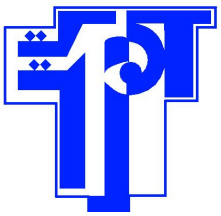
Tunisia Polytechnic School



Graduation Project Report

A FRAMEWORK FOR CARDIO-PULMONARY RESUSCITATION (CPR) SCENE RETRIEVAL FROM MEDICAL SIMULATION VIDEOS BASED ON ACTIVITY DETECTION

Elaborated by: **Mohamed CHAABANE**

3^{rd} year engineer student SISY (Signals and Systems)

Supervised by: **Pr. Hichem Frigui**

Professor, CECS at University of Louisville

Pr. Amel BEN AZZA

Professor, Telecommunication at SUP'COM

Academic Year

2015 - 2016

Acknowledgements

I would like to express my sincere thanks to Pr. Frigui, professor of computer engineering and computer science at the University of Louisville for giving me the opportunity to have this internship at the Multimedia Research Lab of Louisville, for being always available for discussing ideas and for his guidance, support and motivation throughout the duration of my internship.

I would like to thank my academic supervisor at Tunisia Polytechnic School, Pr. Benazza for her continuous assistance.

Finally, I am grateful to my parents for all the support they have given throughout these years.

Abstract

Medical simulations, where uncommon clinical situations can be replicated, have proved to provide a more comprehensive training. Simulations involve the use of patient simulators, which are lifelike mannequins. After each session, the physician must manually review and annotate the recordings and then debrief the trainees. This process can be tedious and retrieval of specific video segments should be automated.

In this report, we present activity based scene retrieval framework to detect and classify scenes that involve Cardio-Pulmonary Resuscitation (CPR) activity from training video sessions simulating medical crises.

The first step of our approach consists of segmenting the video into overlapping volumes. Then from each volume, we extract the region of interest which is the chest of the lifelike mannequin since if a CPR activity is taking place, the hands of the trainee will be placed on the center of the chest of lifelike mannequin. To reduce the computational complexity of our approach we discard from processing volumes which don't have significant motion in the region of interest and to improve the efficiency of our framework, we detect the nearest head to the region of interest in order to analyze its motion in addition to the motion of the hands.

After extracting two sub-volumes from each volume, we describe them using spatio-temporal features: histogram of oriented gradients 3D, local binary patterns-three orthogonal planes and scale-invariant feature transform 3D. Then, we use Support Vector Machine (SVM) and K Nearest Neighbors (KNN) classifiers for classification and finally we fuse all the information obtained from different classifiers to

decide finally for each video volume if it belongs to CPR scene or not.

The proposed framework is tested and validated using two simulation videos and the experimental results are presented.

Contents

List of Figures

List of Tables

Introduction

The Simulation for Pediatric Assessment, Resuscitation, and Communication (SPARC) group, within the Department of Pediatrics at the University of Louisville, was developed to teach pediatrics faculty, fellows, and residents how to respond to medical crises. The objective of SPARC is to enhance the care of infants and children by using simulation-based educational methodologies to improve patient safety, strengthen interdisciplinary and clinician-patient interactions.

Simulation sessions involve 4 to 9 people and last approximately 15 minutes to one hour. Human-like mannequins that have respiration and heartbeat and respond to treatment with virtual drugs, are used for these simulations. After each such session, the physician would manually review and annotate the recording, and then debrief the trainees on the session. Video assisted debriefing allows participants to reflect on their experience, teaching them to be more efficient and productive during such real-life scenarios. With the increasing number of simulation sessions, the physicians realized that (1) the manual process of review and annotation is labor intensive; (2) retrieval of specific video segments is not trivial; and (3) there is wealth of information waiting to be mined form these recordings. Providing the physician with automated tools to segment, semantically index and retrieve specific scenes (Figure(1)) from a large database of training sessions will enable him/her to (1) immediately review important sections of the training with the team, (2) allow more efficient debriefing session with the team of trainees, and (3) identify similar circumstances in previously recorded sessions. The longer-term payback is the potential discovery of similar critical elements in a training session that results

in either positive or negative outcomes and thus enhancing the effectiveness of the training.

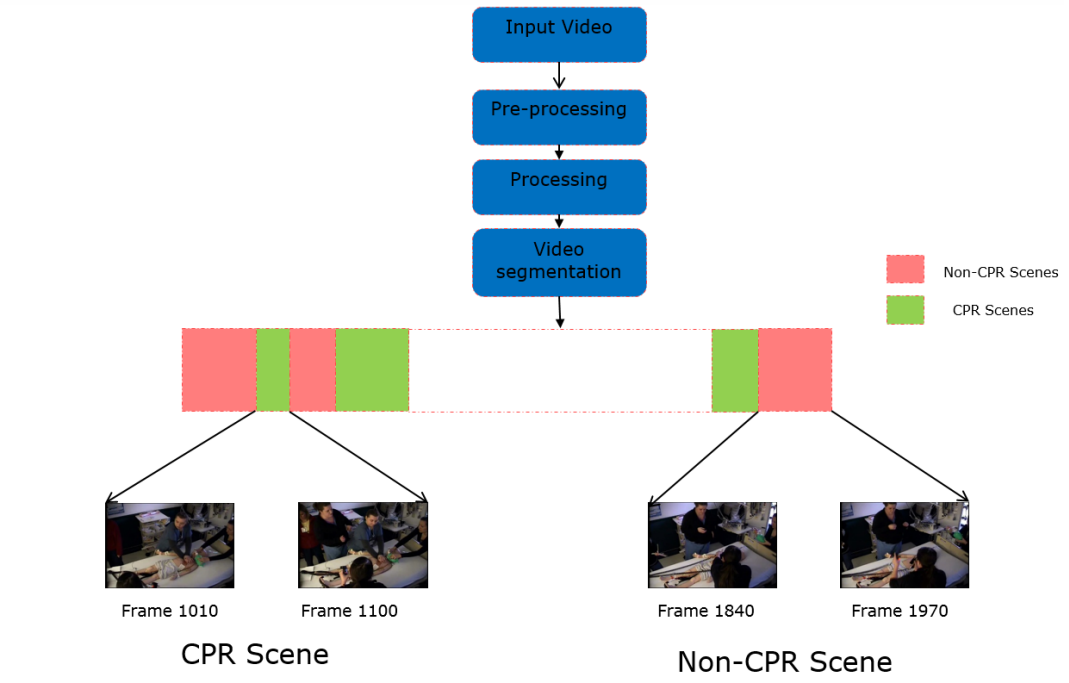


Figure 1: A typical CPR scene retrieval system. Scenes containing CPR actions can be differentiated from non-CPR scenes, and retrieved.

In this project, we focus on detecting and classifying scenes that involve rhythmic activities such as Cardio-Pulmonary Resuscitation (CPR) from training video sessions simulating medical crises. In particular, as illustrated in figure(1), our system takes a video simulation session as input and output a timeline that indicates the occurrences of CPR scenes.

The remainder of this report is organized as follows. In the first chapter, we provide an overview of related work on activity recognition and detection techniques. In the second chapter, we propose the CPR scene identification system using spatio-

temporal features. In the third chapter, we present the experimental results and analysis, and describe our graphical user interface (GUI). Finally, we conclude and discuss future work.

1 Related work

Cardiopulmonary resuscitation (CPR) consists of the use of chest compressions. In an emergency room simulation, a typical CPR procedure for an adult consists of the use of chest compressions: Overlapping hands placed on the center of the sternum of Human-like mannequins (figure(1.1)). Chest compressions are to be delivered at a rate of at least 100 compressions per minute. The main goal of this project is to identify and retrieve CPR scenes. We treat the problem as a general activity recognition, a research topic that has been studied extensively in the past few years.

Figure 1.1: A sample frame taken from a CPR activity scene.

A typical Human Activity Recognition (HAR) system consists of the following three main steps:

1. Pre-processing - Videos may require different preprocessing techniques depending on the quality of video recording and the application at hand. Background subtraction, contrast normalization, filtering, sampling etc. are different types of preprocessing methods generally used for video analysis [1, 2].

2. Feature Extraction - It is important to extract the appropriate features to represent the content of a video. Video features are typically extracted from textual [18, 19], audio or visual modalities [3, 4]. In general, a combination of multiple features from different modalities will be needed to represent the complex content of a video [5]. In such cases, fusion of different features from different modalities are used.

3. Feature Learning - Once the features are extracted, the desired system (classifier, search engine, indexing application etc.) is built by analyzing and learning patterns within the features. Existing learning algorithms use supervised [6, 7] , semi-supervised [8, 9] , or unsupervised [10, 11] techniques for building the system.

HAR is difficult, owing to several reasons such as the high dimensionality of the features representing the video data, large intra-class variability due to difference in scale, illumination changes, camera movements and also the resolution and quality of the video recording.

HAR in medical videos has been gaining attention in the past few years. It finds tremendous use in patient monitoring videos [12, 14, 15], fall detection in elderly

monitoring videos [13], early detection of many diseases like hand tremor or neonatal epilepsy [14, 16], and also in medical simulation videos which are recorded for educational purposes.

In the remaining of this chapter, we give an overview of different types of preprocessing methods, features and classifiers that are related to our project and finally some decision level fusion methods that we have used.

1.1 Preprocessing

In most of the activity datasets, the activity frames have various resolutions and backgrounds, and are typically taken under varying light conditions. Therefore, the preprocessing module is necessary to improve the quality of the frames. At this stage, background information, illumination noise, and unnecessary details are discorded for fast and easy processing.

For preprocessing module, some well-known methods such as histogram equalization (HE), median filter, and homomorphic filter have been employed in order to enhance the quality of the video frames.

For a typical Human Activity Recognition system, human detection is first used to focus interest of future tasks. Background subtraction is simple yet effective method for human detection and silhouette extraction. For example, In [17], background subtraction and application of several noise reduction and smoothing filters used in order to obtain an accurate silhouette representation.

Temporal video segmentation is another common method of preprocessing aiming to divide the video stream into a set of meaningful and manageable segments (shots) that can be used as basic elements for indexing. Each shot is then rep-

resented by selecting key frames and indexed by extracting spatial and temporal features. Video segmentation has been used extensively for Human Activity Recognition [18, 19] and in particular for CPR activity detection [21].

1.2 Feature extraction

Most of the recent action recognition approaches can be divided into the following categories: 2D based methods and 3D spatio-temporal methods.

2D based methods describe an action as a sequence of 2D images or as a 2D template image. Histogram of Oriented Gradients (HOG) [23] is a 2D descriptor that has been used extensively for people activity recognition. Dalal and Triggs [23] explain how a dense histogram of gradient orientations can capture edge information which aids in efficient detection of pedestrians. HOG can also be used in combination with other temporal features for activity recognition. For instance, Laptev et al. in [22] introduced HOG/HOF descriptors which is a fusion of the 2D histogram of oriented spatial gradients (HOG) and histograms of optical flow (HOF). The two histograms are concatenated to form one descriptor. Both descriptors are computed in the space-time neighborhood of the detected interest points.

Human action can also be visualized as a 3D volume created by the trail of the performer. Different 3D features have been used for activities recognition. In [25, 26] volume local binary patterns (VLBP) which is an extension of the local binary patterns (LBP [24]) was experimentally proven to be successful for the task of action recognition. In [27], the VLBP and optical flow descriptors were combined to produce an efficient descriptor for activity recognition, called motion binary pat-

terns (MBP). The VLBP descriptor can be simplified by concatenating the local binary patterns on three orthogonal 2-D planes: XY, XT and YT planes [29] and it is called Local Binary Patterns - Three Orthogonal Planes(LBP-TOP). In [20], Klaser et al. proposed the 3D HOG features (referred to as HOG3D) which are 3D extension of the HOG features into the temporal dimension. HOG3D features are computationally efficient and are able to combine motion and appearance into one representation. In [30], Paul et al. proposed a 3D extension of 2D SIFT descriptor [31] in order to represent the 3D nature of video data in the application of action recognition. In the rest of this chapter, we focus more on the details of spatio-temporal features and in particular on features that are highly relevant to our application.

1.2.1 Spatio-temporal histogram of oriented gradients - HOG3D

The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions.

HOG approximates the two components I_x and I_y of the gradient of the image I by central differences:

$$\begin{cases} I_x(r;c) = I(r;c+1) - I(r;c-1) \\ I_y(r;c) = I(r-1;c) - I(r+1;c) \end{cases} \tag{1}$$

Figure (1.2) illustrate that gradient image in the X-direction measures the horizontal change in intensity and gradient image in the Y-direction measures the vertical change in intensity.

(a) Initial Image I

(b) I_x

(c) I_y

Figure 1.2: A sample image and its gradient in the X and Y directions.

The gradient is then transformed to polar coordinates, with the angle constrained to be between 0 and 180 degrees, so that gradients that point in opposite directions are identified:

$$\begin{cases} \mu = \sqrt{I_x^2 + I_y^2} \\ \theta = \frac{180}{\pi}(tan^{-1}(\frac{I_y}{I_x}) mod \pi) \end{cases} \tag{2}$$

Once the gradient is computed at every pixel of the window, HOG divides the window into adjacent, non-overlapping cells of size $C \times C$ pixels (figure(1.3)). Then, a histogram of the gradient orientations, binned into B bins, is computed for each cell.

After computing a histogram for each cell, HOG groups the cells into overlapping blocks of 2 $\times$2 cells each, so that each block has size $2C \times 2C$ pixels. Then the four cell histograms in each block are concatenated into a single block feature b, and normalized using:

$$b \leftarrow \frac{b}{\sqrt{\|b\|^2 + \epsilon}} \tag{3}$$

In (3), ϵ is a small positive constant that avoids division by zero.

Cell histograms need to be normalized to reduce the effect of changes in contrast between images of the same object (figure(1.4)).

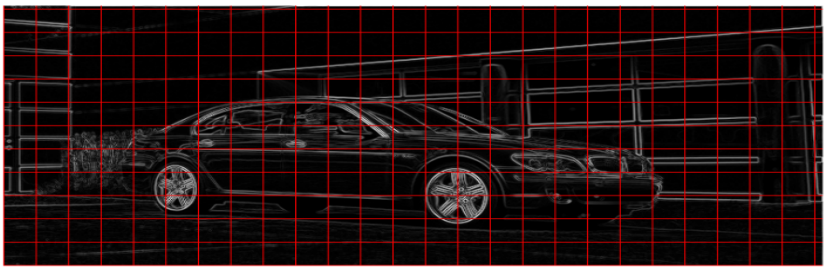


Figure 1.3: Cell division example.

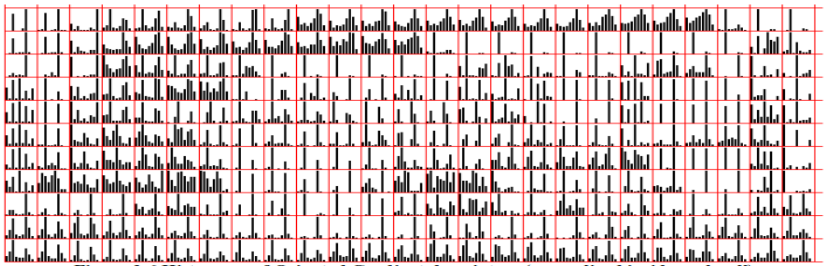


Figure 1.4: Histogram of Oriented Gradient descriptors (normalized inside each cell).

An overview of the HOG computation descriptor is given in figure (1.5). In [20], Klaser et al. proposed the 3D HOG features (referred to as HOG3D), which are fast and simple to compute and are able to combine motion and appearance into one representation. This is a 3D extension of the HOG features into the temporal dimension and is robust to changes in illumination.

Figure (1.6) gives an overview of the HOG3D descriptor computation used in [20]. For each video volume, a sparse set of spatio-temporal interest points are obtained using Harris3D ([28]) operator. Interest points are a subset of all points, that exhibit certain properties which distinguish them from the remaining points. The support region around the interest points are determined and these regions are divided into grids. Figure (1.6(a)), shows the spatio-temporal region of size

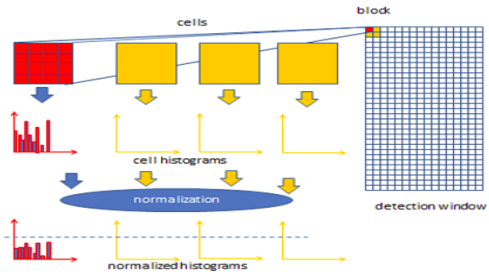


Figure 1.5: overview of the HOG descriptor computation.

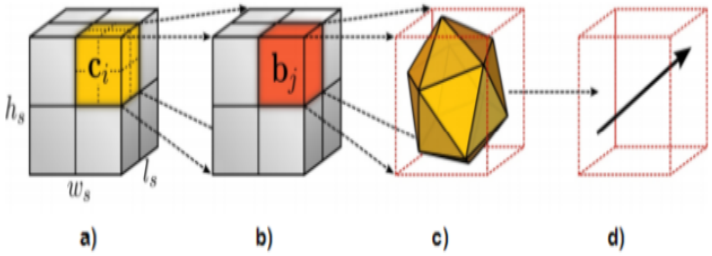


Figure 1.6: Overview of the HOG3D descriptor computation: (a) Support region around a point of interest divided into grid, gradient orientation histogram computed at each sub volume in the grid is concatenated to form final histogram; (b) each histogram is computed over a grid of mean gradients; (c) each gradient orientation is quantized using regular polyhedrons; (d) each mean gradient (courtesy [20]).

(h_s, w_s, l_s), around an interest point. This region is divided into a $M \times M \times N$ grid, (here $M = N = 2$). Each sub-volume C_i is divided into $S \times S \times S$ sub-blocks b_j . The mean gradient is computed at each of the sub-blocks b_j in the grid (figure (1.6(d))). This gradient is projected through a 20- D polyhedron to form 20-D histogram (figure (1.6c)). The histograms from every sub-block are concatenated to form the final feature vector. For activity detection using HOG3D, features are sampled at multiple spatial and temporal scales. The video sequence is then mapped to a feature with fixed dimension using a using a bag-of-words [32] rep-

resentation of the sparse space-time features. A non-linear SVM classifier with χ^2 kernel is then used for classification.

1.2.2 Local binary patterns - three orthogonal planes - LBP-TOP

The local binary pattern (LBP) offers a powerful and attractive texture descriptor that proved to yield excellent results in terms of accuracy and computation complexity in many empirical studies. The original LBP operator represents the pixels of an image with decimal numbers, which are called LBPs or LBP codes that encode the local structure around each pixel. It proceeds as illustrated in Figure(1.7): Each pixel is compared with its eight neighbors in a 3×3 neighborhood by subtracting the center pixel value. In the result, negative values are encoded with 0, and the others with 1. For each given pixel, a binary number is obtained by merging all these binary values in a clockwise direction, which starts from the one of its top-left neighbor. The corresponding decimal value of the generated binary number is then used for labeling the given pixel. The derived binary numbers are referred to be the LBPs or LBP codes.

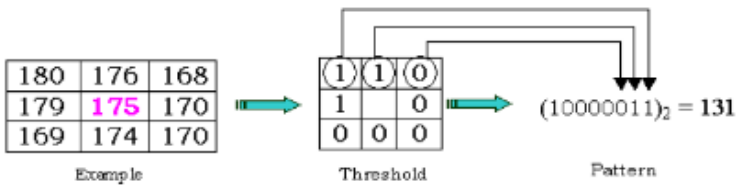


Figure 1.7: Example of LBP calculation.

The most prominent limitation of the basic LBP operator is its small spatial

support area. A feature computed using a 3×3 operator, only relating to a small image structure, that may not necessarily be to capturing the key texture characteristic. Consequently, the LBP operator has been extended to consider different neighbor sizes. For example, the operator $LBP_{4,1}$ uses 4 neighbors while $LBP_{16,2}$ considers the 16 neighbors on a circle of radius 2. In general, the operator $LBP_{P,R}$ refers to a neighborhood size of P equally spaced pixels on a circle of radius R that form a circularly symmetric neighbor set. $LBP_{P,R}$ produces 2^P different output values, corresponding to the 2^P different binary patterns that can be formed by the P pixels in the neighbor set and the corresponding equation is shown below.

$$LBP_{P,R}(x,y) = \sum_{p=0}^{P-1} s(g_p - g_c)2^P \tag{4}$$

where g_c is the intensity in the central pixel, g_p is the intensity in the neighbor pixel, and s is the step function defined as:

$$s(x) = \begin{cases} 1, if x >= 0 \\ 0, \text{otherwise} \end{cases} \tag{5}$$

A histogram $h(i)$ of the local binary patterns is then used to represent the texture of an image i:

$$h(i) = \sum_{x,y} L(LBP_{P,R}(x,y) = i) \mid i \in [0, 2^{P-1}] \tag{6}$$

In Eq.(6), $L\{A\}$ is 1 if A is true and 0 if A is false.

LBP operator has two important properties: it is invariant to monotonic gray scale changes, and its complexity is very low. As a consequence, LBP-based approaches are suitable for many applications, aside from texture recognition [33].

Another feature representation that is based on LBP is the local binary texture descriptor is computed to segment dynamic texture from an input video. This descriptor is used as spatial–texture descriptor when utilized in XY plane of a video. Also, when this descriptor is used in the XT and YT planes, it is called temporal-texture descriptor. Hence this descriptor, called Local Binary Pattern-Three Orthogonal Planes, is spatio-temporal descriptor as it is used in both spatial and temporal domain. XT plane indicates the change/deviation in pixels' row-wise over temporal domain. YT plane indicates the change/deviation in pixels' column-wise over temporal domain. The LBP-TOP descriptor can be obtained simply by concatenating the local binary patterns of three orthogonal 2-D planes: XY, XT and YT planes as shown in figure(1.8)

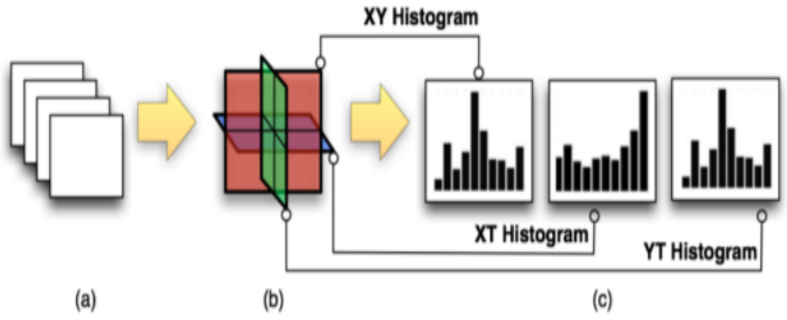


Figure 1.8: Computation of LBP-TOP on a 3D volume as a concatenation of three LBP's extracted from Three Orthogonal Planes.

1.2.3 3D Scale-invariant feature transform (SIFT-3D)

Extensions of the SIFT descriptor to 3 dimensional spatio-temporal data in context of human action recognition in video sequences have been proposed [30, 34]. The computation of local position-dependent histograms in the 2D SIFT algo-

rithm are extended from two to three dimensions to describe SIFT features in a spatio-temporal domain. For application to human action recognition in a video sequence, sampling of the training videos is carried out either at spatio-temporal interest points or at randomly determined locations, times and scales. The spatio-temporal regions around these interest points are then described using the 3D SIFT descriptor.

In the following, we will outline the differences between the 2D SIFT descriptor and the 3D SIFT descriptor. The first step is to compute the overall orientation of the neighborhood. Once this is computed we can create the sub-histograms which will encode our 3D SIFT descriptor. The 2D gradient magnitude and orientation for each pixel in an input image $L(x, y)$ is defined as follows:

$$\begin{cases} m_{2D}(x,y) = \sqrt{L_x^2 + L_y^2} \\ \theta(x,y) = tan^{-1}(L_y/L_x) \end{cases} \tag{7}$$

where

$$\begin{cases} L_x = L(x+1,y,t) - L(x-1,y,t) \\ L_y = L(x,y+1,t) - L(x,y-1,t) \end{cases} \tag{8}$$

For the 3D case, the gradient magnitude and orientations can be computed using [30]:

$$\begin{cases} m_{3D}(x,y,t) = \sqrt{L_x^2 + L_y^2 + L_t^2} \\ \theta(x,y,t) = tan^{-1}(\frac{L_y}{L_x}) \\ \phi(x,y,t) = tan^{-1}(\frac{L_t}{\sqrt{L_x^2+L_y^2}}) \end{cases} \tag{9}$$

In (9), ϕ encodes the angle away from the 2D gradient direction. Each pixel has two values which represent the direction of the gradient in three dimensions. Next, we construct a weighted histogram by dividing θ and ϕ into equally sized bins and creating a 2D histogram.

Bins will need to be normalized by their solid angle (w). The solid angle can be calculated in the following manner:

$$w = \int_{\phi}^{\phi+\Delta\phi} \int_{\theta}^{\theta+\Delta\theta} sin\theta d\theta d\phi \tag{10}$$

$$w = \Delta\phi(cos\theta - cos(\theta + \Delta\theta)) \tag{11}$$

The next step is to compute the SIFT descriptor for which we start by calculating the orientation sub-histograms. The first step in this process will be to rotate the 3D neighborhood surrounding the key point so that the dominant orientation points in the direction of $\theta = \phi = 0$. This is done by taking each (x, y, t) position in the neighborhood and multiplying it by the following matrix:

$$\begin{pmatrix} cos\theta cos\phi & -sin\theta & -cos\theta sin\phi \\ sin\theta cos\phi & cos\theta & -sin\theta sin\phi \\ sin\phi & 0 & cos\phi \end{pmatrix}$$

To create our sub-histograms, we sample the sub-regions surrounding the interest point where each pixel contains orientation values θ and ϕ. For each 3D sub-region, we accumulate the orientations into a histogram. The final descriptor is a vectorization of the sub-histograms.

1.3 Activity classification

After feature extraction, the features are passed to a classifier to learn model and discriminate between different actions.

Some well-known classifiers such as artificial neural networks (ANNs) [35, 36], support vector machines (SVMs) [37, 38], Gaussian mixture models (GMMs) [39], and hidden Markov models (HMMs) [40] have been used extensively in human action recognition.

In the remaining of this chapter, we outline the classifiers that are relevant to our project.

1.3.1 Support vector machine

Support Vector Machines (SVM) are supervised learning models based on training algorithms that maximize the margin between the training patterns and the decision boundary. The classification function essentially depends only on the supporting patterns which are those training examples that are closest to the decision boundary. They are usually a small subset of the training data. Given a set of labeled training data, the SVM algorithm builds a model, that assigns unseen test data into one category or the other, making it a non-probabilistic binary linear classifier. The effective number of parameters is automatically adjusted depending on the complexity of the problem. The solution or model is expressed as a linear combination of the supporting patterns. Thus, an SVM classifier constructs a hyperplane or set of hyperplanes in a high-dimensional space which can be used for classification, regression and other tasks. SVM have proved to perform efficiently in many classification tasks [20] [41] [42], especially when the features have very

high dimension.

1.3.2 K-nearest neighbors

K Nearest Neighbor (KNN) is a are very simple algorithm, yet works very well in practice. KNN is non-parametric lazy learning algorithm [43] that does not rely on the training data samples to do any generalization. In other words, there is no (or very limited) explicit training phase or it is very minimal. Lack of generalization means that KNN keeps all the training data in memory as all of the training data is needed during the testing phase. This is in contrast to other techniques like SVM where you can discard all non-support vectors without any problem. Some KNN versions summarize the training data by few prototypes and maintain only these prototypes in memory for testing new samples.

In KNN, the training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors (or the representative prototypes) and class labels of the training samples. In the classification phase, K is a user-defined constant which decides how many neighbors influence the classification. An unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the K training samples nearest to that query point. Any distance measure, such as: Euclidean distance, Hamming distance, Manhattan distance, Minkowski distance can be used to identify the K nearest neighbors.

Choosing the optimal value for K can be critical in the KNN. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee.

1.4 Fusion methods

Traditional machine learning and pattern recognition systems use features to describe sensor data and a classifier (also called "expert" or "learner") to determine the true class. However, for complex detection and classification problems involving data with large intra-class variations and noisy inputs, perfect solutions are difficult to achieve, and no single source of information can provide a satisfactory solution. As a result, combination of multiple classifiers (or multiple experts) is playing an increasing role in solving these complex problems, and has proven to be a viable alternative to using a single classifier.

Classifier combination have been applied to various fields including character recognition [46], speech recognition [45] and text categorization [44], and have been proved to be superior to single classifier systems both theoretically and experimentally.

Fusion of data/information can be carried out on three levels of abstraction closely connected with the flow of the classification process: data level fusion, feature level fusion, and decision level fusion. Data level fusion, also called low level fusion, combines several sources of raw data to produce new raw data that is expected to be more informative and synthetic than any of the single sources. Feature level fusion, also called intermediate level fusion, combines various features. These features may come from different raw data sources (e.g. sensors) or from the same raw data. In the latter case, the objective is to find relevant features among available features that might come from several feature extraction methods. Decision level fusion, also called high level fusion, combines decisions coming from several experts.

Figure (1.9) shows a generic architecture for the different levels of information

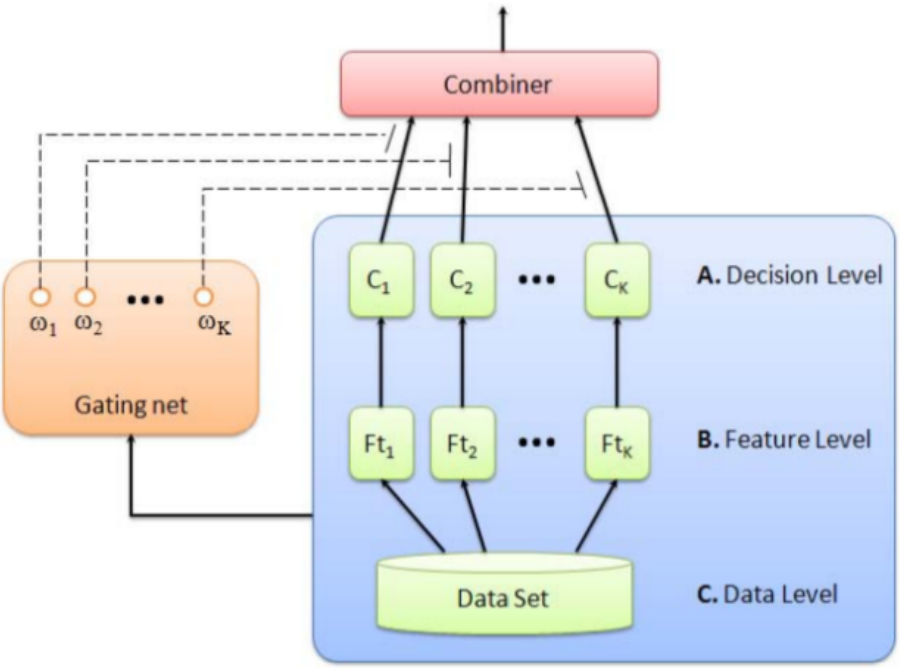


Figure 1.9: A general architecture for information fusion.

fusion. It illustrates three basic ingredients of fusion (fusion level, gating net, and combiner). Different combinations of these different ingredients lead to different specific models for expert combination [47].

Decision level fusion, also called high level fusion, combines decisions coming from several experts. By extension, one speaks of decision fusion even if the experts return a confidence (score) and not a decision. In our project, decision level fusion can be used to improve the accuracy of our approach.

Let $\mathcal{X} = \{x_j | j = 1, ..., N\}$ be a set of N training observations to be classified into one of the M classes: $c_1, ..., c_M$, and Let $e_1, ..., e_K$ denote K classifiers. Each classifier k generates confidence values, $\mathcal{Y}^k = \{{y_j}^k | j = 1, ..., N\}$.

In the following, we outline several classifier fusion methods that are related to our

work. These methods include Artificial Neural Networks, Bayesian Fusion, Fuzzy Integral and supervised Borda-Count.

1.4.1 Artificial neural networks fusion

Artificial Neural Networks (ANN) have been applied successfully to many pattern classification problems. They have also shown promise to the classifier fusion problem. A neural network designed for the purpose of classifier fusion should have one crisp output or alternatively a number of soft outputs equal to the number of classes if there is a need to produce qualitative assignment values to each class. The input of such a network should be associated with the individual classifier outputs [48] [49].

Given a neural network that performs a mapping of K individual classifiers outputs (taken as input) into M outputs corresponding to the level of assignment to each of the M classes. If a crisp decision is required, the output with the highest value is chosen. The input-output mapping in ANN is determined via an iterative learning process. During the learning stage, weights between each pair of connected nodes of the network are adapted in such a way as to minimize the difference between the actual network output and the desired output.

It is quite common for the output of a set of ANNs to be combined using another ANN. Following this approach neural networks working as a mixture can be expanded to a higher dimension by fusing several neural networks [50] or arranging them in an efficient ANN-like structure [51].

1.4.2 Bayesian fusion

The Bayesian methods can be applied to the classifier fusion under the condition that the classifiers' outputs are expressed by posterior probabilities. Effective combination of given likelihoods is also a probability of the same type, which is expected to be higher than the probability of the best individual classifier for the correct class.

Let v represent the output of all L algorithms to be fused, i.e., $v = [y_1, y_2, ..., y_L]$. Within the Bayesian framework, v is considered a random variable with a distribution that depends on the state of nature. Using Bayes formula, we first compute the posterior probability using

$$p(\Omega_i|v) = \frac{p(v|\Omega_i)p(\Omega_i)}{p(v)} \tag{12}$$

Then, v is assigned to the class with maximum posterior probability, i.e.,

$$v \in \Omega_j \quad if \quad p(\Omega_j|v) = \max_{i=1..k} p(\Omega_i|v) \tag{13}$$

In (12), $p(\Omega_i)$ is the prior probability of class i and $p(v|\Omega_i)$ is the class conditional density. The prior $p(\Omega_i)$ is usually provided by an expert, or estimated using the relative proportions of training data from each class. Similarly, $p(v|\Omega_i)$can be estimated from the training data. The Gaussian distribution is usually used as the density function. This is because the Gaussian distribution is easy to handle, and in many cases, the distribution of the sample vectors can be regarded as normal if there are enough samples. The mean vector and covariance matrix are calculated from the vectors.

Let d be the dimension of feature vector. The probability density function of a d-dimensional normal distribution is given by:

$$p(v|\Omega_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\sum_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(v-\mu_i)^T \sum_i{}^{-1}(v-\mu_i)} \tag{14}$$

where v is a d-component vector, μ_i is the mean vector for class i, and $\sum_i$ is the d x d covariance matrix for class i. Then, the posterior probability $p(\Omega_i|v)$ can be computed by Bayes formula using Eq. (12) as:

$$\begin{aligned} p(\Omega_k|v) &= \frac{p(v|\Omega_k)p(\Omega_k)}{p(v)} \\ &= \frac{p(\Omega_k)\, e^{-\frac{1}{2}(v-\mu_k)^T \sum_k{}^{-1}(v-\mu_k)}}{(2\pi)^{\frac{d}{2}} |\sum_k|^{\frac{1}{2}} p(v)} \end{aligned} \tag{15}$$

If the training data can be modeled by a mixture of Gaussian distributions, the Expectation Maximization (EM) algorithm [52] can be used first to build the multiple Gaussian model. Then, this modeling can be used to make the final decision according to the above Bayes rule. The EM algorithm is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. Each iteration of the EM algorithm consists of two processes: The E-step, and the M-step. In the expectation, or E-step, the missing data are estimated given the observed data and current estimate of the model parameters. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimate of the missing data from the E-step are used in lieu of the actual missing data. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration. For the fusion problem, we can first cluster the training data using the EM into M

components. Then, the posterior probability can be computed by generalizing Bayes rule in Eq. (16) to assign a test point into different component or class.

$$
\begin{aligned}
p(\Omega_k|v) &= \sum_{i=1}^{M} \frac{p(v|\Omega_{ki})p(\Omega_{ki})}{p(v)} \\
&= \sum_{i=1}^{M} \frac{p(\Omega_{ki})\, e^{-\frac{1}{2}(v-\mu_{ki})^T {\sum_{ki}}^{-1}(v-\mu_{ki})}}{(2\pi)^{\frac{d}{2}} |\sum_k|^{\frac{1}{2}} p(v)}
\end{aligned}
\tag{16}
$$

1.4.3 Fuzzy integral

The fuzzy integral has been investigated extensively for information fusion [53] [54] [55] [56]. This integral defines a family of generally nonlinear aggregation operators on some function of the algorithm confidence values. The aggregation operator is defined by the fuzzy integral with respect to a non-additive fuzzy measure. As used here, fuzzy measures are real-valued functions defined on sets of algorithms. Thus, the fuzzy integral is a mathematical construct that can be used to optimize the aggregation operator for a specific fusion application.

Definition 1.1. (Fuzzy measure). Let $\mathcal{A} = \{a_1, ..., a_k\}$ be a finite set. A fuzzy measure, g, is a real valued function defined on the power set of $\mathcal{A}$, $\mathscr{P}(\mathcal{A})$, with range [0,1], satisfying the following properties:

1. g($\emptyset$) =0 and g($\mathcal{A}$)=1.

2. given A, B $\in \mathcal{A}$,if A $\subseteq$ B then g(A)$\leq$ g(B).

For the purpose of fusion, the set $\mathcal{A}$ is considered to contain the names of different information sources (algorithms), and for a subset A $\subseteq \mathcal{A}$, g(A) is considered to

be the degree of worthiness of this subset of information. Many fuzzy measures were introduced in the literature. In this work, we limit our study to the Sugeno measures which are a special class of fuzzy measures.

Definition 1.2. (Sugeno measure). A fuzzy measure g is called a Sugeno measure if it satisfies the following additional property: for all A, B $\subseteq \mathcal{A}$ with A $\bigcap$ B = $\emptyset$, there exists $\lambda > -1$ such that

$$g(A \cup B) = g(A) + g(B) + \lambda\, g(A)\, g(B) \tag{17}$$

It can be shown that a set function satisfying the conditions in Definition 1.2 is a fuzzy measure. In particular, equation (17) implicitly imposes the monotonicity constraints on the Sugeno measures. The value of λ can be determined for a finite set $\mathcal{A}$ using (17) and the facts that $\mathcal{A} = \bigcup_{k=1}^{K} \{a_k\}$ and g($\mathcal{A}$)=1, which leads to solving the following equation for λ :

$$1 + \lambda = \prod_{k=1}^{K} (1 + \lambda g(\{a_k\})),\ and\ \lambda > -1 \tag{18}$$

Equation (18) is a polynomial in λ of degree $K - 1$, and can be easily solved numerically [57] [58].

The discrete Choquet integral [59] has proved to be useful tool to fuse evidence supplied by different information sources.

Definition 1.3. (Choquet integral). Let e : $\mathcal{A} \rightarrow$ [0,1].Let $\{a_{\sigma(1)}, ..., a_{\sigma(K)}\}$ denote the reordering of the set $\mathcal{A}$ such that $e(a_{\sigma(1)}) \leq ... \leq e(a_{\sigma(K)})$, and let A_k be a collection of subsets defined by $A_k = \{a_{\sigma(k)}, ..., a_{\sigma(K)}\}$. The discrete Choquet

integral of e with respect to g on $\mathcal{A}$ is defined as:

$$C_g(e) = \sum_{k=1}^{K} [e(a_{\sigma(k)}) - e(a_{\sigma(k-1)})] \; . \; g(A_k) \tag{19}$$

or

$$C_g(e) = \sum_{k=1}^{K} [g(A_k) - g(A_{k+1})] \; . \; e(a_{\sigma(k)}) \tag{20}$$

where $e(a_{\sigma(0)}) = 0$ and $A_{k+1} \equiv \emptyset$.

The function e is a particular instance of the partial support (evidence) supplied by each information source in determining the confidence in an underlying hypothesis. The integral fuses this objective support with the degree of worthiness of the various subsets of the information sources. The analysis of the coefficients of the fuzzy measure can be performed by the calculation of the Shapley values[60].

Definition 1.4. (Shapley value). The Shapley value of g is a k dimensional vector $\Phi_g = [\Phi_g(a_1), ..., \Phi_g(a_k)]$, defined by

$$\Phi_g(a_k) = \sum_{A \subseteq \mathcal{A} \setminus \{a_k\}} \gamma_{\mathcal{A}}(A) \Big(g(A \cup \{a_k\}) - g(A) \Big) \tag{21}$$

with

$$\gamma_{\mathcal{A}}(A) = \frac{(|\mathcal{A}| - |A| - 1)! \times |A|!}{|\mathcal{A}|!} \tag{22}$$

where $|A|$ indicates the cardinality of A. The Shapley value, $\Phi_g(a_k)$ with respect to a fuzzy measure g, represents the global importance of each source a_k with respect to any subset A not containing a_k. It is confined to the interval [0,1]. A value close to zero indicates that the k^{th} algorithm is not relevant for the given data, while a value close to 1 indicates that the given K algorithm is highly relevant for

the given data. It can be proven that $\sum_{k=1}^{K} \Phi_g(a_k) = 1$.
Another way to analyze the coefficient of the fuzzy measure is to compute the interaction index $I_g(a_k, a_l)$ [61] [62] between pairs of information sources.

Definition 1.5. (Interaction index). The mean interaction index between 2 sources k and l with respect to g is defined by

$$I_g(a_k, a_l) = \sum_{A \subseteq \mathcal{A} \setminus \{a_k, a_l\}} \xi_{\mathcal{A}}(A) \Big(g(A \cup \{a_k, a_l\}) - g(A \cup \{a_k, \}) - g(A \cup \{a_l\}) + g(A) \Big) \tag{23}$$

with

$$\xi_{\mathcal{A}}(A) = \frac{(|\mathcal{A}| - |A| - 2)! \times |A|!}{(|\mathcal{A}| - 1)!} \tag{24}$$

A positive value of the interaction index ($I_g(a_k, a_l) > 0$) induces a conjunctive behavior in aggregation. That is, algorithms k and l have to be both satisfied in order to have a good global score. On the other hand, a negative value of the interaction index ($I_g(a_k, a_l) < 0$) induces a disjunctive behavior in aggregation. That is, it suffices to satisfy one of the two algorithms, k or l, to have a good global score. A null value of the interaction index ($I_g(a_k, a_l) = 0$) induces no interaction. In this case, a linear aggregation is sufficient to have a good global score.

1.4.4 Borda count fusion

The Borda Count is a single-winner election method in which voters rank candidates in order of preference [63]. The Borda Count determines the winner of an election by giving each candidate a certain number of points corresponding to the position in which he or she is ranked by each voter. Once all votes have been

counted the candidate with the most points is the winner. Because it sometimes elects broadly acceptable candidates, rather than those preferred by the majority, the Borda Count is often described as a consensus-based electoral system, rather than a majoritarian one.

1.4.4.1 General approach

One approach to combine multiple classifiers with a supervised learning system using rank weighting is to consider each discrimination algorithm to be a voter, and each observation in the training set to be a candidate. Given K algorithms $e_1, ..., e_k$ and N training samples $x_1, .., x_N$, each algorithm maps samples to their confidence values, elements of $\mathbb{R}$. The number of points given to candidates for each ranking is determined by the number of candidates standing in the voting. For each algorithm e_i and for each candidate x_j, a rank $r_i(x_j)$ is assigned to x_j if $e_i(x_j)$ has a confidence value greater than exactly $r_i(x_j) - 1$ other candidate alarms. In other words, a candidate will receive N points for a first preference, $N - 1$ points for a second preference, $N - 2$ for a third, and so on. Thus, r_i is a map from the confidence values assigned by algorithm e_i into the set $\{1, ..., N\}$. The final result of applying the Borda Count to x_j is expressed by the following expression:

$$r(x_j) = \frac{1}{KN} \sum_{k=1}^{K} r_i(x_j) \tag{25}$$

Note that this result is normalized to yield a value in the range [0,1].

1.4.4.2 Weighted Borda count approach

If there are evidences that algorithms e_i and e_j have differing predictive abilities, say e_i is more likely to be correct than e_j , then one should use this prior information and assigns weights w_i and w_j to these algorithms,such that $w_i > w_j$. In general, a weighted Borda scheme assigns a weight w_k to each algorithm e_k such that

$$\sum_{k=1}^{K} w_k = 1 \tag{26}$$

and the weighted Borda Count assigns confidence r to x_j as follows:

$$r(x_j) = \frac{1}{KN} \sum_{k=1}^{K} w_k r_i(x_j) \tag{27}$$

The main advantages of the Borda based fusion is that it makes no assumptions about the underlying distributions of the confidence value assignments. In addition, it maps each of the confidence distribution to a uniform distribution, thus providing a reasonable method for combining decision statistics.

1.5 Conclusion

In this chapter, we explained different spatio-temporal features, classifiers and decision level fusion methods that we have used in our project for the purpose of CPR scenes retrieval. More explanation of the implementation of these algorithms in our framework will be described in the next chapters.

2 Framework for scene retrieval based on activity detection

In this chapter, we elucidate our framework to detect CPR activity from medical simulation videos and retrieve these scenes. Medical simulation videos are recorded by the supervising physicians for educational purposes. Then, they are evaluated, and the results are used to debrief the trainees about their performance during the simulated emergency scenario. The objective is to provide answers to queries that are of interest to the physician supervising the training sessions such as: "Show me all the scenes that have a CPR action from a given video simulation training." An overview of the proposed system is illustrated in figure(2.1). Every input video will be segmented into overlapping volumes on the t-axis. From each volume, region of interest which may contain the movement of hands performing CPR and the nearest face to this region are extracted in order to track them, describe them using spatio-temporal features and classify them with different classifiers and finally fuse the information (confidences) to make the final decision.

In the following, we present the different steps for the preprocessing and features extraction of our proposed framework and more details about our method for CPR retrieval and results are presented in the next chapter.

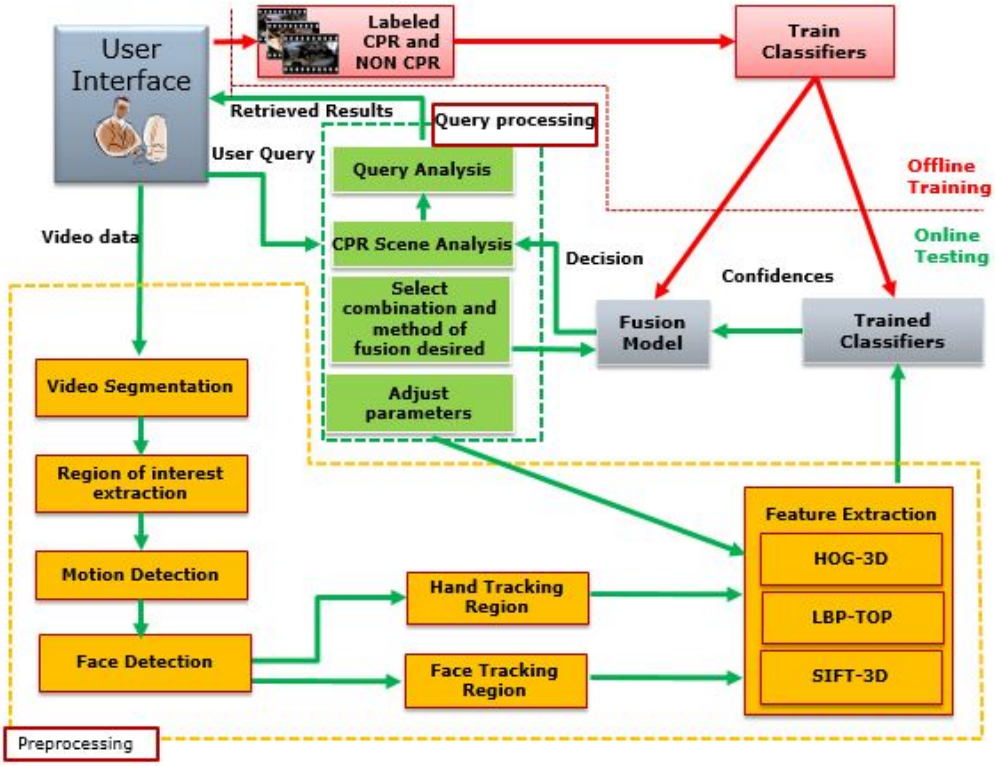


Figure 2.1: Overview of the proposed framework.

2.1 Preprocessing

2.1.1 Video segmentation

Since we are using spatio-temporal features, we are interested in extracting a suitable volume for good description and to be able to classify it if it belongs to CPR activity or not. In order to choose the best value for the side length of overlapping volumes along the t-axis, we rely first on the optical flow in order to have a clear idea about the motion around the hand of the person performing the CPR activity. In the following, we outline The Horn–Schunck method ([64]) that we used to

estimate the optical flow.

The Horn-Schunck algorithm assumes smoothness in the flow over the whole image. Thus, it tries to minimize distortions in flow and prefers solutions which show more smoothness.

The flow is formulated as a global energy functional which is then sought to be minimized. This function is given for two-dimensional image streams as:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x\mathrm{d}y \tag{28}$$

where I_x , I_y and I_t are the derivatives of the image intensity values along the x, y and time dimensions respectively, $\vec{V} = [u(x,y), v(x,y)]^\top$ is the optical flow vector, and the parameter α is a regularization constant. Larger values of α lead to a smoother flow. The function in (28) can be minimized by solving the associated multi-dimensional Euler-Lagrange equations:

$$\begin{cases} \dfrac{\partial L}{\partial u} - \dfrac{\partial}{\partial x}\dfrac{\partial L}{\partial u_x} - \dfrac{\partial}{\partial y}\dfrac{\partial L}{\partial u_y} = 0 \\ \dfrac{\partial L}{\partial v} - \dfrac{\partial}{\partial x}\dfrac{\partial L}{\partial v_x} - \dfrac{\partial}{\partial y}\dfrac{\partial L}{\partial v_y} = 0 \end{cases} \tag{29}$$

where L is the integrand of the energy expression, giving

$$\begin{cases} I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u = 0 \\ I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v = 0 \end{cases} \tag{30}$$

$\Delta = \dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2}$ denotes the Laplace operator. In practice, the Laplacian is approximated numerically using finite differences, and may be written $\Delta u(x,y) = \overline{u}(x,y) - u(x,y)$ where $\overline{u}(x,y)$ is a weighted average of u calculated in a neighborhood around the

pixel at location (x, y). Using this notation, the system of equations in (30) may be written as:

$$\begin{cases} (I_x^2 + \alpha^2)u + I_x I_y v = \alpha^2 \overline{u} - I_x I_t \\ I_x I_y u + (I_y^2 + \alpha^2)v = \alpha^2 \overline{v} - I_y I_t \end{cases} \tag{31}$$

The system of equations in (31) is linear in u and v and may be solved for each pixel in the image. However, since the solution depends on the neighboring values of the flow field, it must be repeated once the neighbors have been updated. The following iterative scheme is used:

$$\begin{cases} u^{k+1} = \overline{u}^k - \dfrac{I_x(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \\ v^{k+1} = \overline{v}^k - \dfrac{I_y(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \end{cases} \tag{32}$$

The superscript k denotes the iteration number.

So, as we can notice, optical flow measures the change in the velocity in terms of speed and direction at each pixel location. For the purpose that we mentioned above, we extract manually region around the hand from CPR scenes labeled manually and we compute the optical flow for all pixels within it. Then, the optical flow of the region is estimated as the average of the optical flow of all of its pixels. Figure (2.10) displays the average optical flow of the bounding box selected manually for a CPR scene region that involves CPR activities:

After analyzing the average optical flow for many CPR scenes and using our knowledge about the typical frequency of CPR and the number of frames per second in the record video, we fix the number of frames for each volume of training

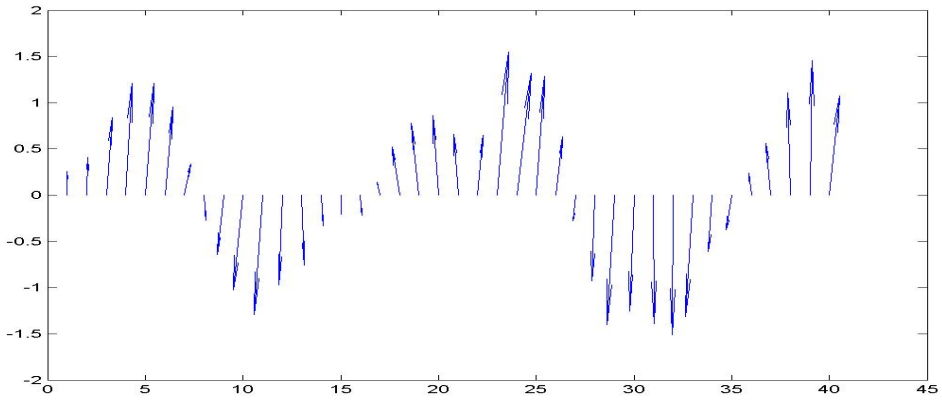


Figure 2.2: Average optical flow of CPR sequence.

and testing to be 18 frames. A sequence of 18 frames(approximately 0.6seconds) of CPR action typically corresponds to one CPR cycle (up-down movement) which is essential for capturing the rhythmic cycle of the CPR activity. Therefore, every query video will be segmented into overlapping volumes of length 18 frames each.

2.1.2 Region of interest extraction

Our objective consists of identifying CPR scenes. This action typically involves the trainee hands placed on the center of the chest of the Human-like mannequins. Thus, in order to reduce the computational complexity of our approach, we start by detecting the chest of the Human-like mannequins.

All subsequent processing steps will be applied to only this region of interest.

We start by identifying regions in the image with skin-like colors. To achieve this task, we use a simple but efficient skin pixel classifier to discriminate between skin and non-skin regions. This classifier needs to be trained to learn the characteristics of skin regions.

We have collected 60 images in which the chest of mannequin is present. For each image, we select small (5x5) patches as shown in figure (2.3).

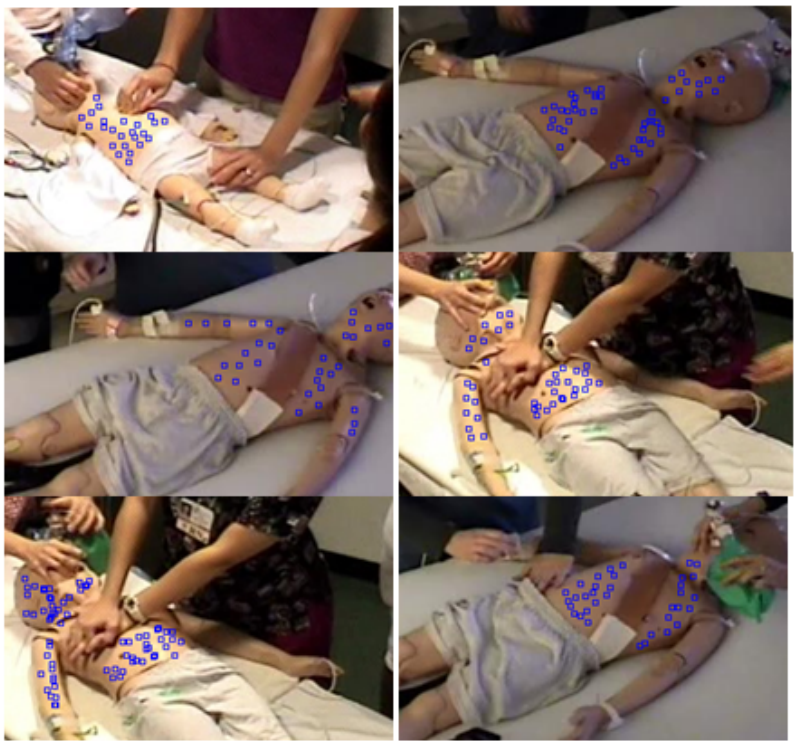


Figure 2.3: Small areas selected from mannequin for the training.

First, each area is mapped into the HSV color space. The H(ue) dimension represents the "color", the S(aturation) dimension represents the dominance of that color and the V(alue) dimension represents the brightness. We use the HSV color space instead of RGB because it is more related to human color perception. We only use Hue and Saturation components to detect skin. Second, the color distribution of all pixels in all training image patches is fitted by a Gaussian model with mean μ and a covariance matrix C. The resulting Gaussian probability density

function that best fits the data is then:

$$p(x \mid \mu, C) = \frac{1}{(2\Pi)^{\frac{d}{2}}(det(C)^{\frac{1}{2}})} exp(-\frac{1}{2}D^2) \tag{33}$$

where

$$D^2 = (x-\mu)C^{-1}(x-\mu)^t \tag{34}$$

is the square Mahalanobis distance and d is the dimensionality of the Gaussian function ($d = 2$ in our case).

To detect the chest of mannequin in a new test image, we map the image into skin likelihood map where the likelihood of each pixel is computed using (33). Then, the likelihood image is smoothed by a low-pass filter.

Finally, the image region with the maximum likelihood score is identified as the mannequin chest. Here, the likelihood score in computed as the sum of the likelihood of each pixel within a region.

More specifically, we scan the entire image with a window of 130 $\times$ 130 pixels and an overlap of 10 pixels and keep only the region which has the highest confidence to be chest.

Figure (2.4) displays an overview of the proposed skin detection process.

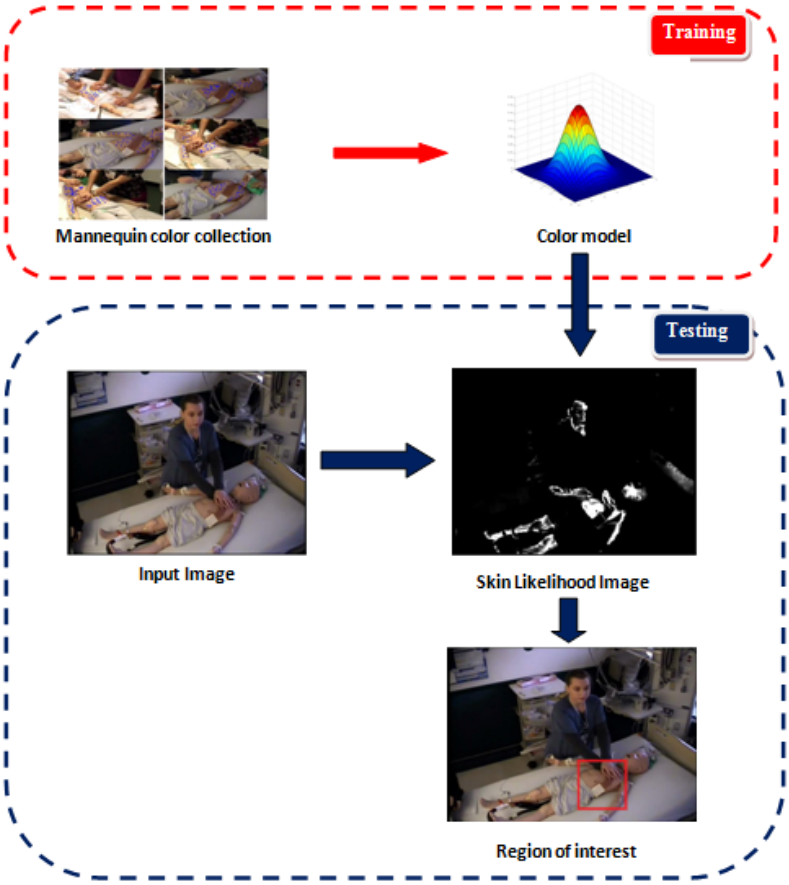


Figure 2.4: Overview of the region of interest extraction process.

2.1.3 Motion detection

The simulation videos may have several static scenes with no significant activities. To improve the efficiency of our approach, we limit the processing to scenes that involve some motion. Thus, the third preprocessing step in our proposed framework is to detect and discard scenes that do not involve motion in the regions of interest. To achieve this task, we use a simple motion detection method.

Suppose that we have Q consecutive image frames $\{I(t), t = 1..Q\}$. First, we

(a) example of binary image in the presence of motion

(b) example of binary image in the absence of motion

Figure 2.5: Binary images in the absence and the presence of motion.

calculate the mean image B by averaging the corresponding pixels, i.e.:

$$B_{(x,y)} = \sum_{t=1}^{Q} \frac{I_{(x,y)}(t)}{Q} \tag{35}$$

After calculating the average image B we can then subtract it from the first image $I(1)$ and threshold it. The resulting binary image $V(x, y)$ is defined as:

$$V(x,y) = \begin{cases} 1, if \mid I(x,y,1) - B(x,y) \mid \geq Th \\ 0, \text{otherwise} \end{cases} \tag{36}$$

The figure (2.5a) shows an example of a binary image obtained for $Q = 18$ images of interest region picked from a CPR scene and the figure(2.5b) shows an example of a binary image obtained for $Q = 18$ images of interest region which have no

motion.

2.1.4 Face detection

CPR activities in our video data collection are correlated with a human action. In fact, when CPR is being administered, the head of the person performing the action tends to parallel motion of his hands (figure(2.6)). Thus, to improve the accuracy

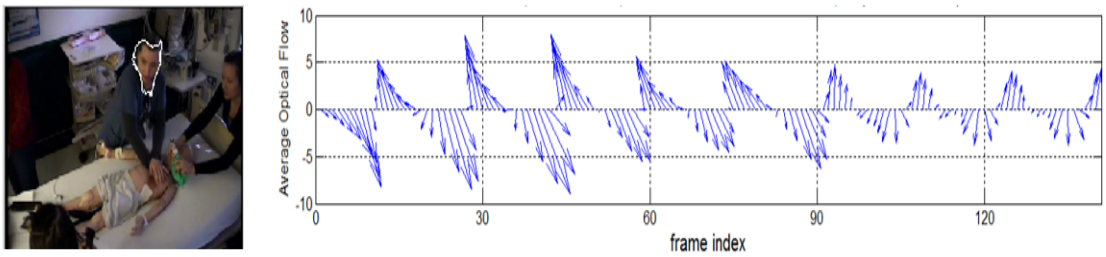


Figure 2.6: Average optical flow for face region extracted manually.

of our CPR video scene retrieval system, we propose detecting and analyzing the motion of the nearest person to the region of interest in addition to analyzing the motion in the region of interest.

Figure (2.7) illustrates our approach to detect faces in the video volume. We start by looking for faces in the first frame of a given video. We use the Viola and Jones [65] face detection method. This algorithm can detect faces in real time with very low false alarm rate using Haar-like features trained by the AdaBoost algorithm [66].

If no face is detected in the video volume's first frame, we keep processing adjacent frames until a face is found or all frames have been processed because sometimes the face is not detected in some frames and detected in others. If no face is detected,

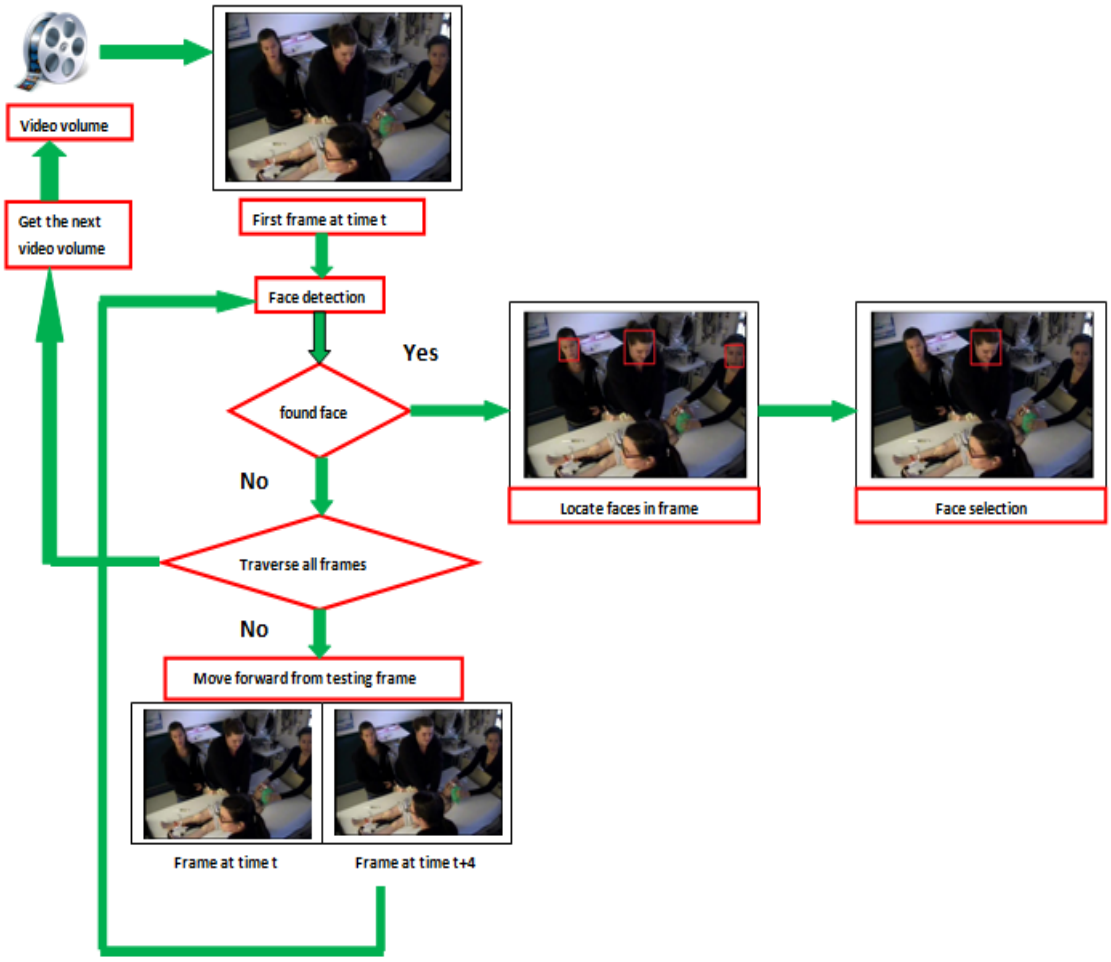


Figure 2.7: Overview of the face detection and selection process.

the video volume will be flagged as a "no-face" video. Otherwise, if multiple faces are found, we keep only the one nearest to the detected region of interest.

Suppose that x_r, y_r are the coordinates of the center of the region of interest and x_f, y_f are the coordinates of the center of the window of the face detected. The distance ρ is defined as:

$$\rho = \sqrt{w * (x_r - x_f)^2 + (y_r - y_f)^2} \tag{37}$$

where $w \geq 1$ is a weight to bias the distance towards horizontal distance since we expect the face to be above the region of interest.

2.2 Feature extraction

The advantage of using spatio-temporal features is that the variations in the space and time dimensions can be represented using a single feature vector. Our objective is to identify CPR activity scenes without video shot segmentation or tracking. This is presumed to be possible with the use of features that represent the spatio-temporal shape of the activity (HOG3D) and spatio-temporal texture of the activity (LBP-TOP) and local informations in space and time in interest points (SIFT-3D). In the rest of this chapter, we provide a more understanding on our methods of describing video volumes using these features.

2.2.1 Spatio-temporal histogram of oriented gradients - HOG3D

The HOG3D is an efficient descriptor to represent the pixel intensity variations in spatial and temporal dimensions. It jointly encodes both appearance and motion information. The HOG3D descriptors are based on spatio-temporal gradient orientations, and hence they are robust to changes in illumination and minor deformations. While performing a CPR activity, the hands of the actor will typically be in one specific posture as shown in figure (3.6)(a).

In our proposed work, we use the HOG3D features to capture the space-time orientation of gradients that represents the structure of the CPR action cycle. The HOG3D features are computed as described below.

We divide each video volume $c = (x_c, y_c, t_c, w_c, h_c, l_c)^T$ where $(x, y, t)^T$ denotes the

position and w, h and l its width, height and length respectively, into cuboids of size $M \times M \times N$ as shown in figure(3.6). This will create the set of $M \times M \times N$ sub-blocks, b_j, each of size $S_1 \times S_2 \times S_3$.

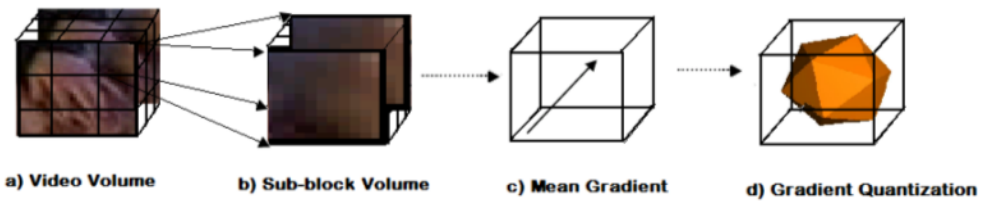


Figure 2.8: Steps for computing HOG3D features.

First, we compute the gradients along x, y, and t directions at every pixel. Then, for each pixel, the gradient orientation is quantized to q_{b_i} by projecting the (dx, dy, dt) vector on a 20-dimensional regular polygon (or icosahedron), with the gradient magnitude as its weight. Then, for each b_j, the weighted gradients are smoothed using a 3D Gaussian filter, with σ determined by the size of b_j. The resulting vector is then thresholded and normalized. The histogram h_c for the region c is found by accumulating the quantized gradients q_{b_i} into 20 bins as shown below

$$h_c = \sum_{i=1}^{S_1.S_2.S_3} q_{b_i} \tag{38}$$

Each of these histograms is then normalized using L2 norm within each sub-block. These histograms are finally concatenated to form the HOG3D descriptor of $M \times M \times N \times 20$ dimensions.

2.2.2 Local binary patterns - three orthogonal planes - LBP-TOP

Dynamic texture is the extension of texture to the temporal domain. The LBP-TOP feature combines the motion and appearance information of the underlying texture. In a typical CPR action cycle, the dynamic texture is expected to display similar pattern for different actors. Moreover, the texture features in a small local neighborhood of the volume are not only insensitive to the translation and rotation, but also robust with respect to the illumination changes. In order to capture the dynamic texture of the CPR cyclic activity we use the LBP-TOP feature. The LBP-TOP is a reduced representation of VLBP, where the local binary patterns are computed only for the three orthogonal planes XY, XT and YT. The XT and YT planes contain the temporal information in the volume. The three orthogonal planes intersect at the center pixel. The feature distributions from separate planes are computed and concatenated together to form the final feature histogram which serves as a global descriptor for the spatial and temporal features. These steps are illustrated in figure (2.9).

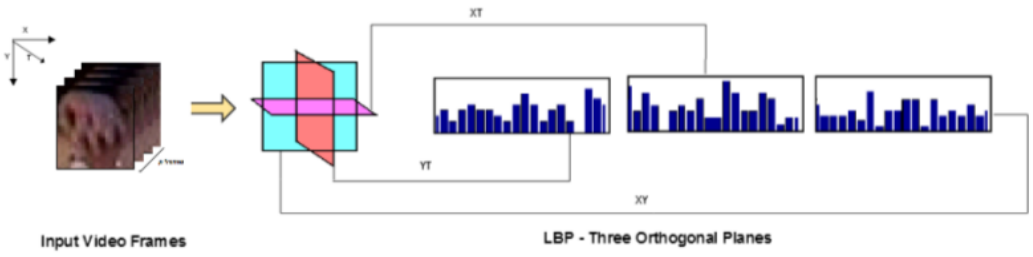


Figure 2.9: Steps for computing LBP-TOP features.

The number of neighborhood points P determines the size of the feature vector. A neighborhood of P points will create 2^P codes. So, the LBP-TOP feature vector

will be 3.2^P dimensional histogram. The radius in time axis that we have used is different from the radius in space axis because of the extend of variation of texture in the time relatively to the space axis. We have applied elliptical neighborhood instead of the conventional circular neighborhood for the XT and YT planes. Say, the radius in X, Y and Z axis are R_X ,R_Y, and R_Z, and the number of neighboring points in XY, XT, and YT planes are P_{XY} , $P_{XT}$and P_{YT} . If the coordinates of the center pixel are (x_c, y_c, t_c),the coordinates of neighboring points in XT plane is given by $(x_c - R_X \sin(2\Pi p/P_{XT}), y_c, t_c - R_T \cos(2\Pi p/P_{XT}))$ and coordinates of neighboring points in YT plane is given by $(x_c, y_c - R_Y \sin(2\Pi p/P_{YT}), t_c - R_T \cos(2\Pi p/P_{YT}))$. For any given video volume, the histogram can be defined as

$$H_{i,j} = \sum_{x,y,t} I(f_j(x, y, t) = i) \tag{39}$$

$i = 0 \ldots n_j - 1, j = 0, 1, 2$ where n_j is the number of different labels produced by the LBP operator in the jth plane (j = 0: XY, 1: XY and 2: YT), $f_i(x, y, t)$ express the LBP code of the center pixel (x, y, t), in the jth plane and

$$I(A) = \begin{cases} 1, \; if \; A \; is \; true \\ 0, \; if \; A \; is \; false \end{cases} \tag{40}$$

We gave also weights (W_{XY}, W_{XT}, W_{YT}) for every histogram in each plan before we concatenate them. We thought for this proposition in order to give more importance to the XT and YT plans which have more significant representation of the texture of CPR action. The final histogram is then normalized to get a coherent and generalized feature descriptor.

2.2.3 3D Scale-invariant feature transform (SIFT-3D)

SIFT-3D is Local spatio-temporal feature. That means first, interest points are extracted from 3-dimensional space-using detectors such as Harris3D [28], Cuboids [67] or Spatio Temporal Interest Points (STIP) [69] to locate spatio-temporal interest points. After detecting the interest points, a typical procedure is to extract a cuboidal volume of interest (VOI) around the interest point, and find spatio-temporal descriptor SIFT-3D within the VOI. The extracted descriptors are quantified with a pre-learned code book, and input videos are typically modeled with Bag of Visual Words [68]. This local descriptor is somewhat successful in capturing and representing local and repeatable properties of actions within a video. These properties make SIFT-3D robust to intra-class variability and deformation to a certain degree. However, this local descriptor cannot effectively discriminate between activities with different high-level motions.

Figure (2.10) summarizes the algorithm to describe a video volume using local descriptors SIFT-3D.

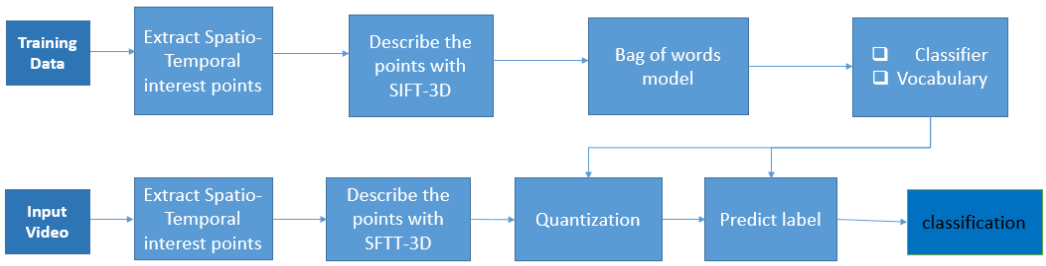


Figure 2.10: Overview of the proposed algorithm.

2.2.3.1 Detecting interest points

In the proposed approach, Harris3D is used to detect interest points. In the following, we explain the Harris3D method.

First, we introduce the spatial interest points and then we extend the idea for a spatio-temporal interest points.

So, suppose we have an image I, we can define L as

$$L(x, y) = g(x, y, \sigma^2) * I(x, y) \tag{41}$$

with

$$g(x, y, \sigma^2) = \frac{1}{2\pi\sigma^2} exp(\frac{-(x^2 + y^2)}{2\sigma^2}) \tag{42}$$

So as we can notice, g is a Gaussian kernel that defines the neighborhood for each position (x, y).

The idea of the Harris interest point detector is to find spatial locations where I has significant changes in both directions. For a given scale of observation, such points can be found using a second moment matrix integrated over a Gaussian window.

$$M = g * (\nabla L(\nabla L)^T) \tag{43}$$

The equation (43) can be rewritten as:

$$M = g * \begin{pmatrix} L_x^2 & L_x L_y \\ L_x L_y & L_y^2 \end{pmatrix} \tag{44}$$

L_x and L_y are Gaussian derivatives computed at local scale. The second moment descriptor can be thought of as the covariance matrix of a two-dimensional distribution of image orientations in the local neighborhood of a point. Hence, the eigenvalues $\lambda 1$, $\lambda 2$, ($\lambda 1 \leq \lambda 2$) of M constitute descriptors of variations in I along the two image directions.

Specifically, two significantly large values of $\lambda 1$, $\lambda 2$ indicate the presence of an interest point. To detect such points, we use the function of Harris to detect positive maximum of the corner function

$$H = \det(M) - k\text{trace}(M)^2 \tag{45}$$

(45) can be rewritten as:

$$H = \lambda 1 \lambda 2 - k(\lambda 1 + \lambda 2)^2 \tag{46}$$

At the positions of the interest points, the ratio of the eigenvalues $\alpha = \frac{\lambda 2}{\lambda 1}$ has to be high.

Now for the spatio-temporal domain, f becomes a function of x, y and t, therefore, the Gaussian kernel is now a function of the variables x, y and t.

$$g(x, y, t, \sigma^2, \tau^2) = \frac{1}{\sqrt{(2\pi)^3 \sigma^4 \tau^2}} exp(\frac{-(x^2+y^2)}{2\sigma^2} - \frac{t^2}{2\tau^2})$$

The neighborhood is now a volume and similar to the spatial domain, we consider a spatio-temporal second-moment matrix, which is a 3-by-3 matrix composed of first order spatial and temporal derivatives averaged using a Gaussian weighting

function

$$M = g * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_t L_y \\ L_x L_t & L_t L_y & L_t^2 \end{pmatrix} \tag{47}$$

To detect interest points, we search for regions in f having significant eigenvalues $\lambda 1$,$\lambda 2$, $\lambda 3$ of M. The Harris corner function defined for the spatial domain is extended into the spatio-temporal domain by combining the determinant and the trace of M as follows:

$$H = det(M) - ktrace(M)^2 \tag{48}$$

(48) can be rewritten as:

$$H = \lambda 1 \lambda 2 \lambda 3 - k(\lambda 1 + \lambda 2 + \lambda 3)^3 \tag{49}$$

To show how positive local maximum of H correspond to points with high values of $\lambda 1$,$\lambda 2$,$\lambda 3$ ($\lambda 1 \leq \lambda 2 \leq \lambda 3$), we define the ratios $\alpha = \frac{\lambda 2}{\lambda 1}$ and $\beta = \frac{\lambda 3}{\lambda 1}$ and (49) can be rewritten as

$$H = \lambda 1^3(\alpha\beta - k(1 + \alpha + \beta)^3) \tag{50}$$

From the requirement $H \geq 0$, we get $k \leq \frac{\alpha\beta}{(1+\alpha+\beta)^3}$ and it follows that k assumes its maximum possible value $k = \frac{1}{27}$ when $\alpha = \beta = 1$.

For sufficiently large values of k, positive local maximum of H correspond to points with high variation of the image values along both the spatial and the temporal

directions.

2.2.3.2 Bag of words model

The bag-of-words model is a simplifying representation used in natural language processing and information retrieval. In this model, a text is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity. Recently, the bag-of-words model has also been used for computer vision. The bag-of-words model is commonly used in methods of document classification where the occurrence of each word is used as a feature for training a classifier.
After the step of feature extraction, each video is a collection of vectors of the same dimension where the order of different vectors is of no importance. The final step for the BoW model is to convert vector-represented patches to "codewords" (analogous to words in text documents), which also produces a "codebook" (analogy to a word dictionary).
To quantize local descriptors into visual words, we must first generate visual vocabulary. Like document in text retrieval, each video is represented as a sparse vector of occurrences through local descriptor quantization, but first we have to build the vocabulary.
A visual vocabulary is often generated by scalable clustering methods which view local descriptors as raw input. By clustering, we treat each cluster as a unique visual word from the vocabulary. It is often the case that centroids of these clusters are stored so that later each video including the query can be easily quantized into visual words. It is worth noticing that generating vocabulary is done off-line and may be time consuming since the number of local descriptors could be huge.
The generating of visual vocabulary could be done by a k-means clustering and

the quantization into visual words could be done using an Euclidean distance.

2.2.3.3 Steps of the developed algorithm

First of all, we have to detect the spatio-temporal interest points using the method mentioned earlier. We can limit the number of points detected to be M interest points for every video volume, it means we take the M strongest points and neglect the others.

Once the points are detected, the second step is to describe the spatio-temporal region around the points using the 3D SIFT descriptor. The length of the descriptor is based on the number of sub-histograms, and the number of bins used to break represent the θ and ϕ angles.

The problem is we can't concatenate the SIFT-3D features of the points detected to build our final descriptor of every video because we can't determine exactly the number of interest points, we can just limit the number and also the concatenation will not be good representation of the video. The solution is to construct the bag of words (BOW) from the detected points and then build histogram for every video data.

Therefore, the descriptors gathered from all the interest points are then quantized by clustering them into a pre-specified number of clusters. For this task, we have used k-means clustering method. The result cluster centers are now called 'words', while the collection of these cluster centers is referred to as the 'spatio-temporal word vocabulary'. Now that our vocabulary is computed, the 3D SIFT descriptors from the videos are matched to each 'word' and the frequency of the words in each video is accumulated into a histogram. This word frequency histogram, referred to as a 'signature', is used to generate the final representation of the video. So, we

have finally our descriptor for every video which is the histogram whose length is equal to the number of cluster centers obtained by k-means.

2.3 Conclusion

In this chapter, we have described all preprocessing steps of our approach and then our methods to extract features from video volumes. In the next chapter, more explanation of our approach will be given and all experimental results are shown. We compare also our system with the method illustrated in [21], as it most closely resembles our application and used similar experimental settings as ours.

3 Experiments and results

3.1 Data sets

The CPR simulation videos are provided by the (SPARC) working group at Kosair Childrens Hospital, Louisville. The duration of these simulation sessions is roughly 30 minutes, with an average frame rate of 29 frames per second. The frames have a resolution of 720×480 pixels. The details of every video are listed in table (1). To provide an idea about the content of the videos, we provide sample frames from the two videos that we analyzed in table (2).

Video	Duration	Number of frames	Number of CPR Scenes	Average number of frames per CPR scene	Std. Deviation of number of frames per CPR scene
CPR1	19m28s	35000	36	258,1	132,2
CPR2	16m1s	28831	36	259,7	173,3

Table 1: Details of the simulation videos provided by the SPARC.

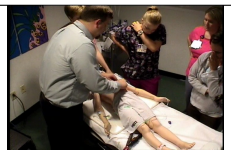
Video	Sample1	Sample2	Sample3	Sample4
CPR1				
CPR2				

Table 2: Sample frames from two medical simulation videos.

3.2 Experimental design

In order to identify CPR scenes from medical simulation videos, we have built two systems:

1. The first system is for classification in which we manually collect all the data for training, testing and validation. The input is a video volume and it just try to classify it if it belongs to CPR activity or not which means that the output is a confidence value between 0 and 1. This system is build to have an idea about the effectiveness of our spatio-temporal features, to select best parameters for features and classifiers and to build models to be used by the second system of retrieval.

2. The second system is for retrieval (figure (1)), the input is the whole record video and the output is a timeline that indicates the occurrences of CPR scenes. The record video is segmented into overlapping volumes of 18 frames each and from each volume a region of interest is extracted with scanning the entire first frame with a window of 130×130. Then, we test the volume obtained for the region of interest ($130 \times 130 \times 18$) if there is some motion using our algorithm of motion detection : If not , we assign 0 for the confidence, otherwise we extract the nearest face to the region of interest with a window of 100×100 and we describe the two obtained sub-volumes of $130 \times 130 \times 18$ and $100 \times 100 \times 18$ using our spatio-temporal features and then we assign a confidence for every sub-volume for every feature and every classifier (SVM and KNN) which means that we can obtain 12 confidences (information) of that volume of $720 \times 480 \times 18$ and finally we fuse these informations to obtain

a better decision.

The both systems designed require features extraction and training. We outline details of these two steps in the next sections.

3.3 Analysis of the proposed system

3.3.1 CPR scene classification

In this section, we describe the first system that we build for classification, testing and validation. We start by manually labeling the CPR scenes and non-CPR scenes for the two videos CPR1 and CPR2.

In order to have a training data where the start and end frame of CPR sequences are consistent, we have detected the frames in which the average optical flow within a region of interest changes sign from negative to positive (figure(3.1)) so that all CPR samples in the training data start with an upward motion and finish with a downward motion.

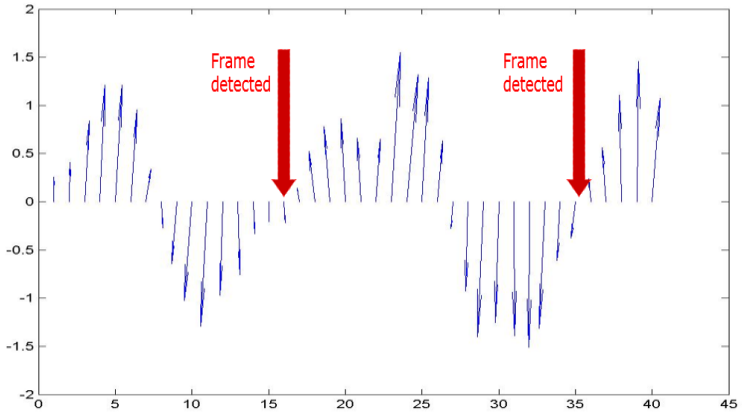


Figure 3.1: Frames detected for training.

In the first experiment, we analyze the effectiveness of HOG3D features for CPR activity detection. After identifying frame sequences or video volumes that contain CPR activity from the video, a 3D bounding box containing only the CPR activity is manually extracted as shown in figure(3.2). After visual inspection of few CPR scenes, the size of the bounding box is fixed to be 55×55 (approximately the size of the hand for most CPR sequences).

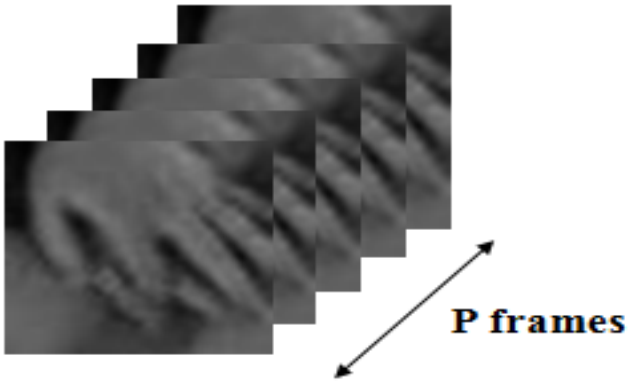


Figure 3.2: Example of a 3D bounding box of a region that correspond to a CPR action sequence : $55 \times 55 \times 18$.

This forms our positive dataset for training, denoted as Tr^P. For non-CPR sequences, we extract blocks of size 55×55 at random spatial and temporal locations that do not overlap with blocks of CPR activity regions. It is important that the extracted non-CPR sequences involve other activity or motion areas in the scene. This will be our negative training dataset, denoted as Tr^N.
More than 20 CPR and non-CPR scenes each, from videos CPR1 and CPR2, are considered to build the training dataset. The HOG3D features are then computed for the bounding boxes volumes. We obtained a total of approximately 800 features for positive training dataset and 1500 features for the negative training dataset.

To evaluate the performance of the features, we use k fold cross validation. In k fold cross validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once.

The main purpose of using K fold cross validation is to find the best parameters for classifiers with the training data extracted manually and find those which give the best accuracy in order to use them for the second system of scene retrieval.

We use a 10-fold cross validation with the training dataset. With each fold, a subset of Tr^p and Tr^p is used for training and the remaining data is used for testing. A binary SVM classifier, with a linear kernel, is used to build the model, with the computed HOG3D features for CPR activity and non-CPR activity. The SVM classifier with 10-fold cross validation gives an accuracy of 89% with only 11% error rate. The K nearest neighbor (KNN) classifier with 10-fold cross validation gives the best accuracy 88% with only 12% error rate with the Euclidean distance and 7 nearest neighbors.

We used 10-fold cross validation to find the parameters that give the highest accuracy. We find M=1 and N=2 are the best parameters for dividing the volume into sub-blocks, that means that every volume is then divided into two sub-blocks. We explain these parameters by the fact that for the positive training data the

mean gradient over the first sub-block represent the upward motion and the mean gradient of the second sub-block represent the downward motion and this can help to better describe the volume. The histogram for each sub-block is found by accumulating the quantized gradients into 20 bins resulting in a feature of a length $20 * 1 * 1 * 2 = 40$. In the second experiment, we analyze the effectiveness of LBP-TOP features for CPR activity detection. The framework for extracting the LBP-TOP features is the same as that of the HOG3D which means the same positive and negative training data used. The SVM classifier (linear kernel) with 10-fold cross validation gives an accuracy of 85% while the KNN classifier with 10-fold cross validation gives an accuracy of 88% with the Euclidean distance and 8 nearest neighbors.

For the LBP-TOP, we use elliptical neighborhood instead of the conventional circular neighborhood for the XT and YT planes. The neighborhood radius we used for this experiment was $x_{radius} = y_{radius} = 1$ and $t_{radius} = 2$ and we choose a neighborhood of 8 pixels for the XY, XT, and YT planes. We obtain finally a feature of length $3 * 2^8 = 768$. For the weights given to each plane, $W_{XY} = 0.25$, $W_{XT} = 0.35$ and $W_{YT} = 0.4$. We explain the significant weight of YT plane by the fact that the motion and especially for the up and down movement is more represented in the YT plan.

In the third experiment, we analyze the effectiveness of SIFT-3D features for CPR activity detection. From video volumes that contain CPR activity, a bounding box of $130 \times 130 \times 18$ of the region of interest (chest of mannequin and hand of person performing CPR) is manually extracted as shown in figure(3.3) to form our positive training dataset. For non-CPR sequences, we extract blocks of size $130 \times 130 \times 18$ at random spatial and temporal locations that do not overlap with

blocks of CPR activity regions. This forms our negative training dataset. Then,

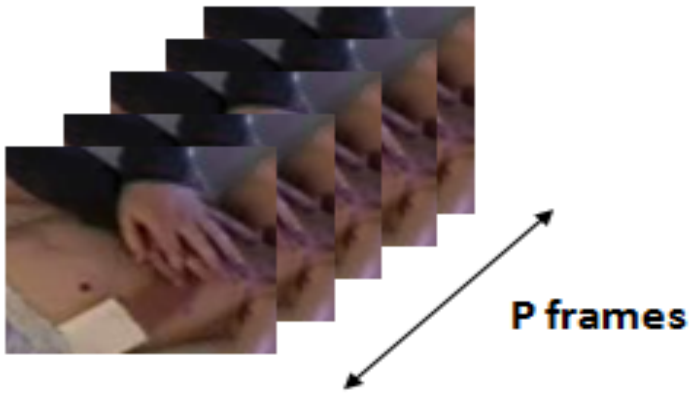


Figure 3.3: Example of a 3D bounding box of a region that correspond to a CPR action sequence: $130 \times 130 \times 18$.

we describe our training data with SIFT-3D descriptor. We obtained a total of approximately 7900 SIFT-3D descriptors of 7900 interest points detected for positive training dataset and 7450 for the negative training dataset.

For the SIFT-3D, for each volume we extract interest points using Harris operator described above. We limit the number of interest points to be 15 which means we take the 15 strongest points and neglect the others.

Then, we describe the spatio-temporal region of $10 \times 10 \times 6$ around each point using the proposed 3D SIFT descriptor. We used $2 \times 2 \times 2$ configurations of sub-histograms and 8×4 histograms to represent θ and ϕ. This yields descriptors of length 256 for every interest point.

The next step which is so crucial is to fix the number of clusters and this step effect extensively the results. For this task, we decided to gather all the descriptors of positive data and cluster them in C clusters and do the same for negative data instead of clustering the mixed descriptors and by doing so the accuracy increases.

In order to have a better idea about the choice of the number of centers, we fix it and then we visualize the centers of the clusters using principal component analysis (PCA) to reduce the dimension from 256 to 2. figure(3.4) shows 40 centers of CPR activity in blue and 40 centers on non-CPR activity in red. So, we variate the

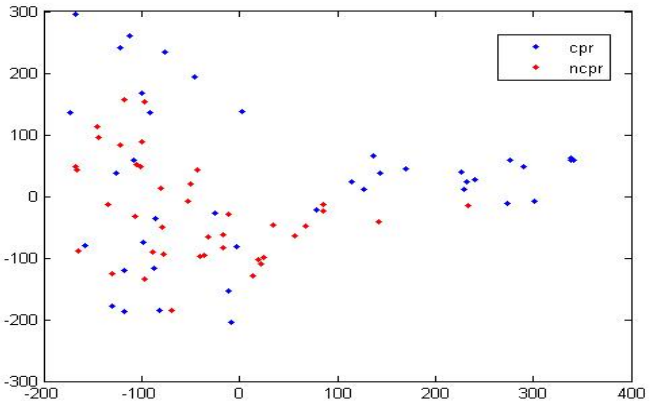


Figure 3.4: Example of visualization of 80 centers of clusters.

number of clusters until we find a better separation between the two group of centers. Finally, we decided to fix it to 15 centers for each group and the visualization of centers are shown in figure (3.5).

Now our vocabulary is computed, the 3D SIFT descriptors from the video volumes are matched to each 'word' and the frequency of the words in each video is accumulated into a histogram. We obtain word frequency histogram whose length is 30 to represent each video and we obtain now the same number of features of HOG3D and LBP-TOP.

Finally, we train a third SVM classifier with linear kernel with the new set of features. The SVM classifier with 10-fold cross validation gives an accuracy of 83% while the KNN classifier with k=5 and the Euclidean distance gives an accuracy

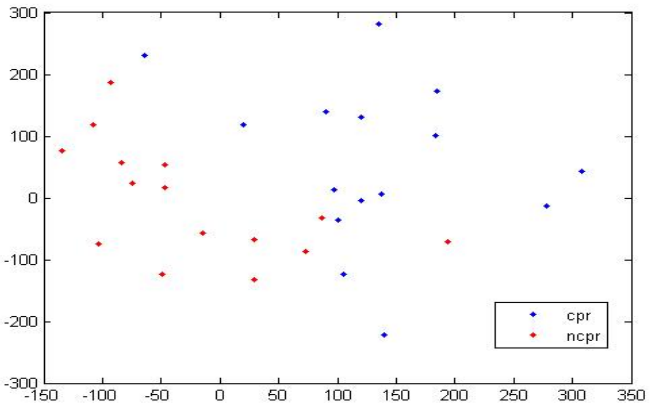


Figure 3.5: Example of visualization of 30 centers of clusters.

of 85%.

The previous work in [21] used an HMM classifier which produced an accuracy of 80% with 10-fold cross validation so we can conclude that our system for scene classification outperform previous work.

Based on the results of the first system, we can conclude that the spatio-temporal features and classifiers that we have used are very efficient to discriminate between CPR and non-CPR activity compared to the previous work. However, in the first system, the region of interest and the region of the hand are extracted manually which will be not the case in the second system of scene retrieval.

3.3.2 CPR scene retrieval

3.3.2.1 Hand-tracking

To evaluate the performance of our method to extract the region of interest, we build automatic system to extract region of interest every 50 frames from CPR1 and CPR2 and count manually the number of miss detection.

The number of frames tested for this task is 1090 for CPR1 and 570 for CPR2 and the number of miss detection is 41 for CPR1 and 26 for CPR2 which means 96,23% accuracy for CPR1 and 95,43% for CPR2.

The video to be tested is first divided into overlapping volumes of 18 frames each. For the HOG3D and LBP-TOP features, as described above bounding boxes containing the region of interest are further divided into $w \times h \times l$ overlapping sub-blocks. In our experiments, we set $w = h = 55$ and $l = 18$. We provide an overlapping of 20 pixels within the XY-grid resulting in 16 overlapped sub-blocks in each $130 \times 130 \times 18$ volume and an overlapping of 15 frames within the temporal grid. The features are computed for each of the sub-block.

The test data is then classified using the trained linear SVM classifier or KNN classifier of the first system, which classifies them into CPR class or non-CPR class based on a certain classification confidence. Since we used overlapping sub-blocks, the confidence in the adjacent sub-blocks are averaged out in the overlapping area. Now we have the confidence values at each pixel, which spans over the entire volume of 18 frames in the given sequence. The confidence of each grid window in any frame is then calculated as the average confidence at corresponding grid window locations, of all video volumes, in which a given frame is part of. Each frame is classified into belonging to CPR or non-CPR scene based on the average confidence of the positively classified grid window. This filtering removes the influence of most of the incorrectly classified grid window volumes.

We try to detect the CPR action by describing much smaller volumes than the volume of region of interest and by dong so, the accuracy increases while increasing also the running time. The SIFT-3D has higher running time compared to HOG3D and LBP-TOP, that's why for SIFT-3D, we describe the volume of region

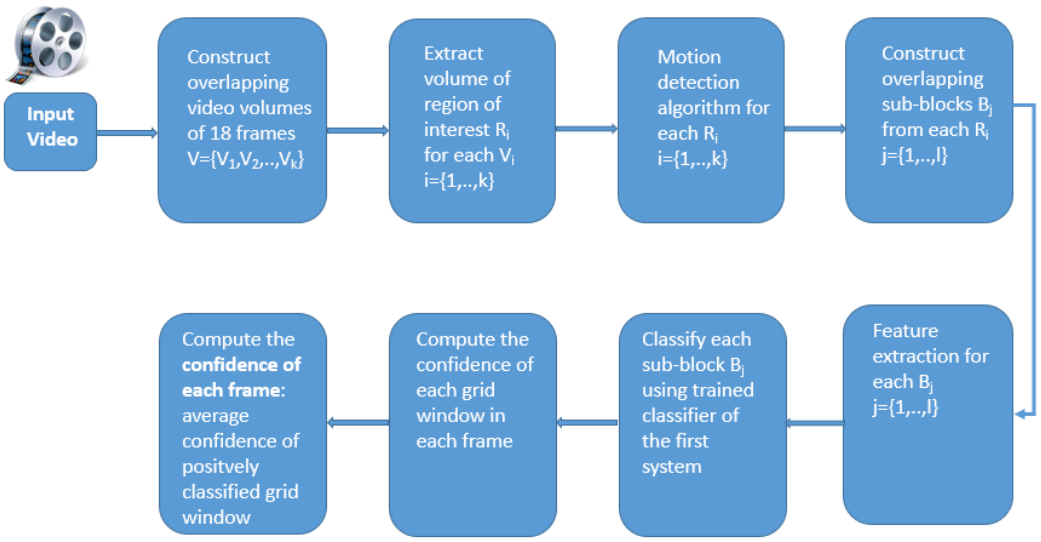


Figure 3.6: Overview of the main steps for CPR scene retrieval system.

of interest without dividing it into sub-blocks.

To evaluate the performance of the proposed method to identify and retrieve CPR scenes, we use the receiver operating characteristic (ROC) curve. ROC is a method to evaluate the expected loss using a range of loss functions, while varying the ratio of costs of false positives and false negatives. The ROC is obtained by plotting the probability of detection (PD) against the probability of false alarms (PFA).

We limit our training data only for the positive and negative 3D bounding boxes extracted manually located only in the first 5 minutes of CPR1 and the first 5 minutes of CPR2 and we use the same parameters cited above for HOG3D and LBP-TOP. For KNN we use the same parameters obtained with cross-validation in the first system and we build two new SVM model for HOG3D and LBP-TOP using the new training data. Finally, we test the remaining data of the CPR1 and CPR2.

The ROC curves obtained using HOG3D features on SVM model and using KNN for the test of CPR1 are shown in figure(3.7a) and for the test of CPR2 is shown in figure(3.7b)

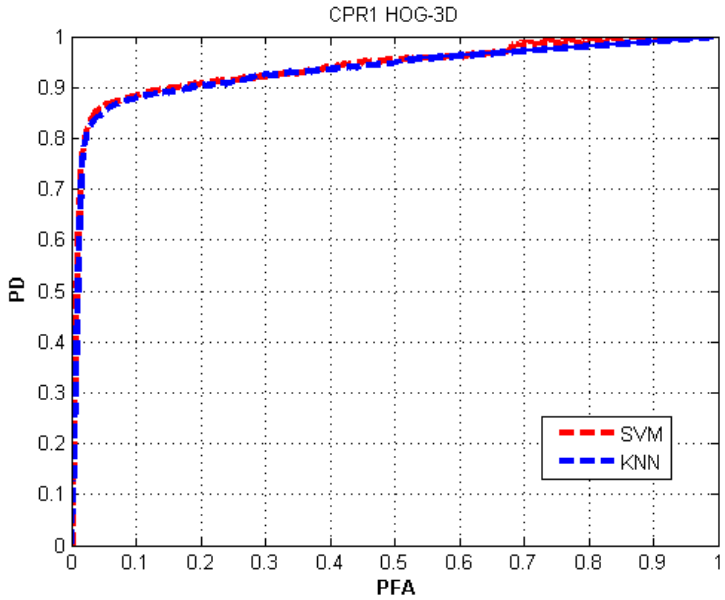
The ROC curves obtained using LBP-TOP features on SVM model and using KNN for the test of CPR1 is shown in figure (3.8a) and for the test of CPR2 is shown in figure (3.8b).

We analyze also the effectiveness of SIFT-3D features for CPR activity detection. We used for training the 3D bounding boxes only the volumes contained in the first 5 minutes of CPR1 and the first 5 minutes of CPR2. The same parameters used for SIFT-3D cited above and we used the same method to fix the number of clusters and this result in 25 clusters for each group and feature of length 50.

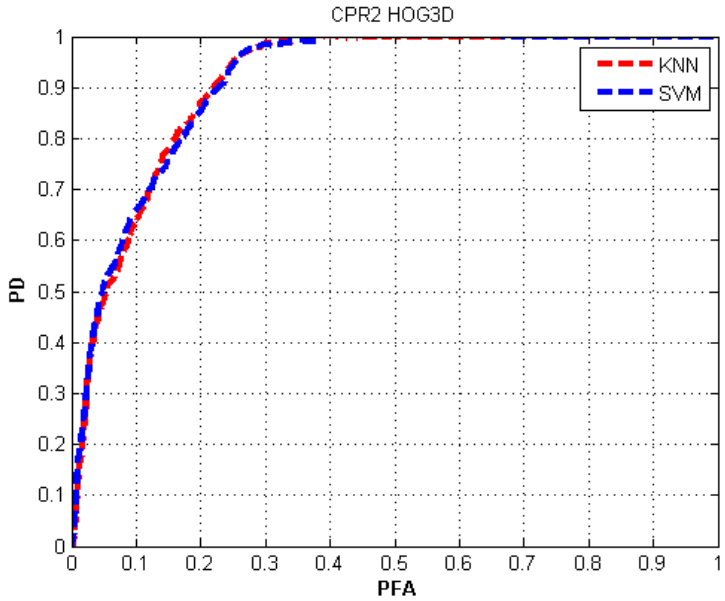
We train a third SVM classifier with the new set of features. Testing is also performed in a similar fashion as described in the previous experiment. The ROC curves obtained using SIFT-3D features on SVM model and using KNN for the test of CPR1 is shown in figure(3.9a) and for the test of CPR2 is shown in figure(3.9b).

For the running time, the time required to extract HOG3D from a bounding box of $55 \times 55 \times 18$ is 0.18 seconds while the LBP-TOP needs only 0.15 seconds to be extracted. Since we have used parallel programming to extract features from 8 bounding boxes at the same time, we need 0.36 seconds to describe the whole video volume of $130 \times 130 \times 18$ with HOG3D while only 0.3 seconds using LBP-TOP. For the SIFT-3D, we need 0.3 seconds to detect interest points and 0.22 seconds to describe each point with SIFT-3D which means a maximum of 0.74 seconds to describe each volume of $130 \times 130 \times 18$ with local feature SIFT-3D.

To test the last 15min of CPR1, it takes 19min45s with HOG3D, 16min45s with LBP-TOP and 37min48s with SIFT-3D which seems logical and acceptable time

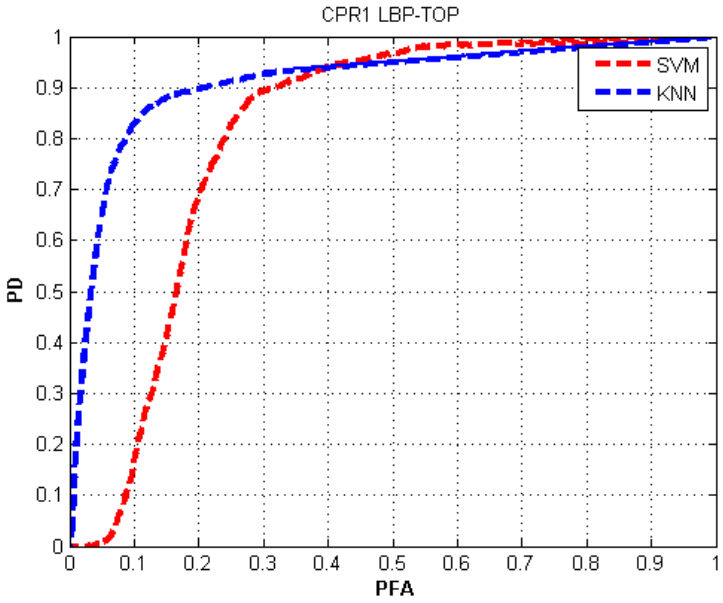


(a) CPR1

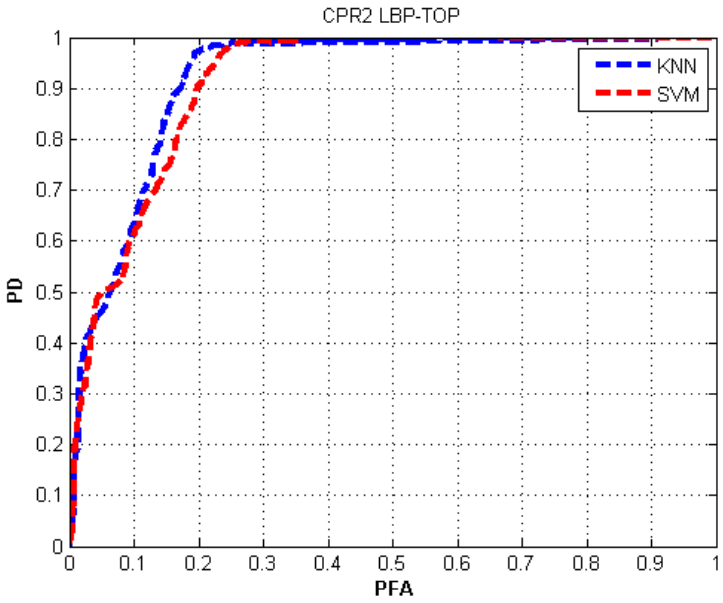


(b) CPR2

Figure 3.7: ROC generated from HOG3D features for CPR1 and CPR2.

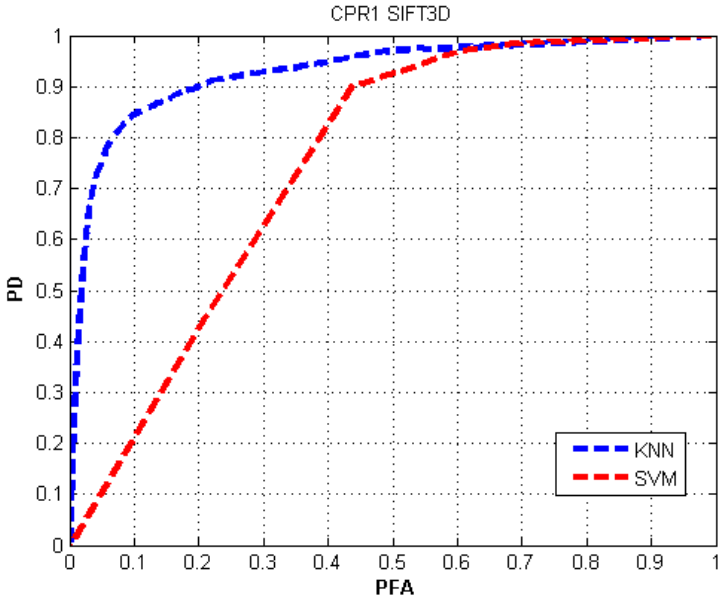


(a) CPR1

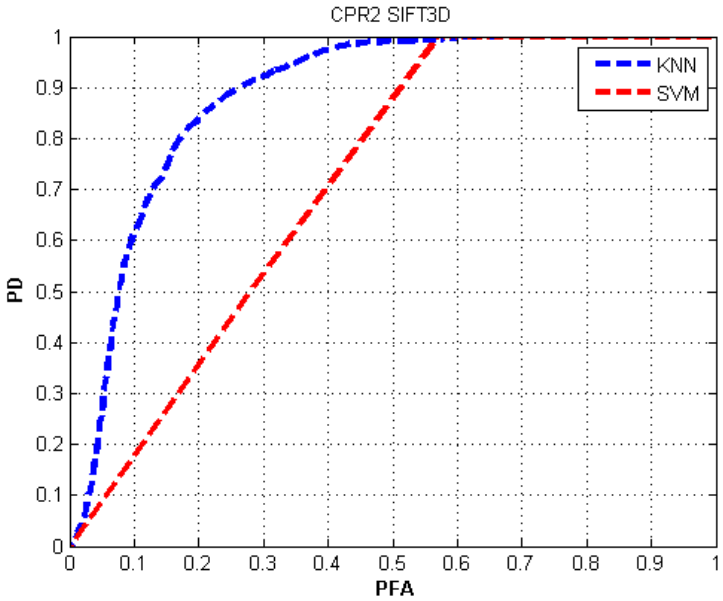


(b) CPR2

Figure 3.8: ROC generated from LBP-TOP features for CPR1 and CPR2.



(a) CPR1



(b) CPR2

Figure 3.9: ROC generated from SIFT-3D features for CPR1 and CPR2.

by the physician.

The accuracy of our system is influenced by the false detection of the region of interest which seems to decrease the performance while decreasing the computation time but on the other hand it is very efficient because when we tried to process the whole video without selection region of interest, we noticed that our system is greatly influenced by the presence of false positives as shown in figure (3.10) where only the positively classified grid windows are color-coded. Red color shows high confidence and blue color shows low confidence. This is mainly because those regions have either their shape structure or texture, similar to that of the CPR volume. For example, the texture of the clothes worn by people have similar edge orientation histograms as that of hands performing CPR (figure(3.10)).

Figure 3.10: Example of incorrect detection using HOG3D.

After analyzing the effectiveness of the spatio-temporal features by the confidences output of the classifiers, now we try to analyze the decision for the binary classification of our application and track the false alarm and miss detection with different features. For this task, we should fix a threshold γ for which the decision

will be 1 (CPR activity) if the confidence is higher than this threshold and 0 otherwise.

To fix the value of γ for every feature, we plot the histograms of confidences for the positive testing data and negative testing data in the same figure and find the intersection between them in order to have the lowest cost as shown for the example of LBP-TOP for CPR2 in figure (3.11) in which the value of γ is approximately 0.7.

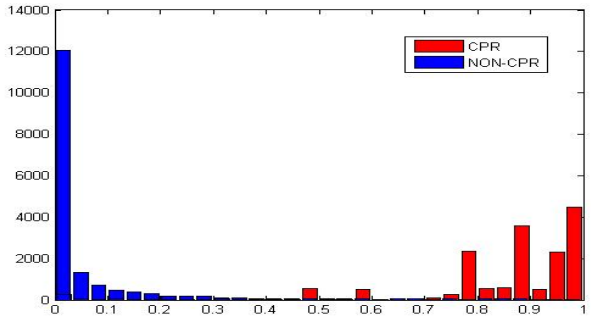
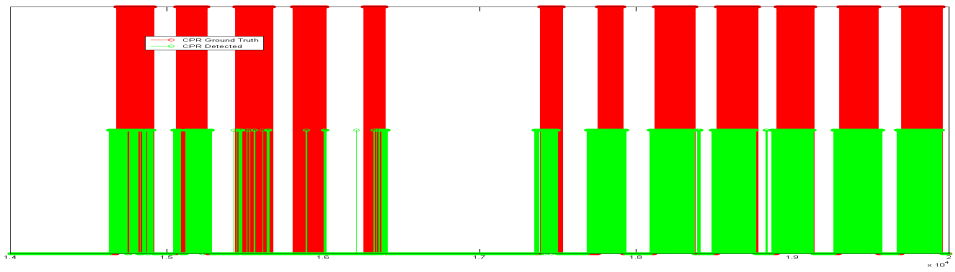
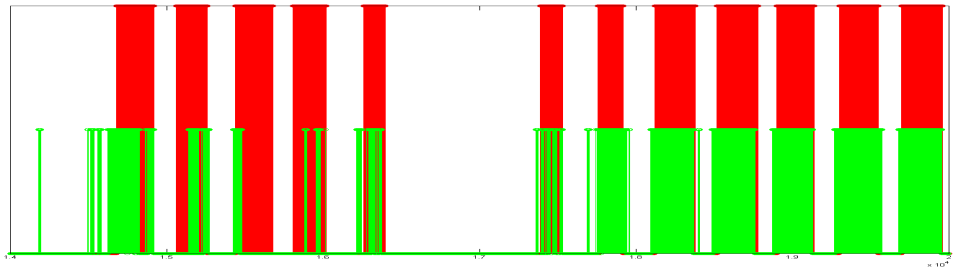


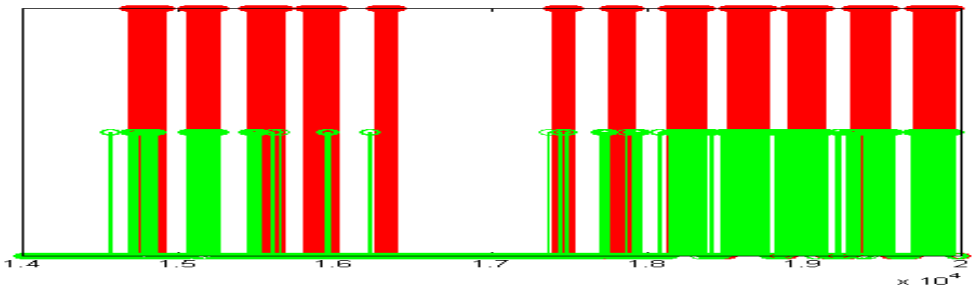
Figure 3.11: Histogram of confidences for the feature LBP-TOP applied for CPR2.



(a) LBP-TOP



(b) HOG3D



(c) SIFT-3D

Figure 3.12: Comparison of CPR activity detections using HOG3D, LBP-TOP and SIFT-3D in CPR2 video (frames:14000 → 20000).

After fixing γ for every feature, we plot the CPR detected by every feature and the ground truth to have a clearer idea about the difference between these features. Figure(3.12) illustrate the ground truth and CPR detected for three features for CPR2 (frames:14000 → 20000)).

By tracking false alarm and miss detection of every feature, we conclude that HOG3D feature mainly computes the edge distribution along the spatial and temporal dimension while the LBP-TOP is representative of the 3-D texture information. LBP-TOP gives good detection because it can detect the texture of the skin region around the hands performing the CPR while the HOG3d is unable to detect the texture in the region, it tries to detect the edge orientation representing the up and down cyclic movement of the CPR action. Hence some scene (where there is partial occlusion), the LBP-TOP gives a better detection (figure(3.14)) compared to the HOG3D (figure(3.15)). So, in scenes where the spatial orientation of the edges is more prominently visible, HOG3D is a better choice compared to LBP-TOP, whereas in scenes where there is partial occlusion, LBP-TOP would be a better choice. Therefore, we expect improvement by combining the two features to utilize the advantage of both the features for CPR activity detection.

Figure 3.13: Region of interest.

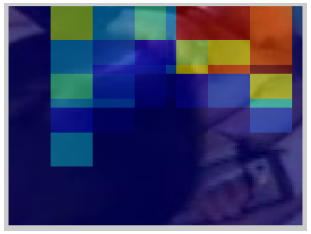


Figure 3.14: Detection results using LBP-TOP.

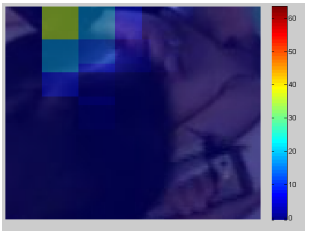


Figure 3.15: Detection results using HOG3D.

However, we have some results that were not expected for the SIFT-3D because it outperforms the other features in all previous works of human activity recognition which is not the case in our results, but after tracking and analyzing our algorithm, we find that the problem is not related to SIFT-3D descriptor but to the interest points extraction which is not always reliable and it detect sometimes many points which don't have noticeable movement. Figure(3.16) shows an exam-

ple of interest points detected in one frame which belongs to CPR activity. That's why we expect an improvement in the performance of the SIFT-3D to describe the movement of the face since all the interest points detected are in the region of the face and all can give good description for the movement.

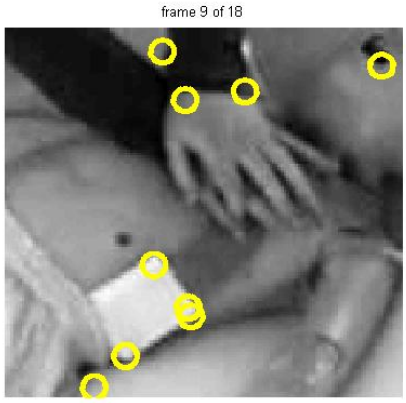


Figure 3.16: Example of interest points detected using Harris3D operator.

3.3.2.2 Face-tracking

The next step of our system of scene retrieval is to track the nearest face to the region of interest. To evaluate the performance of our method to detect the nearest face to the chest of mannequin, we build an automatic system to extract the nearest face every 50 frames from CPR1 and CPR2 and count manually the number of miss detection.

The number of frames tested for this task is 1090 for CPR1 and 570 for CPR2 and the number of miss detection is 77 for CPR1 and 51 for CPR2 which means 92,93% accuracy for CPR1 and 91% for CPR2.

To analyze the motion of the face, we use the same features proposed for analyzing

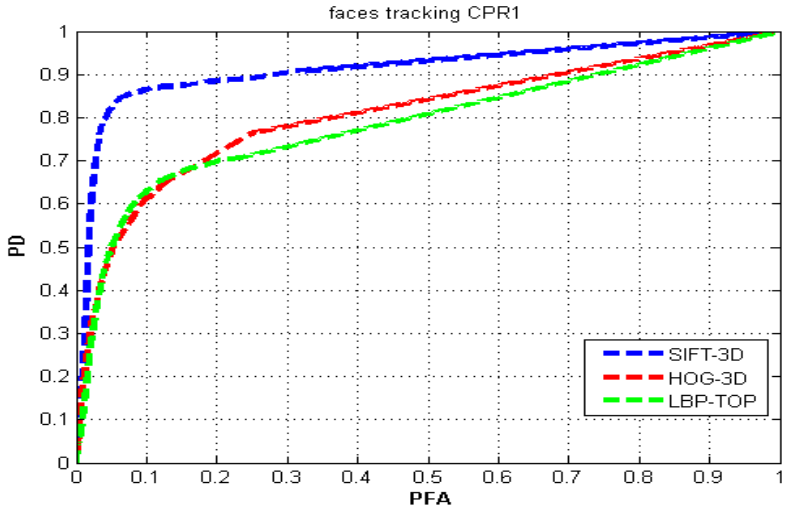
the hand motion (HOG3D;LBP-TOP;SIFT-3D). After visual inspection of few CPR scenes, the size of the bounding box is fixed to be 100×100 (approximately the size of the head for most CPR sequences) (figure(3.17)).

Figure 3.17: Example of 3D Bounding Box of face region.

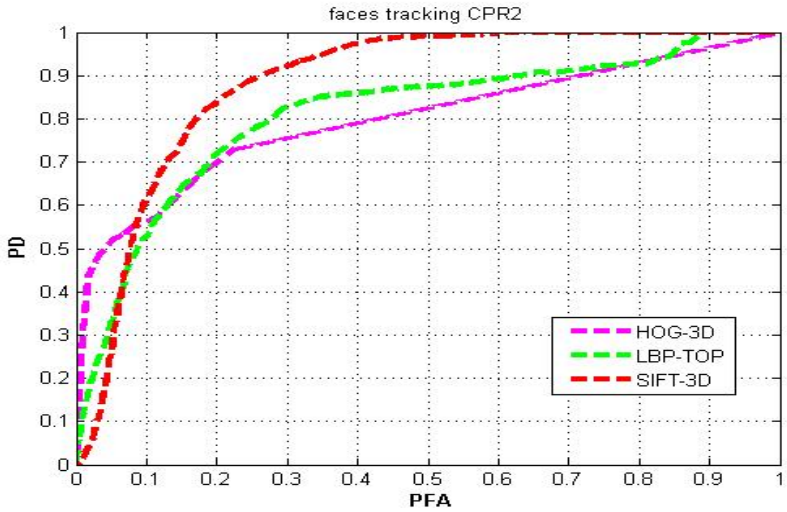
We use the same frame sequences of CPR scenes of the hand activity. The same as for hand-tracking, we use the first 5 minutes of CPR1 and CPR2 for training and the rest for testing.

For training, we select manually 3D bounding boxes of $100 \times 100 \times 18$ of the face of person performing CPR activity to build our positive dataset for training. For non-CPR sequences, we extract blocks at random spatial and temporal locations that do not overlap with blocks of CPR activity regions and we add some blocks selected manually of faces of persons who don't perform CPR activity to build our negative dataset.

We use the same parameters stated above for the HOG3D and LBP-TOP. For SIFT-3D, we use the same parameters for describing the detected interest points and we use the proposed method to fix the number of clusters to finally fix it to be 11 clusters. Finally, we train new three linear SVM classifier with the new set of features. We noticed that SVM outperforms the KNN for three features so we show only the results using SVM classifiers. For The ROC curves obtained using



(a) CPR1



(b) CPR2

Figure 3.18: ROC generated from HOG3D,LBP-TOP and SIFT-3D features for face-tracking for CPR1 and CPR2.

HOG3D, LBP-TOP and SIFT-3D features are shown in figure(3.18a) for CPR1 and figure(3.18b) for CPR2.

As it was expected, SIFT-3D outperforms the other features for face tracking since all the interest points detected are all part of the face region where there is

significant motion and help us to be useful to describe the movement.

3.3.2.3 Decision level fusion

We provided results of using HOG3D, LBP-TOP and SIFT-3D features for hand tracking and face tracking for CPR activity detection. The use of features from different modalities possibly supplements existing feature constraints and their fusion helps in achieving higher performance. Fusion can be achieved with feature level fusion or decision level fusion techniques. Feature level fusion can achieve better performance, but if the individual features are high- dimensional, the fusion results in even higher dimensional features which in turn adversely affects the performance. The primary objective of fusion is to achieve a higher recognition rate with reduced processing time. Therefore, we propose to integrate the confidences obtained using HOG3D, LBP-TOP and SIFT-3D features of hand-tracking and face-tracking to achieve better accuracy of detection.

Many found results motivated us for the fusion. For example, for hand tracking, HOG3D is perfect for CPR1 and not very good for CPR2 and LBP-TOP is perfect for CPR2 and not very good for CPR1 so if we choose one of them it will be good for one and not good for the other.

Also, it is well known that HOG3D is great at capturing edges/corners spatio-temporal shape of the activity while LBP-TOP captures the patterns/spatio-temporal texture of the activity. Ultimately HOG3D and LBP-TOP captures different kinds of information, which make them complimentary to each other. In addition, SIFT-3D can sometimes detect CPR scenes that are not detected with other two features as can be observed from figure(3.12).

Figure(3.19) shows scatter plot of the confidences of HOG3D and LBPTOP for

Hand-tracking for CPR2 where red circles represent Non-CPR sequences while blue circles represent CPR sequences. This can illustrate how for a fixed threshold for each of those confidences, the two features don't detect the same CPR scenes. Therefore, we expect to see an improvement in accuracy by the fusion of all these

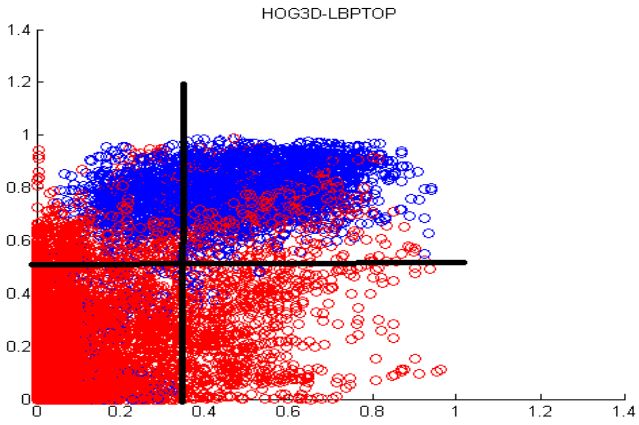


Figure 3.19: Scatter plot of confidences of HOG3D and LBPTOP for Hand-tracking for CPR2 (x-axis: HOG3D, y-axis: LBP-TOP).

features.

First step of our fusion approach is to use the ANN with 0 hidden layer in order to have weights that have meaning for us and we can have an idea about the information that are important (high weights) and what are those which don't affect the output (low weights). So, we use ANN with 12 inputs and 1 output.

Based on the obtained weights we decided to keep only the important information which are HOG3D (KNN and SVM) and LBP-TOP (KNN and SVM) of hand tracking and SIFT-3D (SVM) of face tracking and to fuse the confidences using 2 methods:

1. First method: fuse confidences obtained from HOG3D (KNN and SVM) and

LBP-TOP (KNN and SVM) of hand tracking which means we fuse 4 confidences since the SIFT-3D (SVM) of face tracking takes long running time and see if we can obtain good results without the need of having information of this feature.

2. Second method: fuse confidences obtained from HOG3D (KNN and SVM), LBP-TOP (KNN and SVM) of hand tracking and SIFT-3D (SVM) of face tracking which means we fuse 5 confidences.

For every method, we try 4 fusion techniques described in the second chapter:

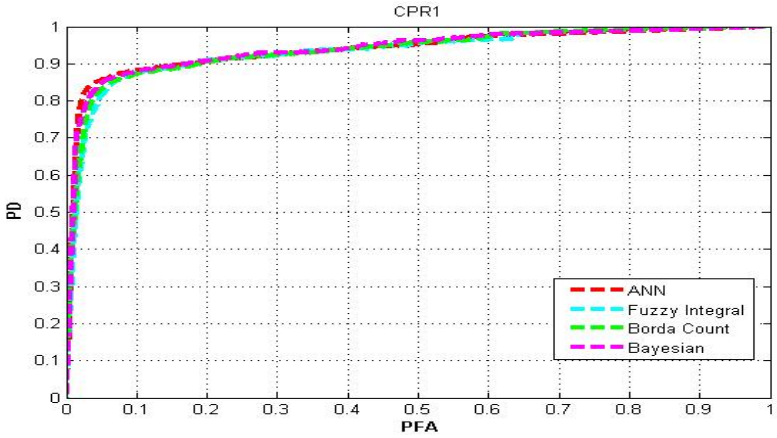
- Fuzzy integral: we have used Sugeno measure. For training, we initialize the fuzzy density values and we find those that minimize the difference between true outputs and the outputs and the value of λ is found by solving the Eq.(18).

- Borda count: we use the weighted approach and assign the values of the weights given from the ANN and then normalize them.

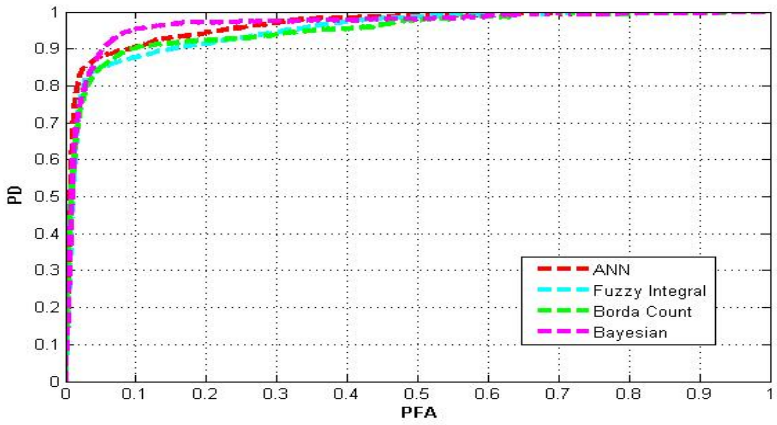
- ANN: we used ANN with 2 hidden layers and this value is determined experimentally since it gives the best results compared to many other number of hidden layers.

- Bayesian fusion: we first cluster the training data using the EM into M components. Then, the posterior probability can be computed by generalizing Bayes rule in Eq. (16) to assign a test point into different component or class. So, there is no need for selecting any parameter for Bayesian fusion.

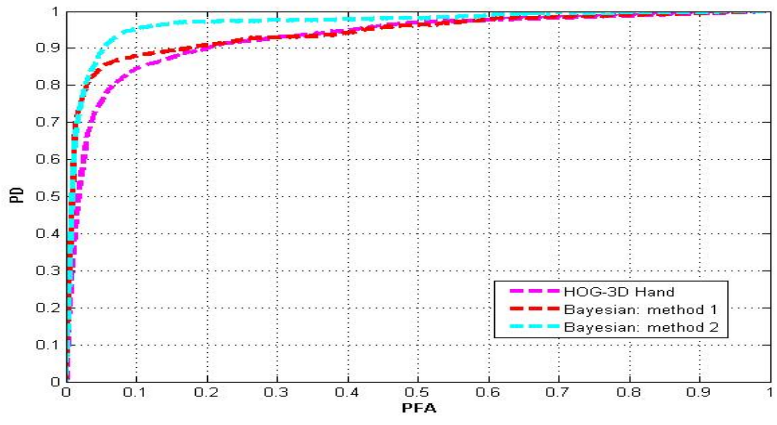
The results of two fusion methods are shown in figure(3.20) for CPR1 and in figure(3.21) for CPR2:



(a) ROC generated using first method of fusion

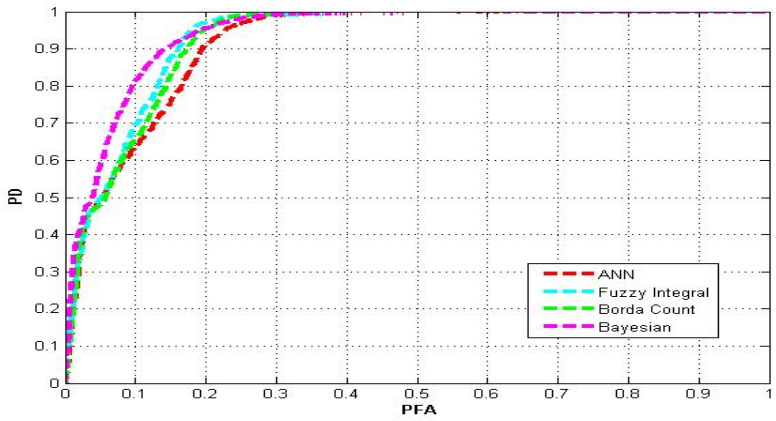


(b) ROC generated using second method of fusion

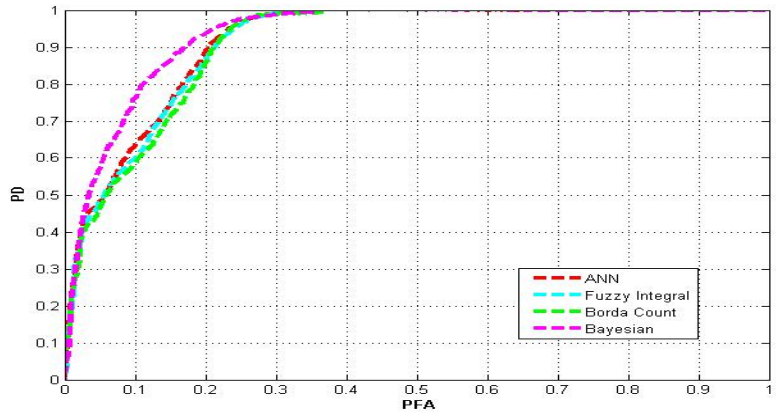


(c) Comparaison of best fusion methods

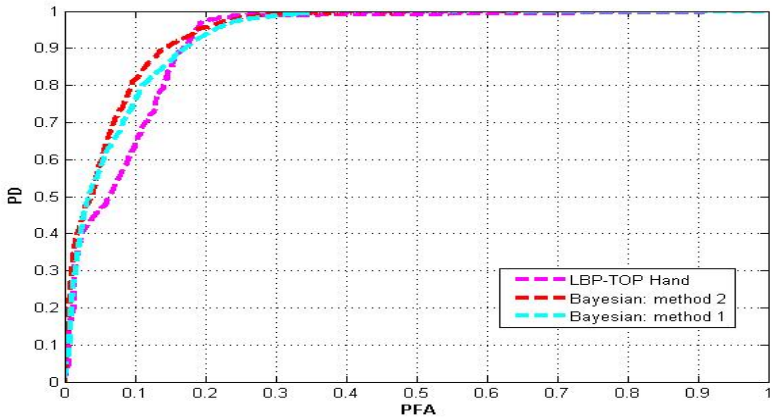
Figure 3.20: Comparaison of different fusion methods for CPR1.



(a) ROC generated using first method of fusion



(b) ROC generated using second method of fusion



(c) Comparaison of best fusion methods

Figure 3.21: Comparaison of different fusion methods for CPR2.

As we can notice from these figures, Bayesian fusion using the second method which fuse information from hand tracking and face tracking outperform all the results using every feature individually and all the other fusion approaches. Bayesian fusion using first method is also very efficient and can be used for our application without the need of increasing the running by extracting SFT-3D from the region of the head. Thus, we conclude to extract features and apply Bayesian fusion with one of the two methods. So, we are here in controversy between performing excellent results and reducing computational time. Thus, we let the physician decide which method to use and this depends on his available time and his need for excellent results. In the next section, a graphical user interface (GUI) for CPR scene retrieval that is explained. This GUI which allows the user to choose any method of fusion desired is provided to the physician.

3.4 Graphical user interface for CPR video scene Retrieval

In order to provide the user an easy way to retrieve CPR scenes from a given video, we build two Graphical User Interfaces (GUI). The GUI allows the use of icons or other visual indicators to interact with electronic devices, rather than using only text via the command line.

The first one (figure(3.22)) used in order to extract features only one time and save them to be used every time the user want to retrieve CPR scenes from a given video. The user can choose to extract feature from the region of interest or from the nearest face to this region. He can also change the parameters for every feature or let the default parameters that we have used in our experiments and that we find the best.

The second GUI (figure(3.23)) is used for two tasks: training and testing.

For the training part, the user can choose any combination of features of hand-tracking and face-tracking with the possibility to change the parameters of every feature and choosing the method of decision level fusion. After choosing the features and method of fusion if more than one feature in used, he can fix the percentage of CPR1 and the percentage of CPR2 used for training. The remaining of these videos is used to have an idea about the performance of this selected combination since we have the ground truth of these videos. the next step is to click in the button "Train classifier" so that we build SVM classifier model with linear kernel for every feature which can be already saved using the first GUI and if not it will be automatically extracted. Then, the ROCs of every feature and the fusion of all features for the rest of the videos are showed. Figure(3.24) illustrate the ROCs

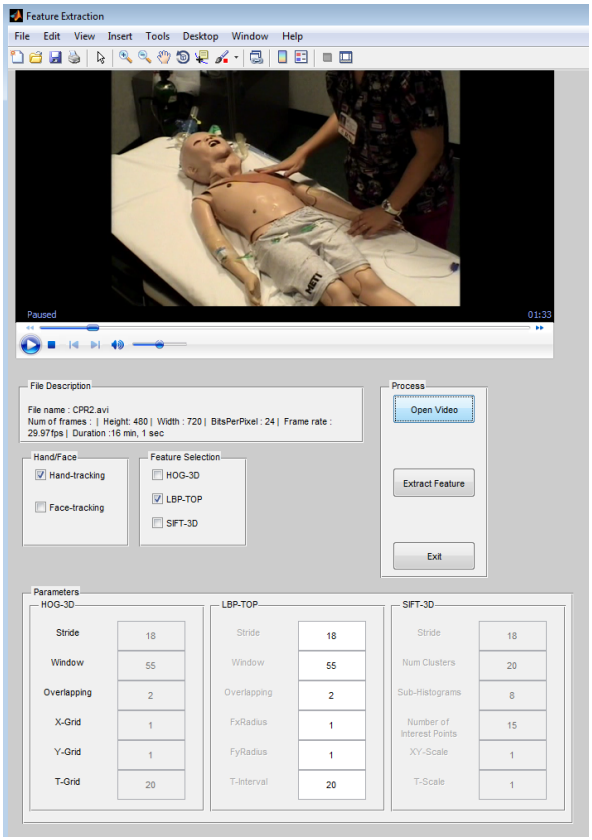


Figure 3.22: GUI of the feature extraction.

of an example of Bayesian fusion of HOG3D and LBP-TOP of Hand-tracking and LBP-TOP of face-tracking.

Based on the ROCs, we can choose either one feature individually or the fusion of all features for the final step of extracting CPR sequences. Then, histograms of confidences of CPR scenes and non-CPR scenes are presented and now the user can variate the confidence threshold used for the decision.

Finally, we can click "Show CPR scenes" and we have with three different colors the first frame of the CPR scenes. The first color represents CPR scenes that are detected correctly with the selected method, the second color repre-

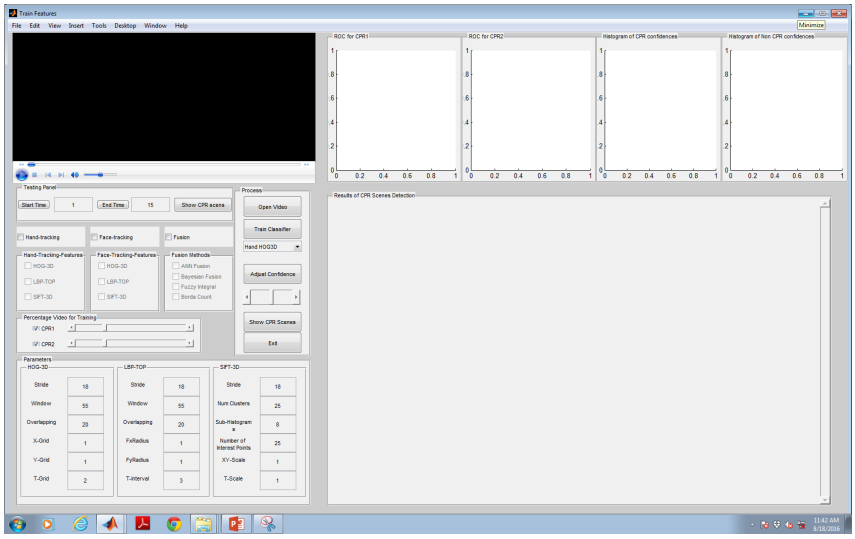
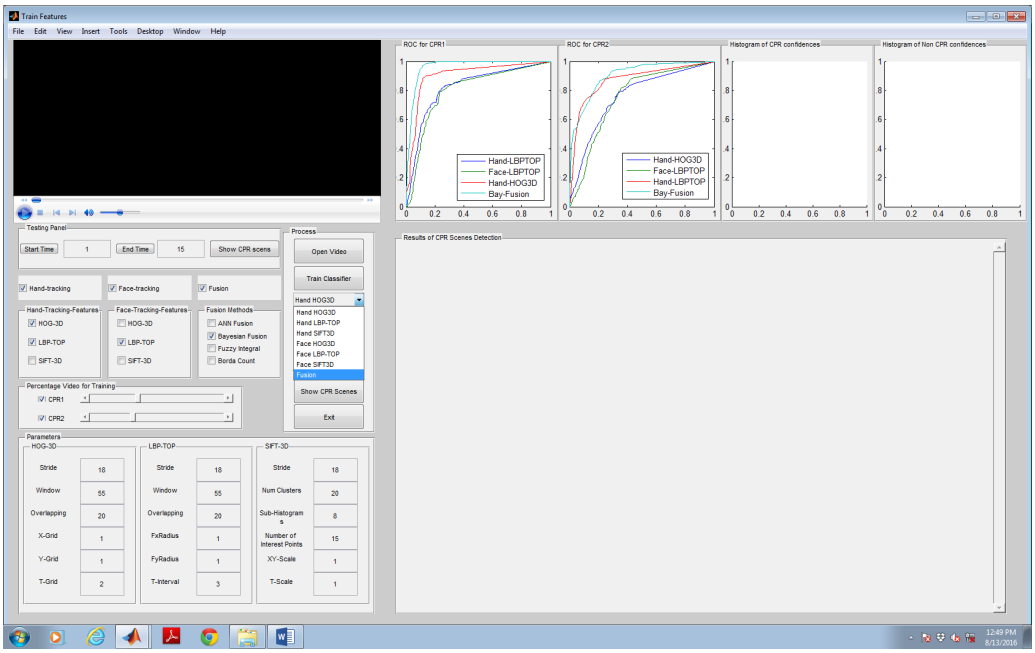


Figure 3.23: GUI of the proposed CPR scene retrieval prototype.

sents the CPR scenes that are not detected and the third represents the false detection(figure(3.25)). With clicking in any image that represent the first frame, the whole sequence is played(figure(3.26)).

For the testing part, we should click "open video" to input the query video. The user will be able to watch a preview of the selected video. The selected video will be processed to identify the CPR activity scenes using the model selected. The user will also have the capability to enter the start and end minute of any scene, to retrieve the particular scene. By clicking on "show CPR scenes" in testing panel, the final results of the CPR sequences retrieved using the selected model are shown (figure(3.27)). The physician can either train classifiers with any combination he wants or test directly using saving model of Bayesian fusion of HOG3D, LBP-TOP of Hand tracking and SIFT-3D of face tracking using the parameters used in our experiments.



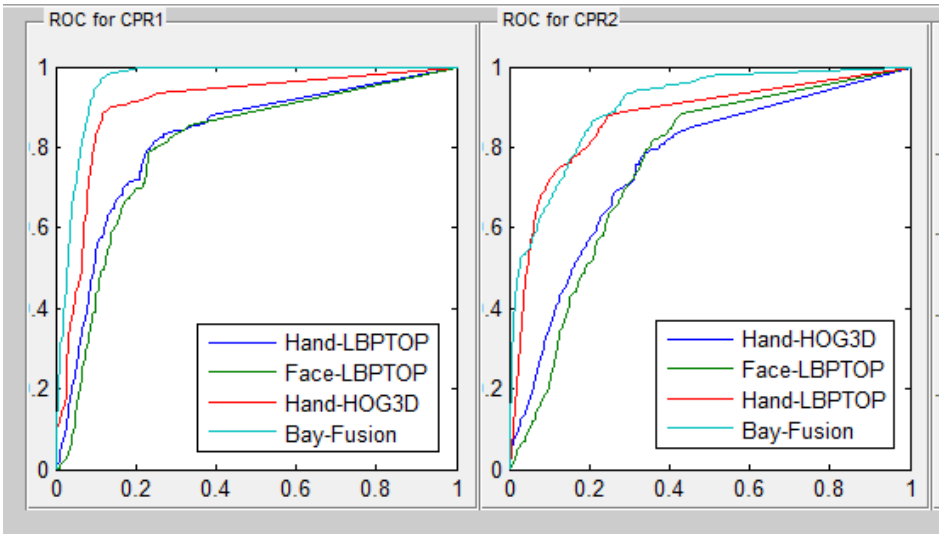


Figure 3.24: ROCs generated for an example chosen randomly.

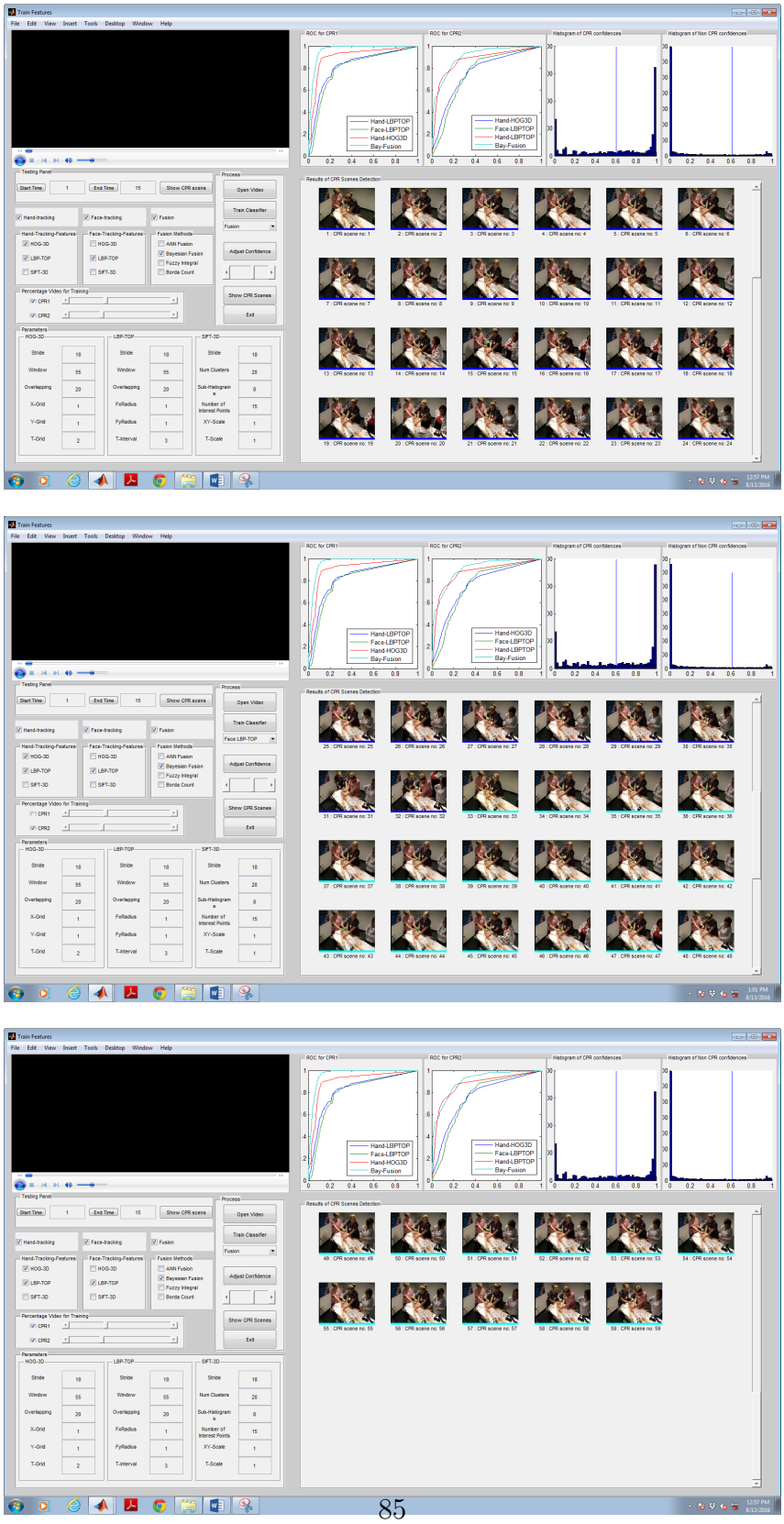


Figure 3.25: First frames of CPR scenes.

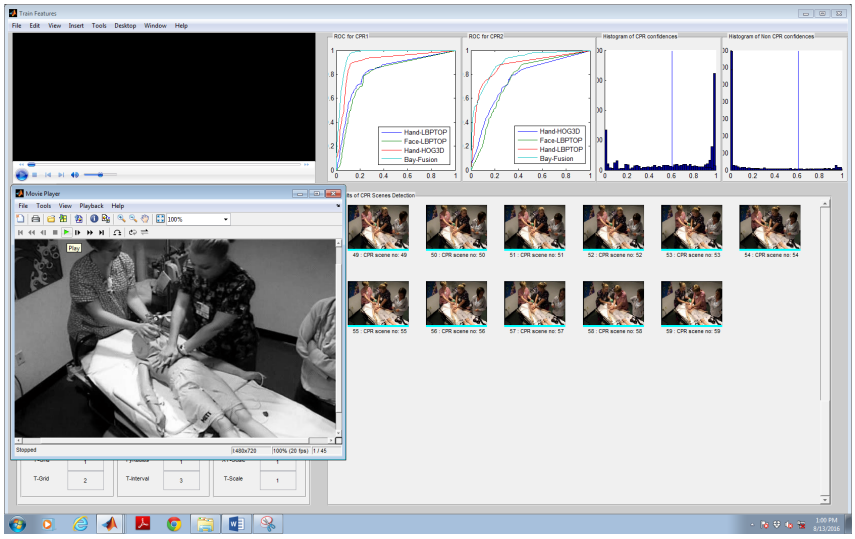


Figure 3.26: Example of a displayed video.

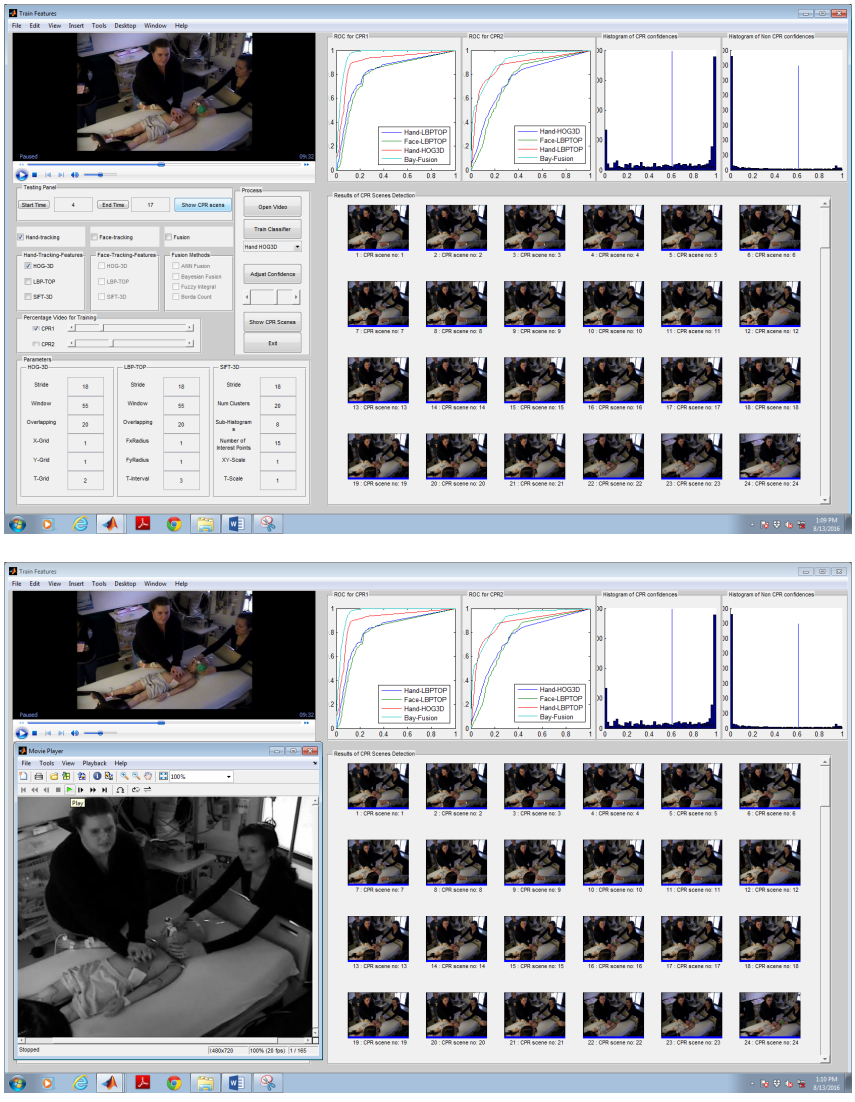


Figure 3.27: Example of testing new video.

4 Conclusion

We have proposed an approach for retrieving scenes from medical simulation videos that contain localized CPR activities. This method of scene retrieval or action retrieval eliminates the need for video shot detection and frame segmentation phases, which are usually the initial steps of most scene retrieval algorithms. The proposed method is straight forward and easier compared to the other methods of activity detection in medical simulations videos. It achieves a better classification accuracy as compared to the HMM classifier in [21].

Future work can includes the integration of detection of specific objects like the breathing bag, defibrillator etc. in the scene, and track its movements and also additional features that the supervising physician has requested include a face recognition component to identify the person performing CPR, and analysis of the audio during the CPR session to identify chaotic environments.

References

[1] *T.Q. Pham and L.J. van Vliet, "Separable bilateral filtering for fast video preprocess- ing," in Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, July 2005, pp. 4 pp.–.*

[2] *Gu Jian-liu, "A processing method of atm monitoring video image under weak light environment," in Knowledge Acquisition and Modeling (KAM), 2011 Fourth Interna- tional Symposium on, Oct 2011, pp. 270–273.*

[3] *Liu Huayong, "Content-based tv sports video retrieval based on audio-visual features and text information," in Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on, Sept 2004, pp. 481–484.*

[4] *Xingquan Zhu, Xindong Wu, A.K. Elmagarmid, Zhe Feng, and Lide Wu, "Video data mining: semantic indexing and event detection from the association perspective," Knowledge and Data Engineering, IEEE Transactions on, vol. 17, no. 5, pp. 665–677, May 2005.*

[5] *Zhao Yaqin, Zheng Jiaqiang, and Zhou Hongping, "News video clip retrieval based on topic caption text and audio information," in Intelligent Systems, 2009. GCIS '09. WRI Global Congress on, May 2009, vol. 4, pp. 477–481.*

[6] *Yongliang Xiao, "An effective video shot boundary detection method based on su- pervised learning," in Advanced Computer Control (ICACC), 2010 2nd International Conference on, March 2010, vol. 4, pp. 371–374.*

[7] *M. Rashid, S.A.R. Abu-Bakar, M. Mokji, and A. Abdu, "Human action concentric video retrieval system using features weight updating method as relevance feedback," in Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on, Nov 2012, pp. 366–370.*

[8] *Tianzhu Zhang, Changsheng Xu, Guangyu Zhu, Si Liu, and Hanqing Lu, "A generic framework for video annotation via semi-supervised learning," Multimedia, IEEE Transactions on, vol. 14, no. 4, pp. 1206–1219, Aug 2012.*

[9] *Yan Song, G.-J. Qi, Xian-Sheng Hua, Li-Rong Dai, and Ren-Hua Wang, "Video anno- tation by active learning and semi-supervised ensembling," in Multimedia and Expo, 2006 IEEE International Conference on, July 2006, pp. 933–936.*

[10] *Sangmin Oh and A. Hoogs, "Unsupervised learning of activities in video using scene context," in Pattern Recognition (ICPR), 2010 20th International Conference on, Aug 2010, pp. 3579–3582.*

[11] *M.M. Ben Ismail, O. Bchir, and A.Z. Emam, "Endoscopy video summarization based on unsupervised learning and feature discrimination," in Visual Communications and Image Processing (VCIP), 2013, Nov 2013, pp. 1–6.*

[12] *Zan Gao, M. Detyniecki, Ming yu Chen, Wen Wu, A.G. Hauptmann, and H.D. Wactlar, "Towards automated assistance for operating home medical devices," in Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, Aug 2010, pp. 2141–2146.*

[13] *F. Hijaz, N. Afzal, T. Ahmad, and O. Hasan, "Survey of fall detection and daily activity monitoring techniques," in Information and Emerging Technologies (ICIET), 2010 International Conference on, June 2010, pp. 1–6.*

[14] *Qiang Liu, R.J. Sclabassi, and Mingui Sun, "A new change detection method and its application to epilepsy monitoring video," in Bioengineering Conference, 2004. Proceedings of the IEEE 30th Annual Northeast, April 2004, pp. 59–60.*

[15] *I. Chakraborty, A. Elgammal, and R.S. Burd, "Video based activity recognition in trauma resuscitation," in Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, April 2013, pp. 1–8.*

[16] *B. Soran, Jenq-Neng Hwang, Su-In Lee, and L. Shapiro, "Tremor detection using motion filtering and svm," in Pattern Recognition (ICPR), 2012 21st International Conference on, Nov 2012, pp. 178–181.*

[17] *Z A Khan and W Sohn. Real Time Human Activity Recognition System based on Radon Transform. IJCA Special Issue on Artificial Intelligence Techniques - Novel Approaches and Practical Applications (4):7–13, 2011.*

[18] *Satkin, S., and Hebert, M. (2010). "Modeling the temporal extent of actions," in Proc. European Conference on Computer Vision (Heraklion), 536–548.*

[19] *Wu, C., Zhang, J., Savarese, S., and Saxena, A. (2015). "Watch-n-patch: unsupervised understanding of actions and relations," in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Boston, MA), 4362–4370.*

[20] *Alexander Klaser, Marcin Marszalek, and Cordelia Schmid, "A spatio-temporal descriptor based on 3d-gradients," in Proceedings of the British Machine Vision Conference 2008, Leeds, September 2008, 2008, pp. 1–10*

[21] *Surangkana Rawungyot, "Identification, indexing, and retrieval of cardiopulmonary resuscitation (CPR) video scenes of simulated medical crisis.", Ph.D. thesis, University of Louisville, 2014.*

[22] *I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, June 2008, pp. 1–8.*

[23] *N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, June 2005, vol. 1, pp. 886–893 vol. 1.*

[24] *Paul Viola and Michael Jones, "Robust real-time object detection," in International Journal of Computer Vision, 2001.*

[25] *J. Fehr and H. Burkhardt, "3d rotation invariant local binary patterns," in Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, Dec 2008, pp. 1–4.*

[26] *M. Topi, O. Timo, P. Matti, and S. Maricor, "Robust texture classification by subsets of local binary patterns," in Pattern Recognition, 2000. Proceedings. 15th International Conference on, 2000, vol. 3, pp. 935–938 vol.3.*

[27] *Florian Baumann, Jie Liao, Arne Ehlers, and Bodo Rosenhahn, "Motion binary patterns for action recognition," in 3rd International Conference on Pattern Recognition Applications and Methods, Mar. 2014.*

[28] *I. Laptev and T. Lindeberg, "Space-time interest points," in Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, Oct 2003, pp. 432–439 vol.1.*

[29] *Kellokumpu, Vili, Guoying Zhao, and Matti Pietikäinen. "Human Activity Recognition Using a Dynamic Texture Based Method."*

[30] *P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In ICME , 2007.*

[31] *Lowe, David G. (1999). "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision. pp. 1150–1157.*

[32] *Sivic, Josef (April 2009). "Efficient visual search of videos cast as text retrieval". IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31, NO. 4. IEEE. pp. 591–605.*

[33] *C. Silva, T. Bouwmans, C. Frelicot, "An eXtended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos", VISAPP 2015, Berlin, Germany, March 2015.*

[34] *Laptev, Ivan , Lindeberg, Tony (2004). "Local descriptors for spatio-temporal recognition". ECCV'04 Workshop on Spatial Coherence for Visual Motion*

Analysis, Springer Lecture Notes in Computer Science, Volume 3667. pp. 91–103.

[35] *Foroughi H., Naseri A., Saberi A., Yazdi H.S. An eigenspace-based approach for human fall detection using integrated time motion image and neural network. Proceedings of the 2008 9th IEEE International Conference on Signal Processing, ICSP 2008; Beijing, China. 26–29 October 2008; pp. 1499–1503.*

[36] *Jain A.K., Duin R.P.W., Mao J. Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Mach. Intell. 2000;22:4–37.*

[37] *Foroughi H., Naseri A., Saberi A., Yazdi H.S. An eigenspace-based approach for human fall detection using integrated time motion image and neural network. Proceedings of the 2008 9th IEEE International Conference on Signal Processing, ICSP 2008; Beijing, China. 26–29 October 2008; pp. 1499–1503.*

[38] *Bengalur M.D. Human activity recognition using body pose features and support vector machine. Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI); Mysore, India. 22–25*

[39] *Kodagoda S., Piyathilaka J. Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features. Proceedings of the 2013 IEEE 8th International Conference on Industrial Electronics and Applications; Melbourne, Australia. 19–21 June 2013; pp. 567–572.*

[40] *Trabelsi D., Mohammed S., Chamroukhi E., Oukhellou L., Amirat Y. An unsupervised approach for automatic activity recognition based on hidden Markov model regression. IEEE Trans. Autom. Sci. Eng. 2013;10:829–835.*

bibitemq966 *Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik, A training algorithm for optimal margin classifiers, in Proceedings of the Fifth Annual Workshop on Computational Learning Theory, New York, NY, USA, 1992, COLT 92, pp. 144–152,ACM.*

[41] *Limin Wang, Yu Qiao, and Xiaoou Tang, "Motionlets: Mid-level 3d parts for human motion recognition," in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, June 2013, pp. 2674–2681.*

[42] *P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, June 2008, pp. 1–8.*

[43] *Cover, T.M., Hart, P.E., 1967. Nearest neighbor pattern classification. IEEE Trans. Information Theory IT-13 (1), 21–27*

[44] *Y. Bi, D. Bell, H. Wang, G. Guo, and J. Guan. Combining multiple classifiers using dempster's rule for text categorization. Applied Artificial Intelligence, 21(3):211–239, 2007.*

[45] *F. Huenupan, N. B. Yoma, C. Molina, and C. Garreton. Confidence based multiple classifier fusion in speaker verification. Pattern Recognition Letters, 29(7):957–966, 2008.*

[46] *K. Sirlantzis, S. Hoque, and M. C. Fairhurst. Trainable multiple classifier schemes for handwritten character recognition. In Proceedings of the 3rd International Workshop on Multiple Classifier Systems, pages 319–322, Cagliari,Italy, 2002.*

[47] *L. Xu and S. ichi Amari. Encyclopedia of Artificial Intelligence, volume 3, chapter Combining Classifiers and Learning Mixture-of-Experts, pages 318–326. IGI Global (IGI) publishing company, 2009.*

[48] *M. Ceccarelli and A. Petrosino. Multi-feature adaptive classifiers for sar image segmentation. Neurocomputing, 14:345–363, 1977.*

[49] *Y. S. Huang and S. C. Y. A method of combining multiple classifiers-a neural network approach. Proc. 12th Int'l Conf. Pattern Recognition and Computer Vision, pages 473–475, 1994.*

[50] *S.-B. Cho and J. H. Kim. Combining multiple neural networks by fuzzy integral for robust classification. In IEEE Transactions on Systems, Man and Cybernetics, volume 25 of 2, pages 380–384, 1995.*

[51] *M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. Neural Computation, 6:181–214, 1994.*

[52] *T. K. Moon, "The expectation-maximization algorithm," IEEE Signal Processing Magazine, vol. 13, no. 6, pp. 47–60, 1996.*

[53] *H. Tahani and J. M. Keller. Information fusion in computer vision using the fuzzy integral. IEEE Transactions on Systems, Man and Cybernetics, 20(3):733–741, 1990.*

[54] *G. Choquet. Theory of capacities. Annales de lŠInstitut Fourier, 5:131–295, 1955.*

[55] *M. Grabisch. A new algorithm for identifying fuzzy measures and its application to pattern recognition. volume 1, pages 145–150, 1995.*

[56] *S. Auephanwiriyakul, J. M. Keller, and P. D. Gader. Generalized choquet fuzzy integral fusion. Information Fusion, 3(1):69 – 85, 2002.*

[57] *K. Leszczynski, P. Penczek, and W. Grochulski. Sugeno's fuzzy measure and fuzzy clustering. Fuzzy Sets and Systems, 15(2):147–158, March 1985.*

[58] *G. Klir and Z. Wang. Fuzzy Measure Theory. Plenum, New York, 1992.*

[59] *T. Murofushi and M. Sugeno. An interpretation of fuzzy measures and the choquet integral as an integral with respect to a fuzzy measure. Fuzzy Sets Syst., 29(2):201–227, 1989.*

[60] *T. Murofushi. A technique for reading fuzzy measures (i): The shapley value with respect to a fuzzy measure. In in 2nd Fuzzy Workshop,page 39-48 Nagaoka, Japan, Oct 1992. in Japanese.*

[61] *TT. Murofushi and S. Soneda. Techniques for reading fuzzy measures (iii): Sapporo, Interaction index. In in 9th Fuzzy Syst. Symp.,page 693-696,Japan, May 1993. in Japanese.*

[62] *M. Grabisch. A graphical interpretation of the choquet integral. IEEE Transactions on Fuzzy Systems, 8:627–631, Oct 2000.*

[63] *J. C. de Borda. Mémoire sur les élections au scrutin. Histoire de l'Académie Royale des Sciences, Paris, 1781.*

[64] *Yicong Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, June 2013, pp. 2642–2649.*

[65] *P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001. IEEE, 2001, vol. 1, pp. I–511.*

[66] *Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," Journal- Japanese Society For Artificial Intelligence, vol. 14, no. 771-780, pp. 1612, 1999.*

bibitema67 *I. Laptev and T. Lindeberg, "Space-time interest points," in Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, Oct 2003, pp. 432–439 vol.1.*

[67] *P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, Oct 2005, pp. 65–72.*

[68] *Xingxing Wang, LiMin Wang, and Yu Qiao, "A comparative study of encoding, pooling and normalization methods for action recognition," in Proceedings of the 11th Asian Conference on Computer Vision - Volume Part III, Berlin, Heidelberg, 2013, ACCV'12, pp. 572–585, Springer-Verlag.*

[69] *Haoyu Ren, Cher-Keng Heng, Wei Zheng, Luhong Liang, and Xilin Chen, "Fast object detection using boosted cooccurrence histograms of oriented gradients," in Image Processing (ICIP), 2010 17th IEEE International Conference on, Sept 2010, pp. 2705– 2708*