Improving Human Activity Recognition Integrating LSTM with Different Data Sources: Features, Object Detection and Skeleton Tracking

JAIME DUQUE DOMINGO\textsuperscript{1}, JAIME GÓMEZ-GARCÍA-BERMEJO\textsuperscript{1, 2}, AND EDUARDO ZALAMA\textsuperscript{1, 2}

\textsuperscript{1}CARTIF Foundation, División de Sistemas Industriales y Digitales, Parque Tecnológico de Borecillo, 47151 Valladolid, Spain
\textsuperscript{2}ITAP-DISA, University of Valladolid, C/Doctor Mergelina s/n, 47011 Valladolid, Spain

Corresponding author: Jaime Duque Domingo (e-mail: jaiduq@cartif.es).

ABSTRACT Over the past few years, technologies in the field of computer vision have greatly advanced. The use of deep neural networks, together with the development of computing capabilities, has made it possible to solve problems of great interest to society. In this work, we focus on one such problem that has seen a great development, the recognition of actions in live videos. Although the problem has been oriented in different ways in the literature, we have focused on indoor residential environments, such as a house or a nursing home. Our system can be used to understand what actions a person or group of people are carrying out. Two of the approaches used to solve the problem have been 3D convolution networks and recurrent networks. In our case, we have created a model that accurately combines several recurrent networks with processed data from different techniques: image feature extraction, object detection and people’s skeletons. The need to integrate these three techniques arises from the search to improve the detection of certain actions by taking advantage of the best recognition offered by each of the methods. In a complete experimentation, where several techniques have been evaluated against different datasets, the classification of the actions has been improved with respect to the existing models.

INDEX TERMS Human Activity Recognition, Recurrent Neural Network, LSTM, YOLO, OpenPose, LRCN

I. INTRODUCTION

THE recognition of actions in videos represents one of the most important current topics in Artificial Intelligence (AI) and Computer Vision (CV). Knowing what action is being carried out in a live-video has many possibilities, from the creation of smart environments to the development of security systems. Although the literature is extensive, only a limited number of works focus on the detection of actions in indoor environments, such as a nursing home or a house, in spite of its increasing interest for robotics and other fields. In our case, this need is key, since our next focus is to integrate our system into a social robotics scenario where robots can help people.

We present a novel human action recognition system that accurately combines frame features obtained by convolution layers with object coordinates and skeleton articulations. To the best of our knowledge, this technique has not been explored as a whole in the literature. Some works have used the 2D coordinates of people’s joints but have not been able to integrate them with a model that also combines the features of the convolutional part and the objects. As we will see in the paper, our experiments have shown that this information contributes significantly to the improvement of results.

Starting from the need to have a large dataset, it is important to mention that in some cases, years ago, datasets could include up to 7,000 videos \footnote{\url{[1]}}. However, in our own experience, that number is too small when the number of action classes is large. From our point of view, we consider that a number of 800 to 1,000 videos per class would be recommended to obtain realistic results. Fortunately, there has recently been a great development in the datasets suited to testing the performances of the different proposed architectures. Nowadays, it is possible to find datasets with many more videos, developed mainly in collaborative social
environments (crowdsource workers), to better test the architectures. Another important aspect to take into account is the required computing effort. Until recent years, it was not possible to evaluate large models against a high number of videos. Moreover, deep neural networks provide excellent results, but require expensive training and powerful equipment or cloud processing.

In our architecture, we use Recurrent Neural Networks (RNN), specifically Long Short-Term Memory (LSTM) networks [2], to process video sequences. Recognition is improved by processing data upon different technologies: image feature extraction, object detection and people skeleton tracking. The proposed architecture has been verified with several public datasets and improves the results obtained in the state of the art in the field of recognition of human activities in video sequences. The novelty of the architecture is that it combines into a single neural network model movement information of the skeleton of the people involved, the characterization of the objects that appear in the scene and the extraction of characteristics from the sequences of images.

The LRCN architecture for visual recognition and description had combined convolutional layers and long-range temporal recursion in previous works [3]. It integrates LSTM with image feature extraction. This model offered good results in detecting actions, given that image features represent prominent image elements. After successive layers of a convolution network, a feature vector is obtained that allows a classification network to be applied in order to obtain a certain class. However, this basic model can be enhanced by adding recent object and skeleton extraction techniques that have been greatly developed over the years. So, for example, when a person is drinking a cup of coffee, a person standing next to a cup object reinforces the idea that the action is someone who is drinking. Or when a person is helping another one to walk, there are usually two skeletons close together with a particular configuration of the respective joints. All this information has been integrated within the same model in the proposed approach, leading to significant improvements on the existing results.

Our architecture has been evaluated against different public datasets. On the one hand, some experiments have been carried out with STAIR dataset [4], including 64,282 and 78 classes. We present all the training data and confusion matrices in order to show which actions were not correctly classified in LRCN and have been improved with the new architecture, as well as all the performance evaluation values of the model. On the other hand, the system has been evaluated against NTU-RGB-D [5] and NTU-RGB-D 120 [6] datasets. NTU-RGB-D includes 56,880 elements and 60 classes, while NTU-RGB-D 120 extends NTU-RGB-D to 120 classes with 114,480 elements. These evaluations have allowed us to progress towards application in realistic situations.

The present paper is structured as follows: Section [II] explores the state-of-art of the technologies considered. Section [III] presents the description of the proposed method. In Section [IV] the different experiments and results obtained with the system are reported. An overall discussion on the obtained results is set out. Finally, Section [V] notes the advantages and limitations of the presented system and suggests future developments.

II. OVERVIEW OF RELATED WORK

Human Activity Recognition (HAR) has been an important field of research over recent years. The recognition of activities or actions can be useful to improve the experience of people in smart spaces [7], to monitor the health of a person [8] or groups in risk situations, such as elderly people [9], or even to be an input stimulus for the perception of a robot interacting with humans. An action is the most elementary human-surrounding interaction with a meaning [10].

The HAR process can be achieved using either on-body sensors or non-intrusive methods, such as computer vision [11]. Regarding on-body sensors, this process can use specific sensors or complete devices that include them, such as smartphones. In the first case, of on-body sensors, the authors of [12] developed a system that detected the activity with an accuracy of 99.89% using just two on-body sensors (chest and ankle). They detected 4 different activities using nine recurrent LSTM layers from the “Localization Data for Posture Reconstruction” dataset [13]. Non-intrusive methods are mainly based on computer vision. Unlike the intrusive methods presented above, the current approach does not require people to wear a specific powered device or sensor, and take advantage of additional information on objects near the user, therefore it can be more easily deployed and escalated to several users and identify more activities. Moreover, the determination of the user joints is done automatically without requiring inertial sensors or external marks such as stickers. The authors of [14] have recently presented a survey where HAR methods are classified according to feature extraction, recognition process, the source of the input data or the machine learning supervision level.

HAR processing follows a sequence of steps: Preprocessing, including Background construction and Foreground extraction; Feature extraction, including Global and Local descriptors (e.g., SIFT) or Semantic representations (e.g., Pose estimation); and Learning process [11]. An approach to the problem of HAR is the use of semantic features. In [15], the authors presented a survey of semantic HAR methods. According to the authors, semantics make the recognition task more reliable, especially when the same actions look visually different due to the variety of action executions. The semantic space includes the semantic features of an action such as the human body (pose and poselet), attributes, related objects and scene context. As an example, given a human action, there may be objects related to that action (e.g., a phone close to the face of a person could be related to the action of phoning). The authors established four groups of activities: atomic actions, people interactions, human-object interactions and group activities. A semantic approach to the HAR problem was published by [16], who...
proposed a hybrid framework between knowledge-driven and probabilistic-driven methods for event representation and recognition, separating the semantic modeling from raw sensor data by using an intermediate semantic representation, namely concepts.

An important step during the recognition process is the classification of the actions, usually carried out by supervised methods, which require training based on actions already recorded and catalogued. Some supervised methods are: Support Vector Machines (SVM), which uses hyperplanes and transformations to separate classes [17]; Hidden Markov Models (HMM), which allow the segmentation problem of the activities to be solved, creating Markov chains which are finite state automaton with a probability value on each arc [19]–[21]. Artificial Neural Networks (ANN), inspired in the human brain, are able to classify using non-linear discriminants and are able to give an approximation of non-linear functions for regression. The output of the Multi-Layer Perceptron (MLP) is the linear combination of the non-linear basis function values given by the hidden units [22]. Deep Neural Networks (DNNs) are a subtype of ANN composed of several different layers which produce different computations across the network on the input data. The authors of [9] have recently presented a system for monitoring elderly people at home that makes use of a Faster R-CNN [23] (Regions with CNN features) to detect people and DeepHAR [24], [25] (Deep Human Action Recognition) to detect the actions. In [26], the authors have recently published a relevant work that integrates depth-based 3-channel Motion History Images (MHI) with local spatial and temporal patterns obtained from skeleton graphs. MHI compress a sequence of motion into a single image. In addition, that work proposes a semantic approach where the object/action dependency is also considered. Finally, the authors of [27] have shown that the use of pose data to estimate people’s action, and more specifically using the STAIR dataset, allows obtaining a validation accuracy of 82.9% using LSTM. The authors selected 3 classes for the experiment, writing, reading newspaper and bowing, and cut 1 second of each video (10 frames). They evaluated different techniques, such as SVM, Random Forest or LSTM. In addition, they proposed to process the frames in edge computing, to send only the pose data and perform the processing in the cloud. In our approach, there are different 2D streams that are independently connected to LSTM layers. LRCN is used in conjunction with object and skeleton detection to significantly improve the results. Moreover, our system has been evaluated against a much larger set of actions (78 for STAIR, 60 for NTU-RGB-D and 120 for NTU-RGB-D-120. Additionally, our system could also work in a hybrid environment, with edge (frame processing) and cloud computing (action recognition).

Among the methods based on neural networks, two types of recently used architectures should be highlighted. The first consists in using a LSTM network, which is capable of learning long-term order dependences in sequence prediction problems. As shown in Figure 1 there are several classes composed of different videos with $n$ frames. The first step in this architecture is to transform each frame into a feature vector using any known Convolutional Neural Network (CNN), such as Inception [28]. These CNNs have been previously trained with datasets, such as ImageNet [29], and allow feature vectors suitable for different types of images to be obtained. This process also reduces the dimensionality of the dataset. When the feature vectors have been obtained, the LSTM model is trained. Although RNN networks are commonly used in prediction (e.g., autocompleting of texts), for the action recognition problem they are used to estimate the next vector from a sequence of vectors. Such a vector is connected to a Fully-Connected layer (FC-layer) and to a categorization function (e.g., Softmax). So, in this case, the vector is used to determine the concrete action. In [30], a work was published using this architecture.

The second architecture consists in using 3D tensors that include all the frames of a video sequence. Neural networks can be trained with 2D, but also with 3D tensors. CNNs for image classification usually rely on the use of 2D convolution or pooling layers, although they can use 3D tensors for the different colors. However, it is possible to use "cubes" representing the complete sequence of a video $(\text{number of frames} \cdot \text{width} \cdot \text{height})$. The convolutions and poolings are applied against these 3D tensors. A problem with this architecture is the high computational training cost. A possible solution to this problem is preloading weights obtained by the training of a common classification CNN. These weights work well for the classification of images and are repeated in the network for the parts corresponding to each frame (considering that the cube integrates the frames). In [31], the authors presented a model named "Two-Stream Inflated 3D ConvNets" (I3D), which used this architecture. They used video frames in one stream and optical flow in the other. Each stream used a 3D network with the same architecture, based on ImageNet pre-trained Inception-v1. The authors used the Kinetics dataset [32] and averaged their predictions with the two streams at test time. In [33], the authors have recently presented a dataset named "Moments" which is composed of one million 3 second videos. The authors have combined different models of action recognition. They combine spatial information (using a ResNet 50 pretrained with ImageNet over 6 random equidistant frames of the video), spatio-temporal information (using said I3D model) and auditory information (using SoundNet). They concatenate the features from the final hidden layer of each modality and train a linear SVM to predict the moment categories (SVM). Another outstanding work using 3D CNN networks has been published by [34]. They created a system that directly learns spatio-temporal features from raw depth sequences, then computes a joint based feature vector for each sequence by taking into account the position and angle information between skeleton joints, and finally applies SVM to predict the actions. In our architecture, the 2D convolutions and LSTM layers have been used instead of 3D convolutions because, on the one hand, that approach was faster during
Several datasets have been created over the last few years concerning the experimentation and evaluation of the different methods. The selection of a realistic and sufficiently large dataset is a first requirement for our research. In fact, there are different existing datasets, but most of them simultaneously consider indoor/outdoor actions. Even within the same category, it is possible to find actions that can be carried out indoors or outdoors, such as jumping. In some very extensive datasets, such as Moments in time [33], composed of more than a million videos, there are some similar categories where two people might differ in which category to put each video. For example, the categories jumping, leaping or skipping can include a person jumping on a horse in a riding tournament, a person skipping with a rope or a person jumping from a pool diving board. The actions of such categories do not represent a significant difference according to the videos we have observed. Other categories include videos with great differences and even videos of old computer games or cartoons. A person would not normally categorize a game video as running when watching a fast-moving character on the screen. In the swimming category, it is possible to see a fish swimming, a diver and even a mermaid. UCF101 [35] is a dataset composed of 101 actions in more than 13,000 videos. It has been widely used [31], [36]–[40]. However, although this dataset is correctly catalogued, it includes many outdoor actions related to sports: horse racing, pole vault, surfing, kayaking, etc. In addition, we considered that the number of videos was too small to train our system. HMDB-51 [1] is another well-known and commonly used dataset [36]–[39]. HMDB-51 is composed of 51 actions organized in 7,000 videos. Each category includes 101 videos. An advantage of this dataset is that the tagging was validated by at least two observers. However, as with UCF101, many actions are outdoors. The Something-Something dataset [41] is a dataset composed of about 108,000 videos organized into 174 classes. It is a current but widely used dataset [42]–[45]. However, it focuses on how a person manipulates objects with their hands. Most of the actions are videos of hands doing something. This does not help us in our approach, where our ultimate goal is to recognize people doing something. The Kinetics-400/600/700 [32], [46], [47] dataset has also been widely used [48]–[51]. It is composed of between 400 and 700 classes and approximately 1,000 videos per class. As in other datasets, many actions took place outdoors or represented classes related to sports. ActivityNet [52] is another widely used dataset [37], [53], [54]. It includes 200 well-catalogued classes with 100 videos per class; although, again, the vast majority of videos correspond to outdoor actions and many are related to sports. Finally, the STAIR dataset [4], the NTU-RGB-D dataset [5] and the NTU-RGB-D 120 dataset [6] are the only ones that mainly focus on indoor actions, offering up to 100 classes (STAIR), 60 classes (NTU-RGB-D) and 120 classes (NTU-RGB-D 120), and including about 1,000 videos per class. The rest of the explored datasets have a limited number of activities in indoor environments.

As seen previously, there are many published papers that have explored action recognition using different methods. Each of them has been tested with different databases (and different scenarios, time durations, etc.) that makes it difficult to compare the results. The results, which are very different, depending on the database selected. In addition, they use different metrics and procedures to measure accuracy, such as top-1 validation accuracy, top-1 test accuracy, top-5 validation accuracy or top-5 validation test, k-fold validation, etc. In our work, we use STAIR [4], NTU-RGB-D [5] and NTU-RGB-D 120 [6] datasets, which allows us to recognize activities in the home environment for the assistance and care of elderly people by means of social robots.
III. ANALYSIS OF THE SYSTEM

A new architecture has been defined to improve the action recognition process with respect to existing approaches. It accurately combines three different models within a single deep neural network and improves the results obtained against the chosen dataset with respect to other existing models. Our system focuses on recognizing actions carried out in a residential environment, such as a home or a nursing home.

Among all the datasets explored, STAIR [4], NTU-RGB-D [5] and NTU-RGB-D 120 [6] are the only ones that mainly focus on indoor actions. Focusing our analysis on STAIR, it offers up to 100 classes and includes about 1,000 videos per class. Many STAIR videos were recorded in a social collaboration and other videos have come from portals such as YouTube. Among the initial 100,000 videos, 64,282 corresponding to those recorded in a collaborative way have been selected. The rest, from YouTube, either corresponded to actions that were not of interest in our application scenario, such as doing origami, or were not available. A list of 78 classes that represent a complete set of possible actions has been chosen (see Table 1). All the selected classes have an average of over 800 videos, which we believe is an acceptable number to train a good classification model. The authors of STAIR have distributed the dataset into training and testing. 5,866 videos, covering all possible classes and representing 9.13% of the total number, were selected for testing. This distribution allows our model to be evaluated against a test set independent from the training. In addition, for our experiments, the training set has been divided into training and validation (90% and 10% respectively, e.g., 52,575 and 5,841 videos). All datasets were previously randomly balanced in order to avoid overfitting during training.

### TABLE 1. STAIR action list

| Id | Action                          | Id | Action                          | Id | Action                          |
|----|--------------------------------|----|--------------------------------|----|--------------------------------|
| 0  | assisting in getting up         | 27 | giving massage                  | 53 | reading newspaper               |
| 1  | assisting in walking            | 28 | going out of room               | 54 | sewing                         |
| 2  | baby crying                     | 29 | going up or down stairs         | 55 | shaking hands                  |
| 3  | being angry                     | 30 | hanging out or in laundry        | 56 | shaking head                   |
| 4  | being surprised                 | 31 | sitting down                    | 57 | shuffling                      |
| 5  | bottle-feeding baby             | 32 | sleeping on bed                 | 58 | sitting down                   |
| 6  | bowing                          | 33 | smoking                         | 59 | sleeping on bed                |
| 7  | brushing teeth                  | 34 | standing on chair or table       | 60 | smoking                        |
| 8  | caressing head                  | 35 | table or stepladder             | 61 | standing on chair or table      |
| 9  | changing baby diaper            | 36 | standing up                     | 62 | table or stepladder             |
| 10 | clapping hands                  | 37 | studying                        | 63 | studying                       |
| 11 | crying                          | 38 | taking photo                    | 64 | telephoning                    |
| 12 | cutting food                    | 39 | throwing                        | 65 | throwing                       |
| 13 | cutting wood                    | 40 | throwing trash                  | 66 | washing dish                   |
| 14 | doing high five                 | 41 | washing face                    | 67 | shaving                        |
| 15 | doing paper-rock-scissors       | 42 | washing hands                   | 68 | walking with stick             |
| 16 | drinking                        | 43 | watching TV                     | 69 | walking with stick             |
| 17 | drying hair with blower         | 44 | watching TV                     | 70 | washing dish                   |
| 18 | eating meal                     | 45 | washing face                    | 71 | washing hands                  |
| 19 | eating meat                     | 46 | putting on clothes              | 72 | watching TV                    |
| 20 | eating snack                    | 47 | putting on clothes              | 73 | watching TV                    |
| 21 | eating sweets                   | 48 | putting on glasses              | 74 | watching TV                    |
| 22 | exercising                      | 49 | putting on shoes                | 75 | watching TV                    |
| 23 | feeding baby                    | 50 | putting on a tie                | 76 | wiping window                  |
| 24 | fighting                        | 51 | putting on a tie                | 77 | wiping window                  |
| 25 | folding laundry                 | 52 | reading book                    | 78 | writing                        |

Figure 2 shows the complete architecture. Our system is composed of three different submodules integrated into the same model: LRCN, object detection and skeleton tracking. Each of these submodules uses its own LSTM and the outputs of the three layers are concatenated and presented to the FC layer.

The need to integrate these three submodules arises from the search to improve the recognition of certain actions that the LRCN model is not able to appreciate. Thus, for example, in the experiments, we see how the action of a person telephoning is reinforced by finding a phone object close to the person’s head.

### A. LRCN/FEATURES SUBMODULE

A first submodule is based on LRCN [55], a model that combines features extracted from a CNN, such as Inception [28], with an LSTM layer, capable of learning long-term order dependencies in sequence prediction problems. This LSTM layer receives the output of the average pooling layer, prior to the FC layer. The features of the whole images are integrated with our system, as in other works [40]. We have used Inception V3 [56], a classification CNN that implements Inception modules, designed to reduce the computational expense of CNNs. These Inception modules work by performing a convolution on an input with three different filter sizes (1x1, 3x3, 5x5). In addition, a max pooling is applied before concatenating these parallel convolutions. Before the Full-Connected layer, an average pooling is carried out. That layer represents the set of main characteristics of the image before the classification. This is why the output of this layer is used instead of the actual output of the network.

Transfer learning has been used with Inception V3 using the weights obtained with ImageNet [29] training. The average pool vector of the Inception v3 model consists of 2048 floating values (named from here on “features vector”). In our architecture, 90 frames are processed for each video, corresponding to a 3-second sequence (with a frame rate of 30Hz). For each frame, the features vector is obtained and the LSTM receives that sequence of 90 * 2048 values. The output of the LSTM is, in turn, a vector of another 2048 values that would represent a possible value of the sequence, but in our case are concatenated with the outputs of the other submodules and connected to an FC layer to classify the video into the 78 different actions selected. Figure 3 shows the feature maps obtained by Inception V3 at different layers.

For the input, the frames are resized to fit into the net input. They are also normalized scaling the pixel values between -1 and 1, using sample-wise (by frame) (3b). The last average pool vector (see feature map (3f)) consists of 2048 floating values \( f_0, f_1, ..., f_{2047} \). Then, a window of 90 feature vectors \( F_0(f_0, f_1, ..., f_{2047}) \), \( F_1(f_0, f_1, ..., f_{2047}) \), ..., \( F_{90}(f_0, f_1, ..., f_{2047}) \) is processed by the LSTM layer to obtain a new vector of 2048 elements (LRCN\( f_0, f_1, ..., f_{2047} \)).

### B. OBJECT DETECTION SUBMODULE

Our architecture does not only use features. To improve the LRCN model, other data are considered, such as the movement of objects and people’s skeletons. The movement of people and objects in the scene is an important aspect to complement the information in the model. Thus, someone who is drinking a coffee will have a “person” object next to them.
to a “cup” that will follow a similar movement among different videos. The You Only Look Once (YOLO) algorithm [57]–[60] is a state-of-the-art, open source, real-time object detection system that uses a single CNN to detect objects in images and that has achieved some of the best results against ImageNet. YOLO returns the confidence and position of an object in an image and uses modules inspired by Inception to reduce the number of operations. One of the differences between YOLO and other networks is the design of its cost function, intended to evaluate the bounding box and confidence that an object is in a particular position of the image rather than simply a classification function of an image.

In our model, YOLO v5 has been trained against the COCO 2017 dataset [61]. The model has 87.8 million parameters and has offered superior results to those of the previous versions, proving to be much faster than other object detection models, such as EfficientDet [62]. In our experiments, YOLO v5 has managed to process at 60 Hz, although our videos are limited to 30 fps, so it has not slowed down our processes. When working indoors, we made an analysis of the 50 most commonly appearing objects in the STAIR videos and we rejected objects that practically did not appear or had been detected incorrectly (see Table 2).

The second submodule includes a vector formed by 51 objects, including 5 parameters per object: detection confidence and the bounding box positions of each object: \((x_0, y_0) - (x_1, y_1)\). If several objects of the same type appear in a frame, we only select the one with the highest confidence. For people, we select the two largest persons that appear in the image. Thus, the number of objects is 51, since a maximum of two persons are considered. In total, after processing the sequence of 90 frames with YOLO v5, we have \(90 \cdot (51 \cdot 5) \approx 90 \cdot (255)\) floating values. These values are normalized to the width and height of the image and the sequence represents the input of the second LSTM that analyzes the sequence of object movements in a scene. Figure 4 shows four frames of videos corresponding to the actions: Sleeping on bed (4a), Drinking (4b), Hugging (4d) and Reading book (4c). Different objects appear in the scene, but we can appreciate the relationship between the persons and the objects that are important for the scene: person with bed in Sleeping on bed, person with cup in Drinking, two persons together in Hugging and person with book in Reading book.

C. SKELETON TRACKING SUBMODULE

The third submodule of the system is the one corresponding to the skeletons of the largest people that appear in the scene. In our model, the 18 coordinates of the skeletons returned by OpenPose [63] are used. OpenPose pipeline is composed of a CNN trained using the COCO and MPII datasets [64] that returns a heatmap and the Partial Affinity Fields (PAFs) [65], used to obtain the location of the part candidates from the heatmaps, and an assignment problem to connect the body parts of each person. As we want to see the movement of one or two persons along the 90 frame sequence, we look for the skeletons of one or two persons along the sequence and extract the bounding box that would cover all the skeletons along the whole sequence. This allows us to make the system independent of the translation of the scene. For example, it

![FIGURE 2. New architecture for the action recognition process.](image)
does not matter if a person entering a room is situated to the left or right of a video, since we will keep the bounding box where the person moves. To avoid null coordinates, 1% of the width and height of the image is subtracted/added to the bounding box. Thus, no skeleton coordinates will be at the bounding box limits. Finally, the skeleton joints are normalized against the width/height of the bounding box. The input of the LSTM is composed of \(90 \cdot (2 \cdot 18)\) floating values, 90 frames and 18 joints \((x,y)\). The output of the LSTM is a vector of \(2 \cdot 18\) values that are concatenated with the output of the preceding submodules.

Figure 5 shows three frames of videos corresponding to the actions: **Assisting in walking** (5a), **Assisting in getting up** (5c) and **Clapping hands** (5e). In these cases, the skeletons clearly show the actions carried out (5b, 5d and 5f respectively). Two people standing together who walk slowly indicates the action **Assisting in walking**, a person who approaches the hands in a video sequence represents the action **Clapping hands**, while a person who helps another who is lying shows the action **Assisting in getting up**. The points shown are those given by Openpose that are used in the Skeleton tracking submodule. These points, \(P_{i,j}\), represent the coordinates \((x, y)\) in the image for a person \(i\) and a key point \(j\), where the value of \(j\) ranges from 0 to 17.

### D. INTEGRATING LRCN, OBJECT DETECTION AND SKELETON TRACKING

The model input receives a fixed sequence of 3 seconds long (90 frames), covering the duration of all activities. It allows working in real time by analyzing the last 3 seconds of video. The frames are processed by each submodel: Inception v3, OpenPose and YOLO to obtain a set of features (see Input in Table 3). As previously explained, the features of great interest are filtered and normalized (e.g., OpenPose coordinates are normalized with respect to the bounding box obtained from the frame sequence). They include 51 objects (YOLO), the coordinates of 2 persons (OpenPose), and the feature vector of \(2048\) elements obtained using the convolutional net. The produced features are processed by the respective three LSTMs. These layers analyze the temporal sequence of data by reducing the dimensionality to the features (see Output in Table 3). The output of the 3 LSTMs: \(F(f_0, ..., f_{2047})\), \(Object_0(\text{conf}, x_0, y_0, x_1, y_1)\), \(..., Object_{50}(\text{conf}, x_0, y_0, x_1, y_1)\), \(Person_0(x_0, y_0, ..., x_{17}, y_{17})\), and \(Person_1(x_0, y_0, ..., x_{17}, y_{17})\) are concatenated and connected to an FC layer and a classification layer. In addition, we use batch normalization of the FC layer to avoid out-of-range values (outliers), and dropout before the LSTMs and in the FC-layer to avoid overfitting.

Let \(I_f\) be the tensor produced by Inception v3 for the frame
This filtering process produces a vector of 51 objects, \( O_f = [\text{Object}_0(\text{conf}, x_0, y_0, x_1, y_1), \ldots] \) for a frame \( f \). The second LSTM processes a sequence of 90 vectors, previously transformed into a tensor with dimensions \((90, 51 \cdot 5)\), producing a single output tensor with dimension \((51 \cdot 5)\), \( O = \text{LSTM}_2(\text{tensor}(O_0, \ldots, O_{89})) \).

OpenPose’s output is a tensor \( J_f \) with dimensions \((p, j, d)\), where \( p \) represents the number of detected people, \( j \) the number of joints and \( d \) the data \([x, y, \text{conf}]\), where \((x, y)\) are the coordinates of a joint and \text{conf} the confidence. This tensor is normalized and filtered as explained before to obtain a vector, \( P_f \), with the skeletons of two people: \( P_f = [\text{Person}_0(x_0, y_0, \ldots, x_{17}, y_{17}), \text{Person}_1(x_0, y_0, \ldots, x_{17}, y_{17})] \). The third LSTM processes a sequence of 90 vectors, previously transformed into a tensor with dimensions \((90, 2 \cdot 18 \cdot 2)\), producing a single output tensor with dimension \((2 \cdot 18 \cdot 2)\), \( P = \text{LSTM}_3(\text{tensor}(P_0, \ldots, P_{89})) \).

The three output tensors are concatenated \((F || Y || P)\) and then connected to a FC layer and a classification output.
Equation (1) shows the complete process before the FC layer.

\[
\begin{align*}
F &= \text{LSTM}_1(\text{tensor}(F_0, \ldots, F_{89})) \\
  &= \text{LSTM}_1(\text{tensor}(\text{avg}(I_0), \ldots, \text{avg}(I_{89}))) \\
Y &= \text{LSTM}_2(\text{tensor}(O_0, \ldots, O_{89})) \\
  &= \text{LSTM}_2(\text{tensor}(\text{norm}(\text{fil}(Y_0)), \ldots, \text{norm}(\text{fil}(Y_{89})))) \\
P &= \text{LSTM}_3(\text{tensor}(P_0, \ldots, P_{89})) \\
  &= \text{LSTM}_3(\text{tensor}(\text{norm}(\text{fil}(J_0)), \ldots, \text{norm}(\text{fil}(J_{89})))) \\
\text{out} &= F \| Y \| P
\end{align*}
\]

(1)

The model input is a fixed sequence of 90 frames of a video.

| Model       | Input                                                                 | Output                                                                 |
|-------------|----------------------------------------------------------------------|------------------------------------------------------------------------|
| Features    | (*) for each frame [0 ... 89] (f_0, f_1, ..., f_89)                  | (f_0, f_1, ..., f_89)                                                 |
| YOLO objects| Object\{conf, x_{fil}, y_{fil}, x_{roi}, y_{roi}\}_{0}               | Object\{conf, x_{fil}, y_{fil}, x_{roi}, y_{roi}\}_{0}               |
|             | Object\{conf, x_{roi}, y_{roi}, x_{roi}, y_{roi}\}_{1}               | Object\{conf, x_{roi}, y_{roi}, x_{roi}, y_{roi}\}_{1}               |
| OpenPose    | Person\{x_{0}, y_{0}, x_{17}, y_{17}\}_{a}                         | Person\{x_{0}, y_{0}, x_{17}, y_{17}\}_{a}                         |
|             | Person\{x_{0}, y_{0}, x_{17}, y_{17}\}_{b}                         | Person\{x_{0}, y_{0}, x_{17}, y_{17}\}_{b}                         |

TABLE 3. LSTM inputs/outputs.

Figure A.1 shows the flow of the proposed model where the outputs of the three different LSTM blocks are concatenated.

For our experiments, we have not only tested the model composed of Features and data from YOLO and OpenPose. We have also tested the three models separately: Features, YOLO and OpenPose, as well as the combination of OpenPose and YOLO. Although, apparently, the accuracy obtained from OpenPose and YOLO are lower than those produced by Features, the union of different types of independent input data improves the results. This has been proven in previous works, where the fusion of different neural networks improved the results [66].

In addition to the proposed model, a model that integrates the outputs of the 3 LSTM layers with SVM [67] has been tested. In this case, the weights and parameters used for the LSTM layers have been taken from the feature network and YOLO-OP trainings. This architecture is shown in Figure 6.

IV. Experiments and Results Discussion

Several public datasets were considered before the experiments. Among them, STAIR [4] was mainly focused on indoor actions. This dataset offers up to 100 classes and includes about 1,000 videos per class (100,000 videos in total). Many STAIR videos were recorded by crowdsource workers or stored on streaming platforms (YouTube). As previously explained, among the complete dataset, only 64,282 videos corresponding to those recorded in a collaborative way were selected. A list of 78 classes that represent a complete set of possible actions was chosen (see previous Table 1). The dataset was randomly divided into training, validation and testing, covering all possible classes in each of them. This division is: Training, 81.79% (52,575 videos); Validation, 9.09% (5,841 videos); and Testing, 9.13% (5,866 videos). These three datasets were randomly balanced to avoid overfitting during training. The test dataset is completely independent of the training process.

The input of the model was composed of features, 51 objects (including 2 people) and 2 skeletons with 18 joints each. Among the different hyper-parameters, only one FC layer was enough to get good results. The hidden neurons of this layer were different depending on the inputs and outputs of the model. The LRCN model (Features) was trained with 512 hidden neurons, while the Features+YOLO+OP model was trained with 594 and the YOLO+OP model with 296 neurons. Dropout was used in different points of the model: after the input layers, a dropout of 0.05%; after the LSTM layers, a dropout of 0.5%; and after the FC layer, another dropout of 0.5% was included. Following a grid search approach, the training sessions were carried out with different parameters values. Among these parameters, an Adam optimizer with a learning rate of 10^{-5} and a decay rate of 10^{-6} were selected. Two metrics were added during the training: accuracy and top 5 categorical accuracy. The sessions used different number of epochs (50, 100, 200) and, in some cases, early-stopping. The best results were achieved with 200 epochs, but in all the different trainings, with the same or different number of epochs (50, 100, 200) and validation sets, our model considerably improved on the values of LRCN. Table 4 shows the obtained results of the different models during a 200 epoch training.

Although a priori the results of YOLO and OpenPose may not seem as good as those of Features, we can see that the integration of OpenPose and YOLO improves over their separate use. In the same way, the integration of OpenPose, YOLO and Features improves the recognition of actions by 1.5% (0.8650 against 0.8500). Figures 7a and 7b show the accuracy and loss for the training and validation set during the different epochs of the model training that integrates Features, OpenPose and YOLO. From about 100 epochs on, the values are stable, so the training can be interrupted by early stopping. There is some difference between the training and validation results due to the difference between the videos of the two sets, training and validation. This overfitting was reduced by dropout, as explained above.

A significant computing effort was required to train our models. We used an i9-10900k server with 128GB RAM and 2 GPU RTX-3090 with 24GB GDDR6X. This server required 29,519 seconds (8.19 hours) to train the Features+YOLO+OP model and 29,504 seconds (8.19 hours) to train the Features model. It is worth noting that the features, objects and skeletons had been previously extracted.

Although all evaluations have been obtained at the individual action level, in order to be able to show confusion matrices with a reduced number of classes, the activities have been grouped into similar categories, as shown in Table 5. Table A.1 shows the confusion matrix of the model that...
combines Features, OpenPose and YOLO. Table A.2 shows the confusion matrix of the model that only uses Features. Table A.3 shows the improvement in the YOLO-OP-Features model against the Features model.

The elements in green on the improvement matrix diagonal correspond to improvements of the integrated model (OP + YOLO + FEAT) with respect to the model that only used features. In the same way, the elements that are in red but are not in the diagonal, are also improvements since they correct the error that occurred in the incorrect detections of the model that only used features. Most elements in the diagonal are improved, including some particular cases such as Standing/Smoking actions, improved by more than 11% (from about 76% to 87%).

Although the matrix reflects grouped values, an analysis of individual actions shows significant improvements, such as the one produced in the action “Using device” 65 (telephoneing) and 68 (using computer). They are improved by 0.21% and 0.28%, respectively, in YOLO+OP+Features with respect to the Features model. This is due to the fact that more elements are involved in the decision, in particular the knowledge that a person is in a relative position with respect to some objects: phone and computer, respectively. In the same way, most errors are produced in the actions which are really difficult to distinguish, even for the human eye. As an example, the actions reading book, reading newspaper, studying or even writing are very similar and it is very difficult to recognize the correct action.

Our model has been evaluated against different metrics, as shown in Table 6. A Precision-Recall curve with the average precision score, micro-averaged over all classes, is displayed in Figure 8. In addition, a Multi-Class Precision-Recall curve, with all classes, is displayed in Figure 9. The evaluation of the model shows a good performance, leading the AUC (Area Under the Curve) close to 1 in all cases.

Table 6. Evaluation of the model with different metrics (%).

In addition to our model training based on a pure Deep Neural Network, a comparison with SVM was carried out (see Figure 6). After an exhaustive grid search to obtain the parameters ($C = 1000, \gamma = 0.0001$), SVM was
TABLE 5. Action grouping to show confusion matrices.

| Group                        | Action                                                                                           |
|------------------------------|--------------------------------------------------------------------------------------------------|
| Assisting                    | Assisting in getting up, assisting in walking, holding someone, holding someone on back, passing something |
| Baby crawling/crying         | Baby crawling, baby crying                                                                      |
| Drinking                     | Drinking, Pouring tea or coffee                                                                  |
| Eating                       | Cutting food, eating meal, eating snack                                                          |
| Emotional actions            | Being angry, being surprised, crying                                                             |
| Exercising                   | Bowing, clapping hands, exercising, fighting, shaking hands, shaking head                        |
| Feeding/assisting baby       | Bottle feeding baby, changing baby diaper, feeding baby                                          |
| Housework                    | Folding laundry, gardening, hanging out or taking in laundry, housecleaning, ironing, polishing shoe, wiping window |
| Massage                      | Caressing head, giving massage                                                                  |
| Moving                       | Entering room, going out of room, going up or downstairs, walking with stick                     |
| Opening something            | opening or closing container, opening refrigerator door                                           |
| Personal care                | Brushing teeth, drying hair with blower, gaging, manicuring, shaving                             |
| Playing something            | Doing high five, doing paper rock scissors, playing board game, playing computer game, using computer |
| Putting clothes              | Putting off cloth, putting on cloth                                                              |
| Reading/Writing              | Reading book, reading newspaper, studying, writing                                              |
| Resting/Watching TV          | Jumping on sofa or bed, lying on floor, sitting down, sleeping on bed, watching TV               |
| Sewing/Knitting              | Knitting or stitching, sewing                                                                    |
| Standing/Smoking             | Smoking, standing on chair or table or stepladder, standing up                                  |
| Throwing something           | Throwing, throwing trash                                                                        |
| Using device                 | Listening to music with headphones, operating remote control, taking photo, telephoning         |
| Washing something            | Washing dish, washing face, washing hands                                                       |
| Wearing something            | Wearing glass, wearing shoes, wearing tie                                                       |

Our method can work in real-time given that it does not take much longer than the feature-based method. Although it requires more time-consuming training, the system takes about the same time as the LRCN in real-time inference. This is because each of the three channels is distributed in independent areas of the GPU, parallelizing the convolutional part, pose fetching and object fetching. Finally, the three flows are integrated in the last layer, but this layer is very fast, practically negligible in time. Each received video frame is processed to obtain features, objects and skeletons, and saved in a sliding window that behaves as a FIFO stack of 90 elements, as shown in Figure 10 which is used entirely for input to the model.

Our method can work in real-time given that it does not take much longer than the feature-based method. Although it requires more time-consuming training, the system takes about the same time as the LRCN in real-time inference. This is because each of the three channels is distributed in independent areas of the GPU, parallelizing the convolutional part, pose fetching and object fetching. Finally, the three flows are integrated in the last layer, but this layer is very fast, practically negligible in time. Each received video frame is processed to obtain features, objects and skeletons, and saved in a sliding window that behaves as a FIFO stack of 90 elements, as shown in Figure 10 which is used entirely for input to the model.
Finally, Table 8 shows the comparison of the test accuracy between our method and current state methods applied to the STAIR dataset. As explained in the Overview of related work (Section II), the Two-stream CNN method uses video frames in one stream and optical flow in the second one. Both networks are 3D convolutional networks. Although LRCN produces better results (0.85%) than Two-stream CNN (0.737%), 3DCNN (0.765%) and even YOLO+OP (0.65%), the integration of YOLO+OP+Features (LRCN) improves the accuracy to 0.865%. In addition, the approach using SVM with our model has shown higher results (0.873%). Our model, using either an FC-layer or SVM, considerably improves the existing methods. Regarding the comparison with [27], there are some differences. On the one hand, the authors use 3 activities and 10 frames per video (1 sec), while we use 78 activities and 90 frames. In addition, they offer validation results while we present test results. Although we have also observed that OpenPose contributes to discern what action the user performs (40.60% over test for OpenPose), it is through integration with the joint model that it shows its potential (87.35% over test).

In addition to the STAIR dataset [4], two additional experiments with NTU-RGB-D [5] and NTU-RGB-D 120 [6] have been conducted. NTU-RGB-D dataset consists of 60 classes with 56,880 elements including RGB videos, depth sequences, skeleton data (3D locations of 25 major body joints) and infrared frames. The data was captured from 40 different human subjects, using multiple Microsoft Kinect v2 devices. NTU-RGB-D 120 extends NTU-RGB-D dataset to 120 classes with 114,480 elements. NTU-RGB-D is a challenging dataset for our method since our system only uses RGB images. Our algorithm obtains the 2D coordinates of the skeletons using OpenPose (18 joints) and we do not need special devices, such as the Kinect. In addition, NTU-RGB-D has a lot of variability in the length of the activities and people are far away from the camera. In the case of these experiments, two additional dense layers with 594 and 512.
neurons respectively and L2 regularization have been added to reduce overfitting in both models: LRCN and LRCN + OpenPose + YOLO. In [5], different methods were evaluated against NTU-RGB-D using 3D information, ranging from 30.56% to 62.93% class accuracy. In our case, taking into account that we only use RGB videos, our method has obtained 52.11% accuracy for LRCN + OpenPose + YOLO and 40.57% for the classic LRCN, showing the effectiveness of the integration of the three techniques proposed in our approach. The top-5 accuracy has been 86.11% for LRCN + OpenPose + YOLO and 77.49% for the classic LRCN.

Regarding the NTU-RGB-D 120 dataset, several state-of-the-art action recognition methods were evaluated in [6] using 3D information, ranging from 26.30% to 66.90% class accuracy. In our case, only using RGB videos, the method has obtained 47.00% accuracy for LRCN + OpenPose + YOLO and 35.96% for the classic LRCN, showing again the effectiveness of the integration of the three techniques proposed in our approach. The top-5 accuracy has been 83.19% for LRCN + OpenPose + YOLO and 74.73% for LRCN. Figure 11 shows the accuracy and loss curves for the NTU-RGB-D 120 training. Since the 75 epoch, the results have been stable. In this case, there is some overfitting due to the high difference of the videos between training and validation, and the effect of dropout has been lower than with STAIR.

Table 9 shows a comparison between some of them and the evaluation of our method. In the description column, a brief explanation of the method is shown. The models compared by the authors mainly used RGB frame data, 3D skeletons and depth map data. In the Two-stream VGG method [68], the authors used a two-stream ensemble model with video frames in one stream and optical flow in the other. To perform the classification, they averaged the output of the individual classifiers. Despite producing better classification results with this RGB-based method in the NTU-RGB-D dataset, our model outperforms the results of the same against the STAIR dataset (73.7% for Two-stream CNN vs 87.3% for LRCN+OpenPose+YOLO).

The results obtained with the STAIR dataset are significantly better than those obtained with NTU-RGB-D and NTU-RGB-D 120 for two main reasons. On the one hand, in NTU-RGB-D and NTU-RGB-D 120, the people appearing...
in the RGB videos are at a much greater distance from the camera, which considerably affects the convolution part. The people and objects that appear are very small in relation to the scene and that makes that when the image enters the model, the part of interest is very small. The convolution network used in our model operates with an input size of 299x299 pixels, compressing the original frames to this size. Therefore, people are also reduced by occupying a very small percentage of these input pixels. On the other hand, NTU-RGB-D and NTU-RGB-D 120 videos have a highly variable duration, in contrast to STAIR, with videos of a more similar duration. This significant difference means that the action is not bounded and is difficult to recognize in a fixed number of frames.

3D convolutional networks (3D-C3D) can be integrated with our system by substituting the convolutional and LSTM part relative to the CNN features. As 3D itself obtains a tensor of spatio-temporal features prior to classification for an entire video, it is not necessary to use LSTM. However, this tensor must be adapted and concatenated with the output of the OpenPose and YOLO-LSTMs. As the results of this technology are worse than those returned by LRCN (see Table 5), the experiments focused on LRCN.

Our future line of research will involve the use of deeper convolution networks with better image feature extraction capabilities. For example, the replacement of the Inception v3 [56] module with ResNeXt-101 [70] will help us to improve the results of the model.

The code showing how the model is defined and trained is available on the Internet at the URL: https://github.com/jaiduqdom/LRCN_OP_YOLO.git

V. CONCLUSIONS

Our paper presents the novelty of accurately combining three different types of information extracted from RGB video and how the selection and normalization of the necessary elements has been performed. Such selection and normalization has been studied in order to improve the problem of action classification in videos. It is mainly focused on scenes carried out in residential environments, such as houses or nursing homes. Different technologies have been analyzed and evaluated in this research.

The recognition of indoor activities is a key element for analysis and decision making in smart environments and social robotics, particularly when interacting with elderly people. Activity recognition from video sequences is complex as there can be very diverse scenes that are difficult to recognize, even for the human eye. In recent years, with the advance of CNNs, the recognition rate has greatly improved. Several datasets have appeared and the authors use different metrics to evaluate their models, which makes it difficult to compare results. In our work, different existing datasets have been evaluated: STAIR [4], NTU-RGB-D [5] and NTU-RGB-D 120 [6]. STAIR, NTU-RGB-D and NTU-RGB-D 120 were chosen as they are oriented towards indoor activity recognition and highlighting the advantages they bring with respect to the rest (significant number of samples, numerous indoor categories, correct division of activities and training/test sets). Our system integrates information from features, environmental objects and people’s skeletons to infer the activity performed in indoor environments. It has been presented in two different ways: one that integrates the three data sources into a single deep neural network and another one that uses SVM for its final classification. It makes use of recurrent networks, specifically LSTM, to gather the data of the 90 frames that make up a 3-second video sequence.

The system has been tested against STAIR dataset [4] where we have selected 78 classes and 64,282 videos. Our architecture has shown that the results obtained by other models, which use exclusively the features of the different frames of the video, have been improved. Although LRCN produces better results (0.85%) than YOLO+OP (0.65%), the combination of YOLO+OP+Features (LRCN) improves the accuracy by 0.865%. The results obtained significantly improve the previous results obtained on STAIR: Two-stream CNN (0.737%), 3DCNN (0.765%). In addition, a different approach has been evaluated using SVM with our model and promising results have been obtained (0.873%). Additionally, our method has been tested with NTU-RGB-D [5] and NTU-RGB-D 120 [6], datasets that provide RGB videos and depth information, including 3D skeletons. The results have shown that, with these datasets, our integrated model greatly improves the results compared to the classical LRCN technique (52.11% vs. 40.57% with NTU-RGB-D, and 47.00% vs. 35.96% with NTU-RGB-D 120). Our model requires the processing of different types of information; however, due to the advance in hardware, it is possible to operate in real time supported by the use of several GPUs. Our trainings have been carried out in a server with two GPUs and can work in real-time.

Our future projects will involve the use of deeper convolution networks with better image feature extraction capabilities, and the integration of our action recognition system in a residential environment with humanoid robots. In a social robotics environment, knowing what action people are taking allows the robot’s actions and movements to be improved for a more intelligent and proactive behavior. In addition, we are working on the integration of this system with elderly people [71], where we can control specific actions that trigger alarms.

| Method       | Description                          | Accuracy  |
|--------------|--------------------------------------|-----------|
| 2-stream VGG | Ensemble of convolutional net and optical flow with RGB | 54.905%   |
| Integrated architecture (ours) with RGB videos | LRCN + OpenPose + YOLO | 47.00%    |
| Soft RNN     | MSRRN with RGB-D features            | 44.90%    |
| LRCN         | LSTM with RGB videos                 | 35.96%    |
| Part-Aware LSTM | LSTM from body parts of 3D skeleton data | 26.305%   |

TABLE 9. Comparison with other methods based on RGB video processing used in the NTU-RGB-D 120 dataset. [5]
APPENDIX. FIGURES AND TABLES

FIGURE A.1. Proposed YOLO-OP-Features architecture.

TABLE A.1. Confusion matrix of the Features+OpenPose+YOLO model.
TABLE A.2. Confusion matrix of the Features model.

| Actual | Predicted |
|--------|-----------|
| Assisting | Baby crawling/crying |
| Assisting | Drinking |
| Assisting | Eating |
| Assisting | Emotional actions |
| Assisting | Exercising |
| Assisting | Feeding/assisting baby |
| Assisting | Housework |
| Assisting | Massage |
| Assisting | Moving |
| Assisting | Opening something |
| Assisting | Personal care |
| Assisting | Playing something |
| Assisting | Putting clothes |
| Assisting | Reading/Writing |
| Assisting | Resting |
| Assisting | Sewing/Knitting |
| Assisting | Standing/Sitdowning |
| Assisting | Throwing something |
| Assisting | Using device |
| Assisting | Wearing something |

TABLE A.3. Improvement matrix of the YOLO-OP-Features model against the Features model.
FUNDING
The present research has been partially financed by the "Programa Retos Investigación del Ministerio de Ciencia, Innovación y Universidades (Ref. RTI2018-096652-B-100)" and by the "Programa de Apoyo a Proyectos de Investigación de la Junta de Castilla y León (Ref. VA233P18)", cofinanced with FEDER funds.

REFERENCES
[1] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In 2011 International Conference on Computer Vision, pages 2556–2563. IEEE, 2011.
[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
[3] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2625–2634, 2015.
[4] Yuya Yoshikawa, Jiaqing Lin, and Akikazu Takeuchi. Stair actions: A video dataset of everyday home actions. arXiv preprint arXiv:1804.04326, 2018.
[5] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1010–1019, 2016.
[6] Jun Liu, Amir Shahroudy, Muricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. IEEE transactions on pattern analysis and machine intelligence, 42(10):2684–2701, 2019.
[7] Alexandros André Chaaraoui, Pau Climent-Pérez, and Francisco Florínez-Revuelta. A review on vision techniques applied to human behaviour analysis for assisted-living experts. Expert Systems with Applications, 39(12):10873–10888, 2012.
[8] Abdullhamit Subasi, Khoioud Keithateb, Tayeb Brahim, and Aïka Sarirete. Human activity recognition using machine learning methods in a smart healthcare environment. In Innovation in Health Informatics, pages 123–144. Elsevier, 2020.
[9] Marco Buzzelli, Alessio Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261, 2016.
[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrea Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3):211–252, 2015.
[11] Amin Ullah, Janil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Action recognition in video sequences using deep bi-directional lstm with cnns features. IEEE Access, 6:1155–1166, 2017.
[12] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6299–6308, 2017.
[13] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1725–1734, 2014.
[14] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vandrick, et al. Moments in time: Dataset and algorithm for human activity recognition using multitask deep learning. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[15] Ye Gu, Xiaofeng Ye, Weihsueh Sheng, Yongsheng Ou, and Yongqing Li. Multiple stream deep learning model for human action recognition. Image and Vision Computing, 93:103818, 2020.
[16] Chikako Takasaki, Atsuko Takefusa, Hidemoto Nakada, and Masato Oguchi. A study of action recognition using pose data toward distributed processing over edge and cloud. In 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pages 111–118. IEEE, 2019.
[17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261, 2016.
[18] Ethem Alpaydin. Introduction to machine learning. MIT press, 2020.
[19] Jaesik Chord, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vandrick, et al. Moments in time: Dataset and algorithm for human activity recognition using multitask deep learning. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[20] Kwok-Yan Lam, Efthymios Kompatsiaris, and Francois Bremond. Semantic event fusion of different visual modality concepts for activity recognition. IEEE Transactions on Multimedia, pages 1–47, 2020.
[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrea Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3):211–252, 2015.
[23] Jia Deng, Jiajun Zhan, Li Fei-Fei, Li Song, and Honglak Lee. Omniglot. In ICML, volume 20, pages 205–212, 2013.
[24] Diogo C. Luz, Vicente Moreira, and Luís M. Kratz. The kinetics dataset: High-quality ground-truth annotation for large-scale action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1725–1732, 2014.
[25] Evangelos Synchropoulos and Andriy Grabocka. Multi-task deep learning for real-time 3d human pose estimation and action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–1, 2020.
[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
[27] Matt Cutts, Alessandro Moschitti, and Sunita Sarawagi. A new model and the kinetics dataset. In proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–1, 2020.
[28] Oliver Groue, Yee Whye Teh, and Gadiel Syeda-Mahmood. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1725–1734, 2014.
[29] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vandrick, et al. Moments in time dataset: one million videos for event understanding. IEEE transactions on pattern analysis and machine intelligence, 42(2):502–508, 2019.
[30] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. A large scale dataset for 3d human activity analysis. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1010–1019, 2016.
[31] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In 2011 International Conference on Computer Vision, pages 2556–2563. IEEE, 2011.
JAIME GÓMEZ-GARCÍA-BERMEJO is Full Professor in the Department of Systems Engineering and Automation at the University of Valladolid (Spain). He holds a MEng from this University, a MSc in Image Processing from the École Nationale Supérieure de Télécommunications of Paris (France) and a PhD in Industrial Engineering from the University of Valladolid. His research career has been largely oriented to applied research and the training of researchers in technology transfer to companies, public institutions and society in general. With this spirit, in 1994 he joined the CARTIF Technology Center, a private foundation where he is Scientific Director of the Computer Vision Area. He has participated in 100+ funded research projects, most in cooperation with companies or public entities, and 125+ research contracts with companies and public entities, in many of which he has acted as principal investigator. He is also co-author of several patents and software licensed for use by important companies. He is co-author of 60+ relevant scientific papers and 120+ contributions to congresses, many of high international level. He is an evaluator of projects for diverse agencies and has been hired as an R&D Expert on numerous occasions for the evaluation and accreditation of research activities.

EDUARDO ZALAMA received the Ph.D. Degree in Control Engineering from the University of Valladolid (Spain) in 1994. He is Full Professor in the Department of Systems Engineering and Automation, of the School of Industrial Engineering at the University of Valladolid. He has been a Visiting Professor at the Universities of Boston and Carnegie Mellon (Pittsburgh). He is director of the Industrial and Digital Systems division of the Cartif Technology Center. He has leaded numerous research projects in the field of service robotics, automation and computer vision, at national and international level. He has participated in more than 125 research contracts with companies and public entities, in many of which he has acted as principal researcher. He is the author of more than 150 peer-review articles in prestigious journals and international conferences in the field of robotics and artificial vision. His line of research focuses on the development of service robots, and particularly social robots with the ability to interact with people.