Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition

Di Wu, Lionel Pigou, Pieter-Jan Kindermans, Nam Le, Ling Shao, Joni Dambre, and Jean-Marc Odobez

Abstract

This paper describes a novel method called deep dynamic neural networks *(DDNN)* for multimodal gesture recognition. More precisely, a semi-supervised hierarchical dynamic framework based on a Hidden Markov Model (HMM) is proposed for simultaneous gesture segmentation and recognition where skeleton joint information, depth and RGB images are the multimodal input observations. Unlike most traditional approaches which rely on the construction of complex handcrafted features as HMM input features, our approach learns high-level spatio-temporal representations using deep neural networks suited to the input modality: a Gaussian-Bernouilli deep belief networks (*DBN*) to handle skeletal dynamics, and a 3D convolutional neural networks (*3DCNN*) to manage and fuse batches of depth and RGB images. This achieved through the modeling and learning of the emission probabilities of the HMM required to infer the gesture sequence. This purely data driven approach achieves a score of ***0.81*** in the ChaLearn LAP gesture spotting challenge. The performance is on par with a variety of the state-of-the-art hand-tuned feature based approaches and other learning based methods. Thus opening the door for using deep learning techniques to further explore multimodal time series.

Index Terms

Deep learning, convolutional neural networks, deep belief networks, hidden Markov models, gesture recognition.

I. Introduction

In recent years, human action recognition has drawn increasing attention of researchers, primarily due to its potential in areas such as video surveillance, robotics, human-computer interaction, user interface design, and multimedia video retrieval.

Previous works on video-based motion recognition [**?**], [**?**], [**?**] mainly focused on adapting handcrafted features. These methods usually have two stages: an optional feature detection stage followed by a feature description stage. Well-known feature detection methods ("interest point detectors") are Harris3D [**?**], Cuboids [**?**] and Hessian3D [**?**]. For descriptors, popular methods are Cuboids [**?**], HOG/HOF [**?**], HOG3D [**?**] and Extended SURF [**?**]. In recent work of Wang *et al.* [**?**], dense trajectories with improved motion based descriptors epitomised the pinnacle of handcrafted features and achieved state-of-the-art results on a variety of "in the wild" datasets. Based on the current trends, challenges and interests within the action recognition community, it is to be expected that many successes will follow. However, the very high-dimensional and dense trajectory features usually require the use of advanced dimensionality reduction methods to make them computationally feasible.

Furthermore, as discussed in the evaluation paper of Wang *et al.* [**?**], no universally best hand-engineered feature exists and the best performing feature descriptor is often dataset dependent. This clearly indicates that the ability to learn dataset specific feature extractors can be highly beneficial. For this reason, even though handcrafted features have dominated image recognition in previous years, there has been a growing interest in learning low-level and mid-level features, either in supervised, unsupervised, or semi-supervised settings [**?**], [**?**], [**?**].

Since the recent resurgence of neural networks invoked by Hinton and others [**?**], deep neural architectures serve as an effective solution for extracting high-level features from data. Deep artificial neural networks have won numerous contests in pattern recognition and representation learning. Schmidhuber [**?**] compiled a historical survey compactly summarising relevant works with more than 850 entries of credited works. From this overview we see that these models have been successfully applied to a plethora of different domains: the GPU-based cuda-convnet [**?**] classifies 1.2 million high-resolution images into 1000 different classes; multi-column deep neural networks [**?**] achieve near-human performance on the handwritten digits and traffic signs recognition benchmarks; 3D convolutional neural networks [**?**] [**?**] recognise human actions in surveillance videos; deep belief networks combined with hidden Markov models [**?**] [**?**] for acoustic and skeletal joints modelling outperform the decade-dominating paradigm of Gaussian mixture models in conjunction with hidden Markov models. Multimodal deep learning technique were also investigated [**?**] to learn cross-modality representation, for instance in the context of audio-visual speech recognition. And recently, Baidu research proposed a DeepSpeech system [**?**] that combines a well-optimised recurrent neural network (RNN) training system, achieving the best error rate on a noisy speech dataset. In these fields, deep architectures have shown great capacity to discover and extract higher level relevant features.

However, direct and unconstrained learning of complex problems remains difficult, since (i) the amount

of required training data increases steeply with the complexity of the prediction model and (ii) training highly complex models with very general learning algorithms is extremely difficult. It is therefore a common practice to restrain the complexity of the model. This is generally done by operating on small patches to reduce the input dimension and diversity [**?**], or by training the model in an unsupervised manner [**?**], or by forcing the model parameters to be identical for different input locations (as in convolutional neural networks [**?**], [**?**], [**?**]).

On the sensor side, due to the immense popularity of Microsoft Kinect [**?**] [**?**], there has been a recent interest in developing methods for human gesture and action recognition from 3D skeletal data and depth images. A number of new datasets [**?**], [**?**], [**?**], [**?**] have provided researchers with the opportunity to design novel representations and algorithms, and test them on a much larger number of sequences. While gesture recognition based on 3D joint positions may seem trivial, it is actually not the case due to several factors. A first one is the high dimensionality and the large amount of variability of the pose space itself. A second aspect that further complicates the recognition is the segmentation of the different gestures. While in practice segmentation is as important as the recognition, it is an often neglected aspect of the current action recognition research which often assume the availability of segmented inputs [**?**] [**?**] [**?**].

In this paper we aim to address these issues by proposing a data driven system, focusing on analysis of acyclic video sequence labelling problems, *i.e.* video sequences that are non-repetitive as opposed to longer repetitive activities, *e.g.* jogging, walking and running. By integrating deep neural networks within an HMM temporal framework, our work allows the online joint recognition and segmentation of gestures. This framework is inspired by discriminant HMM, which embedded multi-layer perceptron inside HMM, in continuous speech recognition [**?**] [**?**] This paper is an extension of the works of [**?**], [**?**] and [**?**]. The key contributions can be summarised as follows:

- A Gaussian-Bernoulli Deep Belief Network is proposed to extract high-level skeletal joint features and the learned representation is used to estimate the emission probability needed to infer gesture sequences;
- A 3D Convolutional Neural Network is proposed to extract features from 2D multiple channel inputs like depth and RGB images stacked along the 1D temporal domain;
- Intermediate and late fusion strategies are investigated within the temporal modelling. The result of both mechanisms show that multiple-channel fusions outperform individual modules.

The remainder of this paper is organised as follows. Section II reviews related works for gesture recognition with various temporal models and recent deep learning work on RGB-D data. Section III introduces the formulation of our Deep Dynamic Neural Network model and the intuition behind the high

level feature extraction. Section IV details the model implementation. Section V details the experimental analysis and Section VI concludes the paper with discussions related to future works.

II. Related Work

Gesture recognition has drawn increasing attention from researchers, primarily due to its growing potential in areas such as robotics, human-computer interaction and user interface design. Different temporal models have been proposed. Nowozin and Shotton [**?**] proposed the notion of "action points" to serve as natural temporal anchors of simple human actions using a Hidden Markov Model. Wang *et al.* [**?**] introduced a more elaborated discriminative hidden-state approach for the recognition of human gestures. However, relying on only one layer of hidden states, their model alone might not be powerful enough to learn a higher level representation of the data and take advantage of very large corpus. In this paper, we adopt a different approach by focusing on deep feature learning within a temporal model.

There have been a few works exploring deep learning for action recognition in videos. For instance, Ji *et al.* [**?**] proposed using 3D convolutional neural network for automated recognition of human actions in surveillance videos. Their model extracts features from both the spatial and the temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. To further boost the performance, they proposed regularising the outputs with high-level features and combining the predictions of a variety of different models. Taylor *et al.* [**?**] also explored 3D convolutional networks for learning spatio-temporal features for videos. The experiments in [**?**] show that multiple network averaging works better than a single individual network and larger nets will generally perform better than smaller nets. Providing there is enough data, averaging multi-column nets [**?**] applied to action recognition could also further improve the performance.

However, the advent of Kinect-like sensors has put more emphasis on RGB-D data for gesture recognition, but not only. For instance, the benefits of deep learning using RGB-D data have been explored for object detection or classification tasks. Dosovistskiy *et al.* [**?**] presented generic feature learning for training a convolutional network using only unlabeled data. In contrast to supervised network training, the resulting feature representation is not class specific and is advantageous on geometric matching problems, outperforming the SIFT descriptor. Socher *et al.* [**?**] proposed a single convolutional neural net layer for each modality as inputs to multiple, fixed-tree RNNs in order to compose higher order features for 3D object classification. The single convolutional neural net layer provides useful translational invariance of low level features such as edges and allows parts of an object to be deformable to some extent. To address object detection, Gupta *et al.* [**?**] proposed a geocentric embedding for depth images that encodes height above ground and angle with gravity for each pixel in addition to the horizontal disparity. This

augmented representation allows CNN to learn stronger features than when using disparity (or depth) alone.

Recently, the gesture recognition domain itself has been stimulated by the collection of large public corpus. In particular, the ChaLearn LAP [**?**] gesture spotting challenge has collected around 14,000 gestures drawn from a vocabulary of 20 Italian sign gesture categories. The emphasis is on multi-modal automatic learning gestures performed by several different users, with the aim of performing user independent continuous gesture spotting. Some of the top winning methods in the ChaLearn LAP gesture spotting challenge require a set of complicated handcrafted features for either skeletal input, RGB-D input, or both. For instance, Neveroa *et al.* [**?**] proposed a pose descriptor consisting of 7 logical subsets for skeleton features while Monnier *et al.* [**?**] proposed to use 4 types of features for skeleton features (normalised joint positions; joint quaternion angles; Euclidean distances between specific joints; and directed distances between pairs of joints, based on the features proposed by Yao *et al.* [**?**]) and histograms of oriented gradients (HOG) descriptor for RGB-D images around hand regions. In [**?**], the state-of-the-art dense trajectory [**?**] handcrafted features are adopted for the RGB module.

There is a gradual trend to learn the features for gesture recognition in videos. For instance, the recent methods in [**?**], [**?**] focused on single modalities, used deep networks to learn representations from skeleton data [**?**] or from RGB-D data [**?**]. Neveroa *et al.* [**?**] presents a multi-scale and multimodal deep network for gesture detection and localisation. Key to their technique is a training strategy that exploits i) careful initialisation of individual modalities and ii) gradual fusion of modalities from the strongest to weakest cross-modality structure. One major difference compared to what we propose is the treatment of the time factor: rather than using a temporal model, they used frames within a fixed interval as the input of their neural networks for the prediction of the final gesture class, an approach that requires require to train several multi-scale temporal networks to cope with gestures performed at different speeds. Furthermore, their skeleton features used in their network are sets of ad-hoc hand crafted features, rather than being learned from raw data.

III. Model Formulation & overall approach

Inspired by the framework successfully applied to speech recognition [**?**], the proposed model is a data driven learning system. This results in an integrated model, where the amount of prior knowledge and engineering is minimised. On top of that, this approach works without the need for additional complicated preprocessing and dimensionality reduction methods as it is naturally embedded in the framework.

The proposed approach relies on a Hidden Markov Model (HMM) for the temporal part, and neural networks to model the emission probabilities. In the remainder of this section, we will first present our

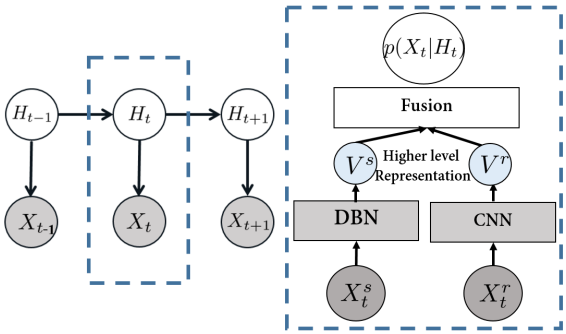


Fig. 1: Gesture recognition model: the temporal model is a HMM (left), whose emission probability $p(X_t|H_t)$ (right) is modeled by forward-linked neural networks. More precisely, observations X_t (skeletal features X_t^s, or RGB-D image features X_t^r) are first passed through appropriate deep neural nets (a Deep Belief Network -*DBN*- pretrained with Gaussian-Bernouilli Restricted Boltzmann Machines for the skeleton modality, and a 3D convolutional neural network -*3DCNN*- for the RGB-D modality) to implictly extract high-level features (V^s and V^r) which are further fused to produce an estimate of $p(X_t|H_t)$.

temporal model and then introduce its main component. The details of the two distinct neural networks and fusion mechanisms along with post-processing will be provided in Section IV.

A. Deep Dynamic Neural Networks

The proposed deep dynamic neural network *(DDNN)* can be seen as an extension of [**?**], where instead of only using the restricted Boltzmann machines to model human motion, various connectivity layers (fully connected layers, convolutional layers) are stacked together to learn higher level features justified by a variational bound [**?**] from different input modules.

A continuous-observation HMM is adopted for modelling higher level temporal relationships. At each time step t, we have one observed random variable X_t composed of the skeleton input X_t^s and RGB-D input images X_t^r as shown in the graphical representation in Fig. 1. The hidden state variable H_t takes on values in a finite set $\mathcal{H}$ composed of $N_{\mathcal{H}}$ states related to the different gestures. The intuition motivating the HMM model is that a gesture is composed of a sequence of poses where the relative duration of each pose may vary. This variance is captured by allowing flexible forward transitions within a Markov chain. In practice, H_t can be interpreted as being in a particular phase of a gesture $\boldsymbol{a}$.

Classically under the HMM assumption, the joint probability of observations and states is given by:

$$p(H_{1:T}, X_{1:T}) = p(H_1)p(X_1|H_1)\prod_{t=2}^{T} p(X_t|H_t)p(H_t|H_{t-1}), \tag{1}$$

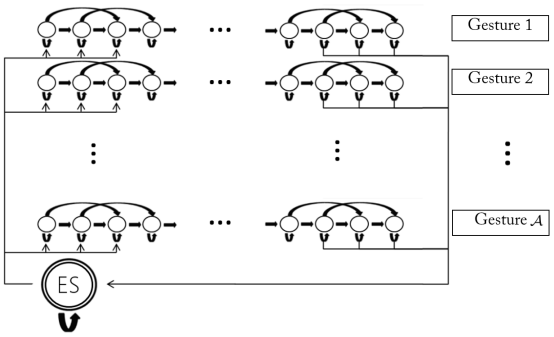


Fig. 2: State diagram of the *ES-HMM* model for low-latency gesture segmentation and recognition. An ergodic state ($\mathcal{ES}$) is used to model the resting position between gesture sequences. Each node represents a single state and each row represents a single gesture model. The arrows indicate possible transitions between states.

where $p(H_1)$ is the prior on the first hidden state, $p(H_t|H_{t-1})$ is the transition dynamics modelling the allowed state transitions and their probabilities, and $p(X_t|H_t)$ is the emission probability of the observation, modelled by deep neural networks in our case. These elements are presented below.

B. State-transition model and inference

The HMM framework can be used for simultaneous gesture segmentation and recognition. This is achieved by defining the state transition diagram as shown in Fig 2. For each given gesture $a \in \mathcal{A}$, a set of states $\mathcal{H}_a$ is introduced to defined a Markov model of that gesture. For example, for action sequence "tennis serving", the action sequence can implictly be dissected into $h_{a_1}, h_{a_2}, h_{a_3}$ as: 1) raising one arm 2) raising the racket 3) hitting the ball. More precisely, since our goal is to capture the variation in speed of the performed gestures, we set the transition matrix $p(H_t|H_{t-1})$ in the following way: when being in a particular node n at time t, moving to time $t+1$, we can either stay in the same node (slower), move to node $n+1$, or move to node $n+2$ (faster). Furthermore, to allow the segmentation of gestures, we add an ergodic state ($\mathcal{ES}$) which resembles the silence state for speech recognition and which serve as a catch-all state. From this state we can move to the first three nodes of any gesture class, and from the last three nodes of any gesture class we can move to $\mathcal{ES}$. Hence, the hidden variable H_t can take values within the finite set $\mathcal{H} = (\bigcup_{a\in\mathcal{A}} \mathcal{H}_a) \bigcup \{\mathcal{ES}\}$.

Overall, we refer to the model as the ergodic states hidden Markov model (*ES-HMM*) for simultaneous gesture segmentation and recognition. It differs from the firing hidden Markov model of [**?**] in that we strictly follow a left-right HMM structure without allowing backward transition, forbidding inter-states

transverse, assuming that the considered gestures do not undergo cyclic repetitions as in walking for instance.

Once we have the trained model, we can use standard techniques to infer online the filtering distribution $p(H_t|X_{1:t})$, or offline (or with delay) the smoothed distribution $p(H_t|X_{1:T})$ where T denotes the end of the sequence. Because the graph for the hidden Markov model is a directed tree, this problem can be solved exactly and efficiently using the max-sum algorithm also known as Viterbi algorithm. This algorithm searches the space of paths efficiently to find the most probable path with a computational cost that grows only linearly with the length of the chain [**?**]. The result of the Viterbi algorithm is a path–sequence $\hat{h}_{t:T}$ of nodes going through the state diagram of Fig.2 and from which we can easily infer the class of the gesture as illustrated in Fig. 8.

C. Learning the emission probability

Traditionally, emission probabilities for activity recognition have been trained with Gaussian Mixture Models (GMM), one per state. Alternatively, in this work we propose to model this term in a discriminative fashion. More precisely, since the input features have a high dimensionality, we propose to learn them using two distinctive types of neural networks each suited to the input modality, as summarized in the right of Fig. 1.

Unfortunately, estimating a probability density such as an emission probability remains quite a difficult problem, especially in high dimensions. Theoretically, discriminative neural networks estimate posterior probabilities $p(H_t|X_t)$. We should divide posteriors by priors to get the scaled likelihoods which are required by the HMM for decoding. However, using scaled likelihood may not be beneficial if estimated priors do not match the priors in the test set [**?**]. Therefore, we employ the posteriors directly without dividing by the priors. This is equivalent to assuming that all priors are equal. or, in other words, inference in the HMM only depends only on the ratio between emission probabilities for the different states. One can interpret that the models are trained to directly predict the ratio between emission probabilities. This is similar to the approach used by Kindermans et al. to integrate transfer learning and an HMM based language model into a single probabilistic model [**?**]. One should think of the predicted emission probability ratio as an unnormalized version of the true emission probability. Nevertheless, to simplify the discussion of our models for readers with a basic understanding of HMMs, we will refer to the predicted emission probability ratio simply as emission probabilities since the underlying model remains unchanged.

For the skeletal features, we rely on a Deep-Belief Network (DBN) trained in two steps [**?**]: in the first step, stacked Restricted Boltzmann Machines (RBM) are trained in an unsupervised fashion using

only observation data to learn high-level feature representations; in the second step, the model is used as a Deep-Belief Network whose weights are further fine-tuned for learning the emission probability. For the RGB and depth (RGB-D) video data, we rely on a 3D (2D for space and 1D for time) convolutional neural networks (3DCNN) to model the emission probabilities. Finally, a fusion method combines in an intermediate or in a late stage the contributions of both modalities. In all cases (including the fusion), the supervised training is conducted by learning to predict the state label (an element of $\mathcal{H}$) associated to each training or testing frame.

Such an approach present several advantages over the traditional GMM paradigm. First, while GMMs are easy to fit when they have diagonal covariance matrices and, with enough components, can model any distribution, they have been shown to statistically inefficient at modeling high-dimensional features with many componential structure as explained in [**?**]. For instance, assume that the components of the input feature space can be separated into two subspaces characterized by N and M significantly different patterns in the training data, respectively, and that the occurences of these patterns are relatively independent[1]. A GMM will requires $N * M$ components to model this structure because each component must generate all the input features. On the other hand, a stacked RBMs model that can explains the data using multiple causes only requires $N + M$ components (in the ideal fully independent case), each of which is specific to a particular subspace. This exponential inefficiency of GMMs at modeling factorial structures leads to GMM+HMM systems having a very large number of Gaussians, most of which must be estimated from a very small fraction of the data.

Secondly, the approach for training the skeleton DBN model, first using variational learning to train stacked RBMs with unlabeled data, then in a discriminative fashion [**?**] has been shown to have several advantages. It has been observed that variational learning [**?**], which tries to optimize the data-likelihood while minizing the Kullback-Leibler divergence between the true posterior distribution of the hidden state (i.e. hidden layer variables of the RBMs in our case) and an approximation of this distribution, tends to produce unimodal distributions. This is beneficial, as this means that similar sensory inputs will be mapped to similar hidden variables. Thus, the intuition for using DBN for modeling the emission probability $p(X_t|H_t)$ from skeleton joints is that by learning the multi-layer network layer by layer, semantically meaningful high level features for skeleton configuration will be extracted while at the same time a parametric prior of human pose is learned. In our case, using the pairwise joints features as raw input, the data-driven approach network will be able to extract relational multi-joint features which are relevant to the target classes. For instance, from the “toss” action data, a wrist joints rotating

[1]In our case, intuitively these spaces could be the features from different body parts, like left/right arm or torso features.

around shoulder joints feature is expected to be extracted from the backpropagation learning, and be the equivalent of those task specific *ad hoc* hard wired sets of joint configurations defined in [**?**] [**?**] [**?**] [**?**].

The benefit of such a learning approach is even more important when large amount of unlabelled data (e.g. skeleton data inferred from depth images of people performing unknown gestures) is available in addition to the labeled ones (this was not the case in this paper). Naturally, many of the features learned in this unsupervised way might be irrelevant for making the required discriminations, even though they are important for explaining the input data. However, this is a price worth paying if data availability and computation are cheap and lead to a stable mapping of the high-dimensional input into high-level features that are very good for discriminating between classes of interest. In this view, it is important to notice that each weight in a neural network is usually constrained by a larger fraction of the training samples than each parameter in a GMM, a point that has been masked by other differences in training. In particular, neural networks have traditionally been training discriminatively, whereas GMMs are typically trained as generative models, which given their parametric nature partially compensates the fact that each mixture of a large GMM is usually trained on a very small fraction of the data.

In summary, the feed forward neural networks offer several potential advantages over GMMs:

- their estimation of emission probabilities does not require detailed assumptions about the data distribution;
- they allow an easy combination of diverse features, including both discrete and continuous features;
- they use far more of the data to constrain each parameter because the output on each training case is sensitive to a large fraction of the weights.

IV. Model Implementation

In this section, we detail the different component of the proposed Deep Dynamic Neural Network approach.

A. Ergodic States Hidden Markov Model

In all our experiments, the different modelling elements are specified as follows.

The number of states $N_{\mathcal{H}_a}$ associated to an individual gesture has been set to 5. Therefore, in total, the number of states is $N_{\mathcal{H}} = 20 \times 5 + 1 = 101$ when conducting experiments on the Chalearn dataset containing 20 classes. Note that intuitively, 5 states represents a good granularity as most gestures in the Clalearn are composed of 5 phases: an onset, followed by arm motions to reach a more static pose (often

Fig. 3: Left: A point cloud projection of a depth image and the 3D positional features. Right: A *DBN* is trained to predict the emission probability $p(X_t^s|H_t)$ from the skeleton input $\mathbf{f}_t$. The double arrows indicate that the intermediate weights are first trained in an unsupervised fashion using stacked RBMs.

characterized by a distinct hand posture), and the motion back to the rest place. In the future, optimal section of this number[2] and of different number of states per gesture could be investigated.

The training data in Chalearn is given as a set of sequences $\mathbf{x}_i = [x_{i,1}, \ldots, x_{i,t}, \ldots, x_{i,T_i}]$ where $x_{i,t} = [x_{i,t}^s, x_{i,t}^r]$ corresponds to the skeleton and RGB-D input. As only a single gesture label is provided for each sequence, we need to define $\mathbf{y}_i = [y_{i,1}, \ldots, y_{i,t}, \ldots, y_{i,T_i}]$, the sequence of state labels $y_{i,t}$ associated to each frame. To do so, a forced alignment is used which means that if the i^{th} sequence is a gesture $\boldsymbol{a}$, then the first $\lfloor \frac{T_i}{5} \rfloor$ frames are assigned to state $h_{\boldsymbol{a}}^1$ (the first state of gesture $\boldsymbol{a}$), the following $\lfloor \frac{T_i}{5} \rfloor$ frames are assigned to $h_{\boldsymbol{a}}^2$, and so forth.

Note that each gesture sequence comes with the video frames preceeding and following the gesture. In practice, we extracted 5 frames before and after each gesture sequence and labelled them with the ergodic state ($\mathcal{ES}$) label. The transitional matrix $p(H_t|H_{t-1})$ was learned by simply collecting the transition statistics from the label sequences $\mathbf{y}_i$, allowing 5 frame jumps to accommodate skipping states.

B. Skeleton Module

1) Skeleton input features: Given our task, only the $N_j = 11$ upper body joints are relevant and considered, namely *"ElbowLeft, WristLeft, ShoulderLeft, HandLeft, ElbowRight, WristRight, ShoulderRight, HandRight, Head, Spine, HipCenter"*. The raw skeleton features of time t are defined as $x_t^s = [x_t^{s,1}, \ldots, x_t^{s,N_j}]$. To capture the gesture dynamics, rather than using x_t^s as raw input to our data driven

[2]Experiments with 10 states led to similar performance.

approach, we follow the approach of [**?**] and compute the 3D positional pairwise differences of joints as well as temporal derivatives, defined as (as shown in Fig. 3) [3]:

$$f_t^{cc} = \{x_t^{s,i} - x_t^{s,j} | i, j = 1, 2, \ldots, N_j; i \neq j\} \tag{2}$$

$$f_t^{cp} = \{x_{t+1}^{s,i} - x_t^{s,i} | i = 1, 2, \ldots, N_j\} \tag{3}$$

$$f_t^{ca} = \{x_{t+1}^{s,i} - 2 \times x_t^{s,i} + x_{t-1}^{s,i} | i = 1, 2, \ldots, N_j\} \tag{4}$$

This results in an input feature vector $\mathbf{f}_t = [f_t^{cc}, f_t^{cp}, f_t^{ca}]$ of dimension $N_{\mathbf{f}} = N_j \times (\frac{N_j}{2} + N_j + N_j) * 3 = 891$. Admittedly, here we do not completely neglect human prior knowledge about information extraction for relevant static postures, velocity and acceleration of overall dynamics of motion data. While we have indeed used prior knowledge to define our relevant features, we believe they remain quite general and do not need dataset specific tuning. Note that the feature extraction process resembles the computation of the *Mel Frequency Cepstral Coefficients (MFCCs)* and their temporal derivatives typically used in the speech recognition community [**?**].

2) Modeling X_t^s using Deep Belief Networks: Given the input skeleton feature $\mathbf{f}$, a *DBN* model is used to predict the emission probability, as shown in Fig. 3. As explained in Section III-C, the learning proceeds in two steps: in the first one, the network is considered as a stack of RBMs, and trained using a greedy, layer-by-layer unsupervised learning algorithm [**?**]; in the second one, a softmax network layer is added on top of the RBMs to create a *DBN* architecture, where the weights of the first step are used to initialize the corresponding weights in the *DBN*, and the *DBN* is further trained and fine-tuned in a supervised fashion to predict the emission probability. The number of nodes at each layer of the *DBN* are $[N_{\mathbf{f}}, 2000, 2000, 1000, N_{\mathcal{H}}]$. We provide below further details on the model and training.

Gaussian-Bernoulli RBM. Restriced Boltzmann machine (RBM) are undirected graphical models involving visible and hidden variables, with symmetric connections between the hidden and visible units but no connection within hidden variables or visible variables. In most cases, RBMs rely on binary random variables. However, in our case the visible unit in the first layer are the vector of skeleton features $\mathbf{f} \in \mathbf{R}^{N_{\mathbf{f}}}$, which are continous. To account for this situation, we thus resort to a Gaussian-Bernoulli RBM (*GRBM*) [**?**], whose main difference w.r.t. standard RBM lies in the following: the energy term of our first layer $\mathbf{f}$ to the hidden binary stochastic units $\mathbf{h} \in \{0, 1\}^F$ is given by:

$$E(\mathbf{f}, \mathbf{h}; \theta) = -\sum_i \frac{(f_i - b_i)^2}{2\sigma_i^2} - \sum_i \sum_j W_{ij} h_j \frac{f_i}{\sigma_i} - \sum_{j=1} a_j h_j \tag{5}$$

[3]Note that the offset features used in [**?**] depend on the first frame. Thus if the initialisation fails which is a very common scenario, the feature descriptor will be generally very noisy. Hence, here we do not use these offset features.

where $\theta = \{W, b, a\}$ are the free parameters: $W_{i,j}$ serves as the symmetric synergy term between visible unit i and hidden unit j, while b_i and a_j are the bias term of the visible and hidden units, respectively. The conditional distributions needed for inference and generation are given by the traditional logistic function g for the binary hidden variable, and the normal distribution $\mathcal{N}$ for the continous variable:

$$P(h_j = 1|\mathbf{f}) = g(\sum_i W_{ij} f_i + a_j) \quad (6)$$

$$P(f_i = f|\mathbf{h}) = \mathcal{N}(f|\mu_i, \sigma_i^2). \quad (7)$$

where $\mu_i = b_i + \sigma_i^2 \sum_j W_{ij}$. In practice, we normalise the data (mean substraction and standard deviation division) in the preprocessing phase. Hence, instead of learning σ_i^2, one typically used $\sigma_i^2 = 1$ during training.

We ran 100 epochs using a fixed recipe based on stochastic gradient descent with a mini-batch size of 200 training cases to train the stacked RBM, in which the learning rate is fixed at 0.001 for the Gaussian-Bernoulli RBMs, and at 0.01 for the following binary-binary RBMs.

***DBN* forward training.** The *DBN* is initialized with the result of the previous step, a method which tends to avoid suboptimal local minima and increase the networks performance stability. The learning rate for the parameter fine tuning starts at 1 with 0.99999 mini-batch scaling. During the experiments, early stopping occurs around epoch 440. The optimisation completes with a frame based validation error rate of 16.5%.

C. RGB & Depth 3D Module

1) Preprocessing: Although DeepMind technology [**?**] presented the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using deep reinforcement learning, working directly with raw input Kinect recorded data frames, which are 640×480 pixel images, can be computationally demanding. Therefore, our first step in the preprocessing stage consists of cropping the highest hand and the upper body using the given joint information. In the Chalearn dataset, we determined that the highest hand is the most interesting. When both hands are used, they normally perform the same (mirrored) movement, and when one hand is used, it is always the highest one which is relevant. Furthermore, to be invariant to handedness, we always train the model with the right hand view. That is, when the left hand is actually the performing hand, the video was mirrored.

The preprocessing results in four video samples (body and hand with grayscale and depth) of resolution 64×64. Furthermore, the noise in the depth maps is reduced by removing the background using the automatically produced segmentation mask provided with the data, and applying a median filtering. Depth

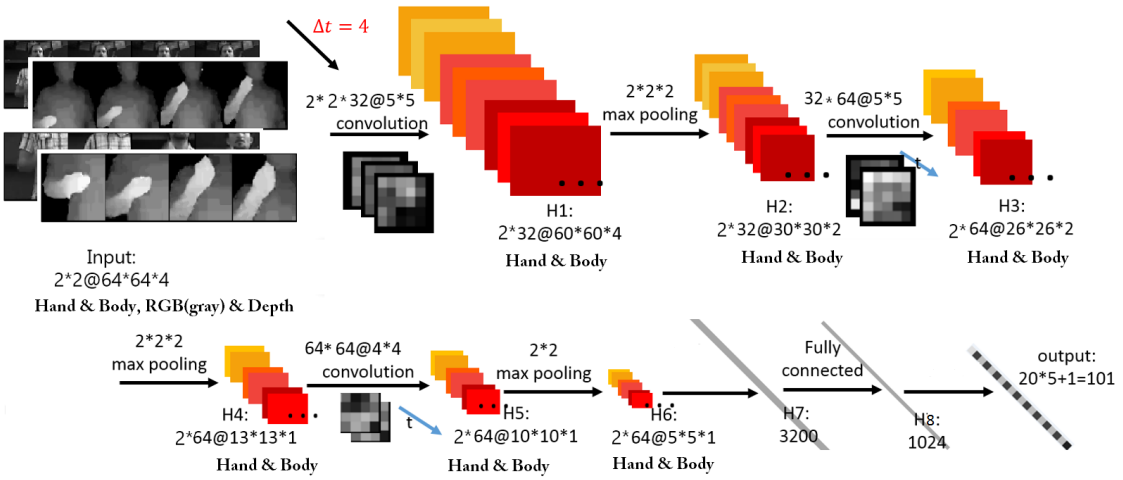


Fig. 4: 3DCNN architecture. The input is $2 \times 2@64 * 64 * 4$, meaning 2 modalities (depth and RGB) for the hand and body regions, each being 4 consecutive 64 by 64 frames stacked together. See text for further details.

images are Z-normalised (the mean is subtracted -as it is rather irrelevant to the gesture subclass- and the result divided by the standard deviation), whereas RGB images are only normalized by the image standard deviation. The outcome is illustrated in Fig. 5.

2) 3DCNN Architecture: This architecture consists of a series of layers composed of either convolution, pooling or, in the last layer, fully connected layers. The 3D convolution itself is achieved by convolving a 3D kernel to the cuboid formed by stacking multiple contiguous frames together. We follow the nomenclature of in [**?**]. However, instead of using $tanh$ units [**?**], Rectified Linear Units (*ReLUs*) [**?**] were used in order to speed up training. Formally, the value of a unit at position (x, y, z) (z here corresponds to the time-axis) in the j-th feature map in the i-th layer, denoted as v_{ij}^{xyz}, is given by:

$$v_{ij}^{xyz} = max(0, (b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(t+r)})) \quad (8)$$

The complete 3DCNN architecture is depicted in Fig. 4: 4 types of input contextual frames are stacked as size $64 \times 64 \times 4$ (as illustrated in Fig. 5). The first layer (H1) consists of 32 feature maps produced by 5×5 spatial convolutional kernels, followed by local contrast normalisation (LCN) [**?**]. Note that the filter response maps of the Depth and RGB images of the hand (and body) are summed to produce a single feature map, thus resulting in H1 in 32 feature maps for each of the hand and for the body region. A 3D max pooling with strides $(2, 2, 2)$ is then applied. The second layer uses 64 feature maps with 5×5 kernels followed by LCN and 3D max pooling with strides $(2, 2, 2)$. The third layer is composed

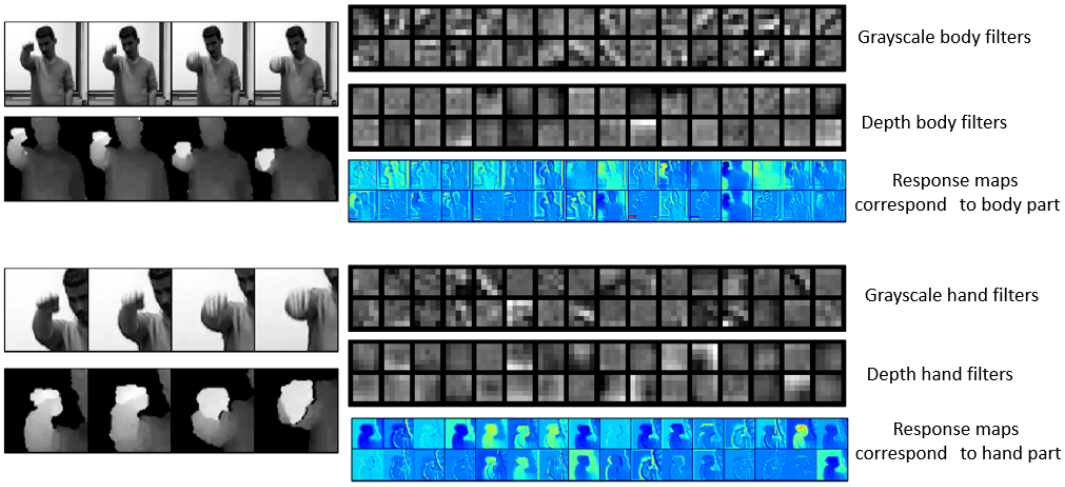


Fig. 5: Visualisation of input frames, first convolutional layer 5×5 filters, and corresponding response maps. As depth images are smoother than the grayscale ones, the corresponding filter are smoother as well.

of 64 feature maps with 4×4 kernels followed by 3D max pooling with strides $(1, 2, 2)$. All hand and body convolutional layer outputs of H6 are flattened in H7, and fed into one fully connected layer of size 1024. Finally, the output layer has $N_{\mathcal{H}}$ values, the number of states in the HMM state diagram (see Fig. 2).

3) Details of Learning: During training, dropout [**?**] is used as main regularisation approach to reduce overfitting. Nesterovs accelerated gradient descent (NAG) [**?**] with a fixed momentum-coefficient of 0.9 and mini-batches of size 64 are also used. The learning rate is initialised at 0.003 with a 5% decrease after each epoch. The weights of the 3DCNNs are randomly initialised with a normal distribution with $\mu = 0$ and $\sigma = 0.04$. The frame based validation error rate is 39.06% after 40 epochs. Compared with the skeleton module (16.5% validation error rate), the 3DCNN has a notable higher frame based error rate.

4) Looking into the Networks: visualisation of Filter Banks: The convolutional filter weights of the first layer are depicted in Fig. 5. The unique characteristics from the kernels are clearly visible: as hand input images (RGB and depth) have larger homogenous areas than the body inputs, the resulting filters are smoother than their body counterpart. In addition, while being smoother overall than the grayscale filters, depth filters exhibit stronger edges, as also reported in [**?**]. Finally, by looking at the joint depth-image response maps, we can notice that some filters better capture segmentation like information, while other are more edge oriented.

D. Multimodal Fusion

To combine the two modalities, two strategies can be used, as shown in Fig. 6: a late fusion approach and an intermediate fusion approach.

1) Late Fusion: This scheme fuses the combination of the emission probabilities estimated from the different input. While different combination schemes exist, here we considered a simple linear combination:

$$\log p(X_t|H_t) \propto \alpha \cdot \log p(X_t^s|H_t) + (1-\alpha) \cdot \log p(X_t^r|H_t) \tag{9}$$

where the different emission probabilities are provided by the modules described in IV-B and IV-C, and α is a coefficient that controls the contributions of each source of information and which is estimated by cross validation. Interestingly, the best performing α is close to 0.5, indicating that both modalities are equally important.

2) Intermediate Fusion: As an alternative to the late fusion scheme, we can take advantage of the high-level representation implicitly learned by each module (and represented by the V^s and V^r nodes of the penultimate layer of the respective networks, before the softmax) to fuse the modality in an intermediate fashion by concatenating these two layers in one layer of 2024 hidden unites and learning a cross-modality emission probability predictive network. Note that this is very similar in spirit to the approach proposed in [**?**] for audio-visual speech recognition. An important difference is that in [**?**], the same stacked RBMs/*DBN* architecture was used to represent both modalities before fusion, whereas in our case, a stacked RBMs/*DBN* and a *3DCNN* are used. Also, [**?**] proposed the use of a multimodal autoencoder to handle predictions when potentially only one modality migth be present, a point that we do not address.

The resulting architecture is trained by first initializing the weights of the deeper layers from the previously trained module, and then jointly fine tuning the whole network (including learning the last layer parameters). The training is stopped when the validation error rate stops decreasing ($\sim$15 epochs). We argue that using the "pre-trained" parameters is important due to the heterogeneity of the inputs of the system, and that the joint training should adjust parameters to handle this heterogeneity and produce the final estimates.

V. Experiments and Analysis

This Section reports the experiments performed to validate our model. First, we will introduce the ChaLearn dataset, and then present the experimental protocol we followed. In Section V-C, we will present

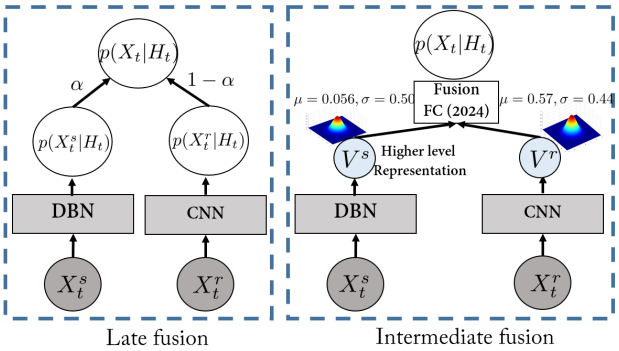


Fig. 6: Multimodal dynamic networks with late fusion scheme (left) and intermediate fusion scheme (right). The late approach simply combines the emission probabilities from two modalities. In the intermediate fusion scheme, each modality (skeleton and RGB-D) is first pre-trained separately, and their high-level representation V^s and V^r (the penultimate node layers of their neural networks) are concatenated to generate a shared representation. The resulting architecture is trained jointly.

and analyse the obtained results, including a discussion on the modeling elements. Finally, Section V-D will briefly discuss the computational complexity of the approach.

A. Chalearn LAP Dataset

The dataset used in this work is provided by the ChaLearn LAP [**?**] gesture spotting challenge[4]. The focus is on "multiple instance, user independent spotting" of gestures, which means learning to recognize gestures from several instances for each category performed by different users, drawn from a gesture vocabulary of 20 Italian cultural/anthropological signs. A gesture vocabulary is a set of unique gestures, generally related to a particular task.

The challenge dataset contains 940 videos sequences, each performed by a single person and composed of 10 to 20 gesture instances for a total of around $14,000$ gestures. There are 20 gesture classes, *i.e. vattene, vieniqui, perfetto, furbo, cheduepalle, chevuoi, daccordo, seipazzo, combinato, freganiente , ok, cosatifarei, basta, prendere, noncenepiu, fame, tantotempo, buonissimo, messidaccordo, sonostufo*, with a number a samples well balanced between classes. The average length of gestures is 39 frames, the minimum frame number for a gesture is 16 and the maximum frame number is 104.

This dataset is challenging because the "user independent" setting and some of gestures differ primarily in hand pose but not the overall arm motions as illustrated in Fig. 7 In terms of data, three modalities are

[4]http://gesture.chalearn.org/2014-looking-at-people-challenge/data-2014-challenge

(a) “Perfetto” (b) “Perfetto” (c) “Buonissimo”

(d) “OK” (e) “Non ce ne piu” (f) “Combinato”

(g) “Basta” (h) “Cheduepalle” (i) “Daccordo”

Fig. 7: Examples of gestures in the Chalearn dataset. This dataset is challenging because the “user independent” setting (a)&(b), some of gestures differ primarily in hand pose but not the overall arm motions (d)&(e) and some gestures require both hands to perform (g,h,i). Subtle hand movement (c) and differences in performing speed and range (f) also make recognising tasks challenging.

provided with the input videos: the sequence of skeleton joints, and the RGB and depth images (including a segmentation of the person performing the gesture).

B. Experimental protocol

1) Training and evaluation protocol: We follow the ChaLearn experimental protocol, in which the input sequences are split into 700 videos for training, and 240 sequences for testing and reporting results. Note that the test sequences are not segmented a priori and the gestures must be detected within a continuous data stream which, in addition to the targeted gestures, also contains noisy and out-of-vocabulary gestures. Furthermore, in the experiments, we split the training videos into 650 videos for learning the actual neural network model parameters, and 50 videos used as validation data for monitoring the training performance or selecting hyper-parameters.

2) Performance measures: Several measures can be used to evaluate the gesture recognition performance. In this work, we adopted the ChaLearn performance measure known as the Jaccard index, which

relies on a frame-by-frame prediction accuracy. More precisely, if GT_i denotes the sequence of ground truth labels in video i, and R_i the algorithm output, the Jaccard index of the video is defined as:

$$JI_i(GT_i, R_i, g) = \frac{N_s(GT_i, R_i, g)}{N_u(GT_i, R_i, g)}, \tag{10}$$

$$\text{and } JI_i = \frac{1}{|\mathcal{G}_i)|} \sum_{g \in \mathcal{G}_i} JI_i(GT_i, R_i, g) \tag{11}$$

where $N_s(GT_i, R_i, g)$ denotes the number of frames where the ground truth and result agree on the gesture class g, and $N_u(GT_i, R_i, g)$ denotes the number of frames labeled as a gesture frame g by either the ground truth or the algorithm, and $\mathcal{G}_i$ denotes the set of gestures either in the ground truth or detected by the algorithm in sequence i[5]. The average of the JI_i over all test videos is reported as performance measure. Note that experimentally, this measure tends to favours having more false positives than missing true positives, in order to increase the numerator.

Being defined at the frame level, the Jaccard index can vary due to variations of the segmentation (both in the ground truth and recognition) at gesture boundaries, which can be irrelevant from an application viewpoint. Thus, we also defined performance at the gesture event level by following the commonly used PASCAL challenge intersection over union criterion. More precisely, if for a gesture segment G, we have $\frac{G \cap R}{G \cup R} > 0.5$, where R denotes a recognized gesture segment of the same class, then the gesture is said to be recognized. If the same relation holds but with a gesture segment of another class, the prediction is incorrect. Otherwise the gesture is rated as undetected. This allows us to define the *Recognized*, *Confused* and *Missed* performance measures at the video level, which are further averaged over test sequences for reporting.

3) Tested systems: We evaluated the recognition performance made by the HMM applied to the emission probabilities estimated from either the skeleton data, the RGB-D image data, the late fusion scheme, and the intermediate fusion scheme. Note that in all cases the HMM output was further filtered to avoid false alarms, by considering gesture segments of less than 20 frames as noise and discarding them.

C. Results

Overall results. The overal performance of the algorithms are given in Tables I and II. As can be observed from both performance measures, the skeleton module usually performs better than the RGB-D module. In addition, its generalization capability is better than that of the RGB-D module, especially when

[5]Note that 'non gesture' frames are thus excluded from the counts.

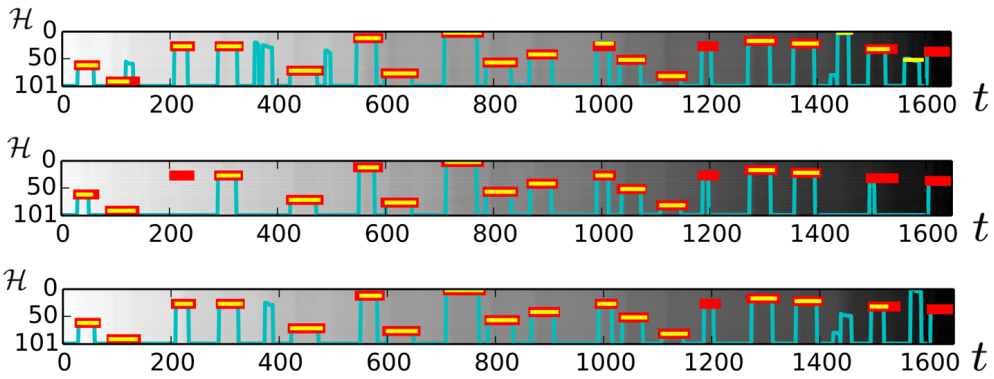


Fig. 8: Viterbi decoding of sample sequence #700, using skeleton (top), RGB-D (middle) and late fusion system (bottom). The x-axis represents time and the y-axis represents the hidden states of all classes and of the ergodic state (state 101) constituting the finite set $\mathcal{H}$. The cyan lines represent the viterbi shortest path, while red lines denote the ground truth labels, and the yellow segments are the predicted labels. The fusion method exploits the complementary properties of individual modules, *e.g.* around frame 200 the skeleton help solving the missed detection from the 3DCNN module, while around frame 1450, the 3DCNN module can help suppress the false positive prediction made by the skeleton module.

Module	Validation	Test
Skeleton – DBDN	0.783	0.779
RGB-D – 3DCNN	0.752	0.717
Multimodal Late Fusion	0.817	0.809
Multimodal Inter. Fusion	0.800	0.798

TABLE I: Results in terms of Jaccard index *JI* for the different network structures and modalities modeling the emission probabilities.

measured with the Jaccard index where there is almost no drop of performance between the validation and test data. One possible explanation is that the information in the skeleton data is more robust, as it benefited from training using huge and highly varied data [**?**]: around on million images from both realistic and synthetic depth images were used to train the decision forest classifiers involved in the joints extraction. On the over hand, as the RGB-D module relies on the raw data and was learned only from the ChaLearn training set, it may suffer from some overfitting. Another interesting conclusion that can be drawn from Table II is that while most errors from the RGB-D module are due to under detection (the *Missed* rate is 19.7%, whereas it is only 4.1% for the skeleton), the skeleton module is more reactive to gesture activity, but makes more mistakes (the *Confused* rate is 12.3% vs 4.5% for RGB-D).

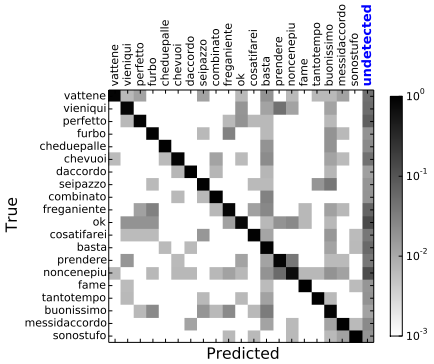
Finally, the results also demonstrate that the combination of both modalities is more robust, as shown

	%	Validation	Test
Skeleton - DBDN	*Recognized*	86.3	83.6
	Confused	11.4	12.3
	Missed	2.3	4.1
RGB-D - 3DCNN	*Recognized*	78.7	75.8
	Confused	5.2	4.5
	Missed	16.1	19.7
Multimodal Late Fusion	*Recognized*	87.9	86.4
	Confused	9.1	8.7
	Missed	3.0	4.9
Multimodal Inter. Fusion	*Recognized*	86.5	85.5
	Confused	7.3	6.8
	Missed	6.2	7.7

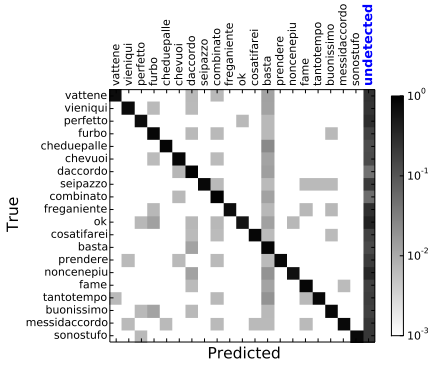
TABLE II: Gesture classification performance at the event level, in percentage of the number of ground truth gestures.

by the recognition rate increase and the smaller drop in the generalization performance (for instance the decrease of the *Recognized* rate is lower than for the skeleton data alone).

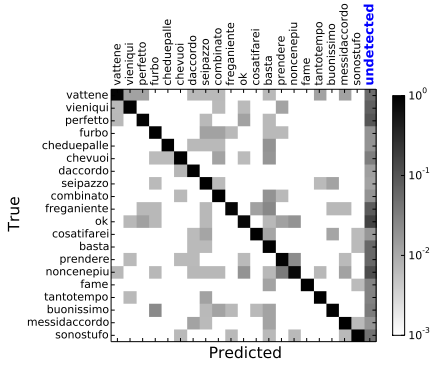
Confusion matrices. The confusion matrices (in log-form) in Fig. 9 better illustrate the complementarity of the behaviors of the two modalities. The higher underdetection of RGB-D is clearly visible (whiter matrix, except last 'undetected' column). We can also notice that some gestures are more easily recognized than others, or catch the difficult instances of other gestures. This is the case of the "Basta" gesture, whose arms motion resembles the start and end of the arm motion of many other gesture (see Fig. 7). Whatever the modality, its model thus tends to recognize few instance of all other gesture classes, whenever their likelihood are low when being evaluated using the HMM states associated with their true label due to too much variability. Similarly, the hand movement and pose of the "Buenissimo" gesture is present in several other gesture classes, whose instances are then often confused with "Buenissimo" when relying on the skeleton information alone. However, as these gestures differ primarily in their hand pose, such confusion is much more reduced using the RGB-D domain, or when fusing the skeleton and RGB-D modules. The complementary properties of the two modalities is also illustrated from the Viterbi path decoding plot in Fig. 8. In general, the benefit of this complementarity between arm pose/gesture and hand pose can be observed from the whiter confusion matrix than in the skeleton case (less confusion due to hand pose information from RGB-D) and much less under-detection than in the RGB-D case (better upper-body pose discrimination thanks to skeleton input).



(a) Skeleton - DBN



(b) RGB-D - 3DCNN



(c) Multimodal Late Fusion

Fig. 9: Confusion Matrices (log-norm) for the different modalities.

However, the modalities by themselves have more difficulties in correcting the recognition errors which are due to variations coming from the performer, like differentiating people that gesticulate more (see Fig. 11).

Late vs. Intermediate fusion. The results in Tab. I and II show that the intermediate fusion system improved individual modalities, but without outperforming the late fusion system. The result is counter-intuitive, as we would have expected the cross-modality learning in the intermediate fusion scheme to result in better emission probability predictions, as compared to the simple score fusion in the late system. One possible explanation is that the independence assumption of the late scheme better preserves both the complementarity and redundancy of the different modalities, properties which are important for fusion. Another related explanation is that in the intermediate fusion learning process, one modality may dominate and skew the network towards learning that specific module and lowering the importance of the other one. The large difference between the mean activations of the skeleton module neurons which are predominantly larger than those of the RGB-D ConvNet's (0.57 *vs.* 0.056) can be an indicator of such a bias during the multimodal fine-tuning phase and support this conjecture, even if these mean activations are not directly comparable due to the neuron heterogeneity (the skeleton DBN has logistic units whereas the 3DCNN ConvNet has relu units). Note that such heterogeneity was not present when fusing modalities in [**?**], where better registration and less spatial registration variability in lip images allowed to also resort to the same stacked RBMs for the visual modality (rather than *3DCNN*) and the audio one. More investigation on how to handle heterogeneous networks should be conducted.

HMM benefit. As the emission probabilities are learned in a discriminative manner, one could wonder whether the HMM brings benefit beyond smoothing. To investigate this issue, we removed the temporal structure as follows: for a given gesture $\boldsymbol{a}$, we computed its score at time t, $Score(\boldsymbol{a}, t)$, by summing the emission probabilities $p(X_t|H_t = h)$ for all nodes associated to that gesture, *i.e.* $h \in \mathcal{H}_{\boldsymbol{a}}$. This score is then smoothed in the temporal domain (using a window of 5 frames) to obtain $\widehat{Score}(\boldsymbol{a}, t)$. Finally, following [**?**], the gesture recognition proceeds in two steps: first finding gesture segments by thresholding the score of the ergodic state; then, for each resulting gesture segment, the recognized gesture is defined as the one whose average score within the segment is the highest. Fig. 10 illustrates this process along with the DDNN and ground-truth. In general, we could observe that better decisions on the presence of gestures and on the boundaries where a gesture starts and ends are achieved with the proposed *DDNN* thanks to the use of the state-diagram defined in Fig. 2, as compared to the above method, where deciding on a gesture detection threshold is rather unstable and quite sequence dependent. Indeed, the overall performance of the above scheme without the HMM temporal sequencing is reduced to $JI = 0.66$, while the *Recognized*, *Confused* and *Missed* corresponding to Table II for the test set are 76.6 , 5.3 and 18.1. However, note that the above method relying on only the gesture probability learned using neural networks on 5 frame inputs still outperforms the Jaccard index of 0.413 obtained by [**?**]

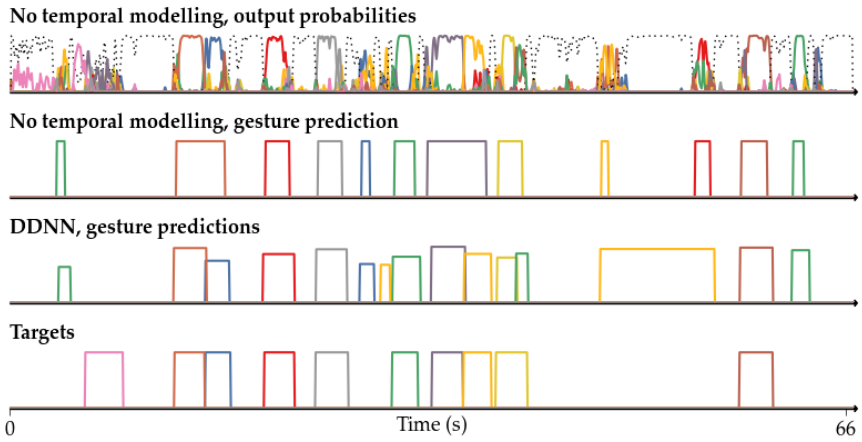


Fig. 10: HMM temporal contribution. First row: output emission probabilities for each gesture as given by the late fusion scheme (see text) for the test set #703. The dashed line represents the probability of the Resting/Other gesture state, while other colour represent different gestures. Second row: resulting recognized gestures, without HMM modeling. Third row: HMM output. Fourth row: ground truth segmentation. Without temporal modelling, the decision boundary of a gesture will be more rugged and it is more difficult to make hard decisions of where the gesture starts or ends. Hence, in general, it causes miss-detection and miss-merging. Thanks to the HMM temporal modelling and Viterbi path decoding, gesture boundaries are usually cleaner defined from the Resting state to the gesture states, resembling the behavior of the manual annotators with better accuracy.

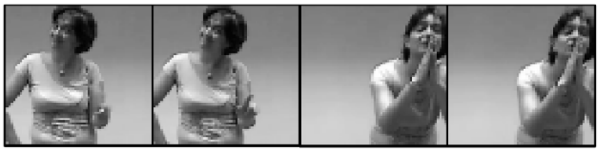
when using a 5 frames template matching system where all the features are handcrafted.

Comparison with the state-of-the-art. The performance of recent state-of-the-art techniques is given in Table III. The first half of the table resorts to hand crafted feature approaches and then usually a second stage classifier. Our proposed system performs on par with the top two methods. However, hand crafted feature methods' performance are saturated regardless of the increase training data. The representation learning methods in the second half of the Table perform comparably with the best hand crafted feature approaches and the top representation method achieves the best Jaccard index score. Given more training data, it is expected that the networks will be able to be more adapted to the "user independent" setting. It also worths noting that our proposed system is the only method that incorporates the temporal modelling element rather than sliding window approach. We believe this is an interesting research direction that can be more adapted to various lengths of gestures and relevant temporal factors.

D. Computational Complexity

We can distinguish between two complexities: the training one, and the test one.

(a) Sample #806



(b) Sample #702

Fig. 11: Examples of performer variations in the upper body dynamic. Most performers tend to keep their upper-body static while performing the gesture, leading to good recognition performance (Jaccard index of person on the top is 0.95 for the late fusion system). Some persons are more involved and move more vehemently (person at the bootom, Jaccard index of 0.61), which can affect the recognition algorithm itself (bottom left samples) or even the skeleton tracking (bottom right; note that normally cropped images are centered vertically on the head position).

Module	Skeleton	RGB-D	Fusion
[**?**] 3 set skeletal & HOG, Boosted classifier	0.791	-	0.822
[**?**] 3D skeletal pose & HOG, MRF	0.790	-	0.827
[**?**] Dense trajectory (HOG, HOF, MBH)	-	0.792	-
[**?**] Template based Random Forest Classifier	-	-	0.747
[**?**] Fisher Vector, Dynamic Programming	0.745	-	-
[**?**] Independent Subspace Analysis, RF	-	0.649	-
[**?**] PHOG, SVM, HMM	0.454	0.462	0.597
[**?**] Representation Learning (multiscal)	0.808	0.809	**0.849**
[**?**] CNN	-	0.789	-
[**?**] Deep Neural Networks	0.747	0.637	0.804
DDNN (this work)	0.779	0.717	0.809

TABLE III: Comparison of results in terms of the ChaLearn Jaccard index with state-of-the-art related works.

Complexity at training time. Although training deep neural network using stochastic gradient descent is computationally intensive, the reuse of pre-trained network parameters as done in our case can help with better initialisation and lead to faster convergence. We can observe different training time in function of the modality (and architecture). Specifically, using a modern GPU (GeForece GTX TITAN Black) and conv op. by Theano [**?**], the training time of each epoch of the DBN skeleton module is less than 300 seconds and allows training the required 500 epochs within 2 days. The training time of each epoch

of the 3DCNN RGB-D module is much more expensive, taking more than 10,000 seconds. Hence, 40 epochs takes around 5 days to train. The fusion network being initialised with the individual module parameters, its training time is half that of the 3DCNN.

Complexity at test time. Given the learned models, our framework with the above GPU can perform real-time video sequence labelling with a low inference cost. More specifically, a single feed forward neural network incurs linear computational time ($\mathcal{O}(T)$), and is efficient because it requires only matrix products and convolution operations. The complexity of the Viterbi algorithm is itself of $\mathcal{O}(T * |S|^2)$, where T is the number of frames and $|S|$ the number of states, and thus performs real-time given our state-space. In practice, our multimodal neural network can be deployed at 90 FPS. The preprocessing part takes most of the time and our un-optimized version runs at 25 FPS, while the Viterbi decoding runs at 90 FPS. Hence, the overall system can achieve faster than real-time performance.

VI. Conclusion and future work

Hand-engineered, task-specific features are often time-consuming to design but they are limited to some certain tasks. This difficulty is even more pronounced with multimodal data as we would like the features to relate to multiple data sources. In this paper, we presented a novel deep dynamic neural network (DDNN) for learning contextual frame-level representations and modelling emission probabilities in the framework of an HMM temporal model. Different feature learning methods (DBN and 3DCNN) suited to the heterogeneous inputs from skeletal joints, RGB images, and depth images were proposed, as well as different fusion schemes. Experimental results on bi-modal gesture time series show that the multimodal DDNN framework can learn good models of the joint space of multiple sensory inputs, improving over unimodal input.

There are several directions for future work. Our results with those of other recent works suggest that learning features directly from data is a very important research direction and that with more and more data and flops-free computational power, learning-based methods are not only more generalisable to many domains, but also are powerful in combination with other well-studied probabilistic graphical models for dynamical modelling and reasoning. In this view, the learning of better shared and complementary representation among multimodal and heterogeneous inputs, as done in [**?**], requires more exploration. In addition, while the proposed HMM provided a good basis for the temporal modeling of gestures, other more discriminant temporal approaches such as Conditional Random Field or further and better variants [**?**] could be directly exploited at their advantage in conjuction with our deep neural network learning approach. Ultimately, in a logical way, these two research directions converge into the investigation of a

single and unified deep learning framework fusing heterogeneous modalities by using recent Recurrent Neural Networks such as Long Short Term Memory [**?**] for modelling the temporal component of the problem.

APPENDIX A
DETAILS OF THE CODE

The python code using Theano [**?**] for this work can be found at:
`https://github.com/stevenwudi/chalearn2014_wudi_lio`

ACKNOWLEDGMENT

The authors would like to thank Sander Dieleman for his guidance in building, training and initialising convolutional neural networks.

Di Wu Biography text here.

Lionel Pigou Biography text here.

Pieter-Jan Kindermans Biography text here.

Nam Le Biography text here.

Ling Shao Biography text here.

Joni Dambre Biography text here.

Jean-Marc Odobez Biography text here.

Response to Reviewer 1

Comment: *After reading the responses and revised paper, I am happy to say that my confusions and criticisms are rebutted. I think the authors for their diligence in indulging my requests for clarification and additional experiments. I believe the paper is much improved and should be accepted.*

Specifically: - The additional discussion convinced me of sufficient contribution (jointly learning features and emission probabilities from heterogeneous input streams for gesture recognition). - I agree with the authors that "More investigation on how to handle heterogeneous networks should be conducted". The additional experiments have identified important problems (e.g. differences in mean activation) for future work, which is a contribution in itself. - The qualitative analysis is greatly improved, giving insight. - I am happy to see intermediate fusion and fine tuning have been considered.

Below I enumerate my few remaining concerns, which I believe the authors should consider in a minor revision:

A few minor substantive points: 4.1: You might consider a latent variable model for the state labels $y_{i,t}$ instead of 'force alignment'. It might help to briefly justify why you think forced assignment is good enough.

Figure 8: Label the y-axis on the figure. The information is in the caption, but readers are lazy.

5.3, Late vs. Intermediate fusion: If done right, the intermediate fusion should work at-least-as-well as the late fusion? But, as the authors discuss, it is not trivial to 'do it right'. I think they have done their due-diligence here and, importantly, exposed a problem.

A few minor grammatical problems:

*The authors jointly learn features and emission probabilities from *heterogeneous input streams* for dynamic gesture recognition. After reading the rebuttal and spending some time searching the literature I agree the work is new and interesting. The authors identify problems and challenges for future work. It isn't a revolutionary paper, but it seems an important and valid milestone which contributes understanding to our community.*

Response to Reviewer 2

Comment: *The authors made an excellent effort to improve the paper based on all the reviewers' comments. I recommend acceptance. I recommend the authors to make an additional pass to correct typos e.g. combinatin =¿ combination.*

The paper proposes the fusion of the output from a Gaussian-Bernoulli Deep Belief network operating on skeletal features and the output of a Convolutional Neural Network operating on RGBD data to perform gesture segmentation and recognition. The paper advances the field of gesture recognition by using both data sources and deep learning architectures within a Hidden Markov Model chain. The results are improved compared to using either architecture independently.

The revised version of the paper includes another fusion scheme: instead of averaging the outputs of the two networks (called "late fusion"), the paper proposes to concatenate the high-level representations produced by their penultimate layers and process them through another classification layer to get the emission probabilities (called "intermediate fusion"). Unfortunately, this type of fusion, which made more sense to me, performs slightly worse. I suspect that the reason is that different types of units (ReLU vs sigmoid units) are used in the two networks causing incompatible scales in the outputs of the penultimate layers, as the authors indicate. In any case, this add-on is useful and can inspire future investigation on this topic and better implementations.

Response to Reviewer 3

Comment: *In this second revision, readability and organization of the manuscript have been significantly improved. Additional experiments have been conducted, and interesting insights and discussions have been added, including comparison of intermediate vs late fusion and temporal modeling vs aggregation of per-frame predictions by voting. Finally, more consistent and complete overview of related work is included. There are no changes in the method though: it remains interesting from a practical point of view, but not particularly novel.*

In section II, please note that a year earlier, ChaLearn 2013 competition was dominated by exploiting HMMs (also RNNs) for the same task and on the same dataset. http://gesture.chalearn.org/2013-multi-modal-challenge/workshop-2013-challenge Some works are published: [HMM] Fusing multi-modal features for gesture recognition, ICMI workshop, 2013 [HMM] Nandakumar et al., A Multi-modal Gesture Recognition System Using Audio, Video, and Skeletal Joint Data, ICMI workshop, 2013 [RNN] Neverova et al., A multi-scale approach to gesture detection and recognition, ICCV workshop, 2013

Some sections of the manuscript require additional proof reading: please check for typos (they are many, especially in section III), verb endings and spelling of names in your references.

In terms of performance, the described DBN for treating raw skeleton data looks promising (16.5

The authors propose a deep learning framework for multi-modal gesture recognition based on color, depth and skeleton streams provided by the Kinect sensor. One of the key contributions is combining feature learning with HMM-based temporal modeling. Although exploring the deep learning approaches in the given context is certainly promising, the proposed implementation is rather straightforward and the reported results are clearly behind the state-of-the-art.