# Continuous Hand Gesture Recognition using Deep Coarse and Fine Hand Features

Hazem Wannous, Jean-Philippe Vandeborre

# Continuous Hand Gesture Recognition using Deep Coarse and Fine Hand Features

Hazem Wannous
hazem.wannous@univ-lille.fr
Jean-Philippe Vandeborre
jean-philippe.vandeborre@imt-nord-europe.fr

IMT Nord Europe
University of Lille, CNRS
UMR 9189 - CRIStAL
F-59000 Lille, France

**Abstract**

Using hand gestures as a HCI modality introduces intuitive and easy-to-use interfaces for a wide range of applications. However, the hand is an object with a high number of degrees of freedom and with high similarities derived from the heterogeneities of possible gestures. Moreover, the online detection of a gesture as soon as it happens in a video stream is a very challenging problem. To address these difficulties, we introduce an effective deep learning based approach, which takes advantage of the combined description of the hand shape and its temporal variation. First, we employ a transfer learning strategy to learn coarse and fine hand features from depth image dataset originally created for hand pose estimation. Then, we model the temporal aspect separately of the hand poses and its shape variations over the time using recurrent models, before merging. Our approach achieve significant performance for the task of hand gesture detection and recognition. In online scenario, results show that our approach is able to detect an occurring gesture and to recognize it far before its end, making our system efficient for real-time interactive applications.

## 1 Introduction

Hand gestures are the most natural and intuitive non-verbal communication medium while using a computer, and related research efforts have recently boosted interest. Additionally, the analysis and the interpretation of human behavior from visual input is one of the most trendy computer vision fields of research. Hand gesture analysis has been widely investigated in the literature, especially from 2D videos captured with RGB cameras. This last decade, the 3D information provided by depth sensors has prompted much research around this cue for hand pose estimation and hand gesture recognition tasks.

Despite an increasing amount of methods proposed over the last few years, defining an online gesture recognition system, robust enough to work in end-user applications, remains a challenge. Dynamic hand gestures can be defined by shape variations of the hand during sequences (e.g. gestures performed by fingers), or by hand movements (e.g. swipe gestures), and often both. These multiple characteristics, which have to be taken into account, make the process of feature learning harder as it has to learn mutually spatial and temporal information. In order to fully extract relevant features of complex hand gestures using raw data, neural networks need a large number of layers which increase their computation complexity. However, the computation complexity has to be small enough so that the algorithm can

predict an incoming gesture in real time. Some methods present acceptable runtime results using very deep networks but these methods require a powerful hardware with several GPUs. Currently, this hardware configuration is too expensive and not suitable for real-world applications. This work is motivated firstly by the powerful capability of deep neural networks to learn discriminative representations for images and videos, and second by the successful usage of recent and inexpensive depth sensors for hand pose estimation. In particular, we build a deep neural network architecture performing an online hand gesture detection and recognition. We take over the whole pipeline of the hand gesture analysis from the hand pose estimation step to the recognition process. The contribution presented in this paper is twofold: (1) We introduce of a new dataset of heterogeneous gestures recorded in an online scenario by a depth camera. (2) We propose a light but efficient approach for online recognition of hand gestures. Simplicity and lightness is one of our goals for HCI applications. Our approach takes into account fine and coarse features of hand shape with a fine tuning fusion.

The rest of this paper is structured as follows. Related work on 3D hand gesture recognition are reviewed in Section 2. Our approach is described in Section 3. In Section 4, the strengths of our approach in terms of online detection and recognition are demonstrated on two datasets before concluding in Section 5.

## 2 Related Work

Gesture recognition has been a widely explored topic in computer vision. We will focus here only in reviewing the works on 3D hand gesture recognition we consider relevant to two main categories – handcrafted and deep learning based methods – using depth images. In the **handcrafted** approaches, 3D depth information is used to recognize hand silhouettes or simply hand areas in order to extract features from a segmented hand region [14, 27, 28, 32]. The temporal aspect of hand motion is also exploited by considering the gesture as a sequence of hand shapes [6, 8, 13, 20, 29, 36]. In order to study hand gesture recognition in a real-time scenario for automotive interfaces, Ohn-Bar and Trivedi [25] made a publicly available dataset of 19 gestures performed in a car captured with a Microsoft Kinect. They compared the accuracy of gesture recognition using several known depth features (HOG, HOG3D, HOG$^2$). De Smedt *et al.* [4, 5] investigate the use of a hand skeleton model in a dynamic hand gesture recognition solution.

Like many research areas in pattern recognition, **deep learning** approaches have recently shown a particularity high performances for hand gesture recognition. Convolutional neural networks [15] designed to take images as input have been used for static hand gesture recognition using RGB data [17, 21] and/or depth maps [16]. Neverova *et al.* [23] designed a multi-modal deep learning framework which takes as inputs: RGB, depth, audio stream and body skeleton data. Molchanov *et al.* [18] proposed a dynamic hand gesture algorithm using a two-stream 3DCNN which takes as inputs stacked image gradients and depth maps to classify sequence of images. They later enhanced their method and proposed a dynamic hand gesture algorithm – called R3DCNN [19] – using a larger 3DCNN composed of eight convolutions on sequences of RGB and depth images. In addition, they used a Connectionist Temporal Classification [9] as the cost function. To overcome the hungriness of deep learning algorithms, they pre-trained their model on the large-scale Sport-1M [12] human action recognition dataset. If they claimed to obtain real-time results, they used a powerful hardware configuration not suitable for public use. Finally, Narayana *et al.* [22] present an interesting approach with a significant improvement, using convolutional networks of sev-

eral modality of data, by focusing attention within the scene. However, except Molchanov's approach [18], all these methods do not take into account the online aspect of the gesture and do not perform a continuous recognition of gesture.

# 3 Approach

The dynamic aspect of gesture sequences requires the use of time-series based models, such as 3DCNN on sub-sequences of depth images, or Recurrent Neural Network (RNN) on lighter data sequences, like hand joints resulting from a hand pose estimation method. The first one requires a powerful hardware configuration and computational complexity, whereas the second one lacks of efficiency related to the loss of information due to the lack of robustness of current hand pose estimators. On the other hand, to describe the gesture, hand postures along the sequence are relevant features, but also the temporal variation of both hand shape and its motions need to be considered. Taking into account these multiple characteristics makes harder the learning process as it has to learn both spatial and temporal information. All these considerations lead us to address the problem of hand gesture recognition within a framework based separately on learned temporal variation of coarse and fine features across the gesture sequence. Those features are both extracted from a Convolutional Neural Network (CNN) trained on depth frames of the gesture sequence.

We first propose to use a transfer learning strategy to extract coarse and fine hand features for hand gesture recognition purpose. To do so, we train a CNN for hand pose estimation using the ICVL dataset [30]. Note that a training set of depth images of this dataset labeled with the 3D joint locations is available, and it is the only one of a close application. The training of the CNN allows to obtain two distinct representations for each time step of a hand depth image sequence: hand fine features $J_t$, which represent hand pose, and a hand coarse features $X_t$ which represents the coarse hand shape in a high dimensional space. Thus, original hand depth image sequences, $s_{original} = \{I_t\}_{t=1...N}$, are transformed into two different sets of sequences as follows: $s_{fine} = \{J_t\}_{t=1...N}$ and $s_{coarse} = \{X_t\}_{t=1...N}$ for a sequence of $N$ frames. Both sequences are fed to two RNNs: $RNN_{fine}$ and the $RNN_{coarse}$, in order to model the temporal aspect separately of the hand poses and the shape variations over the time. Finally, results are merged to perform the recognition of hand gestures. Figure 1 summarizes the architecture of our approach.

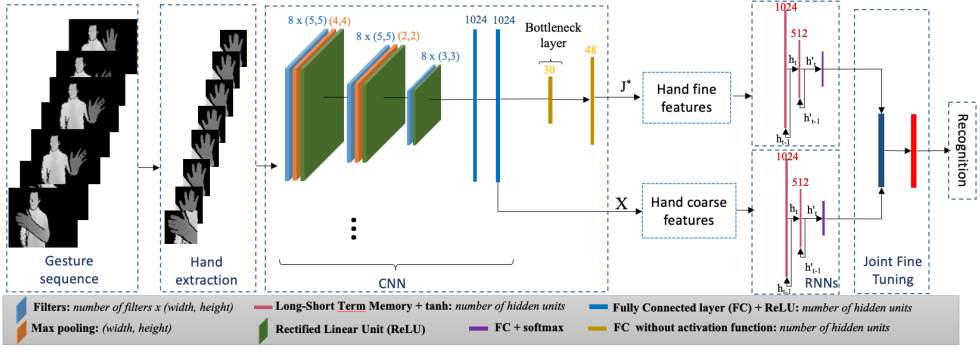



Figure 1: Overview of our proposed approach.

Note that a pre-processing step is first applied to extract the region-of-interest (ROI) of the hand assuming the hand is the closest object to the camera. The estimation is then refined using a 3D bounding box around the center of the mass, from which we can extract a cropped image of the hand and we compute its center of the mass in the original image space.

## 3.1 Spatial CNN feature extraction

The fist part of our architecture is a CNN model which uses prior enforcement implemented by introducing a *bottleneck* in the penultimate layer, having a smaller size than the final one which outputs the 3D hand joint coordinates. The network is then forced to learn a low representation of the data, describing physical constraints of the hand. Indeed, there are strong correlations between 3D joint locations, and a lower dimensional space of $3 \times K$ are sufficient to parameterize a 3D hand pose of $K$ joints [34], but not enough to describe the gesture. Thereby, the CNN first maps the image to a high dimensional space vector. Then, it follows a "bottleneck" layer with a smaller size that the desired output to model the physical constraints over the hand topology. Using this model, we extract two feature vectors from a hand depth image at coarse and fine level, respectively hand shape features $X$ and hand joint features $J^*$. $X$ describes the coarse hand shape without taking into account the details of its topology, and lying in $R^{1024}$, whereas $J^*$ represents 3D hand joint locations in the original depth image, and lying in $R^{3*K}$ with $K$ the number of hand joints.

Parameters of the CNN model have to be initialized before the training step. A common way to generate the values is to use a random normal distribution. An exception is made for the bottleneck layer as we can help the network using prior knowledge. We initialize its weights with the 30 major components from a Principal Component Analysis (PCA) of the hand joint label space of the training set. As the cost function, we minimize the Huber loss to evaluate the differences between the hand pose ground-truth and the output of the network. As a cost function, we minimize the Huber loss to evaluate the differences between the hand pose ground-truth $J$ and the output of the network noted $\hat{J}$:

$$H(J,\hat{J},\delta) = \begin{cases} \frac{1}{2}(J-\hat{J})^2 & \text{for } |J-\hat{J}| \leq \delta, \\ \delta\,|J-\hat{J}| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases} \tag{1}$$

The Hubert loss is thus quadratic when the error is small ($\leq \delta$) and linear when it becomes larger. Consequently, this loss function is less sensitive to noisy annotations (which imply large errors) than the squared error loss function.

## 3.2 Temporal learning

The hand gesture can be represented, according to the coarse feature cue, as a sequence of feature vector $s = \{X_t\}_{t=1...N}$ which is an ordered list of $N$ vectors and $X_t \in R^{1024}$. It can also be represented, according to the fine feature cue, as a sequence of feature vector $s = \{[j_1,\ldots,j_K]\}_{t=1...N}$ be an ordered list of $N$ vectors and $j_i \in J$. To model the temporal aspect of gestures, we feed separately these sequences to two RNNs, noted respectively $RNN_{coarse}$ and $RNN_{fine}$, each one composed of two stacked LSTM layers.

During the training phase of both RNNs, a weight decay and a dropout strategy are applied to prevent overfitting. Networks are trained using the Back-Propagation-Through-Time (BPTT) algorithm [33]. BPTT is equivalent of unrolling the recurrent layers, transforming them into a multi-layer feed-forward network of depth $N$; where $N$ is the number of frames

in the gesture sequence. The standard gradient-based back-propagation is then used. We average the gradients to consolidate weight updates to duplicated unrolling. The learning rate decreases following the number of epochs $ne$ by $lr = 0.001 \times N_0 e^{-\lambda \times ne}$. Networks try to minimize the cross-entropy cost. To increase variability in the training examples, we apply random horizontal, vertical and depth translations on depth image sequences before each learning iteration. Since recurrent connections can learn the specific order of gesture sequences in the training set, we randomly permute the training gesture videos before each new epoch.

To fuse the outputs of $RNN_{coarse}$ and $RNN_{fine}$, we propose a joint-fine-tuning method in order to enhance the classification process. It consists in retraining the two last softmax layers of the 2 *RNNs* while forcing their sum to be a representation of both networks. This strategy allows the network to learn a joint representation of network outputs, without adding parameters to the model and so, does not increase its complexity. Since both networks are trained separately, we retrain last fully connected layers before the softmax activation functions with a new cost function, noted $\mathcal{L}_{fusion}$, defined as follows:

$$\mathcal{L}_{fusion} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \lambda_3 \mathcal{L}_3 \tag{2}$$

where $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ are respectively loss functions computed on the $RNN_{coarse}$, the $RNN_{fine}$ and the sum of both outputs. The $\lambda_1$, $\lambda_2$ and $\lambda_3$ are tuning parameters. Each cost function is a cross entropy function. Let $l_1$ and $l_2$ be respectively the output values of the network $RNN_{fine}$ and $RNN_{coarse}$, $\mathcal{L}_3$ is then defined as follow:

$$y_3^i = softmax(l_1^i + l_2^i) \tag{3}$$

$$\mathcal{L}_3 = -\frac{1}{N} \sum_{i=1}^{N} \left[ y^i \log \hat{y}_3^i + (1 - y^i) \log(1 - \hat{y}_3^i) \right] \tag{4}$$

The final decision is obtained using $y_3^i$: $\hat{y} = {}_i(y_3^i)$

As a result, we utilize three loss functions in the training step: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$. $\mathcal{L}_1$ and $\mathcal{L}_2$ are used to regulate, respectively, both streams and avoid that one of them vanished under the weight of the other. $\mathcal{L}_3$ is used to optimize the fusion of the two modalities. Consequently, we use only $y_3$ for prediction as it is impacted by both RNNs.

## 3.3 Continuous recognition issue

An unsegmented stream of gestures contains a lot of unwanted and meaningless hand motions that do not belong to none of the gesture categories. First, hand gesture movements are often composed of three phases: (1) *pre-stroke* phase, which is composed of hand motions happening before the relevant gesture when the user needs to put his/her hand in a starting position; (2) *nucleus* phase, where the hand gesture is performed and has meanings. (3) *post-stroke* phase, which is composed of hand motions happening after the relevant gesture when the user wants to move back his/her hand to a restful position. Additionally, a stream of gestures contains motions between the gestures.

To tackle with the continuous recognition issues, our approach consists of extending the dictionary of existing gestures by adding a *garbage* class such as: $Y' = Y \cup \{ no\_gesture \}$. All frames which do not belong to a nucleus phase are labeled with this new class. To analyze the online detection and recognition capacity of our method, we use three metrics:

the Receiver Operating Characteristic (ROC) curve [1] and the Normalized Time to Detect (NTtD) [10] for the detection analysis and finally the recognition accuracy to analyze the recognition process. First, the ROC plots the True Positive Rate (TPR) – when the detector fires inside a nucleus phase – versus the False Positive Rate (FPR) – when the detector fires outside the nucleus phase. We use the area under the ROC for evaluating the detector accuracy. Second, the NTtD defines the fraction of the nucleus that has occurred, from $a$ to $b$, before the system fires a successful detection, $a \leq t \leq b$: $NTtD = \frac{t-a+1}{b-a+1}$ By adjusting the detection threshold chosen using the ROC curve, one can achieve lower NTtD at the cost of higher FPR and vice versa.

# 4 Experiments

We address in this work the online recognition of hand gestures in challenging conditions, including the heterogeneity of the hand shape depending on the set of fingers used to perform fine-grained gestures. Two public datasets meet these requirements/challenges which are DHG-14/28 [9] and NVIDIA [19]. Recently, we recorded a new challenging dataset called *Online Dynamic Hand Gesture (Online DHG)*[1] similarly to DHG-14/28 dataset [5], but with one difference that the gesture sequences are captured in continuous scenario. Indeed, Online DHG dataset provides 280 sequences of 10 continuous (unsegmented) gestures occurring sequentially. Between meaningful gestures, participants were free to take back a restful position without any suggestions from the recorders. In addition, comparing to the sequence-wise labeling of the first version, a label for each frame of the gesture sequences is provided. Frames between meaningful gestures are labeled as belonging to a *no gesture* class. Our experiments are limited to these two datasets, since they are, to our knowledge, the only have been captured in an online scenario with different phases of a continuous depth stream of gestures and with frame-wise annotation.

## 4.1 Implementation details

To extract the deep features of hand pose and its shape, we train our CNN on the ICVL dataset [30], which comprises a training set of over 180,000 depth images showing various hand poses. The dataset is recorded using a time-of-flight *Intel Creative Interactive Gesture Camera* and has 16 annotated 3D joints for each depth image. Depth images have a high quality with little noise. We use the Hubert loss function defined in Equation 1. We choose a sensitive factor to error $\delta = 500$ (we remind that 3D hand coordinates annotations are given in *mm*). It means that errors on the joint location prediction superior to half a centimeter is linear while smaller errors are quadratic. Weight decay is applied with a regularization factor equal to 0.001. The networks are trained with a batch size of 128 for 100 epochs. The initial learning rate $lr$ is set to 0.01 with a momentum of 0.9. To avoid overfitting while training the recurrent layers in the two streams of RNN, weight decay is applied with a regularization factor equal to 0.001. The dropout strategy has a probabilistic value equal to 25%. We stop the training after 30 epochs to avoid learning training dataset specification. The initial learning rate $lr$ is set to 0.001. For the data augmentation step, the ranges of the horizontal and vertical translations are $\pm 20$ pixels and the range of the depth translation is $\pm 100$. Parameters for each translation are drawn from a uniform distribution. For the

[1] http://www-rech.telecom-lille.fr/online-dhg

experimental tests, we randomly split the two datasets into a training (70%) and a test (30%) sets, resulting in 196 training and 84 test unsegmented videos for Online DHG and 1050 training and 482 test sequences for NVIDIA Dataset.

## 4.2 Online DHG dataset

Let's first analyze the 'offline' recognition process on the Online GHG dataset by extracting the gesture nucleus, which results in 2800 pre-segmented gestures of either 14 or 28 distinct labels, where the same gestures performed with one finger or the whole hand (SHREC'17 dataset [3]). In the following experiments, we call SHREC'17 to indicate the pre-segmented sequences and *ODHG* to indicate the unsegmented sequences (online). On SHREC'17 dataset, our approach achieves accuracies of 94.4% and 90.7% respectively for 14 and 28 gestures. Table 1 presents a comparison of recognition accuracies obtained by our approach and most state-of-the-art methods including traditional hand-crafted feature approaches [4, 25, 26], deep learning based approaches [2, 3, 11, 24].

For the online scenario, our solution is considering adding a garbage class $\{no\_gesture\}$. Consequently, the softmax layer outputs a class-conditional probability for this additional garbage class. All frames which do not belong to a nucleus phase are labeled with this new class. To detect the presence of any one of the 25 gestures relative to $\{no\_gesture\}$, we compare the highest current class probability output of our approach to a threshold $\xi \in [0,1]$. When the detection threshold is exceeded, a classification label is assigned to the most probable class. To evaluate the online capability of our approach, we used the unsegmented sequences of gestures labeled following 28 gestures. After plotting the ROC curve, we obtained an Area Under the Curve equal to 0.91. We choose a gesture detection threshold equal to 0.15 as it shows a good trade-off between a high TPR (85%) and a low FPR (17%).

The average NTtD across all gesture classes is 0.2104, which means that, in average, a gesture can be detected after only 21% of its nucleus. We note that nucleus of *fine* gestures are shorter than those of *coarse* gestures. Moreover, *Swipe* gestures that contain multiple motions, such as *Swipe V*, *X* and +, have naturally the longest nucleus. Using the detection upstream step, we obtain an overall online recognition accuracy of 82.2%. Due to the detection issues of gestures, the recognition accuracy decreases by 8.3% compared to the easiest task of pre-segmented gesture recognition. This difference can result from incorrect gesture detection or from confusion between gestures with similar parts. For example, the *Swipe Down* gesture performed with the whole hand obtains an *offline* recognition accuracy up to 87%. In the online scenario, the accuracy decreases by 57% and a high confusion up to 24% appears with the *Swipe V* gesture. This can be explained as the first half a *Swipe V* gesture is extremely similar to the *Swipe Down* gesture. In addition, the recognition process suffers from an incorrect prior gesture detection.

### 4.2.1 NVIDIA dataset

The dataset has been captured following a HCI based on hand gestures in a car scenario. While the user is not performing a gesture, his/her hands still move to control the vehicle and, so, is highly suitable to study gesture detection. To detect the presence of any one of all gestures (25 classes) relative to *no_gesture*, we compare the highest current class probability output provided by our approach to a threshold $\xi \in [0,1]$. When the detection threshold is exceeded, a label is assigned to the most probable class. First, we do it in a frame-wise manner and compute the ROC curve. Using it, we choose a detection threshold $\xi$ equal to 0.16

Table 1: SHREC'17 dataset.

| Method | 14 ges. | 28 ges. |
|---|---|---|
| Guerry *et al.* [8] | 82.9 | 71.9 |
| Ohn-Bar *et al.* [25] | 83.8 | 76.5 |
| Oreifej *et al.* [26] | 78.5 | 74.0 |
| De Smedt *et al.* [4] | 88.2 | 81.9 |
| Hou *et al.* [11] | 93.6 | **90.7** |
| Chen *et al.* [2] | **94.4** | **90.7** |
| Ours | 94.2 | 90.5 |

Table 2: NVIDIA dataset.

| Method | Features | Accuracy |
|---|---|---|
| Human | | 88.4% |
| HOG$^2$ [25] | HC | 36.3% |
| SNV [35] | HC | 70.7% |
| C3D [31] | DL | 78.8% |
| R3DCNN [19] | DL | 80.3% |
| FOANET [22] | DL | 73.7% |
| Ours | DL | **81.3%** |

as it shows a good trade-off between a high TPR (85%) and a low FPR (17%). The average NTtD across all gesture classes is 0.2158 which means that, in average, a hand gesture can be detected after only 22% of its nucleus. In general, static gestures require the finest portion of the nucleus to be seen before classification (around 10%), while dynamic gesture are classified on average within 25%. Static gestures have longest nucleus phases. Intuitively, NTtD differences between dynamic and static gestures are explained as users letting their hand a long time in front of the camera to express a static gesture but the algorithm can detect it using few frames. Finally, we compute the overall recognition accuracy obtained by our approach for an online hand gesture recognition scenario. We obtained an accuracy of 81.3%.

We compare our approach to several state-of-the-art methods, including handcrafted (HC) [25, 35] and deep learning (DL) approaches [19, 22, 31], as well as human labeling accuracy provided by Molchanov *et al.* [19] (see Table 2). We have excluded newer methods that do not perform a continuous recognition [7]. We note that handcrafted give lower results that deep learning methods. Our approach achieves the best performances, meanwhile it is still below human accuracy (88.4%). Note here that Molchanov *et al.* [19] Narayana*et al.* [22] have obtained respectively 83.8% and 91.3%. However, they use a combination of modalities (RGB and depth). Moreover, Narayana*et al.* do not address the online scenario, and employ the HandSegNet method [37] for prior detection from the pre-segmente sequences. For all these reasons, Table 2 contains only comparable results.

### 4.2.2 Ablation study

**Coarse and fine features.** We first analyze the individual components of our approach, the $RNN_{coarse}$ model and the $RNN_{fine}$ model, and evaluate its usefulness according to the type of gestures and then to asses the benefits coming from their fusion. On SHREC'17 dataset [8], for 14 gestures, it achieves accuracies of 84.5% and 80% respectively using the $RNN_{fine}$ and the $RNN_{coarse}$. A joint fine tuning fusion of the two models provides an overall accuracy of 94.2%. For 28 gestures, it achieves accuracies of 76.3% and 76.7% respectively using $RNN_{fine}$ and $RNN_{coarse}$, and an overall accuracy of 90.5% for the fusion. This result illustrates the outstanding potential of fusing shape and posture features to perform fine hand gesture recognition. We note that $RNN_{fine}$ alone does not outperform some previous handcrafted approach proposed in [4]. Only by adding the coarse feature cue, our approach can outperform this method by 6%. The results obtained by our approach are comparable to the best performance with a slight difference.

Table 3: Formulas of the number of parameters for different layers with a number of hidden parameters equal to $n$ and an input of size $m$.

| Layers | Formulas | Ex. (m=64, n=9) |
|---|---|---|
| FC Layer | $m \times n$ | 576 |
| Recurrent layer | $m \times n + n^2$ | 657 |
| LSTM | $4 \times (m \times n + n^2)$ | 2628 |
| CNN | $m \times n$ | 576 |

**Focus on the number of parameters in networks.** The capacity of a neural network model can be define following its size and its depth. Higher are the size and the depth, higher is the number of parameters. Differences in the computational complexity between models are not exactly linearly comparable to their number of parameters, as some layers can see their computational time decreases dramatically using parallel computing. However, it is a good start to study the overall complexity differences between models. Formulas giving the number of parameters of different layers are shown in Table 3.

The R3DCNN [19] contains 79,116,288 parameters distributed as follows: they extract spatiotemporal features using a 3D CNN of 8 convolutional steps and two fully connected layers of size 4096 which together contains 77,885,776 parameters. They append a recurrent layer of size 256 (1,114,112 parameters) and a softmax layer (6,400 parameters).

Our approach extracts hand shape and joint features from a single light 2D CNN with 3 convolutional layers and two fully connected of size 1024 which together contains 3,182,414 parameters. Our method uses also two-stacked LSTM layers, both containing 14,680,064 parameters and ends on a single *softmax* layer of 12,800 parameters. The whole pipeline of our approach contains 32,555,342 parameters, so, less than half the number contained in the R3DCNN [19] network and still outperforms their accuracy result by 1%. The transfer strategy using a hand pose estimator to extract hand shape and joint features allowed us to perform better while using a far less complex network

**Limitations and discussion.** Experiments show that the proposed solution guarantee an effective recognition, but still not exceed the human performance, and gestures which contain high hand shape similarities still showed confusions. Some of these confusion due to the fact that different phases of inverse gestures may contain high similarities. For example, as depicted in Figure 2, the pre-stroke phase of a *Swipe left* consists in moving the hand to the right so that the camera is able to see the entire gesture. However, this movement to the right can be seen as a *Swipe right* nucleus by the algorithm and not as a pre-stroke phase of a *Swipe left*.

Evaluation results for the online detection and recognition show that we can detect an arising gesture after only 21% of its nucleus. However, some missclassification appear during the first few frames of gestures where the algorithm has been able to detect a gesture in progress but does not yet have sufficient information to correctly recognize its type. Figure 3 (a) illustrates the output of our approach on a test sequence of gestures. In this case, the result is almost perfect, each of the 10 gestures is correctly labeled after only few frames. In contrast, Figure 3 (b) shows a test sequence where 5 out of 10 gestures have a misclassification during the first few frames. The issues resulting from those misclassifications could be overcome by firing an incoming gesture only if its length is longer than a threshold.

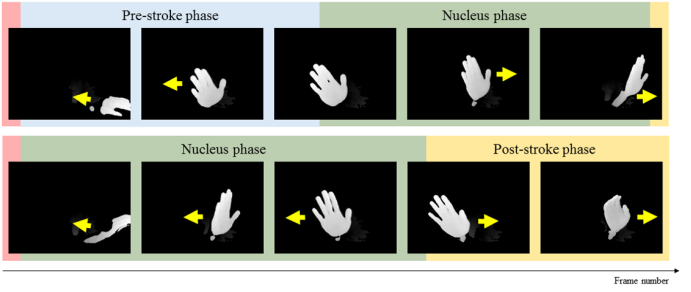
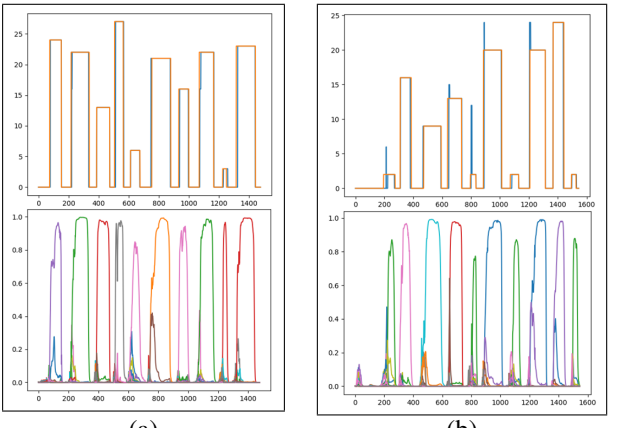



Figure 2: An example of a gesture *Swipe Left* (up) and *Swipe Right* (down), both hand open, and their respective phases (blue: pre-stroke, green: nucleus, orange: post-stroke).



(a) (b)

Figure 3: Gesture detection performance of our approach on a continuous video of 10 gestures. Obtained results (blue) versus the ground-truth (orange) where the x-axis is the time in number of frames and the y-axis represents the class outputs.

# 5 Conclusion

We proposed an online recognition system capable to detect the presence of a gesture in an unsegmented video stream and to recognize it before its end, which is an essential capacity for real-world applications. In our approach, we have taken over the whole pipeline of the recognition process, from hand pose estimation to the classification step, and used the power of deep learning models to increase the efficiency and the robustness of our system. We also presented in this work a new challenging dataset recorded using a depth camera in continuous scenario.

Experimental results demonstrated that the proposed approach is capable to recognize hand gestures and to improve state-of-the-art results. In addition, the experiments showed that our framework is able to detect an occurring gesture after only 21% and 22% of the nucleus phase, respectively for the Online DHG and NVIDIA datasets. The use of the transfer learning strategy allowed us to outperform state-of-the-art deep learning approaches using less than half of the number of parameters of the baseline model.

# References


[1] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[2] Yuxiao Chen, Long Zhao, Xi Peng, Jianbo Yuan, and Dimitris N. Metaxas. Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. *CoRR*, abs/1907.08871, 2019.

[3] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat. 3d hand gesture recognition using a depth and skeletal dataset: Shrec'17 track. In *Proceedings of the Workshop on 3D Object Retrieval*, 3Dor '17, page 33–38, Goslar, DEU, 2017. Eurographics Association. doi: 10.2312/3dor.20171049.

[4] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016.

[5] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Heterogeneous hand gesture recognition using 3d dynamic skeletal data. *Computer Vision and Image Understanding*, (In Press, Corrected Proof, Available online 14 February 2019), 2019. doi: 10.1016/j.cviu.2019.01.008.

[6] Maxime Devanne, Hazem Wannous, Mohamed Daoudi, Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Learning Shape Variations of Motion Trajectories for Gait Analysis. In *International Conference on Pattern Recognition (ICPR 2016)*, pages 895 – 900, Cancun, Mexico, December 2016. doi: 10.1109/ICPR.2016.7899749.

[7] Andrea D'Eusanio, Alessandro Simoni, Stefano Pini, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. A transformer-based network for dynamic hand gesture recognition. In *2020 International Conference on 3D Vision (3DV)*, pages 623–632, 2020. doi: 10.1109/3DV50981.2020.00072.

[8] Sergio Escalera, Xavier Baró, Jordi Gonzalez, Miguel Ángel Bautista, Meysam Madadi, Miguel Reyes, Víctor Ponce-López, Hugo Jair Escalante, Jamie Shotton, and Isabelle Guyon. Chalearn looking at people challenge 2014: Dataset and results. In *ECCV Workshops (1)*, pages 459–473, 2014.

[9] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[10] Minh Hoai and Fernando De la Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014.

[11] Jingxuan Hou, Guijin Wang, Xinghao Chen, Jing-Hao Xue, Rui Zhu, and Huazhong Yang. *Spatial-Temporal Attention Res-TCN for Skeleton-Based Dynamic Hand Gesture Recognition: Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, pages 273–286. 01 2019. ISBN 978-3-030-11023-9. doi: 10.1007/978-3-030-11024-6_18.

[12] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[13] Alexey Kurakin, Zhengyou Zhang, and Zicheng Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1975–1979. IEEE, 2012.

[14] Alina Kuznetsova, Laura Leal-Taixé, and Bodo Rosenhahn. Real-time sign language recognition using a consumer depth camera. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 83–90, 2013.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] Shao-Zi Li, Bin Yu, Wei Wu, Song-Zhi Su, and Rong-Rong Ji. Feature learning based on sae–pca network for human gesture recognition in rgbd images. *Neurocomputing*, 151:565–573, 2015.

[17] Hsien-I Lin, Ming-Hsiang Hsu, and Wei-Kai Chen. Human hand gesture recognition using a convolution neural network. In *IEEE International Conference on Automation Science and Engineering*, pages 1038–1043. IEEE, 2014.

[18] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. Hand gesture recognition with 3d convolutional neural networks. In *IEEE conference on computer vision and pattern recognition workshops*, pages 1–7, 2015.

[19] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016.

[20] Camille Monnier, Stan German, and Andrey Ost. A multi-scale boosted detector for efficient and robust gesture recognition. In *ECCV Workshops (1)*, pages 491–502, 2014.

[21] Jawad Nagi, Frederick Ducatelle, Gianni A Di Caro, Dan Cireşan, Ueli Meier, Alessandro Giusti, Farrukh Nagi, Jürgen Schmidhuber, and Luca Maria Gambardella. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pages 342–347. IEEE, 2011.

[22] Pradyumna Narayana, J. Ross Beveridge, and Bruce A. Draper. Gesture recognition: Focus on the hands. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5235–5244, 2018.

[23] Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. Moddrop: adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38 (8):1692–1706, 2016.

[24] Juan C. Nez, Ral Cabido, Juan J. Pantrigo, Antonio S. Montemayor, and Jos F. Vlez. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recogn.*, 76(C):80–94, April 2018. ISSN 0031-3203. doi: 10.1016/j.patcog.2017.10.033.

[25] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014.

[26] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.

[27] Nicolas Pugeault and Richard Bowden. Spelling it out: Real-time asl fingerspelling recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1114–1119. IEEE, 2011.

[28] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE transactions on multimedia*, 15(5):1110–1120, 2013.

[29] R. Slama, H. Wannous, and M. Daoudi. 3D human motion analysis framework for shape similarity and retrieval. *Image and Vision Computing*, 32(2):131 – 154, 2014. ISSN 0262-8856. doi: 10.1016/j.imavis.2013.12.011.

[30] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014.

[31] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[32] Chong Wang, Zhong Liu, and Shing-Chow Chan. Superpixel-based hand gesture recognition with kinect depth camera. *IEEE transactions on multimedia*, 17(1):29–39, 2015.

[33] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[34] Ying Wu, John Y Lin, and Thomas S Huang. Capturing natural hand articulation. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 426–432. IEEE, 2001.

[35] Xiaodong Yang and YingLi Tian. Super normal vector for activity recognition using depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 804–811, 2014.

[36] Chenyang Zhang, Xiaodong Yang, and YingLi Tian. Histogram of 3d facets: A characteristic descriptor for hand gesture recognition. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE, 2013.

[37] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.