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 labelling videos in a frame-by-frame The key contributions can be summarised as follows:

- A Gaussian-Bernoulli Deep Belief Network is proposed to extracts 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, RGB images;
- The proposed temporal framework labels a video sequence in a frame-to-frame mechanism, rendering it possible for online gesture segmentation and recognition.
- 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, their model relying on only one layer of hidden states might not alone 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 almost will certainly further improve the performance [17].

However, with advent of Kinect has put more emphasis on RGB-D data. For instance, the benefits of deep Learning using RGB-D data have been explored for object detection or claudication 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. For addressing 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.

The gesture recognition doamin has been stimulated by the collection of large public corpus. In particular, the ChaLearn LAP [37] gesture spotting challenge has collected than 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, RGBD input, or both. For instance, Neveroa *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 recognition in videos. Neveroa *et al.* [38] presents a multi-

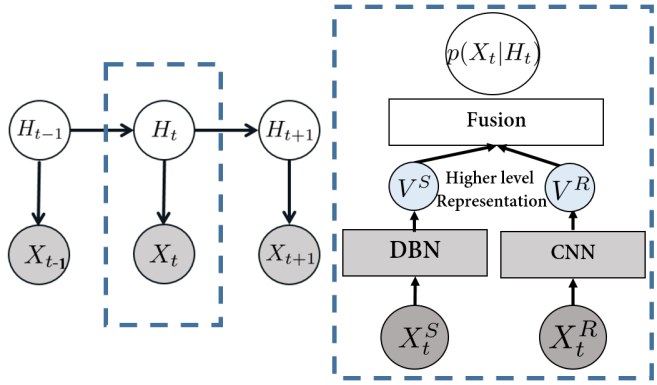


Fig. 1: Per-gesture model: the temporal model is a HMM (left), whose emission probability $p(X_t|H_t)$ (right) is modeled by a forward-linked chain. The HMM observations X_t (skeletal features X_t^s, or RGB-D image features X_t^r) are first passed through deep neural nets: a Deep Belief Network (DBN) for skeletal modality or a 3D convolutional neural network (3DCNN) for the RGB-D modality, to extract high-level features (V^s and V^r). The outputs of the neural networks are the emission probabilities of the hidden states $p(X_t|H_t)$.

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 strongest to weakest cross-modality structure. One major difference between our proposed method and their works is the treatment of the time factor: fixed length of frames are served as the input of their neural networks for the prediction of the final gesture class. To cope with gestures performed at different speeds, several multi-scale networks are trained. Moreover, the skeleton modules used in their network are sets of ad-hoc hand crafted features.

3 Model Formulation

Inspired by the framework successfully applied to speech recognition [20], the proposed model is a data driven learning system, relying on a pure learning approach. 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, where the emission probabilities are modelled by two distinctive types of neural networks appropriate for input modality. More specifically, the first model works on skeletal features and the neural network for the emission probabilities is a deep boltzmann machine. The second model, on the other hand, uses convolutional neural networks to model the emission probabilities related to RGB and depth (RGBD) video data. In the remainder of this section, we will first present our temporal model and then introduce its main component. Details of two distinct neural networks and fusion mechanisms along with post-processing will be provided in Section 4.

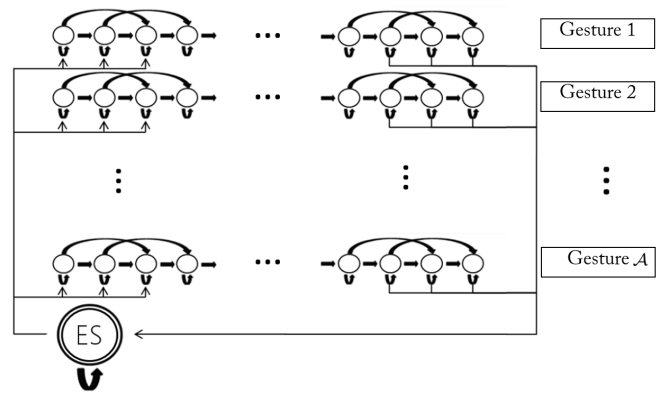


Fig. 2: State diagram of the *ES-HMM* model for low-latency gesture segmentation and recognition. An ergodic state ($\mathcal{ES}$) is used to model the resting position between gesture sequences. Each node represents a single state and each row represents a single gesture model. The arrows indicate 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 represents 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 taking on values in a finite set composed of $\mathcal{H}= (\bigcup_{a\in\mathcal{A}} \mathcal{H}_a)$, where $\mathcal{H}_a$ is a set of states associated with an individual gesture $\boldsymbol{a}$ related to the different gesture. The unobserved variable H_t can be interpreted as a segment of a gesture $\boldsymbol{a}$.

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:

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

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 probability and $p(X_t|H_t)$ is the emission probability modelled by the deep neural nets.

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 state $\mathcal{H}_a$ is introduced to defined a Markov model of that gesture. For example, for action sequence "tennis serving", the action sequence can 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).

Further more, to alow segmentation of gesture, we add an ergodic state ($\mathcal{ES}$) which resembles the silence state for speech recognition which serve as a catch-all state. From the $\mathcal{ES}$ 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 the $\mathcal{ES}$. Hence, the hidden variable H_t can take values within the finite set, defined as $\mathcal{H} = (\bigcup_{a\in\mathcal{A}} \mathcal{H}_a) \bigcup \{\mathcal{ES}\}$, where $\mathcal{ES}$ is the ergodic state as the resting position between gestures. We refer to the model as the ergodic states hidden Markov model (*ES-HMM*) for simultaneously gesture segmentation and recognition.

The *ES-HMM* framework 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 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 this 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 $h_{t:T}$ 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. 12.

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

Traditionally, GMMs and HMMs co-evolved as a way of doing speech recognition when computers were too slow to explore more computationally intensive approaches. GMMs are easy to fit when they have diagonal covariance matrices and, with enough components, they can model any distribution. They are, however, statistically inefficient at modeling high-dimensional data that has many kind of componential structure as explained in [20]. Suppose, for example, that $\mathcal{N}$ significantly different patterns can occur in one sub-band and $\mathcal{M}$ significantly different patterns can occur in another sub-band. Suppose also that which pattern occurs in each sub-band is approximately independent. A GMM requires $\mathcal{N} * \mathcal{M}$ components to model this structure because each component must generate both sub-bands(each piece of data has only a single latent cause). On the other hand, a model that explains the data using multiple causes only requires $\mathcal{N} + \mathcal{M}$ components, each of which is specific to a particular sub-band. This exponential inefficiency of GMMs for modeling factorial structures leads to the GMMs+HMMs systems that have a very large number of Gaussians, most of which must be estimated from a very small fraction of the data.

The benefit of learning a generative model is greatly magnified when there is a large supply of unlabeled skeletal, RGB and depth data either acquired by motion capture systems or inferred from depth images in addition to the training data that has been labeled by a forced HMM alignment. We do not make use of unlabeled data in this paper, but it could only improve our results relative to purely discriminatively approaches.

Naturally, many of the high-level features learned by the generative model may be irrelevant for making the required discriminations, even though they are important for explaining the input data. However, this is a price worth paying if computation is cheap and high-level features are very good for discriminating between classes of interest. The benefit of each weight in a neural network being constrained by a larger faction of training case than each parameter in a GMM has been masked by other differences in training. Neural networks have traditionally been training discriminatively, whereas GMMs are typically trained as generative models (even if discriminative training is performed later in the training procedure). Generative training allows the data to impose many more bits of constraints on the parameters, hence partially compensating for the fact that each component of a large GMM must be trained on a very small fraction of the data.

Feed forward neural networks offer several potential advantages over GMMs:

- Their estimation of the posterior probabilities of HMM states does not require detailed assumptions about the data distribution.
- They allow an easy way of combining 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.

Learning the higher level representation for skeleton joints features:
Neal and Hinton [43] demonstrated that the negative log probability of a single data vector, $\mathbf{v}^0$, under the multi-layer generative model is bounded by a variational free energy, which is the expected energy under the approximating distribution, $Q(\mathbf{h}^0|\mathbf{v}^0)$, minus the entropy of that distribution. For a directed model, the "energy" of the configuration $\mathbf{v}^0$,$\mathbf{h}^0$ is given by $E(\mathbf{v}^0, \mathbf{h}^0) = -[\log p(\mathbf{h}^0) + \log p(\mathbf{v}^0|\mathbf{h}^0)]$. So the bound is

$$\log p(\mathbf{v}^0) \geqslant \sum_{\mathbf{h}^0} Q(\mathbf{h}^0|\mathbf{v}^0)[\log p(\mathbf{h}^0) + \log p(\mathbf{v}^0|\mathbf{h}^0)] - \sum_{\mathbf{h}^0} Q(\mathbf{h}^0|\mathbf{v}^0) \log Q(\mathbf{h}^0|\mathbf{v}^0)$$

The intuition using deep belief networks for modeling marginal distribution $p(X_t|H_t)$ in skeleton joints action recognition is that by constructing multi-layer networks, semantically meaningful high level features for skeleton configuration will be extracted whilst learning the parametric prior of human pose from mass pool of skeleton joints data. In the recent work of [44] a non-parametric bayesian network is adopted for human pose prior estima-

tion, whereas in our framework, the parametric networks are incorporated.

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 target frame class. E.g., for "toss" action, wrist joints is rotating around shoulder joints would be extracted from the backpropagation via target frame as those task specific, *ad hoc* hard wired sets of joints configurations as in [45] [46] [33] [47].

The outputs of the neural net are the hidden states learned by force alignment during the supervised training process. Once we have model, we can use the normal online or offline smoothing, inferring the hidden marginal distributions of every node (frame) of the test video.

The overall algorithm for training and testing are presented in Algorithm 1 and 2.

Algorithm 1: Multimodal deep dynamic networks – training

Data:
x_i=$[x_{i,1}, \ldots, x_{i,t}, \ldots, x_{i,T}]$
$\mathbf{X^1} = \{\mathbf{x_i^1}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - raw input (skeletal) feature sequence.
$\mathbf{X^2} = \{\mathbf{x_i^2}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - raw input (depth) feature sequence in the form of $M_1 \times M_2 \times T$, where M_1, M_2 are the height and width of the input image and T is the number of contiguous frames of the spatio-temporal cuboid.
$\mathbf{Y} = \{\mathbf{y_i}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - frame based local label (achieved by semi-supervised forced-aligment), where $\mathbf{y_i} \in \{N_{\mathcal{A}} * N_{\mathcal{H}_a} + \boldsymbol{1}\}$ with $N_{\mathcal{A}}$ the number of gesture classes and $N_{\mathcal{H}_a}$ is the number of states associated to an individual gesture a and $\boldsymbol{1}$ as ergodic state.

1 Preprocess skeletal data $\mathbf{X^1}$ as in Eq.2, 3, 4.
2 Normalise (zero mean, unit variance per dimension) the above features and feed it to Eq.12.
3 Pre-train the networks using *Contrastive Divergence*.
4 Supervised fine-tuning of the deep belief networks using $\mathbf{Y}$ by standard mini-batch *SGD* backpropagation.
5 Preprocess the depth and RGB data $\mathbf{X^2}$ as in 4.3.1.
6 Feed the above features to Eq.15.
7 Train the 3D convolutional neural networks using $\mathbf{Y}$.

Result:
GDBN - a Gaussian-Bernoulli visible layer deep belief network to generate the emission probabilities for the hidden Markov model.
3DCNN - a 3D deep convolutional neural network to generate the emission probabilities for the hidden Markov model.
$\mathbf{p(X_t|H_t)}$ emission probability.
$\mathbf{p(H_1)}$ - prior probability for $\mathbf{Y}$ by accumulating and normalising labels.
$\mathbf{p(H_t|H_{t-1})}$ - transition probability for $\mathbf{Y}$, enforcing the beginning and ending of a sequence can only start from the first or the last state.

Algorithm 2: Multimodal deep dynamic networks – testing

Data:
$\mathbf{X^1} = \{\mathbf{x_i^1}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - raw input (skeletal) feature sequence.
$\mathbf{X^2} = \{\mathbf{x_i^2}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - raw input (depth) feature sequencein the form of $M_1 \times M_2 \times T$.
GDBN - trained Gaussian-Bernoulli visible layer deep belief network to generate the emission probabilities for the hidden Markov model.
3DCNN - trained 3D deep convolutional neural network to generate the emission probabilities for the hidden Markov model.
$\mathbf{p(H_1)}$ - prior probability for $\mathbf{Y}$.
$\mathbf{p(H_t|H_{t-1})}$ - transition probability for $\mathbf{Y}$.

1 Preprocessing and normalising the skeletal data $\mathbf{X^1}$ as in Eq. 2, 3, 4.
2 Feedforwarding network **GDBN** to generate the emission probability $\mathbf{p(X_t|H_t)}$ in Eq.1.
3 Generating the score probability matrix $\mathbf{S^1} = \mathbf{p(H_{1:T}, X_{1:T})}$.
4 Preprocessing(median filtering the depth data) and normalising data RGBD data $\mathbf{X^2}$.
5 Feedforwarding **3DCNN** to generate the emission probability $\mathbf{S^2} = \mathbf{p(X_t|H_t)}$ in Eq.1.
6 Generating the score probability matrix $\mathbf{S^2} = \mathbf{p(H_{1:T}, X_{1:T})}$.
7 Fuse the score matrix $\mathbf{S} = \alpha * \mathbf{S^1} + (\mathbf{1} - \alpha) * \mathbf{S^2}$ OR the learnt joint representation.
8 Finding the best path $\mathbf{V_{t,\mathcal{H}}}$ using $\mathbf{S}$ by Viterbi decoding as in Eq.**??**.

Result:
$\mathbf{Y} = \{\mathbf{y_i}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - frame based local label
$\mathbf{C}$ - global label, the anchor point is chosen as the middle state frame.

4 Model Implementation

In this section, we detail the proposed Deep Dynamic Neural Networks and its individual composition implementations.

4.1 Ergodic States Hidden Markov Model

In all our experiments, the different modelling elements are specified as follows: the number of states $N_{\mathcal{H}_a}$ associated to an individual gesture has been set to 5. Therefore, in total, the number of $N_{\mathcal{H}}$= $20(classes) \times 5 + 1 = 101$ as experiments are conducted on the Chalearn dataset. The training data in Chalearn is given as a set of sequences x_i=$[x_{i,1}, \ldots, x_{i,t}, \ldots, x_{i,T}]$ where $x_{i,t} = [x^s_{i,t}, x^r_{i,t}]$ corresponds to the skeleton and RGB-D input. As the gesture label is provided for each sequence, we need to define y_i=$[y_{i,1}, \ldots, y_{i,t}, \ldots, y_{i,T}]$, 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 ith sequence is a gesture $\boldsymbol{a}$, then the first $\lfloor \frac{T_i}{5} \rfloor$ frames are assigned to state h^1_a (the first state of gesture $\boldsymbol{a}$), the following $\lfloor \frac{T_i}{5} \rfloor$ frames are assigned to h^2_a, and so forth. Note that each gesture sequence comes with the video frames proceeding and following the gesture. In

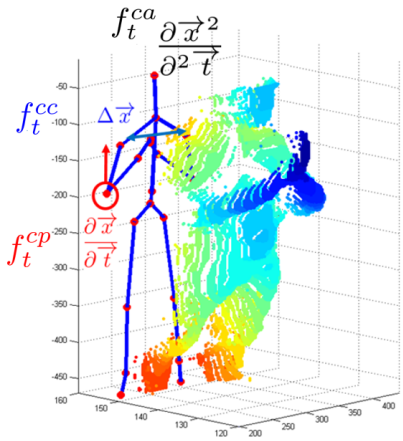


Fig. 3: A point cloud projection of a depth image and the 3D positional features.

practice, we dictate 5 frames before and after each gesture sequence and labelled this shot sequence with the ergodic state ($\mathcal{ES}$) label. Transitional matrix $p(H_t|H_{t-1})$ is collected from the statistics of labelling stage, allowing 5 frame jumps to accommodate skipping states.

4.2 Skeleton Module

Given our task, only upper body joints are relevant. Accordingly, only the N_j= 11 upper body joints are considered, namely *"ElbowLeft, WristLeft, ShoulderLeft, HandLeft, ElbowRight, WristRight, ShoulderRight, HandRight, Head, Spine, HipCenter"*.

Skeleton features of time t are defined as $x_t^s = x_t^{s,1}, \ldots, x_t^{s,N_j}$. To capture the gesture dynamics, rather than using x_t^s as raw input to our data drive approach, we follow the features used in [21] and compute 3D positional pairwise differences of joints as well as temporal derivatives, defined as (as shown in Fig. 3) [1]:

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

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

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

This results in an input feature $f_t = [f_t^{cc}, f_t^{cp}, f_t^{ca}]$ of dimension $N_f = N_j \times (\frac{N_j}{2} + N_j + N_j) * 3 = 891$. 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 data-set specific tuning. Note that feature extraction process resembles the computation of *Mel Frequency Cepstral Coefficients (MFCCs)* and their temporal derivatives for the speech recognition community [20].

4.2.1 Modeling X_t^s using Deep Belief Networks

Once we have the skeleton input feature f_t 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. We briefly introduce the building elements of the network and a more detailed introduction can be found at [48]:

Boltzmann Machines (BMs) are a special structure of Markov Random Field (MRF), *i.e.* the energy function is linear in term of its corresponding free parameters. To empower the expressiveness of the model so as to encode complex distributions, the hidden variables are introduced to enhance the modelling capability of the Boltzmann Machines

Restricted Boltzmann Machines (RBMs) is a subtype of BMs in that there is no connections between visible to visible or hidden to hidden variables. RBM, as a special type of Markov random field with a two-layer structure, has the visible binary stochastic units $v \in \{0,1\}^D$ connected to the hidden binary stochastic units $h \in \{0,1\}^F$.

The energy of the state $\{v,h\}$is:

$$E(v,h;\theta) = -v^\top W h - b^\top v - a^\top h \quad (5)$$

$$= -\sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} v_i h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} a_j h_j \quad (6)$$

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 joint distribution over the visible and hidden units is defined by:

$$P(v,h;\theta) = \frac{1}{Z(\theta)} \exp(-E(v,h;\theta)); \quad (7)$$

$$Z(\theta) = \sum_v \sum_h exp(-E(v,h;\theta)) \quad (8)$$

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

$$P(h_{j=1}|\mathbf{v}) = g(\sum_i W_{ij} v_i + a_j)); \quad (9)$$

$$P(v_{i=1}|\mathbf{h}) = g(\sum_j W_{ij} h_j + b_i)) \quad (10)$$

where $g(x) = \frac{1}{1+exp(-x)}$ is the logistic function.

The derivative of the log-likelihood with respect to the model parameter from Eq. 8 is expressed as: $E_{P_{data}}[vh^\top] - E_{P_{model}}[vh^\top]$ where E denotes the expectation. Due to the intractability of the second term, an approximation is generally required. This approximation is called the "Constrative Divergence":

$$\Delta W = \alpha(E_{P_{data}}[\mathbf{vh}^T] - E_{P_T}[\mathbf{vh}^T]). \quad (11)$$

where α is the learning rate and P_T is the distribution obtained by running a Gibbs chain, initialized with the visible units given by the data, for T full steps.

Because input skeletal features f_t are continuous instead of binomial features, we use the Gaussian-Bernoulli RBM (*GRBM*) to model the energy term of first layer. More precisely:

$$E(v,h;\theta) = -\sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} h_j \frac{v_i}{\sigma_i} - \sum_{j=1}^{F} a_j h_j \quad (12)$$

1. Note that offset features used in [21] depends on the first frame, if the initialisation fails which is a very common scenario, the feature descriptor will be generally very noisy. Hence, the offset features are discarded and only the three more robust features $[f_t^{cc}, f_t^{cp}, f_t^{ca}]$

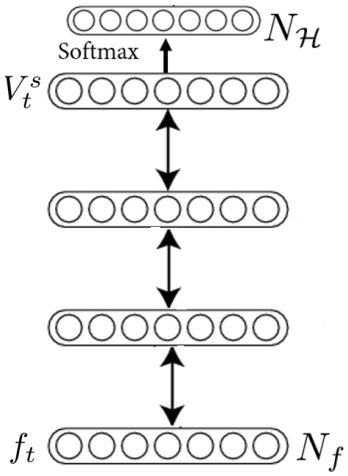


Fig. 4: Deep Belief Network using skeleton input f_t to predict emission probability $p(X_t|H_t)$.

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

$$P(h_{j=1}|\mathbf{v}) = g(\sum_i W_{ij}v_i + a_j)); \tag{13}$$

$$P(v_{i=1}|\mathbf{h}) = \mathcal{N}(v_i|\mu_i, \sigma_i^2). \tag{14}$$

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 substraction 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 $[N_f, 2000, 2000, 1000, N_{\mathcal{H}}]$, where $N_f = 891$ is the observation domain dimensionality; $N_{\mathcal{H}}$ 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). 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 .

The performance of the skeleton module is shown in Tab. 1.

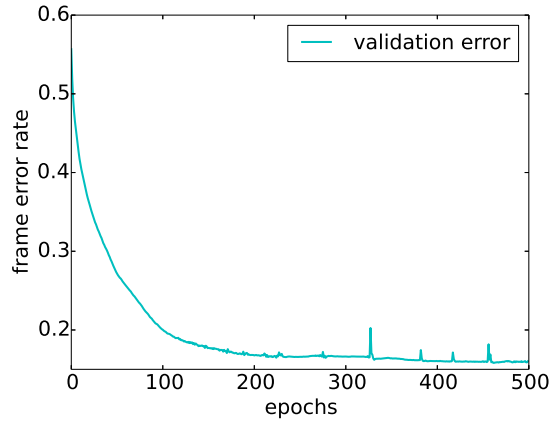


Fig. 5: Deep belief network frame based validation error rate for the skeleton input module. The 0.05 frame error rate indicates the well generalisation Deep Belief Network of skeleton modules.

4.3 RGB & Depth 3D Module

4.3.1 Preprocessing

Although DeepMind technology [49] 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 handiness, we 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 thresholding, background removal using the user index, and median filtering. The depth images are normalised with mean substraction for each image because the median of depth images are irrelevant to the gesture subclass. And both RGB and depth images are normalised by dividing the standard deviation of each image. The outcome is illustrated in Fig. 6.

4.3.2 3DCNN Architecture

This deep neural network architecture consists of a series of layers composed of either convolution, pooling or, in the last layer, fully connected layers. The 3D convolution itself is achieved by convolving a 3D kernel to the cuboid formed by stacking multiple contiguous frames together. We follow the nomenclature as in [19]. However, instead of using $tanh$ units as in [19], Rectified Linear Units (*ReLUs*) [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)

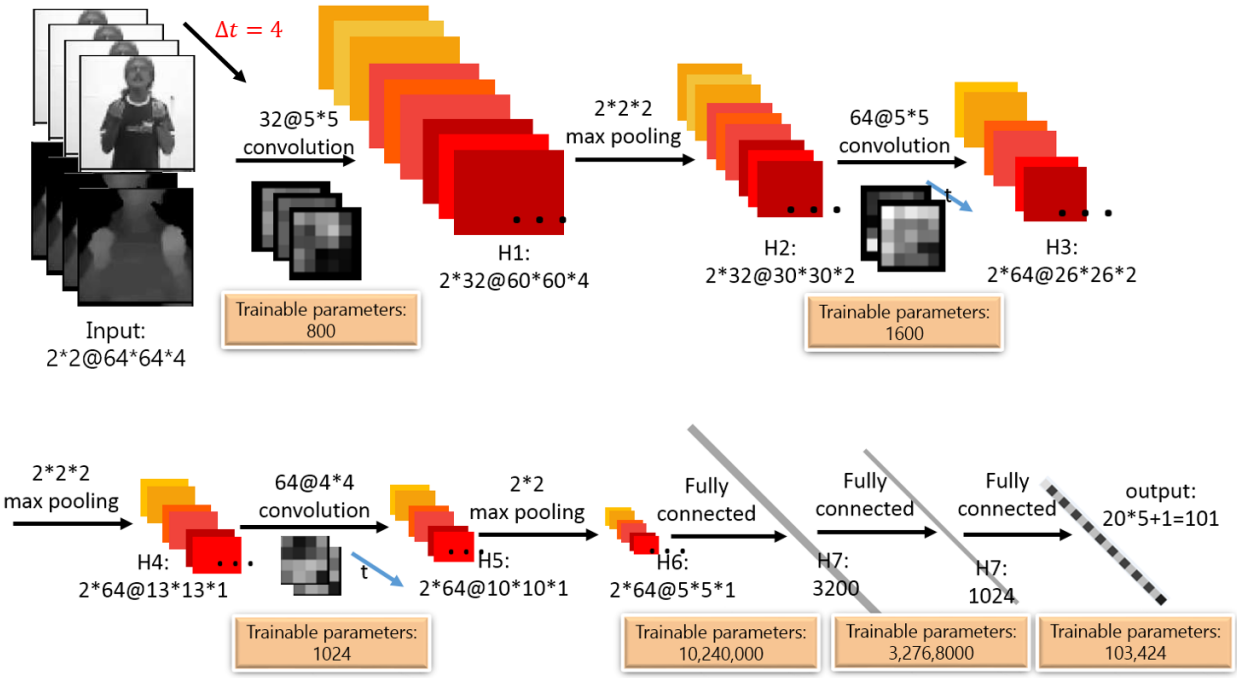


Fig. 7: An illustration of the architecture of the 3DCNN architecture. The 3rd dimension of the input is time with 4 frames stacked together. The depth and RGB data are stacked (concatenated) together at Input. Hand and body part output are concatenated at H7.

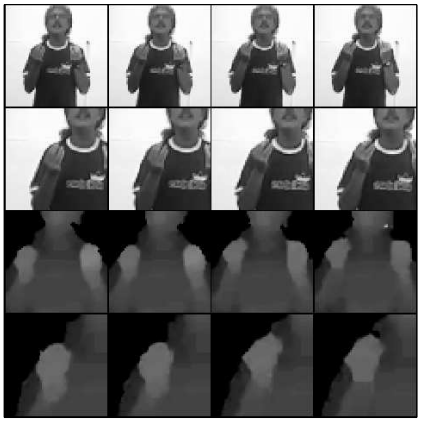


Fig. 6: Preprocessing result. Inputs from top to bottom: 1) grayscale body input, 2) grayscale hand input, 3) depth body input, 4) depth hand input.

in the j-th feature map in the i-th layer, denoted as v_{ij}^{xyz}, is given by:

$$v_{ij}^{xyz} = max(0, (b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(t+r)})) \quad (15)$$

The complete 3DCNN architecture is depicted in Fig. 7 : 4 types (Fig. 6) of input contextual frames are stacked as size $64 \times 64 \times 4$. The first layer consists of 32 feature maps produced by 5×5 spatial convolutional kernels followed by local contrast normalisation (LCN) [50] and 3D max pooling with strides $(2, 2, 2)$, then the grayscale channel and depth channel are concatenated. The second layer has 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 convolutional layer outputs are flattened with the body channel and hand channel concatenated, and fed into one fully connected layer of size 1024. The output layer has $N_{\mathcal{H}}$ values, the number of states in the HMM. is of size $101 = 5 \times 20 + 1$ (number of hidden states for each class $\times$ number of classes plus one ergodic state).

4.3.3 Details of Learning

During training, dropout [51] is used as main regularisation approach to reduce overfitting. Nesterovs accelerated gradient descent (NAG) [52] with a fixed momentum-coefficient of 0.9 and mini-batches of size 64 are also used. The learning rate is initialised at 0.003 with a 5% decrease after each epoch. The weights of the 3DCNNs are randomly initialised with a normal distribution with $\mu = 0$ and $\sigma = 0.04$. The frame based validation error rate is 39.06% after 40 epochs as shown in Fig. 8. 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. 9. The unique characteristics from the kernels are clearly visible: For body and hand filters, both inputs are of the same size : 64×64. Hand inputs have larger homogenous

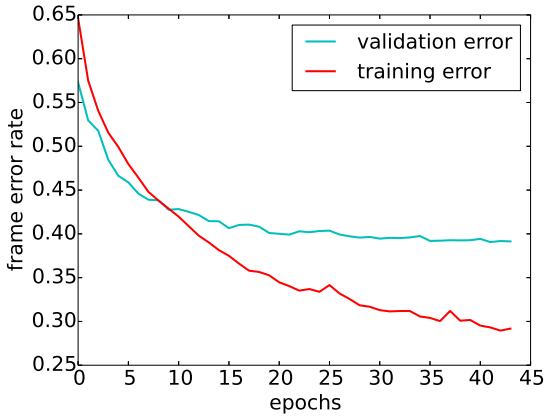


Fig. 8: The frame based error rate for training 3DCNN. The frame error rates from 3DCNN are much higher than the skeleton module in 5 which indicates learning from images may be more difficult than from skeleton modules.

areas than the body inputs, hence the hand part filters are smoother than the body part filters.

Depth images have sharper edges as also observed in [35] and generally are smoother than the grayscale filters, though the distinctions are less obvious compared with the body versus hand filters.

4.4 Multimodal Fusion

To fuse both model predictions, the strategies shown in Fig. 10 are adopted.

4.4.1 Late Fusion

Formally, the multimodal fusion is a score fusion defined by:

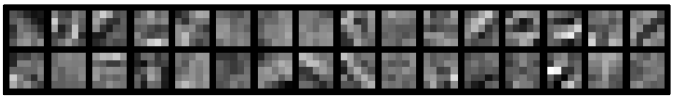
$$\mathbf{S} = \alpha * \mathbf{S^1} + (1 - \alpha) * \mathbf{S^2} \tag{16}$$

where $\mathbf{S^1}$ and $\mathbf{S^2}$ are the score probability matrices as in Algo. 2, corresponding to the skeletal input and RGBD input, and α is the coefficient that controls the score balance obtained by cross validation. Interestingly, the best performing α is close to 0.5, thus indicating that both approaches perform comparably.

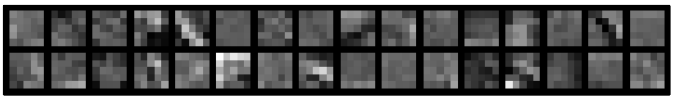
The complementary properties of both modules can be seen from the Viterbi path decoding plot in Fig. 12.

4.4.2 Early Fusion

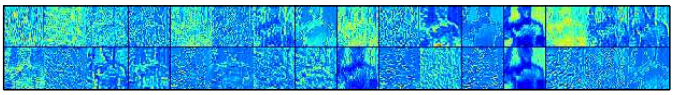
Instead of traditional late fusion, we adopt another layer of perceptron(with 2024 hidden unites) for cross modality learning taking the input from each individual net's penultimate layer. The parameters of two neural networks (for skeleton and depth) are loaded from the previously trained individual module. We argue that this "pre-trained" parameters are important because the heterogenous inputs of the system. We fine-tune the network and the stop the training when validation error rate stop decreasing ($\sim$15 epochs). The result for early fusion system are reported in Tab. 1.



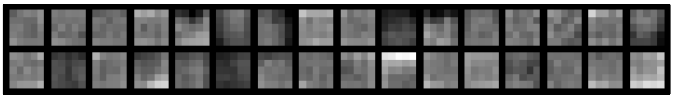
(a) Grayscale body filters.



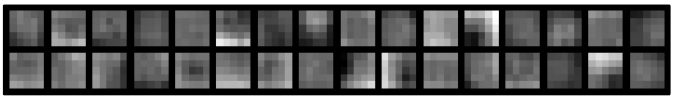
(b) Depth body filters.



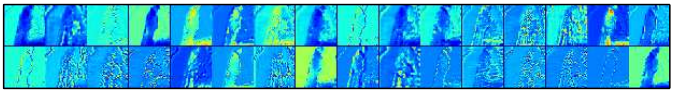
(c) Response maps corresponds to body part.



(d) Grayscale hand filters.



(e) Depth hand filters.



(f) Response maps corresponds to hand part.

Fig. 9: 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. And response maps correspond to hand part are smoother than those correspond to body part.

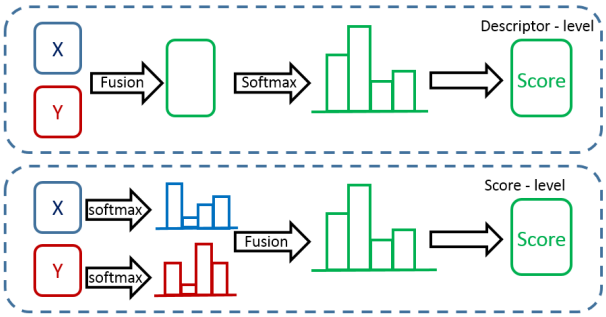


Fig. 10: Different Pipelines for descriptor fusion.

However, we can see from Tab. 1 that the early fusion system didn't outperform the late fusion system. The result is counter-intuitive because we expect the early fusion multimodal feature learning will extract semantically meaningful shared representations, outperforming individual modalities, and the early fusion schemes efficacy against the traditional method of late fusion. One possible expla-

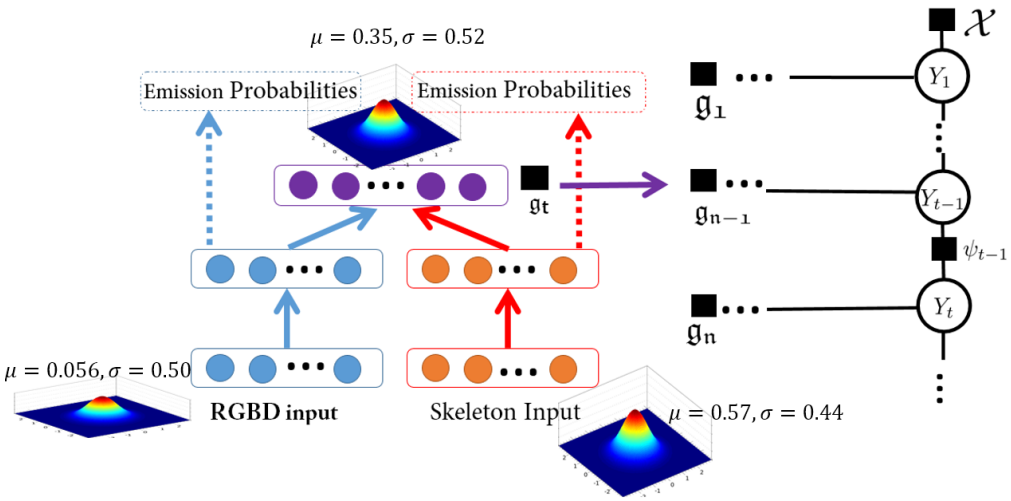


Fig. 11: Architecture of the multimodal dynamic networks: each modality (RGBD and skeleton) is first pre-trained by a Deep Neural Network, and their penultimate layers are fused together to generate a shared representation for dual modalities. The outputs are the emission probabilities $\mathfrak{g}_t$ for temporal dynamic modeling. The Gaussians represent activations of the neurons for each input modality and the final fusion layer. The mean activation of skeleton module neurons is predominantly larger than the RGBD ConvNet's (0.57 vs. 0.056). Note that skeleton module has logistic units whereas ConvNet module has leaky relu unit [53], hence the mean activations of the two are not directly comparable. However, 10 times the difference of mean activation indicates the bias during the multimodal fine-tuning phase that could cause the less than expected performance.

nation 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 modules in Fig. 11 indicate the aforementioned conjecture.

5 Experiments and Analysis

5.1 Chalearn LAP Data Set & Evaluation Metrics

The data set[2] used in this work is provided by the ChaLearn LAP [37] gesture spotting challenge. The development set consists of 700 video sequences and 240 sequences are used for testing. The testing sequences however are not segmented a priori and the gestures must be detected within a continuous data stream. In total, there are more than 14 000 performed gestures.

For the input sequences, there are three modalities provided, *i.e.* skeleton, RGB and depth images (with user segmentation). In the following experiments, the first 650 videos equences are used for training, 50 for validation and the other 240 for testing where each sequence contains around 10 to 20 gestures with some noisy non-meaningful vocabulary tokens.

The evaluation of this data set is performed using the Jaccard index, which computes the overlap between the ground truth and the predictions on a frame-by-frame basis:

$$J(A, B) = \frac{A \bigcap B}{A \bigcup B} \tag{17}$$

where A is the ground truth gesture label and B is the predicted gesture label.

2. http://gesture.chalearn.org/homewebsourcereferrals

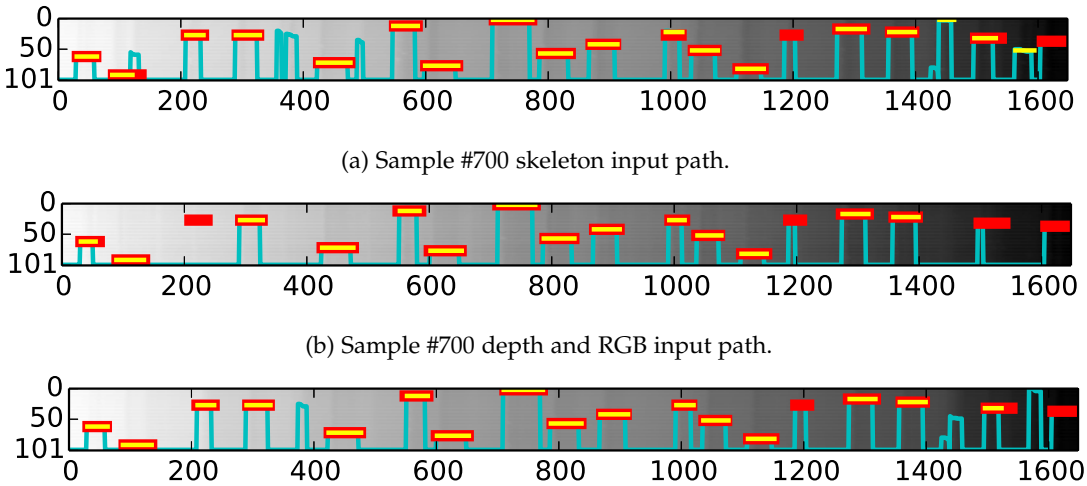
5.2 Post-processing

The predicted tokens that happen to be less than 20 frames are discarded as noisy tokens. Note that there are many noisy gesture tokens predicted by viterbi decoding. One way to sift through the noisy tokens is to discard the token path log probability smaller than a certain threshold. However, because we use the Jaccard index as evaluation score, it strongly penalises false negatives. Experiments show that the evaluation metric favours having more false positives than missing true positives. Effective ways to detect false positives should be an interesting aspect of future work.

5.3 Results

The individual module results and the fusion results are shown in Tab. 1. Note that the skeleton module generally performs better than the depth module, one reason could be that the skeleton joints learnt from [23] lie in success of utilising huge and highly varied training data: from both realistic and synthetic depth images, a total number of 1 million images were used to train the deep randomised decision forest classifier in order to avoid overfitting. Hence, skeleton data is more robust.

From the frame based prediction, we also evaluate the gesture token classification rate using the commonly-used PASCAL overlap criterion: if the gesture is predicted correctly with more than 50% overlap with the ground truth label, then the prediction is counted as a true positive. The results of the two individual modules and the score of the fused modules are shown in Tab. 2. From the confusion matrices in Fig. 13 we can observe the complimentary information between the skeleton input and the RGBD input. While



(a) Sample #700 skeleton input path.

(b) Sample #700 depth and RGB input path.

(c) Sample #700 combined input path.

Fig. 12: Viterbi decoding of the two modules and their fusion of sample sequence #700. Top to bottom: skeleton, RGBD, multimodal fusion with x-axis representing the time and y-axis representing the hidden states of all the classes with the ergodic state at the bottom. Red lines are the ground truth labels, cyan lines represent the viterbi shortest path and yellow lines are the predicted labels. There are some complementary information of the two modules and generally the skeletal module outperforms the depth module. The fusion of the two could exploit the uncertainty, *e.g.* around frame 200 the skeleton can help with the false negative predictions given by the 3DCNN module. Around frame 1450, the 3DCNN module can help suppress the false positive prediction given by skeleton module.

Module	Validation	Test
Skeleton – DBDN	0.78266	0.77920
RGBD – 3DCNN	0.75163	0.71678
Multimodal Late Fusion	0.81744	0.80910
Multimodal Early Fusion	0.80014	0.79800

TABLE 1: Comparison of results in terms of Jaccard index between different network structures and various modules.

		Validation	Test
Skeleton	Acc	0.8633	0.8360
	UnRate	0.0230	0.0412
RGBD	Acc	0.7871	0.7581
	UnRate	0.1612	0.1976
Multimodal Late Fusion	Acc	0.8791	0.8642
	UnRate	0.0302	0.0485
Multimodal Early Fusion	Acc	0.8650	0.8545
	UnRate	0.0623	0.0768

TABLE 2: Gesture classification accuracy (*Acc*) and undetected rate (*UnRate*): if the prediction overlaps with the ground truth with more than 50%, it's considered a true positive.

many of the gestures in this data set could be mainly differentiated by examining the positions and motions of large joints such as the elbows and wrists, some gestures differ primarily in hand pose, *e.g.* Fig. 15.

5.4 Computational Complexity

Although training the deep neural network using stochastic gradient descent is computationally intensive, once the model finishes training, our framework can perform real-time video sequence labelling with a low inference cost. Specifically, a single feed forward neural network incurs linear computational time ($\mathcal{O}(T)$) and is efficient because it

Module	Skeleton	RGBD	Fusion
[38] Deep Learning (Step 4)	0.7891	0.7990	**0.8449**
[38] Deep Learning (multiscal)	0.8080	0.8096	**0.8488**
[39] 3 Set Skeletal & HOG	0.791	-	0.8220
[54] Handcrafted features	**0.7948**	-	0.8268
[41] Dense Trajectory	-	**0.7919**	-
[32] CNN	-	0.7888	-
[31] Deep Learning	0.7468	0.6371	0.8045
DDNN (this work)	0.7792	0.7168	0.8091

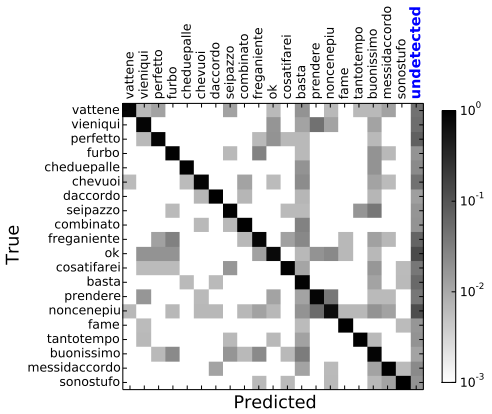
TABLE 3: Comparison of results in terms of Jaccard index between different network structures and various modules.

requires only matrix products and convolution operations. The complexity of the Viterbi algorithm is $\mathcal{O}(T * |S|^2)$ with T the number of frames and $|S|$ the number of states.

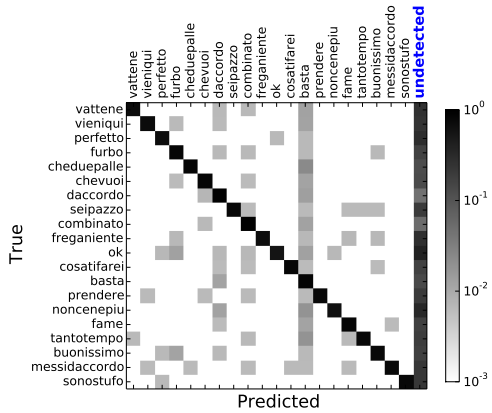
Using a modern GPU (GeForece GTX TITAN Black), our multimodal neural network can be deployed at 90 FPS using Theano library [55] which is 6 times of real time (15 FPS). The preprocessing part takes most of the time and our un-optimized preprocessing pipeline is 25 FPS and Viterbi decoding is 90 FPS. Hence, the overall system can achieve faster than real-time performance.

6 Conclusion and Discussion

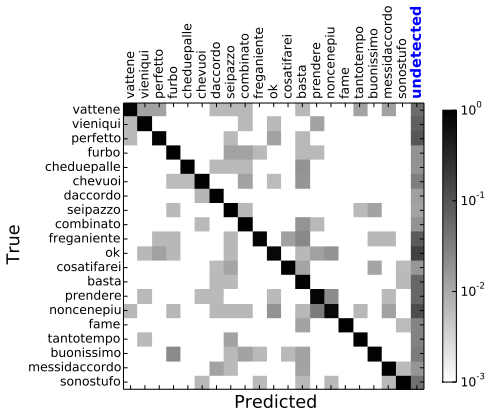
Hand-engineered, task-specific features are often less adaptive and time-consuming to design. This difficulty is more pronounced with multimodal data as the features have to relate with multiple data sources. In this paper, we presented a novel deep dynamic neural network (DDNN) framework that utilises deep belief networks and 3D convolutional neural networks for learning contextual frame-level representations and modelling emission probabilities



(a) Skeletal input prediction result.



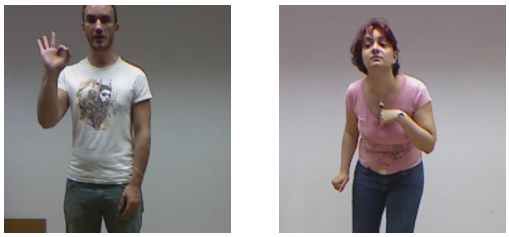
(b) RGBD input prediction result.



(c) Multimodal fusion prediction result.

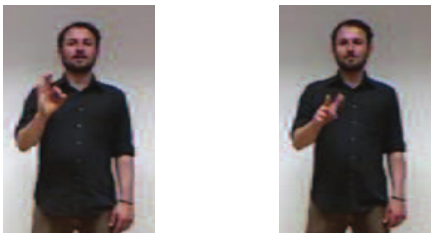
Fig. 13: Confusion Matrix for the skeletal input, RGBD input and multimodal fusion result. Some gestures, *e.g.* "OK" and "Non ce ne piu" differ primarily in hand poses. Hence, they are easier to be differentiated using the RGBD module than the skeleton module.

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



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

Fig. 14: Examples of overall upper body movement's influence on system performance. Left (score: 0.94) performer almost kept static upper body whilst performing Italian sign language. Right (score: 0.34) performer moved vehemently when performing the gestures.



(a) "OK" (b) "Non ce ne piu"

Fig. 15: Examples of gestures that differ primarily in hand pose but not the arm motions.

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 at the penultimate layer and backpropagating the gradient in the share space in a unified representation.

Appendix A
Details of the Code

The python code for this work can be found at:
`https://github.com/stevenwudi/chalearn2014_wudi_lio`

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. Gonzlez, 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," *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. Gonzlez, 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. 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.

[44] A. Lehrmann, P. Gehler, and S. Nowozin, "A non-parametric bayesian network prior of human pose," in *International Conference on Computer Vision*, 2013.

[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] R. Salakhutdinov, "Learning deep generative models," Ph.D. dissertation, University of Toronto, 2009.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] L. Pigou, A. V. D. Oord, S. Dieleman, M. V. Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *CoRR*, vol. abs/1506.01911, 2015. [Online]. Available: http://arxiv.org/abs/1506.01911
[54] J. Y. Chang, "Nonparametric gesture labeling from multi-modal data," in *European Conference on Computer Vision and Pattern Recognition Workshops*, 2014.
[55] 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.
[56] 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.
[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

APPENDIX B
REVIEW

B.1 Editor Comments

Associate Editor Comments to the Author: 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: Wed 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 $s = a * s1 + (1 - a) * s2$ 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. 11. 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 1.
- 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 9.
- Explanation of intuition behind higher level presentation of the skeleton features. We include Section 3.1 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.

B.2 Reviewer Comments

B.2.1 Reviewer1

Recommendation: Accept With No Changes

Comments: *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.

B.2.2 Reviewer2

Recommendation: Revise and resubmit as new

Comments: *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."

Comments: *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. 1. 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 ($\sim$15 epochs)...However, we can see from Tab. 1 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 [56]. 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. 11 indicate the aforementioned conjecture."

Comments: *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.

Comments: *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."

B.2.3 Reviewer3

Recommendation: Author Should Prepare A Minor Revision

Comments:

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, 1) we include the Problem formation in Sec 3.2 that explains the intuition behind the higher level representation and the advantages offered by feed forward neural over GMMs. 2) we include another section for learning the higher level representation for skeleton joints features in Section 3.2 from a pre-training point of view. The pre-training step is of crucial importance in learning the right initialisation of the deep belief networks using the sigmoid activation function.

Comments: *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 skeletan 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 $\boldsymbol{a}$ by force-alignment. The unobserved variable H_t can be interpreted as a segment of an action $\boldsymbol{a}$. 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."

Comments: *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.

Comments: *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.

Comments: *"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 Markove 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.

Comments: *"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."

Comments: *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, [57] 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."

Comments: *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".

Comments: *"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."

Comments: *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."

B.2.4 Reviewer4

Recommendation: Author Should Prepare A Major Revision For A Second Review

Comments: *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 [56] 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 Section4.4.2. "However, we can see from Table 1 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 [56]. 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. 11 indicate the aforementioned conjecture."

Comments: *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"

Comments: *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.

Comments: *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 subsequences. 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."

Comments: *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., Arbelez, P., & Malik, J. (2014). Learning Rich Features from RGB-D Images for Object Detection and Segmentation. arXiv Preprint arXiv:1407.5736, 116. doi:10.1007978-3-319-10584-0-23
[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 RBG-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."