Image and Vision Computing xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

# Comparing random forest approaches to segmenting and classifying gestures ☆

Ajjen Joshi[a,*], Camille Monnier[b], Margrit Betke[a], Stan Sclaroff[a]

[a] Department of Computer Science, Boston University, Boston, MA 02215, USA
[b] Charles River Analytics, Cambridge, MA 02138, USA

ARTICLE INFO

*Article history:*
Received 1 October 2015
Received in revised form 30 May 2016
Accepted 5 June 2016
Available online xxxx

*Keywords:*
Gesture spotting
Gesture classification
Random forest classifier

ABSTRACT

A complete gesture recognition system should localize and classify each gesture from a given gesture vocabulary, within a continuous video stream. In this work, we compare two approaches: a method that performs the tasks of temporal segmentation and classification simultaneously with another that performs the tasks sequentially. The first method trains a single random forest model to recognize gestures from a given vocabulary, as presented in a training dataset of video plus 3D body joint locations, as well as out-of-vocabulary (non-gesture) instances. The second method employs a cascaded approach, training a binary random forest model to distinguish gestures from background and a multi-class random forest model to classify segmented gestures. Given a test input video stream, both frameworks are applied using sliding windows at multiple temporal scales. We evaluated our formulation in segmenting and recognizing gestures from two different benchmark datasets: the NATOPS dataset of 9600 gesture instances from a vocabulary of 24 aircraft handling signals, and the ChaLearn dataset of 7754 gesture instances from a vocabulary of 20 Italian communication gestures. The performance of our method compares favorably with state-of-the-art methods that employ Hidden Markov Models or Hidden Conditional Random Fields on the NATOPS dataset. We conclude with a discussion of the advantages of using our model for the task of gesture recognition and segmentation, and outline weaknesses which need to be addressed in the future.



## 1. Introduction

The problem of spotting and recognizing meaningful gestures has been an important research endeavor in the fields of computer vision and pattern recognition. Research in this domain has a broad scope of applications such as recognizing sign-language symbols, enabling video surveillance, establishing new idioms in gaming and entertainment, and developing new modes of human–computer interaction, among others.

A specific example of a gesture recognition application can be explored in the setting of a flight deck of an aircraft carrier. Deck officers use a vocabulary of gestures to communicate commands such as All clear, Move ahead, Turn left/right, and Slow down to aircraft pilots. However, the advent of unmanned air vehicles (UAV) has engendered the need to create a system capable of communicating the same set of commands to these unmanned aircrafts. Equipping a UAV with a computer vision system capable of accurately and automatically recognizing the existing set of gestures while they are being performed by deck officers would provide the most efficient solution to this problem, as it would permit the continued operation of the current method of communication.

Another example of an application in gesture recognition lies in the domain of understanding the context provided by communication gestures. Human beings communicate with words as well as gestures. A computer vision system capable of deciphering the gestures used in specific languages, such as Italian, can provide contextual information that aids the task of translating a foreign language.

It is important that the start and end points of a gesture be accurately identified in a temporal stream, in order to maximize the probability of correctly estimating the gesture label. One approach in solving the segmentation and classification problem involves separating them into two sub-problems where the task of segmentation precedes the task of recognition (Fig. 1). In this method, the focus is on first finding the gesture segmentation boundaries in time. The candidate gestures produced by the segmentation algorithm is then classified.

☆ This paper has been recommended for acceptance by Vitomir Štruc.

* Corresponding author.
*E-mail address:* ajjendj@bu.edu (A. Joshi).

http://dx.doi.org/10.1016/j.imavis.2016.06.001
0262-8856/

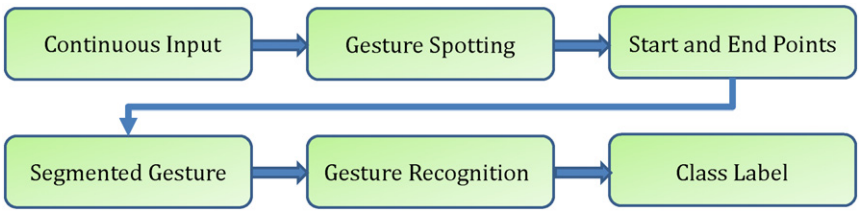


**Fig. 1.** Pipeline view of framework where gesture segmentation is followed by classification.

Another approach simultaneously performs the tasks of segmentation and classification (Fig. 2). In methods such as this, gesture intervals for which above-threshold scores are given by the classifier are deemed to be the labeled and segmented gesture. Given a training set of multi-modal videos with multiple examples of all gestures in a gesture vocabulary, we provide a comparison of the two approaches, highlighting the strengths and weaknesses of each.

With the advent of cameras capable of capturing depth information of a scene, gesture recognition datasets often contain 3D skeletal information of the user as well as intensity information from image frames. Designers of gesture recognition systems can therefore extract features from both skeletal as well as image data. In our random forest classification model, gestures are represented by a combination of both skeletal and image-based features. Because the classifier requires uniform-length feature descriptors for all gestures, we temporally divide the gestures into some number of segments, from which feature vectors are extracted and finally concatenated.

The key contributions of this work are:

- the comparison of a framework that employs a single multi-class random forest classification model to distinguish gestures from a given vocabulary in a continuous video stream with a framework that uses a cascaded approach,
- the fusion of joint-based and image-based features to create an accurate feature representation of gestures that is robust to variations in user height, distance of user to sensor and speed of execution of gesture, and
- the creation of a uniform feature descriptor for gestures to account for the variability in their length by division of gesture into a fixed number of temporal segments followed by the concatenation of the representative feature vectors of each temporal segment.

## 2. Related work

Here, we list and briefly explain some of the important methods that have been used in gesture recognition and are relevant to our work. A more comprehensive survey of gesture recognition techniques can be found elsewhere [1, 2].

Nearest neighbor models are often used in gesture classification problems. Malassiotis et al. [3] used a k-NN classifier to classify static sign language hand gestures. A normalized cross-correlation measure was used to compare the feature vector of an input image with those in the k-NN model. Dynamic Time Warping (DTW) can be used to compute a matching score between two temporal sequences, a variant of which was used by Alon et al. [4]. A drawback of k-NN models is the difficulty in defining distance measures that clearly demarcate different classes of time series observations.

A Hidden Markov Model (HMM) is another widely used tool in temporal pattern recognition, having been implemented in applications of speech recognition, handwriting recognition, as well as gesture recognition. Starner et al. [5] employed an HMM-based system to recognize American Sign Language symbols. One difficulty while implementing HMMs is to determine an appropriate number of hidden states, which can be domain-dependent.

The Conditional Random Field (CRF), introduced by Lafferty et al. [6] is a discriminative graphical model with an advantage over generative models, such as HMMs: the CRF does not assume that observations are independent given the values of the hidden variables. Hidden Conditional Random Fields (HCRFs) use hidden variables to model the latent structure of the input signals by defining a joint distribution over the class label and hidden state labels conditioned on the observations [7]. HCRFs can model the dependence between each state and the entire observation sequence, unlike HMMs, which only capture the dependencies between each state and its corresponding observation. Song et al. used a Gaussian temporal-smoothing HCRF [8] to classify gestures that combine both body and hand signals. They also presented continuous Latent Dynamic CRFs [9] to classify unsegmented gestures from a continuous input stream of gestures.

Random forest models perform well in many classification tasks, work efficiently on large datasets, and are very fast. Random forests have been applied to good effect in real-time human pose recognition [10], object segmentation [11], image classification [12], and sign language recognition [13] among others. Decision forest models have also been used variedly in gesture and action recognition tasks [14–18]. Miranda et al. [19] used a gesture recognition scheme based on decision forests, where each node in a tree in the forest represented a keypose, and the leaves of the trees represented gestures corresponding to the sequence of keyposes that constitute the gesture as one traverses down a tree from root to leaf. Demirdjian and Varri [20] proposed the use of temporal random forests in order to recognize temporal events. Camgöz et al. [18] use random forests to perform gesture spotting and classification. In contrast to our work, they perform frame-level gesture classification by training a model where every individual frame is considered a separate training sample. Randomized decision forests have been shown to be robust to the effects of noise and outliers. Moreover, they generalize well to

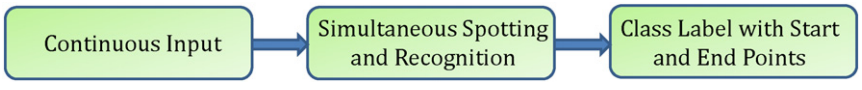


**Fig. 2.** Pipeline view of framework where gesture segmentation and classification is performed simultaneously.

Please cite this article as: A. Joshi, et al., Comparing random forest approaches to segmenting and classifying gestures, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.06.001

variations in data [21]. Thus, random forests are suitable for classification tasks involving data such as gestures because data collected by image and depth sensors can be sensitive to noise and their execution can exhibit a high level of variance.

More recently, deep learning approaches have gained popularity in gesture spotting and recognition tasks. Neverova et al. [22] present a gesture localization and recognition scheme based on multi-modal deep learning operating at various spatial as well as temporal scales. Pigou et al. [23] present an end-to-end neural network architecture incorporating temporal convolutions and bidirectional recurrence to perform gesture spotting and recognition.

Cameras equipped with depth sensors combined with skeleton detection algorithms enable researchers to use features extracted from 3D joint positions in gesture and action recognition problems. Yao et al. [24] used concatenated raw coordinates of body joints for gesture classification whereas Xia et al. [25] employed histograms of 3D joint locations for the task of human action recognition. Raptis et al. [26] formulated an angular representation of user skeletons as features for the problem of dance gesture recognition. In some problems, it is advantageous to include in the feature representation, information that 3D body joint locations are unable to capture, e.g. hand shape. The salient properties of hand shape can be captured using image-based features such as Histograms of Oriented Gradients (HOG) [27]. Song et al. [8] combined features extracted from images of the user hands with joint features to classify gestures.

## 3. System overview

Here, we describe in detail the formulation of both gesture recognition systems. We first explain the differences in the procedures used in training our random forest frameworks, and then illustrate how the classifiers are used to spot and classify gestures from a continuous stream. Overviews of training the two frameworks are depicted in Figs. 3 and 4 .

### 3.1. Training

The training set of gestures used in our experiments is labeled with true temporal segmentation as well as classification values. That is, each video sample used in training is associated with a file that describes the class labels of the gestures that are present in the video, along with their start and end frames.

#### 3.1.1. Simultaneous spotting and classification framework

Let $n$ be the number of different gestures that are present in the gesture vocabulary. We trained a $n + 1$-class random forest classifier using all examples of the $n$ different gestures in the training set, as well as some randomly selected examples of non-gestures (found in intervals between two gestures). Non-gestural examples may contain a sequence of gestural silence, that is when the user is relatively static, or they may contain non-gestural movements, that is when the user is moving or performing out-of-vocabulary gestures.

#### 3.1.2. Cascaded spotting and classification framework

For the cascaded framework, we trained a binary random forest classifier using all instances of the $n$ different gestures in the training set as positive examples and an equivalent number of randomly selected instances of non-gestures (found in intervals between two gestures) as negative examples. This binary classifier was used during test time to distinguish a gesture from the background. Additionally, we trained an $n$-class random forest classifier using all examples of the $n$ different gestures in the training set. This multi-class classifier was used during test time to predict the class label of a candidate gesture spotted by the binary classifier.

#### 3.1.3. Feature extraction

Each training example consists of a varying number of frames, each of which is described by a feature descriptor. In both frameworks, our system computes normalized positional and velocity features for nine different skeletal body joints (left and right shoulders, elbows, wrists and hands, as well as the head joint). Since gestures are performed by subjects with different heights, at different distances from the camera sensor, we first normalized the positional coordinates of the users' joints using the length of the user's torso as a reference. The normalized position vector for joint $j$ at time $t$ is:

$$\mathbf{W}_j(t) = \frac{\mathbf{W}^r_j(t) - \mathbf{W}^r_{hip}(t)}{l}, \tag{1}$$

where $\mathbf{W}^r_j(t)$ is the raw position vector for joint $j$ at time $t$, $\mathbf{W}^r_{hip}(t)$ is the raw position vector for the hip joint at time $t$, and $l$ is the length of the torso defined as:

$$l = \| (\mathbf{W}_{head} - \mathbf{W}_{hip}) \| . \tag{2}$$

Our system uses the normalized positional coordinates $(W_x, W_y, W_z)$ of these nine joints along with their rotational values $(R_x, R_y, R_z, R_w)$, which are provided with the dataset, and computes values for their velocities $\left(W'_x, W'_y, W'_z, R'_x, R'_y, R'_z, R'_w\right)$.

Thus, there are 126 feature descriptors extracted from 3D skeletal data for every frame. In addition, we augment our skeletal feature vector with HOG features on $32 \times 32$ pixel squares centered on the left and right hands. Each $32 \times 32$ pixel square window is divided

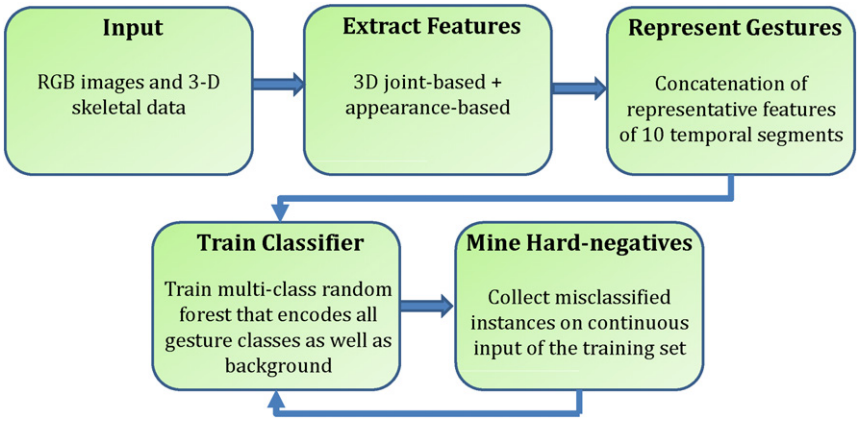


**Fig. 3.** Pipeline view of training our gesture recognition framework that performs simultaneous spotting and classification.

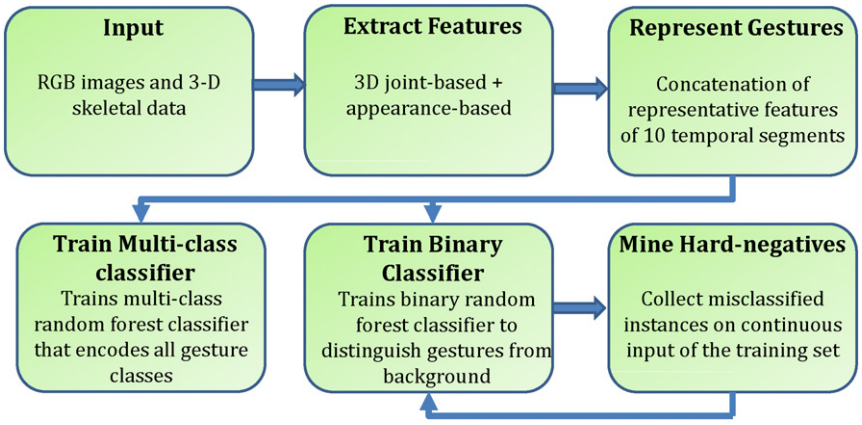


**Fig. 4.** Pipeline view of training our cascaded gesture recognition framework that first spots a gesture before classifying it.

into 4 × 4 cells. Each window is also divided into 3 × 3 overlapping blocks (each block contains 2 × 2 cells) to perform normalization.We obtained a dimensionality-reduced representation of the HOG features by performing Principal Component Analysis (PCA) and using the first 20 principal components for each hand. The first 20 components explained about half of the variance (0.44 % and 0.43 % for the left and right hands respectively) and were chosen so that the resulting feature space was a balanced combination of both the skeletal features obtained from joints as well as hand-appearance features obtained from HOG representations.Thus, every frame of every instance in our training set is represented by a 166 dimensional feature descriptor.

### 3.1.4. Gesture representation

In order to remove the effects of noisy measurements, we first smoothed all features using a moving average filter spanning 5 frames. Smoothing features slightly improved classification accuracy (an increase in classifier accuracy of 1.4% on a validation set on the NATOPS dataset). Because instances of gestures and non-gestures in our training set are temporal sequences of varying length, there arises the need to represent every gesture with a feature vector of the same length. We achieved this by dividing the gesture into 10 equal-length temporal segments, and representing each temporal segment with a vector of the median elements of all features. Using 10 temporal segments provided a balance between keeping the feature representation concise, while encapsulating enough temporal information useful in discerning the gesture classes. The representative vectors of each temporal segment were then concatenated into a single feature vector.

### 3.1.5. Random forest training

We defined the training set as $\mathcal{D} = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$. Here, $(\mathbf{X}_1, \ldots, \mathbf{X}_n)$ corresponds to the uniform-length feature vector representing each gesture or non-gesture, and $(Y_1, \ldots, Y_n)$ represents their corresponding class labels.

A random forest classification model consists of several decision tree classifiers $\{t(\mathbf{x}, \phi_k), k = 1, \ldots\}$ [21]. Each decision tree $t(\mathbf{x}, \phi_k)$ in the forest is constructed until they are fully grown. Here $\mathbf{x}$ is an input vector and $\phi_k$ is a random vector used to generate a bootstrap sample of objects from the training set $\mathcal{D}$. The ideal number of trees in our random forest model was determined to be 500 by studying the Out-of-Bag (OOB) error rate in the training data.

Let $d$ be the dimensionality of the feature vector of the inputs. At each internal node of the tree, $m$ features are selected randomly from the available $d$, such that $m < d$. $m = \sqrt{d}$ provided the highest accuracy among other common choices for $m$ $(1, 0.5\sqrt{d}, 2\sqrt{d}, d)$. From the $m$ chosen features, the feature that provides the most information gain is selected to split the node. Information gain $(I)$ can be defined as:

$$I_j = H(S_j) - \sum_{k\epsilon(L,R)} \frac{|S_j^k|}{|S|} H\left(S_j^k\right), \tag{3}$$

where $S_j$ is the set of training points at node $j$, $H(S_j)$ is the Shannon entropy at node $j$ before the split, and $S_j^L$ and $S_j^R$ are the sets of points at the right child and left child respectively of the parent node $j$ after the split.

The Shannon entropy can be defined as:

$$H(S) = -\sum_{c\epsilon C} p_c \log(p_c), \tag{4}$$

where $S$ is the set of training points and $p_c$ is the probability of a sample being class $c$.

We trained and saved a random forest classification model based on the features that we extracted. There is a need to strengthen the classifier's ability to accurately detect intervals of non-gestures because the randomly chosen intervals of non-gestural examples fail to fully model the class of non-gestures. In order to achieve this, we applied the random forest model on continuous input of the training set and collected false positives and false negatives, which are examples of intervals from the training set that the classifier fails to classify correctly. The set of false positive and false negative instances is then added to the original training set, and the random forest is re-trained using the new extended set of training examples. This process of bootstrapping, as performed by Marin et al. [28], is performed iteratively until the number of false positives is reduced below a threshold.

## 3.2. Testing

The task during testing is to use our trained random forest model to determine the temporal segmentation of gestures in a continuous video and accurately classify the segmented gesture. A sample test video contains a number of frames, and the same features collected during training are computed for every frame. Unlike training videos, test videos do not contain information about where gestures start and end. Therefore, we perform multi-scale sliding window classification to predict the class labels of the gestures, as well as their start and end-points.

Please cite this article as: A. Joshi, et al., Comparing random forest approaches to segmenting and classifying gestures, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.06.001

### 3.3. Multi-scale sliding window classification

We performed multi-scale sliding window classification to predict the class labels of the gestures, as well as their start and end points.

For each input video, gesture candidates were constructed at different temporal scales. Let $f_s$ be the number of frames in the shortest gesture in the training set and $f_l$ be the number of frames in the longest gesture in the training set. Then, the temporal scales ranged from length $f_s$ to length $f_l$, in increments of 5 frames. Let, $\mathcal{G} = \{g_1, \ldots g_n\}$ be the set of gesture candidates at different temporal scales. At each scale, a candidate gesture $g_i$ was constructed by concatenating the feature vectors at an interval specified by the temporal scale, so that the dimensions of the feature vector matched those of the gestures used to train the classification model.

Within a buffer of length larger than the longest temporal scale, a sliding window was used to construct gesture candidates at each temporal scale. For a buffer of size $b$, the number of gesture candidates at scale $s_i$ is equal to $b - s_i + 1$. We chose $b$ to be 100 frames, which is marginally greater than the maximum length of a gesture in the training set. Overviews of training the two frameworks are depicted in Figs. 5 and 6 .

#### 3.3.1. Simultaneous spotting and classification framework

Gesture candidates generated by the sliding window within the temporal neighborhood defined by the buffer at each scale were classified by our trained random forest model and competed to generate a likely gesture candidate $G_{s_i}$ at that scale. Since gesture candidates at the neighborhood of where the gesture is truly temporally located tend to be classified as the same gesture, we performed Non-Maxima Suppression to select the most likely gesture candidate.That is, for each scale $s_i$, $b - s_i + 1$ gesture candidates were generated and the one classified with the highest confidence $(G_{s_i})$ within a temporal neighborhood was selected. The confidence score is the percentage of decision trees that vote for the predicted class. Finally, the likely gesture candidates at the various scales competed to generate the final predicted gesture within the buffer.

Therefore, within the buffer, the scale of the final predicted gesture helps determine the segmentation boundaries of the gesture, whereas its class label is that which is predicted by the random forest classifier. The end point of the predicted gesture was chosen to be the start point of the new buffer. This process was then repeated until the end of the test video was reached.

#### 3.3.2. Cascaded spotting and classification framework

In our cascaded framework, the multi-scale sliding window mechanism outputted whether the gesture candidate was of the gesture or background class, instead of predicting the final class label. Non-overlapping candidates predicted as gestures by the upper-level binary classifier were then given their final gesture label by the multi-class random forest classifier.

#### 3.3.3. Evaluation

In order to evaluate the performance of our gesture spotting and classification frameworks, we use the Jaccard Index score. The Jaccard Index score, in the context of gesture spotting and recognition, is an intersection over union measure that incorporates the evaluation of the predicted gesture label as well as the predicted gesture start and end points [29] and is a common measure for such tasks [18, 22, 23]. For a given sequence of test frames that contains a gesture, the Jaccard Index score can be computed when the ground truth gesture label, the ground truth gesture start and end points, the predicted gesture label and the predicted gesture start and end points are given (as illustrated in Fig. 7).

## 4. Datasets

Here, we describe in detail the nature of the datasets we have used to test our gesture recognition system.

### 4.1. NATOPS

The Naval Air Training and Operating Procedures Standardization (NATOPS) gesture vocabulary comprises of a set of gestures used to communicate commands to naval aircraft pilots by officers on an aircraft carrier deck. The NATOPS dataset [30] consists of 24 unique aircraft handling signals, which is a subset of the set of gestures in the NATOPS vocabulary, performed by 20 different subjects, where each gesture has been performed 20 times by all subjects. Thus, each gesture has 400 samples. The samples were recorded at 20 FPS using a stereo camera at a resolution of 320 × 240 pixels. The videos were recorded in such a way that the position of the camera and the subject relative to the camera was fixed, and changes in illumination and background was avoided. The dataset includes RGB color images, depth maps, and mask images for each frame of all videos. A 12 dimensional vector of body features (angular joint velocities for the right and left elbows and wrists), as well as an 8 dimensional vector of hand features (probability values for hand shapes for the left and right hands) collected by Song et al. [30] was also provided for all frames of all videos of the dataset (Fig. 8).

### 4.2. ChaLearn

The ChaLearn dataset was provided as part of the 2014 Looking at People Gesture Recognition Challenge [31]. The focus of the gesture recognition challenge was to create a gesture recognition system

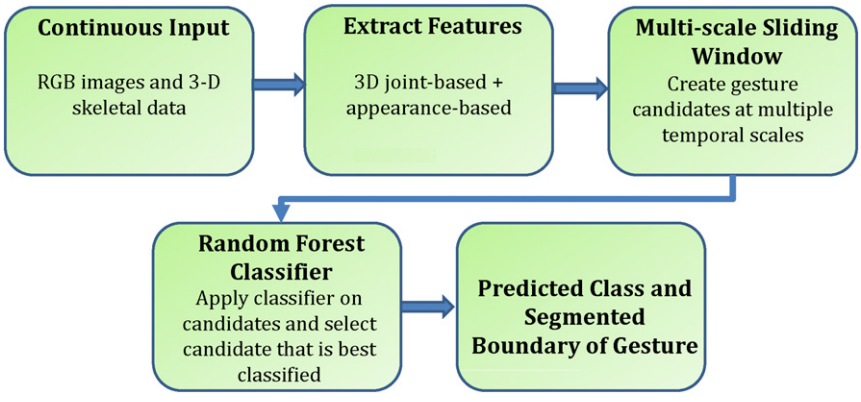


**Fig. 5.** Pipeline view of testing our gesture recognition framework that performs simultaneous spotting and classification.

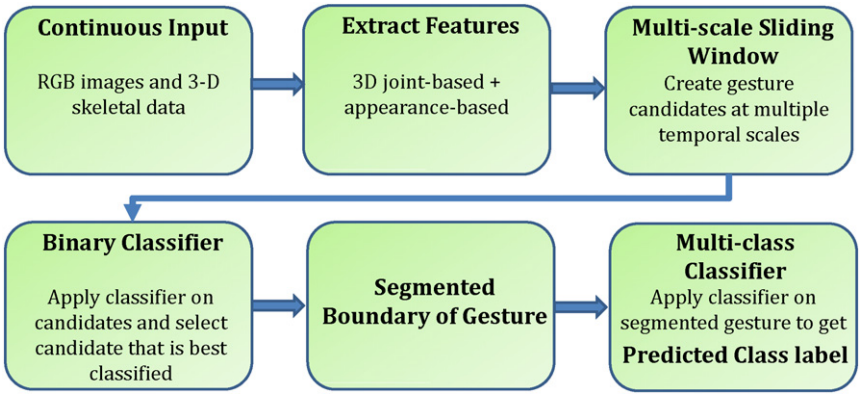


**Fig. 6.** Pipeline view of testing our cascaded gesture recognition framework that first spots a gesture before classifying it.

trained on several examples of each gesture category performed by various users. The gesture vocabulary contains 20 unique Italian cultural and anthropological signs. Gestural communication is a major part of communication in Italian culture, and developing systems to recognize such gestures is a task that can have many applications.

The development data used to train the recognition system contains a total of 7754 manually labeled gestures. Additionally, a validation set with 3363 labeled gestures was provided to test the performance of the trained classifier. During the final evaluation phase, another 2742 gestures were provided. The gesture examples are contained in several video clips. Along with the RGB data, depth data, user mask data along with skeletal information was also provided. Skeletal information was contained in a .csv file, where world coordinates, rotation values and pixel coordinates were provided for 20 different joints of the user in each frame of the video clip (Fig. 9).

## 5. Experiments

Here we describe the experiments performed to evaluate our gesture recognition system on the two datasets. We used the NATOPS dataset to evaluate our gesture classification system in a non-continuous setting. We used a set of gesture samples to train our gesture classifier, and tested its performance on a test-set of pre-segmented gestures. The ChaLearn dataset consists of training and test videos where the user performs both in-vocabulary and out-of-vocabulary gestures, with intervals of gestural silence or transitions. Thus, we used the ChaLearn dataset to test the performance of our system on continuous input.

The difference in evaluation metrics (we use Average Classification Accuracy for NATOPS and Jaccard Index for ChaLearn) is a consequence of the differences in the nature of the datasets. The NATOPS dataset consists of pre-segmented gesture examples, hence the primary task is to formulate methods to do gesture classification. The ChaLearn dataset consists of continuous videos where segments of gesture performance is interspersed with segments of non-gestures. Thus, the challenge is to both *spot* the gesture and *classify* the spotted gesture.

From the NATOPS dataset, we trained our gesture recognition model with the following features sets in order to formulate a good feature representation:

(a) 3D skeletal joints and hand-shape based feature set (SK + HS): This feature set [8] consists of 20 unique features for each timeframe for every gesture. The extracted features are angular joint velocities for the right and left elbows and wrists, as well as probability values of hand shapes for the left and right hands. Since each gesture instance is described by a single feature descriptor obtained by concatenating 10 representative feature vectors, the feature vector representing a gesture instance is of length 200.
(b) Appearance-based feature set (EOD): Each frame of the gesture instances is represented by a 400 dimensional feature vector, which was calculated using randomly pooled edge-orientation and edge-density features. Each gesture example is represented by a single-dimension feature vector of length 4000.
(c) EODPCA: In this feature representation, we reduced the above 4000-d feature space into a 200-d feature space via Principal Component Analysis (PCA).
(d) SK + HS + EODPCA: This feature set was obtained by concatenating the 200-d 3D skeletal joints and hand-shape based (SK + HS) feature descriptor of a gesture with the corresponding dimensionality-reduced edge orientation and density (EOD-PCA) feature descriptor to form a 400-d feature vector for every gesture.

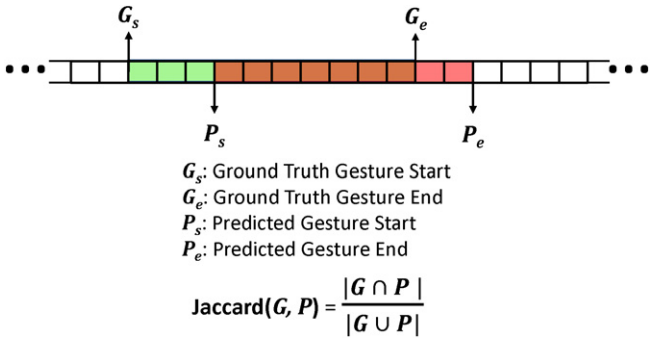


**Fig. 7.** An example illustration of the Jaccard score.

For each feature set described above, we trained random forests with 500 trees on 19 subjects and tested on the remaining subject in a leave-one-out cross-validation approach.

We computed the average recognition accuracy (averaged across all subjects and all gestures) of the random forest classifier on the four different feature sets (a) - (d) of the NATOPS dataset for all 20 test subjects each performing the 24 gestures in the vocabulary (Table 1). The feature set containing 3D skeletal joints and hand-shape features (SK+HS) is correctly classified 84.77% of the time, whereas the feature set containing features based on edge density and orientation is correctly classified 76.63% of the time. This suggests, in our case, that 3D joint-based based features encode more class-discerning information than features based on edge density and orientation. However, the highest classification accuracy of

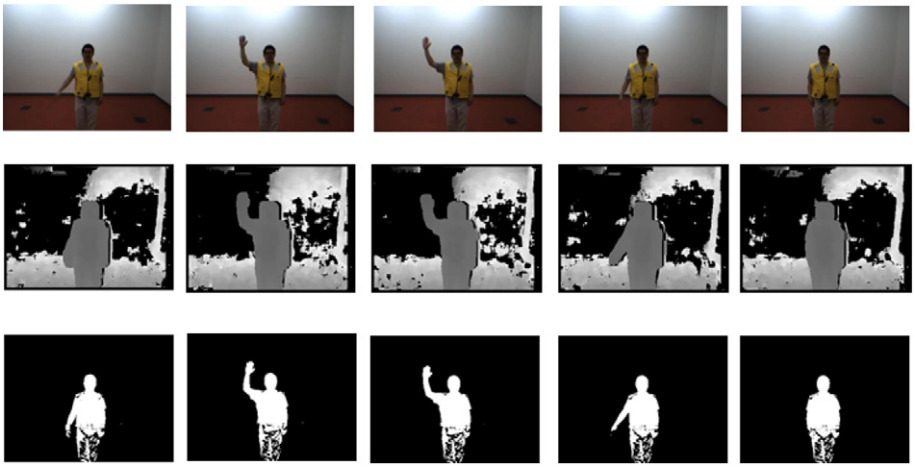

**Fig. 8.** RGB, Depth, and User-Mask Segmentation of a subject performing gesture 1 'I Have' in the NATOPS dataset.

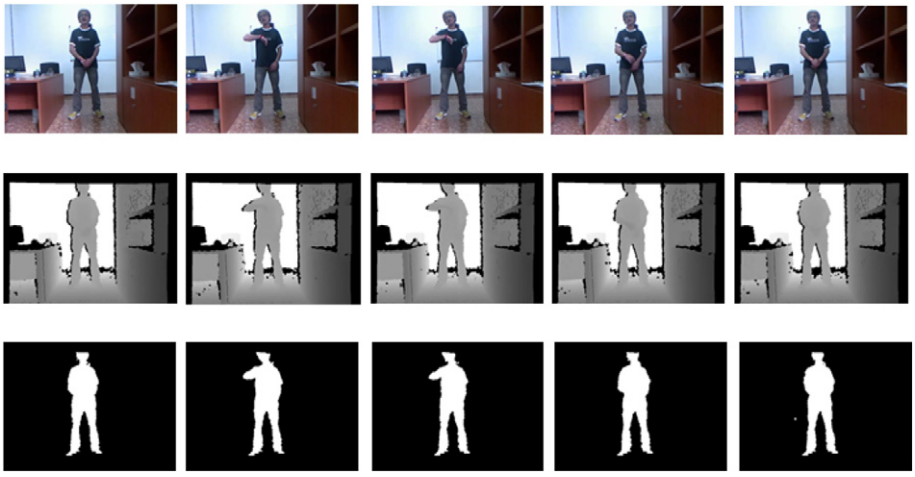

**Fig. 9.** RGB, Depth, and User-Mask Segmentation of a subject performing gesture 1 'sonostufo' in the ChaLearn dataset.

87.35% is achieved on the feature set that combines joint-based features with appearance-based features, suggesting the benefit of combining the two approaches of collecting features.

Gesture pairs (2,3), (10, 11) and (20, 21) were confused, often getting misclassified as the other (Fig. 10). Fig. 11 uses a confusion matrix to illustrate the misclassifications between these pairs of similar gestures.

We compared the classification performance of our random forest classifier with the performance of other classifiers that have been used on this dataset (Table 2). Our random forest approach on the challenging subset of similar gestures, tested on samples from 5 subjects as specified by Song et al. [32], yields results that exceeds those produced by the state-of-the-art (Linked HCRF) (Table 2). The graphical models presented by Song et al. [32] were trained using feature set a (SK+HS), whereas we use feature set d (SK+HS+EODPCA) to train our gesture recognition model.

From the ChaLearn dataset, we trained our gesture recognition model with the following feature sets:

**Table 1**
Average classification accuracy on all 24 gestures of the NATOPS dataset.

| Feature set | Average classification accuracy | Standard deviation across subjects |
|---|---|---|
| Feature set a (SK + HS) | 84.7% | 5.1 |
| Feature set b (EOD) | 76.6% | 8.4 |
| Feature set c (EODPCA) | 67.7% | 9.5 |
| Feature set d (SK + HS + EODPCA) | 87.3% | 4.9 |

(a) Raw 3D skeletal joint data (RAW): Features contain unedited raw skeleton data, that is, each frame consists of 9 values for all 20 joints. The feature vector per frame has 180 dimensions, and per gesture has 1800 dimensions.
(b) Normalized skeletal joint positions and velocities (SKPV): This feature set contains normalized positional and velocity data for 9 joints. The feature vector per frame has 126 dimensions, and per gesture has 1260 dimensions.
(c) Normalized skeletal joint positions, velocities and accelerations (SKPVA): This feature set contains positional, velocity, and acceleration data for 9 joints. The feature vector per frame has 189 dimensions, and per gesture has 1890 dimensions.
(d) SK + HOGPCA: This feature set was obtained by concatenating the 1260-d feature vector of normalized skeletal joint positions and velocities (SK) with the 400-d feature vector of HOG data for $32 \times 32$ pixel squares around the left and right

Please cite this article as: A. Joshi, et al., Comparing random forest approaches to segmenting and classifying gestures, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.06.001

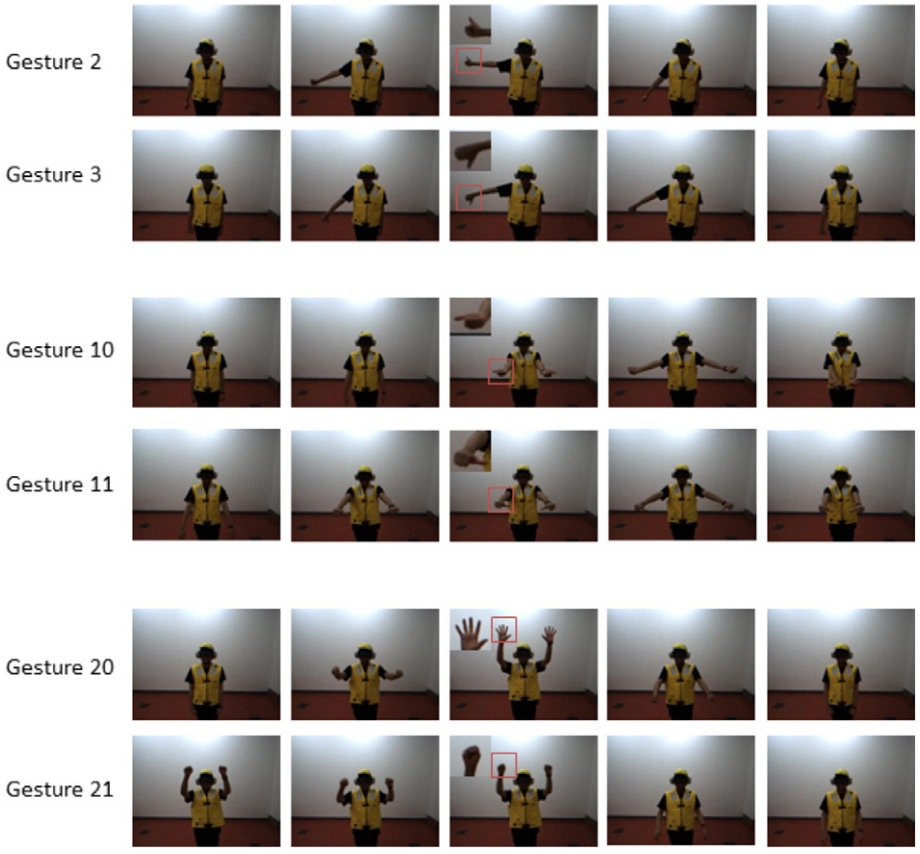


**Fig. 10.** Some pairs of similar gestures in the NATOPS dataset.

hands whose dimensionality has been reduced by PCA. The resultant feature vector per gesture example is 1660-d.

For each feature set described above, we trained random forests with 500 trees on gesture instances from the training and validation sets, and tested the performance of our classifier on the test dataset. The division of the data into training, validation and test sets has been described earlier [29].

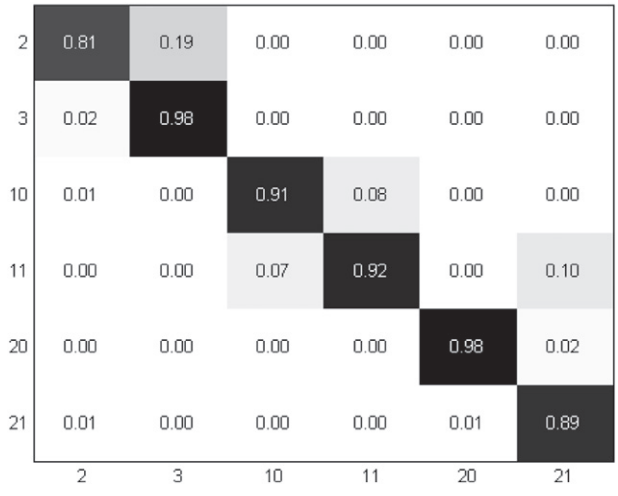


**Fig. 11.** Confusion matrix for pairs of similar gestures in the NATOPS dataset.

The feature set that combines the normalized positional and velocity information (SKPV), with HOG features of the hands (HOG-PCA), is correctly classified correctly 88.91% of the time (Table 3), which is the highest average classification accuracy of all feature sets.

The iterative procedure of training a random forest improves its capacity to correctly classify and segment gestures for both methods. This is evident in the increase in Jaccard scores on the training sets (Figs. 12 and 13 ).

Table 4 shows the Jaccard score of our method compared with the winning scores of the ChaLearn gesture recognition challenge. The competition winner used information from skeleton joints, intensity and depth videos in a deep neural network framework to achieve a Jaccard score of 0.84 [33] (subsequently improved to 0.87 [22]). Our classifier achieves a good recognition accuracy of 88.91% on pre-segmented gestures. One benefit of using a cascaded gesture spotting and classification framework is that it enables separate evaluations of the spotting and classification schemes. The framework which performs spotting and classification simultaneously achieves a Jaccard score of 0.68 whereas the cascaded framework that first spots a gesture before classifying it achieves a score of 0.72.

**Table 2**
Performance comparison on pairs of similar gestures in the NATOPS dataset with other approaches (The HMM, HCRF, and Linked HCRF) presented by Song et al. [32].

| Classifier | Average classification accuracy |
|---|---|
| HMM | 77.6% |
| HCRF | 78.0% |
| Linked HCRF | 87.0% |
| Random forest (our) | 88.1% |

Please cite this article as: A. Joshi, et al., Comparing random forest approaches to segmenting and classifying gestures, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.06.001

**Table 3**
Average classification accuracy on all 20 gestures of the ChaLearn dataset.

| Feature set | Average classification accuracy |
|---|---|
| Feature set a (RAW) | 81.4% |
| Feature set b (SKPV) | 88.1% |
| Feature set c (SKPVA) | 83.5% |
| Feature set d (SK + HOGPCA) | 88.9% |

## 6. Conclusion

Our method consists of first creating a uniform fixed-dimensional feature representation of all gesture samples, and then using all training samples to train a random forest. On a challenging subset of the NATOPS dataset, our approach yields results comparable to those produced by graphical models such as HCRFs. Although a random forest classifier does not explicitly model the inherent temporal nature of gestural data as done by graphical models, its performance in accuracy on this particular dataset exceeds that achieved by graphical models such as HMMs, and different variants of HCRFs, which are presented by Song et al. [32]. Additionally our experiments also show that classification accuracy was improved by combining 3D skeletal joint-based features with appearance-based features, thus underlying the importance of a well-chosen feature set for a classification task.

Although a simple approach has yielded good results with this dataset, there are areas where improvements can be made. As our results have shown, there are some gestures in both datasets that are easily confused and hence hinder the ability of the classifier to achieve maximum accuracy. For example, some gesture pairs (e.g. (2, 3), (10, 11)) in the NATOPS dataset are very similar in structure and therefore there are several instances where the classifier misclassifies one as the other. For example, gestures 2 and 3 in the dataset have the same hand movements and only differ in hand-shape (gesture 2 is performed with a thumbs-up hand-shape, while gesture 3 is performed with a thumbs-down). Although probability measures for hand-shapes are part of the feature description for each gesture, the probability of their selection during tree construction is low due to the randomized nature of selecting features while building individual decision trees. A possible fix for this problem is to modify the process of feature selection during tree-building by encoding a weighting scheme that emphasizes the selection of more discriminative features. Another approach would involve classifying sub-gestural units as done by Cooper and Bowden [34], and then use some Markov chain structure to classify the hand gesture.

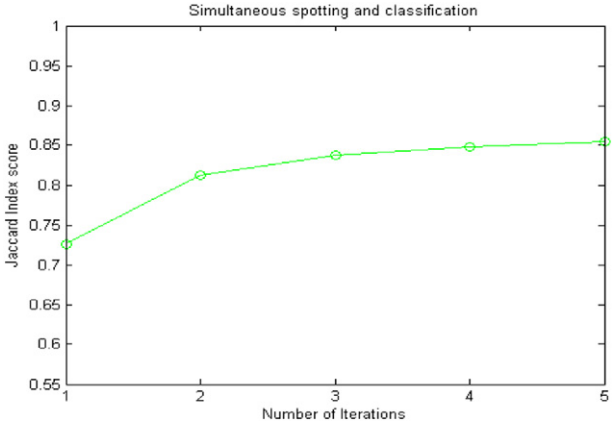


**Fig. 12.** Plot of number of misclassifications and Jaccard Index score with number of iterations of training the simultaneous classifier.

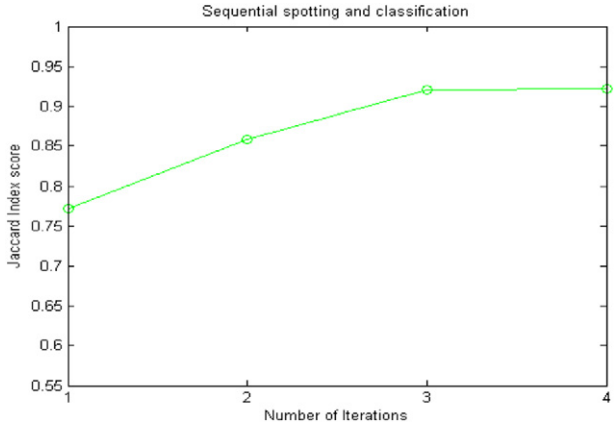


**Fig. 13.** Plot of number of misclassifications and Jaccard Index score with number of iterations of training the sequential classifier.

We have presented a comparison of random forest frameworks for a multi-gesture classification problem on a continuous setting. On the ChaLearn dataset, our classifier yields an average accuracy of 88.91% when tested on a set of segmented gestures. However, the task of simultaneously detecting and classifying gestures is a more difficult challenge than classifying accurately segmented gestures. Doing gesture spotting and classification by employing a cascaded framework yields better results than doing simultaneous spotting and classification, suggesting that solving the two problems sequentially is advantageous, especially in datasets where gestures are separated by background.

The strengths of our two frameworks lie in their simplicity to train and their capacity to generalize well to variations in user size, distance to the sensor, and speeds at which the gestures are performed, and their robustness to the effects of sensor noise. One area of the framework that can be improved is the process of selecting and creating better feature sets. Many additional features, such as joint-pair distances used by Yao et al. [35], can be experimented with in order to improve the accuracy of our framework. Additionally, selecting a small group of features over an interval of frames to split a node in a decision tree, instead of selecting a single feature at a single frame, might be better suited to the purpose of learning complex spatio-temporal objects such as gestures. However, computing more features may hamper the random forest framework's speed during test time.

As stated above, there are ambiguities between similar gesture pairs in both datasets, which the random forest classifier cannot differentiate well. A potential idea for further exploration is to use another layer of tree-forest classifiers to identify the features that can differentiate the ambiguities in order to further refine classification results. In general, gesture classification can be performed in a hierarchical framework, where random forests at the top-most level will accurately separate a dynamically-defined set of super-classes, each of which will be subject to further classification by classifiers at subsequent layers, until all classes are well-separated.

**Table 4**
Jaccard Index scores on ChaLearn gesture recognition challenge 2014 [29].

| Method | Jaccard Index score |
|---|---|
| Deep neural network [22] | 0.87 |
| Simultaneous spotting and classification | 0.68 |
| Sequential spotting and classification | 0.72 |

Please cite this article as: A. Joshi, et al., Comparing random forest approaches to segmenting and classifying gestures, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.06.001

## Acknowledgments

This work was supported in part by National Science Foundation grants IIS-0910908 and MRI-1337866 and US Navy contract N00014-13-P-1152.

## References

[1] S. Mitra, T. Acharya, Gesture recognition: a survey, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 37 (3) (2007) 311–324.
[2] S.S. Rautaray, A. Agrawal, Vision based hand gesture recognition for human computer interaction: a survey, Artif. Intell. Rev. 43 (1) (2015) 1–54.
[3] S. Malassiotis, N. Aifanti, M.G. Strintzis, A gesture recognition system using 3D data, Proceedings First International Symposium on 3D Data Processing Visualization and Transmission, 2002, IEEE. 2002, pp. 190–193.
[4] J. Alon, V. Athitsos, Q. Yuan, S. Sclaroff, Simultaneous localization and recognition of dynamic hand gestures, Seventh IEEE Workshops on Application of Computer Vision, 2005. WACV/MOTIONS'05, 2, IEEE. 2005, pp. 254–260.
[5] T. Starner, A. Pentland, Real-time American sign language recognition from video using hidden Markov models, Motion-Based Recognition Springer. 1997, pp. 227–243.
[6] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, Proceedings of the Eighteenth International Conference on Machine Learning, ACM. 2001, pp. 282–289.
[7] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, T. Darrell, Hidden conditional random fields, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1848–1852.
[8] Y. Song, D. Demirdjian, R. Davis, Multi-signal gesture recognition using temporal smoothing hidden conditional random fields, 2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011), IEEE. 2011, pp. 388–393.
[9] Y. Song, D. Demirdjian, R. Davis, Continuous body and hand gesture recognition for natural human–computer interaction, ACM Trans. Interact. Intell. Syst. (TiiS) 2 (1) (2012) 5.
[10] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, R. Moore, Real-time human pose recognition in parts from single depth images, Commun. ACM 56 (1) (2013) 116–124.
[11] F. Schroff, A. Criminisi, A. Zisserman, Object class segmentation using random forests, Proceedings of the British Machine Vision Conference, 2008.
[12] A. Bosch, A. Zisserman, X. Muoz, Image classification using random forests and ferns, IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007, IEEE. 2007, pp. 1–8.
[13] A. Kuznetsova, L. Leal-Taixé, B. Rosenhahn, Real-time sign language recognition using a consumer depth camera, Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, IEEE. 2013, pp. 83–90.
[14] J. Gall, A. Yao, N. Razavi, L. Van Gool, V. Lempitsky, Hough forests for object detection, tracking, and action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2188–2202.
[15] G. Yu, N.A. Goussies, J. Yuan, Z. Liu, Fast action detection via discriminative random forest voting and top-k subvolume search, IEEE Trans. Multimedia 13 (3) (2011) 507–517.
[16] T.-H. Yu, T.-K. Kim, R. Cipolla, Real-time action recognition by spatiotemporal semantic and structural forests, Proceedings of the British Machine Vision Conference, 2, 2010. pp. 6.
[17] L. Xu, K. Fujimura, Real-time driver activity recognition with random forests, Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, ACM. 2014, pp. 1–8.
[18] N.C. Camgöz, A.A. Kindiroglu, L. Akarun, Gesture recognition using template based random forest classifiers, Computer Vision-ECCV 2014 Workshops, Springer. 2014, pp. 579–594.
[19] L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A.W. Vieira, M.F. Campos, Real–time gesture recognition from depth data through key poses learning and decision forests, 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2012, IEEE. 2012, pp. 268–275.
[20] D. Demirdjian, C. Varri, Recognizing events with temporal random forests, Proceedings of the 2009 International Conference on Multimodal Interfaces, ACM. 2009, pp. 293–296.
[21] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[22] N. Neverova, C. Wolf, G. Taylor, F. Nebout, ModDrop: adaptive multi-modal gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2015)
[23] L. Pigou, A. v. d. Oord, S. Dieleman, M. Van Herreweghe, J. Dambre, Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video, 2015. arXiv preprint arXiv:1506.01911
[24] A. Yao, L. Van Gool, P. Kohli, Gesture recognition portfolios for personalization, Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014. pp. 1923–1930.
[25] L. Xia, C.-C. Chen, J. Aggarwal, View invariant human action recognition using histograms of 3d joints, Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, IEEE. 2012, pp. 20–27.
[26] M. Raptis, D. Kirovski, H. Hoppe, Real-time classification of dance gestures from skeleton animation, Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM. 2011, pp. 147–156.
[27] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, 1, IEEE. 2005, pp. 886–893.
[28] J. Marin, D. Vázquez, A.M. López, J. Amores, B. Leibe, Random forests of local experts for pedestrian detection, 2013 IEEE International Conference on Computer Vision (ICCV), IEEE. 2013, pp. 2592–2599.
[29] S. Escalera, X. Baró, J. Gonzalez, M.A. Bautista, M. Madadi, M. Reyes, V. Ponce, H.J. Escalante, J. Shotton, I. Guyon, ChaLearn looking at people challenge 2014: dataset and results, Proceedings of the 2014 IEEE European Conference on Computer Vision (ECCV 2014) ChaLearn Workshop on Looking at People, IEEE. 2014.
[30] Y. Song, D. Demirdjian, R. Davis, Tracking body and hands for gesture recognition: NATOPS aircraft handling signals database, 2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011), IEEE. 2011, pp. 500–506.
[31] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, H. Escalante, Multi-modal gesture recognition challenge 2013: dataset and results, Proceedings of the 15th ACM on International conference on multimodal interaction, ACM. 2013, pp. 445–452.
[32] Y. Song, L. Morency, R. Davis, Multi-view latent variable discriminative models for action recognition, 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. 2012, pp. 2120–2127.
[33] N. Neverova, C. Wolf, G.W. Taylor, F. Nebout, Multi-scale deep learning for gesture detection and localization, Proceedings of the 2014 IEEE European Conference on Computer Vision (ECCV 2014) ChaLearn Workshop on Looking at People, 2014.
[34] H. Cooper, R. Bowden, Large lexicon detection of sign language, Human–computer Interaction Springer. 2007, pp. 88–97.
[35] A. Yao, J. Gall, G. Fanelli, L.J. Van Gool, Does human action recognition benefit from pose estimation? Proceedings of the British Machine Vision Conference, 3, 2011. pp. 6.

Please cite this article as: A. Joshi, et al., Comparing random forest approaches to segmenting and classifying gestures, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.06.001