Uncovering Human Multimodal Activity Recognition with a Deep Learning Approach

1st Caetano M. Ranieri
ICMC
University of São Paulo (USP)
São Carlos, SP, Brazil
cmranieri@usp.br

2rd Patricia A. Vargas
Edinburgh Centre for Robotics (ECR)
Heriot-Watt University (HWU)
Edinburgh, Scotland, UK
p.a.vargas@hw.ac.uk

3th Roseli A. F. Romero
ICMC
University of São Paulo (USP)
São Carlos, SP, Brazil
rafrance@icmc.usp.br

***Abstract*—Recent breakthroughs on deep learning and computer vision have encouraged the use of multimodal human activity recognition aiming at applications in human-robot-interaction. The wide availability of videos at online platforms has made this modality one of the most promising for this task, whereas some researchers have tried to enhance the video data with wearable sensors attached to human subjects. However, temporal information on both video and inertial sensors are still under investigation. Most of the current work focusing on daily activities do not present comparative studies considering different temporal approaches. In this paper, we are proposing a new model build upon a Two-Stream ConvNet for action recognition, enhanced with Long Short-Term Memory (LSTM) and a Temporal Convolution Networks (TCN) to investigate the temporal information on videos and inertial sensors. A feature-level fusion approach prior to temporal modelling is also proposed and evaluated. Experiments have been conducted on the egocentric multimodal dataset and on the UTD-MHAD. LSTM and TCN showed competitive results, with the TCN performing slightly better for most applications. The feature-level fusion approach also performed well on the UTD-MHAD with some overfitting on the egocentric multimodal dataset. Overall the proposed model presented promising results on both datasets compatible with the state-of-the-art, providing insights on the use of deep learning for human-robot-interaction applications.**

***Index Terms*—Deep learning, CNN, LSTM, TCN, RNN, human activity recognition, human-robot-interaction.**

I. Introduction

Current development on different research fields have risen interest on applications of social robots as interactive tools to assist humans, usually elderly people or people with special needs. In real-world scenarios, roboticists may rely on human activity recognition [1]. This consists in processing sensing data from smartphones and wearable devices to identify semantically understandable interactions amongst the user, the environment and the robot. These technologies are important for the development of automated solutions for human-robot interaction applications that are still mostly based on Wizard of Oz approaches [2].

São Paulo Research Foundation (FAPESP), grants 2017/02377-5, 2018/25902-0 and 2017/01687-0, and Brazilian National Council for Scientific and Technological Development (CNPq), grant 306151/2018-9. This research was carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP, grant 2013/07375-0. Additional resources were provided by the Nvidia Grants program.

Here we address this challenge by proposing a deep learning model for human activity recognition from videos and inertial sensors. Inertial data may be made available from smartphones or wearable devices such as smartwatches. In situations in which social robots are present, video data may also be obtained from the robot's camera(s). Regardless of the modality, deep learning techniques have shown promising results on activity recognition, although feature-based approaches are still competitive in some cases [3]. Most advances on video classification were built on the Two-Stream ConvNet [4], whereas satisfactory results on inertial data have been provided by the combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) [5]. Some other attempts dealing with multimodal data, focused on fusing inertial data with depth images [6].

In our investigation we have used two datasets for daily activities: the egocentric dataset presented by Song *et al.* [1] and the University of Texas at Dallas Multimodal Human Activity Recognition Dataset (UTD-MHAD) [7]. Our proposed model relied on RGB videos and inertial data, experimenting different possibilities for modelling the temporal dependencies on both modalities. In this regard, first, we are proposing to add Temporal Convolutional Networks (TCN) [8], which consists on an feasible alternative to Recurrent Neural Networks (RNN) on sequence modelling. Second, a feature-level fusion approach is considered as an alternative to the late fusion generally used when dealing with video temporal streams.

II. Human Activity Recognition

Human activity recognition comprises of a wide research field, involving different input modalities and classification of activities on distinct levels of abstraction. In the case of videos, this modality may rely not only on structured data built on controlled environments, but also on unconstrained videos obtained from the Internet [9]. The same has not been true for raw sensors such as inertial measurement units (IMU), which may aggregate, for instance, 3D accelerometer, gyroscope and magnetometer [10]. For those, datasets are typically designed and recorded under controlled environments. In this work, we address a multimodal approach, in which data from inertial sensors has been applied to enhance video-based activity recognition. Even though, single-modality approaches

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

also influenced our research, providing guidelines on several developments on our proposed methods.

The UCF101 dataset [11] is probably the most relevant benchmark for video classification available. It is composed by 101 categories distributed into human-object interaction, body movements, human-human interaction, musical instruments playing and sports. More recently, large-scale datasets have been deployed and the most relevant one is the Kinetics dataset [12]. Since its volume of data may take several terabytes, it is not always feasible to work directly with datasets of such scale. As we discuss thoroughly on Section III, a Convolutional Neural Network (CNN) trained from scratch on data derived from the UCF-101 dataset has been adopted as a building block for most of our proposed architectures.

A detailed literature review regarding methods for video classification was presented by Herath *et al.* [13]. Most deep learning approaches may be belong into two categories: multiple stream networks or spatio-temporal networks. The most influential multiple stream network is the Two-Stream ConvNet proposed by Simonyan *et al.* [14]. It was composed by a spatial CNN trained to classify RGB frames and a temporal CNN trained on stacks of dense optical flows from sequential frames. This approach have evolved and an important advance was the Temporal Segment Network [15]. Spatio-temporal networks are characterised by combinations between CNN and LSTM, such as the Long-term Recurrent Convolutional Networks (LRCN) [16], or 3D ConvNets (C3D), as presented by Tran *et al.* [17]. Our approach is composed of multiple streams. However, the video temporal streams were built with similar basic principles as the LRCN.

For inertial sensors, a dataset often used in studies centred on wearable devices is the PAMAP2 [18]. The OPPORTUNITY [10] dataset is also relevant, as it provides a large set of sensors not only wearable, but also placed on objects or distributed around an environment. The neural networks architectures used to classify those datasets are almost always based on combinations between CNN and LSTM. A systematical analysis of deep learning techniques for inertial data, experimented in datasets such as the both mentioned, was performed on Hammerla *et al.* [19], in which regular deep neural networks (DNN) were compared to CNNs and three LSTM-based architectures. In Rueda and Fink [20], features extracted from CNNs were on the basis of three architectures: a regular CNN, a variation called DeepConvLSTM, in which LSTM layers would replace fully-connected layers, and the CNN-IMU, composed of parallel convolutional blocks whose outputs were concatenated and fed to fully-connected layers. The InnoHAR architecture [21] consists of a stack of Inception modules followed by two recurrent layers based on Gated Recurrent Units (GRU), and led to improved results on both PAMAP2 and OPPORTUNITY datasets. A detailed overview of the literature regarding smartphone sensors was provided on the recent work of Sousa Lima *et al.* [22], in which different datasets and algorithms, including deep networks, were broadly revised.

Regarding multimodal datasets with videos and inertial sensors, most of them were recorded with depth cameras, as discussed on the survey provided by Chen *et al.* [6]. Datasets such as the UTD-MHAD [7], adopted in the experiments, and the 50 Salads [23] provide not only video and inertial measurements, but also positioning of skeleton joints, which are often used as an important input for the proposed methods [24]. In Chen *et al.* [7], Depth Motion Maps (DMM) were obtained from depth images, statistical descriptors were adopted for the inertial data and the RGB videos were not considered. Classification was performed with Collaborative Representation Classifiers (CRC). Song *et al.* [3], another object of our analysis, brought a different approach, in which scripted actions were performed by 10 participants and recorded with a Google Glass. In a following paper [1], the authors applied the two-stream ConvNet to classify the videos from their dataset and a DeepConvLSTM to classify the sensor data, performing fusion by averaging or max-pooling their outputs.

III. Proposed Model

In this article, we propose to build on the Two-Stream ConvNet [14] and extend it to the case in which another modality composed by IMU sensor data is present. This modality, comprised by multivariate 1D temporal series, has been considered as an additional stream, called *inertial*, as illustrated in Fig. 1. An Inception-V3 network [25], adapted to take pairs of optical flow matrices (U, V) as inputs, has been previously trained on the UCF-101 dataset. Therefore, instead of taking three input color channels of the RGB images, the network would take the two optical flow channels: vertical and horizontal. Further, its last layer was removed, in order to provide a feature vector for each timestep of the video. In other words, the penultimate layer of the Inception-V3 would generate a feature vector of length 2048 of a given timestep, and this network would be applied independently for each timestep considered. A much simpler CNN was implemented to extract features from the inertial stream, which could be used as inputs to a LSTM or a TCN block. Those outputs could be concatenated to the features obtained from other time-dependent streams, particularly the video temporal stream. In the later case, we are assuming that both of them are related to the same amount of time on the sample, so that $c = t_s \times \omega_s$, where t_s is the number of timesteps of stream $s \in \{\text{video}, \text{inertial}\}$, ω_s is its frequency and c is the time amount, in seconds, shared between the streams. Given such assumption, discrepancies on the number of timesteps at the time of the concatenation could be resolved by sampling from the stream with more timesteps.

More precisely, the LSTM and TCN models for temporal modelling were applied to the features extracted by CNNs, and its outputs were fed to a softmax layer for classification. Although LSTM was already applied for video classification on previous literature [26], the suitability of TCN, which has shown to lead to equivalent or even better results in sequence modelling [8], has not been extensively applied to this context.

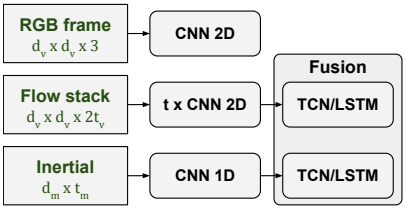


Fig. 1: Proposed framework for multimodal activity recognition, where d refers to the number of features and t, to the number of timesteps considered for a given stream. Fusion may be performed at feature-level, by combining the features from different modalities obtained from the CNNs. Both LSTM and TCN layers were considered for modelling long-term dependencies.

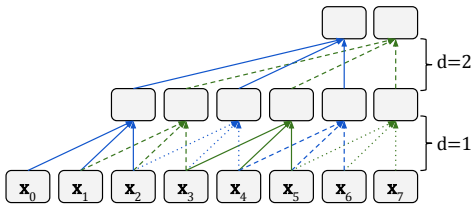
A. Temporal Convolutional Networks

The LSTM architecture is a classical approach for dealing with long-term temporal dependencies in sequences [27]. Recently, it has led to several advances on deep learning, especially regarding language and speech recognition [28]. The success of the LSTM and of its most famous variation, the Gated Recurrent Unit (GRU) [29] turned recurrent neural networks the standard starting point when dealing with deep learning for sequence modelling. However, as Bai *et al.* [8] argued, approaches based solely on convolutional networks could provide results as good as recurrent approaches, and therefore it may be worth to consider them as well. In this context, the temporal convolutional network (TCN) comprises of a neural architecture capable of dealing with long-term dependencies.

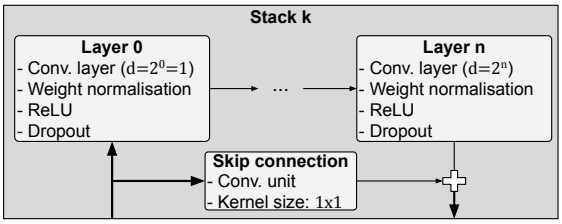
The temporal information would be dealt in such networks by stacks of *dilated causal convolutions*, which are illustrated in Fig. 2a. The *causal* denomination is derived from the connections between the layers. A filter of size k processes a timestep t plus the $k-1$ preceding timesteps, in order to capture the idea of causality. The *dilated* denomination refers to the inclusion of a dilation factor d, responsible for amplifying exponentially the receptive field of the convolutions, as more levels are added to the network. A regular convolution is the particular case in which $d = 1$. Considering a 1D input sequence $\mathbf{x} \in \mathbb{R}^n$ and a filter $f : \{0, \ldots, k-1\} \rightarrow \mathbb{R}$, the dilated convolution operator may be defined as in equation 1, where $s - d \cdot i$ refers to the direction of the receptive field to the past.

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i} \qquad (1)$$

Each convolutional stack would be followed by weight normalisation, an activation function (e.g., ReLU) and spatial dropout, composing residual blocks as shown in Fig. 2b. The advantage of such blocks is the so-called skip connections, which allow the input data to be fed directly not only to the next block, but also to each of the following blocks. To fix the differences of dimensions, 1×1 convolutions may be applied to adjust the previous inputs before they are combined to the output of a block .



(a)



(b)

Fig. 2: Elements of a TCN. (a) Stack of dilated causal convolutions, with $k = 3$ and dilation factors $d = \{1, 2\}$. The convolutional layers within each stack are comprised by dilated causal convolutions. (b) Generic residual stack, with n dilations, where the dilation factor is increased each layer as a power of 2. Multiple stacks may be concatenated one after another, as in a residual neural network.

B. Video Classification

The architecture for activity recognition on videos was based on the Two-Stream ConvNet, in which the spatial features are extracted by a CNN with RGB frames as input, and temporal dependencies, by a CNN which takes optical flow matrices. Before being fed to the correspondent neural network, RGB frames or optical flow matrices were supposed to be cropped to $d \times d$. For both streams, the InceptionV3 network [25] was adopted as a base model. For the spatial model, we applied a straightforward transfer learning from a model previously trained on the ImageNet dataset [30], in which only the softmax layer was replaced and further trained with the weights of all other layers being fixed.

For the temporal stream, illustrated in Fig. 3, t_v successive pairs of optical flow with shape $d_v \times d_v \times 2$, each corresponding to a single timestep of a sequence, were fed independently to the CNN. This approach is different from Simonyan *et al.* [14], in which a CNN took as input a stack of optical flow matrices related to successive timesteps, i.e., the architecture was composed by a single CNN with input shape $d_v \times d_v \times 2t_v$. Here, a determined CNN, trained from scratch to classify the UCF-101 dataset and deprived from its last softmax layer, would process the pairs of optical flow matrices. The result would consist of a feature vector with shape (a_v, t_v), where a is the number of features generated by the output of the CNN - in the case of the network InceptionV3, $a_v = 2048$. In other words, this feature vector would be a multivariate time series with a_v variables

and t_v timesteps. LSTM networks are commonly seen as a good choice for modelling such one-dimensional signals, so as TCNs, as discussed in subsection III-A. Therefore, LSTM and TCN were both considered as candidate layers for this part of the proposed architecture. Finally, the last output of whichever network was used would be fed to the softmax layer for the classification.

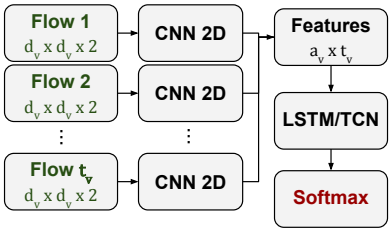


Fig. 3: Network architecture for the temporal stream. The inputs are the pairs of dense optical flows from a frame sequence. Each pair with shape $d_v \times d_v \times 2$, is processed by a shared CNN, and the a_v features obtained from the last layer of the CNN (prior to the softmax layer previously withdrawn) are taken as timesteps for a LSTM or TCN.

C. Inertial Data Classification

In order to classify the inertial data, it has been adopted a one-dimensional version of the same principle as that one for video temporal stream: a CNN to extract features, followed by a LSTM or TCN to model long-term temporal dependencies. This approach has similarities to the work of Rueda and Fink [20]. However, a network architecture was deployed having in mind the particular issues that would arise when performing a fusion with the video temporal stream. Particularly, since the convolutions on the inertial data would be performed on the time domain, and each pooling layer would reduce the resolution at this given domain to the ratio of its kernel, we had to be cautious with the increasing of the depth of this CNN. With this aim, we have considered only two Conv1D layers: the first one with kernel size 1, to increase the number of feature maps, and the second, with size 3, to perform feature extraction. Those layers were followed by a maximum pooling of kernel size 2, which would reduce the number of timesteps t_m to its half, t_n, while still representing the same amount of time (i.e., the time resolution has dropped). The CNN architecture is shown in Fig. 4a.

The a_n features extracted from this CNN were, then, applied as input to a LSTM or TCN block, whose last output was connected to a softmax layer for classification (see Fig. 4b). An important difference between this neural network and that of the video temporal stream is that all the free parameters of both CNN and LSTM/TCN were set to be trainable, i.e., training would be performed end-to-end.

D. Temporal Fusion

In most research on activity recognition based on multiple-stream deep neural networks, fusion was performed at a later stage. For instance, by averaging the outputs of the last layer. Song *et al.* [1] adapted this approach to fuse the video features to those extracted from the inertial data of their egocentric multimodal dataset. Regarding video-only classification, Feichtenhofer, Pinz and Zisserman [31] analysed different methods for feature-level fusion in two-stream ConvNets. Most of the techniques they proposed rely on the spatial dependencies shared by the video temporal and spatial streams. Therefore, they are not suitable for fusion with the inertial stream. However, we could adapt the concatenation of features presented by them to build our feature-fusion approach, since it does not make assumptions on the spatial dependencies between features.

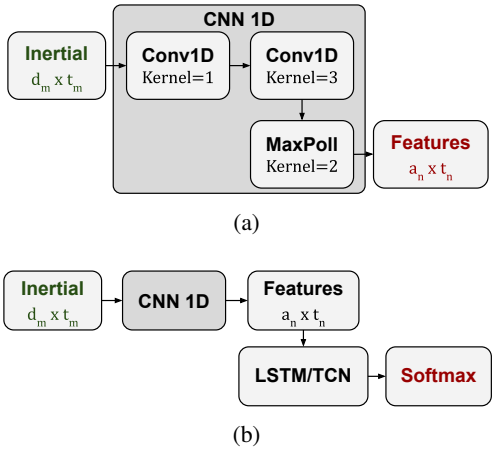


Fig. 4: Neural network applied for the classification of the inertial stream. (a) CNN applied prior to the LSTM or TCN module. Since the convolutions are performed in the time domain, the maximum pooling with kernel size 2 reduces the initial temporal resolution t_m to $t_n = \frac{t_m}{2}$. The number of output features a_n was determined by the number of filters of the second Conv1D layer. (b) For the inertial stream, the inputs are one-dimensional sample sequences. The whole sequence is processed by a CNN in the time domain, which reduces the number of timesteps from d_i to d_j.

The proposed method here, shown in Fig 5, builds on two assumptions: the numbers of timesteps t_v on the videos and t_m on the inertial data are synchronised, referring to the same period of the sample on both streams despite each modality having a different temporal resolution; and that $t_v \leq t_n$. Thus, after applying each of the t_v $(d_v \times d_v \times 2)$ optical flow matrices to CNN 2D and stacking the outputs, and applying the $d_m \times t_m$ inertial data sample to CNN 1D, two feature vectors would be obtained, with shapes $a_v \times t_v$ and $a_n \times t_n$. If $t_v \neq t_n$, the inertial feature vector should be adjusted, what would be done by sampling points that were equidistant in the time domain. After such adjustment, both feature vectors would have the same number of timesteps t_v. Therefore, they may be concatenated in this dimension, resulting in a feature vector of shape $(a_v + a_n) \times t_v$. This feature vector would be fed to a LSTM or TCN block, whose output would be connected to a softmax layer. It is worth to remind that CNN 2D has fixed

weights, already optimised in an ad-hoc manner.

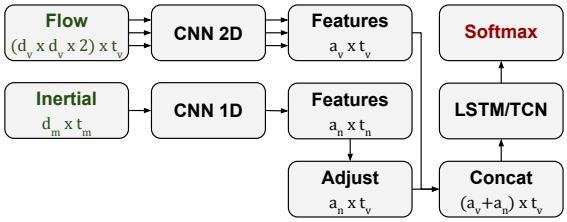


Fig. 5: Framework for feature-level temporal fusion. Features extracted by the video CNN are concatenated to the features extracted by the inertial CNN, composing a feature vector related to a single timestep. The frequencies at each modality are different, thus adjustment by down-sampling is applied to the inertial stream before concatenating, so that the timesteps at all streams are synchronised. After concatenation, the multimodal feature vectors at each timestep are fed to a temporal neural network, be it a LSTM or a TCN module.)

IV. Experimental Setup

All implementations were developed in Python language, using the Keras framework with TensorFlow backend. Before any preprocessing, all videos were proportionally resized so that the smallest side would have size 256. Optical flow was calculated with the TVL1 algorithm [32]. This algorithm has shown, in exploratory experiments, to provide significantly the best classification results among others, although being significantly slower than the Farnebäck algorithm [33], which may be relevant to real-time applications. The networks for the video streams were set to get input frames with shape 224×224, which would be achieved by cropping. Split 1 of UCF101 dataset, suggested by the authors [11], was used to train the CNN for feature extraction in the video temporal stream. The following subsections will present the datasets and the settings for each condition, allowing the results to be reproduced. The code was made available at https://github.com/cmranieri/Deep-Activity-Recognition.

A. Datasets

The experiments were performed on two multimodal datasets: egocentric multimodal [1] and the UTD-MHAD [7]. Those datasets were chosen for their suitability to activities of daily living. Both of them provide the same amount of data from each subject and with respect to each activity. Besides, as they are significantly different in nature, interesting conclusions could be drawn from comparative results.

1) Egocentric Multimodal Dataset: This dataset was generated by a group of 10 participants. They performed a set of 20 activities wearing a Google Glass. Each session length was about 10 seconds. These activities were recorded in different and heterogeneous environments, which provides a lot of visual information, in addition to the movement. Activities were divided into four categories: ambulation, office work, daily activities and exercises. The videos (RGB only) were sampled at 30 Hz, while the sensor data was sampled at 15 Hz. The sensors provided 19 features: the 3D acceleration, magnetic field, linear acceleration, gravity, rotation vector and gyroscope. The data was preprocessed using the L2-norm.

2) UTD-MHAD: This dataset was recorded in a more controlled condition, with 8 participants performing a set of 27 activities, 4 repetitions each. Recordings were performed by a depth and RGB camera (only RGB video was considered in this work) and by two 3D accelerometers. One placed at a band on the user's fist, and the other was placed at the user's waist. Each session lasted about 3 seconds, and the recordings were performed in a controlled room, with the subjects posed facing the camera, at a constant distance and with constant background. The videos were sampled at 15 Hz, and the sensor data, with 6 dimensions corresponding to the two 3D accelerometers, were sampled at 50 Hz.

B. Network Setting

All conditions described in this subsection were experimented on both datasets described in the previous section. The datasets were split following the k-fold cross-validation procedure, with $k = 10$ for the egocentric multimodal dataset and $k = 8$ for the UTD-MHAD, so that data provided by one subject was used for testing, and the remaining data, for training. For the data stream, only one condition was considered, in order to allow for late fusion: an InceptionV3 CNN. As previously stated, transfer learning was applied to a model trained on Imagenet dataset, keeping all weights fixed except for the softmax layer, replaced to match the number of classes of the datasets considered.

The temporal and inertial streams were considered separately and followed the fusion approach of Fig. 5. The CNN applied to extract features of the inertial stream was composed by 256 filters in the first convolutional layer and 512 in the second. As the InceptionV3 ouputs 2048 features, the feature-based fusion provides a vector with video and inertial features in a ratio of 4:1. Both LSTM and TCN were experimented as blocks for temporal modelling, with 128 units and output dropout of 0.3. Regarding the TCN, the kernel size was set to 3, dilations were set do $d = \{1, 2, 4\}$, and the number of residual blocks (i.e., stacks) to 3.

1) Training: The training procedure was adapted from Simonyan *et al.* [14] and Song *et al.* [3]. All models were optimised using the softmax cross-entropy as loss function. The pre-training of CNN for optical flow pairs performed on the split 1 of UCF-101 dataset was ran with the Stochastic Gradient Descent (SGD) optimiser, for $200,000$ steps. In the videos of the goal datasets, data augmentation was performed by random cropping and in the egocentric multimodal dataset, random flipping. We decided not to flip the videos from UTD-MHAD, since some of the activities on that dataset were somewhat symmetric (e.g., wave left and wave right). For the spatial stream, we used SGD with learning rate 10^{-2}, momentum 0.9 and weight decay 10^{-4}, and training was also performed for $30,000$ steps, with batches of size 32. Optimisation on the temporal and inertial streams was

performed in batches of size 16, for $30,000$ training steps, using the RMSProp optimiser [34] with learning rate 10^{-3}.

The number of timesteps was selected so that each snippet would represent 2 seconds of a trial. To reduce the number of video frames, we sampled them so that $t_v = 15$. With the egocentric multimodal dataset, the model was sampled once every 4 frames at the video stream, and the timesteps were set to $t_m = 30$ and $t_n = 15$ for the inertial stream. We sampled once every 2 frames with UTD-MHAD, the timesteps of the inertial stream being set to $t_m = 100$ and $t_n = 50$. Therefore, we had to apply the adjustment depicted in Fig. 5. The same settings were kept when training the inertial stream alone, except the adjustment by sampling in UTD-MHAD.

2) Evaluation: For testing we used the same procedure adopted in the reference papers: a number snippets was considered, with equal time between them, and all of them were submitted to cropping on their four corners and centre. For the egocentric mutimodal dataset, 5 snippets were used to test each video, and the videos from the resulting sequences were also horizontally flipped. For the UTD-MHAD, we considered 2 snippets and no flipping. To make a prediction, output vectors from all snippets of a given sample were averaged.

This procedure was adopted for all models that ran end-to-end, i.e., the models for single-stream and feature-level fusion. For late fusion, one model for each stream was run separately and the output vectors were combined by weighted averaging. The same was done when combining to the spatial stream.

V. Results and Discussion

Fig. 6 shows the number of parameters of each model built for each stream on the egocentric multimodal dataset (UTD-MHAD was fairly alike), including the hybrid model for feature-level fusion. *Late fusion* was not considered a model on itself, since it consists on combining the *spatial* models outputs with one of the *temporal* models. Therefore, at inference time, its number of parameters equals the sum of those present on the models adopted. The *temporal* or *feature fusion* models embed a CNN similar to that of *stream* model. Therefore, their complexity is dependent on the base CNN model adopted.

Since InceptionV3 (adopted on all of our models except for the inertial ones) is expressively more complex than the remaining parts of the architecture, variations on the number of parameters are proportionally small. But yet relevant, since the weights relative to this block are fixed during training. It is noteworthy that TCN model was more complex than LSTM for inertial stream, while the opposite happened for the temporal and feature fusion models. Due to the fact that the temporal block on the inertial stream has shape $512 \times t_v$, against $2048 \times t_v$ on the video stream, thus $2560 \times t_v$ in the feature-fusion models, it may be inferred that the number of input features of the temporal block impacts less the number of parameters in TCN-based than in LSTM-based models. This is expected due to the sparser connectivity of convolutional layers.

The InceptionV3 CNN, which was embedded on the *temporal* and *feature fusion* models to extract features based on single optical flow pairs, was trained separately, prior to the

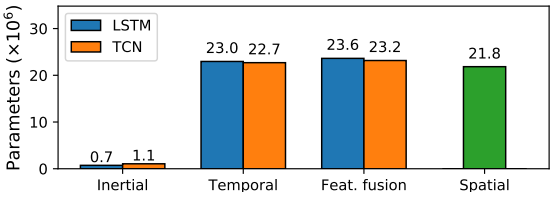


Fig. 6: Number of free parameters of each model analysed, without the softmax layer. The substantially higher number of parameters at the models that involve video processing is given to the InceptionV3 neural network contained on it, which consists of more than 21 million trainable parameters, as made explicit by the number of parameters of the spatial stream.

TABLE I: Mean accuracy of each model for the temporal and inertial streams, using 10 folds for the egocentric dataset and eight for the UTD-MHAD, providing splits such that the test set was composed by all recordings of one subject.

Dataset	Stream	Model	
		LSTM (%)	TCN (%)
Egocentric	Inertial	45.50 ± 7.39	45.50 ± 8.50
	Temporal	69.00 ± 10.68	72.50 ± 11.01
	Feat. fusion	55.50 ± 9.60	53.00 ± 10.77
	Late fusion	74.50 ± 8.20	72.50 ± 9.35
UTD-MHAD	Inertial	63.28 ± 5.71	65.36 ± 9.24
	Temporal	80.02 ± 6.00	81.77 ± 6.49
	Feat. fusion	82.58 ± 5.56	85.47 ± 5.56
	Late fusion	84.90 ± 4.78	83.51 ± 6.25

experiments presented in this paper. It achieved accuracy of **75.15%** on the split 1 of UCF-101, using the same training and evaluation protocol as Simonyan *et al.* [14]. The resulting layers were added as blocks of our architecture, as discussed in section III, and its weights were kept fixed. This was different for the *inertial* stream, whose features were extracted by a simpler network randomly initialised to be optimised together with following layers for temporal modelling. For all models on both datasets, LSTM and TCN blocks were investigated. The mean accuracy of each model for the temporal and inertial streams is shown in Table I.

As some of the results in Table I are close to each other, it may be convenient to compare the performances of each model with respect to some additional aspects. In Fig. 7, we also present the macro F1-score of the models, that is, the average harmonic mean of precision and recall. By penalizing both incompleteness and inconsistency, this measure is a trade-off between type-I and type-II errors per class. The means between the evaluations on each fold were presented in the bars, with standard deviations proportional to the length of the vertical traces on the top of it.

The *spatial* model was obtained by a procedure similar to that of the base CNN block of the *temporal* models. However, it took RGB frames as inputs, instead of pairs of optical flow matrices; and was initialised with ImageNet weights, instead of being trained from scratch. This model was used to build classifications using the three mentioned streams, by fusing it to the models presented in Table I by weighted averaging.

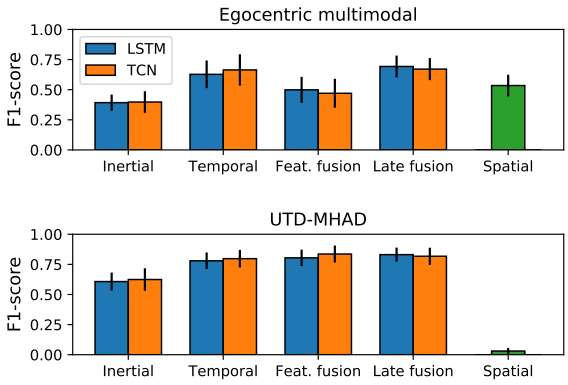


Fig. 7: Macro F1-scores for each model.

TABLE II: Accuracy, for the egocentric multimodal dataset, of the spatial stream models and late fusion by weighted averaging with each of the fused temporal and inertial models. If w_s and w_t are the weights for the spatial and temporal/hybrid streams, the weights are shown using notation $w_s : w_t$.

Temporal model		Weights	Accuracy (%)
Spatial only		1:0	60.50 ± 8.50
LSTM	Video	1:1	72.50 ± 6.42
	Feat. fus.	3:1	69.00 ± 9.69
	Late fus.	1:6	78.50 ± 9.23
TCN	Video	1:2	78.00 ± 10.54
	Feat. fus.	2:1	70.25 ± 9.16
	Late fus.	1:6	80.62 ± 8.81

The fusion weights were selected so that the accuracy was the largest obtained in our experiments. As UTD-MHAD dataset was built on a controlled environment with constant background and without significant differences on objects able to distinguish between activities, the spatial stream was not significantly informative, with accuracy of $(\mathbf{6.74 \pm 3.23})\%$, only slightly above random choice (i.e., 3.70%, given that there were 27 classes). For this reason, fusion between the three streams were made only for the egocentric multimodal dataset. Results are reported in Table II.

A. Discussion

Results from LSTM and TCN-based models were generally quite close to each other, with a slight tendency in favor of TCN models for most single-stream approaches and all models combined with the spatial stream (Tables I and II). The feature-level fusion approach was successful in the UDT-MHAD, surpassing the accuracy of the late fusion when coupled with a TCN block and achieving the best accuracy for this dataset, of 85.47%. Since this dataset is endowed with other modalities, skeleton joints and depth frames, it was expected that the proposed model would perform below the most accurate models on the literature. Still, our proposal may be seen as competitive, since most of our results outperformed those reported on the reference paper [7], which achieved, at most, overall accuracy of 79.10%. It might be noticed that our approach relies only on RGB and inertial data, which are more widely available and may be included in different sorts of systems. With a more complex model, in which LSTM networks also modelled depth information, Li *et al.* [35] achieved an accuracy as high as 95.31%.

On the egocentric multimodal dataset, feature-fusion approaches had suffered from overfitting, with fast optimisation and very high training accuracy. However, average test accuracy is below the temporal stream alone, which has shown lower accuracy during all the training procedure, and actually was harder to optimise than the other models. Considering inertial and temporal streams, the best accuracy was achieved by the late fusion of LSTM-based models (74.50%). This result was curiously different when the models were further combined with the spatial stream, with the late fusion of TCN models achieving the best overall accuracy for this dataset among our models, e.g., 80.62%. Although this was only compatible to the best multiple stream CNN model presented by Song *et al.* [1] which reported 80.50%, it might be noticed that our approach presents some advantages. As we relied on a previously trained CNN to extract features from the optical flow matrices, with a very reduced set of parameters left to be optimised in a LSTM or TCN block, it provides the flexibility to work with different and arbitrarily complex CNNs for this aim. Moreover, since the number of parameters left to be optimised is relatively low, with our approach one can work with larger sequences of data even with a modest hardware.

The F1-scores shown in Fig. 7 were consistent with the accuracy results, thus there were no issues regarding classes with very high precision and low recall or the opposite. Besides, models with higher accuracy have also shown higher F1-score, i.e., both measures were suitable to make comparisons.

The proposed framework may contribute to further applications on human-robot interaction [36] [37], especially on scenarios which demand social interaction between user and robot [38] [39].

VI. Concluding Remarks and Future Work

In this paper, a new model for human activity recognition on videos and data from inertial sensors was proposed. First, different neural networks were analysed as building blocks for the temporal processing, particularly Long Short-Term Memory (LSTM) and Temporal Convolutional Networks (TCN). Second, fusion between the inertial and video temporal streams were not only performed through late fusion of the output layers, but also at feature-level. All those approaches were analysed separately, for different sets of modalities, and thorough comparisons were done.

Focus was given to modelling the temporal dependencies in sequences of tuples of inertial data, features extracted from optical flow and fusion between those approaches. For the temporal feature extraction, we adopted Long Sort-Term Memory (LSTM) units and Temporal Convolutional Networks (TCN). A feature-fusion approach was also proposed and compared to the more traditional late fusion approach, commonly adopted on multiple stream CNNs. The RGB frames were also contemplated, with output features from a spatial CNN

further combined to the other models through weighted averaging, achieving accuracies up to 80.62% for the egocentric multimodal dataset, and 85.47% for the UTD-MHAD without considering depth data.

Experiments were performed on the egocentric multimodal dataset and UTD-MHAD. Models obtained with LSTM and TCN blocks both led to excellent accuracies, with TCN, which we have brought as a novelty to this application, performing slightly better in many circumstances. The feature-fusion approach led to good results in UTD-MHAD dataset. However, it was unable to generalize well on the egocentric multimodal. Overall, the proposed model presented promising results on both datasets compatible with the state-of-the-art, which provided further insights on the use of deep learning for human-robot-interaction applications.

Future work will contemplate depth images as an additional stream, since this may be introduced to social robots in several circumstances. We have already built a multimodal dataset for activities in domestic environments, with videos and inertial data from smartwatches and smartphones, to be used on deep learning models in human-robot interaction applications. This dataset will be made publicly available once we finish the anonymisation procedures.

REFERENCES

[1] S. Song, V. Chandrasekhar, B. Mandal, L. Li, J.-H. Lim, G. S. Babu, P. P. San, and N.-M. Cheung, "Multimodal multi-stream deep learning for egocentric activity recognition," in *IEEE CVPRW*, 2016.

[2] J. T. Browne, "Wizard of oz prototyping for machine learning experiences," in *2019 Conference on Human Factors in Computing Systems*, 2019, pp. 1–6.

[3] S. Song, N.-M. Cheung, V. Chandrasekhar, B. Mandal, and J. Liri, "Egocentric activity recognition with multimodal fisher vector," in *IEEE ICASSP*. IEEE, 2016, pp. 2717–2721.

[4] C. Feichtenhofer, A. Pinz, R. P. Wildes, and A. Zisserman, "Deep insights into convolutional networks for video recognition," *International Journal of Computer Vision*, pp. 1–18, 2019.

[5] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 1 2016.

[6] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4405–4425, 2 2017.

[7] ——, "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *2015 IEEE ICIP*, 2015, pp. 168–172.

[8] S. Bai, J. Zico Kolter, and V. Koltun, "Convolutional sequence modeling revisited," in *6th ICLR*. OpenReview.net, 2018.

[9] C. Gan, C. Sun, L. Duan, and B. Gong, "Webly-supervised video recognition by mutually voting for relevant web images and web video frames," in *ECCV*. Springer, Cham, 2016, pp. 849–866.

[10] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. d. R. Millán, and D. Roggen, "The Opportunity challenge: a benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[11] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild," in *CVPR*, 2012.

[12] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics Human Action Video Dataset," *arXiv:1705.06950*, 2017.

[13] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: a survey," *Image and Vision Computing*, vol. 60, 2017.

[14] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014, pp. 568–576.

[15] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: towards good practices for deep action recognition," in *ECCV*. Springer, Cham, 2016, pp. 20–36.

[16] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 39, no. 4, pp. 677–691, 4 2017.

[17] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *ICCV*. IEEE, 12 2015, pp. 4489–4497.

[18] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *16th ISWC*. IEEE, 2012, pp. 108–109.

[19] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *IJCAI)*. AAAI Press, 2016, pp. 1533–1540.

[20] F. M. Rueda and G. A. Fink, "Learning attribute representation for human activity recognition," in *ICPR*. IEEE, 2018, pp. 523–528.

[21] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "Innohar: a deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902, 2019.

[22] W. Sousa Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, "Human activity recognition using inertial sensors in a smartphone: An overview," *Sensors*, vol. 19, no. 14, p. 3213, 2019.

[23] S. Stein and S. J. Mckenna, "Combining embedded accelerometers with computer vision for recognizing food preparation activities," in *UbiComp*. Zurich, Switzerland: ACM, 2013.

[24] C. Chen, R. Jafari, and N. Kehtarnavaz, "A real-time human action recognition system using depth and inertial sensor fusion," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773–781, 2 2016.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE CVPR*. Las Vegas, NV, USA: IEEE, 2016, pp. 2818–2826.

[26] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *23rd ICM*. ACM, 2015, pp. 461–470.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[28] M. Sundermeyer, H. Ney, and R. Schluter, "From feedforward torRecurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. on Audio, Speech, and Lang. Proc.*, vol. 23, no. 3, pp. 517–529, 2015.

[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS*, 2014.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 12 2015.

[31] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *CVPR*. IEEE, 6 2016, pp. 1933–1941.

[32] C. Zach, T. Pock, and H. Bischof, "A duality based approach for real-time TV-L 1 optical flow," in *Patt. Recog. Symp.*, 2007, pp. 214–223.

[33] G. Farnebäck, "Two-Frame Motion Estimation Based on Polynomial Expansion," in *Scandinavian Conference on Image Analysis (SCIA)*. Halmstad, Sweden: Springer, Berlin, Heidelberg, 2003, pp. 363–370.

[34] M. C. Mukkamala and M. Hein, "Variants of RMSProp and Adagrad with logarithmic regret bounds," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. Sydney, NSW, Australia: ACM, 2017, pp. 2545–2553.

[35] K. Li, X. Zhao, J. Bian, and M. Tan, "Sequential learning for multimodal 3D human activity recognition with long short-term memory," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*. Takamatsu, Japan: IEEE, 8 2017, pp. 1556–1561.

[36] P. Vargas, Y. Fernaeus, M. Lim, S. Enz, W. Ho, M. Jacobson, and R. Aylett, "Advocating an ethical memory model for artificial companions from a human-centred perspective," *AI Society*, vol. 26, pp. 329–337, 2011.

[37] B. V. Ferreira, E. Carvalho, M. R. Ferreira, P. A. Vargas, J. Ueyama, and G. Pessin, "Exploiting the use of convolutional neural networks for localization in indoor environments," *Applied Artificial Intelligence*, vol. 31, no. 3, pp. 279–287, 2017.

[38] S. Enz, M. Diruf, C. Spielhagen, C. Zoll, and P. A. Vargas, "The social role of robots in the future—explorative measurement of hopes and fears," *Int J of Soc Robotics*, vol. 3, no. 263, 2011.

[39] C. Rizzi, C. G. Johnson, F. Fabris, and P. A. Vargas, "A situation-aware fear learning (safel) model for robots," *Neurocomputing*, vol. 221, pp. 32 – 47, 2017.