

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. A semi-supervised hierarchical dynamic framework is proposed for simultaneous gesture segmentation and recognition taking skeleton, depth and RGB images as input observations. Unlike the traditional construction of complex handcrafted features, all inputs are learnt by deep neural networks: the skeletal module is modelled by deep belief networks (*DBN*); the depth and RGB module are modelled by 3D convolutional neural networks (*3DCNN*) to extract high-level spatio-temporal features. The learned representations are then used for estimating emission probabilities of the hidden Markov models to infer a gesture sequence. The framework can be easily extended by including an ergodic state to segment and recognise video sequences by a frame-to-frame mechanism, making online segmentation and recognition possible. 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 approaches and other learning based methods, opening the doors for using deep learning techniques to explore multimodal time series.

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

1 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 [1], [2], [3] 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 [4], Cuboids [5] and Hessian3D [6]. For descriptors, popular methods are Cuboids [7], HOG/HOF [4], HOG3D [8] and Extended SURF [6]. In recent work of Wang *et al.* [9], 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.* [10], 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 [11], [12], [13].

Due to the recent resurgence of neural networks invoked

by Hinton and others [14], 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 [15] 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 [16] classifies 1.2 million high-resolution images into 1000 different classes; multi-column deep neural networks [17] achieve near-human performance on the handwritten digits and traffic signs recognition benchmarks; 3D convolutional neural networks [18] [19] recognise human actions in surveillance videos; deep belief networks combined with hidden Markov models [20] [21] for acoustic and skeletal joints modelling outperform the decade-dominating paradigm of Gaussian mixture models in conjunction with hidden Markov models. And recently, Baidu research proposed a DeepSpeech system [22] that combines a well-optimised recurrent neural network (RNN) training system, achieving the best error rate on 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 [13], or by training the model in an unsupervised manner [12], or by forcing the model parameters to be identical for different input locations (as in convolutional neural networks [16], [17], [18]).

On the sensor side, due to the immense popularity of Microsoft Kinect [23] [24], 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 [25], [26], [27], [28] 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 [4] [29] [30].

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. This paper is an extension of the works of [21], [31] and [32]. Within a temporal framework allowing the labelling of videos in a frame-by-frame fashion, thus making it possible for online gesture recognition, 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 multiple channel inputs such as depth and RGB images;
- Early 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 works for 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 conducts the experimental analysis and Section VI concludes the paper with discussions related to future works.

2 RELATED WORK

Gesture recognition has drawn increasing attention of 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 [33] proposed the notion of “action points” to serve as natural temporal anchors of simple human actions using a Hidden Markov Model. Wang *et al.* [34] 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 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.* [19] 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.* [11] also explored 3D convolutional networks for learning spatio-temporal features for videos. The experiments in [31] 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 [17] 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. Socher *et al.* [35] 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.* [36] 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 [37] 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, Nevero *et al.* [38] proposed a pose descriptor consisting of 7 logical subsets for skeleton features while Monnier *et al.* [39] 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.* [40]) and histograms of oriented gradients (HOG) descriptor for RGB-D images around hand regions. In [41], the state-of-the-art dense trajectory [9] handcrafted features are adopted for the RGB module.

There is a gradual trend to learn the features for gesture

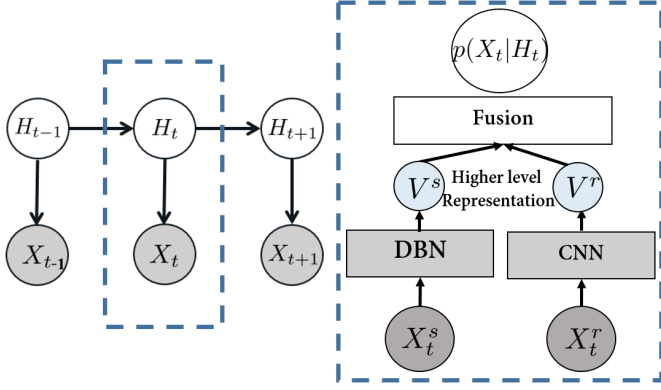


Fig. 1: Gesture recognition model: the temporal model is a HMM (left), whose emission probability $p(X_t|H_t)$ (right) is modeled by a forward-linked neural network chain. More precisely, observations X_t (skeletal features X_t^s , or RGB-D image features X_t^r) are first passed through appropriate deep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

recognition in videos. For instance, the recent methods in [31], [32] focused on single modalities, used deep networks to learn representations from skeleton data [31] or from RGB-D data [32]. Nevero *et al.* [38] 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.

3 MODEL FORMULATION

Inspired by the framework successfully applied to speech recognition [20], 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 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 4.

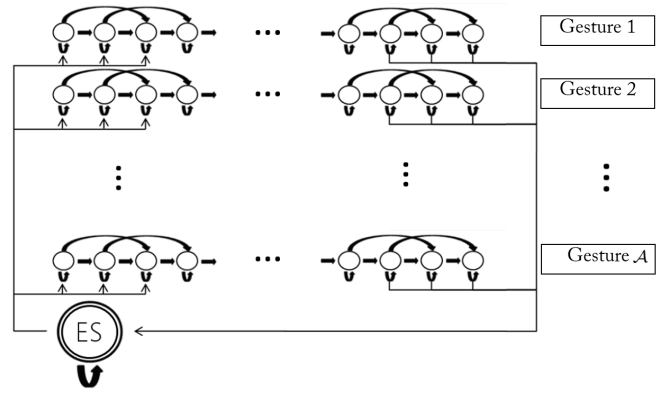


Fig. 2: State diagram of the *ES-HMM* model for low-latency gesture segmentation and recognition. An ergodic state (\mathcal{ES}) is used to model the resting position between gesture sequences. Each node represents a single state and each row represents a single gesture model. The arrows indicate possible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

3.1 Deep Dynamic Neural Networks

The proposed deep dynamic neural network (*DDNN*) can be seen as an extension of [21], 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 [14] from different input modules.

A continuous-observation HMM with discrete hidden states 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 X_t^r as shown in the graphical representation in Fig. 1. The unobserved variable H_t takes on values in a finite set \mathcal{H} composed of $N_{\mathcal{H}}$ states related to the different gestures. In practice, H_t can be interpreted as a segment of a gesture a .

The intuition motivating this construction 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 the chain. Classically under the HMM assumption, the joint probability of observations and states is given by:

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

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.

3.2 State-transition model

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 define a Markov model of that gesture. For example, for action sequence “tennis serving”, the action sequence can implicitly 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 $\mathcal{H} = (\bigcup_{a \in \mathcal{A}} \mathcal{H}_a) \cup \{\mathcal{ES}\}$.

Overall, we refer to the model as the ergodic states hidden Markov model (*ES-HMM*) for simultaneously gesture segmentation and recognition. It differs from the firing hidden Markov model of [33] 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 [42]. 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. 11.

3.3 Learning the emission probability $p(X_t|H_t)$

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 suited to the input modality, as summarized in the right of Fig. 1.

More specifically, for the skeletal features, we rely on a Deep-Belief Network (DBN) trained in two steps [43]: in the first step, stacked Restricted Boltzmann Machines (RBM) is 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 early 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 [20]. For instance, assume that the components of the input feature space can be separated into two sub-spaces characterized by N and M significantly different patterns in the training data, respectively, and that the occurrences of these patterns are relatively independent¹. 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 explain 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 [43] has been shown to have several advantages. It has been observed that variational learning [44], which tries to optimize the data-likelihood while minimizing 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 pair wise joints features as raw input, the data-driven approach network will be able to extract relational multi-joints features which are relevant to the target classes. For instance, from the “toss” action data, a wrist joints rotating 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 joints configurations defined in [45] [46] [33] [47].

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

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

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 compensate 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.

4 MODEL IMPLEMENTATION

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

4.1 Ergodic States Hidden Markov Model

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

The number of states N_{H_a} associated to an individual gesture has been set to 5. Therefore, in total, the number of states is $N_H = 20 \times 5 + 1 = 101$ when conducting experiments on the Chalearn dataset containing 20 classes.

The training data in Chalearn is given as a set of sequences $\mathbf{x}_i = [x_{i,1}, \dots, x_{i,t}, \dots, x_{i,T_i}]$ 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 $\mathbf{y}_i = [y_{i,1}, \dots, y_{i,t}, \dots, y_{i,T_i}]$, the sequence of state labels $y_{i,t}$ associated to each frame. To do so, a force alignment is used which means that if the i^{th} sequence is a gesture \mathbf{a} , then the first $\lfloor \frac{T_i}{5} \rfloor$ frames are assigned to state h_a^1 (the first state of gesture \mathbf{a}), the following $\lfloor \frac{T_i}{5} \rfloor$ frames are assigned to h_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.

4.2 Skeleton Module

4.2.1 Skeleton input features

Given our task, only the $N_j = 11$ upper body joints are relevant and considered, namely “*ElbowLeft, WristLeft,*

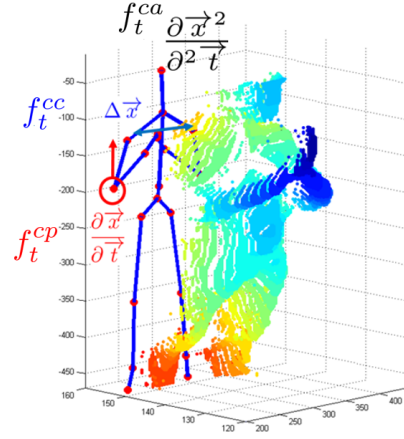


Fig. 3: A point cloud projection of a depth image and the 3D positional features.

ShoulderLeft, HandLeft, ElbowRight, WristRight, ShoulderRight, HandRight, Head, Spine, HipCenter”. The raw skeleton features of time t are defined as $x_t^s = [x_t^{s,1}, \dots, x_t^{s,N_j}]$. To capture the gesture dynamics, rather than using x_t^s as raw input to our data driven approach, we follow the approach of [21] and compute the 3D positional pairwise differences of joints as well as temporal derivatives, defined as (as shown in Fig. 3) ²:

$$f_t^{cc} = \{x_t^{s,i} - x_t^{s,j} | i, j = 1, 2, \dots, N_j; i \neq j\} \quad (2)$$

$$f_t^{cp} = \{x_{t+1}^{s,i} - x_t^{s,i} | i = 1, 2, \dots, N_j\} \quad (3)$$

$$f_t^{ca} = \{x_{t+1}^{s,i} - 2 \times x_t^{s,i} + x_{t-1}^{s,i} | i = 1, 2, \dots, N_j\} \quad (4)$$

This results in an input feature vector $f_t = [f_t^{cc}, f_t^{cp}, f_t^{ca}]$ of dimension $N_f = N_j \times (\frac{N_j}{2} + N_j + N_j) \times 3 = 891$. 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 [20].

4.2.2 Modeling X_t^s using Deep Belief Networks

Once we have the skeleton input feature f which served as the input X_t^s for DBN in Fig. 1. The emission probability $p(X_t|H_t)$ can be generated using a Deep Belief Networks. Because input skeletal features f are continuous instead of binomial features, we use the Gaussian-Bernoulli RBM (GRBM) [43] to model the energy term of first layer with

2. Note that the offset features used in [21] 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.

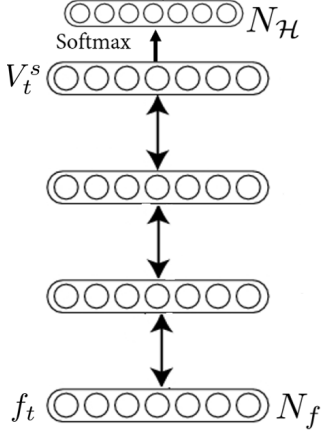


Fig. 4: Deep Belief Network using skeleton input f to predict emission probability $p(X_t|H_t)$.

continuous visible units v of dimension D connected to the hidden binary stochastic units $h \in \{0, 1\}^F$. More precisely:

$$E(v, h; \theta) = - \sum_{i=1}^D \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i=1}^D \sum_{j=1}^F W_{ij} h_j \frac{v_i}{\sigma_i} - \sum_{j=1}^F a_j h_j \quad (5)$$

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 ; b_i is the bias term of the visible units and a_j is the bias term of the hidden units.

The conditional distributions needed for inference and generation are given by:

$$P(h_{j=1}|\mathbf{v}) = g(\sum_i W_{ij} v_i + a_j); \quad (6)$$

$$P(v_{i=1}|\mathbf{h}) = \mathcal{N}(v_i | \mu_i, \sigma_i^2). \quad (7)$$

where $\mu_i = b_i + \sigma_i^2 \sum_j W_{ij} h_j$ and \mathcal{N} is the normal distribution. In general, we normalise the data (mean subtraction and standard deviation division) in the preprocessing phase. Hence, in practice, instead of learning σ_i^2 , one would typically use a fixed, predetermined unit value 1 for σ_i^2 .

By stacking a softmax layer on top of the RBMs as in Fig. 4, a Deep Belief Network is built for predicting emission probability $p(X_t|H_t)$. The weights are fine-tuned slightly from the top.

For high-level skeleton feature extraction, the network architectures is $[N_f, 2000, 2000, 1000, N_H]$, where $N_f = 891$ is the observation domain dimensionality; N_H is the output target class.

In the training set, there are in total 400 117 frames. During the training of the DBN, 90% is used for training, 8% for validation (for the purpose of early stopping) 2% is used for test evaluation. The feed forward networks are pre-trained with a fixed recipe using stochastic gradient decent with a mini-batch size of 200 training cases. Unsupervised initialisations (we run 100 epochs for unsupervised pre-training) tend to avoid suboptimal local minima and increase the networks performance stability. For Gaussian-Bernoulli RBMs, the learning rate is fixed at 0.001 while for binary-binary RBMs the learning is 0.01 (note that in general, training GRBMs requires smaller learning rates).

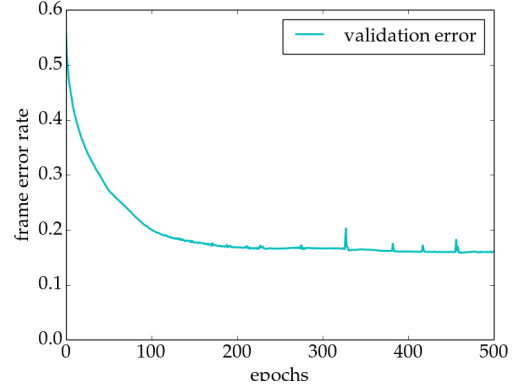


Fig. 5: Deep Belief Network frame based validation error rate for the skeleton modality. The 0.05 frame error rate indicates the good generalisation property of the DBN skeleton module.

For fine-tuning, the learning rate 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%, with 16.15% on the test set. The frame based validation error rate is shown in Fig 5 .

4.3 RGB & Depth 3D Module

4.3.1 Preprocessing

Although DeepMind technology [48] 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 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. 8.

4.3.2 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

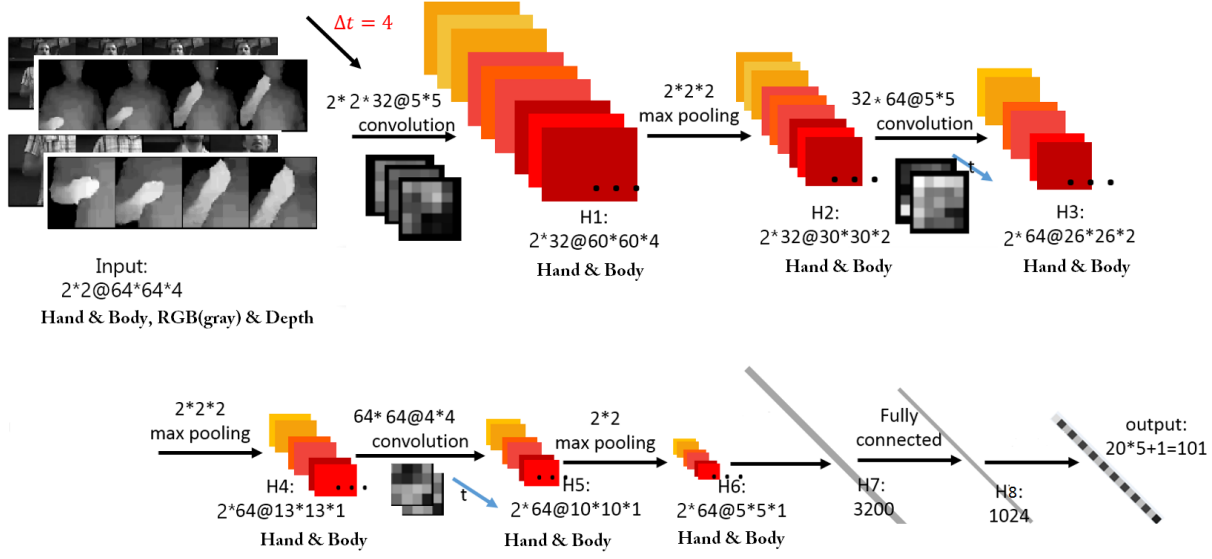


Fig. 6: 3DCNN architecture. The input is $2 \times 2 @ 64 \times 64 \times 4$, meaning 2 modalities (depth and RGB) for the hand and body regions, each being 4 consecutive 64 by 64 frames stacked together.

convolving a 3D kernel to the cuboid formed by stacking multiple contiguous frames together. We follow the nomenclature of in [19]. However, instead of using *tanh* units [19], Rectified Linear Units (*ReLU*s) [16] 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. 6: 4 types of input contextual frames are stacked as size $64 \times 64 \times 4$ (as illustrated in Fig. 8). The first layer (H1) consists of 32 feature maps produced by 5×5 spatial convolutional kernels, followed by local contrast normalisation (LCN) [49]. 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 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 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_H values, the number of states in the HMM state diagram (see Fig. 2).

4.3.3 Details of Learning

During training, dropout [50] is used as main regularisation approach to reduce overfitting. Nesterovs accelerated gradient descent (NAG) [51] 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

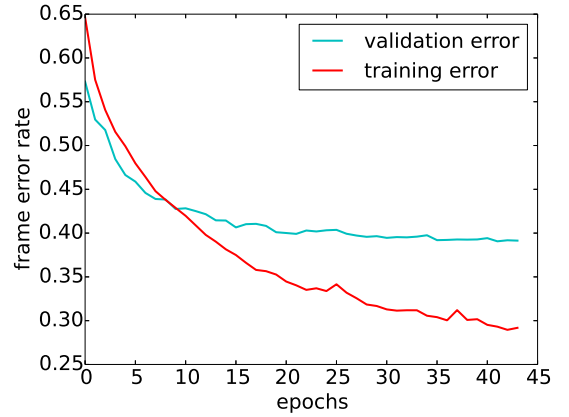


Fig. 7: Frame based error rate of the 3DCNN. The error rates are much higher than when using the skeleton module (see Fig. 5) indicating that learning from images is more difficult.

with a normal distribution with $\mu = 0$ and $\sigma = 0.04$. The frame based validation error rate is 39.06% after 40 epochs as shown in Fig. 7. Compared with the skeleton module (16.5% validation error rate), the 3DCNN has a notable higher frame based error rate.

4.3.4 Looking into the Networks: visualisation of Filter Banks

The convolutional filter weights of the first layer are depicted in Fig. 8. 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 [35].

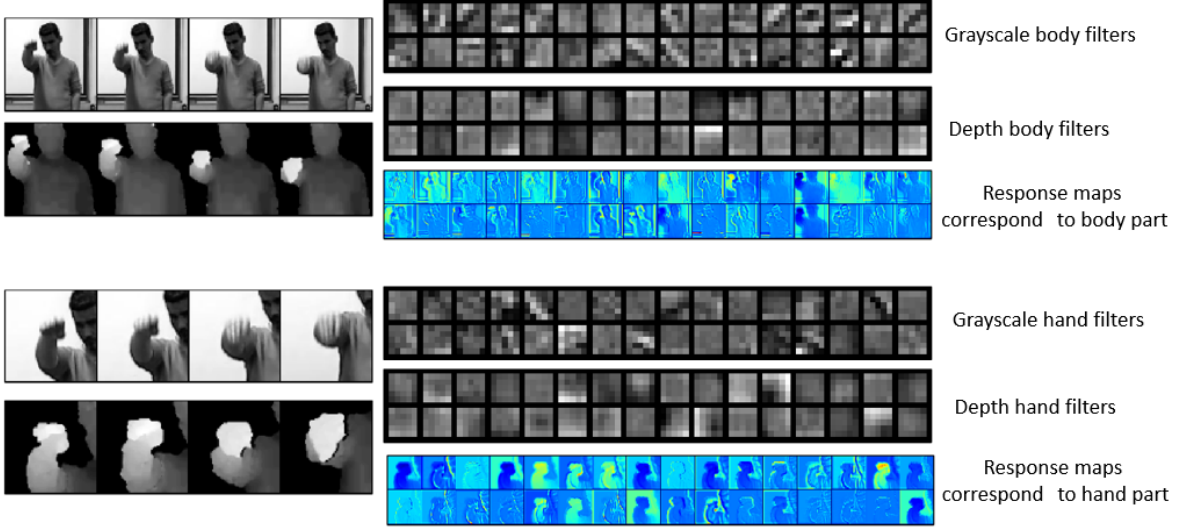


Fig. 8: Visualisation of input frames, first convolutional layer 5×5 filters, and corresponding response maps. As depth images are smoother than the grayscale ones, the corresponding filter are smoother as well.

4.4 Multimodal Fusion

To combine the two modalities, two strategies can be used, as shown in Fig. 9: a late fusion approach and an early fusion approach.

4.4.1 Late Fusion

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

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

where the different emission probabilities are provided by the modules described in 4.2 and 4.3, 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.

4.4.2 Early 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 early fashion by concatenating these two layers in one layer of 2024 hidden unites and learning a cross-modality emission probability predictive network. 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) and stop the training when the validation error rate stops decreasing (~ 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.

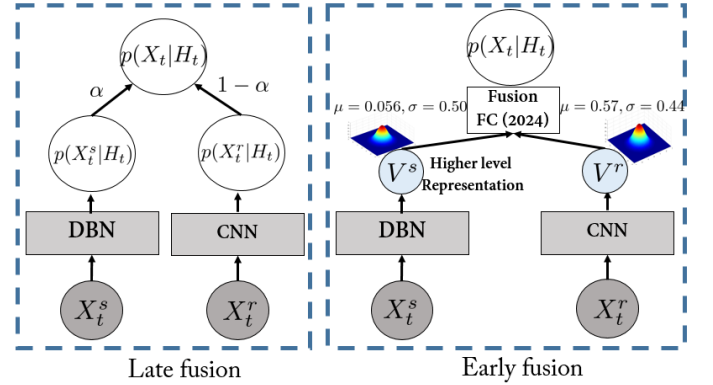


Fig. 9: Multimodal dynamic networks with late fusion scheme (left) and early fusion scheme (right). The late approach simply combines the emission probabilities from two modalities. In the early fusion scheme, each modality (skeleton and RGBD) is first pre-trained separately, and their high-level representation 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.

5 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 5.3, we will present and analyse the obtained results, including a discussion on the modeling elements. Finally, Section 5.4 will briefly discuss the computational complexity of the approach.

5.1 Chalearn LAP Dataset

JMO comment: In this part, provide more information about the dataset: description, content, types of gestures



(a) "OK"



(b) "Non ce ne piu"



(c) "Basta"



(d) "Buonissimo"



(e) "Daccordo"



(f) "Combinato"

Fig. 10:

JMO comment: Add other examples of gestures, interesting to comment upon take them from chalearn website. From the confusion matrix, i would propose: "Basta", buenissimo, daccordo, combinato

DWU comment: We may not show all the examples here

Examples of gestures in the Chalearn dataset. Note that some gestures primarily differ primarily in hand pose but not the arm motions, like "OK" vs "Non ce ne piu". "Basta" and "Combinato" are the gestures that got mostly misclassified.

(what are the classes), illustration of the gestures with some pictures (esp. for gestures that have impact on data -for example, expand fig 14 where you have the two ok and noncepiu examples with other ones, and put it earlier in the document. why is the dataset challenging ?

Also, provide some statistics about the dataset: rough duration of the gestures (average, min max). How many person are performing the gestures, how many occurrences

Gesture	Length	MAP
vattene	38.0	0.882
vieniqui	36.1	0.902
perfetto	36.7	0.877
furbo	40.1	0.945
cheduepalle	36.9	0.949
chevuoi	35.9	0.926
daccordo	40.2	0.982
seipazzo	43.5	0.963
combinato	45.6	0.956
freganiente	37.8	0.852
ok	30.8	0.778
cosatifarei	41.9	0.942
basta	35.7	0.924
prendere	37.9	0.892
noncenepiu	37.5	0.774
fame	41.3	0.958
tantotempo	39.1	0.955
buonissimo	43.5	0.905
messidaccordo	44.5	0.951
sonostufo	39.9	0.926

TABLE 1:

DWU comment: Average length of a gesture, this table won't be included in the final version, but just to show to JMO.

per class, is the dataset balanced

The dataset used in this work is provided by the ChaLearn LAP [37] gesture spotting challenge³. 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. 10. In terms of data, three modalities are 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).

5.2 Experimental protocol

5.2.1 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

3. <http://gesture.chalearn.org/2014-looking-at-people-challenge/data-2014-challenge>

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.

5.2.2 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:

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

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

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 ⁴. 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.

5.2.3 Tested systems

We evaluated the recognition performance made by the HMM applied to the emission probabilities estimated from either the skeleton data, the RGBD image data, the late fusion scheme, and the early 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.

4. Note that 'non gesture' frames are thus excluded from the counts.

Module	Validation	Test
Skeleton – DBDN	0.783	0.779
RGBD – 3DCNN	0.752	0.717
Multimodal Late Fusion	0.817	0.809
Multimodal Early Fusion	0.800	0.798

TABLE 2: Results in terms of Jaccard index JI for the different network structures and modalities modeling the emission probabilities.

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

TABLE 3: Gesture classification performance at the event level, in percentage of the number of ground truth gestures.

5.3 Results

JMO comment: *Confusion matrices are fine. However, would it be possible to know the classification accuracy of the different classes, for each modality? (i.e the diagonal elements of the confusion matrix? i.e. for instance to look at lowest/highest results? and it improves with fusion?*

Overall results. The overall performance of the algorithms are given in Tables 2 and 3. 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 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 [23]: 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 RGBD 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 3 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 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 Fig. 12 better illustrate the complementarity of the behaviors of the two modalities. The higher underdetection of RGB-D is clearly

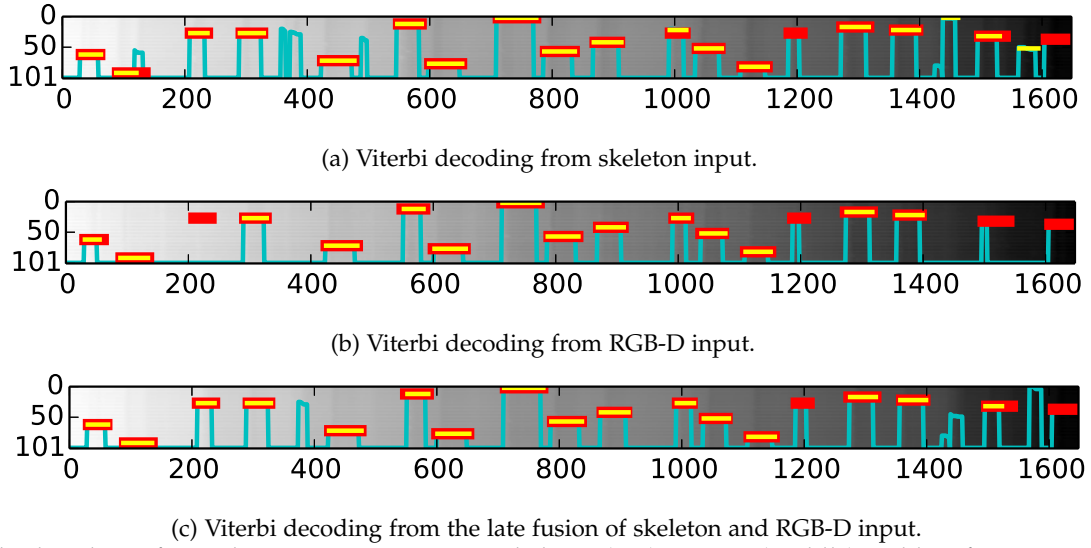


Fig. 11: Viterbi decoding of sample sequence #700, using skeleton (top), RGB-D (middle) and late fusion system (bottom). The x-axis represents time and the y-axis represents the hidden states of all classes and of the ergodic state (state 101). The cyan lines represent the viterbi shortest path, while 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.

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 for instance of the “Basta” gesture, whose arms motion resembles the start and end of the arm motion of many other gesture (see Fig. 10). Whatever the modality, it thus tends to be over-detected for all gesture classes, whenever the likelihood of the instances 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. 11. 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).

However, the modalities by themselves have more difficulties to correct the recognition errors which are due to variations coming from the performer, like differentiating people that gesticulate more (see Fig. 14).

Late vs. Early fusion.

The results in Table 2 and 3 show that the early 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 early fusion scheme to result in better e-

mission probability predictions, as compared to the simple score fusion in the late system. One possible explanation is that the independance 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 early 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 RGBD 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). Further investigation on how to better address this heterogeneity should be conducted.

HMM benefit. To investigate the benefit of temporal modelling of the proposed system, we compare the model that discards temporal modelling. Specially, once, we collect the emission probability $p(X_t|H_t)$, we average the hidden states for each gesture class and obtain the frame based gesture class prediction $p(G_t|X_t)$ where G_t denotes the gesture variables. After temporal smoothing for each frame, a threshold for cutting the begin and end of active gesture is chosen by a validation set.

To investigate the benefit of temporal modelling of the proposed system, we compare the model that discards temporal modelling. Specially, once, we collect the emission probability $p(X_t|H_t)$, we average the hidden states for each gesture class and obtain the frame based gesture class prediction $p(G_t|X_t)$ where G_t denotes the gesture variables. After temporal smoothing for each frame, a threshold for

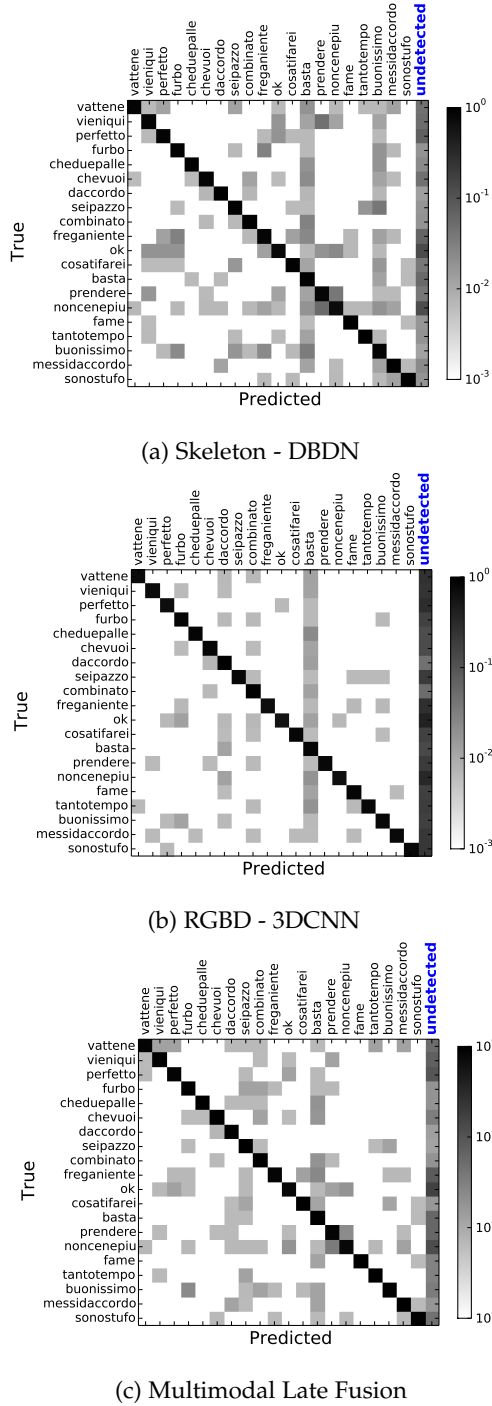


Fig. 12: Confusion Matrices (log-norm) for the different modalities.

cutting the begin and end of active gesture is chosen by a validation set. Fig. 13 illustrates the effect of temporal modelling. The decision boundary of where the gesture starts and ends in the proposed *DDNN* is clean-cut due to the state-diagram defined in Fig. 2. And the overall performance in terms of Jaccard index is reduced to 0.607 for test set. Note that, this representation learning template matching method using only 5 frames still outperforms the Jaccard index 0.413 using 5 frames template matching system in [52] where all the features are handcrafted.



(a) Well-recognised (b) Poorly-recognised

Fig. 14:

JMO comment: Show more sample to illustrate the dynamic - would be good to show the instances for the same gestures - the jaccard index below are obtained using which module?

Examples of performer variations in the upper body dynamic. While most performers tend to keep their upper-body static while performing the gesture leading to good recognition performance (left, Jaccard index of 0.94), others vehemently move their body when doing the gestures, making them harder to recognize (right, Jaccard index of 0.34).

Module	Skeleton	RGBD	Fusion
[39] 3 set skeletal & HOG, Boosted classifier	0.791	-	0.822
[53] 3D skeletal pose & HOG, MRF	0.795	-	0.827
[41] Dense trajectory	-	0.792	-
[52] Template based Random Forest Classifier	-	-	0.747
[54] Fisher Vector, Dynamic Programming	0.745	-	-
[55] Independent Subspace Analysis, RF	-	0.649	-
[56] PHOG, SVM, HMM	0.454	0.462	0.597
[38] Representation Learning (multiscal)	0.808	0.809	0.849
[32] CNN	-	0.789	-
[31] Deep Neural Networks	0.747	0.637	0.804
<i>DDNN</i> (this work)	0.779	0.717	0.809

TABLE 4: Comparison of results in terms of the ChaLearn Jaccard index with state-of-the-art related works.

Comparison with the state-of-the-art. The performance of recent state-of-the-art techniques is given in Table 4. The first half of the table resort

5.4 Computational Complexity

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

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 (GeForce GTX TITAN Black) and conv op. by Theano [57], 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 RGBD 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

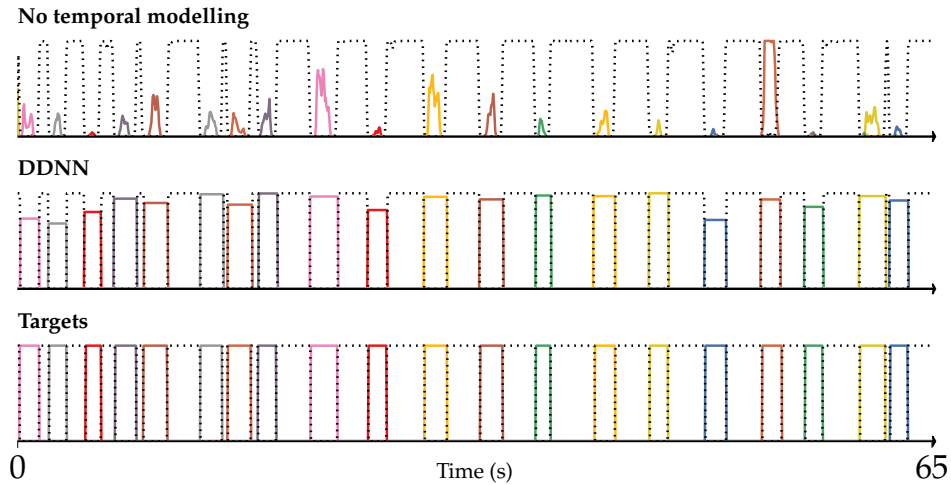


Fig. 13: The output probabilities are shown for a sequence fragment for the test set #881. The dashed line represents silence and different colour represent different gestures. Without temporal modelling, the decision boundary of a gesture will be more rugged and it is more difficult to make hard decisions of where the gesture starts or ends. Because of the temporal modelling and Viterbi path decoding, the gesture boundary will have a clean jump from the ergodic state to the gesture states, resembling the behavior of the manual annotators with better accuracy.

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.

6 CONCLUSION AND DISCUSSION

Hand-engineered, task-specific features are often less adaptive and time-consuming to design. This difficulty is more pronounced with multimodal data as the features have to relate with multiple data sources. In this paper, we presented a novel deep dynamic neural network (DDNN) framework that utilises deep belief networks and 3D convolutional neural networks for learning contextual frame-level representations and modelling emission probabilities for Markov fields. The heterogeneous inputs from skeletal joints, RGB and depth images require different feature learning methods and the late fusion scheme is adopted at the score level. The experimental results on bi-modal time series data show that the multimodal DDNN framework can learn a good model of the joint space of multiple sensory inputs, and is consistently as good as or better than the unimodal input, opening the door for exploring the complementary representation among multimodal inputs. It also suggests that learning features directly from data is a very important research direction and with more and more data and flops-free computational power, the learning-based methods are not only more generalisable to many domains, but also are powerful in combining with other well-studied probabilistic graphical models for modelling and reasoning

dynamic sequences. Future works include learning the share representation amongst the heterogeneous inputs in a more meaningful fashion as in [58] and integrate the overall system into a unified deep learning framework by replacing the temporal modelling part with Recurrent Neural Nets such as the work of [59].

APPENDIX A DETAILS OF THE CODE

The python code using for this work can be found at:
https://github.com/stevenwudi/chalearn2014_wudi_luo

ACKNOWLEDGMENTS

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

REFERENCES

- [1] L. Liu, L. Shao, F. Zheng, and X. Li, "Realistic action recognition via sparsely-constructed gaussian processes," *Pattern Recognition*, doi: 10.1016/j.patcog.2014.07.006., 2014.
- [2] L. Shao, X. Zhen, and X. Li, "Spatio-temporal laplacian pyramid coding for action recognition," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 817-827, 2014.
- [3] D. Wu and L. Shao, "Silhouette analysis-based action recognition via exploiting human poses," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 236-243, 2013.
- [4] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, 2005.
- [5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. IEEE, 2005.
- [6] G. Willems, T. Tuytelaars, and L. V. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *European Conference on Computer Vision*. Springer, 2008.
- [7] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *International Conference on Multimedia*. ACM, 2007.
- [8] A. Klaser, M. Marszalek, and C. Schmid, "A Spatio-Temporal Descriptor Based on 3D-Gradients," in *British Machine Vision Conference*, 2008.

- [9] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, 2013.
- [10] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid *et al.*, "Evaluation of local spatio-temporal features for action recognition," in *British Machine Vision Conference*, 2009.
- [11] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *European Conference on Computer Vision*. Springer, 2010.
- [12] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [13] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Spatio-temporal convolutional sparse auto-encoder for sequence classification," in *British Machine Vision Conference*, 2012.
- [14] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, 2006.
- [15] J. Schmidhuber, "Deep learning in neural networks: An overview," *arXiv preprint arXiv:1404.7828*, 2014.
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012.
- [17] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [18] M. Y. Shuiwang Ji, Wei Xu and K. Yu, "3d convolutional neural networks for human action recognition," in *International Conference on Machine Learning*. IEEE, 2010.
- [19] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2013.
- [20] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, 2012.
- [21] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [22] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deepspeech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.
- [23] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [24] J. Han, L. Shao, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1317-1333, 2013.
- [25] S. Escalera, J. Gonzalez, X. Bar, M. Reyes, O. Lops, I. Guyon, V. Athitsos, and H. J. Escalante, "Multi-modal gesture recognition challenge 2013: Dataset and results," in *ACM ChAlearn Multi-Modal Gesture Recognition Grand Challenge and Workshop*, 2013. [Online]. Available: <http://gesture.chalearn.org/>
- [26] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *ACM Computer Human Interaction*, 2012.
- [27] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. J. Escalante, "Chalearn gesture challenge: Design and first results," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2012.
- [28] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [29] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.
- [30] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.
- [31] D. Wu and L. Shao, "Deep dynamic neural networks for gesture segmentation and recognition," *European Conference on Computer Vision and Pattern Recognition Workshops*, 2014.
- [32] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *European Conference on Computer Vision and Pattern Recognition Workshops*, 2014.
- [33] S. Nowozin and J. Shotton, "Action points: A representation for low-latency online human action recognition," *Tech. Rep.*, 2012.
- [34] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 1521-1527.
- [35] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Advances in Neural Information Processing Systems*, 2012.
- [36] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *ECCV*. Springer, 2014.
- [37] S. Escalera, X. Bar, J. Gonzalez, M. Bautista, M. Madadi, M. Reyes, V. Ponce, H. Escalante, J. Shotton, and I. Guyon, "Chalearn looking at people challenge 2014: Dataset and results," in *European Conference on Computer Vision workshop*, 2014.
- [38] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *European Conference on Computer Vision and Pattern Recognition Workshops*, 2014.
- [39] C. Monnier, S. German, and A. Ost, "A multi-scale boosted detector for efficient and robust gesture recognition," in *European Conference on Computer Vision and Pattern Recognition Workshops*, 2014.
- [40] A. Yao, J. Gall, G. Fanelli, and L. J. Van Gool, "Does human action recognition benefit from pose estimation?," in *BMVC*, 2011.
- [41] X. Peng, L. Wang, and Z. Cai, "Action and gesture temporal spotting with super vector representation," in *European Conference on Computer Vision and Pattern Recognition Workshops*, 2014.
- [42] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [43] R. Salakhutdinov, "Learning deep generative models," Ph.D. dissertation, University of Toronto, 2009.
- [44] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*. Springer, 1998.
- [45] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3d discriminative skeletal features for human action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013.
- [46] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2006.
- [47] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation*, 2013.
- [48] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.
- [49] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2146-2153.
- [50] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.
- [51] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1139-1147.
- [52] N. C. Camgöz, A. A. Kindiroglu, and L. Akarun, "Gesture recognition using template based random forest classifiers," in *Computer Vision-ECCV 2014 Workshops*. Springer, 2014, pp. 579-594.
- [53] J. Y. Chang, "Nonparametric gesture labeling from multi-modal data," in *European Conference on Computer Vision and Pattern Recognition Workshops*, 2014.
- [54] G. D. Evangelidis, G. Singh, and R. Horaud, "Continuous gesture recognition from articulated poses," in *Computer Vision-ECCV 2014 Workshops*. Springer, 2014, pp. 595-607.
- [55] G. Chen, D. Clarke, M. Giuliani, A. Gaschler, D. Wu, D. Weikersdorfer, and A. Knoll, "Multi-modality gesture detection and recognition with un-supervision, randomization and discrimination," in *Computer Vision-ECCV 2014 Workshops*. Springer, 2014, pp. 608-622.

- [56] B. Liang and L. Zheng, "Multi-modal gesture recognition using skeletal joints and motion trail model," in *Computer Vision-ECCV 2014 Workshops*. Springer, 2014, pp. 623–638.
- [57] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," in *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*, 2012.
- [58] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Moddrop: adaptive multi-modal gesture recognition," *arXiv preprint arXiv:1501.00102*, 2014.
- [59] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 855–868, 2009.
- [60] D. Wu and L. Shao, "Multimodal dynamic networks for gesture recognition," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 945–948.
- [61] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

Response to Associate Editor

Comment: *Reviews of the paper have been received. In general comments are positive. Still one reviewer suggests some interesting extra analyses of the proposed method. Please carefully address all reviewers comments for a second review round of the paper. Please provide your revision by July 31th.*

Response: We would like to thank the associate editor's general positive comments on our paper. We also would like to thank reviewers' careful and insightful suggestions for improving the paper. Following are some common points that were mentioned by the reviewers and we list the most notable changes below. The point to point responses are provided afterwards.

- In the previous version for multimodal fusion, we used the late fusion scheme $s = a * s_1 + (1 - a) * s_2$ where a is chosen by cross-validation. In this revision, we implement an early fusion scheme in Section 4.4.2 that a new-top level perceptron layer is created to combine two models' output as in Fig. 9. The new multimodal neural network's parameters are initialised by the previously trained individual module, taking advantage of different modules intrinsic properties and making the network converge much faster. The early fusion system uses pre-trained weights. The results are reported in Section 4.4.2 and Table 2.
- The Related Works section has been moved after the Introduction section. We also follow reviewers' suggestions and include discussions of related works concerning works of: 1) exploiting temporal models in the context of gesture recognition; 2) literature for RGBD data using deep learning - "Wang *et al.* [34] introduced a discriminative hidden-state approach for the recognition of human gestures. However, their discriminative training is limited for pre-segmented gesture sequences and one layer of hidden states might not learn powerful enough high level representations for a larger corpus." "In the field of deep learning from RGBD data, Socher *et al.* [35] proposed a single convolutional neural net layer for each modality as input 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. The pooled filter responses are then given to a recursive neural network to learn compositional features and part interactions. Gupta *et al.* [36] 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. The depth image was represented by three channels: horizontal disparity, height above ground and angle with gravity. This augmented representation allows the CNN to learn strong features than by using disparity (or depth) alone."
- Experimental analysis. We have included some time analysis in Section 5.4 and visualisation of response maps after learnt filters in Fig 8.
- Explanation of intuition behind higher level presentation of the skeleton features. We include Section 3.3 to explain the intuition behind higher level representation for skeleton joint features which appeared in our previous CVPR paper but was not included in the previous submission. We think this part is one of major contributions of the paper and inclusion of this section makes the journal paper more self-contained.

Response to Reviewer 1

We thank the reviewer for his time and comments and positive appreciation of the paper. Below, we provide our answers to his comments and to the unclear points that were raised, and what we have done to clarify the paper and take comments into account. Note that in our answer, all references (Equations, Figures) refer to the new version unless stated otherwise.

Comment: *The article is easy to read and well structured. The methodology is not strictly novel but its application in the gesture domain with the multimodal fusion makes the article worth reading. Although results are arguably a little behind the maximum performance ones the overall impression of the article is favorable and I believe the community may benefit to check the ideas included in this paper.*

The article proposes a framework for dynamic data augmenting a HMM with deep learning techniques and apply this to gesture segmentation and recognition. Gestures segmentation and recognition is a difficult problem. The article tackles this difficulty by means of pure data driven approaches similar to the ones used for speech recognition. The particularities of the computer vision domain are handled accordingly.

Response: Thank you for your review and positive outlook of the paper. We are also aware that our results are arguably a little behind the maximum performance, and this may be due to the network initialisation and multimodal neural network learning. We have included extra experimental analysis and early fusion implementation to further extend the broadness of the paper.

Response to Reviewer 2

We thank the reviewer for his time and comments. Note that the article has been reworked significantly to better introduce and enhance the modeling elements. Below, we provide our specific answers to his comments and to the unclear points that were raised, and what we have done to clarify the paper and take comments into account. Note that in our answer, all references (Equations, Figures) refer to the new version unless stated otherwise.

Comment: *In general, the manuscript is well written and is easy to follow. In the given case, it would be preferable to have the "Related work" section right after the introduction, as otherwise paper's contributions are not completely clear. Furthermore, there is certainly a vast literature on exploiting HMMs in the context of gesture recognition (as well as other temporal models, such as recurrent neural networks), which should be briefly summarized, the differences with the proposed solution should be highlighted.*

Response: Thank you for your comments and the recognition of easy readability of the paper. We have moved the related work section after the introduction section. Moreover, we have included the discussions of literature that utilise temporal models, e.g., "Wang *et al.* [34] introduced a discriminative hidden-state approach for the recognition of human gestures. However, their discriminative training is limited for pre-segmented gesture sequences and one layer of hidden states might not learn powerful enough high level representations for a larger corpus", "[35] proposed a single convolutional neural net layer for each modality as input to multiple, fixed-tree RNNs in order to compose higher order features for 3D object classification."

Comment: *Authors claim to learn a model in the joint multi-modal space is a slight overstatement, as neural networks processing different modalities are trained completely independently with following averaging of produced scores.*

Response: Thank you for your comments. In this revision, we implement the early fusion scheme in Section 4.4.2 that "we adopt another layer of perceptron for cross modality learning taking the input from each individual net's penultimate layer. The parameters of two neural networks (for skeleton and depth) are loaded from the previously trained individual module...The results for the early fusion system are reported in Tab. 2. The fusion network is initialised by the pre-trained model and stacked with one hidden layer with 2024 hidden unites. We fine-tune the network and stop the training when the validation error rate stops decreasing (~15 epochs)...However, we can see from Tab. 2 that the early fusion system does not outperform the late fusion system. The result is counter-intuitive because we expect that the early fusion multimodal feature learning would extract semantically meaningful shared representations, outperforming individual modalities, and the early fusion schemes efficacy against the traditional method of late fusion [60]. One possible explanation could be that one individual module has dominant effect on the learning process so as to screw the network towards learning that specific module. The mean activations of the neurons for each module in Fig. 9 indicate the aforementioned conjecture."

Comment: *State of the art in the experimental section should be mentioned more consistently. For fair comparison, first three lines in Table 3 should be: [39] Deep learning (step 4): skeleton 0.7891, video 0.7990, fusion 0.8449 [39] Deep learning (multiscale): skeleton 0.8080, video 0.8096, fusion 0.8488 [40] 3 sets of skeletal features and HoG: skeleton 0.791, fusion 0.8220 Therefore, it shows that both learning-based and feature extraction-based approaches outperform the proposed method on each modality, as well as on a combination of them. Furthermore, it would be interesting to see how the HMM contributes in the performance in comparison with simple voting based on frame-based predictions.*

Response: Thank you for your detailed comments. We have amended the results table accordingly. The less than maximum performance could be due to the less than ideal settings and initialisations of the neural network. Nonetheless, we would like to argue that one major contribution of the paper is using the learning method for feature extraction and the utilisation of HMM for simultaneous gesture segmentation and recognition. We also present some brief analysis of why the fusion network didn't achieve expected performance gain and

hope the experimental analysis could cast some light on the future research directions of the related problems.

Comment: Visualization of the filter banks (section 3.3.4) in its current state is unnecessary as it does not provide any interesting insights on the interpretation of the learned features. Instead, the poorly formed filters rather indicate undertraining, or lack of training data given the model complexity, or suboptimality of training procedure.

Response: We include the response maps after filtering for both body and hand parts. We observe some interesting properties from the visualisation of the filter banks as in [35] that “Depth images have sharper edges and generally are smoother than the grayscale filters, though the distinctions are less obvious compared with the body versus hand filters.”

Response to Reviewer 3

Comment: In general, I would be more excited if shared representations were learned from the skeleton and the RGB data, as done in multimodal deep learning. This is left for future work. On the positive side, the CNN and DBN are technically sound and the results from their fusion are interesting. One would expect that the journal version of the paper would be more self-contained and easier to follow than the conference versions, but here I observe the opposite trend. For example, the older conference version [21] explains the intuition behind the higher level representation of the skeleton features, but the journal version does not. The conference paper explains how the coordinate frames are built for the features, while this paper skips this part. The conference paper explains the datasets and visualizes the Viterbi paths better.

Response: Thank you for your careful and positive review. We agree that in this journal version of the paper, some self-contained information has been omitted from the conference paper. In this revision, we have included more details. Specifically, we include the Problem formation in Sec 3.3 that explains the intuition behind the higher level representation and the advantages offered by feed forward neural over GMMs. The pre-training step is of crucial importance in learning the right initialisation of the deep belief networks using the sigmoid activation function.

Comment: Section 2 does not help much the reader understand the formulation. For example: “At each time step, we have one observed random variable X_t : explain what these variables represent early (raw skeleton input / RGBD) we have an unobserved variable H_t : describe at a high level the information that the unobserved variables capture, mention examples

Response: We have included the interpretation part as follows: “A continuous-observation HMM with discrete hidden states is adopted for modelling higher level temporal relationships. At each time step t , we have one observed random variable X_t which can be the skeleton input or depth/RGB input. The unobserved variable H_t taking on values in a finite set $\mathcal{H} = (\bigcup_{a \in \mathcal{A}} \mathcal{H}_a)$, where \mathcal{H}_a is a set of states associated with an individual gesture a by force-alignment. The unobserved variable H_t can be interpreted as a segment of an action a . For example, for action sequence “tennis serving”, the action sequence can be dissected into $\mathcal{H}_{a_1}, \mathcal{H}_{a_2}, \mathcal{H}_{a_3}$ as: 1) raising one arm, 2) raising the racket, and 3) hitting the ball.”

Comment: The related work section is out of place after the technical sections and before the experiments.

Response: We have moved the related work section after the introduction section.

Comment: There is no point writing a loop for $m=1:2$ in Algorithm 1 and 2.

Response: We have rewritten Algorithms 1 and 2 and merge the algorithms into a more succinct format.

Comment: “the number of states ... is chosen as 5”: any intuition here?

Response: Thank you for the comment and this is a very good observation. The number of hidden states chosen uniformly as 5 in the paper might not be the optimal way of setting the number of hidden states for each gesture. We also experimented segmenting gestures into 10 states and obtained similar results. We reduce the hidden states from 10 to 5 in order to reduce the number of predicting classes and avoiding overfitting. The interpretation of choosing the number of hidden states for Markov Model is similar to choosing the number of hidden states for neural networks: it’s more heuristically based. Ideally, we could set the number of hidden states according to the average length of the gesture sequence. But due to time constraint, we didn’t train such neural networks.

Comment: "10 frames are assigned to hidden state ...": why 10?

Response: Thank you for the careful observation. This is actually a written error due to different number of hidden states used for our experiments. We rewrote the section as: "**Hidden states (\mathcal{H}_a):** Force alignment is used to extract the hidden states, i.e. if a gesture token is 100 frames, the first $20 = \frac{100}{5(N_{\mathcal{H}_a})}$ frames are assigned to hidden state 1, the following 20 frames are assigned to hidden state 2, and so forth."

Comment: it is hard to interpret the learned features on Figure 8. There is no intuition what the depth filters capture.

Response: Because our filter size is 5×5 (smaller filter sizes tend to generalize better, [61] used 3×3 convolution filters), it will be hard to interpret. However, we observe the similar effect as in paper [35] for depth image filters and we include the following analysis: "Visualisation of the 5×5 filters in the first layer for the different input channels. Interestingly, we observe the same effect as [35] that the resulting filters from depth images have sharper edges which arise due to the strong discontinuities at object boundaries. While the depth channel is often quite noisy, most of the features are still smooth."

Comment: Citations that could be added in the context of deep learning from RGBD data: "Convolutional-Recursive Deep Learning for 3D Object Classification", Socher et al., NIPS 2012

Response: Thank you for the suggested citation. And we now include in the related works section as follows: "In the field of deep learning from RGBD data, Socher et al. [35] proposed a single convolutional neural net layer for each modality as input 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. The pooled filter responses are then given to a recursive neural network to learn compositional features and part interactions."

We also observe the same CNN filters as the paper that "one interesting result is that depth channel edges are much sharper. This is due to the large discontinuities between object boundaries and background. While the depth channel is often quite noisy, most of the features are still smooth".

Comment: "Learning Rich Features from RGB-D Images for Object Detection and Segmentation", Gupta et al., ECCV 2014

Response: We find the works in Gupta et al. [36] interesting in a sense that CNN does not necessarily need to be trained from the raw images, and some handcrafted features may better help the network to learn more meaningful, higher level representations. And we have included this related work as follows. "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. The depth image was represented by three channels: horizontal disparity, height above ground and angle with gravity. This augmented representation allows the CNN to learn strong features than by using disparity (or depth) alone."

Comment: Another related work is the "Multimodal Deep Learning" by Ngiam et al., ICML 11. I would also like to see some discussion wrt "Hidden Conditional Random Fields for Gesture Recognition", Wang et al, CVPR 2006

Response: Thank you for suggesting very relevant works. We came across both papers. "Multimodal Deep Learning" essentially is the prototype for an early fusion model. Wang et al. [34] observed that one hidden layer is limited for learning a larger class corpus. Feature learning for skeleton modules is an essential part of this paper and we believe a higher level representation is more beneficial for gesture classification. Moreover, the partition function of CRF makes the discriminative training more difficult to train. The similarity in the aforementioned paper with our proposed method is that both methods used a hidden layer for learning higher level representations.

Recent advancement in feature learning and pre-training for DBN renders our proposed method more meaningful. We have included the following in the related works section: "Wang et al. [34] introduced a discriminative hidden-state approach for the recognition of human gestures. However, their discriminative training is limited for pre-segmented gesture sequences and one layer of hidden states might not learn powerful enough high level representation for a larger corpus."

Response to Reviewer 4

Comment: Late fusion: my greatest technical concern is that two deep models are trained and then combined with a weighted average: $s = a * s1 + (1-a)*s2$ where a is chosen by cross-validation. Instead, the authors could combine the two models by creating a new top-level perceptron layer which takes the two models as input. Then this whole structure could be trained jointly with back-propagation. I'd expect results to be (1) at least as good and (2) more philosophically unified.

Response: We agree with your insightful observation and continue experimenting with the early fusion scheme in this revision. Our previous paper [60] utilized the early fusion scheme for audio and skeleton modules for action recognition. We followed that strategy and perform the early fusion scheme using penultimate layer in Section 4.4.2. "However, we can see from Table 2 that the early fusion system does not outperform the late fusion system. The result is counter-intuitive because we expect that the early fusion multimodal feature learning would extract semantically meaningful shared representations, outperforming individual modalities, and the early fusion schemes efficacy against the traditional method of late fusion [60]. One possible explanation could be that one individual module has the dominant effect on the learning process so as to screw the network towards learning that specific module. The mean activations of the neurons for each modules in Fig. 9 indicate the aforementioned conjecture."

Comment: The analysis is a bit brief. More experiments and ablative analysis could be added. Specifically, can we interpret the failure patterns of the proposed model(s) and prior work? It would be interesting to see statements like [40] fails more often on gestures of X kind because HOG erases Y useful information or [39] does worse for Z because it handles time at an earlier stage of the pipeline. Then, also giving some qualitative examples of these failures.

Response: We agree that there is a lack of experiments analysis, especially the failure patterns and lessons learnt from the experiments. We have included more analysis in the Experiment and Analysis section as follows: "Examples of overall upper body movements influence on the system performance. Left (score: 0.94) performer almost kept a static upper body whilst performing Italian sign language. Right (score: 0.34) performer moved vehemently when performing the gestures.14"

Comment: These extra experiments (considering joint training of a combined emission probability model) and qualitative interpretation could significantly affect the paper. Overall, the research is solid but needs significantly more work before publication. RCNN: Last, it is entirely possible to train a recurrent neural network to perform Viterbi decoding. This may be difficult (requiring more training data) but would make the entire paper fit into a the deep learning framework. I cannot hold this against the authors, but some discussion might help.

Response: We have included more qualitative interpretation of the results. We agree that a recurrent neural network could potentially replace the Viterbi decoding part to make the system as a more unified end-to-end system. This, however, may be left to the future work.

Comment: They use a 3d convolutional network. While the introduction makes it sound like this is for multiple-channels (e.g. RGB + Depth), sec. 3.3.2 makes it clear the 3rd dimension is time as the model processes 4 frame sub-sequences. I think, Fig. 6 could be clearer.

Response: Thank you for your detailed observation. Yes, the 3rd dimension of the input network is indeed the time axis. However, RGB and Depth data are treated as the two channels during the input phase. We detailed the description of the Figure as follows. "The 3rd dimension of the input is time with 4 frames stacked together. The depth and RGB data are stacked (concatenated) together at Input. Hand and body part outputs are concatenated at H7."

Comment: Using RGB-D with deep learning is a common idea, explored by many concurrent works e.g. [A,B,C]. [A] Dosovitskiy, A., Springenberg, J.

T., Riedmiller, M., & Brox, T. (2014). Discriminative Unsupervised Feature Learning with Convolutional Neural Networks. *arXiv Preprint arXiv: 113*.
 [B] Gupta, S., Girshick, R., Arbelaz, P., & Malik, J. (2014). Learning Rich Features from RGB-D Images for Object Detection and Segmentation. *arXiv Preprint arXiv:1407.5736*, 116. doi:10.1007/978-3-319-10584-0-23
 [C] Socher, R., Huval, B., Bhat, B., Manning, C. D., & Ng, A. Y. (2012). Convolutional-Recursive Deep Learning for 3D Object Classification. *Advances in Neural Information Processing Systems*

Response: Thank you for suggesting related works. We find the works in Gupta *et al.* [36] interesting in a sense that CNN does not necessarily need to train from the raw images, and some handcrafted features may better help the network to learn more meaningful, higher level representations. We have included the papers using deep learning for RGB-D data with discussions in the related works section: "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. The depth image was represented by three channels: horizontal disparity, height above ground and angle with gravity. This augmented representation allows the CNN to learn strong features than by using disparity (or depth) alone." "In the field of deep learning from RGBD data, Socher *et al.* proposed a single convolutional neural net layer for each modality as input 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. The pooled filter responses are then given to a recursive neural network to learn compositional features and part interactions."