IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 7, 2021, accepted October 2, 2021, date of publication October 8, 2021, date of current version October 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3118829

A New Approach for Video Action Recognition: CSP-Based Filtering for Video to Image Transformation

ITSASO RODRÍGUEZ-MORENO, JOSÉ MARÍA MARTÍNEZ-OTZETA, IZARO GOIENETXEA, IGOR RODRIGUEZ, AND BASILIO SIERRA

Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), 20018 Donostia-San Sebastián, Spain

Corresponding author: Itsaso Rodríguez-Moreno (itsaso.rodriguez@ehu.eus)

This work was supported in part by the Basque Government Research Teams under Grant IT900-16, in part by ELKARTEK 3KIA Project under Grant KK-2020/00049, in part by the Spanish Ministry of Science (MCIU), the State Research Agency (AEI), and the European Regional Development Fund (FEDER) (MCIU/AEI/FEDER, European Union (EU)) under Grant RTI2018-093337-B-I100, and in part by the Spanish Ministry of Science, Innovation and Universities under Grant FPU18/04737 (predoctoral grant).

ABSTRACT In this paper we report on the design of a pipeline involving Common Spatial Patterns (CSP), a signal processing approach commonly used in the field of electroencephalography (EEG), matrix representation of features and image classification to categorize videos taken by a humanoid robot. The ultimate goal is to endow the robot with action recognition capabilities for a more natural social interaction. Summarizing, we apply the CSP algorithm to a set of signals obtained for each video by extracting skeleton joints of the person performing the action. From the transformed signals a summary image is obtained for each video, and these images are then classified using two different approaches; global visual descriptors and convolutional neural networks. The presented approach has been tested on two data sets that represent two scenarios with common characteristics. The first one is a data set with 46 individuals performing 6 different actions. In order to create the group of signals of each video, OpenPose has been used to extract the skeleton joints of the person performing the actions. The second data set is an Argentinian Sign Language data set (LSA64) from which the signs performed using just the right hand have been used. In this case the joint signals have been obtained using MediaPipe. The results obtained with the presented method have been compared with a Long Short-Term Memory (LSTM) method, achieving promising results.

INDEX TERMS Action recognition, social robotics, global visual descriptors, common spatial patterns, sign language recognition.

I. INTRODUCTION

Video action recognition is a task which involves recognizing the action that is being performed in a sequence of observations. It is mainly used in computer vision, since the visual features provide basic information about what is happening in the image sequence, and has many real-life applications, such as visual surveillance, rehabilitation, human-computer interaction or entertainment.

Due to the fast growth of the technology, the demand for automatic interpretation of human behavior within videos is also growing, making video action recognition a highly active area. Even though many different approaches have been presented throughout the years trying to solve the problem of the identification of actions in videos, action recognition has not seen the gains in performance that have been achieved in image classification or human face recognition. The main reason is the complexity of combining both spatial and temporal information, which makes this problem harder than image analysis.

In this paper, a pipeline for a video action recognition method is presented, which has been applied to solve two problems with common characteristics. The first application where the presented method has been tested is human-robot interaction. Human-robot interaction (HRI) aims to understand, design and evaluate robotic systems to be used by or with humans. Specially when dealing with social robots, a highly evolved type of interaction is required, since these

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

(a) Image captured by the robot. (b) Expected reaction of the robot.

FIGURE 1. Interaction example.

robots cannot be merely teleoperated, and they are expected to meet high operational standards in order to be accepted by the general public.

The presented method aims to endow a pseudo-humanoid robot with the ability to understand the action that an actor is performing, in order to be able to give an adequate response, thus enhancing the social capabilities of the robot. A data set with six different actions performed by different people has been created to test the method. In Fig. 1 an interaction example between a person and the robot is displayed.

The second application is the sign language recognition. Nowadays, a large number of people has some degree of hearing impairments, about 466 million, and this number is expected to grow in the next years. Many of those people use sign languages to communicate with others, but since these languages are not commonly known among the hearing community, people with hearing problems often face communication difficulties in environments where no interpreter is available. In order to try to break the barrier between the hearing impaired community and the rest of the society, significant work is being carried out in Sign Language Recognition (SLR), where computer vision is playing a major role.

In order to improve the interaction between the people with hearing impairments and the robot, it is interesting to endow the robot with the ability to recognize certain gestures and react in different ways. Driven by the results obtained in [1], it has been decided to test the method presented in this paper on the recognition of some signs that are included in an Argentinian Sign Language database.

The approach presented herein continues with the work presented in [2], where Common Spatial Patterns (CSP), a method commonly used in Brain Computer Interface (BCI) for ElectroEncephaloGram (EEG) systems [3], [4], is used as feature extraction method for a video action recognition task.

In order to apply CSP, the information about the person performing the action to identify must be extracted. To that end, two different technologies have been used: OpenPose [5] to extract the skeletons of the action recognition videos and MediaPipe [6] to extract hand landmarks of the sign language data set.

The positions of the joints of the skeletons are used as input for the CSP, as presented in the previous work [2]. In this new approach, after computing the CSP algorithm, a matrix multiplication is applied and the transformed signals are represented as images. The features for the classification are extracted from those images using several visual global descriptors, and different classifiers have been tested to perform the classification.

Several experiments have been performed with the proposed approach in both databases, and their results have been compared to a Long Short-Term Memory (LSTM) paradigm in order to validate it.

The rest of the paper is organized as follows. First, in Section II some related works are described in order to introduce the topic. In Section III the proposed approach is introduced, explaining the process that has been carried out. Then, in Section IV the databases are presented and the experiments are explained further. Next, in Section V the obtained results are shown, and finally, in Section VI the conclusions extracted from this work are presented and future work is pointed out.

II. RELATED WORKS

Many approaches for video action recognition have been introduced lately. These techniques make use of the visual features extracted from the video, both static and temporal. The temporal features mix the static image features with time information, so that the temporal information of the video is maintained.

In [7] the authors use a temporal template which is based on a static vector-image where the value of the vector at each point represents a function of the motion properties at the corresponding spatial location in an image sequence. Local spatio-temporal interest points can be used to recognize complex motion patterns as it is demonstrated in [8]. A hybrid hierarchical model is presented in [9] where collections of spatial and spatio-temporal features are used to represent video sequences. Many other methods make use of Histograms of Oriented Gradients (HOG) or Histogram of Oriented Optical Flow (HOOF) [10]–[12]. Motion descriptors based on the direction of optical flow have also been introduced [13], [14]. The use of depth data captured by depth cameras has also grown due to the advances in imaging technology [15], [16].

With these two publications [17], [18] as a starting point, deep learning has continued to be used for activity recognition, mainly with Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) [19]. Very deep two-stream ConvNets are presented in [20] which, according to the authors, get close to image domain deep models. Convolutional Neural Networks (CNN) and deep bidirectional LSTM (DB-LSTM) networks are used in [21]. In [22] the authors combine 3D-CNN and LSTM networks. Motion maps, which integrate temporal information, are iteratively extracted from videos using a kind of deep 3-dimensional

CNN (C3D), acquiring a final motion map of the whole video. LSTM is used for the final prediction.

As two-stream CNNs are unable to model long-term temporal structures, Wang *et al.* [23] developed a temporal segment network (TSN) which is able to model dynamics throughout the whole video. TSN extracts short snippets over a long video sequence with a sparse sampling scheme, this way modeling long-range temporal structures and preserving relevant information. Temporal Relation Network (TRN) [24] is a network module which enables temporal relational reasoning and can be easily plugged into an existing neural network. The module tries to describe the temporal relations between observations in videos. While TSN uses average pooling ignoring the temporal order, TRN replaces the average pooling with an interpretable relational module. Authors of [25] proposed a Temporal Shift Module (TSM) which shifts the channels forward or backward along the temporal dimension to exchange information between adjacent frames. The Gate Shift Module (GSM) [26] has a learnable spatial gating block which controls spatio-temporal interactions. Other authors [27] present Channel-Separated Convolutional Networks (CSN), which factorize 3D convolutions in pointwise $1 \times 1 \times 1$ convolutions for channel interaction or depthwise $k \times k \times k$ (usually $k = 3$) convolutional operations for local spatio-temporal interactions. Temporal Pyramid Network (TPN) [28] models the visual tempo at feature level, extracting temporal features by combining features obtained at different tempos.

Skeleton data has also been used to perform activity recognition. The authors of [29] use a LSTM network to focus on the significant joints of the skeleton within each frame and, according to that, the outputs of different frames are weighted. In [30] the authors present a representation where a human pose estimator is used and heatmaps are extracted for the human joints in each frame. In [31] a method for encoding geometric relational features into color texture images is presented, where temporal variations of different features are converted into the color variations of their corresponding images. They use a multi-stream CNN model to classify the images. The authors of [32] propose a two-stream adaptive graph convolutional network (2s-AGCN), where both the coordinates of the joints and the bones between the joints are used as features for classification.

Regarding Sign Language Recognition (SLR), different techniques have been used in recent years [33]–[36]. On the one hand, we can find methods that make use of intrusive sensors which must be placed on the person who is performing the signs. These wearable markers or data gloves are used to detect the body and hand movements [37]–[39]. In the case of non-intrusive systems, there are techniques that make use of sensors such as Leap Motion or Microsoft Kinect [40]–[43] and others that focus on the information obtained by cameras, vision-based methods [44]–[46]. Most of the presented methods use neural networks to perform the classification, like CNNs and LSTMs [47]–[49], although Hidden Markov Models (HMM) have been widely used for SLR too [50]–[52]. As a practical application, it is possible to mention [53], where the authors develop a software system for hearing impaired children with articulation disorders.

III. PROPOSED APPROACH

The method presented in this paper is a continuation of the work presented in [2], which uses CSP applied to skeleton information for video action recognition. In this work, an image is obtained for each video that summarizes the information of the video and that can be then classified using image classifiers. Therefore, we transform the video classification problem into an image classification problem. An overview of the proposed approach can be seen in Fig. 2.

As seen in the overview of the method, the first step is the extraction of the skeletons of the person performing the action or sign to be recognized. The positions of the joints in the skeletons are then used to create signals. The created signals are the input for the Common Spatial Patterns algorithm.

The Common Spatial Patterns (CSP) algorithm [54], a mathematical technique for signal processing, has been widely used in Brain Computer Interface (BCI) applications for electroencephalography (EEG) systems [55], [56]. It has also been applied in the field of electrocardiography (ECG) [57], electromyography (EMG) [58], [59] or even in astronomical images for planet detection [60]. CSP was presented as an extension of Principal Component Analysis (PCA) and it consists of finding an optimum spatial filter which reduces the dimensionality of the original signals. Considering just two different classes, a CSP filter (1) maximizes the difference of the variances between the classes, maximizing the variance of filtered signals of EEG of one of the targets while minimizing the variance for the other.

$$\mathbf{W} = argmax \frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{1}$$

As the feature vectors of the spatial filter W are sorted by variance, the first and the last q vectors, which produce the smallest variance for one class and the largest variance for the other class, are used to project the original signals (2). Finally, the feature vector is obtained by calculating the variance of the transformed signals Z (3). The feature vector value for the p-th component of the i-th trial is the logarithm of the normalized variance.

$$Z = W^T X \tag{2}$$

$$f_p^i = log\left(\frac{var_p(Z_i)}{\sum_{p=1}^{2q} var_p(Z_i)}\right) \tag{3}$$

The CSP algorithm can only work with pairs of classes, but multiclass classification is possible using pairwise classification approaches, such as One versus One (OVO) as a class binarization technique [61].

The CSP-filtered signals are further processed applying two matrix operations. Being $M \in R^{KxL}$ a matrix formed by the extracted video signals where K is the number of signals and L is the maximum length value, on the one hand, a matrix

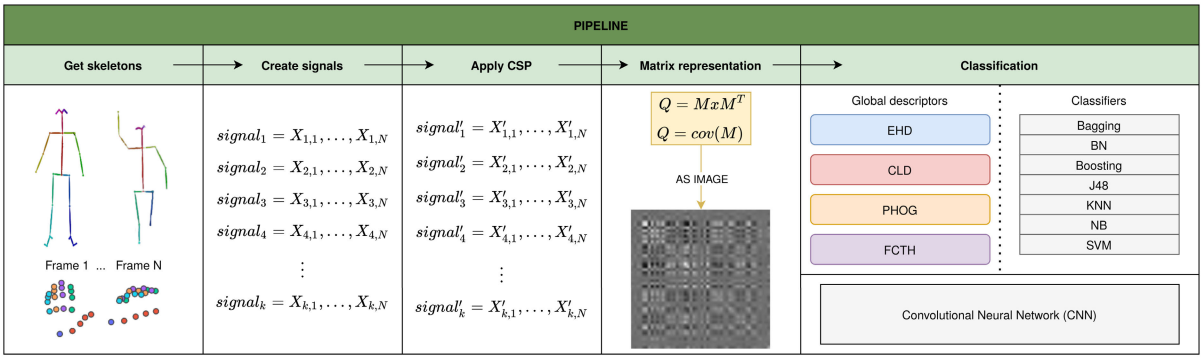


FIGURE 2. Overview of the presented approach.

multiplication is performed (Eq. 4) and, on the other hand, the covariance matrix is calculated (Eq. 5). The motivation behind these transformations is that one of the dimensions of the matrix representing the signals is the number of signals, but the other could be arbitrary long, as it is the number of time steps or frames. Therefore a matrix multiplication by its transpose reduces the data to a manageable size. On the other side, centering a matrix, multiplying by its transpose and dividing by the number of rows - 1 produces the covariance matrix, which provides information about global characteristics of the signals.

$$Q = M * M^T \tag{4}$$

$$Q = cov(M) = \frac{1}{n-1}\sum_{j=1}^{n}(M_j - \overline{M})(M_j - \overline{M})^T \tag{5}$$

A $K \times K$ matrix is obtained, being K the number of signals, the number of rows of the matrix. These matrices are then treated as images; that is, for each video one image is obtained.

The created images are then classified to identify the action that has been performed on the original video.

IV. EXPERIMENTAL SETUP

A. DATA SETS

In the experiments presented in this paper two data sets have been used: one has been collected by us, and the other is a public available database.

1) ACTION RECOGNITION (AR)

This database has been created by recording videos with the camera of the semi-humanoid robot Pepper. It consists of 272 videos with 6 action categories. There are around 45 clips in each category, performed by 46 different people. When recording the actions, the robot adjusts the orientation of its head according to the location of the face of the person appearing in its field of view.

The action categories and the information about the videos can be seen in Table 1.

These are the 6 categories that are included in the data set:

1) Come: gesture for telling the robot to come to you.
2) Five: gesture of 'high five'.

TABLE 1. Characteristics of each action category.

Category	#video	Resolution	FPS
Come	46	320×480	10
Five	45	320×480	10
Handshake	45	320×480	10
Hello	44	320×480	10
Ignore	46	320×480	10
Look at	46	320×480	10

TABLE 2. LSA64 signs used for classification and their characteristics.

CLASS	ID	ENV.	CLASS	ID	ENV.	CLASS	ID	ENV.
Opaque	001	Indoor	*Born*	015	Indoor	*Birthday*	030	Outdoor
Red	002	Indoor	*Learn*	016	Indoor	*Hungry*	033	Outdoor
Green	003	Indoor	*Call*	017	Indoor	*Ship*	037	Outdoor
Yellow	004	Indoor	*Skimmer*	018	Indoor	*None*	038	Outdoor
Bright	005	Indoor	*Bitter*	019	Indoor	*Name*	039	Outdoor
Light-blue	006	Indoor	*Sweet milk*	020	Indoor	*Patience*	040	Outdoor
Colors	007	Indoor	*Milk*	021	Indoor	*Perfume*	041	Outdoor
Red2	008	Indoor	*Water*	022	Indoor	*Deaf*	042	Outdoor
Women	009	Indoor	*Food*	023	Indoor	*Candy*	046	Outdoor
Enemy	010	Indoor	*Argentina*	024	Outdoor	*Chewing-gum*	047	Outdoor
Son	011	Indoor	*Uruguay*	025	Outdoor	*Shut down*	052	Outdoor
Man	012	Indoor	*Country*	026	Outdoor	*Buy*	059	Outdoor
Away	013	Indoor	*Last name*	027	Outdoor	*Realize*	062	Outdoor
Drawer	014	Indoor	*Where*	028	Outdoor	*Find*	064	Outdoor
FPS: 60			**Resolution:** 1920x1080			**Pos. camera:** 2m away		

3) Handshake: gesture of handshaking with the robot.
4) Hello: gesture for telling hello to the robot.
5) Ignore: ignore the robot, pass by.
6) Look at: stare at the robot in front of it.

2) SIGN LANGUAGE RECOGNITION(SLR)

For the SLR task an Argentinian Sign Language (LSA) data set, LSA64 data set [62] is used, which is composed of 64 different LSA signs. The videos were recorded by 10 non-expert subjects, who repeat each sign 5 times. Among the performed signs, both one-handed (42 signs performed with the right hand) and two-handed (22 signs) signs can be found. In order to simplify the classification problem, a subset of the data set has been selected, precisely the 42 one-handed videos have been used. The name and information of the used signs can be seen in Table 2. Thus, the subset used is composed by 2100 videos, where 1150 videos were recorded outdoors with natural lighting (23 signs, 10 signers, 5 repetitions) and 950 videos were recorded indoors with artificial lighting (19 signs, 10 signers, 5 repetitions).

The signers wore black clothes and colored gloves (red and green), and they were recorded with a white wall as background. The colored gloves (red and green) are used in order to facilitate the task of hand segmentation, although this

is not helpful in the approach presented in this paper, as no hand segmentation is performed. It must be mentioned that the subjects do not make use of the facial expression when performing the signs, they just focus on the movements of the hands.

B. METHOD APPLICATION

The method described in Section III has been applied to both of the presented data sets. Even though both data sets correspond to scenarios where the action or the sign performed by the person in front of the camera needs to be identified, some differences have been made on the application of the method. Different classifiers have also been tested on the classification step of the images that are created from the videos.

The different setups that have been tried are described below.

1) GET SKELETONS AND CREATE SIGNALS

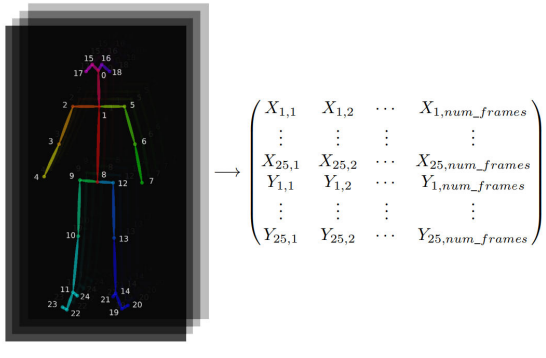
The selected data sets have different purposes; on one hand the AR data set is an action recognition data set where different subjects perform general actions where the whole body is involved. On the other hand, on the SLR data set the focus is always on the upper body of the signers, specially on their hands. Due to this dissimilarity, different methods have been selected to extract the skeletal information of the videos of the different databases.

On the AR data set, it has been decided to use OpenPose [5] to extract the skeletons of the people of the scene. This tool is a real-time multi-person system to detect human body on single images. In this case, the actions that have to be recognized are centered in the actor who perform them. Therefore, the skeleton of the actor has been extracted in every frame of each video. The system has been designed with the restriction that only a person ought to be in the field of view of the camera. In any case, as OpenPose allows for restricting the detection to only one person in order to speed up the processing and tracking, this approach ignores people in the background.

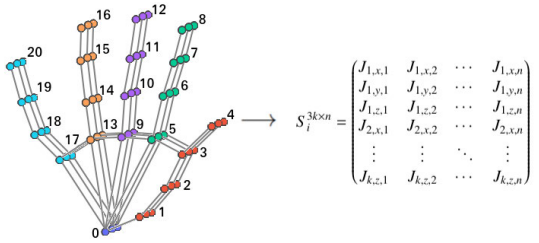
OpenPose returns the (x, y) positions of 25-keypoints (joints). After obtaining the skeleton information for every frame of each video, we can create 50 different signals to represent each video, where each signal will be the position of a skeleton keypoint over time. This way, there will be 50 signals (25 for the x position of the joints and another 25 for the y positions) with the same length as the original video (one skeleton per frame). The appearance of the skeleton and the matrix extracted from the skeletons can be seen in Fig. 3a.

For the SLR data set a technology called MediaPipe [6] has been used to track the positions of the hands in each frame of the video. More precisely the MediaPipe Holistic solution is used, which integrates separate models for pose, face and hand components. This solution offers a real-time hand tracking, which includes 21 hand landmarks for each hand.

It has been noticed that due to the speed of the movements or the use of color gloves, MediaPipe is not able to track the



(a) AR skeletons.



(b) SLR hand landmarks.

FIGURE 3. Joints positions and matrix representation of the extracted signals.

hands in 52 videos of the SLR data set. In order to try to solve this issue, the original videos have been converted from RGB color space to black and white. This way, the performance of Mediapipe has been improved and the number of videos where the hand is not detected in any frame has dropped to 6.

Each landmark returned by MediaPipe is composed of three coordinates (x, y, z), where (x, y) denote its position and the z coordinate represents the depth of each joint in reference to the position of the wrist. Once the landmark values are obtained, a set of signals is created for every video of the database.

In Fig. 3b a graphical explanation of the hand landmarks and the extracted set of signals S for video i are shown, where k is the number of joint features, n is the number of frames and $J_{u,c,v}$ is the landmark value for joint u, coordinate $c : x, y, z$ and frame v. For each frame 21 joints $(k = 21)$ are extracted, and as each landmark is composed of (x, y, z) values, the signal matrix has 63 rows: 3 values (x, y, z) for each one of the 21 joints $(3 \times 21 = 63)$. In Fig. 4 an example of the sequence obtained from a video is shown, both for the action recognition data set and the LSA64 data set. In 4a the skeleton obtained by OpenPose is presented and, in 4b, the hand landmarks extracted with MediaPipe.

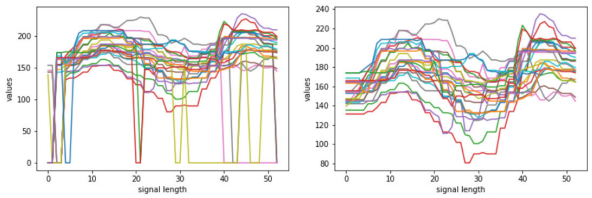
2) APPLY THE COMMON SPATIAL PATTERNS ALGORITHM

In order to compute the CSP algorithm, the signals have been preprocessed first. On the one hand, it has to be considered that some joints could be missing from the captured skeletons when the actor does not fit entirely in the camera range or OpenPose and MediaPipe are not able to capture some of the landmarks. In these cases, the missing joints values are estimated by a linear interpolation, using the previous

(a) Example of skeleton obtained for an action sequence.

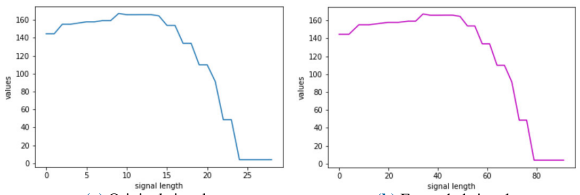
(b) Example of hand landmarks obtained for a sign sequence.

FIGURE 4. Frame sequences examples for different categories.



(a) Before interpolation. (b) After interpolation.

FIGURE 5. Interpolation example to avoid missing values.



(a) Original signal. (b) Expanded signal.

FIGURE 6. Interpolation example to enlarge signals.

and next values of that joint. The interpolation is done to avoid having missing values, and assuming that consecutive values of joints positions follow a smooth curve. An example of 25 signals of the x poses of a joint can be seen in Fig. 5, where the signals before and after the interpolation are shown.

Furthermore, all the signals of every video have been set to the same length. Since this is not the case of the videos used in the experiments, the longest video has been selected and all the signals have been enlarged to the number of frames of that video. To assign the same length to all the videos, new values have been introduced between the original values of the joints, uniformly. The added values are interpolated with the original values among which they are found. An example of a single signal extension is shown in Fig. 6.

Once the landmarks are processed and, hence, the signals are formed, the CSP is computed in order to separate the classes according to their variance. Since in both data sets a multiclass classification needs to be performed, a pairwise approach is used. In Fig. 7 an example of the variances obtained from the signals transformed applying the CSP algorithm can be seen.

As it has been explained, the CSP filter tries to separate the given classes by variance, where the first q vectors produce

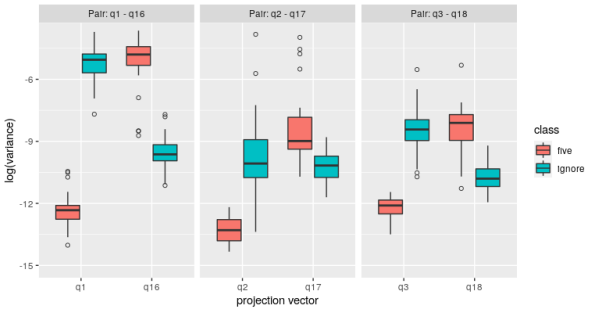
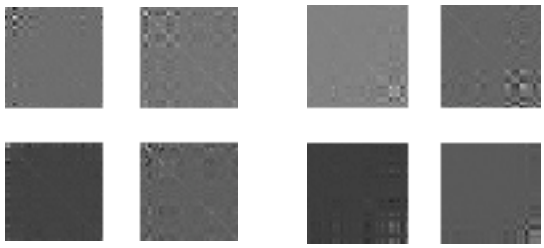


FIGURE 7. Boxplot of variances obtained from different projection vectors, by class.



(a) First class images (b) Second class images

FIGURE 8. Examples of achieved images, after applying CSP and matrix operation. On the left images for class *come* are shown and on the right, images for class *five*.

the smallest variance for one class and the largest for the other, while the last q vectors produce the opposite. In Fig. 7, three pairs of vectors are shown ($q1 - q16$, $q2 - q17$ an $q3 - q18$) and it is clearly noticeable the difference between the variances of the classes (*ignore* and *five*) in each of the vectors, where the first q vectors ($q = 15$ in this case) minimize the variance of class *five* and maximize the variance of class *ignore*, and the last q minimize the variance of class *ignore* and maximize the variance of class *five*.

3) MATRIX REPRESENTATION

In Fig. 8 some examples of obtained images are shown, for the classes *come* and *five* of the presented AR data set (as mentioned before, all the process is computed in pairs). The images are low-dimensional since they come from 50×50 matrices.

4) CLASSIFICATION

Once the summary images are created, different classification strategies can be used in order to classify them into the original action classes. In this work two paths have been explored; a global descriptor strategy and the use of a Convolutional Neural Network (CNN).

a: VISUAL GLOBAL DESCRIPTORS

Some commonly used visual descriptors (image descriptors) have been used to extract useful information from the created summary images. These descriptors describe visual features of images or videos, encoding interesting information into a list of numbers. They describe basic characteristics such as shape, color, texture or motion. In our approach four different descriptors have been used:

- The Color Layout Descriptor (CLD): a technique, proposed by the MPEG-7 standard, designed to capture the spatial distribution of color of an image.
- The Pyramid Histogram of Oriented Gradients (PHOG) descriptor [63]: it represents an image by its local shape and the spatial information of the shape.
- The Fuzzy Color and Texture Histogram (FCTH) descriptor [64]: it joins color and texture information in a single histogram.
- The Edge Histogram Descriptor (EHD) [65]: it extracts MPEG-7 edge histogram features from images, a summary of the edges directions across an image.

Different classifiers have been trained with the feature vectors constructed from the descriptors. These classifiers are: Bagging, Bayesian Network (BN), Boosting, J48 classification Tree, K-Nearest Neighbors (KNN), Naive Bayes (NB) and Support Vector Machine (SVM). For each descriptor type these seven classifiers have been trained and evaluated by a 10-fold Cross Validation.

b: CONVOLUTIONAL NEURAL NETWORKS

Additionally, a Convolutional Neural Network (CNN) has been applied to classify the summary images obtained for each video. Its performance might drastically vary between several hyperparameter configurations, and therefore, in order to provide a fair comparison, we have used Keras Tuner Hypermodel, with a RandomSearch tuner to look for good configurations automatically. The input is composed by one image per video with a shape of $50 \times 50 \times 1$ in the case of the AR data set and $63 \times 63 \times 1$ when it refers to LSA64 database, since the images are gray-scale.

Convolutional layers, dropout layers, max pooling layers and a final dense layer of two units (as the classification is performed by pairs) make up the network. Adam is used as optimizer and categorical cross-entropy as loss function. The learning rate, activation functions, number of filters and dropout rate hyperparameters have been tuned.

5) COMPARISON

To finish, the proposed approach is compared with a type of neural network widely used for video action recognition task, a Long Short-Term Memory (LSTM) network. The LSTM network has the signals obtained from the joints of the skeleton as input, so its input is bidimensional (number of frames, number of joints) and the output is of 64 units. Then a dense layer of 2 units has been placed, since the classification is carried out between two classes. Regarding the rest of the hyperparameters, Adam optimizer and categorical cross-entropy loss function have been used, and the network has been trained for 100 epochs with a batch size of 25.

V. EXPERIMENTAL RESULTS

In this section the obtained results for the experimentation that has been carried out are presented. First, the results obtained for the action recognition data set are shown and afterwards the outcomes obtained for the LSA64 are explained.

A. ACTION RECOGNITION DATA SET

Table 3 and 4 show the results obtained using the characteristics extracted by the global descriptors from the images obtained by matrix multiplication (equation 4) and from the images of covariance matrices (equation 5), respectively. A mean accuracy value is also presented for each pair of classes, with the best values highlighted in bold. In the tables, to summarize, the classes have been represented as follows: C (come), F (five), H (handshake), He (hello), I (ignore) and L (look at).

First, if the two tables (Table 3 and Table 4) are compared with each other, both the matrix multiplication and the covariance matrix obtain good results. After applying the descriptors, both representations yield a mean over all the entries of $\sim$0.86 and a median of $\sim$0.92. Thus, in general the results are encouraging.

Next, three types of comparisons are made: by classifier type, by image descriptor type, and by class pairs.

Regarding the classifiers, on average they all get even results, there is no one that stands out from the rest. Even so, it could be said that Naive Bayes (NB) has been the worst of all in both representations and the best average result is achieved by Support Vector Machine (SVM).

Concerning the descriptors, the difference is more noticeable. EHD and PHOG get outstanding results with an average accuracy of $\sim$0.95. The CLD descriptor does not get bad results either ($\sim$0.87 on average). The worst results, by far, are achieved with the characteristics obtained using the FCTH descriptor.

Finally, in relation to the pairs of classes, the good results of *handshake-hello (H-He)*, *handshake-ignore (H-I)* or *hello-look at (He-L)* can be highlighted. For instance, the pairs *five-ignore (F-I)* and *come-ignore (C-I)* achieve very good results with all the descriptors except FCTH, which as it has been already mentioned is the descriptor with the worst results overall. However, the worst pair of classes (*five-hello (F-He)*) obtains an average of 0.71 accuracy, therefore very good results have been achieved in the experiment.

Table 5 shows the average accuracy values obtained for each type of descriptor and image of the presented approach, along with the results obtained using the CNN network taking as input the matrix representation images and the results achieved by the LSTM mentioned before, where the best values are highlighted in boldface. The results obtained with a previous approach [2] are also shown.

The obtained accuracy values show that our new approach beats LSTM method and the previous approach for every class pairs. Furthermore, observing this table it is evident that the best results are achieved using the CNN and, EHD or PHOG when it comes to global descriptors. Although the best mean value corresponds to the use of PHOG descriptor, in 9 out of 15 pair of classes CNN performs better. Regarding the type of images, generally better outcomes are obtained using

TABLE 3. Accuracy values for every pair of classes, with each type of descriptor and classifier. Approach: matrix multiplication, AR data set.

Descrip.	Classif.	C-F	C-H	C-He	C-I	C-L	F-H	F-He	F-I	F-L	H-He	H-I	H-L	He-I	He-L	I-L
CLD	Bag.	0.7033	0.8022	0.7222	0.9674	0.8152	0.8333	0.7416	0.8791	0.8791	0.809	0.967	0.8791	0.9667	0.9556	0.9565
	BN	0.6264	0.8462	0.7556	0.9674	0.8152	0.8556	0.7865	0.9451	0.9121	0.8989	0.967	0.9011	0.9778	0.9444	0.9891
	Boost.	0.7473	0.7692	0.7556	0.9674	0.8804	0.8556	0.764	0.9341	0.9231	0.8876	0.9451	0.8791	0.9556	0.9667	0.9783
	J48	0.7143	0.7253	0.7778	0.9565	0.8478	0.8333	0.7753	0.9341	0.9011	0.8989	0.9341	0.9121	0.9556	0.9222	0.9674
	KNN	0.7033	0.7912	0.7111	0.9674	0.8587	0.8333	0.7079	0.8901	0.8791	0.809	0.967	0.8901	0.9667	0.9556	0.9565
	NB	0.6923	0.8242	0.7889	0.9565	0.8478	0.8667	0.7978	0.9341	0.8901	0.8764	0.967	0.8791	0.9778	0.9556	0.9783
	SVM	0.6813	0.7692	0.7111	0.9674	0.8587	0.8111	0.7753	0.9341	0.9231	0.8315	0.9451	0.9341	0.9667	0.9667	0.9783
EHD	Bag.	0.8681	0.9231	0.8333	1	0.9783	0.9778	0.6854	1	0.989	0.9551	1	0.9341	0.9889	0.9889	1
	BN	0.8022	0.8681	0.7111	1	0.9891	0.9778	0.6629	1	1	0.9663	0.978	0.9341	0.9889	1	1
	Boost.	0.8791	0.8901	0.7556	1	0.9565	0.9556	0.5843	0.978	1	0.9888	0.989	0.8791	0.9889	1	0.9891
	J48	0.8132	0.8462	0.7	0.9891	0.9457	0.9667	0.5843	0.989	1	0.9663	0.967	0.8462	0.9889	1	0.9674
	KNN	0.8571	0.9341	0.8111	1	0.9783	0.9778	0.6629	1	0.989	0.9551	1	0.9341	0.9889	0.9889	1
	NB	0.8022	0.8462	0.7	1	0.9457	0.9556	0.6966	1	0.978	0.9326	0.967	0.8791	0.9889	0.9778	1
	SVM	0.9011	0.967	0.8	1	1	0.9778	0.7865	1	1	0.9888	1	0.956	1	1	1
FCTH	Bag.	0.7582	0.7143	0.5778	0.5543	0.7935	0.5111	0.5506	0.5275	0.7143	0.9438	0.8791	0.6923	0.5333	0.9222	0.6957
	BN	0.7582	0.7692	0.6222	0.4565	0.7609	0.4444	0.5056	0.4505	0.7143	0.9438	0.8462	0.4396	0.5111	0.8889	0.4565
	Boost.	0.8022	0.7692	0.6333	0.5761	0.7717	0.4778	0.5393	0.4396	0.7143	0.9551	0.8352	0.6264	0.5222	0.9	0.5978
	J48	0.8022	0.7363	0.6778	0.5761	0.8261	0.4222	0.5618	0.4396	0.7473	0.9551	0.8352	0.5934	0.4889	0.9	0.6957
	KNN	0.7473	0.7143	0.5556	0.5761	0.7935	0.5	0.5506	0.5385	0.7143	0.9438	0.8571	0.6923	0.5111	0.8889	0.6957
	NB	0.8022	0.6484	0.7222	0.5652	0.6413	0.5333	0.5393	0.4396	0.7253	0.8989	0.7912	0.6593	0.5	0.9333	0.6848
	SVM	0.8242	0.6813	0.6889	0.5326	0.75	0.4778	0.573	0.4505	0.6593	0.8764	0.8352	0.7033	0.4778	0.9222	0.663
PHOG	Bag.	0.9231	0.956	0.9333	1	1	0.9	0.8989	1	0.978	0.9438	1	0.956	1	0.9778	1
	BN	0.978	0.989	0.9444	0.9891	1	0.9889	0.9551	1	0.978	0.9775	1	0.956	1	1	0.9891
	Boost.	0.967	0.989	0.9444	0.9891	1	1	0.8764	1	0.978	0.9888	0.989	0.967	1	1	1
	J48	0.9011	0.989	0.8556	1	1	1	0.8539	1	0.978	0.9888	1	0.956	1	0.9889	1
	KNN	0.9231	0.967	0.9222	0.9891	1	0.9	0.8989	1	0.978	0.9438	1	0.9451	1	0.9889	1
	NB	0.8242	0.8791	0.8	0.9674	0.913	0.9556	0.7416	0.978	0.956	0.9551	0.978	0.8791	0.9889	0.9667	1
	SVM	0.978	0.989	0.9444	0.9891	1	1	0.8876	1	0.978	0.9663	1	0.967	1	1	1
MEAN		0.8136	0.8426	0.7627	0.8750	0.8917	0.8139	0.7123	0.8458	0.8956	**0.9302**	**0.9443**	0.8454	0.8655	**0.9607**	0.9014

TABLE 4. Accuracy values for every pair of classes, with each type of descriptor and classifier. Approach: covariance matrix, AR data set.

Descrip.	Classif.	C-F	C-H	C-He	C-I	C-L	F-H	F-He	F-I	F-L	H-He	H-I	H-L	He-I	He-L	I-L
CLD	Bag.	0.6593	0.8022	0.7667	0.9783	0.9348	0.8333	0.6966	0.978	0.8681	0.8652	0.9341	0.8571	0.9667	0.9444	0.9457
	BN	0.7473	0.8571	0.7111	0.9783	0.9022	0.8778	0.6629	0.978	0.9011	0.8876	0.9341	0.9231	0.9667	0.8889	0.9565
	Boost.	0.7582	0.8242	0.8333	0.9783	0.9239	0.8556	0.6742	0.967	0.8681	0.8652	0.956	0.9121	0.9222	0.9222	0.9348
	J48	0.7802	0.7473	0.8	0.9891	0.913	0.8444	0.6854	0.978	0.8791	0.8539	0.9341	0.9011	0.9	0.8556	0.9239
	KNN	0.6593	0.7802	0.7778	0.9891	0.9239	0.8444	0.6966	0.978	0.8571	0.8764	0.9341	0.8571	0.9667	0.9333	0.9565
	NB	0.7582	0.8352	0.7444	0.9783	0.913	0.9111	0.764	0.956	0.9011	0.8764	0.9341	0.9011	0.9778	0.9444	0.9783
	SVM	0.7143	0.8022	0.8333	0.9891	0.9239	0.8667	0.764	0.978	0.8681	0.8427	0.9341	0.8901	0.9444	0.9	0.9457
EHD	Bag.	0.8681	0.9451	0.8222	1	0.9891	1	0.7191	1	1	0.9888	0.989	0.9451	0.9889	1	0.9891
	BN	0.8242	0.956	0.7667	1	0.9783	0.9778	0.7191	1	0.989	0.9663	0.989	0.9451	0.9889	1	0.9783
	Boost.	0.9011	0.9121	0.8111	1	0.9239	0.9667	0.7416	1	0.989	0.9663	0.989	0.9231	0.9778	0.9778	1
	J48	0.8352	0.9451	0.7889	0.9674	0.9457	0.9556	0.6067	1	1	0.9326	0.956	0.8791	0.9778	0.9778	0.9457
	KNN	0.8681	0.9341	0.8444	1	0.9891	1	0.7416	1	1	0.9775	0.989	0.9451	0.9889	1	0.9891
	NB	0.7802	0.8132	0.7333	1	0.9348	0.9444	0.6742	1	0.989	0.9551	0.978	0.9011	0.9778	0.9556	0.9674
	SVM	0.956	0.9341	0.8667	1	1	0.9889	0.8202	1	1	0.9775	1	0.967	1	1	0.9891
FCTH	Bag.	0.7363	0.6484	0.7111	0.5652	0.7717	0.5556	0.573	0.5495	0.6593	0.9438	0.8901	0.6703	0.5667	0.8889	0.6196
	BN	0.7802	0.7582	0.7333	0.4565	0.7174	0.4444	0.4607	0.4505	0.6923	0.9551	0.8901	0.4505	0.5111	0.8778	0.4783
	Boost.	0.7912	0.7692	0.7	0.5652	0.6739	0.4889	0.6629	0.4396	0.6593	0.9663	0.9011	0.6154	0.5556	0.9	0.587
	J48	0.7692	0.7582	0.7444	0.5435	0.7717	0.4667	0.5955	0.4286	0.6923	0.9663	0.9011	0.6154	0.5444	0.8667	0.6196
	KNN	0.7363	0.6923	0.6778	0.5652	0.7609	0.5111	0.5955	0.5385	0.6593	0.9551	0.9121	0.6374	0.5444	0.8444	0.6196
	NB	0.7912	0.6374	0.7	0.5761	0.6304	0.5444	0.5955	0.4725	0.5165	0.9213	0.8571	0.6154	0.5222	0.9	0.663
	SVM	0.8132	0.7033	0.7444	0.5543	0.7283	0.5667	0.6517	0.4396	0.5934	0.8652	0.9011	0.6484	0.5444	0.9111	0.5978
PHOG	Bag.	0.9451	0.8901	0.8889	1	1	0.9111	0.8315	0.989	0.967	0.9551	1	0.9231	1	0.9444	1
	BN	0.967	0.956	0.9111	0.9891	0.9783	0.9667	0.9326	0.989	1	0.9888	1	0.967	1	0.9667	0.9891
	Boost.	0.9341	0.956	0.9111	0.9891	0.9891	0.9778	0.8764	1	1	0.9663	1	0.967	1	0.9778	0.9783
	J48	0.9231	0.956	0.8333	1	0.9891	0.9889	0.8427	1	1	0.9888	1	0.978	1	0.9889	0.9891
	KNN	0.9451	0.9011	0.8889	1	0.9783	0.9111	0.8202	1	0.967	0.9663	1	0.9231	1	0.9333	1
	NB	0.8132	0.8681	0.7667	0.9891	0.837	0.9556	0.7865	0.989	0.9451	0.9888	0.967	0.8462	1	0.9667	1
	SVM	0.9341	0.9451	0.9222	1	0.9891	0.9667	0.8202	1	0.989	0.9888	1	0.956	1	0.9333	0.9891
MEAN		0.8210	0.8403	0.7940	0.8800	0.8932	0.8258	0.7147	0.8607	0.8732	**0.9374**	**0.9525**	0.8414	0.8691	**0.9357**	0.8797

the features extracted from the covariance matrices. However, the best mean value is attained using the images created from the matrix multiplication.

It is concluded that our approach beats LSTM network since only the configurations which use the FCTH descriptor obtain worse mean accuracy values than the LSTM method.

B. SIGN LANGUAGE RECOGNITION DATA SET, LSA64

The results obtained after using the global descriptors to extract the features from the images created by matrix multiplication and extracting them from the images of the covariance matrix for the LSA64 data set are shown in Table 6 and Table 7, respectively. As the experiments have been performed pairwise and there are 42 different classes, a total of 861 ($42 \times 41/2$) tests have been performed for each classifier and descriptor type. In order to summarize them, in the results some statistics are shown: minimum, 1st quartile, mean, median, 3rd quartile and maximum values.

When looking at the results, it can be seen that the obtained mean values are around $0.8 - 0.9$ for most descriptors and classifiers. Although the minimum values obtained are quite low, the Q1 values already show that the 75% of the pairs of classes get accuracy values higher than 0.9 with EHD and PHOG descriptors, and greater than 0.8 for CLD descriptor. There are some pairs of classes which are perfectly classified (accuracy value 1) with every configuration. The best average result is 0.9911 for matrix multiplication, which is obtained with EHD descriptor and BN classifier. For covariance matrix, the best mean value is 0.9878, obtained with EHD descriptor and SVM classifier instead.

TABLE 5. Comparison between presented approaches, previous approach and a LSTM network for AR data set.

Pair of Categories	Matrix multiplication					Covariance matrix					LSTM	Previous approach
	CLD	EHD	FCTH	PHOG	CNN	CLD	EHD	FCTH	PHOG	CNN		
COME-FIVE	0.6955	0.8461	0.7849	0.9278	0.9355	0.7253	0.8618	0.7739	0.9231	**0.9677**	0.8628	0.7579
COME-HANDSHAKE	0.7896	0.8964	0.7190	0.9654	0.8710	0.8069	0.9200	0.7096	0.9246	**1**	0.7739	0.8668
COME-HELLO	0.7460	0.7587	0.6397	0.9063	**1**	0.7809	0.8048	0.7159	0.8746	0.9333	0.7336	0.5334
COME-IGNORE	0.9643	**0.9984**	0.5481	0.9891	0.9677	0.9829	0.9953	0.5466	0.9953	0.9355	0.9575	0.9779
COME-LOOK_AT	0.8463	0.9705	0.7624	0.9876	**1**	0.9192	0.9658	0.7220	0.9658	**1**	0.7849	0.8678
FIVE-HANDSHAKE	0.8413	0.9699	0.4809	0.9635	0.8667	0.8619	0.9762	0.5111	0.9540	**1**	0.8125	0.9557
FIVE-HELLO	0.7641	0.6661	0.5457	0.8732	**0.9333**	0.7062	0.7175	0.5907	0.8443	**0.9333**	0.9125	0.8208
FIVE-IGNORE	0.9215	0.9953	0.4694	0.9969	**1**	0.9733	**1**	0.4741	0.9953	**1**	0.9789	0.9668
FIVE-LOOK_AT	0.9011	0.9937	0.7127	0.9749	**1**	0.8775	0.9953	0.6389	0.9812	0.9677	0.8889	0.9667
HANDSHAKE-HELLO	0.8588	0.9647	0.9310	0.9663	0.8667	0.8668	0.9663	0.9390	**0.9776**	0.8333	0.7108	0.7431
HANDSHAKE-IGNORE	0.9560	0.9859	0.8399	**0.9953**	0.9677	0.9372	0.9843	0.8932	**0.9953**	0.9032	0.9764	0.9889
HANDSHAKE-LOOK_AT	0.8964	0.9090	0.6295	0.9466	0.9355	0.8917	0.9294	0.6075	0.9372	**0.9677**	0.8350	0.8235
HELLO-IGNORE	0.9667	0.9905	0.5063	0.9984	0.9667	0.9492	0.9857	0.5413	**1**	0.9667	0.9789	0.9333
HELLO-LOOK_AT	0.9524	**0.9937**	0.9079	0.9889	0.9333	0.9127	0.9873	0.8841	0.9587	0.9333	0.5733	0.8445
IGNORE-LOOK_AT	0.9721	0.9938	0.6413	**0.9984**	0.8387	0.9488	0.9798	0.5978	0.9922	0.8710	0.9775	0.9889
MEAN	0.8715	0.9288	0.6746	**0.9652**	0.9389	0.8760	0.9380	0.6764	0.9546	0.9475	0.8505	0.8691

TABLE 6. Results obtained with matrix multiplication and visual global descriptors approach for LSA64 data set.

Descrip.	Classif.	MIN	Q1	**MEAN**	MEDIAN	Q3	MAX
CLD	Bag.	0.5263	0.8300	0.8770	0.8900	0.9424	1
	BN	0.4800	0.8300	0.8801	0.9000	0.9500	1
	Boost.	0.5600	0.8400	0.8847	0.9000	0.9500	1
	J48	0.5500	0.8400	0.8812	0.8900	0.9400	1
	KNN	0.5400	0.8300	0.8773	0.8900	0.9495	1
	NB	0.4900	0.8469	0.8918	0.9000	0.9500	1
	SVM	0.5800	0.8400	0.8855	0.9000	0.9500	1
EHD	Bag.	0.5600	0.9800	0.9837	0.9900	1	1
	BN	0.5000	0.9900	**0.9911**	1	1	1
	Boost.	0.5500	0.9900	0.9883	0.9900	1	1
	J48	0.5700	0.9800	0.9846	0.9900	1	1
	KNN	0.5700	0.9800	0.9822	0.9900	1	1
	NB	0.5600	0.9900	0.9893	1	1	1
	SVM	0.5400	0.9900	0.9889	1	1	1
FCTH	Bag.	0.3434	0.6176	0.7120	0.7100	0.8100	1
	BN	0.4700	0.5000	0.6811	0.6900	0.8025	1
	Boost.	0.4000	0.6200	0.7139	0.7100	0.8100	1
	J48	0.3800	0.6200	0.7140	0.7100	0.8100	1
	KNN	0.3434	0.6100	0.7103	0.7100	0.8101	1
	NB	0.3200	0.6100	0.7024	0.7000	0.7900	1
	SVM	0.3900	0.6100	0.7070	0.7000	0.8000	1
PHOG	Bag.	0.5200	0.9400	0.9530	0.9800	0.9900	1
	BN	0.4900	0.9700	0.9753	0.9900	1	1
	Boost.	0.4500	0.9700	0.9718	0.9900	1	1
	J48	0.4900	0.9600	0.9686	0.9900	1	1
	KNN	0.5300	0.9400	0.9541	0.9800	1	1
	NB	0.5300	0.9500	0.9598	0.9800	0.9900	1
	SVM	0.5300	0.9700	0.9726	0.9900	1	1

TABLE 7. Results obtained with covariance matrix and visual global descriptors approach for LSA64 data set.

Descrip.	Classif.	MIN	Q1	**MEAN**	MEDIAN	Q3	MAX
CLD	Bag.	0.5263	0.7696	0.8309	0.8400	0.9100	1
	BN	0.4700	0.7600	0.8261	0.8586	0.9200	1
	Boost.	0.5100	0.7800	0.8419	0.8500	0.9194	1
	J48	0.5100	0.7800	0.8404	0.8500	0.9100	1
	KNN	0.5053	0.7600	0.8292	0.8400	0.9100	1
	NB	0.4600	0.7900	0.8517	0.8700	0.9200	1
	SVM	0.4400	0.7800	0.8413	0.8500	0.9100	1
EHD	Bag.	0.7100	0.9697	0.9750	0.9800	0.9900	1
	BN	0.7500	0.9800	0.9868	0.9900	1	1
	Boost.	0.7600	0.9800	0.9859	0.9900	1	1
	J48	0.7400	0.9684	0.9778	0.9895	1	1
	KNN	0.7400	0.9600	0.9722	0.9800	0.9900	1
	NB	0.7200	0.9800	0.9860	0.9900	1	1
	SVM	0.6600	0.9895	**0.9878**	1	1	1
FCTH	Bag.	0.4200	0.6000	0.7017	0.6900	0.8000	1
	BN	0.4600	0.5000	0.6645	0.6667	0.7980	1
	Boost.	0.4100	0.6100	0.7042	0.7000	0.8081	1
	J48	0.4343	0.6100	0.7027	0.6900	0.8000	1
	KNN	0.3600	0.6000	0.6998	0.6900	0.7985	1
	NB	0.3700	0.6000	0.6899	0.6737	0.7700	1
	SVM	0.3700	0.6000	0.6933	0.6737	0.7900	1
PHOG	Bag.	0.5400	0.8800	0.9159	0.9500	0.9800	1
	BN	0.4949	0.9300	0.9443	0.9800	0.9900	1
	Boost.	0.4900	0.9200	0.9400	0.9700	0.9900	1
	J48	0.5053	0.9100	0.9344	0.9700	0.9900	1
	KNN	0.5100	0.8900	0.9170	0.9500	0.9800	1
	NB	0.5100	0.8900	0.9219	0.9600	0.9800	1
	SVM	0.4848	0.9200	0.9421	0.9700	0.9900	1

Comparing both tables (Table 6 and Table 7) better results are obtained when using the matrix multiplication. However, there is not a noticeable difference between them.

Regarding the used descriptors, there is a clear difference between the outcomes obtained with each one of them. EHD descriptor is the one which achieves better results. In Table 6 the mean accuracy values are over 0.98 and the 75% of the pairs of classes obtain higher than 0.98 accuracy values. In Table 7 the mean accuracy values are greater than 0.97 and the Q1 value indicates that the 75% of the pairs of classes achieves at least a 0.96 accuracy value.

The PHOG descriptor also obtains good results. Many pairs of classes obtain a 100% of accuracy and the mean values are ∼0.95 for both matrix representation methods. The less suitable descriptor is FCTH. The results obtained with the features extracted with this descriptor are the lowest, where the mean values are ∼0.7, which shows a great difference when comparing with the others.

Concerning the classifiers, there is not a perceptible contrast between them. As mentioned before, the best mean values have been obtained with BN and SVM. However, the worst average values have also been obtained with the BN classifier and FCTH descriptor. It can be concluded that their performance depends on the configuration used before the classification.

In order to compare the differences between the tested classes, in Table 8 the mean values obtained for each class of the data set are shown. These values have been calculated with the accuracy values of all the test pairs in which each class has participated. These mean values are achieved with the best configuration, in this case, the features obtained after applying the EHD descriptor to the images obtained by the matrix multiplication and performing the classification with a Bayesian Network.

All the classes obtain a mean accuracy value between 0.97 and 1.00. Therefore, not many conclusions can be drawn about the difference of classes, since all of them obtain good

TABLE 8. Mean accuracy values obtained with the best configuration (Matrix Multiplication, EHD descriptor and BN classifier) for each class of LSA64 data set.

Opaque	**Red**	**Greeen**	**Yellow**	**Bright**	**Light-blue**
0.9968	0.9868	0.9958	0.9941	0.9973	0.9932
Colors	**Red 2**	**Women**	**Enemy**	**Son**	**Man**
0.9824	0.9753	0.9875	0.9954	0.9932	0.9921
Away	**Drawer**	**Born**	**Learn**	**Call**	**Skimmer**
0.9946	0.9893	0.9971	0.9951	0.9934	0.9963
Bitter	**Sweet-milk**	**Milk**	**Water**	**Food**	**Argentina**
0.9815	0.9932	0.9844	0.9900	0.9876	0.9961
Uruguay	**Country**	**Last name**	**Where**	**Birthday**	**Hungry**
0.9929	0.9934	0.9849	0.9759	0.9910	0.9941
Ship	**None**	**Name**	**Patience**	**Perfume**	**Deaf**
0.9893	0.9953	0.9868	0.9959	0.9909	0.9912
Candy	**Chewing-gum**	**Shut down**	**Buy**	**Realize**	**Find**
0.9973	0.9873	0.9971	0.9878	0.9938	0.9912

TABLE 9. Comparison between presented approaches and LSTM for SLR.

	Matrix multiplication			Covariance matrix			LSTM
	EHD	PHOG	CNN	EHD	PHOG	CNN	
MIN	0.5000	0.4500	0.5455	0.6600	0.4848	0.5000	0.6100
Q1	0.9899	0.9600	0.8788	0.9700	0.9100	0.8788	0.8900
MEAN	**0.9869**	0.9650	0.9267	0.9816	0.9308	0.9171	0.9186
MEDIAN	1	0.9900	0.9394	0.9900	0.9600	0.9394	0.9300
Q3	1	1	1	1	0.9900	0.9697	0.9600
MAX	1	1	1	1	1	1	1

results. *Bright*, *Born*, *Candy* and *Shut down* classes get the highest values and classes like *Red2*, *Where* and *Bitter* get slightly worse values.

A comparison with a LSTM network is performed and the results are shown in Table 9. Some statistics of the obtained accuracy values are displayed, which refer to all pairs of classes. For the comparison, it has been decided to show only the results obtained with EHD and PHOG descriptors, as they are the ones which performed best. The results achieved by applying a CNN after creating the images are also presented.

Although the minimum obtained value among all the pairs of classes is lower in the presented approaches, the rest of the statistics show that better accuracy values are obtained than with the LSTM method. While the LSTM method obtains an average of 0.9186, our approach achieves a mean value of 0.9869. Regarding the approaches which use global descriptors to extract characteristics from the images to train the classifiers, both the median and the Q1 and Q3 values indicate that a greater number of pairs of classes obtain better results than with the LSTM.

All of the 4 configurations presented in the table are able to surpass the LSTM method, being the EHD descriptor the one that suits best, as mentioned above. When using the CNN, although the median and Q3 values are higher, there are several pairs of classes that achieve lower results than with the LSTM, as indicated by the minimum and Q1 values.

Finally, and in order to understand these results better, Table 10 shows the mean of the values obtained for each class using the Convolutional Neural Network with both types of images (MM: Matrix Multiplication, COV: covariance matrix).

All the classes obtain high mean accuracy values, which vary between 0.85 and - 0.96. For the matrix multiplication images the best value is obtained by the *Sweet-milk* class (0.9601) and the worst by *Find* (0.8853), whereas for the covariance matrix images the best value is obtained by the *Opaque* class (0.9504) and the worst by the *Red2* class (0.8703). In short, the results obtained for all of the classes are similar and no matrix method stands out.

TABLE 10. Mean accuracy values obtained for each class of LSA64 data set with CNN.

	Opaque	**Red**	**Greeen**	**Yellow**	**Bright**	**Light-blue**
MM	0.9069	0.9475	0.9209	0.9038	0.9401	0.9438
COV	0.9504	0.9283	0.9208	0.9075	0.9416	0.9209
	Colors	**Red 2**	**Women**	**Enemy**	**Son**	**Man**
MM	0.9157	0.9259	0.9356	0.9186	0.9149	0.9319
COV	0.9216	0.8703	0.9416	0.9193	0.9430	0.8868
	Away	**Drawer**	**Born**	**Learn**	**Call**	**Skimmer**
MM	0.9119	0.9261	0.9208	0.9231	0.9061	0.9379
COV	0.9290	0.9349	0.8831	0.9141	0.9157	0.9372
	Bitter	**Sweet-milk**	**Milk**	**Water**	**Food**	**Argentina**
MM	0.9091	0.9601	0.9172	0.9222	0.9348	0.9341
COV	0.9148	0.9246	0.9111	0.9331	0.8854	0.9172
	Uruguay	**Country**	**Last name**	**Where**	**Birthday**	**Hungry**
MM	0.9453	0.9387	0.9401	0.9483	0.9172	0.9194
COV	0.9335	0.9275	0.9157	0.9067	0.9163	0.9105
	Ship	**None**	**Name**	**Patience**	**Perfume**	**Deaf**
MM	0.9460	0.9326	0.9164	0.9216	0.9261	0.9200
COV	0.9231	0.9260	0.9238	0.9319	0.9305	0.9194
	Candy	**Chewing-gum**	**Shut down**	**Buy**	**Realize**	**Find**
MM	0.9334	0.9326	0.9318	0.9446	0.9131	0.8853
COV	0.9297	0.9083	0.9097	0.8934	0.8788	0.9128

VI. CONCLUSION AND FUTURE WORK

In this paper a new pipeline for action recognition is presented, which has been applied to two different tasks in this domain: activity recognition and sign language recognition. In the presented approach the Common Spatial Patterns method has been applied to signals created from the positions of the skeleton joints of people performing different actions or signs. From the output of the method some images have been created, which have been then classified. In the classification step two approaches have been tested; one based on Visual Global Descriptors and the other a CNN implementation. The obtained results have been compared to a previous approach by the same authors and also to those obtained by a LSTM, a well-known deep learning method.

As further work, we plan to extend the range of human activities, as well as to implement the presented method in the actual robot. This would allow the robot to react to different actions performed in front of it, or to communicate with people with hearing impairments. Applications in Social Robotics are also to be developed, being this the next envisaged step.

Concerning the sign language recognition, several steps have been identified that would improve the presented method. Facial information of the signers should be added, since it is a crucial feature when interpreting sign language. Signs which use both hands should also be considered, in order to make the recognition system more complete.

On the classification step, other image descriptors could also be used, and in that case a feature subset selection step could be advisable.

ACKNOWLEDGMENT

The authors would like to thank the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, and B. Sierra, "Sign language recognition by means of common spatial patterns," in *Proc. 5th Int. Conf. Mach. Learn. Soft Comput. (ICMLSC)*, Jan. 2021, pp. 96–102.

[2] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, I. Rodriguez-Rodriguez, and B. Sierra, "Shedding light on people action recognition in social robotics by means of common spatial patterns," *Sensors*, vol. 20, no. 8, p. 2436, Apr. 2020.

[3] K. Keng Ang, Z. Yang Chin, H. Zhang, and C. Guan, "Filter bank common spatial pattern (FBCSP) in brain-computer interface," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intelligence)*, Jun. 2008, pp. 2390–2397.

[4] S. Sethi, R. Upadhyay, and H. S. Singh, "Stockwell-common spatial pattern technique for motor imagery-based brain computer interface design," *Comput. Electr. Eng.*, vol. 71, pp. 492–504, Oct. 2018.

[5] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[6] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A framework for building perception pipelines," 2019, *arXiv:1906.08172*. [Online]. Available: http://arxiv.org/abs/1906.08172

[7] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[8] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, 2004, pp. 32–36.

[9] J. C. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.

[10] C.-C. Chen and J. K. Aggarwal, "Recognizing human action from a far field of view," in *Proc. Workshop Motion Video Comput. (WMVC)*, Dec. 2009, pp. 1–7.

[11] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and Binet–Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1932–1939.

[12] F. Ercis, "Comparison of histogram of oriented optical flow based action recognition methods," M.S. thesis, Dept. Elect. Electron. Eng., Middle East Tech. Univ., Ankara, Turkey, 2012.

[13] K. Lertniphonphan, S. Aramvith, and T. H. Chalidabhongse, "Human action recognition using direction histograms of optical flow," in *Proc. 11th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Oct. 2011, pp. 574–579.

[14] S. Akpinar and F. N. Alpaslan, "Video action recognition using an optical flow based representation," in *Proc. Int. Conf. Image Process., Comput. Vis., Pattern Recognit. (IPCV)*. Las Vegas, NV, USA: The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2014, p. 1.

[15] S. Satyamurthi, J. Tian, and M. C. H. Chua, "Action recognition using multi-directional projected depth motion maps," *J. Ambient Intell. Hum. Comput.*, pp. 1–7, Nov. 2018.

[16] M. Liu, H. Liu, and C. Chen, "Robust 3D action recognition through sampling local appearances and global distributions," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 1932–1947, Aug. 2018.

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[18] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[19] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.

[20] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream ConvNets," 2015, *arXiv:1507.02159*. [Online]. Available: http://arxiv.org/abs/1507.02159

[21] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[22] S. Arif, J. Wang, T. Ul Hassan, and Z. Fei, "3D-CNN-based fused feature maps with LSTM applied to action recognition," *Future Internet*, vol. 11, no. 2, p. 42, Feb. 2019.

[23] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 20–36.

[24] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 803–818.

[25] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 7083–7093.

[26] S. Sudhakaran, S. Escalera, and O. Lanz, "Gate-shift networks for video action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1102–1111.

[27] D. Tran, H. Wang, M. Feiszli, and L. Torresani, "Video classification with channel-separated convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5552–5561.

[28] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou, "Temporal pyramid network for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 591–600.

[29] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," 2016, *arXiv:1611.06067*. [Online]. Available: http://arxiv.org/abs/1611.06067

[30] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "PoTion: Pose motion representation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7024–7033.

[31] J. Ren, N. H. Reyes, A. L. C. Barczak, C. Scogings, and M. Liu, "An investigation of skeleton-based optical flow-guided features for 3D action recognition using a multi-stream CNN model," in *Proc. IEEE 3rd Int. Conf. Image, Vis. Comput. (ICIVC)*, Jun. 2018, pp. 199–203.

[32] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 12026–12035.

[33] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 1, pp. 131–153, 2019.

[34] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. B. Lakulu, "A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017," *Sensors*, vol. 18, no. 7, p. 2208, 2018.

[35] A. Wadhawan and P. Kumar, "Sign language recognition systems: A decade systematic literature review," *Arch. Comput. Methods Eng.*, vol. 28, pp. 785–813, May 2021.

[36] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113794.

[37] K. Kudrinko, E. Flavin, X. Zhu, and Q. Li, "Wearable sensor-based sign language recognition: A comprehensive review," *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 82–97, 2021.

[38] P. D. Rosero-Montalvo, P. Godoy-Trujillo, E. Flores-Bosmediano, J. Carrascal-García, S. Otero-Potosi, H. Benitez-Pereira, and D. H. Peluffo-Ordóñez, "Sign language recognition based on intelligent glove using machine learning techniques," in *Proc. IEEE 3rd Ecuador Tech. Chapters Meeting (ETCM)*, Oct. 2018, pp. 1–5.

[39] M. I. Sadek, M. N. Mikhael, and H. A. Mansour, "A new approach for designing a smart glove for Arabic sign language recognition system based on the statistical analysis of the sign language," in *Proc. 34th Nat. Radio Sci. Conf. (NRSC)*, Mar. 2017, pp. 380–388.

[40] C. K. M. Lee, K. K. H. Ng, C.-H. Chen, H. C. W. Lau, S. Y. Chung, and T. Tsoi, "American sign language recognition and training method with recurrent neural network," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114403.

[41] J. J. Bird, A. Ekárt, and D. R. Faria, "British sign language recognition via late fusion of computer vision and leap motion with transfer learning to American sign language," *Sensors*, vol. 20, no. 18, p. 5151, Sep. 2020.

[42] T.-W. Chong and B.-G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors*, vol. 18, no. 10, p. 3554, Oct. 2018.

[43] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A modified LSTM model for continuous sign language recognition using leap motion," *IEEE Sensors J.*, vol. 19, no. 16, pp. 7056–7063, Aug. 2019.

[44] D. Li, C. R. Opazo, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Mar. 2020, pp. 1459–1469.
[45] M. Parelli, K. Papadimitriou, G. Potamianos, G. Pavlakos, and P. Maragos, "Exploiting 3D hand pose estimation in deep learning-based sign language recognition from RGB videos," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 249–263.
[46] M. Jebali, A. Dakhli, and M. Jemni, "Vision-based continuous sign language recognition using multimodal sensor fusion," *Evolving Syst.*, pp. 1–14, Jan. 2021.
[47] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," in *Proc. Conf. Signal Process. Commun. Eng. Syst. (SPACES)*, Jan. 2018, pp. 194–197.
[48] N. Basnin, L. Nahar, and M. S. Hossain, "An integrated CNN-LSTM model for Bangla lexical sign language recognition," in *Proc. Int. Conf. Trends Comput. Cogn. Eng.* Singapore: Springer, 2021, pp. 695–707.
[49] R. Rastgoo, K. Kiani, and S. Escalera, "Hand sign language recognition using multi-view hand skeleton," *Expert Syst. Appl.*, vol. 150, Jul. 2020, Art. no. 113336.
[50] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1311–1325, 2018.
[51] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "Coupled HMM-based multi-sensor data fusion for sign language recognition," *Pattern Recognit. Lett.*, vol. 86, pp. 1–8, Jan. 2017.
[52] S. G. Azar and H. Seyedarabi, "Trajectory-based recognition of dynamic Persian sign language using hidden Markov model," *Comput. Speech Lang.*, vol. 61, May 2020, Art. no. 101053.
[53] A. Bastanfard, N. A. Rezaei, M. Mottaghizadeh, and M. Fazel, "A novel multimedia educational speech therapy system for hearing impaired children," in *Proc. Pacific-Rim Conf. Multimedia*. Berlin, Germany: Springer, 2010, pp. 705–715.
[54] K. Fukunaga and W. L. G. Koontz, "Application of the Karhunen–Loève expansion to feature selection and ordering," *IEEE Trans. Comput.*, vol. C-19, no. 4, pp. 311–318, Apr. 1970.
[55] Y. Wang, S. Gao, and X. Gao, "Common spatial pattern method for channel selection in motor imagery based brain-computer interface," in *Proc. IEEE Eng. Med. Biol. 27th Annu. Conf.*, Jan. 2006, pp. 5392–5395.
[56] Q. Novi, C. Guan, T. H. Dat, and P. Xue, "Sub-band common spatial pattern (SBCSP) for brain-computer interface," in *Proc. 3rd Int. IEEE/EMBS Conf. Neural Eng.*, May 2007, pp. 204–207.
[57] T. N. Alotaiby, S. A. Alshebeili, L. M. Aljafar, and W. M. Alsabhan, "ECG-based subject identification using common spatial pattern and SVM," *J. Sensors*, vol. 2019, pp. 1–9, Mar. 2019.
[58] P. Kim, K.-S. Kim, and S. Kim, "Using common spatial pattern algorithm for unsupervised real-time estimation of fingertip forces from sEMG signals," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 5039–5045.
[59] X. Li, P. Fang, L. Tian, and G. Li, "Increasing the robustness against force variation in EMG motion classification by common spatial patterns," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 406–409.
[60] J. Shapiro, D. Savransky, J.-B. Ruffio, N. Ranganathan, and B. Macintosh, "Detecting planets from direct-imaging observations using common spatial pattern filtering," *Astron. J.*, vol. 158, no. 3, p. 125, Aug. 2019.
[61] I. Mendialdua, J. M. Martínez-Otzeta, I. Rodriguez-Rodriguez, T. Ruiz-Vazquez, and B. Sierra, "Dynamic selection of the best base classifier in one versus one," *Knowl.-Based Syst.*, vol. 85, pp. 298–306, Sep. 2015.
[62] F. Ronchetti, F. Quiroga, C. A. Estrebou, L. C. Lanzarini, and A. Rosete, "LSA64: An Argentinian sign language dataset," in *Proc. XXII Congreso Argentino de Ciencias de la Computación (CACIC)*, 2016, pp. 794–803.
[63] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. 6th ACM Int. Conf. Image Video Retr.*, 2007, pp. 401–408.
[64] S. A. Chatzichristofis and Y. S. Boutalis, "FCTH: Fuzzy color and texture histogram—A low level feature for accurate image retrieval," in *Proc. 9th Int. Workshop Image Anal. Multimedia Interact. Services*, 2008, pp. 191–196.
[65] C. S. Won, D. K. Park, and S.-J. Park, "Efficient use of MPEG-7 edge histogram descriptor," *ETRI J.*, vol. 24, no. 1, pp. 23–30, 2002.

ITSASO RODRÍGUEZ-MORENO received the B.Sc. and M.Sc. degrees in computer science from the University of the Basque Country, in 2018 and 2019, respectively, where she is currently pursuing the Ph.D. degree with the Department of Computer Sciences and Artificial Intelligence, under a FPU Grant from the Spanish Government.

She is currently a member of the Robotics and Autonomous Systems Group. Her research interests include machine learning, computer vision, and robotics.

JOSÉ MARÍA MARTÍNEZ-OTZETA received the B.Sc. and Ph.D. degrees in computer science from the University of the Basque Country, in 1993 and 2008, respectively.

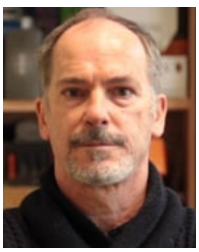
He is currently a Postdoctoral Researcher with the Department of Computer Sciences and Artificial Intelligence, University of the Basque Country. He is also a member of the Robotics and Autonomous Systems Group. His research interests include machine learning, computer vision, and robotics.

IZARO GOIENETXEA received the B.Sc. degree in computer science from the University of the Basque Country, Spain, in 2008, and the Ph.D. degree from the Department of Computer Science and Artificial Intelligence, University of the Basque Country, in 2019.

She is currently a member of the Robotics and Autonomous Systems Research Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country. Her current research interests include music generation and classification, computer vision, and machine learning.

IGOR RODRIGUEZ received the B.Sc. and Ph.D. degrees in computer science from the University of the Basque Country, in 2012 and 2018, respectively.

He is currently a member of the Robotics and Autonomous Systems Research Group, Department of Computer Science and Artificial Intelligence. His current research interests include robotics, human–robot interaction, and machine learning.



BASILIO SIERRA received the B.Sc. degree in computer sciences, the M.Sc. degree in computer science and architecture, and the Ph.D. degree in computer sciences from the University of the Basque Country, Donostia-San Sebastian, Spain, in 1990, 1992, and 2000, respectively.

He is currently a Full Professor with the Department of Computer Sciences and Artificial Intelligence, University of the Basque Country. He is also the Co-Director of the Robotics and Autonomous Systems Group. His research interests include robotics and machine learning, where he is working on the use of different paradigms to improve behaviors.