Noname manuscript No.
(will be inserted by the editor)

Snapture - A Novel Neural Architecture for Combined Static and Dynamic Hand Gesture Recognition

Hassan Ali* · Doreen Jirak · Stefan Wermter

Received: date / Accepted: date

Abstract As robots are expected to get more involved in people's everyday lives, frameworks that enable intuitive user interfaces are in demand. Hand gesture recognition systems provide a natural way of communication and, thus, are an integral part of seamless Human-Robot Interaction (HRI). Recent years have witnessed an immense evolution of computational models powered by deep learning. However, state-of-the-art models fall short in expanding across different gesture domains, such as emblems and co-speech. In this paper, we propose a novel hybrid hand gesture recognition system. Our *Snapture* architecture enables learning both static and dynamic gestures: by capturing a so-called *snapshot* of the gesture performance at its peak, we integrate the hand pose along with the dynamic movement. Moreover, we present a method for analyzing the motion profile of a gesture to uncover its dynamic characteristics and which allows regulating a static channel based on the amount of motion. Our evaluation demonstrates the superiority of our approach on two gesture benchmarks compared to a CNNLSTM baseline. We also provide an analysis on a gesture class basis that unveils the potential of our *Snapture* architecture for performance improvements. Thanks to its modular implementation, our

H. Ali, S. Wermter
University of Hamburg
Knowledge Technology
Department of Informatics
Vogt-Kölln-Str. 30
22527 Hamburg
Germany
**Corresponding author:*
E-mail: 7ali@informatik.uni-hamburg.de

D. Jirak
IIT Central Research Labs Genova
Robotics Brain and Cognitive Sciences
Via Enrico Melen 83
16152 Genova
Italy

framework allows the integration of other multimodal data like facial expressions and head tracking, which are important cues in HRI scenarios, into one architecture. Thus, our work contributes both to gesture recognition research and machine learning applications for non-verbal communication with robots.

Keywords Co-Speech Gestures, Dynamic Gesture Recognition, Convolutional Neural Networks, Long Short-Term Memory

1 Introduction

Gestures are a form of non-verbal communication prominently used in day-to-day human communication. Additionally, they have become a fundamental part of human-robot interaction (HRI). It is common in the literature to categorize gestures as static and dynamic. Static gestures portray particular meanings through hand postures. They can substitute words or be used in harmony with them in the form of signs or emblems. These gestures can be recognized through a precise interpretation of the emphasized hand shape and spelled out finger arrangements [18]. In contrast, a dynamic gesture has a temporal aspect articulated through the movement of the hand. Therefore, recognizing it requires the employment of a different set of techniques, e.g., segmenting and tracking the moving body limb [2].

However, such distinction between gesture types might overlook some of their unique characteristics. More specifically, hand pose is essential for recognizing gestures that share a similar motion path. For example, the gesture commands "stop" and "go forward" have an identical motion with the arm moving from the body side and extending to the front. However, the specific meaning of each command can be distinguished by observing their unique hand and fingers arrangement (open palm vs. extended finger). Furthermore, a precise interpretation of the unique characteristics of each hand gesture is desired for a smooth HRI experience. This becomes more vital in critical robot applications, e.g., the medical or industrial domains in which the confusion between gestures might have severe consequences, such as safety risks, in case of misinterpretation of a robot command.

This kind of precise interpretation is challenging for approaches that use RGB data only. Recently, we have witnessed a rise of multimodal data, such as depth and audio, provided through intelligent sensors. Various multimodal challenges and datasets were proposed recently [8]. However, these devices impose certain operational conditions [21] limiting their flexibility. Therefore, RGB-based approaches are still desirable [6]. In addition to their convenience, they are potentially compatible with low-resource systems, such as robots. Furthermore, they facilitate the reproduction of results, especially as reproducibility issues related to deep learning are getting more attention by the scientific community [16]. Although recent developments were triggered by the deep learning trend with networks like 3DCNN, ResNet, and Inception V3, dynamic gesture recognition is still a challenging task.

State-of-the-art vision-based approaches are challenged by factors such as *indistinctive* and *subtle* movements [1]. *Subtle* movements refer to the small motion of the hand and fingers at the peak with no arm movement. On the other hand, *indistinctive* movements mean that multiple gestures follow a very similar path of motion. One limitation prominent in various state-of-the-art approaches is that they rely on the motion path [3]. Consequently, some approaches lack in the consideration of hand details, i.e., the exact hand shape and finger arrangements. This leads to misclassifications between gestures with similar motion properties. Moreover, it is worth inspecting whether integrating hand details into the classification would refine the performance of such models.

In this study, we propose a modular RGB-based approach called *Snapture*. It aims to address the issues of *indistinctive* and *subtle* movements in dynamic gesture recognition systems. Our architecture is an extension of the CNNLSTM [20] network and is evaluated in the domains of robot commands and co-speech gestures. This study is organized as follows: we present our literature review of some recent gesture recognition frameworks. Then, we describe the used datasets and our proposed method for analyzing gesture motion profiles. Next, we discuss the CNNLSTM architecture and our proposed hybrid gesture recognition framework called *Snapture*. Then, we compare the performance of both models. Finally, we conclude with a discussion and highlight some potential directions for future research.

2 Related Work

It is a common a step in vision-based hand gesture recognition systems to perform hand segmentation, i.e., extracting the moving hand from the background, before feeding the data into the learning model. In this context, Tsironi et al. [20] propose a pre-processing technique called the *differential image* algorithm in which grayscale frames are consecutively subtracted. The processed data passes through a *CNNLSTM* architecture, responsible for the implicit feature extractions and motion tracking of the hand across the time step sequences. The paper reports an accuracy of ∼0.92 of the approach over the *GRIT* dataset [19]. Despite having promising results, the study falls short in the correct classifications of gestures that share similar movement patterns. The authors report confusion between the "hello" and "no" classes, which both follow almost an exact motion path, i.e., *indistinctive* movements. Most subjects perform "hello" and "no" using an open palm and an extended index finger, respectively. Therefore, the distinction lies in the specifics of the handshape and finger arrangements, which is not considered by the author's proposed approach. However, this does not highly influence the overall model's performance since it is limited to a small subset of gesture classes. In addition to this observation, the dataset used for evaluation is small (a total of 543 sequences over ten classes). Thus, it may not give a clear picture of the actual capabilities and limitations of the architecture. On the other hand, it is beneficial to assess the robustness of this approach against another gesture domain in which the correlation in motion across gestures is more pronounced. In our study, we evaluate this approach in the context of co-speech gestures.

Moreover, it was reported in the literature that the performance of models could noticeably change based on the gesture vocabulary used for evaluation. Auge et al. [4] demonstrate that the accuracy of their Numenta's Hierarchical Temporal Memory (HTM) approach drops drastically from 0.95 to 0.7 when evaluated on a subset of gestures. This is done by trimming their dataset of radar data from ten to five classes. Considering this observation, the problem is simplified by ruling out gestures with *subtle* finger movements. This gives another clue on the non-triviality of distinguishing such particular hand and finger movements and their influence on the overall performance of models.

Similar to the CNNLSTM mentioned above, the loss of hand details is also reported in the work of Canuto et al. [17]. In their RGB-based approach, the authors propose a motion representation technique called *RGB star*. This method encodes color information, allowing compatibility with a broad spectrum of modern pre-trained CNN frameworks using transfer learning. In the presented approach, each gesture sequence is divided into three equal parts corresponding to the *pre-stroke*, *stroke*, and *post-stroke* gesture's stages as defined by Kendon [12]. The algorithm generates a motion representation for each said part, further merged using the frame's R, G, and B channels. The pre-processed data is fed into a feature extraction model using pre-trained ResNet50 and ResNet101 CNNs. The features are further weighted using a soft-attention mechanism, while a final classification is accomplished using a two-layered feed-forward network. Similar to our work, the approach is evaluated in both the robot command and co-speech gesture domains. An accuracy of ∼0.98 and ∼0.95 is reported by the paper on the GRIT [20] and Montalbano [7] datasets, respectively. Despite the astonishing performance, the loss of hand details leads to confusion between multiple Italian gestures: "noncenepiu", "ok", "freganiente", and "prendere". These movements share a similar path with the hand raised over the elbow. Additionally, some of them, e.g., "noncenepiu", are *subtle*. Furthermore, the architecture is overly complex, especially considering that the GRIT is a small-scale dataset, as discussed earlier. However, the results of this paper further confirm that the issues of *indistinctive* and *subtle* gestures are not trivial. The authors hypothesize that they can be addressed by integrating the hand information into the system. However, it remains an open question whether that would improve the performance.

The stated hypothesis is supported by the work of Wu et al. [23]. By fusing RGB and depth data alongside a skeleton modality, the authors demonstrate an increase in performance concerning gestures with similar motions. The authors propose an approach called Deep Dynamic Neural Network (DDNN), which consists of three neural networks, corresponding to the following modalities: RGB, depth, and skeleton. RGB and depth data is fed into a 3DCNN, while skeleton data passes through a Deep Belief Network (DBN). The learned features of these networks are fused and further fed into a feed-forward network that produces an emission probability at each time step. A Hidden Markov Model (HMM) is responsible for the temporal modeling gestures and pauses using a set of defined states. Each observation is classified by calculating the most probable path using the Viterbi algorithm. A score of 0.816 is reported on the Montalbano dataset [7] with the Jaccard index. The results hint at the importance of integrating RGB data when preserving the hand pose of gestures. The authors report less confusion regarding *implicit* Montalbano class movements when considering RGB and depth modalities. However, the approach does not consider Kendon's model [12] of co-speech gestures, which states that the gesture's hand pose is uncovered during the *stroke* phase. Hence, the hand pose is treated equally at all gesture phases. Consequently, some gestures are considered similar based on resemblance at the beginning or end of the motion, i.e., during the *rest phase*. Furthermore, their model is computationally intensive and requires long training times of five days. Thus, its robotic applications might be limited.

One approach that attempts to integrate static and dynamic recognition is the architecture of Mazhar et al. [13]. In their CNNLSTM-based framework called StaDNet, they propose an architecture consisting of two Inception V3 CNNs. Each CNN is responsible for the spatial extraction of the left and right hands. The authors claim that by cropping the CNN input to the hand and removing the background, the framework can learn *subtle* movements. The temporal learning is carried out using an LSTM network in which the features from multiple modalities are fused. Besides RGB, a 2D body skeleton model is extracted using OpenPose[1]. Additionally, the approach includes depth estimators extracted using a Kinect sensor. These estimators highlight the area of interest, i.e., the hand. This model requires two datasets for training: one static and one dynamic. The accuracy of 0.8675 and 0.989 is reported on the Chalearn 2016 and OpenSign datasets, respectively. However, the requirement of two independent sets of gesture vocabulary implies that the architecture learns them separately. Thus, the model can not seamlessly classify a gesture based on its dynamic and static characteristics in contrast to the authors' claim. Similarly, despite the model's ability to classify new samples using RGB data only, it requires multiple modalities for training. This contradicts the author's claim of a pure RGB-based framework.

3 Datasets and Motion Profiles

As mentioned in our literature review, the performance of gesture recognition frameworks might be influenced by the choice of the gesture domain. For example, co-speech gestures are more natural than robot commands. Therefore, they might incorporate higher chances of encountering *indistinctive* and *subtle* movements. Therefore, our choice of dataset spans across multiple gesture domains, i.e, the contexts of robot commands and co-speech gestures. In this section, we present the two datasets used to evaluate our framework. Due to the substantial difference between the domains of the datasets, we find it crucial to carry out a temporal analysis of the gesture sequences. Analysing gestures would provide insights into how the motion changes over their time span. Late in this section, we will describe our SSIM-based method for analyzing the motion profile of gestures. We present a motion analysis of several gestures from the said datasets and highlight some of the key distinctions. Based on that, we define two kind of gestures, *paused* and *repeating pattern* gestures, which we also define.

[1] `https://github.com/CMU-Perceptual-Computing-Lab/openpose`

3.1 The GRIT Robot Commands Dataset

In the context of robot commands, the "Gesture commands for Robot InTeraction" (GRIT)[2] [19] is one of the few publicly available dynamic gesture datasets. The corpus contains 543 *isolated gestures* distributed over nine gesture classes and recorded with the help of six participants. However, most movements are designed to have a unique path of motion. An exception is present in the case of classes "hello" and "no" which are *indistinctive movements*, as discussed previously. The recorded gestures vary in length, which is evident for gestures such as "circle" and "turn". Similarly, the subjects were not given any instructions on how to perform the gestures nor which hand to use. Consequently, it becomes more challenging to capture the hand pose of some gestures, especially with the camera's relatively low frame rate and resolution, as we will discuss later. The dataset was collected under lab-controlled settings with a plain white background. All subjects have similar lighting conditions with no noise in the surrounding.

3.2 The Montalbano V1 Co-Speech Dataset

The Montalbano dataset is a publicly available corpus in the domain of co-speech gestures. This dataset was collected as part of the *ChaLearn*[3] *Looking at People challenge* [7]. It contains around 14 000 Italian gestures spreading over 20 gesture classes. We use this dataset to evaluate our approach in the context of gestures "in the wild". Although each recorded video contains one subject only, the data is collected in various day-to-day human environments, such as offices and lecture halls, with different noisy backgrounds. The Montalbano dataset tackles the task of multimodal gesture spotting. Therefore, it includes multiple sensory data, e.g., depth, user index, skeleton, and RGB images. However, we only use RGB data due to the advantages of reproducibility and portability that vision-based approaches provide. Since the gestures were recorded continuously with little to no pause between them, we convert them into *isolated gestures* by identifying the start and end of each movement sequence. The variation in length is also pronounced in this dataset, which is partly due to the high number of subjects ($\sim$50). We make the annotations created for isolating the gestures and source code of the experiments presented in this work publicly available[4].

3.3 Motion Profile Analysis

Our approach for tackling the issues concerning *indistinctive* and *subtle* movements relies on fusing the hand motion and pose. Therefore, we refer to our

[2] https://www.inf.uni-hamburg.de/en/inst/ab/wtm/research/corpora.html

[3] http://chalearnlap.cvc.uab.es/

[4] https://github.com/hassanali-90/snapture/

architecture (described in the next section) as *hybrid*. Motion features can be extracted by exploiting the temporal information across the consecutive frames. The hand pose is most interesting at the *stroke* phase of co-speech gestures, as described by Kendon [12]. However, the relationship between frames and *stroke* in the targeted datasets is not clear. Therefore, we analyze hand gesture sequences in the studied datasets to uncover the characteristics of their dynamics in terms of motion and pause. Due to the lack of approaches for gesture analysis, we utilize the structural similarity index measure (SSIM) [22] as a metric for the similarity between consecutive frames. We carry out the calculation as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}\,, \tag{1}$$

$$C_1 = (K_1 L)^2\,, \tag{2}$$

$$C_2 = (K_2 L)^2\,, \tag{3}$$

where x and y are spatially local windows of the input frames. μ_x and μ_y represent the mean intensity of x and y, respectively. Similarly, σ_x and σ_y denote the standard deviation. Stability constants C_1 and C_2 help avoid a division by zero and are calculated using the pixel range L (255 for 8-bit grayscale frames) and positive values much smaller than 1, i.e., K_1 and K_2.

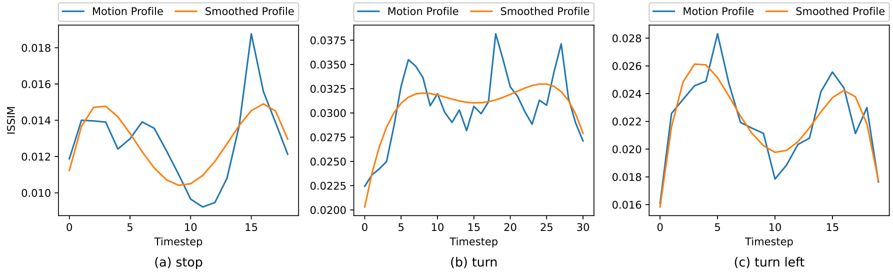
Using the first frame as a reference, we can quantify the amount of motion and pause across the gesture time span. By inverting the equation, we can express change across frames. We refer to that as the *Inverted SSIM (ISSIM)*:

$$ISSIM = 1 - SSIM(I_i, I_0)\,, \tag{4}$$

where I_i and I_0 denote the grayscale frames at time steps i and 0, respectively.

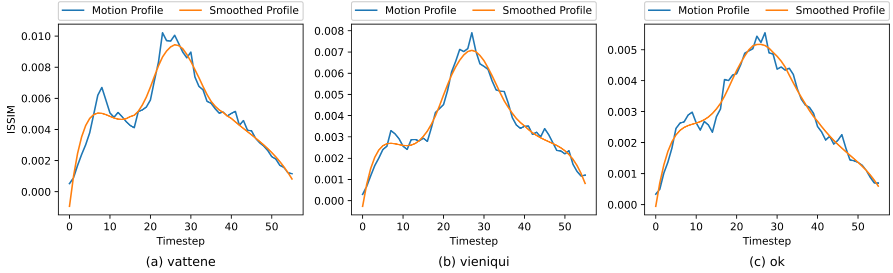
Due to their design as robot commands, we observe two variations of movements in the GRIT dataset based on the analyzed motion profile. *Paused* gestures include a pronounced period of pause around the gesture peak. For example, “stop” (cf. Fig. 1 (a)) and “turn left” (cf. Fig. 1 (c)) lack motion around the peak since participants hold their hand briefly in a fixed position. In contrast, in gestures, such as “turn” (cf. Fig. 1 (b)), subjects continuously repeat a circular pattern across the gesture’s time span. We refer to these movements as *repeating pattern* gestures. As we will see later, these unique characteristics of motion and pause of each gesture influence the design of our approach.

In contrast to the GRIT dataset, the Montalbano gestures are co-speech. Thus, they follow Kendon’s [12] relational model of gesticulation and concurrent speech. The intensity of the movement starts and ends gradually with a clear peak in between. In Fig. 2, we see examples of the motion profile of Montalbano gestures.



(a) stop (b) turn (c) turn left

Fig. 1: The motion profile of GRIT gestures “stop”, “turn left” and “turn”. “stop”, “turn left” are *paused* at their peak, while “turn” is with a *repeating pattern* due to the continuous intensity across its time span.



(a) vattene (b) vieniqui (c) ok

Fig. 2: The motion profile of Montalbano gestures “vattene”, “vieniqui” and “ok”. These co-speech gestures follow Kendon’s model [12], hence, they start and end with low intensity and have a clear peak around the midpoint of the timeline.

4 Snapture - Hybrid Gesture Recognition

One core concept of our hybrid recognition system is to combine the dynamic and static (movement and hand pose) aspect of gestures. The analysis of the GRIT and Montalbano gestures provides insights into their motion profiles. This information will be utilized when extracting the hand pose at the peak, as we will see later. In this section, we will describe our proposed approach called *SNAPshot capTURE* (*Snapture*). Our approach is an extension of the *CNNLSTM* [20] architecture, and aims to find a solution to the problems of *indistinctive* and *subtle gestures*. This is done by integrating the hand details in a dynamic gesture recognition framework, thus performing a hybrid gesture recognition task. A simplified overview can be seen in Fig. 3.

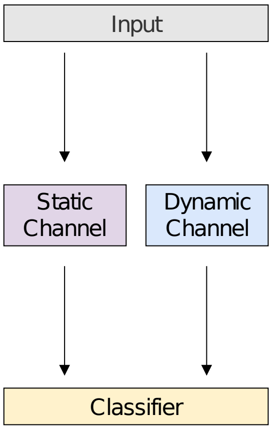


Fig. 3: An overview of the *Snapture* framework. The architecture consists of a dynamic and static channels, fused into a final classifier. Thus, it performs a hybrid hand gesture recognition task.

4.1 Dynamic Channel

The *CNNLSTM* [20] architecture is an RGB approach, which has proven to work quite well classifying various motion patterns of robot commands. The CNNLSTM network consists of a 2-layer stacked convolutional neural network (CNN) followed by a long short-term memory (LSTM) network (cf. Fig. 4). The input frames represent segmented gestures, which means the moving hand is detected and extracted in the input by subtracting subsequent frames. This is calculated using the *differential image* algorithm as described in [20]:

$$\Delta_i = (I_i - I_{i-1}) \wedge (I_{i+1} - I_i)\,, \tag{5}$$

where Δ_i and Δ_{i-1} are the segmented gesture input frames at the current and previous time steps, respectively. I_{i-1}, I_i and I_{i+1} denote the grayscale frames at time steps $i-1$, i and $i+1$, respectively. $\wedge$ is the bitwise AND operator. Additionally, each input sequence represents an *isolated gesture*, i.e., the sequence's start and end is known.

As motivated earlier, we are interested in evaluating our *Snapture* approach across multiple gesture domains, i.e., robot commands and co-speech gestures. The *Snapture* architecture is an extension of the CNNLSTM model. We will use the CNNLSTM method as described in [20] as baseline for comparison. On the other hand, little is known so far about the performance of CNNLSTM in the context of co-speech gestures. Therefore, our evaluation provides some further insights into the performance of CNNLSTM using the Montalbano dataset. Our PyTorch[5] implementation of the CNNLSTM uses the same kernel size and number of filters of the CNN as the original proposal [20]. However,

[5] `https://pytorch.org/`

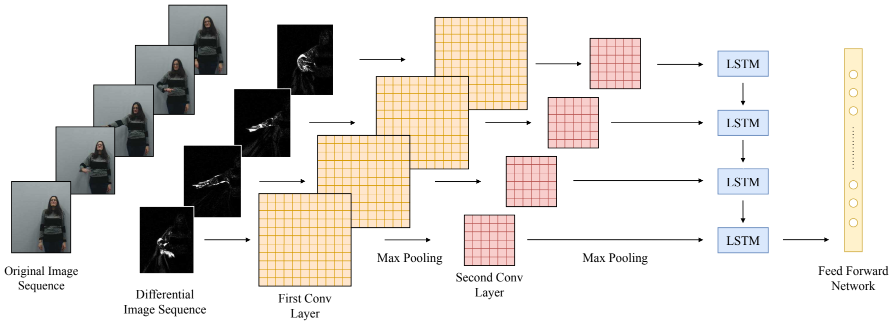
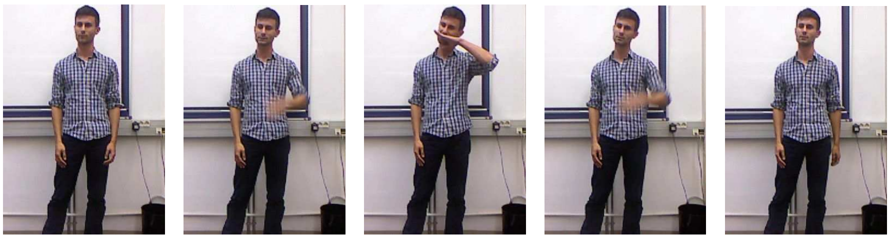


Fig. 4: The dynamic channel of *Snapture* is a CNNLSTM network, consisting of two layers of CNN followed by a LSTM and feed forward network. The *isolated gesture* input is pre-segmented using the *differential image* algorithm. For clarity, we show only five frames and increase the contrast of the differential images.

we tune the rest of the experimental settings. The stacked convolution layers have five and ten kernels of size 11x11 and 6x6, respectively. Each layer has a 1x1 stride, zero-padded input, and a hyperbolic tangent (Tanh) activation function. We chose the non-linearity using grid-search with the rectified linear unit (ReLU) as an additional candidate. A max-pooling layer of size 2x2 follows each convolution layer. Additionally, batch normalization is used after each convolution to reduce internal covariate shift [11] and speed up the training. We initialize the CNN's weights with values from a uniform distribution [9]. The output of the last convolution layer is flattened and propagated through a feed-forward layer.

Due to the input of *isolated gestures*, each mini-batch has all the information needed for the network to produce a classification. Therefore, we opt to use a stateless LSTM. The LSTM's number of layers and neurons are selected using grid search. The optimal number of layers is 2 out of 1, 2, 4, and 8. The optimal number of neurons has resulted differently for the GRIT and Montalbano datasets. We chose 64 and 512 neurons for the GRIT and Montalbano datasets, respectively. We initialize the LSTM with weights from a uniform distribution with zero bias. After passing through dropout [14], the output of the LSTM is further propagated into feed-forward and softmax layers, producing a probability distribution over the gesture classes. The rest of the hyperparameters will be presented in a later section.

Due to the cell state of the LSTM, the CNNLSTM architecture's output can be configured in various ways. In the original CNNLSTM proposal, the authors have opted for a *frame-level* configuration. Therefore, their model predicts a label for each time step. In contrast, we tune our model to produce a *sequence-level* classification for each gesture sequence in its entirety. Therefore, our model requires no additional post-processing steps and fits the concept of capturing a *snapshot* more intuitively.



(a) Rest position (b) Pre-stroke (c) Stroke (d) Post-stroke (e) Rest position

Fig. 5: The five gesture phases of Kendon [12]. Each gesture starts with a *rest phase*. In *pre-stroke*, the limb moves from the rest position into the *stroke* phase. The *stroke* phase contains the most expressive information. In post-stroke, the limb moves away from *stroke* back into *rest phase*.

4.2 Static Channel

This channel is responsible for capturing the specific handshape and finger arrangements through a so-called *snapshot* at the gesture's peak. We detect and extract the gesture at the peak corresponding to the *stroke* phase. This provides the hand pose information, which can be fused alongside CNNLSTM's motion learning outcome. As a result, our method integrates the characteristics of static and dynamic recognition systems.

4.2.1 Gesture Peak Detection

According to Kendon's [12] model of the relationship between gestures and concurrent speech, human gestures are described by five phases (cf. Fig 5). Gestures start with a *rest phase*, which represents a non-movement state of the arms. In the *pre-stroke* or *preparation phase*, a gradual intensity in motion of one or both arms starts to unveil. Next is the *stroke* phase in which the gesture static characteristics, i.e., hand shape and finger configurations, completely unfold. These characteristics start to fade away in the *post-stroke*, or *retraction* phase as the intensity of motion gradually decreases. The gesture ends again with a rest phase. Montalbano gestures have a clear peak through the frames around the midpoint of the gesture sequence (cf. section 3.2). Similar time steps are occupied by a pronounced pause in *paused* gestures (cf. section 3.1). Since the data consists of *isolated gestures*, we identify the peak as the frame in the middle of the gesture sequence.

4.2.2 Gesture Peak Extraction

We follow a skin detection technique to extract the hand from the rest of the frame. Our implementation uses Python and OpenCV[6]. First, the face

[6] `https://opencv.org/opencv-4-5-3/`

of the subject is detected and removed from the frame since skin detection techniques treat all visible skin of body parts in the frame equally. Next, the hand is segmented by converting into the orthogonal color space *YCbCr* [10]: *Y* representing the luminance, while *Cb* and *Cr* indicate the chromaticity. This is done to avoid the high correlation between luminance, hue, and saturation in RGB [15]. Since various lighting conditions highly influence skin tones, we apply the threshold on chrominance only. We use the thresholds *Cb*=[80, 120] and *Cr*=[133, 173] proposed by Basilio et al. [5]. According to the authors, these threshold values are independent of skin tone. However, the datasets used in our study are limited in the diversity of skin tones. An additional step of background removal is applied to the Montalbano data using simple background subtraction. This is due to the complexity of recording scenes, unlike GRIT. Furthermore, we apply the connected component analysis, which describes the *YCbCr* mask in terms of BLOBs. These objects are then sorted by size and position. Due to the noisy background in the Montalbano dataset, we filter out objects that do not belong to the foreground, calculated in the step of background removal.

To avoid any subject's preferred hand assumption, we pick the higher object in the frame. As we observe in the data, all subjects have the hand performing the gesture in an upper position, while the other is usually in rest or slightly raised. For gestures requiring two hands, both always contain the same hand pose. Therefore, our algorithm has the flexibility of picking up either hand in this case. A step of hand smoothing is applied using erosion and dilation morphological transformations. However, we find that omitting this step does not influence the algorithm's output. Finally, an area around the detected hand is extracted from the original frame and further resized to 64x48 pixels matching the CNN input. The gesture peak extraction module is depicted in Fig. 6.

4.2.3 Static Channel Control

Despite the convenience of vision-acquired data, modest RGB cameras tend to have certain limitations. For example, they fail to capture the hand details when a rapid movement is present. This is caused by factors such as camera resolution and exposure time. Consequently, it leads to a blurry hand in the frame (cf. Fig. 7). This is pronounced for *repeating pattern* robot commands due to the high dynamics of movement. This phenomenon raises a challenge to any vision-based approach due to the missing information and limited data source quality. However, we aim to address it by regulating the static channel based on the amount of motion contained in a gesture. More precisely, we integrate the extracted static information, i.e., the hand shape at the peak, only if the amount of motion lies below a threshold., i.e., the *stroke* phase contains a pause sufficient for the *snapshot* extraction.

We use the SSIM based approach presented in section 3.3 as quantitative metrics for the amount of motion and pause. We split each *isolated gesture* into three parts with equal number of frames: 1) the first part represents all

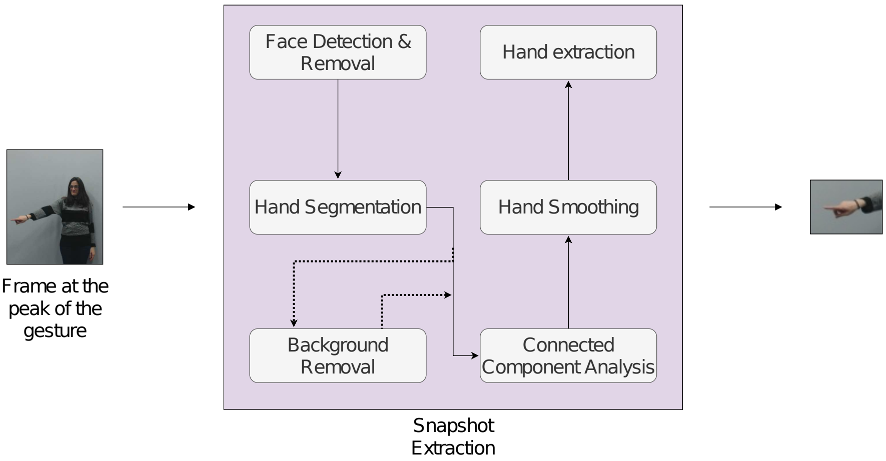
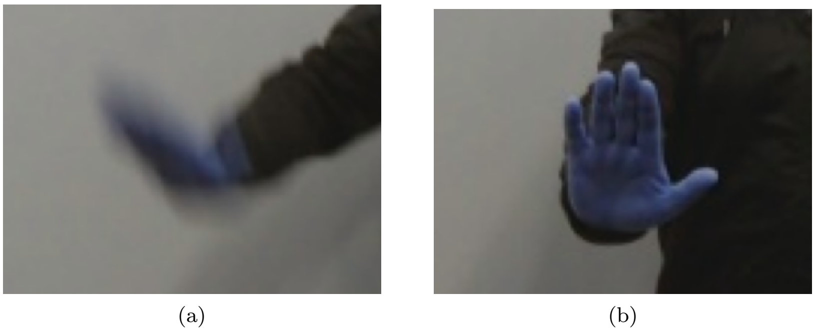


Fig. 6: The gesture peak extraction module of the *Snapture* approach. Using a skin detection technique, the hand shape and finger configurations are extracted from a target frame at the gesture's peak. Background removal is only applied to the Montalbano gestures (dotted line).


(a)

(b)

Fig. 7: *Repeating pattern* gestures, e.g., "circle" (a), contain a *blur* at the peak compared to *paused* movements, e.g., "stop" (b). The blurry hand at the gesture's peak for highly dynamic movements is challenging for RGB-based approaches. We bypass this issue by regulating the static channel of our approach.

the frames in the *rest* and *pre-stroke* phases, 2) the second part contains the frames in *pre-stroke* and *post-stroke* phases, 3) and the third part consists of all the frames consecutively from *post-stroke* to *rest* phases. We assume the three parts to be of equal length for simplicity. The three parts and our defined threshold are visualized for the GRIT (c.f. Fig. 8) and Montalbano (c.f. Fig. 9) datasets. The average amount of motion in part 2 is less than part 1 and part 3, which supports our choice of Kendon's [12] *stroke* phase as the peak of the gesture. Furthermore, most samples of *paused* gestures,

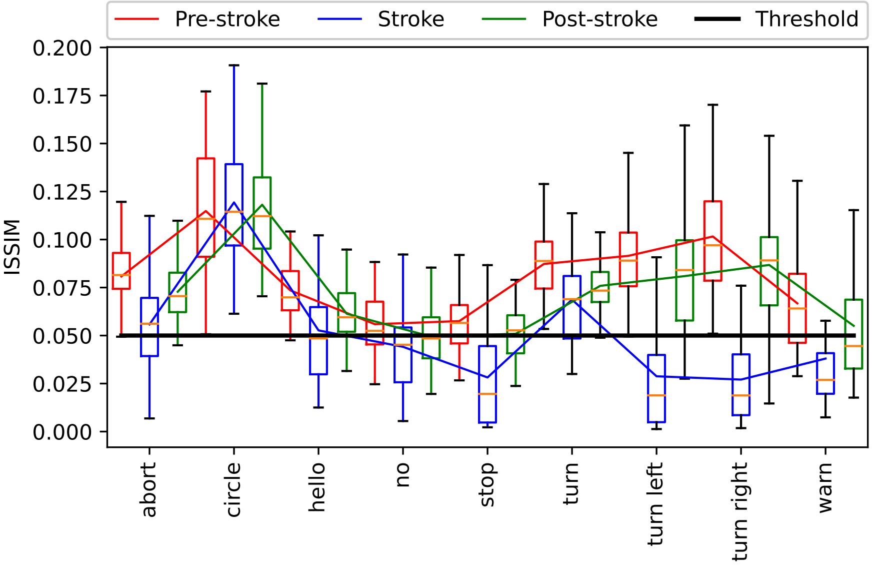


Fig. 8: Our motion analysis of the GRIT dataset after splitting into three parts: *rest* to *pre-stroke* phases, *pre-stroke* to *post-stroke* phases and *post-stroke* to *rest* phases. The second part contains more pause and facilitates capturing a *snapshot*. The black line denotes our defined threshold for regulating the static channel.

such as: “stop”, “turn left” and “turn right”, lie well below the threshold due to their pronounced period of pause. In contrast, the intensity of motion is evidently high for *repeating pattern* gestures, e.g., “circle” (cf. Fig. 8). This goes well with our definition of *repeating pattern* gestures (cf. section 3.1) since the *stroke* phase of these commands does not contain any distinct hand shape information. Thus, the shape of the hand plays a minimal role in the recognition, and by disabling the static channel for “circle” samples, we can reduce the influence of the drastic loss of sharpness in the input. On the other hand, the majority of Montalbano gesture classes contain pause facilitating the capturing of a *snapshot*. The cut-off value lies around the median of each gesture class except for “basta” and “cheduepalle”, which have an explicit arm movement to the side of the body, as we will discuss later.

5 Experiments and Results

In this section, we present the experiments carried out in this study. In each experiment, we evaluate and compare the following: 1) a *sequence-based* CNNLSTM model, which classifies gestures based on motion only and acts as a baseline for comparison, 2) our *Snapture* architecture without the threshold

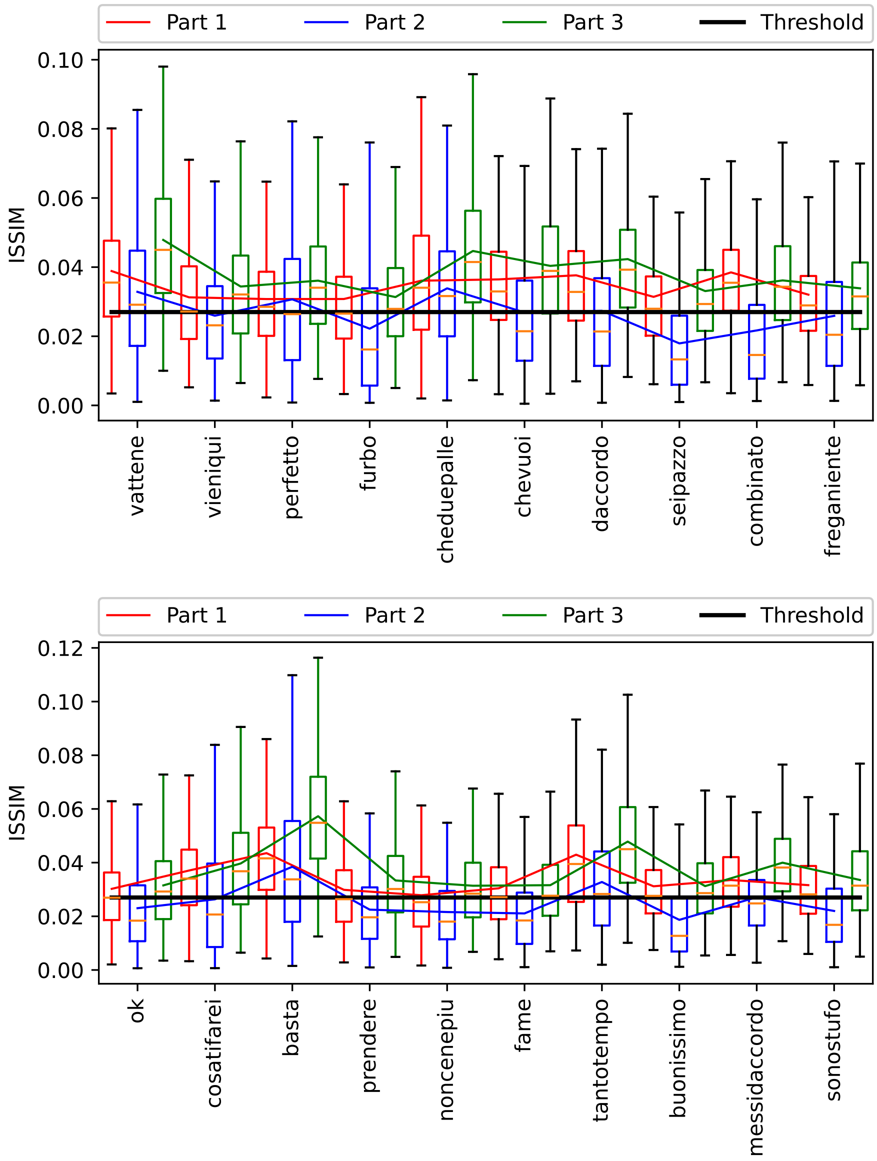


Fig. 9: Our motion analysis of the Montalbano dataset after splitting into three parts: *rest* to *pre-stroke* phases, *pre-stroke* to *post-stroke* phases and *post-stroke* to *rest* phases. Similar to GRIT, the second part contains more pause that facilitates capturing a *snapshot*. Our defined threshold for regulating the static channel is denoted by the black line.

mechanism, which predicts a class by integrating the handshape and motion, and 3) *Snapture* with the threshold-controlled mechanism for regulating the static channel based on the sufficiency of pause to capture a *snapshot*. We will refer to this model as $Snapture_{thold}$. We repeat the experiment in the contexts of robot commands and co-speech gestures using the GRIT and Montalbano datasets, respectively. The purpose is to evaluate the influence of *subtle* and *indistinctive* gestures on the performance of each of the models. These types of movements are more pronounced in co-speech gestures, as described earlier. Additionally, this gives more insights into the performance of each model across different gesture domains.

5.1 Experimental Settings

The training parameters of each experiment were selected using grid search and are listed in the next sections. We run each of the models under similar conditions. The hardware specifications used for training and testing the models are as follows: 1) Ubuntu 18.04.5 LTS operating system; 2) Intel Core i7-4930K 3.40 GHz with six cores; 3) 8 GB of RAM; 4) NVIDIA GeForce GTX 1080 graphics card with 8 GB of memory. The performance of each model is evaluated using accuracy, F1-score, and training time metrics. We report the average performance of each model over five trials. In each trial, we repeat the steps of training and testing. Further classification behavior analysis is done by visualizing and discussing the confusion matrices.

5.2 GRIT Experiment

The search space and optimal hyperparameters of the GRIT experiment are listed in Table 1. In this experiment, the resulting optimal values are identical, which we explain by the similarity in architecture and training procedure across the models. We use the same data split ratio for each trained model to conduct a fair comparison. To avoid data imbalances, we use stratified sampling in terms of class labels. Similar to the original CNNLSTM proposal [20], we use cross-validation. However, we opt to use a 3-fold split, meaning that approximately 33% of the data is held out for testing. The results of the experiment are summarized in Table 2.

Our *Snapture* approach achieves slightly superior results compared to the CNNLSTM in terms of accuracy and F1-score. The scores across the *Snapture* and $Snapture_{thold}$ variations are similar. The three models have a slight deviation across the five trials. We explain the marginal accuracy boost by three factors. First, GRIT robot commands are designed to have a unique motion path, as motivated earlier. Therefore, the CNNLSTM model is sufficient to provide good performance due to its powerful movement learning capabilities. Second, due to the *repeating pattern* gestures, the majority of the GRIT movements do not have sufficient pause for capturing a *snapshot* (approximately 44%) according to our threshold definition. Combined with the small dataset size, our model may not have seen enough training data to learn the unique characteristics of handshapes. Third, since only approximately 44% of GRIT samples include a motion at the peak beneath the defined threshold, the *Snapture*$_{thold}$ acts similar to a CNNLSTM model in 56% of the cases. Therefore, it is not able to contribute to a noticeable accuracy increase.

However, we analyze the results further through the confusion matrix of the average case, i.e., the mean results of five trials (cf. Fig.10). The most confusion in the CNNLSTM model occurs between the classes “hello” and “no”, “hello” and “stop”, “no” and “stop”, and “stop” and “abort”. All of these movements have a similar motion profile but differ in hand shape. Thus, this supports that the *indistinctive* movements negatively influence the performance of CNNLSTM. We explain that by the CNNLSTM’s lack of considering the hand details. Similar findings were reported in the work of Tsironi et al. [19]. On the other hand, the confusion between these classes is less pronounced in *Snapture* (cf. Fig.11) due to the static channel, which provides the hand pose information. However, the misclassification of “hello” samples as “no” still negatively impacts the performance of *Snapture*. We observe that some participants perform “hello” and “no” in a rapid fashion resulting in a *blur* effect and noisy input to the network. Therefore, the *Snapture*$_{thold}$ improves the situation (cf. Fig.12) by excluding the *snapshot* in case the input is not sufficient for interpreting the hand details. However, dealing with the low resolution of the dataset remains a challenge to any approach while extracting meaningful hand shape and finger arrangement. On the other hand, *repeating pattern* movements, e.g., “circle” yields comparable F1-score values across the three architectures (cf. Fig.13) due to the distinctive hand movement. However, the number of false positives and true negatives associated with “circle” drops noticeably in *Snapture*$_{thold}$, further emphasizing that the static channel is indeed counterproductive for such movements.

On a different note, the confusion between the classes “no” and “stop” is less pronounced in *Snapture* and *Snapture*$_{thold}$ compared to CNNLSTM. Despite the dissimilarity between the two classes, some subjects tend to perform “no” with a slight left and right hand movement around the wrist, making it very similar to “stop” in terms of arm movement (raised and direct towards the camera). As motivated earlier, due to the loss in hand details, the CNNLSTM struggles with this sort of *implicit* hand movements. Therefore, our approach improves the performance by integrating the hand shape and finger arrangements.

5.3 Montalbano Experiment

Similar to the previous experiment, we report the hyperparameters in Table 3. Due to the more considerable number of class labels, the search space is extended compared to the GRIT experiment. Since the dataset is part of the *ChaLearn Looking at People* challenge, it is already split into training and test datasets, with each set containing unique subjects. To avoid any influence

Table 1: The search space and optimal hyperparameter values (in bold) of each model in the *GRIT* experiment.

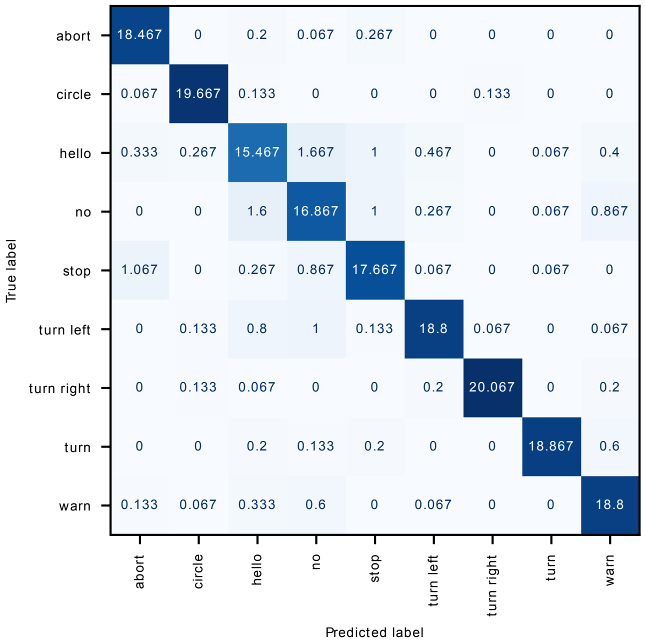
Hyperparameter	CNNLSTM	Snapture*
Learning rate	[0.01, **0.001**, 0.0001]	[0.01, **0.001**, 0.0001]
Number of epochs	[10, 20, **40**]	[10, 20, **40**]
Mini-batch size	[16, 32, **64**, 128]	[16, 32, **64**, 128]
Optimizer	[**Adam**, SGD]	[**Adam**, SGD]

***Similar for** $Snapture_{thold}$**.**

Table 2: The results of the *GRIT* experiment under the described settings. The reported metrics represent the mean of five trials, while the values in parentheses correspond to the standard deviation. The superior accuracy and F1-score values are in bold.

Model	CNNLSTM	Snapture	$\mathbf{Snapture_{thold}}$
Accuracy	0.91 (0.012)	0.924 (0.006)	**0.926** (0.008)
F1-score	0.913 (0.012)	**0.927** (0.005)	0.913 (0.012)
Time*	140.612 (0.255)	170.012 (1.027)	125.156 (1.117)

*In seconds.



True label \ Predicted label	abort	circle	hello	no	stop	turn left	turn right	turn	warn
abort	18.467	0	0.2	0.067	0.267	0	0	0	0
circle	0.067	19.667	0.133	0	0	0	0.133	0	0
hello	0.333	0.267	15.467	1.667	1	0.467	0	0.067	0.4
no	0	0	1.6	16.867	1	0.267	0	0.067	0.867
stop	1.067	0	0.267	0.867	17.667	0.067	0	0.067	0
turn left	0	0.133	0.8	1	0.133	18.8	0.067	0	0.067
turn right	0	0.133	0.067	0	0	0.2	20.067	0	0.2
turn	0	0	0.2	0.133	0.2	0	0	18.867	0.6
warn	0.133	0.067	0.333	0.6	0	0.067	0	0	18.8

Fig. 10: The confusion matrix of the average case for the CNNLSTM on the GRIT dataset. The confusion is pronounced between the classes “hello” and “no”, “hello” and “stop”, “no” and “stop”.

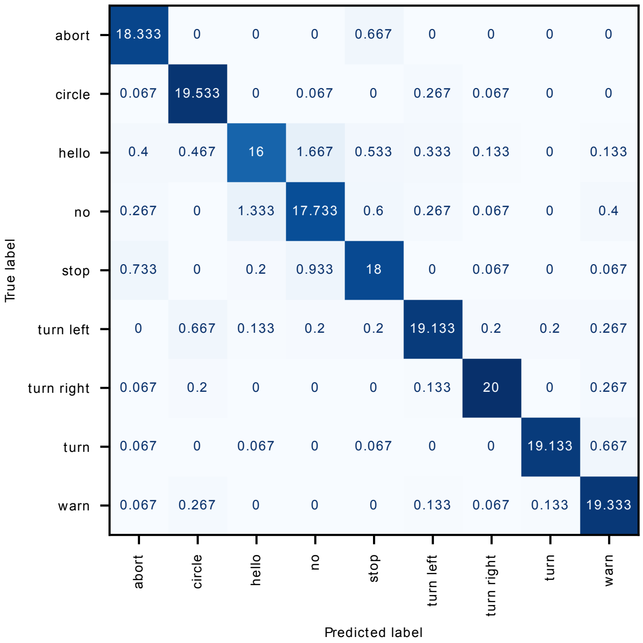
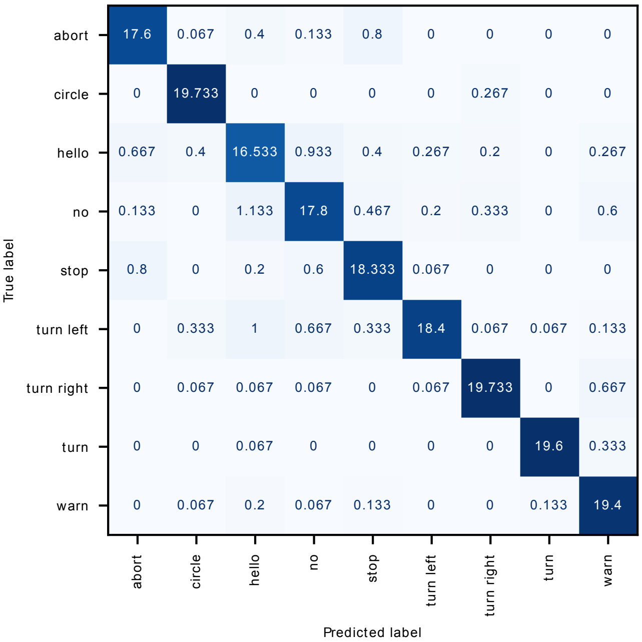


Fig. 11: The confusion matrix of the average case for *Snapture* on the GRIT dataset. The confusion is less pronounced between the classes “hello” and “no”, “hello” and “stop”, “no” and “stop”. However, the performance is still negatively influenced by the false classification of some “hello” samples as “no”.

triggered by subject variability, we implement our own split with data from all participants. We follow this approach since we focus on comparing the classification behavior of the different models rather than comparing them to the benchmark. Furthermore, we increase the size of the test set. Our split consists of 70% and 30% of randomly selected data for training and testing, respectively. Stratified sampling is utilized for an approximately uniform distribution of class labels across the sets.

Our *Snapture* approach scores superior accuracy and F1-score compared to CNNLSTM. Also, the $Snapture_{thold}$ improves the results even further (cf. Table 4). However, we observe a noticeable time increase in $Snapture_{thold}$. We explain that the additional check for each sample to identify where it lies in comparison to the defined threshold. Approximately, 70% of the Montalbano data contain a sufficient pause for a *snapshot*. Thus, it gives more insights into the performance of the $Snapture_{thold}$ approach. By observing per-class performance, *Snapture* achieves superior per-class F1-scores compared to the CNNLSTM with the exception of “basta” (both models achieve an identical



	abort	circle	hello	no	stop	turn left	turn right	turn	warn
abort	17.6	0.067	0.4	0.133	0.8	0	0	0	0
circle	0	19.733	0	0	0	0	0.267	0	0
hello	0.667	0.4	16.533	0.933	0.4	0.267	0.2	0	0.267
no	0.133	0	1.133	17.8	0.467	0.2	0.333	0	0.6
stop	0.8	0	0.2	0.6	18.333	0.067	0	0	0
turn left	0	0.333	1	0.667	0.333	18.4	0.067	0.067	0.133
turn right	0	0.067	0.067	0.067	0	0.067	19.733	0	0.667
turn	0	0	0.067	0	0	0	0	19.6	0.333
warn	0	0.067	0.2	0.067	0.133	0	0	0.133	19.4

Fig. 12: The confusion matrix of the average case for $\mathit{Snapture}_{thold}$ on the GRIT dataset. Less confusion can be observed concerning class "circle", which confirms that the static channel should be disabled for such *repeating pattern* movements.

score). Furthermore, we report a boost in F1-score on all classes with the $\mathit{Snapture}_{thold}$.

In contrast to robot commands, most co-speech gestures in Montalbano have a similar path of motion. Generally, we observe two main categories of movements in the Montalbano dataset: single-handed and two-handed. In single hand movements, the arm is raised above the head level. Most of these gestures have a noticeably similar motion with a distinctive hand and finger arrangement at the peak. Additionally, some single-hand gestures include a delicate hand movement at the peak. Second, two hand movements require synchronization between the two arms.

Indistinctive Movements: In CNNLSTM, multiple observations of classes "vattene" are miscalssified as "vieniqui", "perfetto" or "tantotempo" (cf. Fig. 14). We explain that by the similarity in hand motion. Compared to the CNNLSTM, an addition of ∼19 and ∼32 samples on average are correctly classified by the *Snapture* and $\mathit{Snapture}_{thold}$, respectively (cf. Fig. 15 and Fig. 16). Consequently, we observe F1-score improvements in the respective classes (cf. Fig. 17). Furthermore, the CNNLSTM achieves poor F1-score values (below

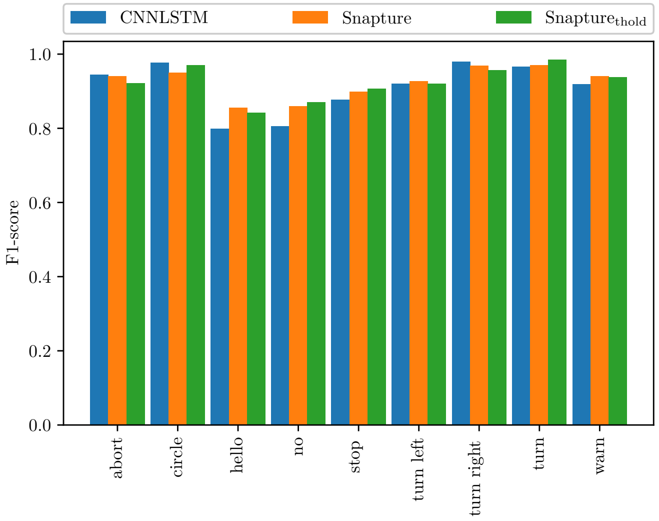
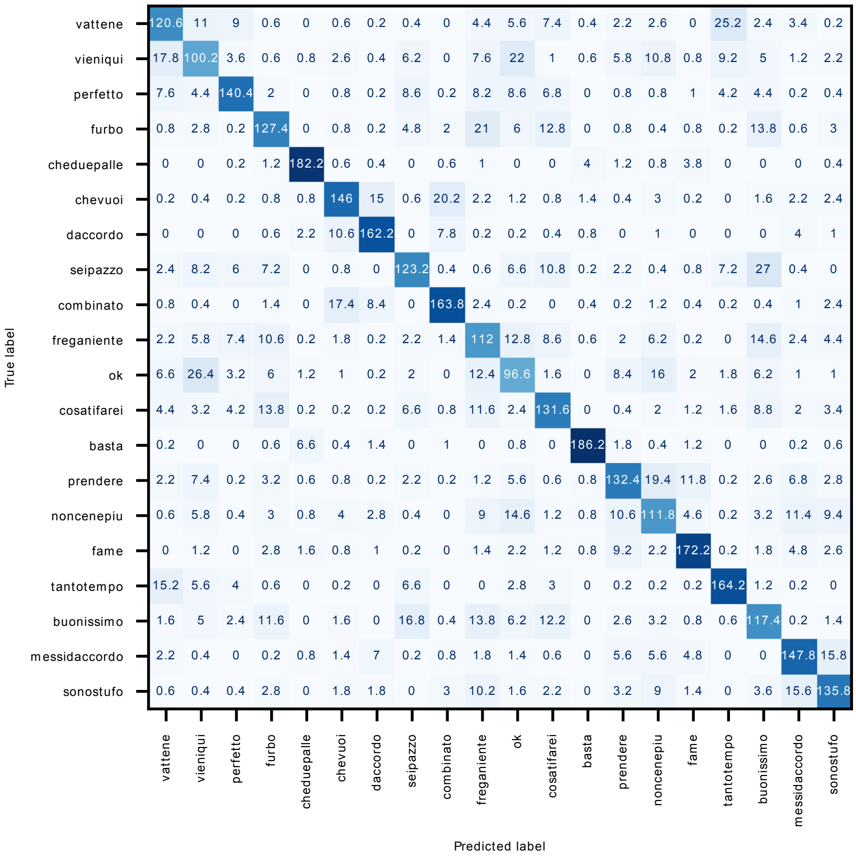


Fig. 13: A comparison of per-class F1-score values between the different approaches on the GRIT dataset. *Snapture* increases the score on classes “hello” and “no”, while the performance across the remaining classes is comparable.

0.6) on classes “vieniqui” and “freganiente”, “ok”, “noncenepiu” and “buonissimo”. Most of the confusion of class “ok” is tied to false positives/negatives with one of the said classes. This can be explained by the similarity in their motion. However, the total number of misclassified “ok” samples drops in *Snapture* and $\textit{Snapture}_{thold}$ by approximately 30. Therefore, we observe an increase in the F1-score. Additionally, *Snapture* and $\textit{Snapture}_{thold}$ enhance the F1-score of class “seipazzo”. Approximately, an additional average of 23 and 25 samples are correctly classified due to less confusion with “buonissimo”.

On a different note, classes that share both the motion and handshape are challenging for our approach. For example, classes “vattene”, “vieniqui” and “tantotempo” use a similar open palm handshape at the peak (cf. Fig. 18). Therefore, the confusion between such these classes is still noticeable in *Snapture* and $\textit{Snapture}_{thold}$ despite the handshape information.

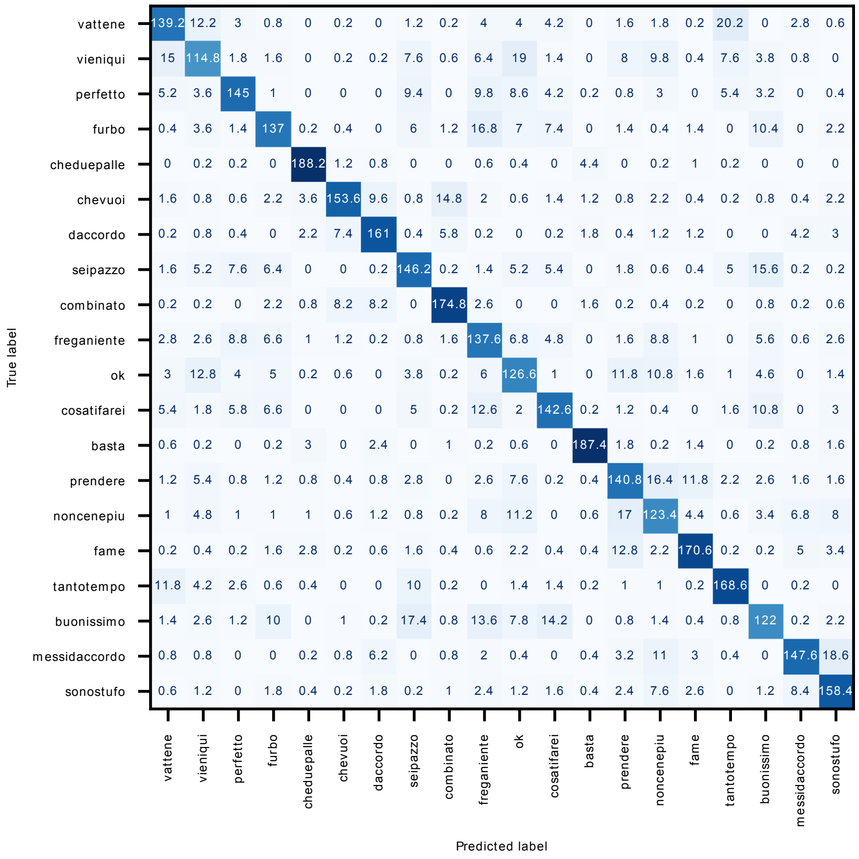
Implicit Movements: Besides the motion similarity, some single-hand gestures include a delicate hand movement at the peak. For example, “sonostufo” includes a subtle movement of the hand against the chest. Similarly, “noncenepiu” and “buonissimo” include a rotational motion of the extended index and thumb fingers around the wrist. Due to the pre-processing, i.e., the *differential images* algorithm, these implicit hand details and movements are lost. Consequently, they are not picked up by the CNNLSTM due to the lack of information at the input. However, the confusion related to these classes is noticeably less in *Snapture* and $\textit{Snapture}_{thold}$ (cf. Fig. 15 and Fig. 16). On the other hand,



	vattene	vieniqui	perfetto	furbo	cheduepalle	chevuoi	daccordo	seipazzo	combinato	freganiente	ok	cosatifarei	basta	prendere	noncenepiu	fame	tantotempo	buonissimo	messidaccordo	sonostufo
vattene	120.6	11	9	0.6	0	0.6	0.2	0.4	0	4.4	5.6	7.4	0.4	2.2	2.6	0	25.2	2.4	3.4	0.2
vieniqui	17.8	100.2	3.6	0.6	0.8	2.6	0.4	6.2	0	7.6	22	1	0.6	5.8	10.8	0.8	9.2	5	1.2	2.2
perfetto	7.6	4.4	140.4	2	0	0.8	0.2	8.6	0.2	8.2	8.6	6.8	0	0.8	0.8	1	4.2	4.4	0.2	0.4
furbo	0.8	2.8	0.2	127.4	0	0.8	0.2	4.8	2	21	6	12.8	0	0.8	0.4	0.8	0.2	13.8	0.6	3
cheduepalle	0	0	0.2	1.2	182.2	0.6	0.4	0	0.6	1	0	0	4	1.2	0.8	3.8	0	0	0	0.4
chevuoi	0.2	0.4	0.2	0.8	0.8	146	15	0.6	20.2	2.2	1.2	0.8	1.4	0.4	3	0.2	0	1.6	2.2	2.4
daccordo	0	0	0	0.6	2.2	10.6	162.2	0	7.8	0.2	0.2	0.4	0.8	0	1	0	0	0	4	1
seipazzo	2.4	8.2	6	7.2	0	0.8	0	123.2	0.4	0.6	6.6	10.8	0.2	2.2	0.4	0.8	7.2	27	0.4	0
combinato	0.8	0.4	0	1.4	0	17.4	8.4	0	163.8	2.4	0.2	0	0.4	0.2	1.2	0.4	0.2	0.4	1	2.4
freganiente	2.2	5.8	7.4	10.6	0.2	1.8	0.2	2.2	1.4	112	12.8	8.6	0.6	2	6.2	0.2	0	14.6	2.4	4.4
ok	6.6	26.4	3.2	6	1.2	1	0.2	2	0	12.4	96.6	1.6	0	8.4	16	2	1.8	6.2	1	1
cosatifarei	4.4	3.2	4.2	13.8	0.2	0.2	0.2	6.6	0.8	11.6	2.4	131.6	0	0.4	2	1.2	1.6	8.8	2	3.4
basta	0.2	0	0	0.6	6.6	0.4	1.4	0	1	0	0.8	0	186.2	1.8	0.4	1.2	0	0	0.2	0.6
prendere	2.2	7.4	0.2	3.2	0.6	0.8	0.2	2.2	0.2	1.2	5.6	0.6	0.8	132.4	19.4	11.8	0.2	2.6	6.8	2.8
noncenepiu	0.6	5.8	0.4	3	0.8	4	2.8	0.4	0	9	14.6	1.2	0.8	10.6	111.8	4.6	0.2	3.2	11.4	9.4
fame	0	1.2	0	2.8	1.6	0.8	1	0.2	0	1.4	2.2	1.2	0.8	9.2	2.2	172.2	0.2	1.8	4.8	2.6
tantotempo	15.2	5.6	4	0.6	0	0.2	0	6.6	0	0	2.8	3	0	0.2	0.2	0.2	164.2	1.2	0.2	0
buonissimo	1.6	5	2.4	11.6	0	1.6	0	16.8	0.4	13.8	6.2	12.2	0	2.6	3.2	0.8	0.6	117.4	0.2	1.4
messidaccordo	2.2	0.4	0	0.2	0.8	1.4	7	0.2	0.8	1.8	1.4	0.6	0	5.6	5.6	4.8	0	0	147.8	15.8
sonostufo	0.6	0.4	0.4	2.8	0	1.8	1.8	0	3	10.2	1.6	2.2	0	3.2	9	1.4	0	3.6	15.6	135.8

Fig. 14: The confusion matrix of the average case for CNNLSTM on the Montalbano dataset. The confusion between gesture classes with *indistinctive movement* is pronounced, e.g., “vattene”, “vieniqui”, “perfetto”, and “tantotempo”.

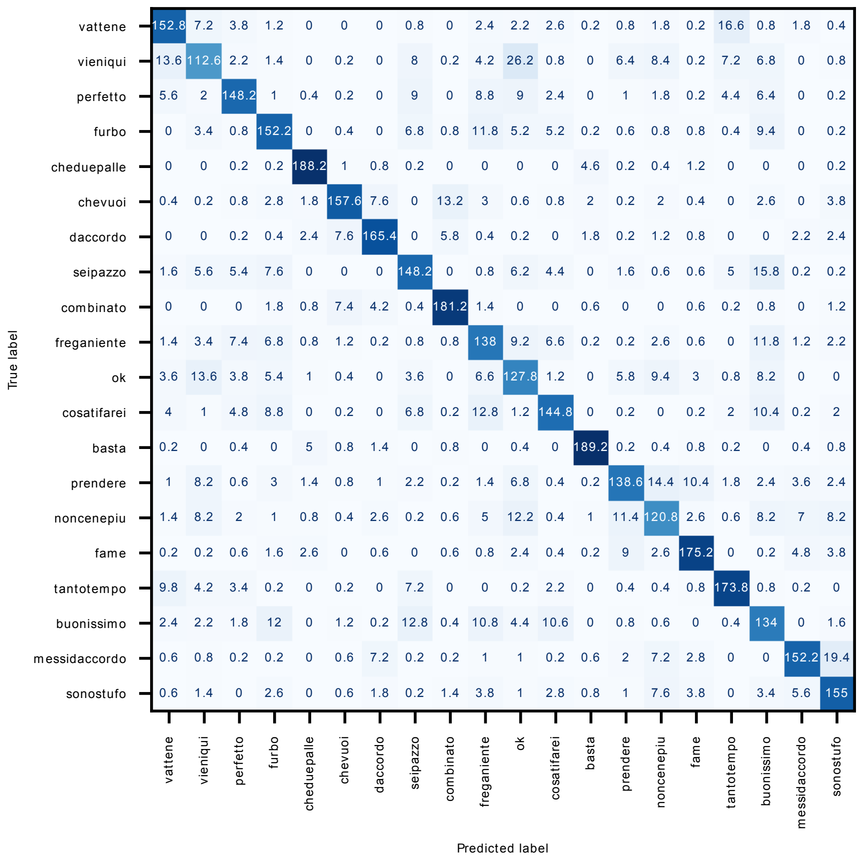
the confusion regarding class “buonissimo” is only slightly boosted in *Snapture* and *Snapture*$_{thold}$. We explain that by observing that “buonissimo” and “furbo” are similar in both the motion and handshape, i.e., extended index finger. The difference lies in the position the finger touches the face (under the eyes vs. on the cheek). Efficiently recognizing these gestures requires additional modalities, which we do not consider in our study. However, we will discuss this point later. Moreover, since *snapshot* is captured using one frame at the gesture’s peak, it is subject to influence by the corresponding hand orientation and light reflection. Thus, it becomes more challenging to distinguish between an open palm and an extended index finger, especially since the input is in grayscale (cf. Fig. 19).



	vattene	vieniqui	perfetto	furbo	cheduepalle	chevuoi	daccordo	seipazzo	combinato	freganiente	ok	cosatifarei	basta	prendere	noncenepiu	fame	tantotempo	buonissimo	messidaccordo	sonostufo
vattene	139.2	12.2	3	0.8	0	0.2	0	1.2	0.2	4	4	4.2	0	1.6	1.8	0.2	20.2	0	2.8	0.6
vieniqui	15	114.8	1.8	1.6	0	0.2	0.2	7.6	0.6	6.4	19	1.4	0	8	9.8	0.4	7.6	3.8	0.8	0
perfetto	5.2	3.6	145	1	0	0	0	9.4	0	9.8	8.6	4.2	0.2	0.8	3	0	5.4	3.2	0	0.4
furbo	0.4	3.6	1.4	137	0.2	0.4	0	6	1.2	16.8	7	7.4	0	1.4	0.4	1.4	0	10.4	0	2.2
cheduepalle	0	0.2	0.2	0	188.2	1.2	0.8	0	0	0.6	0.4	0	4.4	0	0.2	1	0.2	0	0	0
chevuoi	1.6	0.8	0.6	2.2	3.6	153.6	9.6	0.8	14.8	2	0.6	1.4	1.2	0.8	2.2	0.4	0.2	0.8	0.4	2.2
daccordo	0.2	0.8	0.4	0	2.2	7.4	161	0.4	5.8	0.2	0	0.2	1.8	0.4	1.2	1.2	0	0	4.2	3
seipazzo	1.6	5.2	7.6	6.4	0	0	0.2	146.2	0.2	1.4	5.2	5.4	0	1.8	0.6	0.4	5	15.6	0.2	0.2
combinato	0.2	0.2	0	2.2	0.8	8.2	8.2	0	174.8	2.6	0	0	1.6	0.2	0.4	0.2	0	0.8	0.2	0.6
freganiente	2.8	2.6	8.8	6.6	1	1.2	0.2	0.8	1.6	137.6	6.8	4.8	0	1.6	8.8	1	0	5.6	0.6	2.6
ok	3	12.8	4	5	0.2	0.6	0	3.8	0.2	6	126.6	1	0	11.8	10.8	1.6	1	4.6	0	1.4
cosatifarei	5.4	1.8	5.8	6.6	0	0	0	5	0.2	12.6	2	142.6	0.2	1.2	0.4	0	1.6	10.8	0	3
basta	0.6	0.2	0	0.2	3	0	2.4	0	1	0.2	0.6	0	187.4	1.8	0.2	1.4	0	0.2	0.8	1.6
prendere	1.2	5.4	0.8	1.2	0.8	0.4	0.8	2.8	0	2.6	7.6	0.2	0.4	140.8	16.4	11.8	2.2	2.6	1.6	1.6
noncenepiu	1	4.8	1	1	1	0.6	1.2	0.8	0.2	8	11.2	0	0.6	17	123.4	4.4	0.6	3.4	6.8	8
fame	0.2	0.4	0.2	1.6	2.8	0.2	0.6	1.6	0.4	0.6	2.2	0.4	0.4	12.8	2.2	170.6	0.2	0.2	5	3.4
tantotempo	11.8	4.2	2.6	0.6	0.4	0	0	10	0.2	0	1.4	1.4	0.2	1	1	0.2	168.6	0	0.2	0
buonissimo	1.4	2.6	1.2	10	0	1	0.2	17.4	0.8	13.6	7.8	14.2	0	0.8	1.4	0.4	0.8	122	0.2	2.2
messidaccordo	0.8	0.8	0	0	0.2	0.8	6.2	0	0.8	2	0.4	0	0.4	3.2	11	3	0.4	0	147.6	18.6
sonostufo	0.6	1.2	0	1.8	0.4	0.2	1.8	0.2	1	2.4	1.2	1.6	0.4	2.4	7.6	2.6	0	1.2	8.4	158.4

Fig. 15: The confusion matrix of the average case for *Snapture* on the Montalbano dataset. The confusion concerning gesture classes with *indistinctive* and *implicit* movements, e.g., “vattene”, “noncenepiu” and “ok”, is less pronounced than the CNNLSTM.

Explicit Movements: Five Montalbano gesture classes require a synchronized movement of both arms. We observe two types of movements under this category according to the way the arms are extended. “Chevuoi” and “combinato”, are performed using symmetric hand movements in which both arms move from the rest position to making a distinct shape at chest level. Due to the similarity of motion, the CNNLSTM comes short in terms of F1-scores, most noticeable for “chevuoi”. Furthermore, *Snapture* and *Snapture*$_{thold}$ present a noticeable F1-score boost for these classes (cf. Fig. 17). On the other hand, gestures “cheduepalle” and “basta” are also symmetric but made with a movement of both arms to the side of the body. Both gestures are used in a situation where a person is being decisive and implying “enough”. Therefore, the move-



	vattene	vieniqui	perfetto	furbo	cheduepalle	chevuoi	daccordo	seipazzo	combinato	freganiente	ok	cosatifarei	basta	prendere	noncenepiu	fame	tantotempo	buonissimo	messidaccordo	sonostufo
vattene	152.8	7.2	3.8	1.2	0	0	0	0.8	0	2.4	2.2	2.6	0.2	0.8	1.8	0.2	16.6	0.8	1.8	0.4
vieniqui	13.6	112.6	2.2	1.4	0	0.2	0	8	0.2	4.2	26.2	0.8	0	6.4	8.4	0.2	7.2	6.8	0	0.8
perfetto	5.6	2	148.2	1	0.4	0.2	0	9	0	8.8	9	2.4	0	1	1.8	0.2	4.4	6.4	0	0.2
furbo	0	3.4	0.8	152.2	0	0.4	0	6.8	0.8	11.8	5.2	5.2	0.2	0.6	0.8	0.8	0.4	9.4	0	0.2
cheduepalle	0	0	0.2	0.2	188.2	1	0.8	0.2	0	0	0	0	4.6	0.2	0.4	1.2	0	0	0	0.2
chevuoi	0.4	0.2	0.8	2.8	1.8	157.6	7.6	0	13.2	3	0.6	0.8	2	0.2	2	0.4	0	2.6	0	3.8
daccordo	0	0	0.2	0.4	2.4	7.6	165.4	0	5.8	0.4	0.2	0	1.8	0.2	1.2	0.8	0	0	2.2	2.4
seipazzo	1.6	5.6	5.4	7.6	0	0	0	148.2	0	0.8	6.2	4.4	0	1.6	0.6	0.6	5	15.8	0.2	0.2
combinato	0	0	0	1.8	0.8	7.4	4.2	0.4	181.2	1.4	0	0	0.6	0	0	0.6	0.2	0.8	0	1.2
freganiente	1.4	3.4	7.4	6.8	0.8	1.2	0.2	0.8	0.8	138	9.2	6.6	0.2	0.2	2.6	0.6	0	11.8	1.2	2.2
ok	3.6	13.6	3.8	5.4	1	0.4	0	3.6	0	6.6	127.8	1.2	0	5.8	9.4	3	0.8	8.2	0	0
cosatifarei	4	1	4.8	8.8	0	0.2	0	6.8	0.2	12.8	1.2	144.8	0	0.2	0	0.2	2	10.4	0.2	2
basta	0.2	0	0.4	0	5	0.8	1.4	0	0.8	0	0.4	0	189.2	0.2	0.4	0.8	0.2	0	0.4	0.8
prendere	1	8.2	0.6	3	1.4	0.8	1	2.2	0.2	1.4	6.8	0.4	0.2	138.6	14.4	10.4	1.8	2.4	3.6	2.4
noncenepiu	1.4	8.2	2	1	0.8	0.4	2.6	0.2	0.6	5	12.2	0.4	1	11.4	120.8	2.6	0.6	8.2	7	8.2
fame	0.2	0.2	0.6	1.6	2.6	0	0.6	0	0.6	0.8	2.4	0.4	0.2	9	2.6	175.2	0	0.2	4.8	3.8
tantotempo	9.8	4.2	3.4	0.2	0	0.2	0	7.2	0	0	0.2	2.2	0	0.4	0.4	0.8	173.8	0.8	0.2	0
buonissimo	2.4	2.2	1.8	12	0	1.2	0.2	12.8	0.4	10.8	4.4	10.6	0	0.8	0.6	0	0.4	134	0	1.6
messidaccordo	0.6	0.8	0.2	0.2	0	0.6	7.2	0.2	0.2	1	1	0.2	0.6	2	7.2	2.8	0	0	152.2	19.4
sonostufo	0.6	1.4	0	2.6	0	0.6	1.8	0.2	1.4	3.8	1	2.8	0.8	1	7.6	3.8	0	3.4	5.6	155

Fig. 16: The confusion matrix of the average case for *Snapture*$_{thold}$ on the Montalbano dataset. The confusion concerning gesture classes “vattene”, “furbo” and “buonissimo” is less pronounced than the CNNLSTM.

ment of the arm is quite firm, making it unique from the rest of the gesture vocabulary. Consequently, the CNNLSTM is efficient at picking up these movements. *Snapture* and *Snapture*$_{thold}$ only slightly improve over the performance of the CNNLSTM concerning these *explicit* movements since the handshape and finger arrangement play a minimal role in their recognition. In Fig. 20, we display a comparison between an *implicit* and *explicit* movement and their corresponding pre-processing step.

6 Discussion and Future Work

In this study, we proposed a hybrid (static/dynamic) gesture recognition architecture called *Snapture*. Our model integrates the hand pose alongside move-

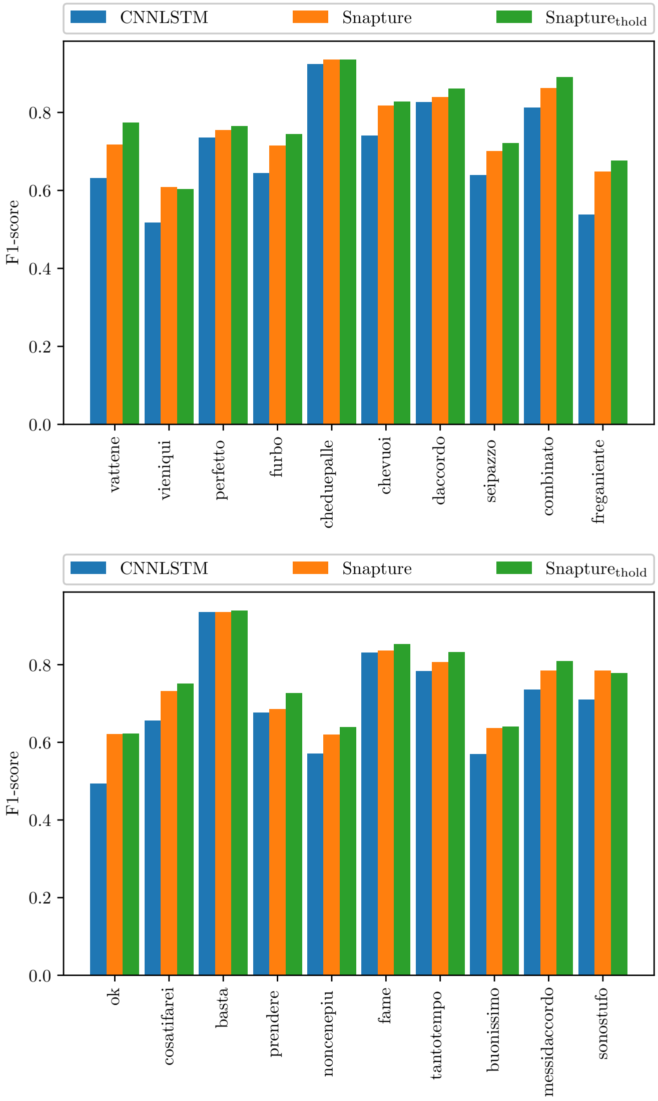
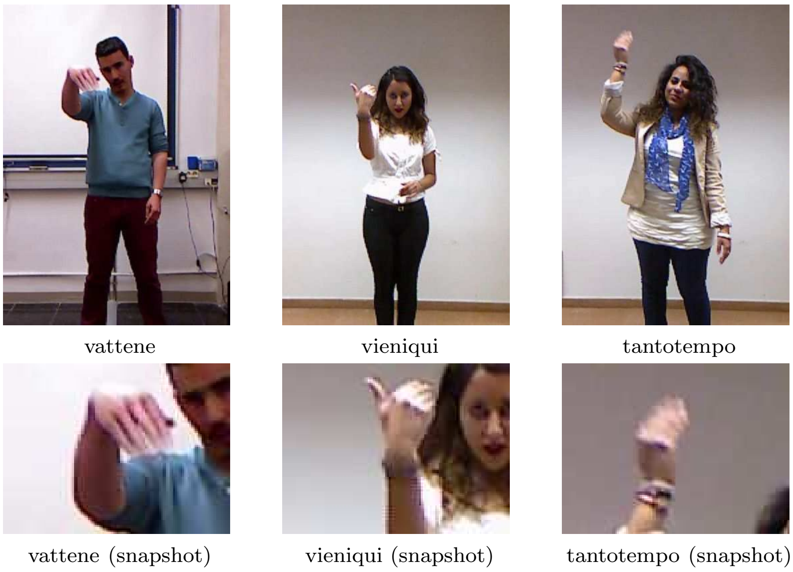


Fig. 17: A comparison of per-class F1-score values between the different approaches on the Montalbano dataset. *Snapture* improves the score on all classes except “basta”. The performance of *explicit* arm movements, e.g., “basta” and “cheduepalle” is comparable across the three models.



vattene

vieniqui

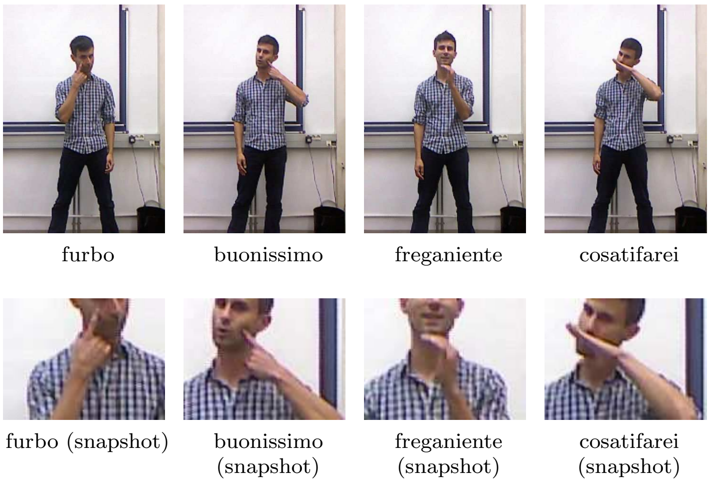
tantotempo

vattene (snapshot)

vieniqui (snapshot)

tantotempo (snapshot)

Fig. 18: Our *snapshot* extraction takes place using a single frame at the peak. Thus, a challenging scenario to our approach is when gestures that have a similar hand pose during the *stroke* phase.



furbo

buonissimo

freganiente

cosatifarei

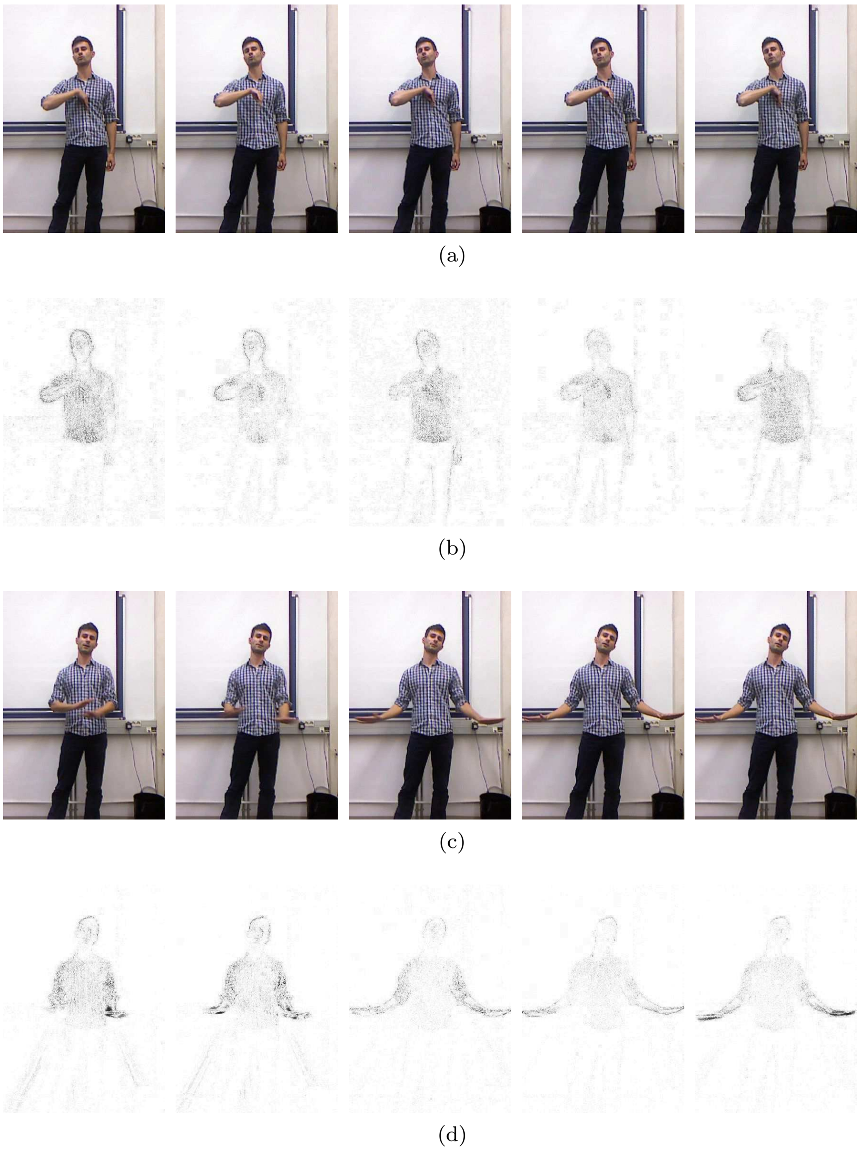
furbo (snapshot)

buonissimo (snapshot)

freganiente (snapshot)

cosatifarei (snapshot)

Fig. 19: Some challenges concerning class “buonissimo”: a) similarity in hand motion and pose with “furbo”. Therefore, another modality is required, which is not considered by approach, b) similarity in hand orientation and light reflection causes misclassifications with “freganiente” and “cosatifarei” in the worst case of our results. It becomes challenging to interpret the open palm under these conditions.



(a)

(b)

(c)

(d)

Fig. 20: A comparison between *implicit* and *explicit* hand movements. We observe missing hand details concerning “sonostufo” (b). In contrast, the *explicit* arm movement of “basta” is conserved (d). (a) and (c) depict the original sequence for clarity. We increase the contrast of (b) and (d) for clarity.

Table 3: The search space and optimal hyperparameter values (in bold) of each model in the *Montalbano* experiment.

Hyperparameter	**CNNLSTM**	**Snapture***
Learning rate	[0.01, **0.001**, 0.0001]	[0.01, **0.001**, 0.0001]
Number of epochs	[20, 40, 60, 80, **100**]	[20, 40, 60, 80, **100**]
Mini-batch size	[16, 32, **64**, 128]	[16, 32, 64, **128**]
Optimizer	[**Adam**, SGD]	[**Adam**, SGD]

*__Similar for *Snapture*$_{thold}$.__

Table 4: The results of the *Montalbano* experiment under the described settings. The reported metrics represent the mean of five trials, while the values in parentheses correspond to the standard deviation. The superior accuracy and F1-score values are in bold.

Model	**CNNLSTM**	**Snapture**	**Snapture**$_{\mathbf{thold}}$
Accuracy	0.699 (0.014)	0.755 (0.021)	**0.77** (0.008)
F1-score	0.701 (0.013)	0.752 (0.021)	**0.772** (0.007)
Time*	234.762 (0.115)	318.578 (0.428)	744.953 (0.724)

*In minutes.

ment through modular static and dynamic channels. Our work is motivated by the limitation of RGB techniques, such as the CNNLSTM network, across different gesture domains. Therefore, we evaluated our approach in the context of robot commands and co-speech gestures. In our experiments, we compared the performance of both our *Snapture* model and the CNNLSTM approach using the GRIT [20] and Montalbano [7] datasets. Moreover, we showed the superiority of our approach in the scope of *indistinctive* and *subtle* movements. Our evaluation and analysis demonstrated that considering the handshape and finger arrangement at the gesture's peak led to superior per-class F1-score values. Furthermore, we identified a lack of literature concerning the analysis of gesture motion profiles. Thus, we proposed an SSIM-based algorithm for analyzing motion profiles. We utilized this technique to propose a threshold-based extension *Snapture*$_{thold}$. The performance is further improved by regulating the static channel and bypassing the *blurriness* issue. We believe the unique characteristics of our approach make it potentially beneficial in the following domains: 1) emblematic hand gestures, which substitute words to convey a particular meaning, and 2) co-speech gestures, which accompany words as means of verbal communication. Although we do not consider speech in our approach, incorporating additional modalities through extra channels is straightforward due to the modularity of our architecture.

Vision-based approaches are highly influenced by similarities in hand movement patterns. Furthermore, they fall short in capturing delicate small-scale hand motions at the peak [17]. The effects of this phenomenon are limited in the GRIT dataset to the classes “hello”, “no”, and “stop”. However, robot commands are motion-oriented, designed to be unique and convey simple meanings, i.e., robot control. On the other hand, the Montalbano Italian gestures are part of human communication. Therefore, they are natural and tend to have a basic motion path, and rather involve particular hand and finger configurations. Our analysis of the classification behavior of the CNNLSTM reports F1-score values of lower than 0.7 across ten classes of the Montalbano classes. This issue is also prominent in recent state-of-the-art approaches. Many of the modern gesture recognition techniques, such as 3DCNN [23], ResNet [17] and Inception V3 [13] require advanced transform learning techniques and relatively long training times. In contrast, our system addresses this problem by using a simple additional static channel. Consequently, our approach facilitates a robot application due to the lightweight architecture. Furthermore, recent approaches that deal with the Montalbano dataset predominately utilize multimodalities, such as skeleton and depth data [23][13] alongside RGB, for the detection and extraction of the hand. Therefore, such approaches require a window of frames and suffer the occasional loss of joint information. We avoid such dependencies by using RGB data only, making our approach one of the few pure RGB-based models that operate on the Montalbano dataset. Thus, it is compatible with any system equipped with a camera, including robots.

Additionally, we show that *Snapture* enhances the performance by limiting the confusion between the classes that share the same movement path. Thus, it has a powerful false-positive limiting characteristic, making it a viable asset in critical scenario applications. One example of safety is shown in the literature in which the human operator has control over a robotic arm [13]. The study addresses a physical safety scenario where the communication is carried out through gestures. However, their network is trained on independent static and dynamic gestures. Thus, their framework can only recognize either a static or dynamic gesture at a given time and does not address the potential risk resulting from gesture confusion. Therefore, this contradicts their claim of a static and dynamic gesture recognition framework. In contrast, our system integrates both the spatial and temporal aspects of the gesture and considers both for classification. Furthermore, the scheme of fusing the static and dynamic features influences the system. Our approach operates on a single frame in the static channel, which has several advantages. First, it matches with Kendon's [12] model of gesticulation and concurrent speech. It was proven in the literature that the *stroke* phase plays an essential role in recognition. Second, the spatial and temporal traits are treated with equal importance. Thus, we can avoid the issues of fusing features at each time step. A dominance of particular modalities (RGB, depth, and skeleton) in the learning is reported by Wu et al. [23], making it more challenging to analyze the influence of each one on the final outcome. In contrast, our experimental design provides concrete evidence that classification performance concerning *indistinctive* and *subtle* movements can be boosted through the learning of hand details.

Furthermore, our observations on the GRIT and Montalbano datasets show high variability in hand preference. Besides hand dominance, fatigue and injuries are among the most common factors that drive the interchangeable use of both hands. Therefore, a robust system that works with subjects regardless of the dominant hand is desired. Our system accomplishes that by extracting the pose of the hand actively used when making the gesture. That makes our system unique to other studies that mirror all videos of left-handed subjects [23] or have a dedicated network for each hand [13]. Thus, our approach facilitates higher flexibility, which leads to less restrictive and guided HRI scenarios. However, this is one step towards a wider research domain.

One of the main future directions of our work is extending the model with the body pose information. By extracting the handshape information at the peak, our architecture is prone to confusion between the classes that share a similar hand pose at the *stroke* phase. Such faulty behavior can be avoided by integrating the body pose information through an additional static channel with a different cropping size. Our modular architectural design facilitates that by incorporating additional channels. Furthermore, gestures such as “furbo” and “buonissimo” are almost identical at the peak with minor distinction. Precise recognition of these classes depends on the context and requires additional speech or facial information. It remains interesting to extend our model with additional facial or speech features through added networks such as CNNs. Finally, one of the main disadvantages of threshold-based approaches is the lack of guarantee of generalizing to new samples. Therefore, introducing robustness by learning the cut-off values of the threshold is desirable. However, our results show that the *blur* phenomena is non-trivial. We hope that our work raises more attention to the quality of collected RGB gesture datasets and encourages more research in the area of producing affordable higher-quality cameras that are compatible with robots.

7 Conclusion

Despite the advantages of RGB-based vision-based hand gesture recognition frameworks, they are still challenged by the confusion between gestures with similar paths as well as the loss of hand details. In this study, we presented a novel architecture called *Snapture* which integrates both the static and dynamic information of a gesture. Our RGB-only dependency and lightweight architecture design allow compatibility with any system equipped with a camera, including robots. We also suggested an algorithm for analyzing gesture motion profiles, which is essential for revealing the unique characteristics of a gesture domain. Our results provide evidence that incorporating the hand pose at the gesture’s peak with motion information offers a better solution to the issues of *indistinctive* and *subtle* movements. They also demonstrate that these challenges are more prominent in the context of co-speech gestures compared to robot commands. Therefore, it hints at the substance of evaluating gesture recognition frameworks across multiple gesture domains. Additionally, our $Snapture_{thold}$ extension highlights the influence of RGB data quality on the performance of the model and provides means for optimization based on data quality. Overall, we hope our work provides a solid step at bridging the gap between static and dynamic gestures and leading to applications that foster immersive and less controlled HRI experiences.

Acknowledgements This work was partially supported by the DFG under project CML (TRR 169) and BMWK under project KI-SIGS. The authors would like to thank Philipp Allgeuer for the insightful comments that helped improve the first draft of this manuscript.

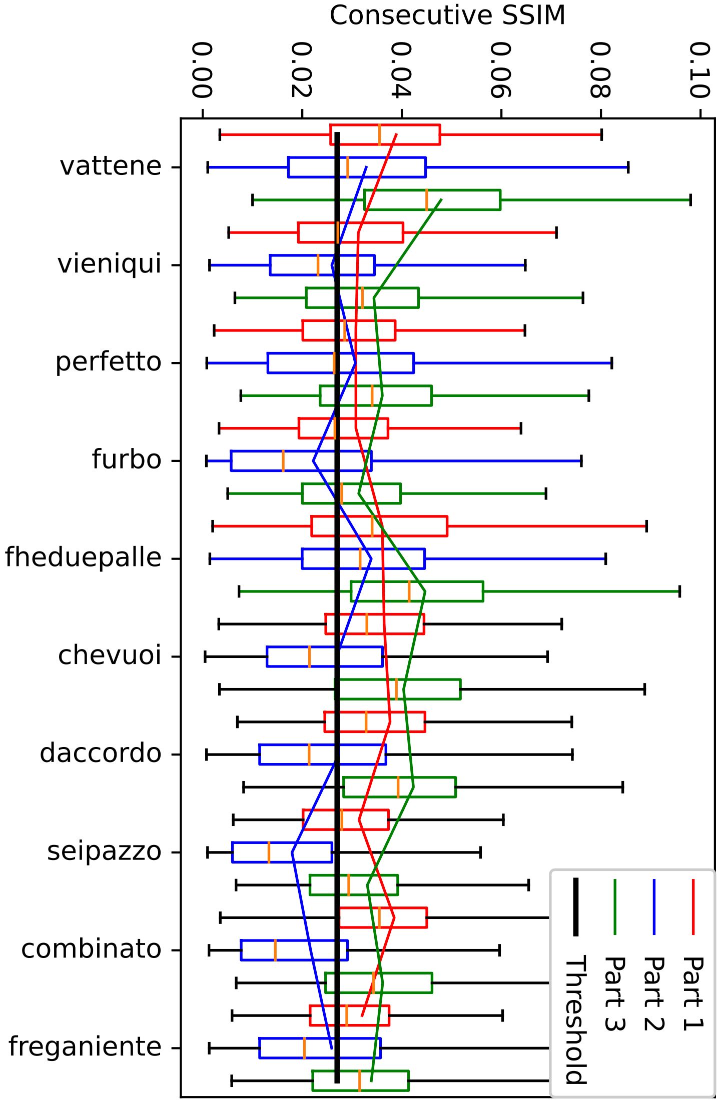
Compliance with Ethical Standards

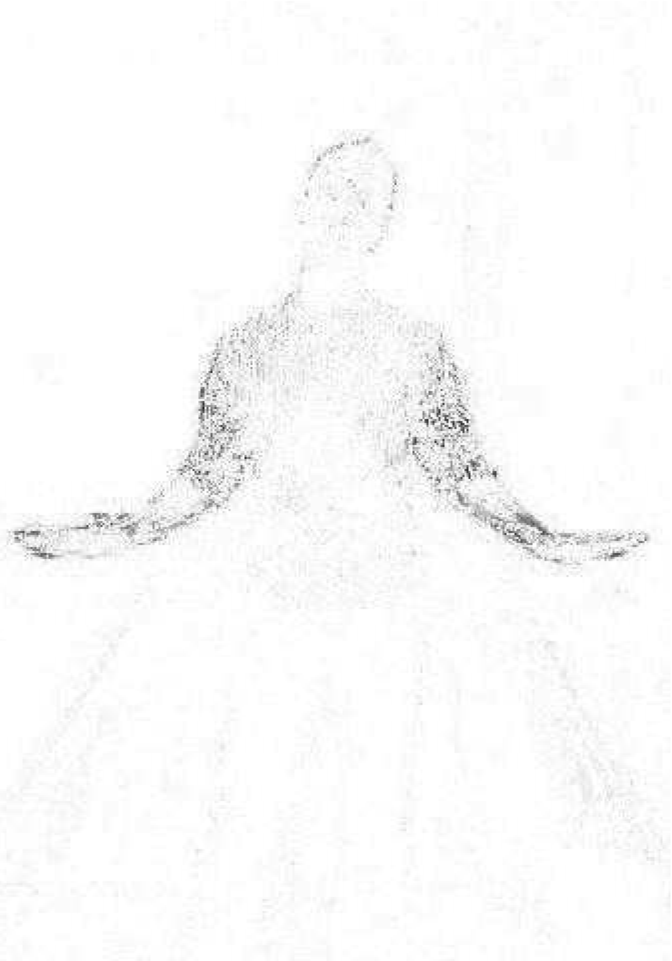
The authors declare that they have no conflict of interest.

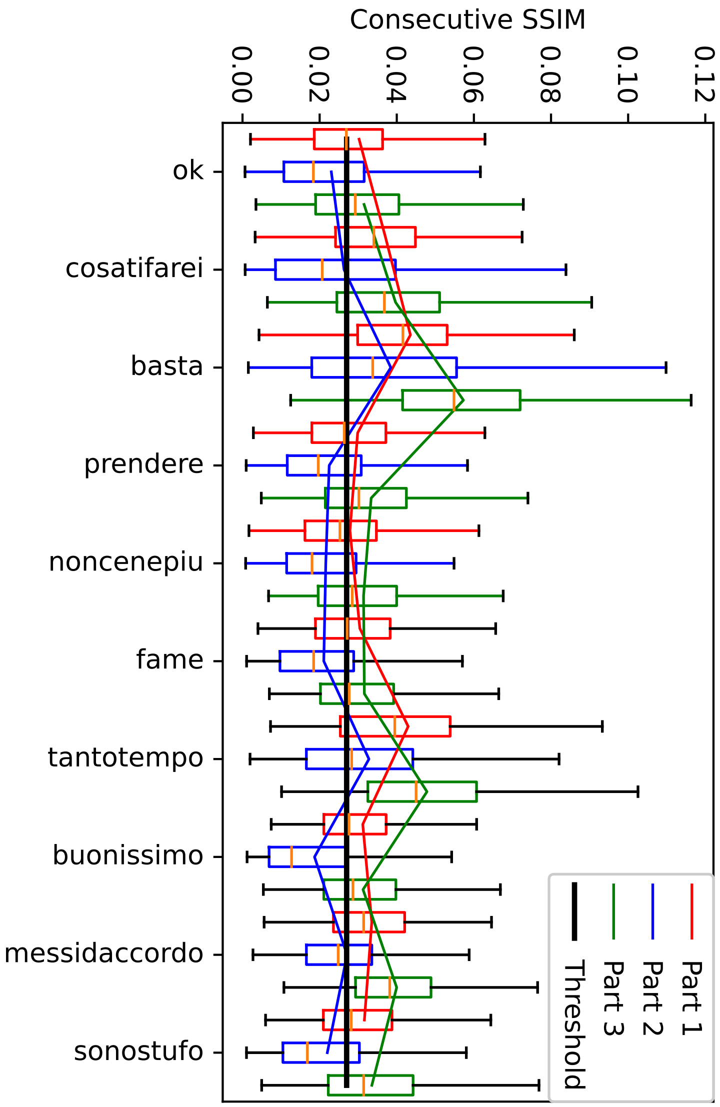
References

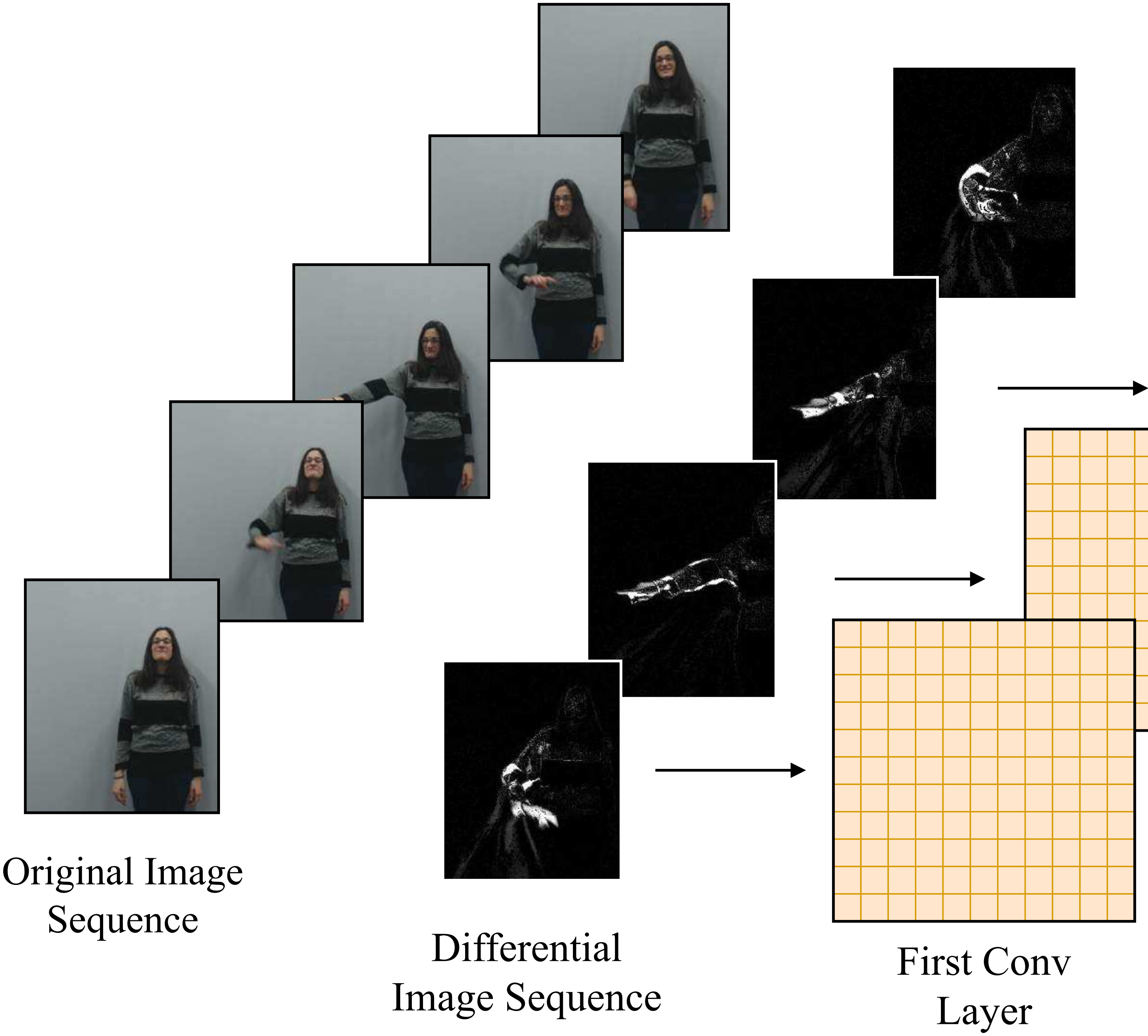
1. van Amsterdam, B., Clarkson, M., Stoyanov, D.: Gesture recognition in robotic surgery: a review (2021)
2. Anwar, S., Sinha, S.K., Vivek, S., Ashank, V.: Hand gesture recognition: A survey. In: V. Nath, J.K. Mandal (eds.) Nanoelectronics, Circuits and Communication Systems, pp. 365–371. Springer Singapore, Singapore (2019)
3. Asadi, M., Clapés, A., Bellantonio, M., Escalante, H.J., Ponce-López, V., Baró, X., Guyon, I., Kasaei, S., Escalera, S.: A survey on deep learning based approaches for action and gesture recognition in image sequences (2017)
4. Auge, D., Wenner, P., Mueller, E.: Hand Gesture Recognition using Hierarchical Temporal Memory on Radar Sequence Data. Bernstein Conference (2020). DOI 10.12751/nncn.bc2020.0022
5. Basilio, J.A.M., Torres, G.A., Pérez, G.S., Medina, L.K.T., Meana, H.M.P.: Explicit image detection using YCbCr space color model as skin detection. In: Proceedings of the 2011 American Conference on Applied Mathematics and the 5th WSEAS International Conference on Computer Engineering and Applications, AMERICAN-MATH'11/CEA'11, p. 123–128. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA (2011)
6. Chakraborty, B., Sarma, D., Bhuyan, M., MacDorman, K.: A review of constraints on vision-based gesture recognition for human-computer interaction. IET Computer Vision **12** (2017). DOI 10.1049/iet-cvi.2017.0052
7. Escalera, S., Baró, X., Gonzàlez, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce-López, V., Escalante, H.J., Shotton, J., Guyon, I.: Chalearn looking at people challenge 2014: Dataset and results. In: L. Agapito, M.M. Bronstein, C. Rother (eds.) Computer Vision - ECCV 2014 Workshops, pp. 459–473. Springer International Publishing, Cham (2015)
8. Escalera, S., Guyon, I., Athitsos, V.: Gesture Recognition, 1st edn. Springer Publishing Company, Incorporated (2018)
9. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research - Proceedings Track **9**, 249–256 (2010)
10. Hsu, R.L., Abdel-Mottaleb, M., Jain, A.: Face detection in color images. Pattern Analysis and Machine Intelligence, IEEE Transactions on **1**, 696–706 (2002). DOI 10.1109/34.1000242
11. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: F. Bach, D. Blei (eds.) Proceedings of the 32nd International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 37, pp. 448–456. PMLR, Lille, France (2015). URL `https://proceedings.mlr.press/v37/ioffe15.html`
12. Kendon, A.: Gesticulation and Speech: Two Aspects of the Process of Utterance, pp. 207–228. De Gruyter Mouton (2011). DOI doi:10.1515/9783110813098.207. URL `https://doi.org/10.1515/9783110813098.207`

13. Mazhar, O., Ramdani, S., Cherubini, A.: A deep learning framework for recognizing both static and dynamic gestures. Sensors **21**, 2227 (2021). DOI 10.3390/s21062227
14. Pham, V., Bluche, T., Kermorvant, C., Louradour, J.: Dropout improves recurrent neural networks for handwriting recognition. In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 285–290 (2014). DOI 10.1109/ICFHR.2014.55
15. Qiu-yu, Z., Lu, J.c., Zhang, M.y., Duan, H.x., Lv, L.: Hand gesture segmentation method based on YCbCr color space and k-means clustering. International Journal of Signal Processing, Image Processing and Pattern Recognition **8**, 105–116 (2015). DOI 10.14257/ijsip.2015.8.5.11
16. Renard, F., Guedria, S., De Palma, N., Vuillerme, N.: Variability and reproducibility in deep learning for medical image segmentation. Scientific Reports **10** (2020). DOI 10.1038/s41598-020-69920-0
17. dos Santos, C.C., Samatelo, J.L.A., Vassallo, R.F.: Dynamic gesture recognition by using CNNs and star RGB: a temporal information condensation. Neurocomputing **400**, 238–254 (2020). DOI https://doi.org/10.1016/j.neucom.2020.03.038. URL `https://www.sciencedirect.com/science/article/pii/S092523122030391X`
18. Siddharth, S., Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey. Artificial Intelligence Review **43**, 1–54 (2015)
19. Tsironi, E., Barros, P., Weber, C., Wermter, S.: An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. Neurocomputing **268**, 76–86 (2017). DOI https://doi.org/10.1016/j.neucom.2016.12.088. URL `http://www.sciencedirect.com/science/article/pii/S0925231217307555`. Advances in artificial neural networks, machine learning and computational intelligence
20. Tsironi, E., Barros, P., Wermter, S.: Gesture recognition with a convolutional long short-term memory recurrent neural network. In: ESANN (2016)
21. Wang, T., Li, Y., Hu, J., Khan, A., Liu, L., Li, C., Hashmi, A., Ran, M.: A survey on vision-based hand gesture recognition. In: A. Basu, S. Berretti (eds.) Smart Multimedia, pp. 219–231. Springer International Publishing, Cham (2018)
22. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004). DOI 10.1109/TIP.2003.819861
23. Wu, D., Pigou, L., Kindermans, P.J., Le, N., Shao, L., Dambre, J., Odobez, J.M.: Deep dynamic neural networks for multimodal gesture segmentation and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**, 1–1 (2016). DOI 10.1109/TPAMI.2016.2537340


Consecutive SSIM
0.00
0.02
0.04
0.06
0.08
0.10
vattene
vieniqui
perfetto
furbo
fheduepalle
chevuoi
daccordo
seipazzo
combinato
freganiente
Part 1
Part 2
Part 3
Threshold




Consecutive SSIM
0.00
0.02
0.04
0.06
0.08
0.10
0.12
ok
cosatifarei
basta
prendere
noncenepiu
fame
tantotempo
buonissimo
messidaccordo
sonostufo
Part 1
Part 2
Part 3
Threshold


Original Image Sequence
Differential Image Sequence
First Conv Layer