Multi-modality Gesture Detection and Recognition With Un-supervision, Randomization and Discrimination

Guang Chen[1], Daniel Clarke[2], David Weikersdorfer[2], Manuel Giuliani[2], Andre Gaschler[2], and Alois Knoll[1]

[1] Technische Universität München, Garching bei München, Germany
[2] fortiss GmbH, Guerickestr. 25, 80805 München, Germany

Abstract. We describe in this paper our gesture detection and recognition system for the 2014 ChaLearn Looking at People (Track 3: Gesture Recognition) organized by ChaLearn in conjunction with the ECCV 2014 conference. The competition's task was to learn a vacabulary of 20 types of Italian gestures and detect them in sequences. Our system adopts a multi-modality approach for detecting as well as recognizing the gestures. The goal of our approach is to identify semantically meaningful contents from dense sampling spatio-temporal feature space for gesture recognition. To achieve this, we develop three concepts under the random forest framework: un-supervision; discrimination; and randomization. Un-supervision learns spatio-temporal features from four channels (grayscale, depth, gradient, surface normal) of RGB-D video in an unsupervised way. Discrimination extracts the information in dense sampling spatio-temporal space effectively. Randomization explores the dense sampling spatio-temporal feature space efficiently. An evaluation of our approach shows that we achieve a mean Jaccard Index of 0.6489, and a mean average accuracy of 90.3% over the test dataset.

Keywords: Multi-Modality Gesture; Unsupervised Learning; Random Forest; Discriminative Training

1 Introduction

Gesture detection and recognition refers to detecting and classifying meaningful motions executed by human. It has become a popular research field in recent years due to the low-cost sensors and its promising application prospects in human-computer interaction.

During the past decades, approaches of gesture recognition were controller-based, in which users had to wear human motion capture systems. The interfaces of users and devices are traditional command line and graphic user interfaces [12]. Recently, vision-based gesture recognition has become the mainstream of the research due to the abilities which enable the controller-free and natural user interactions (NUI) [5, 8]. NUI are based on natural interaction (e.g., gestures) that people use to communicate with the smart objects (e.g., smartphones).

Therefore, NUI have better user experience compared to a more traditional graphic user interface. Kinect, the motion sensing input device developed by Microsoft corporation, features a RGB camera, a depth sensor and a multi-array microphone. With all these features, Kinect serves as an ideal experimental platform for developing new NUI systems of multi-modality gesture detection and recognition.

The primary objective of the 2014 ChaLearn Looking at People (Track 3: Gesture Recognition) [7, 6] was to evaluate the performance of computational methods on gesture recognition. Track 3 of the challenge aims at the recognition of continuous, natural human gestures with the multi-modality nature of the visual cues, as well as technical limitations such as spatial and temporal resolution and unreliable depth cues.

The dataset of this competition is captured by Kinect. More than 14000 gestures are drawn from a vocabulary of 20 Italian sign gesture categories. However, the input samples may include other unrecognized gestures that are not included in the vocabulary. During the development phase, a large database of 7754 manually labeled gestures is available (referred to as the development dataset) and another dataset of 3,362 labeled gestures is provided for algorithm validation (referred to as the validation dataset). The challenge is to make predictions on the evaluation data of 3579 gestures (referred to as the testing dataset) revealed at the final evaluation phase. The evaluation is based on the Jaccard Index. For each one of the $n \leq 20$ gesture categories labelled for each sequence s, the Jaccard Index is defined as follows: $J_{s,n} = (A_{s,n} \cap B_{s,n})/(A_{s,n} \cup B_{s,n})$ where $A_{s,n}$ is the ground truth of action n at sequence s, and $B_{s,n}$ is the prediction for such a gesture at sequence s. $A_{s,n}$ and $B_{s,n}$ are binary vectors where 1-value entries denote frames in which the n-th gesture is being performed.

This paper presents an overview of our approach and gives technical details. In Section 2, we describe the overall architecture of the proposed system. In Section 3, we provide the details of the individual modules that constitute our gesture recognition system. In Section 4, we discuss the results achieved by our system. Finally, we present our conclusions in Section 5.

2 System Architecture

The architecture of the proposed multi-modality gesture detection and recognition system starts with the multi-modality input data. Each input sample in the 2014 ChaLearn Looking at People (Track 3: Gesture Recognition) contains a sequence of gestures performed by a subject and these gestures are typically separated by pauses in between. However, some of the gestures in the input sample are consecutive. Some of input samples include unrecognized gestures except for the gestures corresponding to one of the 20 gestures in the pre-defined gesture vocabulary. The first task of our approach is to detect the candidate gestures and temporally segment them by identifying their start and end frames. We use the skeletal joint data for gesture detection and segmentation. We assign each frame of the input sample a label: *gesture* or *non-gesture*, and train a Support

Vector Machine model for each input sample. Within the prediction labels of the test sample, we segment the sample into several candidate gestures.

Once the given input sample is broken down into candidate gesture segments, the next task is to provide a suitable representation of the candidate gesture contained within each segment. We utilize spatio-temporal features extracted from the RGB-D video data to represent the gesture. In contrast to previous work [10, 14, 15, 9], we extracted the spatio-temporal features using an unsupervised learning approach. At the heart of unsupervised learning approach is the extension of Independent Subspace Analysis algorithm [11, 3, 4] for the use of RGB-D video data. To effectively model the motion patterns of the gestures for the classification, we approach this problem from the perspective of mining a large number of video blocks with arbitrary shapes, spatio-temporal sizes, or locations that carry discriminative gesture video statistics. However, this approach poses a fundamental challenge: without any feature selection, even a modestly sized video will yield millions of video blocks. In addition, as large number of the blocks overlap significantly, these blocks are highly correlated and introduce significant redundancy among these features. To address this issue, we propose a random forest with discriminative decision trees approach to mine video blocks that are highly discriminative for the gesture classification tasks. Unlike traditional decision trees [1], our approach uses a SVM classifier at each node and integrates information at different depths of the tree to effectively mine a very dense sampling space of the video data. The final predicted label for a candidate gesture is assigned to the class which maximizes the average of the posterior probability from the leaf node of each tree.

3 Gesture Detection and Segmentation

We train SVM models to classify a fixed length time window of each input sample and then use a sliding window on the test sample to obtain a probability distribution over time for each window. The predicted labels for the test sample with likelihood scores average the prediction confidence of all the SVM models trained on the training samples. According to the predicted labels of the test sample, we segment the sample into several candidate gestures. A new SVM model is trained to tackle the problem of consecutive gestures in the input sample.

3.1 Segmentation based on skeletal joints

We analyze the skeletal joint data stream from the Kinect sensor to identify the start frame and end frame of each gesture within an input sample. We approach this problem as two-class classification task: classify each frame of the input sample as *gesture* frame or *non-gesture* frame. We only focus on the joints above waist level reducing the number of joints from 20 to 12.

Skeletal Feature Engineering We extract the skeletal feature from the skeletal joint data. The 3D coordinates of these joints are, however, not invariant to

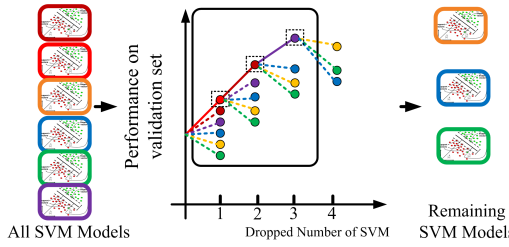


Fig. 1. Illustration of the process of greedy SVM model selection: Left: initial number T of the SVM model ($T = 6$ in this figure). Middle: greedy SVM model selection process (the number of dropped SVM model is $n = 3$). Right: the remaining $T - n$ SVM model that maximize validation performance ($T - n = 3$)

the position of the subject. Therefore we align the skeletal joints of each frame for each sample so that the hip centers of all frames overlap each other. 3D position differences of joints are employed to characterize gesture information including motion feature f_c and hand-based feature f_h. Features $f_{c,t}$ and $f_{h,t}$ are extrated from a 13-frame-long sliding window s_t where the frame t is at the center of this sliding window.

Let $p_{j,t} \in \mathbb{R}^3$ be the 3D world position $(x_{j,t}, y_{j,t}, z_{j,t})$ of joint j at frame t. J represent the 12 joints used in our approach. The motion features $f_{c,t}$ of frame t are defined as the joints differences within the sliding window s_t:

$$f_{c,t} = \{max(p_{j,i} - p_{j,t}) \mid \forall j \in J\}, i \in [t-6, t+6]; i \neq t \tag{1}$$

We designed the hand-based feature f_h to pay attention to hand motion signals as all the gestures are performed by the hands. In particular, we consider only the y-coordinate of the hand joint locations and hip joint locations. We first compute the y-coordinate differences between hand joint and hip joint:

$$\delta_{hh,i} = max(\mid y_{jr,i} - y_{jh,i} \mid, \mid y_{jl,i} - y_{jh,i} \mid) \tag{2}$$

where jr, jl, jh represent the right hand joint, left hand joints, and hip joint, respectively. As the same gesture can be performed by either right hand or left hand, Equation 2 is able to achieve the invariance under different hand performances. To capture the motion property of the hand joints, the hand-based features $f_{h,t}$ of frame t are defined as y-coordinate differences between hand joint and hip joint of each frame within the sliding window s_t:

$$f_{h,t} = \{\delta_{hh,i} \mid i \in [t-6, t+6]\} \tag{3}$$

Skeletal Feature Classifier We extract the motion feature f_c and hand-based feature f_h from each frame of the input sample. In our implementation, each frame is represented by 13-frame-long sliding window where the frame is at the center of the window. We annotate each frame with a label, either *gesture* frame or *non-gesture* frame according to the annotation labels provided by the

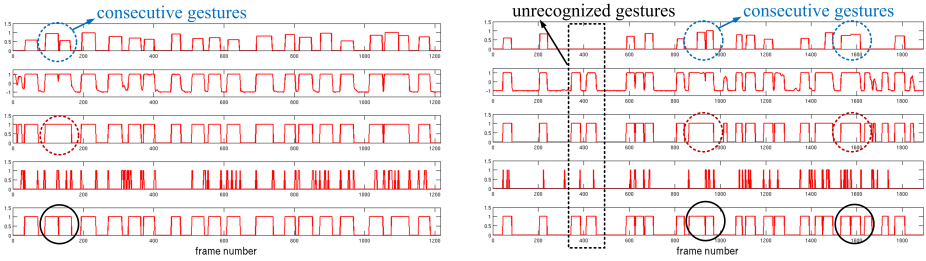


Fig. 2. Segmentation result of Sample 701 (left side) and Sample 707 (right side) in the testing dataset. From top to down: ground truth label of the samples; the labeled results of the SVM models; initial segmentation results; the labeled results of the SVM models for dealing with the consecutive gestures; final segmentation results of the samples

training dataset. However, as the unrecognized gestures in the training dataset were mislabled as *non-gesture* frames, we choose the y-coordinate differences $\mid y_{jr,i} - y_{jl,i} \mid$ between right hand joint jr and left hand joint jl to filter out the false *non-gesture* frames. Any *non-gesture* frame which has the y-coordinate differences $\mid y_{jr,i} - y_{jl,i} \mid$ above a specified threshold are removed from the training data. To eliminate the effect of different sizes of the performers, we train a two-class SVM model for each input sample of the training dataset and validation dataset, in total, having 700 SVM models. Finally, we select a subset of SVM models to maxmize the performance of the validation dataset by following a greedy SVM model selection procedure (see Fig. 1).

3.2 Dealing with consecutive gestures

Normally, each sample includes between 10 and 20 candidate gestures. Most of them are typically separated by *long-pauses* (e.g., the *long-pause* contains tens of *nongesture* frames), but some of them are consecutive gestures (e.g., separated by *short-pause* containing less than 2 frames, which is indicated by the blue dash circles in Fig. 2). The above SVM models may classify the *non-gesture* frames of *short-pause* as *gesture* frames (indicated by the red circles in Fig. 2). To tackle this problem, we train a new SVM model to classify the frames of candidate gestures as *consecutive frame* or *nonconsecutive frame*. To get the training data of the new model, we scan all the samples in the training and validation dataset and find the consecutive gestures where two adjacent gestures are separated by a *short-pause*. We manually annotate the frames in the *short-pause* as *nonconsecutive frame* and the frames in the adjacent gestures as *consecutive frame*. We then train the SVM model based on the labeled training data. For the frames in the candidate gesture, if two consecutive frames are labeled as *consecutive frames* by the new SVM model, we divide the candidate gesture into another two candidate gestures further (indicated by the black circles in Fig. 2). Fig. 2 shows the segmentation results of Sample 701 and 707 in the testing dataset.

4 Gesture Classification

The segmentation results cannot separate the pre-defined gestures from the unrecognized gestures (indicated by the black rectangles in Fig. 2). Thus, during the gesture classification phase, we will perform the classification of 21 classes of gestures (20 pre-defined gestures plus unrecognized gesture) instead of 20 classes of pre-defined gestures. We first explore a 3D dense representation of each candidate gesture. Dense features have shown the advantages in classifying human activities [17]. Inspired from [17], we combine discriminative training and randomization to obtain an effective classifier with good generalizability. This allows us explore a richer feature set efficiently as well as identifies semantically meaningful video blocks that closely match human intuition.

4.1 Spatio-temporal Feature Extraction

We extract spatio-temporal features from four channels (grayscale, gradient, depth, surface normal) of RGB-D video data by using Independent Subspace Analysis (ISA) algorithm [11]. ISA is a popular unsupervised learning algorithm that learns spatio-temporal features from unlabeled video data. An ISA network [11] is described as a two-layer neural network, with square and square-root nonlinearities in the first and second layers respectively. We start with any input subvolume $x^t \in \mathbb{R}^n$ (each subvolume is a sequence of image patches). The activation of each second layer unit is

$$p_i(x^t; W, V) = \sqrt{\sum_{k=1}^{m} V_{ik} (\sum_{j=1}^{n} W_{kj} x_j^t)^2} \tag{4}$$

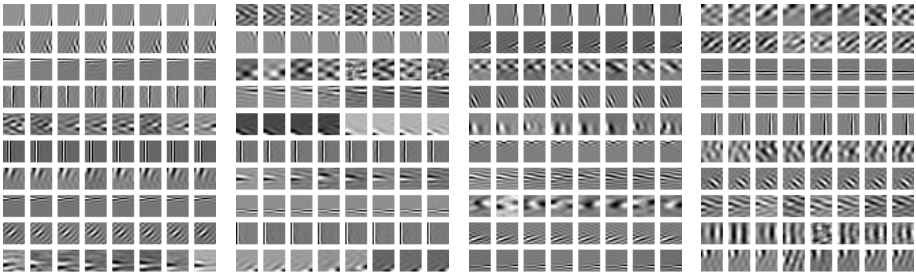
where i is the indicator of the activation of the second layer unit; $j = 1, ..., n$; $k = 1, ..., m$; n and m are the dimension of input unit x^t and the number of units in the second layer, respectively.

ISA learns the parameters W by finding sparse feature representations in the second layer, by solving

$$\begin{aligned} &\min_{W} \sum_{t=1}^{T} \sum_{i=1}^{m} p_i(x^t; W, V) \\ &s.t.\ WW^T = \mathbf{I} \end{aligned} \tag{5}$$

Here, $W \in \mathbb{R}^{u \times n}$ denotes the weights connecting the input units to the first layer units (u denotes the number of units in the first layer); $V \in \mathbb{R}^{m \times u}$ denotes the weights connecting the first layer units to the second layer units (V is typically fixed to represent the subspace structure of the neurons in the first layer); T is the number of the input units x^t. The orthonormal constraint is to ensure the features are sufficiently diverse.

One advantage of unsupervised feature learning is that it readily applies to novel data, such as grayscale and gradient magnitude video data from an RGBD-camera. We learn spatio-temporal features up to four channels of RGB-D video data: grayscale, gradient, depth, and surface normal (z-axis). The learned features are visualized in Fig. 3. These features are interesting to look at and



(a) Grayscale (b) Depth (c) Gradient (d) Surace Normal

Fig. 3. Visualization of randomly selected spatio-temporal features learned from four channels of the RGB-D video data - from left to right, grayscale, depth, gradient magnitude, Z surface normal component. Each row of the figure indicates a spatio-temporal feature

share some similarities. For example, the learned feature (each row of the sub-figure) is able to assign similar patterns into a group, and has sharper edges like Gabor filters.

4.2 Dense Sampling Spatio-temporal Space

Our approach aims to identify discriminative spatio-temporal blocks that are useful for the gesture classification. For example, in order to recognize whether a human is performing the gesture "*Ok*", we want to use the spatio-temporal blocks surrounding the human hands that are closely related to the gesture "*Ok*". We need to identify not only the spatial position of this kind of blocks (the image coordinate of the blocks) but the temporal position of the blocks (the start and end timestamps of the blocks). An algorithm that can reliably locate such regions is expected to achieve high classification accuracy. We achieve this goal by searching over spatio-temporal blocks with arbitrary spatial size, temporal size, and the 3D position of the blocks. We refer to this extensive set of spatio-temporal blocks as the *dense sampling spatio-temporal space*. Considering blocks with arbitrary spatial and temporal sizes, the actual density of spatio-tempral blocks is significantly higher. Richer feature indeed provide enough information for the classification task, however, many spatio-temporal blocks are not discriminative for distinguishing different gesture classes. Additionally, dense sampling introduces many overlapped spatio-temporal blocks which introduces significant redundancy. Therefore, it is challenging to explore this 3D dimensional, noisy and redundant feature space. In this work, we address this problem using the idea of combining discrimination and randomization.

4.3 Discriminative Random Forest Framework

In order to explore the 3D dense sampling feature space for the gesture classification, we combine two ideas: 1) Discriminative training to extract the information

in the spatio-temporal blocks effectively; 2) Randomization to explore the 3D dense feature space efficiently. Specifically, we adopt a random forest framework [17, 2] where each tree node is a SVM classifier that is trained on one spatio-temporal block.

Introduction of Random Forest Framework A random forest is a multi-class classifier consisting of an ensemble of decision trees where each tree is constructed via some randomization. As illustrated in Fig. 4, the leaf nodes of each tree encode a distribution over the gesture classes. All internal nodes contain a binary classifier that splits the data into two parts and sends the two parts to its children nodes. The splitting is stopped when a leaf node is encountered. A candidate gesture is classified by descending each tree and combining the leaf distributions from all the trees of the forest. This method allows the flexibility to explore a rich feature space effectively because it only considers a small subset of features (e.g., several hundreds of spatio-temporal blocks sampled from the video data) in every tree node.

Each tree returns the posterior probability of a test example belonging to the given classes. The posterior probability of a particular class at each leaf node is learned as the proportion of the training videos (each training video contains one gesture) belonging to that class at the given leaf node. The posterior probability of class c_m at leaf l of tree t is denoted as $P^m_{t,l}(c_m)$, where m means the type of the modality used in the representation of video data. Thus, a test candidate gesture can be classified by averaging the posterior probability from all the trees of the forest:

$$\hat{c}_m = \arg\min_{c_m} \frac{1}{T} \sum_{t=1}^{T} P^m_{t,l_f}(c_m) \tag{6}$$

where $\hat{c}_m$ is the predicted labeled using the modality data m, T is the number of the trees of the forest, and l_f is the leaf node that the testing video falls into. To fuse multi-modality data in the random forest framework, we utilize late fusion to classify the test candidate gesture:

$$\hat{c} = \arg\min_{c} \frac{1}{M \times T} \sum_{t=1}^{T} \sum_{m=1}^{M} P^m_{t,l_f}(c_m) \tag{7}$$

where $\hat{c}$ is the predicted labeled using the multi-modality data, M is the number of the types of the modality data.

Sampling the Dense Spatio-temporal Feature As shown in Fig. 4, each internal node in the decision tree corresponds to a set of spatio-temporal video blocks that are sampled from the 3D dense sampling space (Section 4.2), where the spatio-temporal blocks can have many possible spatio-temporal size and spatio-temporal positions. In order to sample candidate spatio-temporal blocks, we first normalize all videos to unit width, height and temporal dimension, and then randomly sample (x_l, y_l), (x_r, y_r) and (t_s, t_e) from a uniform distribution $U([0, 1])$. The coordinates (x_l, y_l) and (x_r, y_r) specify two diagonally opposite

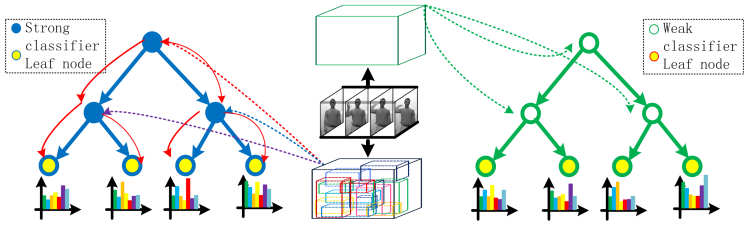


Fig. 4. Comparison of our discriminative decision tree (Left side of the figure) with conventional random decision tree (Right side of the figure). Conventional decision trees use information from the entire video data at each node, which encodes no spatio-temporal information. Our decision trees sample the spatio-temporal blocks from the dense sampling space. The histograms below the leaf nodes illustrate the posterior probability distribution. Our approach use strong classifiers (SVM) in each node, while the conventional method uses weak classifiers.

vertices of the spatial region of the block. The coordinates (t_s, t_e) specify the start and end position along the temporal dimension of the block. Such blocks could correspond to small area of the gesture segment or even the complete gesture segment. This allows the method to capture both the global and local information in the video.

In our approach, each saptio-temporal block is represented by a histogram of spatio-temporal features. The features are augmented with the decision value $w^T f$ (described in Equation 8) of this video segment from its parent node (indicated by the red lines in Fig. 4). Therefore, the feature representation combines the information of all upstream tree nodes that the corresponding video segment has descended from.

Learning the binary classifier of the tree node We describe the process of learning the binary splits of the data using SVM. This is achieved in two steps: 1) Randomly assigning all segments from each class to a binary label; 2) Using SVM to learn a binary split of the data. Assume that we have C classes of gesture segments at a given node. We uniformly sample C binary variables. We then assign all sampled blocks of a particular class c_i a binary class label. As each node performs a binary split of the data, this allows us to learn a simple binary SVM at each node. Using the feature representation f of an spatio-temporal block, we find a binary split of the data:

$$score = w^T f, \begin{cases} score \leq 0, \ go\ to\ left\ child \\ score > 0, \ go\ the\ right\ child \end{cases} \tag{8}$$

where w is the set of weights learned from a linear SVM. We evaluate each binary split that corresponds to a spatio-temporal blocks with the information gain criteria [1], which is computed from the complete training video segments that fall at the current tree node. The splits that maximize the information gain are selected and the splitting processing is repeated with the new splits of the

data. The tree splitting stops if a pre-defined tree depth or a minimum number of samples in the current node has been reached, or the information gain of the current node is larger than a specified threshold.

4.4 Pre-processing and Implementation Details

Pre-processing of the RGB-D video data It is our observation that gestures only relate to upper body movement of the performers. Within the performance of the gestures of each sample, there is little movement of the lower part of the body, especially the foot movement. Therefore, we cut out part of the video data containing only the upper body of the performers from the entire video data. During the gesture classification phase, we extract spatio-temporal features from this partial video instead of the complete video in each sample. We resize this partial video to a fixed spatial size video of 200×200. For the learning of the binary split of the tree node, the randomly sampled spatio-temporal blocks of different gesture segments should have the same spatio-temporal size and spatio-temporal positions. However, the temporal dimension of gesture segments is different. We therefore employed time normalization for the temporal alignment of all gesture segments. We apply the max pooling along the temporal dimension of the dense sampling feature space of the gesture segments. All the gesture segments are normalized to have a fixed temporal size.

Implementation details We densely extract four types of ISA features (Gray-ISA, Depth-ISA, Gradient-ISA, and Normal-ISA feature) on each gesture segment with a spatial spacing of 2 pixels and a temporal spacing of 2 frames. Using k-means clustering, we construct a vocabulary of codewords for each modality. Then we use Locality-constrained Linear Coding [16] to assign the spatio-temporal features to codewords. A bag-of-words histogram representation of the spatio-temporal blocks is used if the spatial size and temporal size of the blocks are smaller than 0.2, while a 2-level spatial pyramid is used if the spatial size of the block is between 0.2 and 0.9. We limit the maximum spatial size and temporal size to 0.9 and 0.8 respectively. For each tree of the forest, we sample 150 spatio-temporal blocks in the root node and the first level nodes respectively, and sample 200 spatio-temporal blocks in all other nodes. Sampling a smaller number of blocks in the root can reduce the correlation between the resulting trees. In total, we have trained 100 trees for each type of ISA features.

5 Results and Discussion

In this section, we present the experimental results to evaluate the performance of our approach. We use the training set and validation set as the final training dataset, and the testing set as the final testing dataset. To best understand the classification performance of our approach, we use the ground truth labels to segment the testing dataset instead of the predicted gesture segmentations.

Table 1. Mean average precision (map) and classification accuracy (acc) on the testing dataset. The Gray-ISA-Drf, Depth-ISA-Drf, Gradient-ISA-Drf, Normal-ISA-Drf and Fusion model were represented by Gray, Depth, Gradient, Normal, Fusion in this table, respectively. Each column shows the results obtained from one model. The last row of the table shows the mean results of the 20 pre-defined gesture classes. The best result is highlighted in bold

Gesture	Gray		Depth		Gradient		Normal		Fusion	
	map	acc	map	acc	map	acc	map	acc	map	acc
vattene	**90.1**	76.4	88.2	73.6	87.2	78.1	81.8	68.5	88.5	**78.6**
vieniqui	**92.1**	**85.7**	87.6	79.1	86.6	75.3	79.0	68.1	90.0	81.3
perfetto	**94.7**	**94.4**	92.7	89.3	92.8	90.4	90.1	87.6	93.1	90.4
furbo	**97.8**	90.4	91.5	92.7	95.9	92.1	90.3	91.6	96.4	**93.8**
cheduepalle	**99.6**	97.7	**99.6**	97.7	99.1	96.5	99.2	**98.8**	**99.6**	98.3
chevuoi	96.1	**89.9**	**96.4**	85.9	93.5	83.8	94.3	81.3	**96.4**	86.9
daccordo	99.3	99.4	99.2	97.5	98.8	98.8	97.7	93.9	**99.6**	**100**
seipazzo	97.5	**96.2**	96.8	92.4	96.3	90.3	94.8	90.8	**97.6**	94.6
combinato	**99.0**	98.3	97.3	97.3	98.5	97.3	98.1	97.3	98.8	**98.4**
freganiente	**92.7**	**88.2**	87.4	75.9	88.2	82.9	81.1	67.0	90.8	82.4
ok	**88.4**	**74.1**	81.5	60.9	84.8	65.5	79.6	40.2	88.0	66.7
cosatifarei	**96.4**	92.0	95.9	90.4	94.8	91.5	92.5	89.4	**96.4**	**92.6**
basta	99.8	98.3	99.8	**99.4**	99.7	98.3	99.6	98.8	**99.8**	98.9
prendere	**93.1**	82.1	89.5	81.5	91.4	**83.7**	84.2	72.3	91.9	80.4
noncenepiu	**83.8**	68.6	75.2	70.9	76.0	60.5	62.9	60.5	80.2	**71.5**
fame	99.0	97.8	99.0	97.8	98.7	94.6	98.3	97.8	**99.1**	**98.4**
tantotempo	**99.0**	96.5	96.9	95.4	98.5	96.5	96.1	96.5	98.4	**97.7**
buonissimo	**94.0**	**92.1**	85.7	77.5	88.7	82.6	83.5	75.8	90.5	84.8
messidaccordo	96.8	94.4	97.4	95.6	94.1	96.7	95.4	91.7	**97.8**	**97.8**
sonostufo	98.0	93.7	98.8	93.7	96.8	85.7	95.6	89.1	**98.8**	**94.3**
	95.3	**90.3**	92.8	87.2	93.0	87.1	89.7	82.9	94.6	89.4

We train our models on 10000 gesture segments of the training dataset, and perform the classification task of 21 gesture classes (20 pre-defined gesture classes and one unrecognized gesture class) on 3579 gesture segments in the testing dataset. However, we only show the results of the 20 pre-defined gesture classes, because the ground truth of testing dataset only provide the annotation of the 20 pre-defined gestures. We used four channels (grayscale, depth, gradient magnitude, surface normal) of the RGB-D video data to train the spatio-temporal features and the discriminative random forest models. Finally, we use four types of spatio-temporal features (Gray-ISA, Depth-ISA, Gradient-ISA, Normal-ISA), and four RF models (Gray-ISA-Drf, Depth-ISA-Drf, Gradient-ISA-Drf, Normal-ISA-Drf) where each model contains 100 decision trees. We also utilize a fusion model which uses a simple late fusion strategy by combining the likelihood scores of the above four RF models.

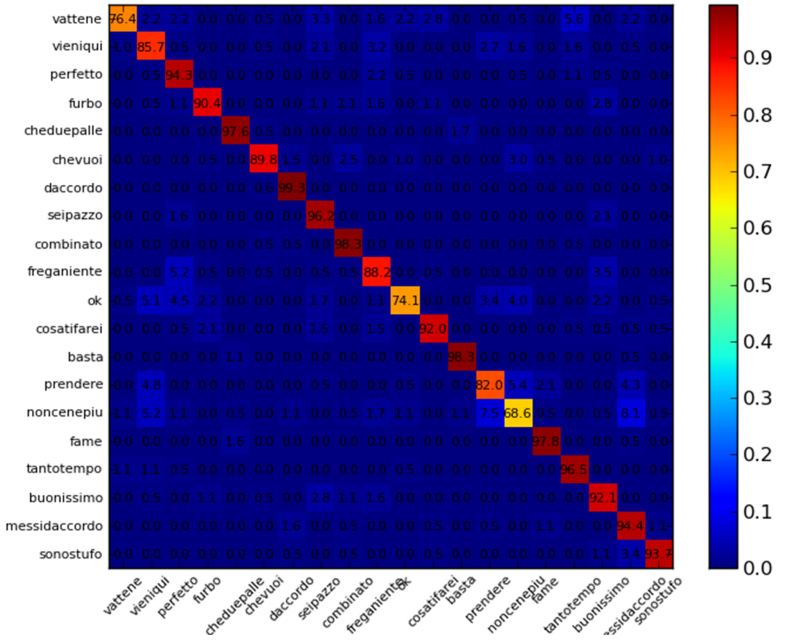


Fig. 5. The fusion matrice on the testing dataset using the the Gray-ISA-Drf model. Rows represent the actual gesture classes, and columns represent predicted classes (best viewed in color)

The classification results measured by mean average precision (*map*) and average accuracy (*acc*) are shown in Table 1. The Gray-ISA-Drf model achieves the best result on the average *map* (95.3%) and *acc* (90.3%) of 20 gesture classes. Note that we achieved this accuracy using very-low resolution videos (200 *pixels* $\times$ 200 *pixels*). In detail, the Gray-ISA-Drf model and fusion model achieve the best result on seven and ten out of 20 classes respectively. While the performance based on the Gray-ISA-Drf/Depth-ISA-Drf/Gradient-ISA-Drf models is promising, the accuracy of the Normal-ISA-Drf model is relative low. This is probably because the process of down-sampling depth video to a lower resolution loses some important information of surface normals. In addition, the fusion model decreased the performance compared with the Gray-ISA-Drf model. It is expected to achieve a better performance by investigating different fusion strategies (e.g., different combination of single models, fusion before training the random forest model). Fig. 5 is the visualization of confusion matrix of the Gray-ISA-Drf model. We can see that 12 out of 20 gesture classes achieved a result of $> 90\%$ accuracy. This is a good performance considering that we use single spatio-temporal feature, without using any hand-engineering spatio-temporal features or skeleton-based feature (for classification task).

In Fig. 6a, we visualize the 2D heat maps of the dominant positions of the first 40 gesture segments in the testing dataset. The 2D heat maps show the distribution of the discriminative positions discovered by our approach for the

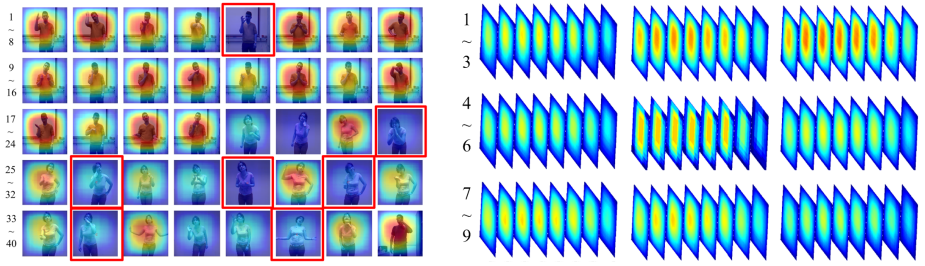


Fig. 6. (a) The 2D heat maps of the dominant positions of the first 40 gesture segments in the testing dataset. Red rectangles mean the mis-classified gesture segments. Red indicates high frequency and blue indicates low frequence. (b) The 3D heat map of the dominant spatio-temporal positions of the first 9 gesture segments in the testing dataset. To achieve a better visualization, we mapped each 3D heat map to a sequence of 2D heat maps where the timestamps of the heat maps range from the start to the end of the gesture segment (best viewed in color)

specific gesture segment. These maps are obtained by aggregating the spatial regions of the spatio-temporal blocks of all the tree nodes in the random forest weighted by the probability of the corresponding gesture class. We can see the difference of distributions for different gesture classes. We observe that they show semantically meaningful locations of where we would expect the discriminative regions of subjects performing different gestures to occur. For example, the regions of corresponding to the hand joint of the human body are usually highlighted. We can also see that the regions corresponding to background or irrelevant joints (e.g., head, hip center) are not frequently selected.

In Fig. 6b, we visualize the 3D heat maps of the dominant spatio-temporal positions of the first 9 gesture segments in the testing dataset. The 3D heat maps are obtained by aggregating the spatio-temporal space of the spatio-temporal blocks of all the tree nodes in the random forest weighted by the probability of the corresponding gesture class. To achieve a better visualization, we mapped the 3D heat maps to a sequence of 2D heat maps where the timestamps of the heat maps range from the start of the gesture segment to the end of the gesture segment. From Fig. 6b, we can clearly see that the different timestamps of the gesture segment have completely different heatmaps of the dominant positions. This means that, at different phases of a gesture, we would expect the different discriminative regions of the subjects performing gesture to occur. In addition, the 3D heat maps show three distinct phases of a gesture: pre-stroke, nucleus, post-stroke (see Fig. 6b), which correllates with the description in previous research on hand gesture [13]. The pre-stroke corresponds to the subject moving from the resting posture to the initial posture, which matches the start-phase of our 3D heat maps. During this phase, the spatio-temporal spaces are not frequently selected by our model (indicated by the blue space in the start-phase of the 3D heat map). The nucleus corresponds to the actual gesture performed by the subject, which matches the middle phase of our 3D heat maps (indicated by

the red space in the middle part of the 3D heat map). Post-stroke corresponds to the hand motions from the end of the gesture back to the resting posture, which matches the end of the 3D heat maps (indicated by the blue space at the end of the 3D heat map).

6 Conclusion

Gesture detection and recognition has a wide variety of applications and the *2014 ChaLearn Looking at People* Challenge serves as an important benchmark of the state-of-the-art in this field. We present our multi-modality gesture detection and recognition system in this paper. Our system utilizes the random forest framework with discriminative decision trees to discover spatio-temporal blocks that are highly discriminative for gesture recognition tasks. We show that our method identifies semantically meaningful spatio-temporal blocks that closely match human intuition. Though the proposed system achieves fairly good performance as indicated by the Jaccard Index of 0.6489 in the final evaluation phase, this performance is still far from what was achieved by the top-ranked team in the challenge. We also observe that the good classification performance (a 90.3% accuracy) of our system failed to bring a high Jaccard Index score. We have identified a few areas of improvement in future, especially in the case of designing better features for gesture detection, in the case of mining mid-level information from the gesture segments and in the case of rejecting unrecognized gestures. We expect that the implementation of these changes will improve the accuracy of our system substantially.

References

1. Bosch, A., Zisserman, A., Muoz, X.: Image classification using random forests and ferns. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. pp. 1–8 (Oct 2007)
2. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
3. Chen, G., Clarke, D., Knoll, A.: Learning weighted joint-based features for action recognition using depth camera. In: International Conference on Computer Vision Theory and Applications (2014)
4. Chen, G., Giuliani, M., Clarke, D., Knoll, A.: Action recognition using ensemble weighted multi-instance learning. In: IEEE International Conference on Robotics and Automation (2014)
5. Chen, G., Zhang, F., Giuliani, M., Buckl, C., Knoll, A.: Unsupervised learning spatio-temporal features for human activity recognition from rgb-d video data. In: International Conference on Social Robotics (2013)
6. Escalera, S., Bar, X., Gonzlez, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce, V., Escalante, H.J., Shotton, J., Guyon, I.: Chalearn looking at people challenge 2014: Dataset and results. In: Proceedings of the ChaLearn Looking at People 2014 workshop. ECCV'14 (2014)
7. Escalera, S., Gonzlez, J., Bar, X., Reyes, M., Lops, O., Guyon, I., Athitsos, V., Escalante, H.J.: Multi-modal gesture recognition challenge 2013: Dataset and results. In: Chalearn Multi-Modal Gesture Recognition Workshop, International Conference on Multimodal Interaction (2013)
8. Gaschler, A., Huth, K., Giuliani, M., Kessler, I., de Ruiter, J., Knoll, A.: Modelling state of interaction from head poses for social Human-Robot Interaction. In: ACM/IEEE HCI Conference on Gaze in Human-Robot Interaction Workshop (2012)
9. Hadfield, S., Bowden, R.: Hollywood 3d: Recognizing actions in 3d natural scenes. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3398–3405 (2013)
10. Laptex, I.: On space-time interest points. International Journal of Computer Vision 64, 107–123 (2005)
11. Le, Q., Zou, W., Yeung, S., Ng, A.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3361–3368 (2011)
12. Lu, D.V., Pileggi, A., Smart, W.D.: Multi-person motion capture dataset for analyzing human interaction. In: RSS 2011 Workshop on Human-Robot Interaction. RSS, Los Angeles, California (July 2011)
13. Pavlovic, V.I., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 677–695 (1997)
14. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3d action recognition with random occupancy patterns. In: Proceedings of the 12th European Conference on Computer Vision - Volume Part II. pp. 872–885. ECCV'12 (2012)
15. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1290–1297 (2012)
16. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3360–3367 (June 2010)

17. Yao, B., Khosla, A., Fei-Fei, L.: Combining randomization and discrimination for fine-grained image categorization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2011)