z^t_{j_i}, x^{t+1}_{j_i} - x^t_{j_i}, y^{t+1}_{j_i} - y^t_{j_i}, z^{t+1}_{j_i} - z^t_{j_i})$$

For the bone-branch, we also have 6 features that include the 3 lengths and the 3 values of the angles for the X, Y, Z axes:

$$bone\_features_{(j_u, j_v)} = (x_{j_u} - x_{j_v}, y_{j_u} - y_{j_v}, z_{j_u} - z_{j_v}, a_{(j_u,j_v),x}, a_{(j_u,j_v),y}, a_{(j_u,j_v),z})$$

where joints j_u and j_v are adjacent, $l_{(j_u,j_v),x} = x_{j_u} - x_{j_v}, l_{(j_u,j_v),y} = y_{j_u} - y_{j_v}, l_{(j_u,j_v),z} = z_{j_u} - z_{j_v}$ and

$$a_{(j_u,j_v),x} = \arccos\left(\frac{l_{(j_u,j_v),x}}{\sqrt{l^2_{(j_u,j_v),x} + l^2_{(j_u,j_v),y} + l^2_{(j_u,j_v),z}}}\right)$$

$$a_{(j_u,j_v),y} = \arccos\left(\frac{l_{(j_u,j_v),y}}{\sqrt{l^2_{(j_u,j_v),x} + l^2_{(j_u,j_v),y} + l^2_{(j_u,j_v),z}}}\right)$$

$$a_{(j_u,j_v),z} = \arccos\left(\frac{l_{(j_u,j_v),z}}{\sqrt{l^2_{(j_u,j_v),x} + l^2_{(j_u,j_v),y} + l^2_{(j_u,j_v),z}}}\right)$$

.

2.2 METHODS FOR REARRANGING JOINTS

To extract spatial dependencies, we proposed two variants of reorganizing the joints: one 2D (shown in Figure 2.1) and one 1D (shown in Figure 2.2).

The 2D variant was proposed earlier in our paper [1] and is based on a 5 × 5 matrix. The 2D variant presented in the Figure 2.1 allows the application of a Temporal Convolutional Network (TCN) type layer based on 3D convolutions. This variant of representation considers the 5 essential parts of the body—left hand, torso, right hand, left foot and right foot.

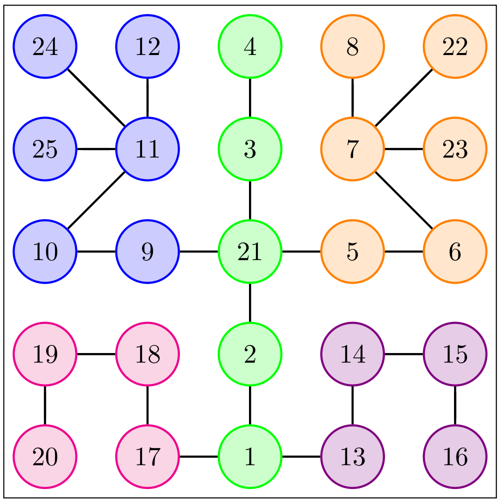
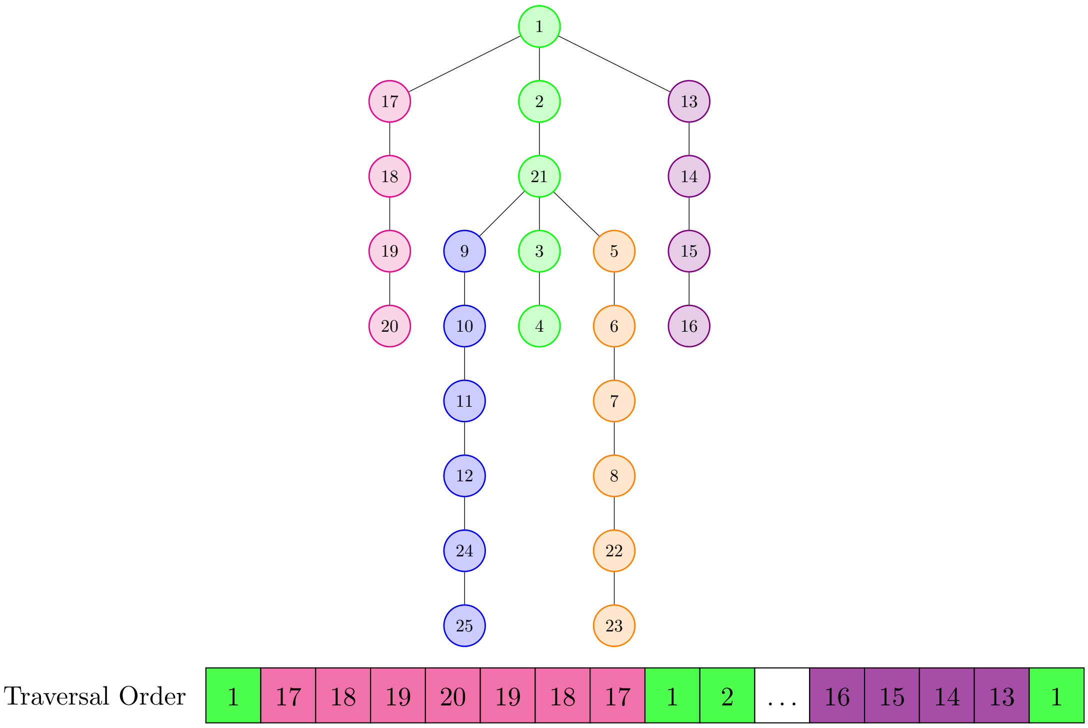


Figure 2.1: A proposal to rearrange the joints in a 2D format

The second proposed reorganization is a linear one and is inspired by Yang et al. [2]. We chose as the root for the proposed tree the central joint (the one with index 1), considering that it has a special importance, reason for which it was also used for the normalization step. The proposed tree is shown in Figure 2.2 and contains all 25 joints. We also considered the order in which the sub-trees for the root were added. Starting from this tree, we made a linear rearrangement of the joints starting from the DFS (Depth-first search) traversal of the tree. In this way, we made sure that any two nodes that appear side by side in the arrangement are also adjacent in the skeleton graph.



1	17	18	19	20	19	18	17	1	2	...	16	15	14	13	1

Figure 2.2: Transformation of the skeleton into a tree having the root of joint 1 (considered the centre of gravity). This tree is used for linearizing the skeleton (made based on a depth traversal applied to the tree).

2.3 TCN-BASED ARCHITECTURES

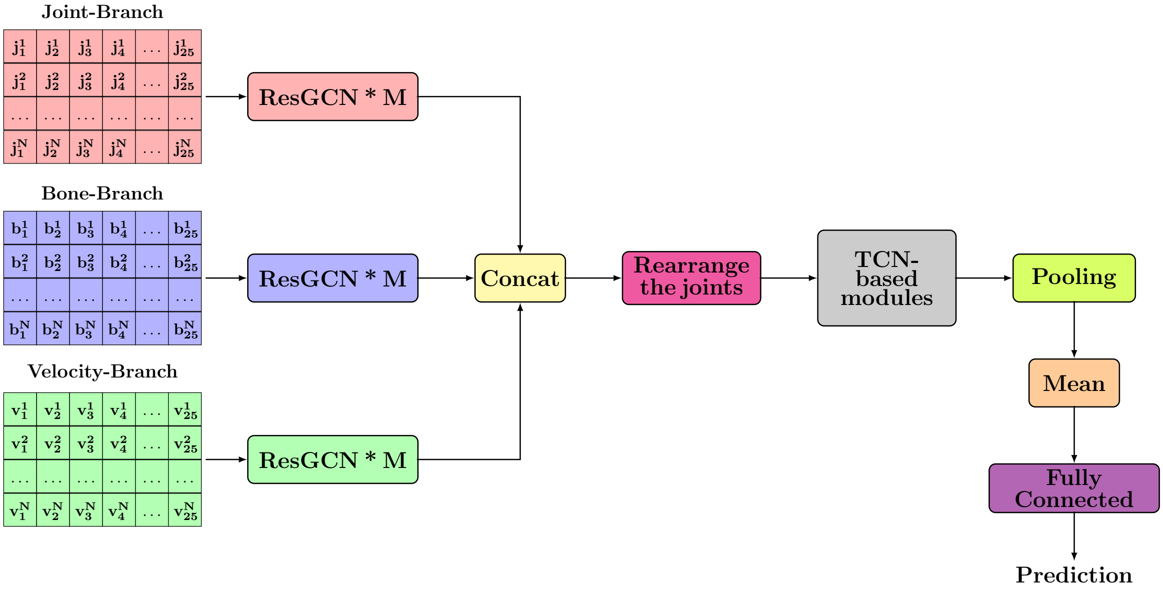

Joint-Branch
j_1^1 j_2^1 j_3^1 j_4^1 ... j_{25}^1
j_1^2 j_2^2 j_3^2 j_4^2 ... j_{25}^2
... ...
j_1^N j_2^N j_3^N j_4^N ... j_{25}^N
Bone-Branch
b_1^1 b_2^1 b_3^1 b_4^1 ... b_{25}^1
b_1^2 b_2^2 b_3^2 b_4^2 ... b_{25}^2
... ...
b_1^N b_2^N b_3^N b_4^N ... b_{25}^N
Velocity-Branch
v_1^1 v_2^1 v_3^1 v_4^1 ... v_{25}^1
v_1^2 v_2^2 v_3^2 v_4^2 ... v_{25}^2
... ...
v_1^N v_2^N v_3^N v_4^N ... v_{25}^N
ResGCN * M
ResGCN * M
ResGCN * M
Concat
Rearrange the joints
TCN-based modules
Pooling
Mean
Fully Connected
Prediction

Figure 2.3: The proposed general architecture for TCN-based approaches

The general scheme of the proposed TCN-based architectures is presented in Figure 2.3. For each branch, *M* layers of ResGCN type are applied to extract spatial features. This part of extracting spatial features is inspired by the architectures proposed by Song et al. [3]. After spatial features have been extracted for each branch, we concatenate all features. Because we propose to use a module based on TCN type layers that will be able to extract both temporal and spatial features simultaneously, it is necessary to perform a rearrangement of the joints / bones. The proposed rearrangements are detailed in Section 2.2.

To analyse the sequence and extract features from a temporal perspective, we decided to use TCN layers. Thus, we started by testing several types of blocks based on TCN layers. Initially, we used a module based on TCN blocks inspired by the models previously proposed in [1]. Their major disadvantage was that they did not preserve the spatial size, because the TCN unit was based on 1D convolution. Therefore, we performed the concatenation of the extracted features for each joint. Then, we used the resulting 2D tensor as input for the TCN units.

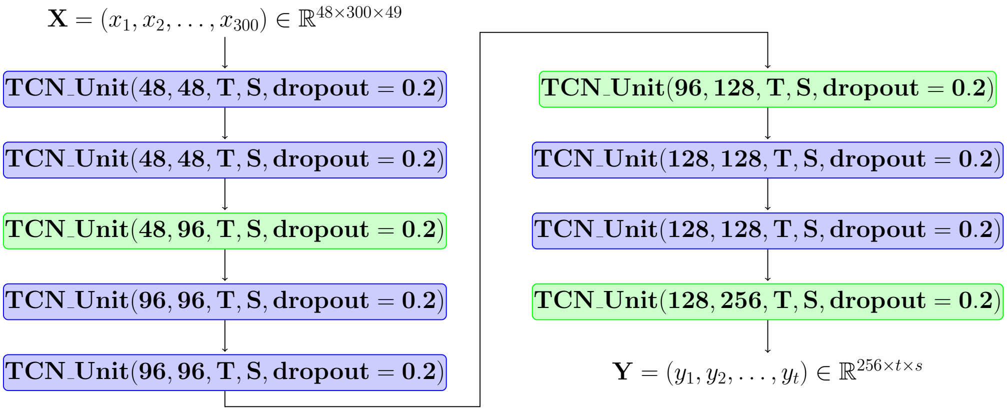

$\mathbf{X} = (x_1, x_2, \ldots, x_{300}) \in \mathbb{R}^{48\times300\times49}$
TCN_Unit(48, 48, T, S, dropout = 0.2)
TCN_Unit(48, 48, T, S, dropout = 0.2)
TCN_Unit(48, 96, T, S, dropout = 0.2)
TCN_Unit(96, 96, T, S, dropout = 0.2)
TCN_Unit(96, 96, T, S, dropout = 0.2)
TCN_Unit(96, 128, T, S, dropout = 0.2)
TCN_Unit(128, 128, T, S, dropout = 0.2)
TCN_Unit(128, 128, T, S, dropout = 0.2)
TCN_Unit(128, 256, T, S, dropout = 0.2)
$\mathbf{Y} = (y_1, y_2, \ldots, y_t) \in \mathbb{R}^{256\times t\times s}$

Figure 2.4: T represents the size of the temporal window, and S represents the size of the spatial window. For the blue blocks, the stride has the value 1, and for the green ones, the stride has the value 2. 300 represents the maximum number of frames, and 49 represents the number of analysed joints (results after the linear rearrangement described in Section 2.2). For each TCN type unit, the padding is determined based on the T and S values.

2.4 RNN-BASED ARCHITECTURES

The architecture used for the Recurrent Neural Network (RNN)-based approach is shown in Figure 2.5. This architecture is similar to the one previously presented in Section 2.3.

In this architecture, the initial layers were applied independently for each skeleton. In the end, a mean of the extracted features was computed. Finally, only the features corresponding to the final hidden state for each sample are kept, and they are passed through a Fully Connected layer to achieve classification.

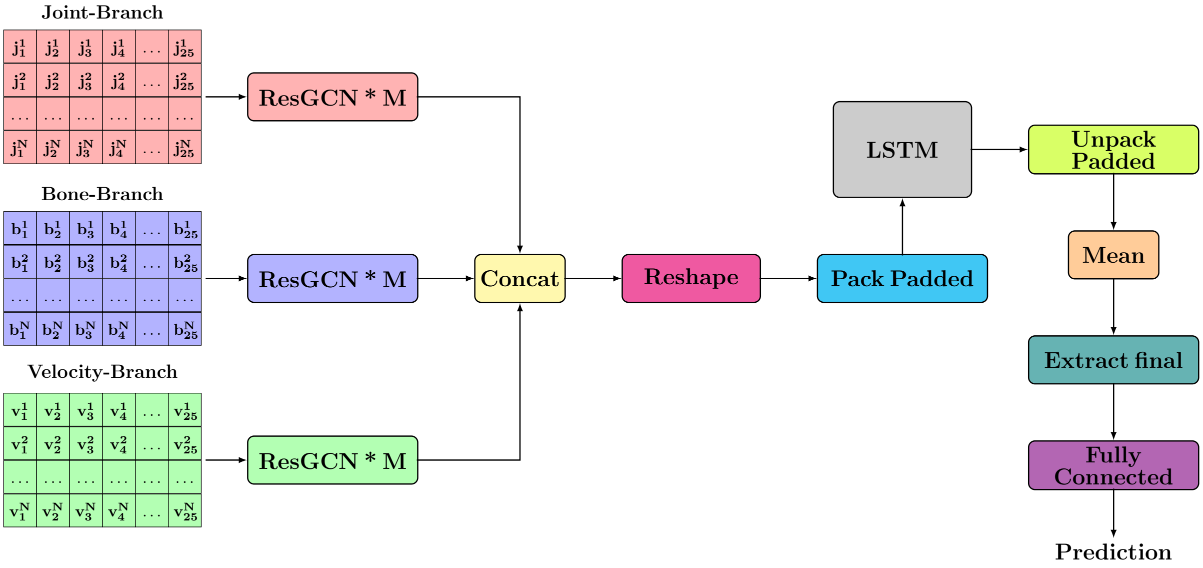


Figure 2.5: The proposed architecture for RNN-based approach

2.5 DISCUSSION

We tested a similar architectural model based on Long short-term memory (LSTM) to highlight the importance of our methods based on an extended TCN type unit. Even if the inference rate is lower for the TCN-based approach, this aspect could be improved if the parallelization property of this type of neural network were used. Unlike RNN, where the computations for later timestamps must wait for their predecessors to complete, convolutions can be computed in parallel even on simple and powerful SIMD architectures like those found in graphic cards because the same kernel or very similar kernels are independently used in each layer repeatedly. In contrast, in terms of performance, the best results were obtained for TCN-based approaches.

In the case of TCN-based methods, for samples that contained less than 300 frames, the padding operation is applied. In contrast, for LSTM-based approaches, this aspect is avoided by using specific optimization operations (e.g., Pack padded sequence in Pytorch). This may be one of the reasons why the inference speed obtained for TCN-based approaches is lower than that obtained when using LSTM.

An important advantage of TCN-based architectures is the ability to change their receptive field size in many ways. For instance, stacking more dilated (causal) convolutional layers, using larger dilation factors, or increasing the kernel size are all possible options, each with its specific advantages and disadvantages depending on the finer details of each implementation. This allowed us to use different values for the receptive field depending on the domain. The best performances were obtained when we used a kernel size equal to 5 for the spatial domain and a kernel size equal to 9 for the temporal domain.

3 HUMAN ACTION RECOGNITION IN AMIRO SOCIAL ROBOTICS FRAMEWORK

The integration of a module specialized in the action recognition within a framework represents a challenging problem. In this chapter, we present the AMIRO robotics platform, emphasizing how we developed the component for HAR within this platform. Thus, we describe the general integration pipeline proposed together with two neural models specialized in the recognition of 8 human actions. These results would not have been possible without the help of Alexandra Ștefania Ghiță.

3.1 AMIRO FRAMEWORK

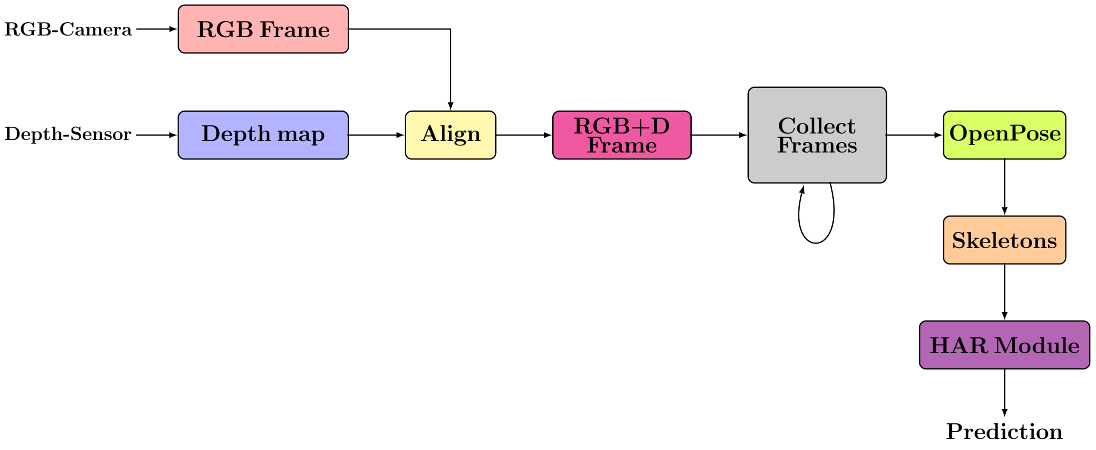


Figure 3.1: The pipeline of the component that recognizes human actions. This component is integrated in the Vision module.

The architecture of the component used for the recognition of human actions is presented in Figure 3.1. As can be seen, this component comprises 3 important submodules:

1. Data Acquisition Module;
2. Feature Extraction Module;
3. HAR Module.

The first module is the one that collects and processes the data provided by the two important sensors of the robot: RGB-Camera and Depth-Sensor. Because the robot is not equipped with a single sensor that allows the collection of both types of data, it was necessary to align the data from a temporal and spatial perspective. After being processed, this data is collected and, when it becomes sufficient, it begins to be analysed by the next module. Because the HAR module works with skeletal data, it was necessary to introduce an intermediate module that aims to transform RGB+D frames into skeletal data.

3.2 GENERAL PIPELINE FOR ACTION RECOGNITION

To have a robot that can interact with humans, the robot must identify the actions performed by the person it is monitoring or to whom it must send a notification. Thus, the module for recognizing human activities was included in the proposed framework for the Pepper robot, to obtain a socially assistive robot.

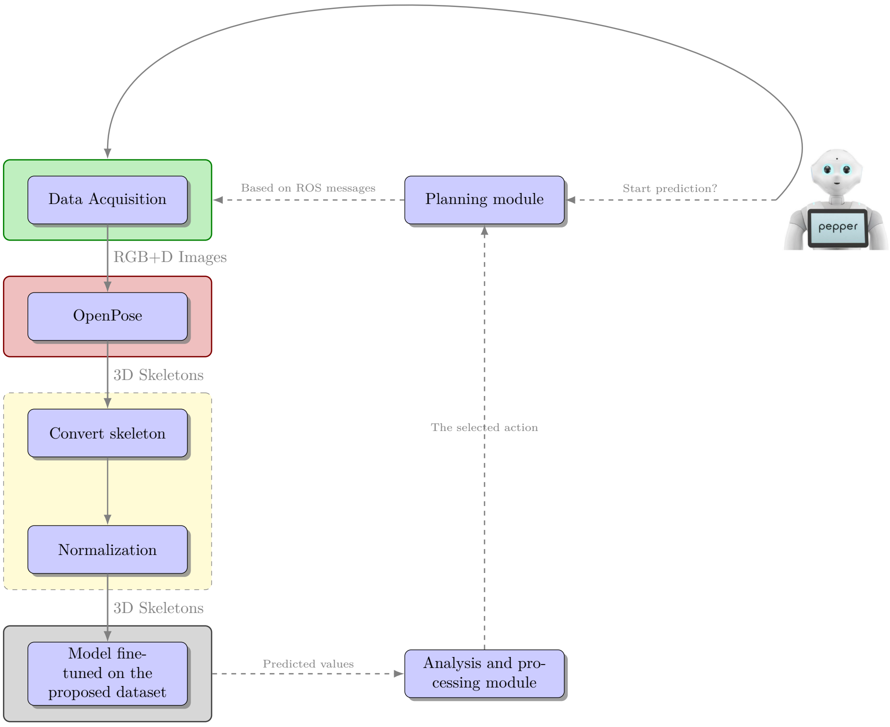


Figure 3.2: The architecture of the complete integration process of the human action recognition module

3.3 MULTI-STAGE ARCHITECTURE

The pipeline of the entire integration process is shown in Figure 3.2 and the network architecture used as a human action classifier is presented in Figure 3.3. In the paper [4], we presented an architecture that failed to correctly differentiate similar actions (such as *drink water* and *sneeze/cough* or *hand waving* and *pointing to something*). Thus, we decided

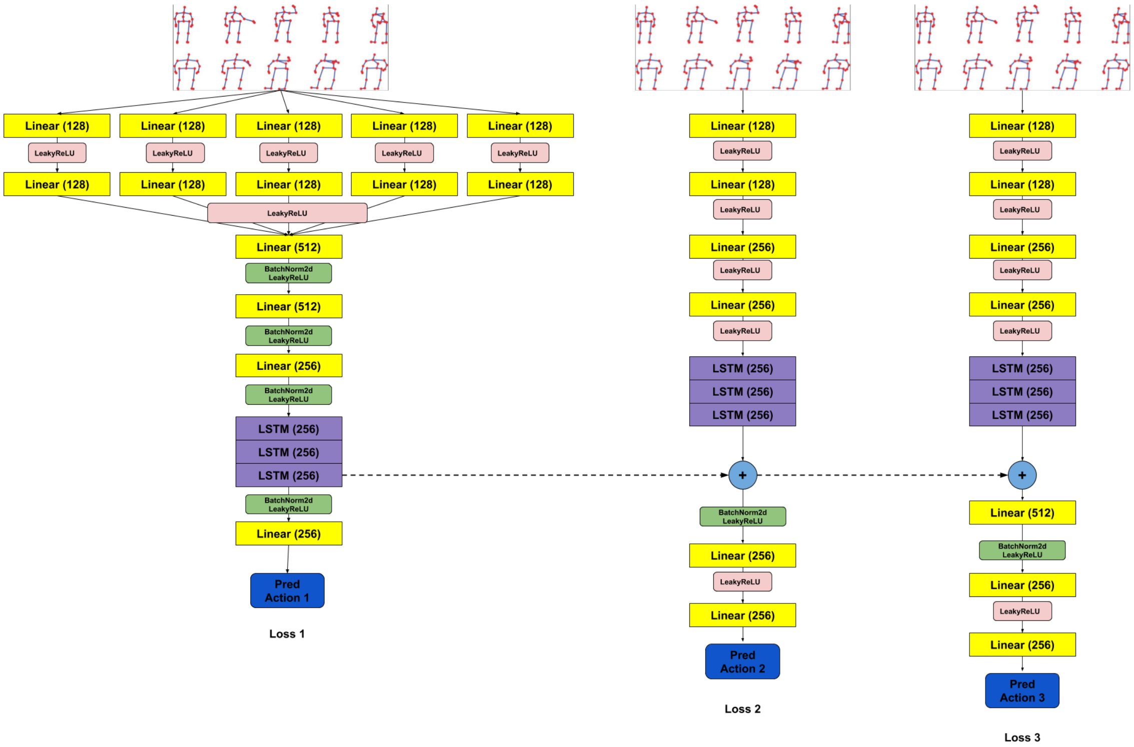


Figure 3.3: The architecture of the neural network used to recognize human action

to propose a new extended architecture that contains two additional stages that receive as input the sequence processed with the coordinates of the skeletons but also takes into account information from the previous stage. Each stage uses a series of linear layers to extract features from the skeletal coordinate sequence and an LSTM network for analyzing these temporal sequences. A loss function was applied for each stage. The action classifier used was trained on the NTU RGB+D dataset [5] and then specialized on a dataset collected using the Pepper robot. The dataset collected with the Pepper robot contains a subset of 8 actions considered to be relevant and challenging for a robot used as a personal assistant. Because there are very similar actions (e.g. *playing with phone/tablet* and *typing on a keyboard*), within this subset of selected actions, a complex model was needed to be able to differentiate the actions correctly. Thus, we proposed an architecture composed of three stages. After this classifier was trained, the result provided by stage three was used as the final prediction. The architecture of the classifier is an improved version of a model tested and analyzed in our previous work [1].

3.4 TCN-BASED ARCHITECTURE

Another approach used for the human action classifier was based on TCN layers. The architecture of this type of classifier is presented in Figure 3.4. Initially, transformations are applied that ensure data preprocessing. These transformations have two fundamental purposes: the normalization of data and the addition of descriptors related to motion (speed and acceleration). After the preprocessing steps, the data are passed through some convolutional layers responsible for extracting relevant spatial information. This information is analysed, from a spatial and temporal perspective, by TCN type layers. The TCN layers coloured in blue in Figure 3.4 are the ones that will preserve the dimensions, and the ones coloured in green are the layers that will change the dimensions (the spatial one - the number of joints and the temporal one - the number of frames). The features determined after the application of TCN type layers are passed through a pooling layer and then classified using a fully connected layer.

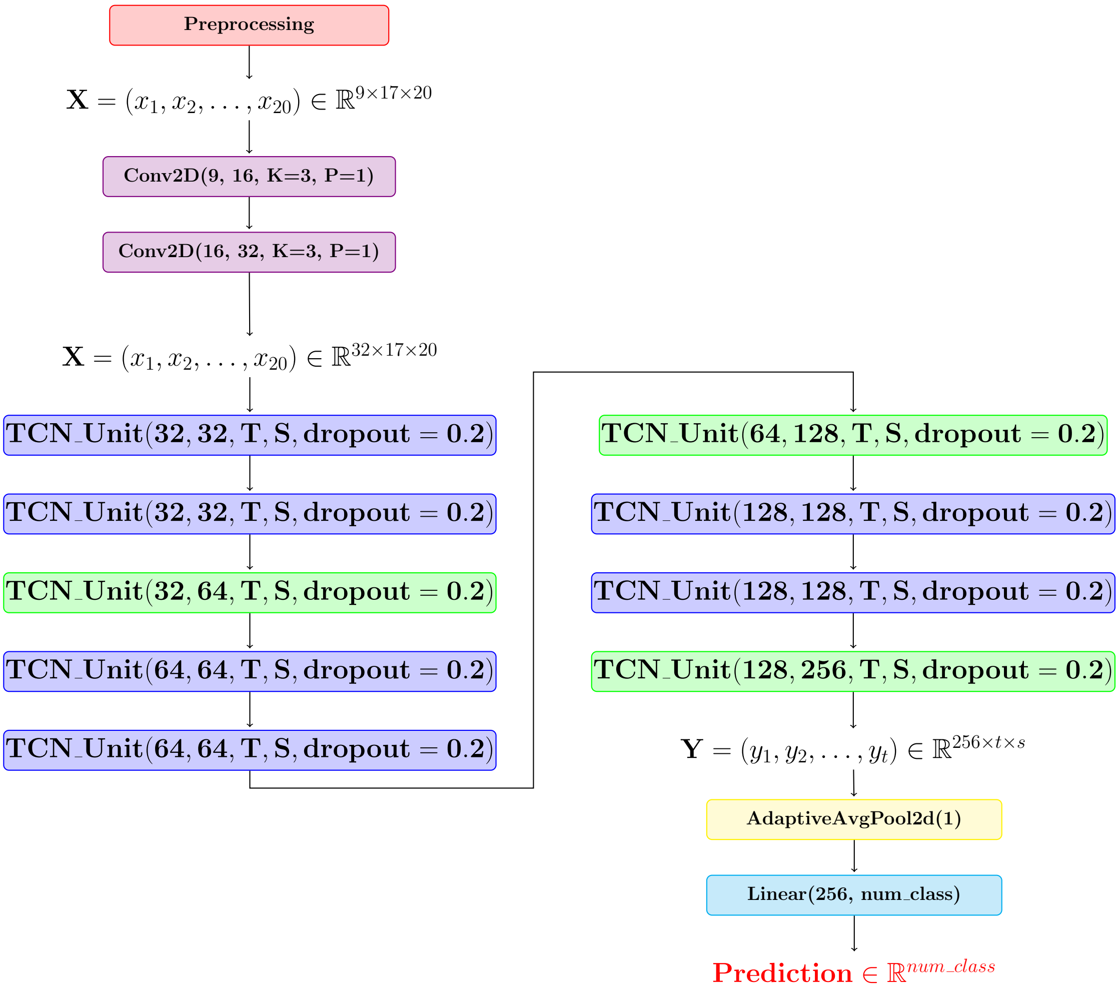

Preprocessing
$\mathbf{X} = (x_1, x_2, \ldots, x_{20}) \in \mathbb{R}^{9\times17\times20}$
Conv2D(9, 16, K=3, P=1)
Conv2D(16, 32, K=3, P=1)
$\mathbf{X} = (x_1, x_2, \ldots, x_{20}) \in \mathbb{R}^{32\times17\times20}$
TCN_Unit(32, 32, T, S, dropout = 0.2)
TCN_Unit(32, 32, T, S, dropout = 0.2)
TCN_Unit(32, 64, T, S, dropout = 0.2)
TCN_Unit(64, 64, T, S, dropout = 0.2)
TCN_Unit(64, 64, T, S, dropout = 0.2)
TCN_Unit(64, 128, T, S, dropout = 0.2)
TCN_Unit(128, 128, T, S, dropout = 0.2)
TCN_Unit(128, 128, T, S, dropout = 0.2)
TCN_Unit(128, 256, T, S, dropout = 0.2)
$\mathbf{Y} = (y_1, y_2, \ldots, y_t) \in \mathbb{R}^{256\times t\times s}$
AdaptiveAvgPool2d(1)
Linear(256, num_class)
Prediction $\in \mathbb{R}^{num\_class}$

Figure 3.4: The architecture of TCN-based classifier. In the case of simple convolutional layers, K represents the kernel size, and P represents the padding. T is the value used for *kernel_size* corresponding to the temporal dimension, and S is the *kernel_size* value corresponding to the spatial dimension.

3.5 ROBOTIC PERCEPTION DATASET

Our dataset contains 720 records, approximately 15% of the size of the NTU RGB+D dataset, with 90 records for each selected action. The actions were performed by 10 participants. Each action was recorded 3 times, from 3 different angles, to simulate the different angles from which the robot can see a person. Each set of recordings for action was filmed in 3 different scenes, the scenes being inside a building, with artificial light. We introduced this data set in the [4].

3.6 DISCUSSION

The proposed evaluation scenario highlights a series of disadvantages presented by the current version of the human action recognition module. If the robot is too close to the subject, then the skeleton predicted by OpenPose [6] is incomplete. Moreover, if there are occlusions with other objects, then the predicted skeleton is incomplete or the coordinates of some joints are incorrectly predicted. Given that the module for recognizing human activities was trained using samples in which the coordinates for all joints appeared, in such situations with a partial skeleton poor results are obtained. Also, when the robot approaches the user, the appearance in the camera may vary, leading to more difficult activity recognition. This can be mitigated in two ways: by training against a more diverse dataset, becoming more robust against joint occlusions or observation distance, as well as by enhancing the *activity recognition task* to include a forwards–backwards movement of the mobile base, so as to obtain a similar user bounding box proportion within the frame, like the ones in the dataset.

4

SPATIO-TEMPORAL NEURAL NETWORK WITH HANDCRAFTED FEATURES

In this chapter, we present the Spatio-Temporal Neural Network with Handcrafted Features approach, which consists of a data preprocessing phase followed by a spatio-temporal neural model to recognize the action. The neural model introduced in this section is one with multiple input branches based on TCN and GCN layers. In contrast to the existing neural architectures, our model presents a reduced inference time, obtain results comparable with state-of-the-art methods, and offers the possibility of determining an activation map that can be useful in the explainability process.

4.1 SPATIO–TEMPORAL MODEL

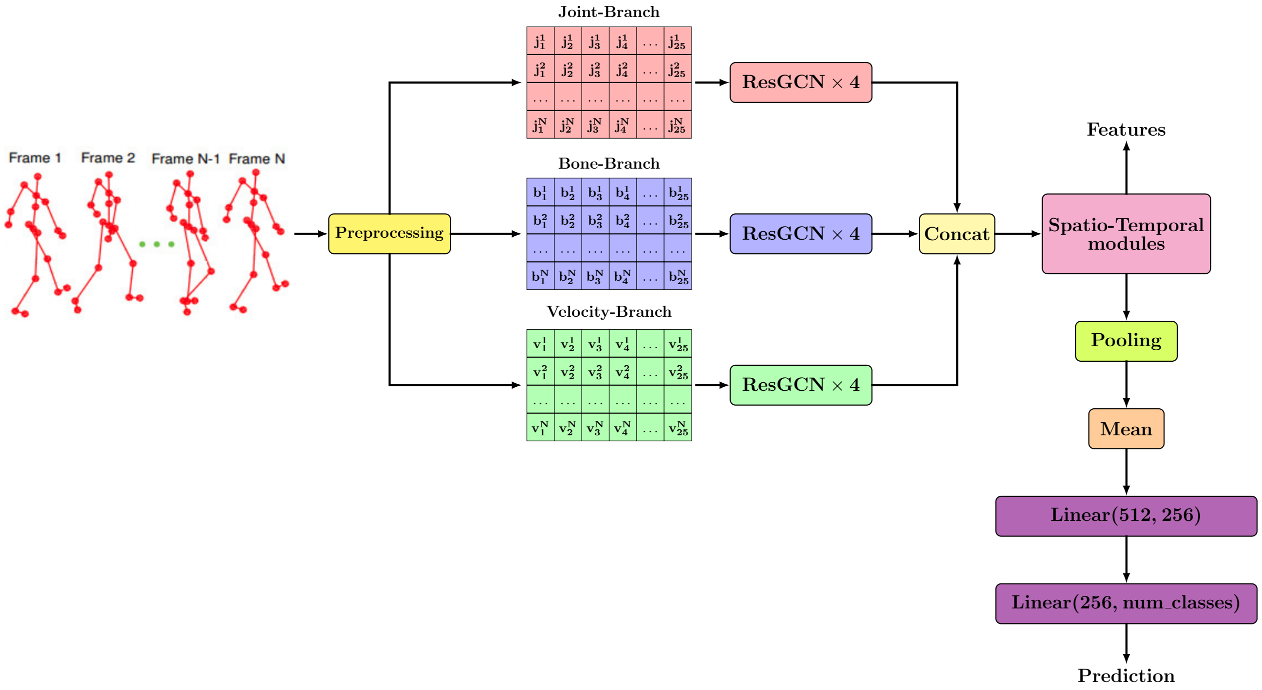

Frame 1
Frame 2
Frame N-1
Frame N
Preprocessing
Joint-Branch
j_1^1 j_2^1 j_3^1 j_4^1 … j_{25}^1
j_1^2 j_2^2 j_3^2 j_4^2 … j_{25}^2
j_1^N j_2^N j_3^N j_4^N … j_{25}^N
Bone-Branch
b_1^1 b_2^1 b_3^1 b_4^1 … b_{25}^1
b_1^2 b_2^2 b_3^2 b_4^2 … b_{25}^2
b_1^N b_2^N b_3^N b_4^N … b_{25}^N
Velocity-Branch
v_1^1 v_2^1 v_3^1 v_4^1 … v_{25}^1
v_1^2 v_2^2 v_3^2 v_4^2 … v_{25}^2
v_1^N v_2^N v_3^N v_4^N … v_{25}^N
ResGCN × 4
ResGCN × 4
ResGCN × 4
Concat
Features
Spatio-Temporal modules
Pooling
Mean
Linear(512, 256)
Linear(256, num_classes)
Prediction

Figure 4.1: The structure proposed for the neural model used to solve the Human Action Recognition problem

The general architecture for the proposed approach is presented in Figure 4.1. To normalize and extend the features resulting from the preprocessing process, we used 4 ResGCN layers for each branch. We concatenate the data resulting from the application of these layers and use the resulting tensor as input for a Spatio-Temporal module. The proposed architecture returns two types of results: the features obtained after the application of the Spatio-Temporal module and the final prediction.

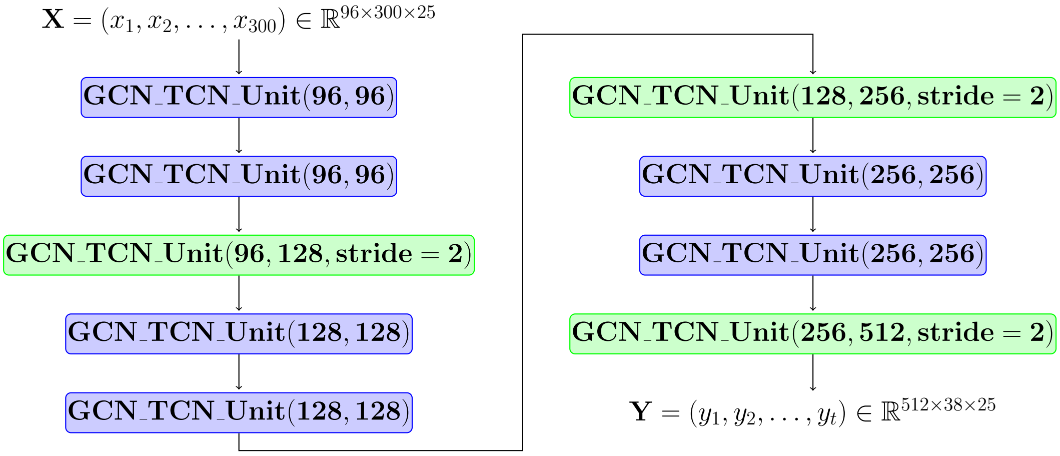

$\mathbf{X} = (x_1, x_2, \dots, x_{300}) \in \mathbb{R}^{96\times300\times25}$
GCN_TCN_Unit(96, 96)
GCN_TCN_Unit(96, 96)
GCN_TCN_Unit(96, 128, stride = 2)
GCN_TCN_Unit(128, 128)
GCN_TCN_Unit(128, 128)
GCN_TCN_Unit(128, 256, stride = 2)
GCN_TCN_Unit(256, 256)
GCN_TCN_Unit(256, 256)
GCN_TCN_Unit(256, 512, stride = 2)
$\mathbf{Y} = (y_1, y_2, \dots, y_t) \in \mathbb{R}^{512\times38\times25}$

Figure 4.2: The structure of the Spatio-Temporal module included in the general architecture presented in Figure 4.1. Layers highlighted in blue use strides with the value equal to one and preserve both dimensions (spatial and temporal)

For the design of the Spatio-Temporal module, we used GCN–TCN type units. The architecture of such a block is shown in Figure 4.3. These blocks were proposed by Chen *et al.* in [7].

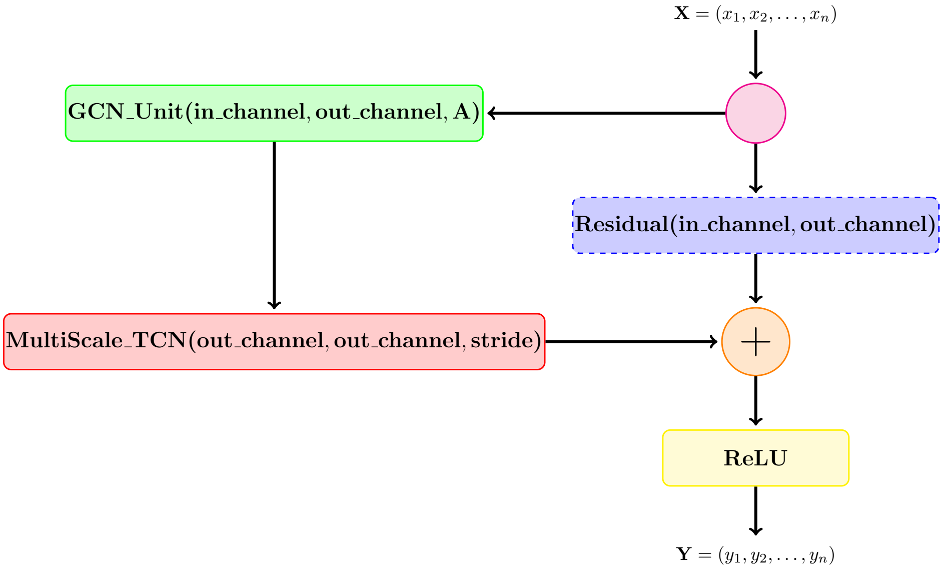

$\mathbf{X} = (x_1, x_2, \dots, x_n)$
GCN_Unit(in_channel, out_channel, A)
Residual(in_channel, out_channel)
MultiScale_TCN(out_channel, out_channel, stride)
+
ReLU
$\mathbf{Y} = (y_1, y_2, \dots, y_n)$

Figure 4.3: Architecture used for GCN–TCN type units. **A** represents the matrix that describes the graph. The residual layer is applied only if *in_channel* $\neq$ *out_channel*. This architecture was proposed by Chen *et al.* in [7]

4.2 EXPLAINING THE NETWORK PREDICTION

An explanation as to why this is happening can bring valuable information both for those who develop applications based on HAR and for future advances on solving this problem. As opposed to methods that explain network prediction for images, there are quite few proposals trying to explain HAR, and most are limited to 3D Convolutional Neural Network (CNN) models. Only a recent paper [3] proposes a visualization of skeletons, which tries to discover the most essential body parts over a whole action sequence, in an attempt to obtain a more explainable representations for different action sequences. We consider that explainability of HAR based on a skeleton model is a very promising and sound path to understanding network prediction and our proposed model was developed starting also from this premise.

To achieve an explanation for the recognized action, we used the features resulting from the application of the Spatio-Temporal module and the weights of the last two linear layers. Starting from these, we determined for each frame which are the most important joints considered by the network in terms of activations and pictured them in an Activation Map. We also considered the importance that the network attaches to each frame. Similarly, in case two new skeletons appear, we checked the importance for each one.

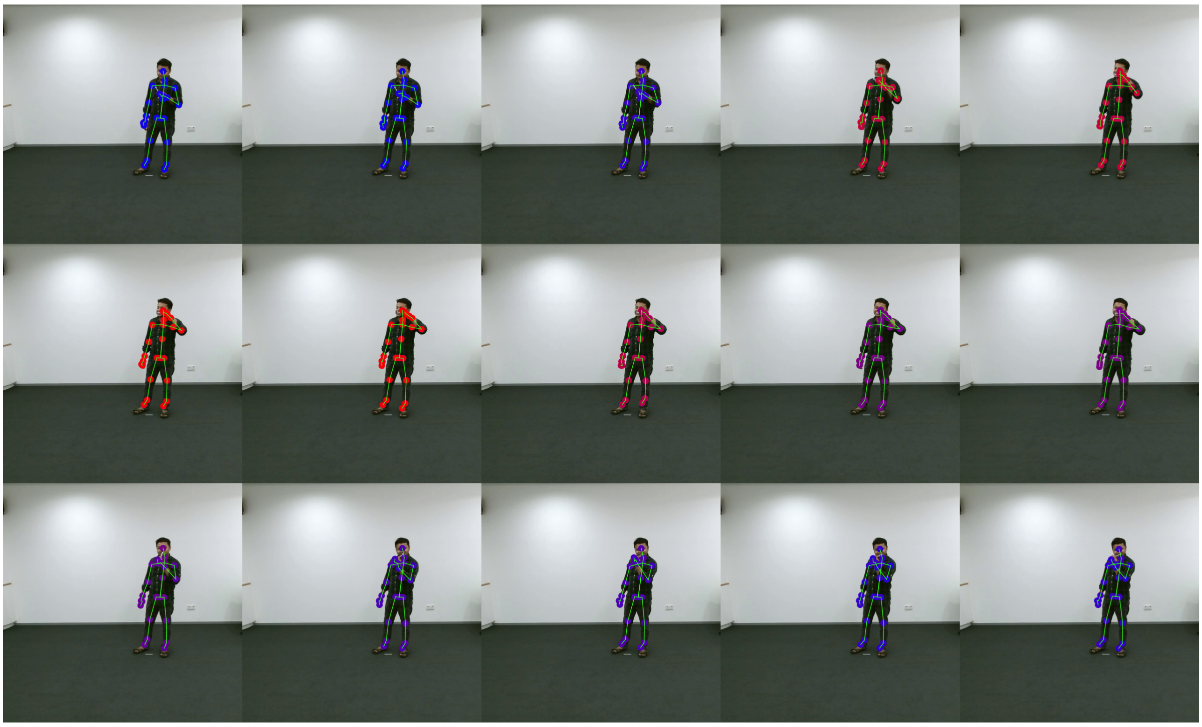


Figure 4.4: Sample from the test subset for the *brushing teeth* action. The network correctly predicts this action with a 100% probability

In order to highlight some qualitative results obtained for samples from the test set, using the protocol for which the network obtained the lowest score, Cross-Subject v2, we performed the testing from the perspective of 3 actions: *brushing teeth* (Figure 4.4), *drink water* (Figure 4.5) and *eat meal or snack* (Figure 4.6).

We presented in Figure 4.4 some frames from a sample for the *brushing teeth* action. We selected these frames using a step of 10 frames. For these frames, we highlight the human skeleton and choose the colours for each joint according to the importance generated by the neural model. We coloured the joints considered by the model unimportant in blue, and, for the rest, we used the red color considering the intensity generated by the model. The network correctly identifies the key moment of the action. The intensity of the joints are highlighted in images 4, 5, 6, 7, 8.

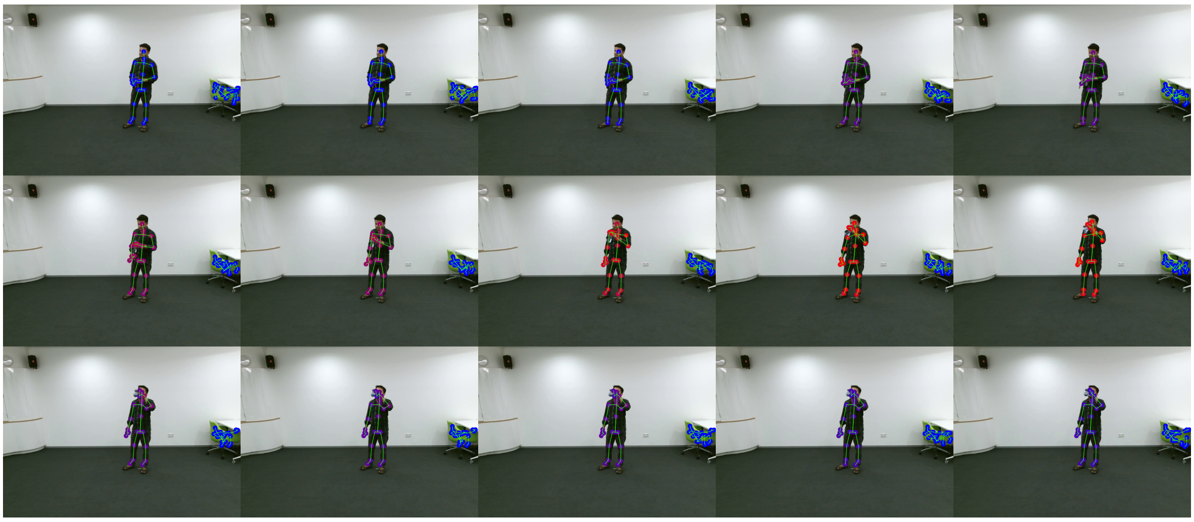


Figure 4.5: Sample from the test subset for the *drink water* action. The network correctly predicts this action with a 100% probability. We sampled the frames with a step of 5 frames

We highlight in Figure 4.5 an example which highlights an error generated by the Kinect sensor. For each frame, two skeletons appear, even if a single person performs the action. The sensor confused the chair in the image with a skeleton, for which it predicts some distorted data. In our approach, we provide the data from the two skeletons as input for the network. The network correctly identifies the skeleton of interest, completely ignoring the false one. This example highlights the robustness of our model and motivate the correctness of the prediction. In all the examples, we distinguish a difference between the colour intensity for the joints of the hand that performs the action and the other ones. This is especially visible in images 6 and 7. We also notice that, in the last frames, the importance associated with the joints decreases because the coordinates do not change considerably. It is worth to note that for this sample, the model makes a correct prediction with a confidence of 100%.

In the last selected qualitative example, we presented a repetitive action. As seen from the frames included in Figure 4.6, the network captures this aspect. In these images the action *eat meal or snack* is presented. The model correctly classifies this action with very high confidence (99.7%) and identifies the frames where the person starts eating. This time, being a repetitive action, the temporal dimension is the one to which the model pays more attention. Therefore, only in the last images could a discrepancy be distinguished between the importance associated with each joint at the same time.

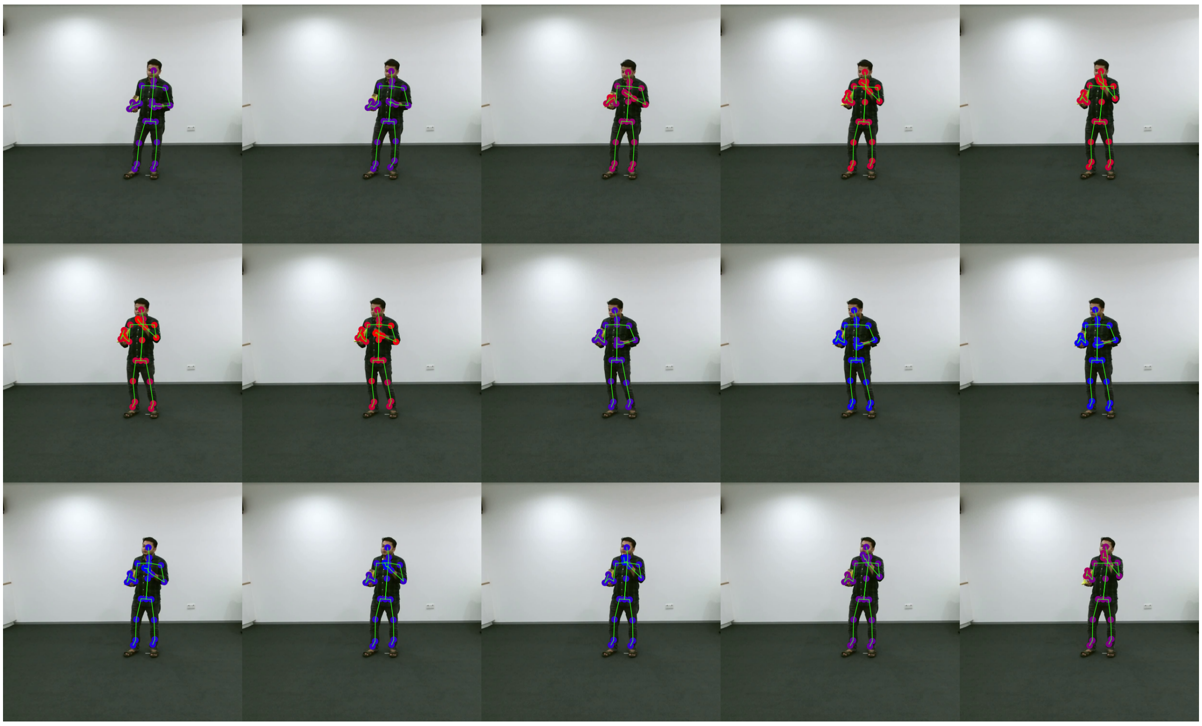


Figure 4.6: Sample from the test subset for the *eat meal or snack* action. The network correctly predicts this action with a 99.7% probability. We sampled the frames with a step of 5 frames

4.3 DISCUSSION

In this chapter we proposed a methodology for Human Action Recognition that consists of a preprocessing stage, in which geometric features and data normalization are used to achieve a better performance, followed by a spatio-temporal neural network architecture that combines TCN and GCN layers to capture both the spatial and the temporal dimension of the action. We showed that our proposed model is able to obtain accuracy results similar to state-of-the-art ones and has a lower inference processing time, a robust behaviour in case of incorrect identified skeletons by the sensors, and the capacity to explain the recognized action (or the incorrectly identified one) by highlighting the most important joints considered by the network in terms of activations and the importance that the network attaches to each frame.

We performed a thorough analysis of the network behavior on the 2 versions of NTU RGB+D (60 and 120 actions) for all the 4 protocols proposed in the literature, in case of both correctly recognized actions and incorrect recognized ones, and we linked this analysis with the explanation capability of the model. We also highlighted the fact that the errors in classification are generated by very similar actions (some even not discernable by a human).

Based on the features provided by the model and the weights from the last two linear type layers, we can generate statistics that present the most important joints considered by the model and the most relevant frames in the performed action. Thus, we performed the analysis and visualization of the reasons behind the predictions for actions performed by one or two people, for single or repetitive actions, showing how the network gives an importance to the spatial dimension and/or the temporal dimension.

5 FAST TEMPORAL GRAPH CONVOLUTIONAL MODEL FOR SKELETON-BASED ACTION RECOGNITION

In this chapter, we present the approach proposed for the problem of human action recognition, which consists of a preprocessing stage and a neural model based on various types of convolutions layers. This approach represents an improvement to the previous contribution described in Chapter 2.

5.1 TEMPORAL GRAPH CONVOLUTIONAL MODEL

The complete pipeline designed for our solution is presented in Figure 5.1. It contains two stages: a stage for calculating the features and a stage for applying the proposed neural model.

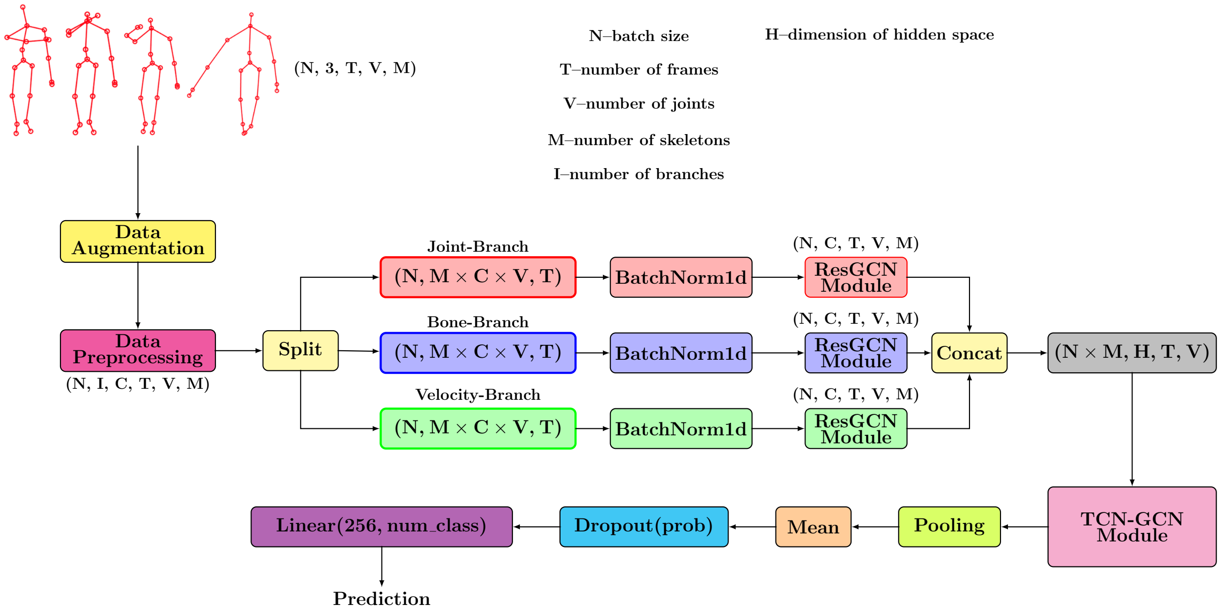


Figure 5.1: The pipeline proposed consists of two fundamental components: the data processing stage and the convolutional model

5.1.1 *ResGCN Module*

In the pipeline proposed within our method, we independently use a module based on ResGCN blocks to extract spatial or temporal dependencies from the data corresponding to each branch. In Figure 5.2, we highlight the proposed structure for this module. The tensor provided as input has the number of channels $C = 6$, and the module reshapes it to be applied independently for each skeleton. Thus, the batch size becomes equal to $N \cdot M$, where N represents the initial batch size, and M is the number of skeletons.

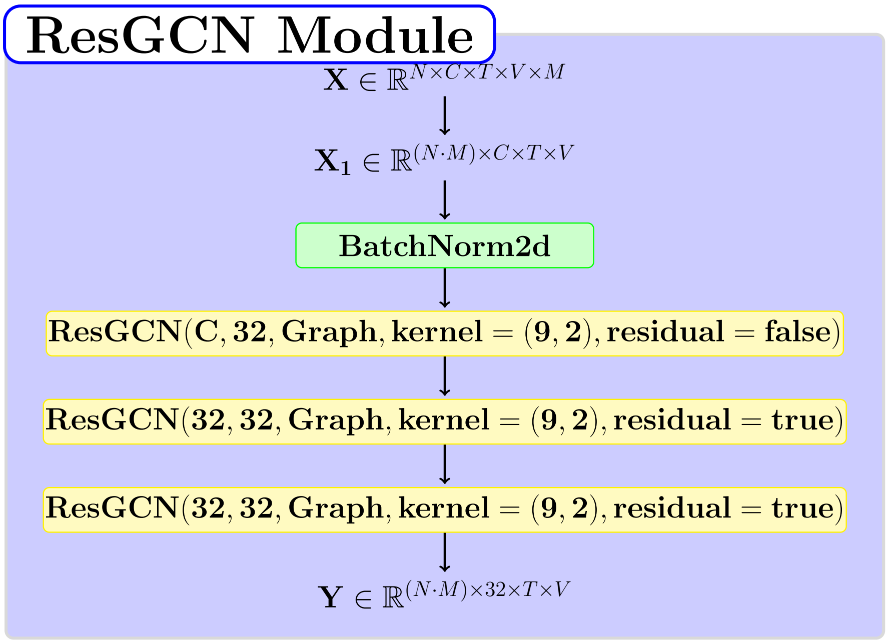

ResGCN Module
$\mathbf{X} \in \mathbb{R}^{N\times C\times T\times V\times M}$
$\mathbf{X_1} \in \mathbb{R}^{(N\cdot M)\times C\times T\times V}$
BatchNorm2d
$\mathbf{ResGCN(C, 32, Graph, kernel = (9, 2), residual = false)}$
$\mathbf{ResGCN(32, 32, Graph, kernel = (9, 2), residual = true)}$
$\mathbf{ResGCN(32, 32, Graph, kernel = (9, 2), residual = true)}$
$\mathbf{Y} \in \mathbb{R}^{(N\cdot M)\times 32\times T\times V}$

Figure 5.2: The structure of the ResGCN layer-based module

The graph used for these ResGCN-type units is one in which the maximum graph distance is set to 2. The bottleneck structure used for the spatial and temporal blocks ensures a model with high inference speed and which requires a low number of epochs for training. Moreover, the module presents an additional optimization, obtained by changing the batch size and applying operations in parallel for the two skeletons.

5.1.2 *TCN-GCN Module*

The spatio-temporal module represents the fundamental part of the proposed neural model, and its structure is described in Figure 5.3. The basic unit of this module is the GCN-TCN block proposed by Chen et al. [7]. This block is composed of two of the most relevant types of neural layers for the human action recognition problem: Graph Convolutional Network (GCN) and TCN. In our approach, we used a graph in which the neighborhood of each node contains the entire skeleton graph for these blocks. This module consists of 9 layers applied sequentially, and each one is a GCN-TCN Unit. Three of these layers halve the temporal dimension by using a stride value of 2. The blue colored layers in Figure 5.3 preserve the temporal dimension.

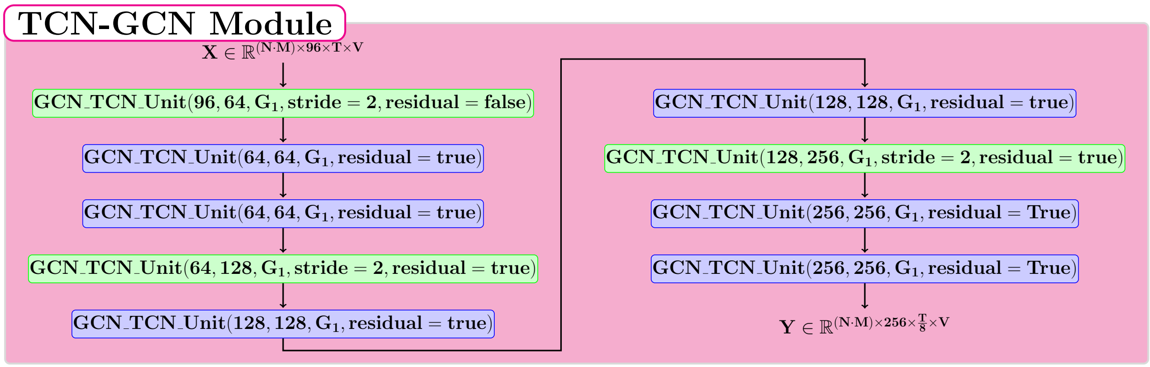

TCN-GCN Module
$X \in \mathbb{R}^{(N \cdot M) \times 96 \times T \times V}$
GCN_TCN_Unit(96, 64, G$_1$, stride = 2, residual = false)
GCN_TCN_Unit(64, 64, G$_1$, residual = true)
GCN_TCN_Unit(64, 64, G$_1$, residual = true)
GCN_TCN_Unit(64, 128, G$_1$, stride = 2, residual = true)
GCN_TCN_Unit(128, 128, G$_1$, residual = true)
GCN_TCN_Unit(128, 128, G$_1$, residual = true)
GCN_TCN_Unit(128, 256, G$_1$, stride = 2, residual = true)
GCN_TCN_Unit(256, 256, G$_1$, residual = True)
GCN_TCN_Unit(256, 256, G$_1$, residual = True)
$Y \in \mathbb{R}^{(N \cdot M) \times 256 \times \frac{T}{8} \times V}$

Figure 5.3: The proposed structure for the TCN-GCN module

5.2 EXPERIMENTAL RESULTS

Table 5.1: Performances achieved by the proposed model for the Cross-Subject protocol (v1—60) depending on the number of epochs set for training

Method	**Total Number of Epochs**	**Top 1**	**Top 5**	**Best Epoch**
Fast Convolutional (200 frames, dropout = 0.3, theta = 0.3)	100	89.37	98.25	97
Fast Convolutional (200 frames, dropout = 0.3, theta = 0.3)	70	89.37	98.17	66
Fast Convolutional (200 frames, dropout = 0.3, theta = 0.3)	50	89.05	98.20	49
Fast Convolutional (200 frames, dropout = 0.3, theta = 0.3)	30	87.65	98.09	29
Fast Convolutional (200 frames, dropout = 0.3, theta = 0.3)	10	82.73	97.13	10

In Table 5.1, we included an analysis of the experimental results achieved by the proposed model for the Cross-Subject test protocol. We obtained these results by changing the number of epochs used in the training process. Based on them, we highlighted the fact that the proposed model does not require a large number of epochs for training. As can be seen, there is no difference between the Top 1 accuracy obtained for 100 epochs and the one obtained for 70. Moreover, the difference between the Top 5 accuracy for these two training configurations is insignificant. Due to the small number of parameters and the transformations proposed in this approach, it is possible to obtain an accuracy of 82.73% using only 10 epochs. In other words, using a larger number of training epochs helps the model distinguish between similar classes for which it predicts close probabilities. Moreover, we can see that by introducing augmentation transforms and using a Dropout-type layer with a probability of 0.3, we managed to obtain a model with a good generalization power that does not overfit. This observation is certified by the results obtained for 100 epochs which are not weaker than those for 70.

5.3 DISCUSSION

This contribution introduces an approach to the problem of human action recognition using skeletal data. The proposed method is based on a neural network designed using some of the most used types of convolutional layers: GCN and TCN. The innovation of our current approach consists in designing a performant and fast pipeline that augments the data, determines geometric features and uses a neural network to identify the action. The neural model included in this pipeline is also innovative and combines the advantages of previously proposed networks.

The current version represents an improvement to our previous method proposed in the paper [8]. The current method starts from the same categories of features but by introducing some optimizations at the level of the structure of the neural model and by applying augmentation techniques, it improves the inference speed, increases the accuracy achieved by the model and reduces the number of parameters. The pipeline proposed in the current version additionally contains the augmentation stage that does not exist in the previous method. The proposed neural model is also different from those analyzed in the previous chapter. The TCN-based architecture introduced previously contains a module in which we integrate only TCN-type layers. For the method described in the current article, we used a module based on GCN-TCN layers. In the current model, the GCN-type layers extracted the spatial dependencies, unlike our previous solution in which the TCN-type layers analyzed the sequence both temporally and spatially. More precisely, this time the size of the spatial window used by the TCN layers is 1, and, in this way, the network preserves the spatial dimension until the Polling block is applied.

6 CONCLUSIONS

The main goal of this thesis was to construct a robust module for solving the HAR problem. In this sense, we identified the challenges that characterize this problem and the shortcomings of the existing approaches. Starting from these observations, we developed a series of approaches focused on the main properties that we considered a solution for a robotic platform must have: generalization capacity, reduced model size, small pre-processing time and high inference speed. The neural models proposed in these approaches are based on the most used types of deep neural networks for tackling the HAR problem: Graph Convolutional Network, Temporal Convolutional Network and Long Short Term Memory Network.

We used one of the most challenging datasets, NTU RGB+D, to evaluate our models. The representation chosen for the proposed approaches is in the form of skeletal data, and the features used are determined using the 3D coordinates predicted by the Kinect sensor. The proposed solutions achieve performances comparable to the state of the art for all testing protocols proposed by the NTU RGB+D dataset. Moreover, we have adapted two approaches to be evaluated using an own collected dataset from a robotics perspective. In this way, we highlighted the generalization capacity of the proposed models and showed that we can apply the concept of transfer learning to this problem.

We have carried out a deep analysis of the HAR problem consisting of a presentation of the main types of modalities used and an overview of the practical applications that require such a module. For each type of modality, we highlighted some characteristics and the limitations presented by the solutions based on the respective representation. To highlight the importance of each field of application, we included some representative approaches. Starting from all this, we outlined the challenges that exist in this research area and identified five categories of sub-tasks for the HAR problem.

6.1 CONTRIBUTIONS

The main original contributions of this thesis are the following:

- In the original paper in which TCN-type networks are introduced, the authors implement these blocks using 1D convolutional layers. In the case of the HAR problem, the transformation of the sequence into one compatible with 1D convolutions does not allow preserving the spatial dimension. To avoid this inconvenience, we proposed a modified version for TCN layers that uses 2D convolutions. In addition, we suggested extending the concept of dilated convolution to the spatial level. The obtained experimental results demonstrated that this modification is beneficial for the performance of the model.

- The spatial dimension is fundamental for the HAR problem. That is why we focused on it by proposing a way to integrate two methods of rearranging the joints. One of the rearrangement methods is based on a 2D matrix and is proposed by us in [1]. The second method is a 1D type and starts from the human skeleton perceived as a tree whose root is the joint considered closest to the center of gravity.

- Integrating a human action recognition module into a robotic framework is a complex operation because we must consider many aspects. We designed a general pipeline integrated into the AMIRO framework and tested using the Pepper humanoid robot. In this pipeline we used the pre-trained OpenPose [6] model to extract the 2D coordinates from the RGB images, extracted the third coordinate from the corresponding depth map and applied an original neural architecture to classify the actions. We obtained good results in terms of accuracy and showed that this pipeline works in real-time scenarios. The integration of the human action recognition module into the AMIRO framework was done in collaboration with Alexandra Ștefania Ghiță.

- We designed two neural architectures that we tested on a dataset collected from a robotics perspective. The first architecture is a multi-stage one that uses Linear layers for extracting features and LSTM cells for analyzing the temporal sequence. This neural network returns 3 predictions, and we calculate separate loss for each stage. The main goal we pursued in its design is to improve the prediction from one stage to another. Thus, stage 2 uses the output resulting from the application of LSTM cells from stage 1, and stage 3 proceeds similarly to the output from stage 2. The second architecture is one in which we use 2D convolutional layers to extract features and then analyze them with the help of a temporal module made up of TCN-type layers.

- Generalization and overfitting are not enough analyzed for the approaches proposed to solve the HAR problem. That is why we decided to investigate this direction of research. For this, we used two neural architectures that we trained using the NTU RGB+D dataset and then tested them on a dataset collected using the Pepper robot. Thus, we analyzed the concept of transfer learning from the perspective of different datasets. It was necessary to introduce a skeleton conversion operation from the format predicted by OpenPose [9] to the format used by the Kinect sensor.

- The features from which the analysis of a neural model starts influence obtained performances. That is why we focused on this topic and we propose a method to determine geometric features that can help the neural model to achieve better performance. This method also contains a variant of data normalization.

- We present a spatio-temporal neural architecture, which combines Temporal Convolutional and Graph Convolutional layers, and we report the results of the proposed model on NTU RGB+D [5, 10] benchmark, for all the test protocols. This approach is a non-black-box solution in which the model outputs a feature tensor together with the results, thus being able to explain predictions.

- Recently, the field of Explainable Artificial Intelligence (XAI) has received a lot of attention and various methods have been proposed to determine an explanation for network prediction. In other words, researchers no longer want to perceive neural models as black boxes. Following this trend, we showed how we can design the neural model such that it is able to explain the prediction from the perspective of the importance of each joint. Moreover, we presented some eloquent examples to highlight the importance of such explanations.
- Data augmentation is a technique by which we can artificially increase the size of the dataset to obtain a model that performs with better precision. Regarding the HAR problem, it is complicated to collect enough samples for action like falling. We also want the trained model to become invariant to the person's position or physical characteristics. To reduce these limitations, we designed a pipeline in which we integrated a data augmentation stage. We performed extensive experimental validation of the approach demonstrating its potentials compared to the state of the art.

6.2 PERSPECTIVES AND FUTURE WORK

We can conclude that the HAR problem is a complex task that cannot be solved using a classical deterministic algorithm. Approaches based on Deep Learning techniques manage to partially solve this problem. We consider that this problem remains an open research topic because the existing solutions are specialized in identifying the subset of possible actions and present various limitations. Moreover, the integration process of these solutions in frameworks used to solve real-life scenarios brings additional challenges.

A first improvement that can be achieved at the architectural level consists in trying to improve the final part of the neural model that deals with classification by proposing an improved architectural model, possibly based on a multi-stage architecture. For this, we can build a complex neural architecture consisting of several sub-models, each one specialized in identifying a subset of actions.

The second direction of research is an attempt to improve the understanding of the neural model by introducing additional data to provide more context about the environment in which the subject performs the action. We consider that adding context information to the process can improve both the performance of the architecture and the explanation capabilities of HAR models.

ACRONYMS

CNN	Convolutional Neural Network
GCN	Graph Convolutional Network
HAR	Human Action Recognition
LSTM	Long Short-term Memory
RNN	Recurrent Neural Network
TCN	Temporal Convolutional Network
XAI	Explainable Artificial Intelligence

BIBLIOGRAPHY

[1] M. Trăscău, M. Nan and A. M. Florea, 'Spatio-temporal features in action recognition using 3d skeletal joints', *Sensors*, vol. 19, no. 2, p. 423, 2019.

[2] Z. Yang, Y. Li, J. Yang and J. Luo, 'Action recognition with spatio-temporal visual attention on skeleton image sequences', 2018.

[3] Y.-F. Song, Z. Zhang, C. Shan and L. Wang, 'Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition', in *Proceedings of the 28th ACM International Conference on Multimedia (ACMMM)*, New York, NY, USA: Association for Computing Machinery, 2020, pp. 1625–1633, ISBN: 9781450379885. DOI: 10.1145/3394171.3413802. [Online]. Available: https://doi.org/10.1145/3394171.3413802.

[4] M. Nan, A. S. Ghiță, A.-F. Gavril, M. Trascau, A. Sorici, B. Cramariuc and A. M. Florea, 'Human action recognition for social robots', in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, IEEE, 2019, pp. 675–681.

[5] A. Shahroudy, J. Liu, T. Ng and G. Wang, 'NTU RGB+D: A large scale dataset for 3d human activity analysis', *CoRR*, vol. abs/1604.02808, 2016.

[6] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, 'Openpose: Realtime multi-person 2d pose estimation using part affinity fields', *arXiv preprint arXiv:1812.08008*, 2018.

[7] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng and W. Hu, 'Channel-wise topology refinement graph convolution for skeleton-based action recognition', in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 359–13 368.

[8] M. Nan, M. Trăscău, A. M. Florea and C. C. Iacob, 'Comparison between recurrent networks and temporal convolutional networks approaches for skeleton-based action recognition', *Sensors*, vol. 21, no. 6, p. 2051, 2021.

[9] Z. Cao, G. Hidalgo, T. Simon, S. Wei and Y. Sheikh, 'Openpose: Realtime multi-person 2d pose estimation using part affinity fields', *CoRR*, vol. abs/1812.08008, 2018. arXiv: 1812.08008. [Online]. Available: http://arxiv.org/abs/1812.08008.

[10] J. Liu, A. Shahroudy, M. L. Perez, G. Wang, L.-Y. Duan and A. K. Chichung, 'Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding', *IEEE transactions on pattern analysis and machine intelligence*, 2019.