Efficient Hand Gesture Recognition for Human-Robot Interaction

Marc Peral[1], Alberto Sanfeliu[1] and Anaís Garrell[1]

***Abstract*—In this paper, we present an efficient and reliable deep-learning approach that allows users to communicate with robots via hand gesture recognition. Contrary to other works which use external devices such as gloves [1] or joysticks [2] to tele-operate robots, the proposed approach uses only visual information to recognize user's instructions that are encoded in a set of pre-defined hand gestures. Particularly, the method consists of two modules which work sequentially to extract 2D landmarks of hands –ie. joints positions– and to predict the hand gesture based on a temporal representation of them. The approach has been validated in a recent state-of-the-art dataset where it outperformed other methods that use multiple pre-processing steps such as optical flow and semantic segmentation. Our method achieves an accuracy of 87,5% and runs at 10 frames per second. Finally, we conducted real-life experiments with our IVO robot to validate the framework during the interaction process.**

I. Introduction

Hand gestures and sign language have been used as a way to express and communicate feelings and thoughts between humans. Precisely, sign language is a well-known structured system which uses hand gestures and signs. This language serves as a useful tool for the daily interaction of deaf and speech-impaired community. Sign language requires the use of different parts of the body, such us hands, fingers, arms or facial expression to convey information [3].

Previously, the communication between humans and robots was achieved using a keyboard or a touch-screen where directives were given to the robot [4]. However, in HRI (Human-Robot Interaction), robot's ability to collaborate naturally with people in a human-centered environment is a crucial aspect. Humans do not cooperate using touch-screens, their interaction requires the recognition and interpretation of speech, gesture or emotion [5]. Therefore, the interest on non-touch-based methods are catching on recently.

Speech recognition is one of the most convenient methods, nevertheless, it has some problems due to the large variety of human accents and failing in out-of-control noisy situations. Alternatively, there exist vision methods that use facial expression, eye tracking or head movements [6]; but the most understood one is gesture recognition. Vision-based HRI technology is a non-touch method that is capable of expressing the most complex information [7].

Fig. 1. The proposed approach is used to interact with robots via hand gesture recognition.

From all human body gestures it is natural to focus on hands, as they are intuitively used in natural human to human communication. The main reason to pursue this touch-less methods is to create an engagement between robots an humans, and thus, achieve a natural interaction between them. Although hand gesture recognition is being deeply studied [8], [9], it has some challenges yet to overcome, like complex and moving backgrounds or changes on illumination conditions [10]. In this paper, we focus on the recognition of hand gestures as a way to communicate with robots on an easy and natural manner. It is crucial for robotics applications such as human-robot interaction or assisted robotics, where the interaction should be fluid, effective, and as less invasive as possible [11]. To this end, we propose an approach based on deep learning, called EUREKA (gEstUre REcognition Key frAmes), see Fig 1, that recognizes hand gestures in images using a social robot.

An overview of this approach is shown in Fig. 2. Specifically, it consists of two modules. The first one is a visual detector which returns the 2D positions of hand landmarks –joints– in the image, see Fig. 2-(b,c). In this work, we rely on the Mediapipe landmark detector that has shown exceptional results in terms of efficiency and accuracy [12], [13]. The second module extracts a temporal representation of the detected landmarks and predicts the hand gesture using a densely connected network, see Fig. 2-(d). This straightforward but effective approach runs in real time and is robust to uncontrolled environments with varying backgrounds and lighting conditions.

Finally, our method has been evaluated extensively in the IPN Hand dataset [14] where it has shown remarkable results in comparison to more complex works in the state of the art. Experiments with our mobile robot were also conducted to validate the framework for human-robot interaction.

The contribution of this paper is threefold: first, a deep learning approach for hand gesture recognition called EU-

Manuscript received: February 24, 2022; Revised: May 23, 2022; Accepted: June 27, 2022.

This paper was recommended for publication by Editor Gentiane Venture upon evaluation of the Associate Editor and Reviewers'comments.

[1]Marc Peral, Alberto Sanfeliu and Anaís Garrell are with the Institut de Robòtica i Informàtica Industrial (CSIC-UPC). Llorens Artigas 4-6, 08028, Barcelona, Spain. {marc.peral, alberto.sanfeliu, anais.garrell}@upc.edu

Work supported under the Spanish State Research Agency through the ROCOTRANSP project (PID2019-106702RB-C21 / AEI / 10.13039/501100011033)) and the EU project CANOPIES (H2020-ICT-2020-2-101016906)

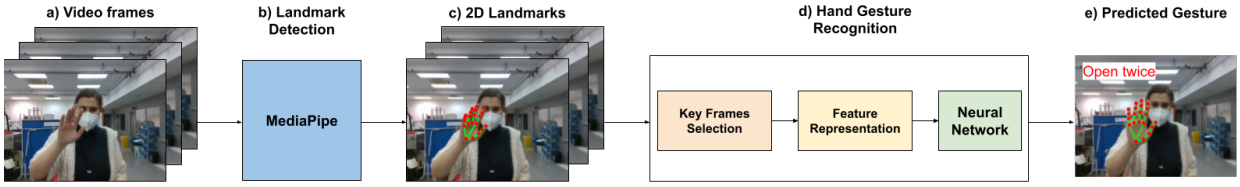

Fig. 2. Overview of the proposed approach for hand gesture recognition. Given a video sequence of images (a), the proposed method uses the Mediapipe pose detector (b) to localize landmarks of hands in each video frame (c). Subsequently, the detected landmarks are encoded temporally and spatially in a feature representation which is then used to perform hand gesture recognition by a densely connected network (d).

REKA was developed. It integrates hand landmark detection with gesture classification in a two-step approach; second, we conducted some ablation studies over the IPN Hand dataset [14] where we report significant improvement over the state-of-the-art results after finetuning; and third, we define a novel batch approach to work on real-life situations, tested on continuous streams of data taken by our robot IVO.

The remainder of this paper is organized as follows. Section II presents some related work for human-robot interaction and hand gesture recognition. Section III describes the components of the proposed method: hand landmarks extraction and gesture recognition. In Section IV, the method is evaluated in a state-of-the-art dataset. Finally, some conclusions and future work is provided in Section V.

II. Related Work

In this section, we present some related work in the state of the art for human-robot communication using gestures, and for hand gesture recognition using deep-learning approaches.

A. Human-Robot Communication Using Gestures

Nowadays, robots are becoming increasingly ever-present in human environments such as homes, malls, hospitals, or schools. Hence, the new human environment requires them to be equipped with social intelligence, successfully qualifying them to assist, interact and collaborate with humans. The research field of human-robot interaction (HRI) has studied how humans and robots can communicate using gestures. Particularly, such research has placed a wide scope of social HRI applications like robots assistance [15], education [16], search and rescue [17], or tour guiding in cities [18].

Moreover, collaborative robotics is an emerging and multidisciplinary research field, in which gesture-based HRI is an active research topic. The use of hand gestures constitutes a natural form of communication among humans and can therefore be an effective method for natural and accessible HRI. In addition to that, hand gestures might offer a better alternative to speech recognition to overcome challenges such as noise, reverberation and distant speech [19]. As a result, hand gesture recognition has been thoroughly studied in the field of HRI.

It has been demonstrated that gestures are one of the most effective and natural mechanisms for reliable HRI [20] as they encourage a natural interaction process. In the HRI context, they have been used for robot tele-operation [21], or to coordinate the interaction process and cooperation activities between human and robot [22].

B. Deep Learning applied to Human-Robot Communication

As stated in [23], non-machine-learning approaches for hand gesture recognition face unstable accuracy problems regarding different light environments and gesture overlapping, making machine-learning algorithms more flexible and able to adapt to real exigent situations.

Research on gestures detection using neural networks has been vastly explored and one could say it is reliable, but when it comes to develop a real-life working method for natural human-robot communication the temporal component supposes the biggest challenge.

Gao et al. [24] designed a method that fusing 2D and 3D fast hand estimations gets a dynamic gesture recogniser with high accuracy. In the videos used for their experimentation, the user began moving the hand into the image and ended removing it. Zhang et al. [25] achieved great results approaching the temporal component with bidirectional ConvLSTM and 3DCNN, and then, fusing them at a higher level from 2DCNN. The use of LSTM based RNN was proven suitable for the various length videos, nevertheless gestures in databases used in their work, ChaLearn IsoGD [26] and SKIG [27], were isolated so each video contained exactly one gesture.

Isolated gesture classifiers are not actually applicable to real-life situations, as what the computer sees is not isolated. This paper aims to get down to real life situations where what the computer analyzes are successions of gestures without starting and ending labeling or signaling.

There has been some recent research on continuous gesture recognition. Benitez-Garcia et al. [14] created a dataset with videos each containing 21 successive gesture instances, see Section IV-A. Authors used a two-stage approach, at first they slide a gesture detector through the video frames and, whenever a gesture is detected, the gesture classifier makes a prediction on the gesture that has been performed. In [28], Gammulle et al. designed a single-stage framework that used distinct feature extraction methods, and then fused these extracted features to make a prediction on each video frame.

EUREKA establishes a gesture recognition method that combines a feature extractor with a neural network that, with the help of advanced pre-processing techniques, outperforms other state-of-the-art methods. Our work also brings up a pioneering batch approach to face real-life situations with continuous gestures.

III. Proposed Method

In this section we describe the proposed method, EUREKA, which is depicted in Fig. 2. It consists of two main modules

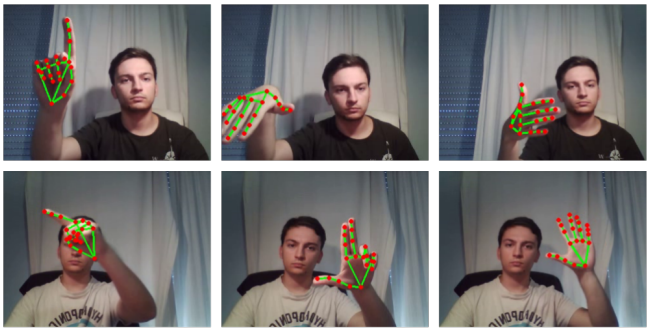

Fig. 3. Example images showing the output of the landmarks detector.

which work together to predict efficiently hand gestures. The first module focuses on the 2D localization of landmarks –i.e. hand joints– in images, while the second one predicts the hand gesture based on a temporal and normalized representation of the detected landmarks. In the following, these modules are explained in more detail.

A. Hand landmarks extraction

The extraction of landmarks is done efficiently by the Mediapipe pose detector which is a cross-platform open source solution that uses machine learning algorithms to track hands on color images [12]. Some example images are shown in Fig. 3. Note that it is able to localize hand landmarks under varying pose configurations. This detector consists of two steps. First, a palm detector is run over the full input image to localize palms using oriented bounding boxes.

Then, a hand model is applied to the extracted bounding boxes to return the 2D position of landmarks associated to 21 joints of a human hand.

This two-step approach vastly reduces the need for data augmentation through its oriented hand bounding box which eludes rotation, translation and scale problems. It allows the landmark predictor to focus mainly on localisation accuracy. Moreover, to speed up the detection process, the approach uses previous landmarks predictions in order to avoid running the palm detector in every frame. This results in a real-time method which is able to detect multiple hands.

B. Hand gesture recognition

The recognition of hand gestures is carried out by a network that uses the landmarks provided by the Mediapipe detector, observe Fig. 2-(c). The input to this network can be the raw landmark positions given in pixel coordinates. However, as we will see in the experiments section, computing a temporal and normalized representation of landmarks leads to better recognition rates. In this section, we introduce different feature representations and the network architecture for hand gesture recognition.

Key frames selection. In order to recognize hand gestures that include motion like waving, it is crucial to include temporal information at the time of performing classification with the network, see Fig. 2-(d). For this goal, we propose a frame selection strategy which combines the landmarks extracted from different frames. This allows to encode the spatial changes among landmarks positions and over time.

We start by defining a number of key frames M. Then, for each gesture instance in the dataset, M frames from that instance are selected. The selection is made by choosing M equally distanced frames in time (see Eq. 1) where L represents the total number of frames in the instance where the hand landmarks were detected.

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ ... \\ f_M \end{bmatrix} = \frac{L}{M} * \begin{bmatrix} 0 \\ 1 \\ 2 \\ ... \\ M-1 \end{bmatrix} \tag{1}$$

Note that the division L/M result is rounded to the nearest integer because the frame number must be an integer. We define $\hat{f}$ the vector that contains the indexes for the selected M frames of a certain gesture instance.

In cases where the number of frames with detected landmarks is smaller than the number of frames in the gesture instance ($L < M$), we proceeded to replicate the detected landmarks until reaching M.

Feature representation. For each video frame we have the 2D positions of 21 landmarks, what results in a 42-dimensional feature vector. When we consider multiple key frames, $M > 1$, the resulting feature vector has a size of 42*M values. This feature vector corresponds to the raw positions of the hand landmarks. We denote this vector as our raw feature representation Fig. 4-(a). In the following, we study different feature representations in order to provide some translation and scale invariance and achieve better classification results.

The second feature representation, called *Distances* Fig. 4-(b), computes the Euclidean distances among all landmarks in the same video frame. For example, let's consider a vector with 3 landmarks with x and y components. The distances computed will start with 1→2, then 1→3 and the last 2→3. Note that the distance from a point to itself is 0 and adds no information, so it is not considered in our experiments. Analog distances like 3→2, while having computed 2→3, are also ignored. Equation 2 can be used to obtain the input vector shape, where $n^{[0]}$ is the length of the input vector, l represents the number of landmarks and c the number of components for each of them.

$$n^{[0]} = \frac{l^2 - l}{2} \cdot c \cdot M \tag{2}$$

In the previous example, the *Raw* and *Distances* feature representations have vectors of size 6*M. However, when we use the full set of landmarks (21 landmarks) the feature vector has a size of 420*M values.

The next feature representation, denoted as *DistAndTime* Fig. 4-(c), takes into account the relative change of landmarks over time. It is computed by adding, to the *Distances* feature representation, the distance from each landmark to itself in the past frame. Coming back to the previous example with 3 landmarks the new vector will have distances 1→2, 1→3, 2→3, 1→1_{past}, 2→2_{past} and 3→3_{past}. Note that for these

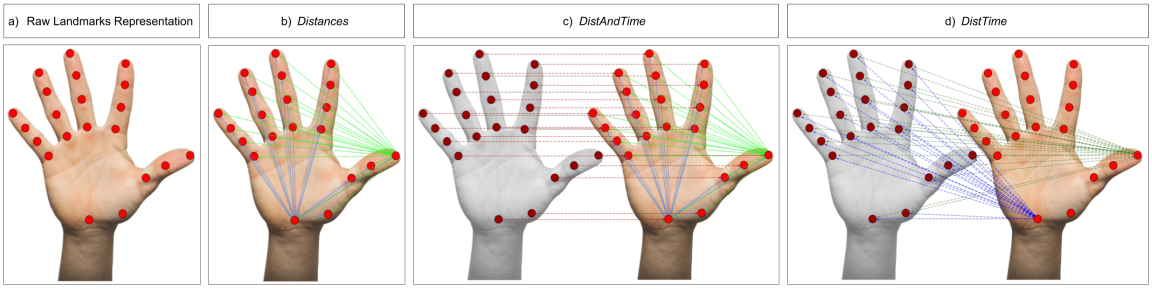

Fig. 4. Depiction of proposed feature representations. a) *Raw* Landmarks Representation: raw landmarks directly taken from the hand joints detector MediaPipe. b) *Distances*: computes the Euclidean distances among all landmarks in the same video frame, in the figure simplified for only 2 of the landmarks. c) *DistAndTime*: computed by adding, to the *Distances* feature representation, the distance from each landmark to itself in the past frame. d) *DistTime*: computed by calculating distances from each of the points in the present frame to each of the points in the past frame, in the figure simplified for only 2 of the landmarks.

operations at least two frames are needed, the present and the past one. As a consequence, this feature representation ends up with a vector's length of $(M-1)$.

$$n^{[0]} = \left(\frac{l^2 - l}{2} \cdot c + l \cdot c\right) \cdot (M-1) \tag{3}$$

The feature vector for the full set of landmarks has a size of 462*$(M-1)$ values.

The last feature representation, called *DistTime* Fig. 4-(d), is a combination of previous ones. It is computed by calculating distances from each of the points in the present frame to each of the points in the past frame. In this representation the distance from $1 \rightarrow 1_{past}$ is no longer 0 and $2 \rightarrow 3_{past}$ is not analog to $3 \rightarrow 2_{past}$.

The following equation is used to compute the size of the feature vector.

$$n^{[0]} = l^2 \cdot c \cdot (M-1) \tag{4}$$

For the 21*M hand landmarks the input vector's length after this representation is 882*$(M-1)$.

Network. Once the feature vector is computed, we use a neural network consisting of densely connected layers to perform hand gesture classification (MLP). That is, the feature vector is classified in one of the gesture classes defined in the database.

More specifically, the network has four layers in which the size of the input layer is set according to the length of the feature representation. Next, the network has two hidden layers with ReLU activation functions. The number of neurons in each layer is tested and evaluated in the experiments section, but we define a baseline network which has 64 and 32 neurons in the first and second hidden layers respectively. The output layer uses softmax activation function to predict the gesture. Dropout and batch normalization is also applied to avoid overfitting.

IV. Experiments

In this section, first the dataset used for the state-of-the-art experiments is described. Next, those tests reaching the best model for EUREKA are presented. At the end of the section, real-life experiments with robot IVO are described.

TABLE I
EXPERIMENT IN WHICH SOME DESIGNED FEATURE REPRESENTATIONS SUPPOSE AN UPGRADE IN FRONT OF USING RAW DATA

Method	Accuracy
Raw	0.831
Distances	0.831
DistTime	**0.841**
DistAndTime	**0.845**

TABLE II
EXPERIMENT SHOWS THAT ADDING A 25% DROPOUT AFTER EACH HIDDEN LAYER IS CONVENIENT TO OBTAIN BETTER ACCURACY AND AVOID OVERFITTING

Method	Dropout	Accuracy
DistTime	0.25	**0.870**
	0	0.847
DistAndTime	0.25	**0.862**
	0	0.831

A. Dataset

The IPN Hand dataset [14] contains 4218 instances of gesture instances but, notably, they are grouped in 200 videos. Each video includes 21 gesture instances, what lets us state that the dataset considers real-life continuous gestures performed without transitional states. It is also remarkable that, as in real-life, gesture instances last different amounts of time despite having the same gesture performed. Therefore, the minimum length of a gesture in the dataset is 9 frames and the maximum is 650 frames. All videos were recorded at 30 frames per second.

Furthermore, the videos, that are only 2D RGB videos, feature 50 subjects with static or dynamic backgrounds, lighter or darker illumination conditions, and interacting with either right or left hand. The dataset includes 2 pointing gestures, 11 performing actions and a non-gesture class, making up a total of 14 gesture classes. A non-gesture class is crucial so that the system leans to distinguish the addressed gestures from other natural gestures such as scratching the nose, drinking water, etc.

The training and testing sets are chosen in the same way as Benitez-Garcia et al. did, they randomly split into 74% training and 26% testing, resulting in 37 training subjects and 13 testing subjects, what is 3117 and 1101 training and testing gesture instances, respectively, so we chose the same subjects for the training and testing sets. Afterwards, 5 of the training subjects are randomly chosen to conform the validation set, that supposes a 10% of the overall dataset.

B. SOTA experiments

In this section we perform some experiments to study the impact of some choices of EUREKA on the hand gesture recognition performance.

The following experiments were run training the neural network with the preprocessed data using the training and validation sets. For each experiment, 5 tests were operated with the same input data and equal hyperparameters, then the average accuracy was computed and also its standard deviation. The following tables show the classification accuracy, where standard deviation values are not shown because they were lower than $0,001$ in all cases.

For each test, 3 models were saved: the default model obtained at the end of the training process, the one after the epoch that got the highest accuracy and the model that presented the lowest loss value. The model that subsequently obtained the best accuracy against the testing set was selected.

Firstly, we assess the different feature representation methods explained in Section III-B. In this work, in those cases where the Mediapipe detector is unable to detect hand landmarks, due mainly to fast movements and blurring effects, the feature representation removes those frames.

The first test directly gets, as the input data, the detected landmarks in each of the selected key frames. After this raw data test is done, the three representation methods are applied. Results in Table I imply that the *Distances* method does not entail an upgrade to the raw data models, but *DistTime* and *DistAndTime* do. Normally we would select *DistTime* as the best feature representation and carry on with it but, as *DistAndTime* accuracy value is near the best, both methods will be evaluated in the next experiments.

The next parameter to be analyzed is M, the number of key frames for each gesture instance. Starting from the base value of $M = 5$, tests were run using $M = 3$ and $M = 9$, a higher and a lower value. The outcome presented a higher accuracy on higher key frames selection, that is why some more tests were performed to determine the trend. The accuracy for each tested value of M can be seen in Figure 7. It shows that once $M = 11$ is reached, the accuracy growth experiments a saturation and stops increasing. At $M = 15$, *DistTime* method gets a higher accuracy than *DistAndTime*, but we'll keep analysing both feature representation methods as they've both reached similar accuracy and this $M = 15$ spike could be spurious.

Then, the number of key frames that were selected from each gesture instance in the dataset was fixed to $M = 15$.

The next experiments influence the neural network hyperparameters to get a better configuration of it. It started by finding out if the dropout applied after each hidden layer was functional, or rather disadvantageous. As Table II displays, dropping out nodes in the network helped increasing overall accuracy, in both tested feature representation routines, by regulating and avoiding overfit on the training data.

TABLE III
EXPERIMENT FINDS OUT THAT ADDING AN EXTRA HIDDEN LAYER WON'T INCREASE ACCURACY

Method	Hidden Layers	Accuracy
DistTime	2	**0.870**
	3	0.851
DistAndTime	2	**0.862**
	3	0.851

TABLE IV
EXPERIMENT SETS OUT THAT HAVING HIGHER NUMBER OF NODES IN FIRST HIDDEN LAYER IMPROVES ACCURACY

Method	Hidden Layers	Accuracy
DistTime	64-32	0.870
	32-64	0.855
	64-64	0.870
	128-64	**0.875**
DistAndTime	64-32	0.862
	32-64	0.850
	64-64	0.859
	128-64	**0.869**

After proving 25% dropout is convenient, we advanced to testing 2-hidden-layers network against 3-hidden-layers. The 3-hidden-layers architecture was $64-32-16$ nodes and the 2-hidden-layers one was the base architecture set before, $64-32$ nodes. Table III concludes that adding a hidden layer was not successful for neither of the feature representation techniques.

The last experiment that was carried out tested diverse node configurations for the two hidden layers of the neural network. The first test combination switched hidden layers order from the defined base network, conforming a $32 - 64$ node net. Also a symmetric $64-64$ and a bigger $128-64$ network were included in the experiment. Table IV exposes that $32-64$ was the lowest accuracy achieving distribution, $64-64$ and $64-32$ respond similarly and the $128-64$ configuration turned out to outperform the rest. This experiment shows that it is essential to have a high number of nodes on the first hidden layer, and it makes sense taking into account the huge length of the input data vectors after the feature representation preprocess, 13230 for *DistTime* and 6930 for *DistAndTime*.

After all these experiments, the optimal method parameters and neural network hyperparameters are rooted out. The ultimate configuration of the method will select 15 key frames for each gesture instance, will preprocess input data using *DistTime* as feature representation technique and its neural network will have a $128-64$ distribution in its 2 hidden layers with a 25% dropout after each of them.

This definitive method obtained an accuracy of 87,5%. Table V contains the comparison between state-of-the-art methods

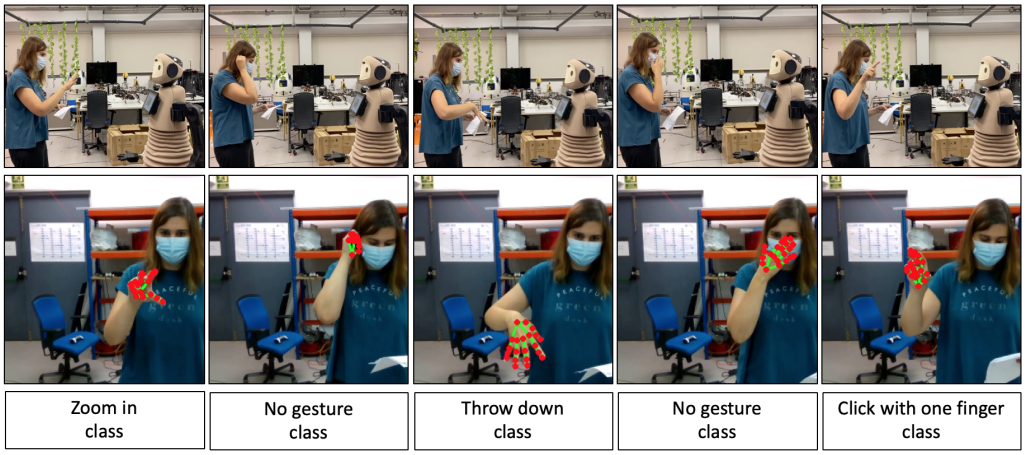

Fig. 5. EUREKA combined with a pioneering batch approach to continuous gestures identifies the hand-gestures performed in front of IVO. Note that when the user is scratching their ear or nose, the robot predicts there is not a command being executed.

TABLE V
COMPARING OUR METHOD'S ACCURACY WITH THE STATE-OF-THE-ART METHODS ON IPN HAND DATASET

Model	Accuracy
C3D	0.778
ResNeXt-101	0.836
ResNet-50	0.731
DistTime	**0.875**
DistAndTime	0.869

exposed in the IPN Hand database and EUREKA using the two overperforming feature representation techniques. One can observe that our method outran the accuracy obtained by Benitez-Garcia et al. in [14] either using *DistTime* or *DistAndTime*.

C. Experiments with the robot

This section unfolds the process followed to test the obtained state-of-the-art model on a real-life situation with the robot IVO, which is a robot designed for citizen assistance in public or outdoor spaces. IVO is equipped with an Intel Realsense D435i infrared camera which is installed in the head of the robot, this camera has been used to test our model with volunteers.

The application designed to recognise hand gestures in images that IVO sees will load the best-performing EUREKA model obtained in IV-B state-of-the-art experiments, the one that uses the *DistTime* feature representation, 15 key frames and a $128-64$ composition in its neural network hidden layers with a 25% dropout after each of them.

The key frames selection is tricky in continuous gesture recognition because the time that a gesture lasts is not known. That is why a batch approach is executed. This approach consists of taking different sized batches containing each a distinct number of frames, then the model emits a prediction for each of the taken batches and the prediction with a higher score will be the trusted one.

To exemplify this batch approach let's imagine there is a gesture instance that lasts 50 frames, like in Fig. 6. If it was in an isolated gesture dataset our method would take 15 from those 50 frames and select them as key frames to do the evaluation. In real-life continuous gesture instances, the batch approach takes for instance 3 batches with 30, 60 and 90 frames. On each of the 3 batches the 15 key frames are selected and evaluated to get 3 model predictions. In the exposed case the 60 frame batch would return the best score prediction, because the 30 frame batch would get part of a gesture and the 90 frame batch obtained a mix of more than a gesture instance.

Figure 5 portraits how the batch approach working with EUREKA recognise gestures (or non-gestures) included in the IPN Hand dataset classes performed in front of the robot IVO. The app runs at 10 frames per second using an unoptimized code. The following subsection presents the developed user study to analyze people perception during the experimentation procedure.

D. User Study

The results presented in the previous section demonstrate that the robot is able to detect and recognize human gesture. A user study was also conducted to determine whether the hand gesture recognition to control our robot enhances the usability and the comfort of the robot from the point of view of the human. We compared our method with the use of a remote controller.

The hypothesis we endeavored to test was as follows: "Participants will perceive difference between the use of hand gesture recognition and the use of a remote controller."

In the first experiment, the human had to use the hand gestures recognition to express to the robot what action must perform. We conducted these experiments in a Wizard-of-Oz way, since using the gesture detector may lead to missing some of the gestures, and it can cause a negative impact on the user perception of gestures.

Then, we repeated the same experiment but this time we gave a remote controller to the human, thus they could teleoperate the robot after some instruction.

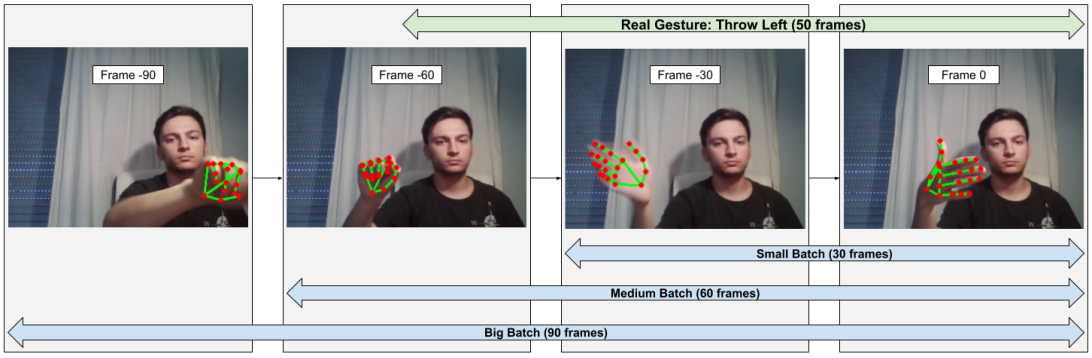

Fig. 6. Example of the batch approach. 3 batches with 30, 60 and 90 frames are taken. On each of the 3 batches, the key frames are selected and evaluated to get 3 model predictions. In the exposed case the 60 frame batch would return the best score prediction, because the 30 frame batch would get part of a gesture and the 90 frame batch obtained a mix of more than a gesture instance.

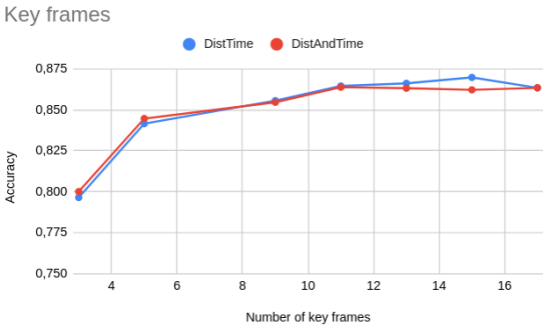

Fig. 7. Testing different amounts of key frames selected for each gesture instance.

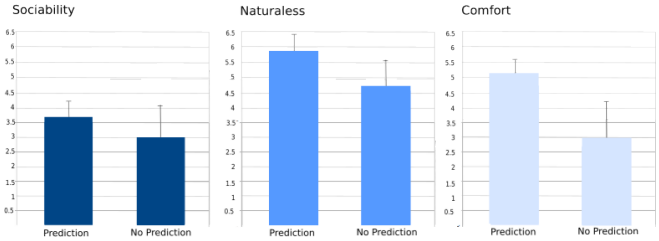

Fig. 8. Evaluation from 1 (low) to 7 (high) of the main aspects related to the robot behavior in hand gesture recognition.

In each case, we asked the volunteers to make all gestures/commands in a random order. We also chose randomly between the gesture communication and the controller as the first experiment for each volunteer in order to avoid possible biases.

For the experiments, we selected 14 people (8 men, 6 women) on the University Campus. Participants ranged in age from 22 to 50 years (M=26.2, SD=10.3), and represented a variety of University majors and occupations including computer science, mathematics, physics, and chemistry. For each selected participant, we randomly activated one of the two behaviors. It should be mentioned that none of the participants had previous experience working or interacting with robots.

Participants were asked to complete a questionnaire. Our independent variables considered whether participant make use of our hand gesture recognition or the remote control. The main dependent variables involved participants' perceptions of the **sociability**, **naturalness**, and **comfort** characteristics. Each of these fields was evaluated by every participant using a questionnaire to fill out after the experiment, based on [29].

Participants were asked to answer a questionnaire, following their encounter with the robot in each mode of behavior. To analyze their responses, we grouped the survey questions into four scales: the first measured sociability robot behavior, while the second naturalness, and the last one evaluated the comfort. Both scales surpassed the commonly used 0.7 level of reliability (Cronbach's alpha).

Each scale response was computed by averaging the results of the survey questions comprising the scale. ANOVAs were run on each scale to highlight differences between the three robot behaviors.

Below, we provide the results of comparing the two different methods. To analyze the source of the difference, four scores were examined: "sociability", "naturalness", and "comfort", plotted in Fig. 8. For all three aspects, the evaluation score plotted in Fig. 8, pairwise comparison with Bonferroni demonstrate there were difference between the two kind of behavior approaches, $p < 0.05$.

Therefore, after analyzing these three components, we may conclude that, if the robot is capable of understanding people's hand gesture, the acceptability of the robots increases and participants perceived the robot as a social entity.

Acknowledgments: The authors would like to thank to Michael Villamizar for the help and advice in the design of the proposed deep-learning approach based on landmarks.

V. Conclusions

This paper has introduced a new deep-learning framework, which allows humans to communicate with robots using hand gesture recognition. The proposed method only makes use of visual information to recognize humans' commands that are predefined in a hand gestures database.

Carried out experiments achieve an over-performing method that surpasses the state-of-the-art methods' accuracy. EUREKA uses *DistTime* as feature representation technique, a key frame selection procedure and a $128 - 64$ architecture in its neural network hidden layers with a 25% dropout after each of them.

Furthermore, the method is tested in real-life situations where gestures are continuous by using a batch approach that

successfully detects the gestures performed in front of the robot IVO.

The communication framework will be very useful for collaborative robots. Our experiments show that our framework is capable of interpreting human instructions with gestures. Furthermore, we are currently working on the design of more advanced deep learning methods with data fusion, and more real-life testing is being performed to enhance the refinement of the batch approach.

REFERENCES

[1] Y.-T. Hsieh, A. Jylhä, and G. Jacucci, "Pointing and selecting with tactile glove in 3d environment," in *International Workshop on Symbiotic Interaction*. Springer, 2015, pp. 133–137.

[2] A. Garrell, M. Villamizar, F. Moreno-Noguer, and A. Sanfeliu, "Teaching robot's proactive behavior using human assistance," *International Journal of Social Robotics*, vol. 9, no. 2, pp. 231–249, 2017.

[3] B.-W. Min, H.-S. Yoon, J. Soh, Y.-M. Yang, and T. Ejima, "Hand gesture recognition using hidden markov models," in *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 5, 1997, pp. 4232–4235.

[4] A. B. Usakli and S. Gurkan, "Design of a novel efficient human–computer interface: An electrooculagram based virtual keyboard," *IEEE transactions on instrumentation and measurement*, vol. 59, no. 8, pp. 2099–2108, 2009.

[5] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human–robot interaction," *Image and vision computing*, vol. 25, no. 12, pp. 1875–1884, 2007.

[6] J. C. Castillo, Á. Castro-González, F. Alonso-Martín, A. Fernández-Caballero, and M. Á. Salichs, "Emotion detection and regulation from personal assistant robot in smart environment," in *Personal assistants: Emerging computational technologies*. Springer, 2018, pp. 179–195.

[7] B. Chen, C. Hua, B. Dai, Y. He, and J. Han, "Online control programming algorithm for human–robot interaction system with a novel real-time human gesture recognition method," *International Journal of Advanced Robotic Systems*, vol. 16, no. 4, p. 1729881419861764, 2019.

[8] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial intelligence review*, vol. 43, no. 1, pp. 1–54, 2015.

[9] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE transactions on multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[10] Q. Gao, J. Liu, Z. Ju, Y. Li, T. Zhang, and L. Zhang, "Static hand gesture recognition with parallel cnns for space human-robot interaction," in *International Conference on Intelligent Robotics and Applications*. Springer, 2017, pp. 462–473.

[11] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[12] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[13] Kazuhito, "Hand gesture recognition using mediapipe," 2020. [Online]. Available: https://github.com/Kazuhito00/hand-gesture-recognition-using-mediapipe/blob/main/README_EN.md

[14] G. Benitez-Garcia, J. Olivares-Mercado, G. Sanchez-Perez, and K. Yanai, "Ipn hand: A video dataset and benchmark for real-time continuous hand gesture recognition," in *2020 25th International Conference on Pattern Recognition*. IEEE, 2021, pp. 4340–4347.

[15] G. Nejat and M. Ficocelli, "Can i be of assistance? the intelligence behind an assistive robot," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3564–3569.

[16] B. Scassellati, L. Boccanfuso, C.-M. Huang, M. Mademtzi, M. Qin, N. Salomons, P. Ventola, and F. Shic, "Improving social skills in children with asd using a long-term, in-home social robot," *Science Robotics*, vol. 3, no. 21, 2018.

[17] R. R. Murphy and J. L. Burke, "Up from the rubble: Lessons learned about hri from search and rescue," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 3. SAGE Publications Sage CA: Los Angeles, CA, 2005, pp. 437–441.

[18] A. Garrell and A. Sanfeliu, "Cooperative social robots to accompany groups of people," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1675–1701, 2012.

[19] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6989–6993.

[20] P. Neto, M. Simão, N. Mendes, and M. Safeea, "Gesture-based human-robot interaction for human assistance in manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 101, no. 1, pp. 119–135, 2019.

[21] W. Qi, S. E. Ovur, Z. Li, A. Marzullo, and R. Song, "Multi-sensor guided hand gesture recognition for a teleoperated robot using a recurrent neural network," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 6039–6045, 2021.

[22] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin, "Effects of nonverbal communication on efficiency and robustness in human-robot teamwork," in *2005 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2005, pp. 708–713.

[23] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed, R. Damaševičius, R. Maskeliūnas, and K. H. Abdulkareem, "Real-time hand gesture recognition based on deep learning yolov3 model," *Applied Sciences*, vol. 11, no. 9, p. 4164, 2021.

[24] Q. Gao, Y. Chen, Z. Ju, and Y. Liang, "Dynamic hand gesture recognition based on 3d hand pose estimation for human-robot interaction," *IEEE Sensors Journal*, 2021.

[25] L. Zhang, G. Zhu, P. Shen, J. Song, S. Afaq Shah, and M. Bennamoun, "Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3120–3128.

[26] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and S. Z. Li, "Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 56–64.

[27] L. Liu and L. Shao, "Learning discriminative representations from rgb-d video data," in *Twenty-third international joint conference on artificial intelligence*, 2013.

[28] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Multi-modal fusion for single-stage continuous gesture recognition," *arXiv preprint arXiv:2011.04945*, 2020.

[29] R. Kirby, *Social robot navigation*. Carnegie Mellon University, 2010.