

Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition

Di Wu, Lionel Pigou, Ling Shao, Pieter-Jan Kindermans, and Joni Dambre

Abstract—This paper describes a novel method called deep dynamic neural networks (*DDNN*) for the dynamic multimodal gesture recognition. A generalised semi-supervised hierarchical dynamic framework is proposed for simultaneous gesture segmentation and recognition taking skeleton, depth and RGB images as input modules. Unlike the traditional construction of complex handcrafted features, all inputs modules 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 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 focused on adapting handcrafted features. For example, [1], [2], [3] have been heavily employed with great success. 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” data sets. 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 method is data set dependent. This clearly indicates that learning a data set specific feature extractor, as is done in this work, 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 the noisy speech data set. 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 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]).

Due to the immense popularity of Microsoft Kinect [23], [24], there has been renewed interest in developing methods for human gesture and action recognition from 3D skeletal data and depth images. A number of new data sets [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 it may seem that the task of gesture recognition based on 3D joint positions is trivial, this is certainly not the case, largely due to the high dimensionality and the huge amount of variability of the pose space. A second aspect that further complicates the detection problem is the segmentation of the different gestures. While this segmentation is as important as the recognition, it is an often neglected aspect of action recognition research.

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.

The key contributions of this work can be summarised as follows:

- We propose a hierarchical dynamic framework that first extracts high-level skeletal joint features and then use the learned representation for estimating emission probability to infer gesture sequences.
- We develop a 3D dynamic convolutional neural network architecture based on the convolution feature extractors for multiple channel inputs, *e.g.* depth, grayscale RGB with hand and upper body as input. The proposed framework labels a video sequence in a frame-to-frame mechanism, rendering it possible for online segmentation and recognition for both RGB and depth images.
- We propose a late fusion strategy for the dynamic hidden Markov model, showing that multiple channel fusion outperforms individual modules by a large margin.

2 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.

The proposed approach combines two distinct machine learning models that each work on a different modality. Please note that the individual models can also be used independently. Both models are based on a hidden Markov model, where the emission probabilities are modeled by a deep neural network. 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, utilises RGB and depth (RGBD) video data and uses convolutional neural networks to model the emission probabilities. In the remainder of this section we will detail the shared concepts behind both

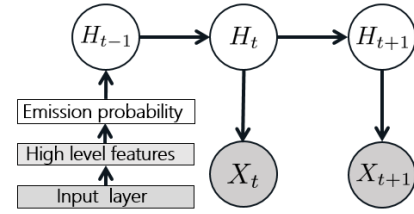


Fig. 1: Per-gesture model: a forward-linked chain. Inputs (skeletal features, or RGBD image features) are first passed through deep neural nets (deep belief networks for skeletal modality or 3D convolutional neural networks for RGBD modality) to extract high-level features. The outputs are the emission probabilities of the hidden states.

models. The next section is dedicated to the specific aspects of the two distinct models and we will also highlight the fusion mechanism and the required post-processing tools.

2.1 Deep Dynamic Neural Networks

The proposed deep dynamic neural network (DDNN) can be seen as an extension to [21], in that instead of only using the restricted Boltzmann machines to model human motion, various connectivity layers, *e.g.* fully connected layers, convolutional layers, *etc.*, 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 . Additionally we have an unobserved variable H_t taking on values in a finite set $\mathcal{H} = (\bigcup_{a \in \mathcal{A}} \mathcal{H}_a)$, where \mathcal{H}_a is a set of states associated with an individual gesture a by force-alignment. The 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. With these definitions, the full probabilistic model is now specified as a hidden Markov model:

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

where $p(H_1)$ is the prior on the first hidden state; $p(H_t|H_{t-1})$ is the transition dynamics model and $p(X_t|H_t)$ is the emission probability modelled by the deep neural nets.

The motivation for using deep neural nets to model the emission probabilities conditional distributions is that by constructing multi-layer networks, semantically meaningful high-level features will be extracted whilst learning the parametric prior of human pose from a massive pool of data. In the recent work of [29], a non-parametric bayesian network is adopted for human pose prior estimation, whereas in the proposed framework, the parametric networks are incorporated. The graphical representation of a per-gesture model is shown in Fig. 1.

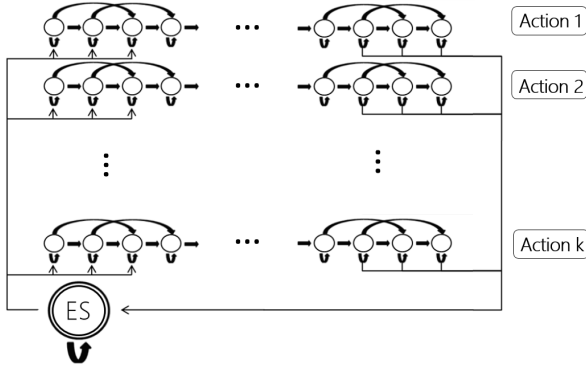


Fig. 2: State diagram of the *ES-HMM* model for low-latency gesture segmentation and recognition. An ergodic state (\mathcal{ES}) shows the resting position between gesture sequences. Each node represents a single frame and each row represents a single gesture model. The arrows indicate possible transitions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

2.2 Ergodic States Hidden Markov Model

The aforementioned framework can be easily adapted for simultaneous gesture segmentation and recognition by adding an ergodic state (\mathcal{ES}) which resembles the silence state for speech recognition. Hence, the hidden variable H_t can take on an extra value within the finite set, which becomes $\mathcal{H} = (\bigcup_{a \in \mathcal{A}} \mathcal{H}_a) \cup \mathcal{ES}$, 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.

Since our goal is to capture the variation in speed of the performed gestures, we set the transition matrix $\mathbf{p}(\mathbf{H}_t | \mathbf{H}_{t-1})$ in the following way as shown in Fig. 2: 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). 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} .

The *ES-HMM* framework differs from the firing hidden Markov model of [30] in that we strictly follow the temporal independent assumption, forbidding inter-states transverse, preconditioned that a non-repetitive sequence would maintain its unique states throughout its performing cycle.

The emission probability of the trained model is represented as a matrix of size $N_{\mathcal{TC}} \times N_{\mathcal{F}}$ where $N_{\mathcal{F}}$ is the number of frames in a test sequence and output target class $N_{\mathcal{TC}} = N_{\mathcal{A}} \times N_{\mathcal{H}_a} + 1$ where $N_{\mathcal{A}}$ is the number of gesture classes and $N_{\mathcal{H}_a}$ is the number of states associated to an individual gesture a and one \mathcal{ES} state (c.f. Fig. 10: x-axis as $N_{\mathcal{F}}$ and y-axis as $N_{\mathcal{TC}}$ with \mathcal{ES} as the bottom y-axis 101).

Once we have the trained model, we can use the normal online or offline smoothing, inferring the conditional distributions $p(H_t | X_t)$ of every node (frame) of the test video. 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. The number of

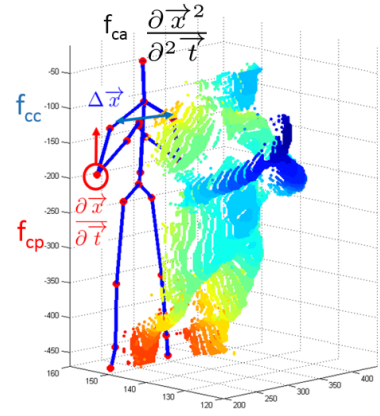


Fig. 3: A point cloud projection of a depth image and the 3D positional features.

possible paths through the lattice grows exponentially with the length of the chain. The Viterbi 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 [31]. We can infer the gesture presence in a new sequence by Viterbi decoding as:

$$V_{t,\mathcal{H}} = \log P(H_t | X_t) + \log(\max_{\mathcal{H} \in \mathcal{H}_a} (V_{t-1,\mathcal{H}})) \quad (2)$$

with the initial state $V_{1,\mathcal{H}} = P(H_1 | X_1)$. From the inference results, we define the probability of a gesture $a \in \mathcal{A}$ as $p(y_t = a | x_{1:t}) = V_{T,\mathcal{H}}$. The result of the Viterbi algorithm is a path-sequence of nodes which corresponds to hidden states of gesture classes. From this path we can infer the class of the gesture (c.f. Fig. 10). The overall algorithm for training and testing are presented in Algorithm 1 and 2.

3 MODEL IMPLEMENTATION

3.1 Hidden Markov Model

In all our experiments the number of states associated to an individual gesture $N_{\mathcal{H}_a}$ is chosen as 5 for modelling the states of a gesture class, therefore $N_{\mathcal{TC}} = 20 \times 5 + 1 = 101$. The labels for each cuboid \mathbf{Y} are specified as follows:

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 10 frames are assigned to hidden state 1, the following 10 frames are assigned to hidden state 2, and so forth.

Ergodic states (\mathcal{ES}): Neutral frames are extracted as 5 frames before or after a gesture token, according to the ground truth labels.

3.2 Skeleton Module

3.2.1 Preprocessing

Only upper body joints are relevant to the discriminative gesture recognition tasks. Therefore, only the 11 upper body joints are considered. The 11 upper body joints used are "ElbowLeft, WristLeft, ShoulderLeft, HandLeft, ElbowRight, WristRight, ShoulderRight, HandRight, Head, Spine, HipCenter".

Algorithm 1: Multimodal deep dynamic networks – training

Data:
 $\mathbf{X}^1 = \{\mathbf{x}_i^1\}_{i \in [1 \dots t]}$ - raw input (skeletal) feature sequence.
 $\mathbf{X}^2 = \{\mathbf{x}_i^2\}_{i \in [1 \dots t]}$ - 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.
 $\mathbf{Y} = \{\mathbf{y}_i\}_{i \in [1 \dots t]}$ - frame based local label (achieved by semi-supervised forced-alignment), where $\mathbf{y}_i \in \{C * S + 1\}$ with C the number of classes, S is the number of hidden states for each class, 1 as ergodic state.

```

1 for  $m \leftarrow 1$  to 2 do
2   if  $m$  is 1 then
3     Preprocess skeletal data  $\mathbf{X}^1$  as in Eq.3, 4, 5.
4     Normalise (zero mean, unit variance per dimension) the above features and feed it to Eq.6.
5     Pre-train the networks using Contrastive Divergence.
6     Supervised fine-tuning of the deep belief networks using  $\mathbf{Y}$  by standard mini-batch SGD backpropagation.
7   else
8     Preprocess the depth and RGB data  $\mathbf{X}^2$  as in 3.3.1.
9     Feed the above features to Eq.9.
10    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.
 $\mathbf{p}(\mathbf{H}_1)$ - prior probability for \mathbf{Y} by accumulating and normalising labels.
 $\mathbf{p}(\mathbf{H}_t | \mathbf{H}_{t-1})$ - transition probability for \mathbf{Y} , enforcing the beginning and ending of a sequence can only start from the first or the last state.

The 3D coordinates of the N joints of frame c are given as: $X_c = \{x_1^c, x_2^c, \dots, x_N^c\}$. 3D positional pairwise differences of joints [21] are used in the representation of the observed variable \mathcal{X} . They capture posture features, motion features by direction concatenation: $\mathcal{X} = [f_{cc}, f_{cp}, f_{ca}]$ as in Eq. 3, 4, 5. 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 $[f_{cc}, f_{cp}, f_{ca}]$ (as shown in Fig. 3) are kept for representing the frame pairwise difference,

Algorithm 2: Multimodal deep dynamic networks – testing

Data:
 $\mathbf{X}^1 = \{\mathbf{x}_i^1\}_{i \in [1 \dots t]}$ - raw input (skeletal) feature sequence.
 $\mathbf{X}^2 = \{\mathbf{x}_i^2\}_{i \in [1 \dots t]}$ - raw input (depth) feature sequence in the form of $M \times M \times T$.
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.
 $\mathbf{p}(\mathbf{H}_1)$ - prior probability for \mathbf{Y} .
 $\mathbf{p}(\mathbf{H}_t | \mathbf{H}_{t-1})$ - transition probability for \mathbf{Y} .

```

1 for  $m \leftarrow 1$  to 2 do
2   if  $m$  is 1 then
3     Preprocessing and normalising the data  $\mathbf{X}^1$  as in Eq. 3, 4, 5.
4     Feedforwarding network GDBN to generate the emission probability  $\mathbf{p}(\mathbf{X}_t | \mathbf{H}_t)$  in Eq.1.
5     Generating the score probability matrix  $\mathbf{S}^1 = \mathbf{p}(\mathbf{H}_{1:T}, \mathbf{X}_{1:T})$ .
6   else
7     Preprocessing the data  $\mathbf{X}^2$  (normalising, median filtering the depth data).
8     Feedforwarding 3DCNN to generate the emission probability  $\mathbf{S}^2 = \mathbf{p}(\mathbf{X}_t | \mathbf{H}_t)$  in Eq.1.
9     Generating the score probability matrix  $\mathbf{S}^2 = \mathbf{p}(\mathbf{H}_{1:T}, \mathbf{X}_{1:T})$ .
10  Fuse the score matrix  $\mathbf{S} = \alpha * \mathbf{S}^1 + (1 - \alpha) * \mathbf{S}^2$ .
11 Finding the best path  $\mathbf{V}_{t, \mathcal{H}}$  using  $\mathbf{S}$  by Viterbi decoding as in Eq.2.
```

Result:
 $\mathbf{Y} = \{\mathbf{y}_i\}_{i \in [1 \dots t]}$ - frame based local label where $\mathbf{y}_i \in \{C * S + 1\}$ with C the number of classes, S is the number of hidden states for each class, 1 as ergodic state.
C - global label, the anchor point is chosen as the middle state frame.

velocity and acceleration elements for skeletal features:

$$f_{cc} = \{x_i^c - x_j^c | i, j = 1, 2, \dots, N; i \neq j\} \quad (3)$$

$$f_{cp} = \{x_i^c - x_i^p | x_i^c \in X_c; x_i^p \in X_p\} \quad (4)$$

$$f_{ca} = \{x_i^p - 2 \times x_i^c + x_i^n | x_i^c \in X_c; x_i^p \in X_p; x_i^n \in X_n\} \quad (5)$$

with X_i^c, X_i^p, X_i^n as the current, previous and next frame skeletal features.

This results in a raw dimensionality of $N_{\mathcal{X}} = N_{joints} * (\frac{N_{joints}}{2} + N_{joints} + N_{joints}) * 3$ where N_{joints} is the number of joints used. Therefore, in the experiment with $N_{joints} = 11$, $N_{\mathcal{X}}$ is equal to 891. Admittedly, 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 about the relevant features, the resulting ones remain quite general and do not need dataset specific tuning. A similar data driven approach has

been adopted in [26] where random forest classifiers were adapted to the problem of recognising gestures using a bundle of 35 frames. These sets of feature extraction processes resemble the *Mel Frequency Cepstral Coefficients* (MFCCs) for the speech recognition community [20].

3.2.2 Gaussian-Bernoulli Restricted Boltzmann Machines

Because input skeletal features (*a.k.a.* observation domain \mathcal{X}) are continuous instead of binomial features, we use the Gaussian-Bernoulli RBM (GRBM) to model the energy term of first visible layer:

$$E(v, h; \theta) = - \sum_{i=1}^D \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i=1}^D \sum_{j=1}^F W_{ij} h_j \frac{v_i}{\sigma_i} - \sum_{j=1}^F a_j h_j \quad (6)$$

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

$$P(h_{j=1} | \mathbf{v}) = g\left(\sum_i W_{ij} v_i + a_j\right); \quad (7)$$

$$P(v_{i=1} | \mathbf{h}) = \mathcal{N}(v_i | \mu_i, \sigma_i^2). \quad (8)$$

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

For high-level skeleton feature extraction, the network architectures is $[N_{\mathcal{X}}, 2000, 2000, 1000, N_{\mathcal{T}\mathcal{C}}]$, where $N_{\mathcal{X}} = 891$ is the observation domain dimensionality; $N_{\mathcal{T}\mathcal{C}}$ is the output target class.

3.2.3 Deep Belief Networks Pretraining & Training Details

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 descent 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 4.

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

3.3 RGB & Depth 3D Module

3.3.1 Preprocessing

Working directly with raw input Kinect recorded data frames, which are 640×480 pixel images, can be computationally demanding. DeepMind technology [32] presented the first deep learning model to successfully learn control

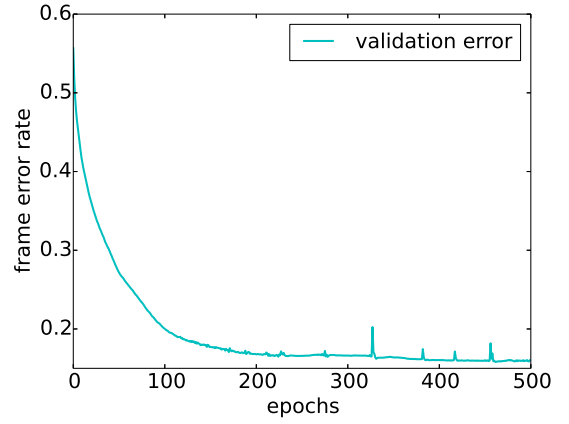


Fig. 4: Deep belief network frame based validation error rate for the skeleton input module.

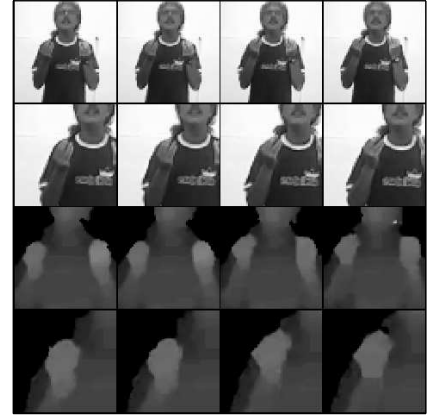


Fig. 5: Preprocessing result. Inputs from top to bottom: 1) grayscale body input, 2) grayscale hand input, 3) depth body input, 4) depth hand input.

policies directly from high-dimensional sensory input using deep reinforcement learning.

Our first step in the preprocessing stage is cropping the highest hand and the upper body using the given joint information. We determined that the highest hand is the most interesting. If both hands are used, they perform the same (mirrored) movement. If one hand is used, it is always the highest one. If the left hand is used, the videos are mirrored. This way, the model only needs to learn one side.

The preprocessing results in four video samples (body and hand with grayscale and depth) of resolution $4 \times 64 \times 64$ (4 frames of size 64×64). Furthermore, the noise in the depth maps is reduced by thresholding, background removal using the user index, and median filtering. The outcome is shown in Fig. 5.

3.3.2 3DCNN Architecture

The 3D convolution 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

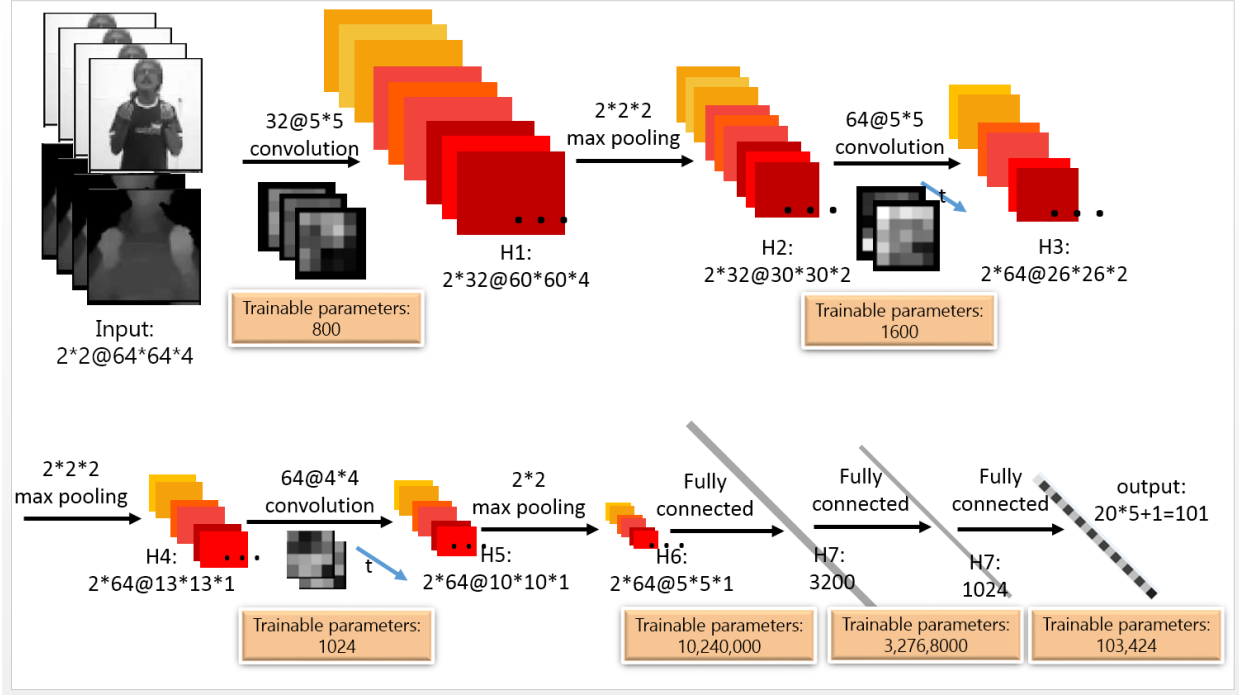


Fig. 6: An illustration of the architecture of the 3DCNN architecture.

Linear Units (*ReLU*s) [16] were used in order to speed up training. Formally, the value of a unit at position (x, y, z) (z here corresponds the time-axis) in the j -th feature map in the i -th layer, denoted as v_{ij}^{xyz} , is given by:

$$v_{ij}^{xyz} = \max(0, (b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(t+r)})) \quad (9)$$

The 3DCNN architecture is depicted in Fig. 6 : the 4 types (Fig. 5) of input contextual frames are stacked as size $64 \times 64 \times 4$. The depth images are normalised with N_{var} and the grayscale images are normalised with N_{std} as in Eq. 10,11 because the median of depth images are irrelevant to the gesture subclass.

$$N_{var} = (x - \text{mean}(x)) / (\text{var}(x))^{1/2} \quad (10)$$

$$N_{std} = x / (\text{var}(x))^{1/2} \quad (11)$$

The first layer consists of 32 feature maps produced by 5×5 convolutional kernels followed by local contrast normalisation (LCN) [33] 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 N_{TC} is of size $101 = 5 \times 20 + 1$ (number of hidden states for each class \times number of classes plus one ergodic state).

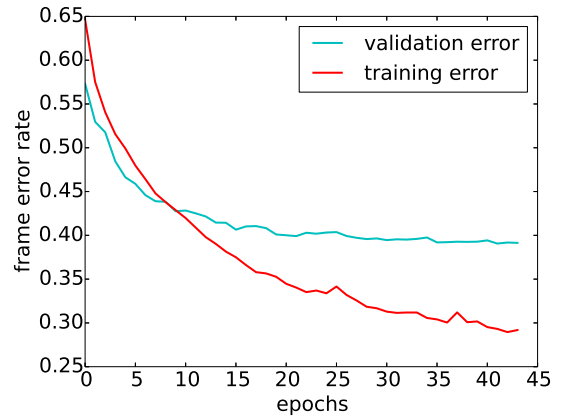


Fig. 7: The frame based error rate for training 3DCNN.

3.3.3 Details of Learning

During training, dropout [34] is used as main regularisation approach to reduce overfitting. Nesterovs accelerated gradient descent (NAG) [35] 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. 7. Compared with the skeleton module (Fig. 4), the 3DCNN has a notable higher frame based error rate.

3.3.4 Looking into the Networks: Visualisation of Filter Banks

The convolutional filter weights of the first layer are depicted in Fig. 8. The unique characteristics from the kernels

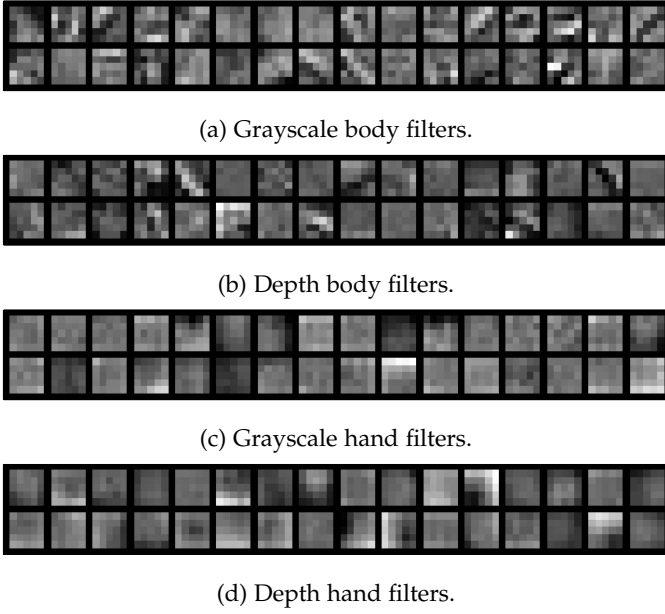


Fig. 8: Visualisation of the 5×5 filters in the first layer for the different input channels.

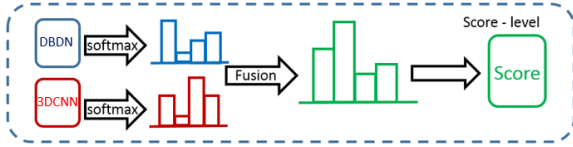


Fig. 9: Illustration of descriptor fusion.

are clearly visible. Depth images generally have less edges. Hence, depth filters are smoother than the grayscale filters, though the distinctions are less obvious compared with the body versus hand filters.

3.4 Multimodal Fusion

To fuse both model predictions, the strategy shown in Fig. 9 is adopted. The complementary properties of both modules can be seen from the Viterbi path decoding plot in Fig. 10.

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

$$\mathbf{S} = \alpha * \mathbf{S}^1 + (1 - \alpha) * \mathbf{S}^2 \quad (12)$$

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.

4 RELATED WORK

This paper is an extension of the work in [36] and [37].

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.

For the ChaLearn LAP [38] gesture spotting challenge, [39] presents a multi-scale and multimodal deep network for gesture detection and localisation. Key to their technique is a training strategy that exploits i) careful initialisation of individual modalities and ii) gradual fusion of modalities from strongest to weakest cross-modality structure. One major difference between our proposed method and theirs is the treatment of the time factor at a very early stage of the deep network to cope with gestures performed at different speeds.

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. *E.g.* [39] formulated a pose descriptor, consisting of 7 logical subsets. [40] proposed four 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.* [41] and a histogram of oriented gradients (HOG) descriptor for hand features. In [42], the state-of-the-art dense trajectory handcrafted features are adopted for the RGB module. Multiple network averaging works better than a single individual network and it can be seen from the experiments in [36] that larger nets will generally perform better than smaller nets. Averaging multi-column nets almost will certainly further improve the performance [17].

5 EXPERIMENTS AND ANALYSIS

5.1 Chalearn LAP Data Set & Evaluation Metrics

The data set¹ used in this work is provided by the ChaLearn LAP [38] gesture spotting challenge. The development set consists of 700 videosequences 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 videosequences 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:

$$J(A, B) = \frac{A \cap B}{A \cup B} \quad (13)$$

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

1. <http://gesture.chalearn.org/homewebsourcerefferrals>

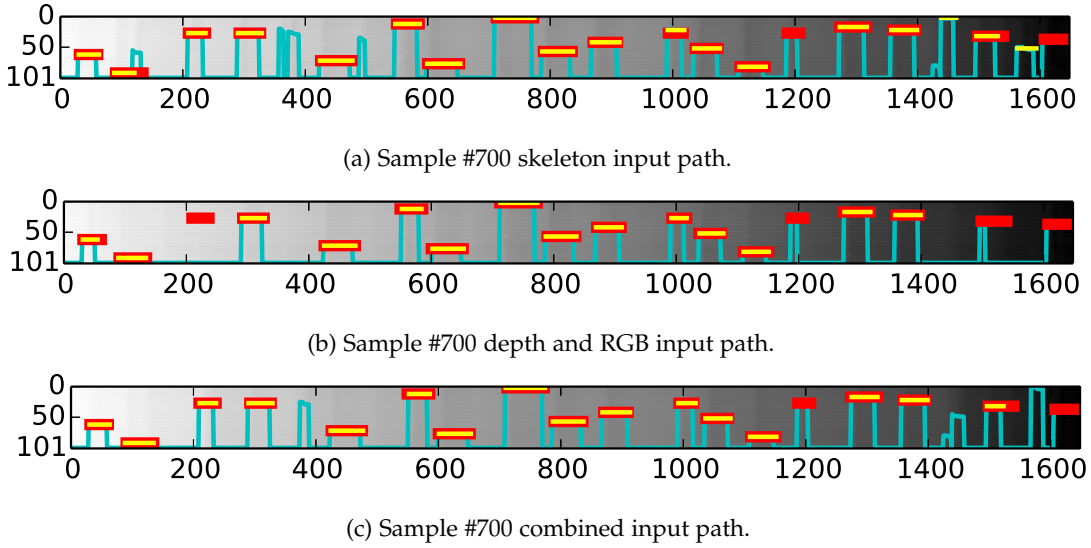


Fig. 10: Viterbi decoding of the two modules and their fusion of sample sequence #700. Top to bottom: skeleton, RGBD, multimodal fusion with x-axis representing the time and y-axis representing the hidden states of all the classes with the ergodic state at the bottom. Red lines are the ground truth labels, 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.

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 it's better to have more false positives than to miss true positives. Effective ways to detect false positives should be an interesting aspect of future works.

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. 11 we can observe the complimentary information between the skeleton input and the RGBD input. While 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. 12.

Evaluation Set		Validation	Test
Module			
Skeleton – DBDN		0.78266	0.77920
RGBD – 3DCNN		0.75163	0.71678
Multimodal Fusion		0.81744	0.80910

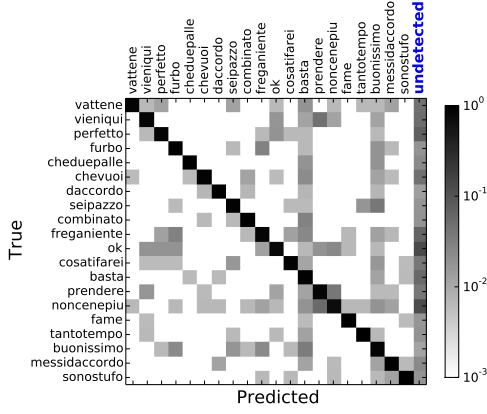
TABLE 1: Comparison of results in terms of Jaccard index between different network structures and various modules.

		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 Fusion	Acc	0.8791	0.8642
	UnRate	0.0302	0.0485

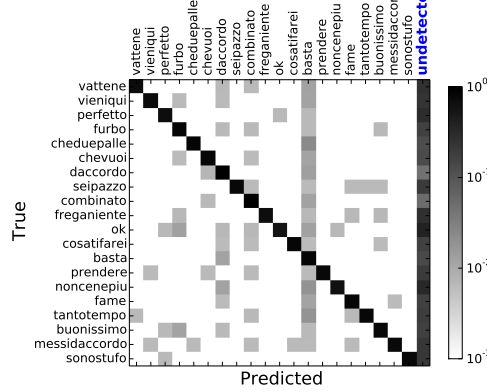
TABLE 2: Gesture classification accuracy (*Acc*) and undetected rate (*UnRate*): if the prediction overlaps with the ground truth with more than 50%, it's considered a true positive.

5.3 Computational Complexity

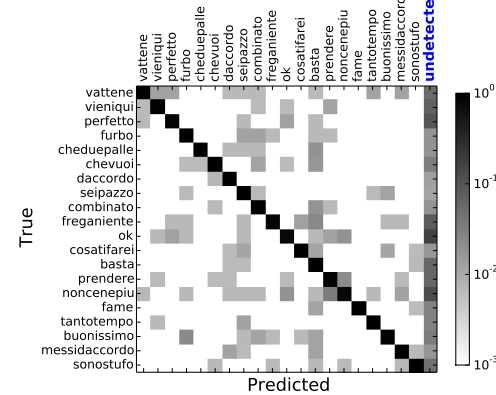
Although training the deep neural network using stochastic gradient descent is computationally intensive, once the model finishes training, our framework can perform real-time video sequence labelling with a low inference cost. Specifically, a single feed forward neural network incurs linear computational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



(a) Skeletal input prediction result.



(b) RGBD input prediction result.



(c) Multimodal fusion prediction result.

Fig. 11: 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.

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-



(a) “OK”



(b) “Non ce ne piu”

Fig. 12: Examples of gestures that differ primarily in hand pose but not the arm motions.

Module	Evaluation Set		
	Skeleton	RGBD	Fusion
[39] Deep Learning (Step 2)	0.6938	0.7862	0.8500
[40] 4 Set Skeletal & HOG	0.7420	-	0.8220
[43] Handcrafted	0.7948	-	0.8268
[42] Dense Trajectory	-	0.7919	-
[37] CNN	-	0.7888	-
[36] 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.

level representations and modelling emission probabilities for Markov fields. The heterogeneous inputs from skeletal joints, RGB and depth images require different feature learning methods and the late fusion scheme is adopted at the score level. The experimental results on bi-modal time series data show that the multimodal DDNN framework can learn a good model of the joint space of multiple sensory inputs, and is consistently as good as or better than the unimodal input, opening the door for exploring the complementary representation among multimodal inputs. It also suggests that learning features directly from data is a very important research direction and with more and more data and flops-free computational power, the learning-based methods are not only more generalisable to many domains, but also are powerful in combining with other well-studied probabilistic graphical models for modelling and reasoning dynamic sequences. Future works include learning the share representation amongst the heterogeneous inputs 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.*, "Deep-speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.
- [23] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [24] J. Han, L. Shao, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1317-1333, 2013.
- [25] S. Escalera, J. Gonzalez, X. Bar, M. Reyes, O. Lops, I. Guyon, V. Athitsos, and H. J. Escalante, "Multi-modal gesture recognition challenge 2013: Dataset and results," in *ACM ChaLearn Multi-Modal Gesture Recognition Grand Challenge and Workshop*, 2013. [Online]. Available: <http://gesture.chalearn.org/>
- [26] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *ACM Computer Human Interaction*, 2012.
- [27] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. J. Escalante, "Chalearn gesture challenge: Design and first results," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2012.
- [28] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [29] A. Lehmman, P. Gehler, and S. Nowozin, "A non-parametric bayesian network prior of human pose," in *International Conference on Computer Vision*, 2013.
- [30] S. Nowozin and J. Shotton, "Action points: A representation for low-latency online human action recognition," Tech. Rep., 2012.
- [31] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] D. Wu and L. Shao, "Deep dynamic neural networks for gesture segmentation and recognition," *European Conference on Computer Vision and Pattern Recognition Workshops*, 2014.
- [37] 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.
- [38] S. Escalera, X. Bar, J. Gonzalez, M. Bautista, M. Madadi, M. Reyes, V. Ponce, H. Escalante, J. Shotton, and I. Guyon, "Chalearn looking at people challenge 2014: Dataset and results," in *European Conference on Computer Vision workshop*, 2014.
- [39] 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.
- [40] 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.
- [41] A. Yao, J. Gall, G. Fanelli, and L. J. Van Gool, "Does human action recognition benefit from pose estimation?," in *BMVC*, 2011.
- [42] 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.
- [43] J. Y. Chang, "Nonparametric gesture labeling from multi-modal data," in *European Conference on Computer Vision and Pattern Recognition Workshops*, 2014.