Master Erasmus Mundus in
Color in Informatics and Media Technology (CIMET)

Adult video content detection using Machine Learning Techniques

Master Thesis Report

Presented by

Victor Manuel Torres Ochoa

and defended at

Gjøvik University College

Academic Supervisor(s): Sule Yildirim Yayilgan
Faouzi Alaya Cheikh

Jury Committee: Prof. Marc Sebban
Prof. Damien Muselet

Adult video content detection using Machine Learning Techniques

Victor Manuel Torres Ochoa

15/07/2012

Abstract

Automatic adult video detection is a problem of interest for many organizations around the globe aiming to restrict the availability of potentially harmful material for young audiences. Being most of the existing techniques a mere extension of the image categorization problem. In the present work we employ video genre classification techniques applied specifically for adult content detection by considering cinematic principles. Shot structure and camera motion in the temporal domain are used as the main features while skin detection and color histograms representation in the spatial domain are utilized as complementary features. Using a data set of more than 7 hours of video, our experiments comparing two different SVM algorithms achieve a high accuracy of **94.44%**.

Keywords: Porn detection, Video categorization, Machine Learning

Preface

My more sincere gratitude to all the people whose intervention made this possible. Firstly I'd like to thank my supervisors, Prof. Sule Yildirim Yayilgan and Prof. Faouzi Alaya Cheikh, for their guidance, support, encouragement, and trust. To Ali Shariq Imran for his participation and advice. Prof. Simon McCallum and Jayson Mackie for their help and concern. Helene Goodsir for constantly looking after us. And last but no least, to all of my friends and family who made this experience valuable and were always there to give a word. Thank you all for being along the way.

Contents

List of Figures

List of Tables

1 Introduction

In the last few decades the use of Internet has grown exponentially posing a series of problems that demand new methods to be effectively tackled. One of the undesirable effects is the proliferation of media that can be potentially harmful and the ease of its distribution and access by virtually anyone. The traditional methods of site blocking based on the so called black lists are losing effectiveness and the scientific community has centered efforts in developing new techniques that classify the huge data collections available in the network based on the content of the file itself rather than the meta data (such as filenames). Pornography is quite spread all over the network and can be easily found and accessed even by novel users. There is a number of concerned organizations all over the world working on preventing children to access this potentially damaging media such as the *Medietlisynet: The Norwegian Media Authority*[1] and the *European Commision* hosting the *Safer Internet Programme*[2]. In parallel, the computer vision field has made enormous progress to provide many applications such as automatic video surveillance, action detection, robot guidance, or video genre classification. All of these share the common task of interpret the semantic meaning of the scenes in a certain degree. The prime goal of multimedia content analysis is then considered to be bridging the so called semantic gap, which is the conceptual space between the raw properties within the media content, known as low-level features, and the actual meaningful interpretation of a scene [1]. By means of these techniques it is possible to identify and block adult content before it reaches the wrong audiences. A number of solutions exist for adult image detection [2, 3, 4, 5, 6], and fewer have been proposed to deal with video mainly extending the image-based methods [7, 8, 9] or using a more sophisticated set of features exploiting the temporal cues in video [10, 11, 12]. Taking into account that adult content detection is a video classification problem, a number of interesting and promising methods to be specifically applied for the mentioned purpose are reviewed [13, 14, 15, 16, 17, 18].

The goal of the present work is to demonstrate that is possible to achieve high accuracy rates for adult content discrimination by using temporal features, particularly shot structure and camera motion, together with image-based features, specifically skin detection and color histograms representation, by means of machine learning techniques, explicitly Support Vector Machines (SVM) broadly used in the field [19, 20, 17, 10]. We experimentally compare different combinations of features with the SVM algorithms using a challenging data set containing more than 7 hours of video. The results in our experiments indicate that our hypothesis are correct.

[1] http://www.medietilsynet.no/en-GB/

[2] http://ec.europa.eu/information_society/activities/sip/index_en.htm

2 Literature review

Adult-content identification in images has been extensively explored [2, 3, 4, 5]. The most typical means of identifying pornography are strongly based in skin detection [21, 6, 22]. In the case of videos, the most common approach is basically an extension of the image detection techniques [7, 8, 9] applied for a set of selected frames representative of the clip known as key frames. More novel techniques utilize the temporal cues as well to extract information such as motion patterns [10, 11, 12]. Based on that data that is representing such characteristics of the video, a number of classification techniques are used to identify whether or not the clip is regarded as adult content. One of the most vanguardist classifiers applied to this and some other visual recognition tasks are the so called Support Vector Machines (SVMs) [19, 20, 17, 10]. Some other works have posed a more general problem in which there are multiple categories in the video files such as films, news, commercials, music videos, etc. The research in this area has been more extensive and a different set of features of those used for the adult content identification are utilized, e.g. the structure of the scene based in shots [13, 14] or the type of camera motion [15, 16, 17, 18]. We will organize the present literature survey by exploring six topics, first the **shot detection techniques** used to recover the scene structure of the clips extensively used for video categorization problem, then the **key frame extraction** required for the image-based classification, followed by the **camera motion identification** whose application for video classification purposes will be discussed later, next the typical **skin detection** techniques applied for adult images, then **motion analysis** for video categorization (not to be confused with camera motion identification), and finally **video classification techniques**. The most interesting works in each of the six topics will be discussed in more details as cases of study.

2.1 Shot detection techniques

Disregarding the audio, a video is merely a coded sequence of images, each of which receives the name of frame within the video. This images can be analyzed separately from the perspective of image categorization problem. However the selection of such frames is not arbitrary. The first step is to segment the video in shots. A shot corresponds to a temporal portion of the sequence containing an action happening at a continuous time with no interruptions. These units are generally identified by detecting abrupt changes or degraded transitions, known as shot boundaries. The general idea is to compare the features between two consecutive frames to detect a significant difference corresponding to the signature of a shot boundary. Lu [23] proposes a refinement method that uses color histogram based features from every frame in each of the already detected shots and applies later post-refining to eliminate both false positives and negatives. Baber [24] on the other hand uses a more simple approach by means of entropic comparisons in the first pass and false positives in the second pass by comparing similitude between SURF descriptors extracted from the candidate boundaries.

2.1.1 Shot detection – Cases of study

Entropic and SURF-based detector

As done by [24], both the abrupt and dissolved transitions can be located by comparing the *entropy*[1] between consecutive frames. For the abrupt changes the difference in entropy between frames will be high, while the *fade-out* [2] or *fade-in*[3] transitions will show either a continuous decrease or increase in the entropy respectively. To recognize such gradual transitions, the concept of a fade signature is introduced to characterize the entropic changes typical of *fades* that are recognized by employing Run Length Encoding (RLE). Since some sudden changes in illumination, extreme motion, or gradual disappearance of an object within a scene can also trigger a big change in the entropy, Baber [24] uses a pair of thresholds to decide. If the entropic change is not big enough to be considered a shot boundary straight away, a point matching correspondence is done using SURF descriptor. For the scenarios described before, there will be certain correspondence between points even if the entropy is very different, so when the euclidean distance between the extracted SURF features of two correspondent frame pairs is big enough, then the shot boundary detection can be confirmed, otherwise the false alarm is just ignored.

Color histogram post-refinement

One of the problems of shot boundary detection algorithms are the recall and precision values. While some methods have an acceptable performance it is also true that they include some false alarms in the final segmentation while failing to detect some shot boundaries, particularly those with degrading transitions. The main focus of the work by Lu [23] is to apply a three-phased post refinement process to a set of shot-based segments within a video by means of Color Histograms computed from the HSV color space (with 12 hue, 4 value and 4 saturation bins). A single feature is extracted by shot, and it's basically the probability density function (PDF) of the normalized distance between the color histograms of each of the frames to the *characteristic color histogram* of the whole shot (bin-wise median histogram of every frame in the segment). Such PDF is represented using a truncated exponential function, this final representation will be used for comparison and it is called *log-likelihood*. In phase 1 all the adjacent small shots are merged since, just as pointed out by [24], we can expect a reasonable minimum duration of every shot within a video structure so the audience can be able to understand it. In the second phase the PDF of the color histograms corresponding to each pair of presumed adjacent shots are compared, if the feature distribution is similar then the shot boundaries are genuine, otherwise the segments must be merged. The third and final pass is for detecting transitions that were not properly identified within the presumed shots (fade-in, fade-out, and the scene dissolving transition, which consists of the first shot becoming progressively more transparent while the opacity of the second increases). When such dissolving scenes occurs, it is possible to notice an abrupt change in the log-likelihood ratios right in the moment the vanishing effect is over. This would correspond either to a valley or a peak in the log-likelihood variation plot. To detect

[1] Entropy is a measure of average information or variability contained in an image

[2] Fade-out refers to the common editing effect in which the scene is gradually dissolved into a black image by progressively decreasing the illumination

[3] An introductory transition done by gradually increasing the illumination from a black image to the scene

such point the first $\Delta J(i)$ and second-order finite-difference, $\Delta^2 J(i)$, of the log-likelihood are computed while sequentially scanning the shot. $\Delta^2 J(i)$ will then reach a peak when the abrupt change is happening, meaning the two adjacent frames being at each side are shot-boundaries. The shot is then divided, and the same process is executed in the second split shot to look for further transitions. *Phase 1* and *3* are used to *eliminate* the false positives while *Phase 2* is for correcting the *false positives*, and each of them can be independently executed according to the post-processing requirements. The result is a robust algorithm with a nearly real-time performance, but still depending on the quality of the previously executed segmentation.

2.2 Key frame extraction

As mentioned earlier, a video sequence is composed of a series of successive frames shown one after another to create the illusion of movement. This means that each of the frames can be analyzed separately as an image, so features can be extracted from each of them to, for instance, provide clues about the category of the whole video sequence. Nevertheless, before actually proceeding to applying image analysis techniques, it is necessary to extract a set of representatives frames that accurately capture the essence of the scene (an arbitrary set of related frames that could be a single shot). This is because analyzing every frame would be very computationally expensive, in the first place, and also will imply a great amount of redundancy since the visual variation among some subsets of frames is very low. To visualize this property in the context of adult content detection, consider the fact that some pornographic scenes might start with an undressing sequence followed immediately by sexual activity without any shot transition. The keyframe extraction techniques are closely related to the shot boundary problem mainly because a very effective way to detect them both is to measure the degree of variation within the visual information among successive frames. Truong [25] published a review of different methods for video summarizations. According to this classification, the number of key frames to be extracted to abstract a given sequence can be fixed *a priori* when constraints such as bandwidth or storage capacity are present in the requirements, or they could be determined *a posteriori* by the degree of visual change in the shot, a task that is basically done by measuring the similarity between the current keyframe and the next candidate sequentially until a threshold in the euclidean distance between them is exceeded. There exists a wide variety of similarity functions used for such purpose. The *a priori* methods are clearly the most simple solutions, both in terms of implementation and computational complexity, and for some applications these are perfectly acceptable. For instance [7] just extracts frames at a fixed interval to save up processing time. In Yinzi's approach [26] the number of key frames to be extracted is defined by the information given by the user. The *a posteriori* methods such the ones used by [27], on the other hand, select the first frame within a shot as a keyframe and use color histograms and global motion feature extraction as the similarity measure. In [28] a single representative frame based on average color histogram is selected. In the following lines we will further describe the two most interesting approaches.

2.2.1 Key frame extraction – Cases of study

Cumulative color histograms
The color histogram based approach presented by Ferman [28] computes first a cumulative color histogram for a group of frames in a sequence (that might be a whole shot) and normalizes it so an average histogram is obtained. Since this representation is vulnerable to out lier frames, a median histogram can be used instead implying the selection of a sole keyframe in all the group having the closest color histogram to the average histogram. More sophisticated refinements for this basic method are proposed within the same work however they won't be further discussed.

Color and displacement
Within the content-based retrieval system proposed by [27], a keyframe extractor module working with a three modality criterion is proposed. These are shot, color, and motion modalities. The shot modality is simply to select as key frame the first frame of every shot. If any subsequent selection is to be done, it will be based on comparisons between this first frame and the following ones within the same shot unit. The *color modality* of the criterion consists of a simple frame to frame comparison (last keyframe against each of the candidates) based on a color histograms descriptor using euclidean distance and a threshold to detect significant differences and decide whether or not to add a new keyframe. When a new keyframe is detected the same process is repeated until no more candidates are available. The *motion* modality first estimates global motion to detect camera movements. These are categorized as either *panning-like*, which includes the panning in every direction as well as the tilting, or as *zooming-like*, which incorporates the dollying and every other motion perpendicular the image's plane. Every time the camera makes a *zooming-like* movement, the first and last frame of such sequence will be selected as key frames. For the *panning-like* movements, the number of frames will depend directly on the displacement detected, a new keyframe will be captured every time the scene displacement is equal or exceeds 30%. Note that global motion information can be also used as either a simple heuristic as done by [9], where the frames are extracted just when the camera is static (global motion below a given threshold) , or as a feature for content classification as in [14]. A deeper discussion about camera motion identification is carried out in the next section.

2.3 Camera motion identification

When every pixel present in the scene is moving following a pattern, the scene is considered as having global motion. This kind of movement implies, in most of the cases, that the camera is performing an operation (panning, translation, zoom). To identify the motion pattern yields to recognize the kind of operation the cameraman is performing. This provides many clues about the intention of the director, or whoever is in charge of recording, providing semantic clues. For instance zooming-in a particular object would show the intention of emphasizing its importance. Also the presence of certain patterns of camera operations are quite distinctive of very specific video genres. A cinematographic film for instance would have a wider variety of camera movement while a news segment will not have motion most of the time (when the anchorman is speaking). Pornographic films are not famous for being particularly rich in either editing ef-

fects or artistic camera movements. Therefore camera motion is a very promising discriminative feature for adult-content detection. To identify these operations from the sequences of images implies first to compute the motion vectors representing the movement for every pair of successive frames in the sequence and the later analysis by means of different techniques. Some methods will be described in this section as a *stand-alone* problem except for the last point in which some details about the use of such features for scene categorization problem are mentioned referring to the work of [14]. The characterization and use of such features for a machine learning classifier will be discussed in section 2.6. The global motion is identified by using the motion vectors extracted from a pair of sequential frames in the video stream. This paragraph will summarize an overview of the existing approaches for camera motion identification. Zhang [27] detects the global motion in the scene and extracts key frames based on the displacement or zooming of the camera. However this information is not used for characterization purposes but rater as a similarity measure criteria in the keyframe extraction process described in section 2.2. Camera motion can also be used as heuristic guidance to detect the frames which are most likely to contain pornographic material, Chang [9], for instance, points out the tendency of obscene scenes to have virtually not global motion and extracts key frames only from them. However that characteristic does not universally hold for adult content. In the work by [29] the camera motion is estimated by dividing the image in 7 blocks and computing mean and standard deviation of the motion vectors magnitudes in each of them. Different combinations among these blocks identify the camera operation as shown by table 1. Lee [30] achieves real-time motion classification by means of templates that are matched against the denoised motion vectors. Liu [31] extracts motion vectors from 100 video frames previously identified as containing either panning, rotation, tilt or zoom camera motion and trains 4 binary SVM classifiers to identify the motion. The works discussed here focus exclusively in the problem of camera operation identification. Some other methods [14] [17] [15] [16] [32] incorporate global motion information as well as statistical information for video categorization.

2.3.1 Camera motion identification − Cases of study

Camera motion annotation

The method proposed by [29] aims to identify the kind of camera motion occurring in the scene by calculating mean values and standard deviation in the optical flow for certain sub-regions in the video canvas. The problem can be seen as a camera parameter estimation consisting of 9 unknown parameters associated with the three coordinates of translation, given by the vector $T = (T_X T_Y T_Z)^t$, the three rotation axis Ω_X (tilt) Ω_Y(pan) Ω_Z(Z-wise rotation), the distance Z to the image plane, the focal length f of the camera, and finally the zoom factor r_{zoom}, all of which are independent from the image's pixels, except for distance Z [29]. The camera model is shown by *Figure 1*. First the optical flow between two frames is computed using Lucas and Kanade's algorithm [33] having the lowest error rate among the approaches studied in Barron's work [34].The next reasonable step is indeed to discern whether or not global motion is present in a given sequence, which can be done by simply computing the overall mean value of the optical flow vectors. If there's indeed some camera motion, then the algorithm further proceeds

to identify it. To discriminate the camera operation, the precise calculation of the vertical and horizontal components of the vectors, $u = (x, y)$ and $v = (x, y)$ respectively, is not required. Furthermore it is possible to identify most of the camera operations by just computing the mean value and standard deviation for some specific sub regions in the canvas of the video. The total area is divided in 7 non overlapping sub-regions consisting of a central one, 2 axis-like stripes forming a cross, and four block like sub-regions for each of the corners. Canvas sub-regions are shown in *Figure 2*. If the mean value of the motion vectors within a given sub-region is below a certain threshold, it can be considered as having no motion, and using the standard deviation, it is possible to find out if a constant velocity motion is occurring in the block. The camera operation can be identified depending on $u = (x, y)$ and $v = (x, y)$ being either constant or zero in a particular combination of sub-regions. Table 1 shows the rules used to identify camera operation based on u and v. From this table, it is evident that *c)* r_{zoom} and T_Z are indistinct by just applying this set of rules, so further analysis is required. The use of just the mean values of u and v will be sufficient to identify in which of the three planes the motion is occurring. The standard deviation values (which help to identify is the velocity is constant or not) are needed to discern between *a)* horizontal translation against panning, and *b)* vertical translation versus tilt. These pairs of operations are easily confused due to optical flow calculation inaccuracies. The x for case *a)* or y for case *b)* are very small, in either tilting or panning movements respectively, and therefore easily confused with 0). To overcome this problem a simple criteria considering that $u_{pan} = (x, y)$ exhibits a considerably greater variation than $u_{horizontal_translation} = (x, y)$ is applied. Something similar is done for the case *b)* and for *c)* where, given that r_{zoom} is independent of Z, it can be assumed it will have smaller variance than T_Z under ordinary circumstances. This method will be still applicable for complex camera movements consisting of a combination of two or more camera operation for it will detect the dominant motion in the segment. Still [29] tests its algorithm just in very particular segments of 11 frames containing such movements. The success of the method then would depend not just in the video segmentation (also proposed by the author within the same article), but also in the optical flow computation, provided a sparse flow would yield to failure.

Camera operation	u(x,y)value	u(x,y) value in region	v(x,y)value	v(x,y) value in region
Rotation (Tilt)	0	at X0 and Y0	Constant	Y0
Rotation Pan	Constant	X0	0	at X0 and Y0
Rotation (Z-wise)	0	Y0	0	X0
Translation (Horizontal)	Constant		0	everywhere
Translation (Vertical)	0	everywhere	Constant	
Translation (Z-wise)	0	X0	0	Y0
Zoom	0	X0	0	Y0

Table 1: Camera movement pattern identification rules. The u and v values (constant or zero) in certain sub-regions of the image can be used to detect every type of camera operation. Note how the conditions for Z-translation and zoom are the same, making them indistinguishable from each other within without further analysis.

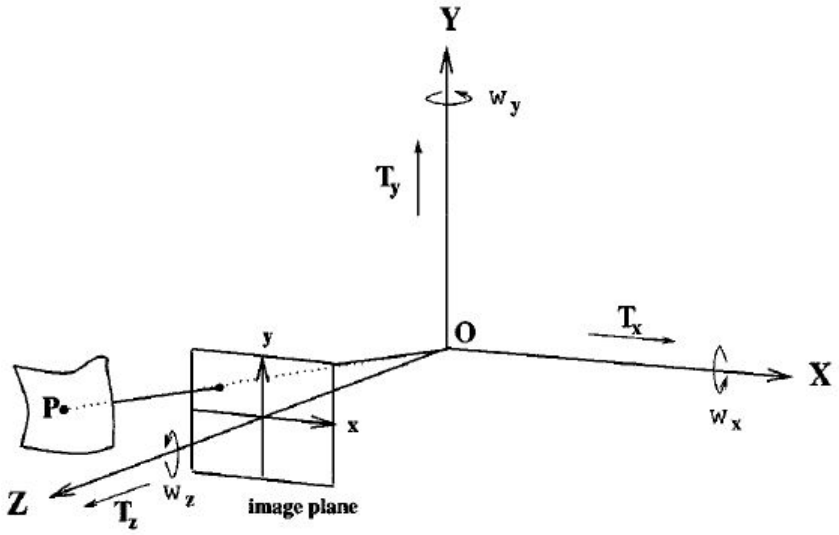

Figure 1: Camera model. Courtesy of [29].

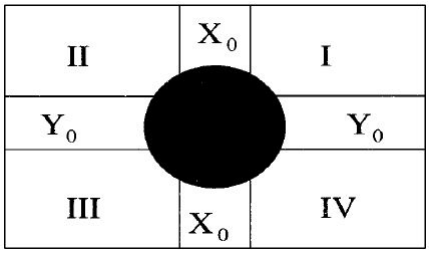

Figure 2: Image subregions. Courtesy of [29].

Scene-related motion features

The interesting part regarding the work presented by [35] are the holistic features extracted from the sequence, aiming to represent the content of the scene globally. This methodology focuses strongly on structural information contained as a whole within both spatial domain (extracted frame-wise) and temporal information (derived from a group of frames). To create such features two different aspects are modeled, *1) Intra-frame* information, corresponding to spatial structure and *2) Inter-frame* content, which are motion patterns. For the *inter-frame* information, a sound distinction between foreground and background is not made, which implies the absence of recognition for individual objects or local movements. The optical flow between pairs of consecutive frames is calculated to estimate vertical and horizontal velocities in the motion vectors. The velocities are analyzed in the grounds that they will be distributed around zero for scenes with no global motion or will exhibit shifted magnitudes (for every vector) in scenes with camera motion. This representation is inspired in the functionality of the Medial Superior Temporal Cortex that allows the human brain to quickly interpret whether or not the scene contains movements. In the same stage of visual processing a quick distinction between moving objects or self movement based on spatial and structural information (and with directional selectivity) is made. The so called scene related features are, in summary, a way to model the human perception of movement and structural information within a scene.

Template matching motion discrimination

As previously discussed in section 2.1, a video sequence can be segmented in the temporal domain into units called shots by identifying scene transitions consisting on either abrupt or gradual changes. Such units might consist of multiple camera operations sequentially executed. Lee [30] parts from the idea that it is possible to further divide each shot by identifying the operations of the camera within the scene. Computing the optical flow results in high computational expenses and inaccuracy in the motion vectors. This makes necessary deeper analysis as pointed out by [29]. To overcome such constraints, [30] makes use of motion vectors that are part of the MPEG coding scheme. The approach pursues three basic properties, a) capacity to work with noisy motion vectors, b) robustness against big object movements, and c) reasonable speed to classify camera operations present in real video sequences. The algorithm *1)* First preprocesses the motion vectors from the bitstream to create the so called motion vector fields (MVF), then *2)* divides them into non overlapping regions (sub-MVF) and finally *3)* matches the background and the object's sub-MVF against a set of predefined templates. In the MPEG standard, the so called P-frames are predicted for the reference I-frames (which contains the richest amount of information and the lowest compression) by using the motion vectors and the error computed from the difference between the prediction and the actual frame. This encoding scheme focuses on decreasing the prediction errors rater than in accurately modeling the optical flow and causes homogeneous regions to be encoded as random noisy patterns. *Step 1) Preprocesing motion vectors*, is done by applying median filters for the suspected noisy regions in both horizontal and vertical component of the motion vectors. Similarly to the work carried out by [29], the image's area is divided in blocks following the same philosophy, a vertical and horizontal cross-like axis in the middle of the field, a central block, and four corner blocks. Refer to *Figure 3*. The set of all

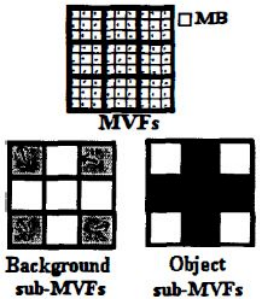

Figure 3: Motion Vectors Field block division. a) The so called MVF sub-regions consist of basically 9 blocks, and those are done separately for b)background and c) objects. Courtesy of [30].

the motion vectors after noise subtraction receives the name of Motion Vector Field (MVF), and each subdivision block is called sub-MVF. Then *Step 2) MVFs magnitude computation* , for each sub-MVF, represented as s, and frame n, the average magnitude of the motion vectors, $M[s,n]$ and the so called argument $\beta[s,n]$, are computed to determine whether or not the frame contains sufficient global motion to be classified. *Step 3) Detect global motion* $D_{bg}[n]$ is computed by basically averaging how many of the background sub-MVFs contain significant motion. Then if conditions are met, *Step 4) Template matching* will be performed using a series of templates. In order to do it, a measure $A_{\theta[i]}$ based in phase histograms is extracted from the motion vectors of the frame, and then each of the 16 bins of the histogram are compared with the corresponding ones in the histogram of each of the templates. If the similitude is above a given threshold τ_{temp} then the frame will be labeled as having the motion represented by the template. In order to declare that a camera operation is occurring, the identified movement needs to persist for at least 10 frames. Although this approach just identifies six basic camera operations, it can be extended to include panning-like and camera rotation as well.

Global motion based classification

Roach [14] addresses the classification problem by solely use of motion features without taking into account the structure of the shots. The lowest error rate reported by this approach is of 6%. The so called video dynamics, are basically the background camera and foregorund object motion. Rather that using the motion vectors of a particular format, the camera motion is based in template matching optical flow computation. Rather than detecting the precise camera operation as some other approaches, the movements are simply categorized into X (including horizontal tranlation and pan), Y(vertical translation and tilt) and Z (Z translation, focal length changes and zooms). According to the author, this 3-movement based abstraction have been predicted to contain enough information for discriminatory purposes. To obtain the *background motion* vectors, a mask is extracted from the current frame to be compared against the previous frame. A search neighborhood, shown by *Figure4*, is defined based in the premise that the expected translation is relatively small. When the highest correlation in the search area is found, the correlation coefficient must be greater than a first threshold to make sure the blocks are similar enough. Then the number of matching blocks is checked against a second threshold, if they are too many, it means

there is an homogeneous area and therefore the computed vector is discarded. To calculate the *foreground motion*, the camera motion is substracted from the optical flow corrsponding to two consecutives frames in a pixel-wise fashion and then is procesed using morphological opening. The process is ilustrated by *Figure 5*. After this is completed, then a second order motion signal δ_t is computed.

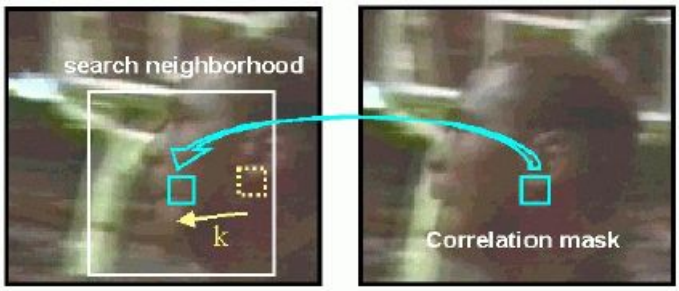

Figure 4: Neighborhood search used to compute optical flow. a) Shows the previous frame and b) the current frame. Block k from in b) will be searched in the area shown in a) to compute the motion vector k. Courtesy of [14].

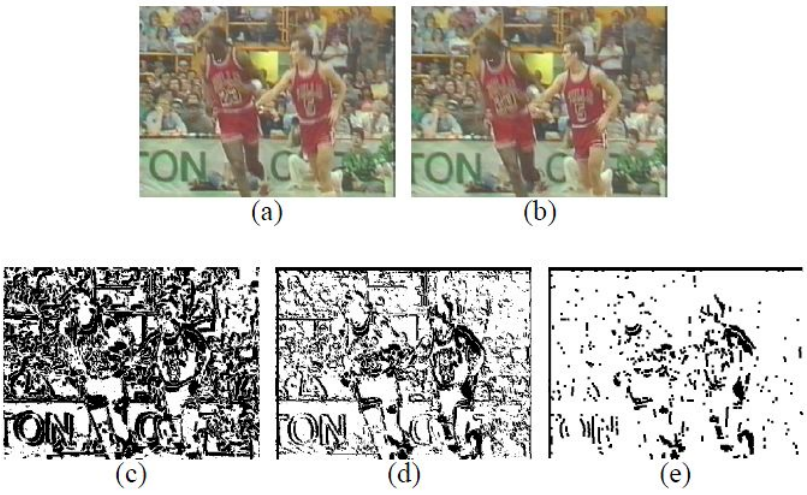

Figure 5: Scene motion. a) Frame i-1, b) frame i, c) Motion map, d) Camera motion compensation, e) After applying opening morphological operation. Each black pixel represents the pixel-wise motion detected over the threshold. As shown by e), the motion is mostly concentrated in players' arms and legs. Courtesy of [14].

2.4 Skin detection

As mentioned early before, one of the more recurred strategies is to extract a set of key frames (topic reviewed in section 2.2) representative of a video sequence and apply to each of them image classification techniques. Such approaches are currently the most exploited in the domain

of adult content classification. When properly combined, the individual scores of the images can contribute in an important degree to the final video score. Evidently the spatial features are the basis for this kind of approaches. Skin detection is probably the most simplistic way to characterize adult image since they commonly exhibit a high rate of human skin exposure. Even among different ethnicities the range of skin colors is well defined mainly because of melanin concentrations present in the dermis affecting just the intensity of the skin color but not its hue [36]. Human skin tends to present yellowish/reddish tinctures, considered as warm colors. When there is sexual activity involved such tinctures, especially the red ones, tend to get accentuated, making this color-based detection specially suitable for treating adult video. Fair approximations can be quickly computed by just considering the values within a particular range for a given channel in a defined color space as skin regions (being HSV, and CIELAB common choices) and building-up a binary image to represent a skin map. More sophisticated approaches are based in calculating the skin pixel likelihood [18] by formulating the problem in a probabilistic framework rather than assuming every color within a sub-space will belong to a skin region. Skin detection has the disadvantage of introducing false positives that occur mainly when: *a)* Some objects in the scene can have the same characteristic colors as human skin introducing false positives, *b)* The scene shows a close up to a given part of the body exposed, which occurs very frequently in faces close-ups. These cases are the motivation behind implementing object recognition approaches for adult video recognition, e.g. detect faces to recognize close-ups. Marcial [37] presents one of the most simple approaches. An image will be considered as pornographic when at least 50% of the image's pixels are identified as skin. Jones [36] proposes an image classification framework in which a set of sample pictures where the skin regions are segmented by hand in order to construct skin color histogram model. The histograms' counts are transformed into a discrete probability distribution by dividing the histogram count associated with an RGB triplet by the total count for every bin all over the whole histogram. When a new image sample comes every pixel is classified by calculating the ratio of probabilities of it as either belonging or not to the skin class. The resulting skin map is used to compute a set of features that are finally classified using a Neural Network classifier. The attributes include percentage of skin regions in the image, average probability of skin pixels, size in pixels of the largest connected skin patch, percentage of unclassified pixels (with 0 probability being part of both skin and no-skin class). In the work by [18] first a skin map is generated by using a bayesian probability estimation in the HSV color space. For every pixel, the probability of it being a skin pixel given the observed color is calculated. If this value surpases a certain threshold then the pixel will be considered as skin region. The resulting binary mask is refined by applying some morphological operators to reduce the noise and finally is evaluated by fitting ellipses on the skin regions to distinguish between adult images and portraits images. Since the criteria defining pornographic often involves the exposure of certain body parts in the scene, there is a family of approaches that aims to automatically detect them in the image domain rather than relying on skin detections. Face position is an auxiliar feature combined with skin layout [21], or nipple detection [38, 6, 22]. Face detection is often used together with skin detection in order to eliminate false positive generated by face close-ups. Stottinger [21] uses multiple face detectors to track all the faces in the image. In parallel a skin map is created. The color space selected is YC_bC_r because it separates chromaticity from

luminance and it is quickly converted from RGB. The C_b and C_r ranges for the area belonging to the detected face will be extracted and used to generate an adapted skin detection model. This procedure can be extended for multiple faces and all of the generated models are kept rather than discarded, all over the whole sequence. The results are good for short online clips but this fails for longer clips such as movies that have more persons on scene and more varied ranges of colors. The measure used to differentiate between adult and legit content is the relation between skin enclosed inside faces and the skin present in the whole image. This measure is averaged for a fixed number of frames. Adult material exhibits a small amount of face skin coverage when compared to the total skin present in the frame. This intuitive criteria is converted into rule based model which is used for classification. When the $face/total.detected.skin$ is under a fixed threshold, and the total skin exposure above a second threshold, the video is flagged as adult content.

Nipple detection

Fuangkhon [38] aims to detect nipples in images to judge the adult content. This is done by using a sliding window of a fixed size to search over the previously detected skin regions in a image. Since providing negative examples is not practical in this case, a class/non class classification scheme is used. Then a neural network using Kohonen's self organizing maps is used for classification. The probability of correct classification of harmless images is 99.78%, however when the image is obscene just the 65.39% of the nipples present in the picture will be identified. Anoter problem with this method is the high sensitivity to the size of the window which might significantly vary in real world images. A more shopisticathed approach is presented by Wang [6] which extracts the skin regions and discards false positives using texture as an auxiliar feature. Instead of directly analyzing the resulting skin regions, a sub set of candidate areas is generated on the basis that the nipples present a more reddish coloration and that this areas differ considerably in color from the sourrounding pixels. The feature extraction phase is done just for candidate regions identified in the previous step, the attributes are modeled in terms of *a)* spatial and *b)* morphological notions. The main notion behind *a)* is that the localization of the nipples in the upper part of the human body, while the intuitions about *b)* is that a positive detection must fit an elliptical shape. The feature training and classification is done by using a *Forward Propagation Neural Network*. Nipple detection as an acceptable method for detecting adult content suffers from two reasoning flaws. One is the fact that the detection might correspond to a male which is not considered offensive in many cultures. Also it is important to keep in mind that the female breast does not represent the most strict criteria to judge an image as pornographic. Moreover it is even widely acceptable in many cultures and contexts, such as medical magazines or breast feeding videos in public sites such as youtube.

Detecting erotogenic parts

Shen [2] uses a sequential detection scheme locating, in the following order, face, trunk, skin area, and finally the genitals. Face detection is achieved by applying the classic and famous Viola and Jones face detection algorithm [39] which was selected because of its robustness in the basis that the rest of the algorithm depends on this first tracking. In a similar fashion as in [21], the

trunk is found by searching colors with similar ranges of those of the face just detected (assuming a highest probability of the trunk being under the face). An ellipse proportional to the size of the head is fitted to characterize the envelope of the trunk and its position. The color space YC_bC_r is used because of its high cohesiveness in the skin color for different ethnics. The skin model is then re-adjusted by incorporating information of the skin color present in the trunk. The first object to detect are nipples, the same set of distinctive characteristics used by [6] are utilized in addition to the fact that more edges are found in the inner nipple area compared with the external one. The last and more characteristic step of this approach is the pubes tracking. The first obvious criteria having located the head and trunk is the pubes position with respect of them. A modified filtering window analogous to that used by the Adaboost detection implemented in the classic solution by [39] is applied in the area of interest. Finally a set of rules considering which visual features to find and in which parts of the window are applied to confirm or dismiss the presence of a pubes. The scheme of progressive object detection in this approach is interesting in the sense that it can be used to measure the degree of erotic content similarly as in [3]. The performance is significantly higher when compared to the traditional skin based methods. However the extension of this methodology for video categorization is not straight forward and it would be interesting to explore and evaluate such capabilities. A clear limitation of this method is the schematic functionality that aims to detect levels of nudity but not scenes with sexual activity. Such scenes would naturally be found much more offensive than the nudity itself, and the possible layouts and combination of objects that might appear on them makes this approach not suitable for them.

Bag of Visual Words (BoVW), an alternative to skin detection

As stated before, video classification tasks are firstly anchored in image classification techniques and secondly in skin detection which is the base of most of the early works on adult content detection. However the simplest kind of such features are insufficient for having robust results [3]. The bag of visual words model is a more sophisticated technique that has increased its popularity among the research community, particularly for the object recognition and image categorization problem. The model applies different algorithms to extract a set of features from a collection of images. While image categorization, generally speaking, focuses on more complex images, object recognition works with datasets containing pictures in which a single object appears such as the flower data set[4]. Deselaers [3] extracts specific patches in the image around interest points defined by difference-of-Gaussian within the BoVW framework. Such features have the advantage to include straightforward color information as opposed to some shape descriptors such as *SIFT* [40]. In the training phase, a Gaussian Mixture model algorithm is employed to generate the visual vocabulary. In the classification phase, log-linear models (LLM) and SVMs are compared using as an input a visual-word histogram. SVMs yield to a slight improvement. For the SVMs, a one-against-the-rest multi class scheme is followed as well as a 5 folds cross validation. The four categories ordered in the degree of erotic content are $\{inoffensive, lightly-dressed, semi-nude, nude \text{ and } pornographic\}$. A minor improvement

[4]The flower data set was created by Maria-Elena Nilsback and Andrew Zisserman for object recognition purposes. Available at http://www.robots.ox.ac.uk/ vgg/data/flowers/

can be seen when skin features are integrated into the model.

2.5 Motion analysis for video categorization

Motion is a typical inter-frame or temporal feature and could be regarded as the main additional cue provided by video-content with respect to static images. The motion patterns that can be detected by using the so called motion vectors of a scene are a very important feature to discriminate the category of video. For instance, news segments where the anchor man is speaking will contain practically no motion except in the face, while action movies in the contrary will have a lot of it. The kind of motion can be mainly classified into two categories, *1) Local motion*, corresponding to objects moving independently from the camera, and *2) Global motion*, which correlates to self-movement in the context of human perception. Since we are dealing with video, the center of the perception is related to the position and movement of the camera instead. Regarding global motion in particular we can say that a) Is a very promising categorization feature because it takes into account the cinematic principles and rules that define the genre of video productions and b) The means of extracting global motion to identify the camera operation is a very complex problem by itself. For these two reasons a whole section has been devoted to the discussion of this issue2.3. For these reasons the global motion is practically left out of this section which will mainly discuss the topic of local motion analysis.

2.5.1 Motion analysis for video categorization – Cases of study

Holistic structural representation

Wang [35] uses a biologically inspired approach that constructs a holistic scene representation by modeling *1) Intra-frame* or *spatial* information and *2) Inter-frame* content (motion patterns). This feature aims to model edges and textures by means of Gabor filters as proposed by [41] without making a sound distinction between background and foreground. This characteristic is inspired in the function of human receptive fields, particularly in the center-on/surround-off antagonic mechanism responsible for the lateral inhibition phenomena. This behavior takes place inside area V1 in the Primary Visual Cortex. A set of Gabor sub band images are created by convolving amplitude of images by Gabor filters to capture *spatial* structure (including texture information). Using them the descriptor called *gist* is created. This feature encompasses texture and edges and, in theory, images belonging to the same category are expected to have similar patterns.

Motion periodicity

Jansohn [10] points out the tendency of pornographic sequences to contain repetitive motions patterns typical from the sexual intercourse and uses them as one of the features to detect adult content. In this work three different methods for detecting motion periodicity are experimentally evaluated. The first of them is based in an autocorrelation function inspired on the works by [11] and [42]. The second one is an slightly modified version of the first one that uses sliding temporal windows of three seconds. The last method is called motion histograms and it is based on the work by Ulges [43]. Within [43] the motion vectors are extracted directly from the MPEG compressed domain by the XViD encoder. After that the area of the motion vectors is divided in 12 blocks and inside each of them an histogram of the 2D components for every vector within

is generated. Each histogram will be of size 7 by 7. This descriptor is useful to have track of the particular motion present as well as the area in which it occurs. The most effective periodic motion descriptor is proven to be *motion histograms* in the experiments conducted by [10] together with the BoVW approach yielding to an error of 6%. Nevertheless motion periodicity as the sole characterization mean (feature), as Jansohn himself highlights, will produce some false positives because the same kind of patterns can be found in not-harmful content such as dancing clips. This fact makes clear that adult content detection is not the only application for cyclic movement analysis. The first example of this is provided by Fujiyoshi [44]. The idea is to construct an star-like skeleton that models the silhouetthe of the foreground objects. The position of the segments related to each other can be, at least potentially, used for gait recognition while the cyclic motion of them provide a mean for action detection (walking, running) which is particularly useful for surveillance applications. The method proposed is computationally efficient and does not require big resolution in the masks of the target objects. Cluther [45] bases this analysis in a self similarity measure whose behavior will be also periodic. By applying time frequency analysis it is possible to characterize the periodic motion.

Camera motion features applied to video categorization
Yuan [17] combines shot information with camera motion. The mining is obviously performed in the spatial and temporal domain. For the temporal domain, 3 statistical 1D features and a single 4D camera movement feature are used. The statistical features are: *1) Average shot-length* is computed by simply using the mean of the duration of every shot after boundary detection, *2) Cut percentage*, calculated as the ratio of cut transitions to the total of every type of transition, and finally *3) Average color difference* which is just a mean of the color difference within the whole video sequence. Each of the 4 dimensions of the camera motion feature consist of the average motion extraction ratio of a particular kind of camera movement to the totality of motion present in the sequence. The four kinds of motion identified are `still` (no motion), `pan` (vertical translation), `zoom` (z-axis translation) and `others`. The three statistical 1D features are quite useful to discriminate between music clips (which are characterized by larger color differences and shorter shot length) and sports (which have less color differences and larger shot duration in average). The camera motion is very useful to identify the sub-genres within *sports*, for instance still motion is predominant in *soccer*, and pan is very common in *volleyball*. Please note that for the camera motion features the raw motion vectors are not used as input for the classifier but rather the ratios of each of the 4 operations against the rest of them.

2.6 Video classification techniques

As a side note, please recall that pornographic content detection is a binary classification problem when the only aim is to differentiate such media from any other possible genres. The so called semantic gap is the conceptual space between the raw properties within the media content, such as color, edges, motion or scene structure, to mention just a few, and the actual meaningful interpretation of a scene. This is the moment when the use of machine learning techniques come into play. The so called classifiers are computational tools that make use of statistical techniques to redistribute the data and estimate boundaries that properly separate each set of data belonging

to several classes. Without going into further details, a classifier within the supervised learning paradigm, is first feed with a number of positive examples previously identified as belonging to a given class (this is where the name of supervised comes from) in the process called *learning* or *training*. After this, the classifier can recognize and label automatically any new sample as belonging or not to a given class. This process is based on similarity comparison between the received sample's and the expected data distribution. This data is originated from a set of *features* that represents certain *aspects* of the media content. This *aspects* or cues, more formally referred to as *features* have to be carefully chosen to maximize the discrimination between genres. This is a general overview of the mechanics involved in a media content categorization system.

Bayesian classifiers

The Bayesian formulation presents the big advantage that some *a priori* knowledge can be included in the inference process of the classifier [13]. Therefore, shot duration and structure can be part of the statistical model as a prior to judge whether a video belongs or not to a given category. Based just on this previously available heuristic information, we could know that a sequence composed by many short shots would be more likely a musical video rather than a film, for instance. Lin [46] takes advantage of transcripts to categorize news video sequences by combining image and text features employing a separate SVM for the classification of each of the feature. Two different combination strategies are then compared, the first one consisting of a simple feature vector concatenation (also known as late fusion) and the second one by using another SVM classifier to take the final decision (meta-classifier).

Support Vector Machines

In late years Support Vector Machines have gained quite a lot of popularity as classifiers, particularly in computer vision tasks [20]. Support vector machines basically execute a mapping process to transform the input space into a higher dimensional space called feature space where a hyperplane will be created to separate the data according to their classes [19]. This is done by means of a kernel function which is utilized to avoid the explicit computation of the mapping by defining a dot product in the feature space using the kernel. This is informally known as the kernel trick [19]. Even though SVMs are just employable for n-binary classification, they have high accuracy ratios and have the interesting property of being insensitive to the relative number of positive and negative examples in the training set [18]. Therefore they are widely used for image and video categorization problems [10, 7, 8]. Among the different kernels functions that can be used to draw the boundary between classes, the Gaussian (RBF) kernel is preferred for content based video genre classification [47] and adult content detection [18] having just the inconvenience of being sensible to the γ parameter. LibSVM tool is referenced by many visual content-based applications [48, 46, 8, 49, 31]. In an experiment conducted by [18], where a specific framework to extract a discriminative feature based on skin detection is also proposed, different SVM kernels are compared. Among Polynomial, Sigmoid, linear, and Gaussian (RFB) kernels, the last one is proved to have a superior performance, yielding to an accuracy 97%, and linear kernel being the worse with just 76% accuracy.

2.6.1 Video classification techniques – Cases of study

Structural features

The holistic representation of structural elements in a scene as described by [35] computes spatial and temporal features which have been discussed in previous sections. Here we will focus on the experimental setup, results and usage for those descriptors in a classification problem. The experiment is done using two different data bases, the first one contains 4 classes {Tennis, Talking heads, Highway, Outdoor} from videos downloaded from youtube, and the second one, with 7 classes {city streets, hallway, industrial & parking lots, inside mall, library, talking heads, and woods}, was obtained using a mobile device. Every clip has the same number of frames: 200. The experiments are carried out using two classifiers, SVMs and Multi-Layered Perceptron (MLP). The training is done by selecting two random clips from each class and using the rest as testing set. For the first database the overall accuracy was 95.88% while for the secong 84.14%. The greater succes in the first database is attributed to the considerably high dissimilarity between the classes. In the second database some difficulties to properly separate classes among human-made scenery (such cities versus libraries) are observed given than 51% of the library frames were misclassified as *street*. When the structural components of two different class are expected to be similar this method is bounded to report low succes. This suggest the sole usage of these two structural features for porn detection is unsuitable.

Image based features

While temporal features can be straight forward used to characterize a video sequence as a whole, the spatial attributes (also called image-based) have to be properly combined so the resulting representation is meaningful for properly judging the category of the clip as a unit. Let's think of a simplistic scenario. A movie containing a single erotic shot exhibiting genitals or any other potentially offensive content. To categorize the movie as pornographic could be wrong (many art cinema movies contain erotic scenes and are clearly not pornography) in one hand while in the other in some strict environments it might be desirable to flag it as containing an inappropriate scene. While this example is trivial it shows the need of having a robust methodology to integrate the results of classifying each of the frames composing a video clip. Lee [7] analyzes frames extracted at regular intervals and utilizes SVM with Radial Basis Function Kernel (also known as Gaussian) to generate two sets of intermediate features. First the skin probability is computed for each of the frames using a Gaussian mixture modeling. A training set of images is selected manually from the frames extracted. Then an SVM is employed to classify each of the images to compose the first group of features X_f. The second group of features, Y_f, is created using a group of frames, extracting from each of them an HSV color histogram based features and then applying another SVM classifier. The results of both classifiers, regarded as a new set features X_f and Y_f, as explained before, are used by a third classifier based in linear discriminant analysis. One of the main disadvantages of this approach is the fact that the frames are extracted in fixed intervals which may lead to either redundant information (as the frames might be highly similar) or bypassing representative frames. Another one is that requires human intervention to manually select the adult images for the training. This might imply a gargantuan job when using

a large corpora. Jansohn [10] applies SVMs classifiers as well and calculates a score for each of the processed frames that are combined following a vote-based scheme to create a global score for the entire clip. This choice is based on the work of [50] proving the reliability and robustness of the vote-based rule for combining the classifiers decisions. The consensus or final class is simply the one receiving more votes from the different classifiers.

Shot-based features

Following the same ideas as [13] Roach [14] classifies video sequences in three categories (cartoons, sports and news) by analyzing the foreground (objects) and background (camera) motion. Rasheed uses similar set of features as well as audio analysis taking also into account cinematic principles such as typical illumination and characteristic camera operations present in movies trailers in accordance to its genre [15] [16]. Nam [32] on the other hand aims to detect violent content by entropic bursts measures associated with explosions and gun fire, as well as color profiles present in scenes with blood or fire.

3 Methodology and implementation

The present work starts from the idea that temporal-based features used for content-based indexing and retrieval, particularly *shot structure* and *camera motion*, can be successfully applied for identifying adult content. In addition to these, spatial-based features, specifically *skin detection* and *color histogram representation*, are used as auxiliary data to cover the cases in which the temporal ones are not sufficient. The objective of our approach is to demonstrate that it is possible to achieve a high accuracy in adult content discrimination when combining this set of features altogether with machine learning techniques, explicitly Support Vector Machines, rather than finding or implementing the optimal classifier.

Shot structure [51, 15, 16], camera motion [14, 29] or the combination of both [17] have been successfully applied as features for automatic video categorization and content-based indexing and retrieval, but not for the specific purpose of detecting adult video content [1]. On the other hand, Support Vector Machines (SVMs) have been widely utilized for adult content detection [10, 52, 7, 53] using different set of features, mostly in the image domain. We aim to combine SVMs with the mentioned temporal features plus a set of auxiliary spatial features to add robustness to the application. The spatial features are computed in a very simple way, to be used as complementary information, and are based on *c) skin detection* and *d) color histogram*. Using them we intend to capture information about the content that is ignored by the temporal features. These features are extracted from a set of representative frames called key frames, each of which corresponds to a single shot. The extracted information is combined by averaging the corresponding numeric values within the feature vector, following a feature fusion scheme [54], and is concatenated with the temporal features to be processed by a single SVM classifier. The details about how and why this features are used for discrimination are discussed in section 3.1 while the extraction process of every feature is treated in section 3.2. More details about the fusion method are given in section 3.3.1. For the experimental evaluation of the methodology, fully discussed in chapter 4, specialized software with machine learning algorithms already implemented will be used. The implementation, development, or improvement of machine learning techniques is out of the scope of the project. A diagram summarizing all the steps involved in our approach is shown by figure 6.

3.1 Selecting features for video content description

This section describes in details the criteria used to select the set of features describing the content of the video sequences. To read about specific details about the implementation of the extraction please review section 3.2. The main set of features are related to the temporal domain while a secondary set of complementary features belonging to the spatial domain are used to em-

[1] For a full discussion about related work please refer to chapter 2, for feature extraction methodologies to section 3.2 for the application of such features in automatic genre categorization.

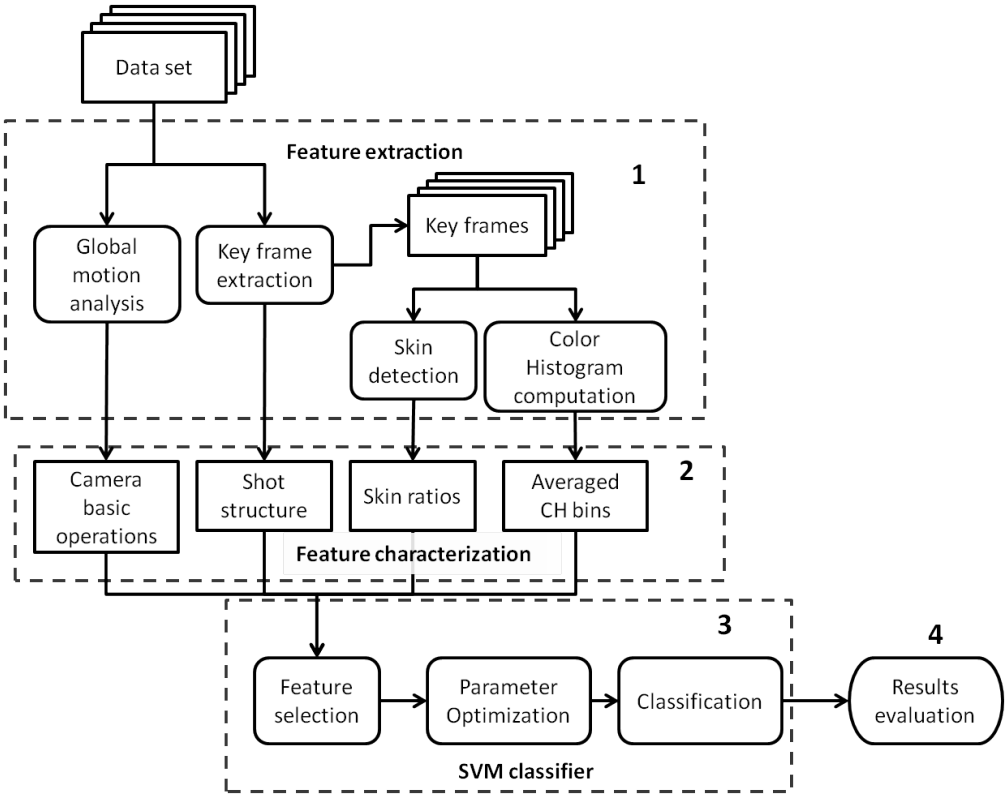

Figure 6: Proposed framework general diagram. This figure summarizes all the steps involved in our proposed methodology. There are 4 main phases. The first one is the feature extraction process for each of the four feature groups. Global motion analysis and Key frame extraction are related to the temporal domain, while skin detection and Color Histogram computation to the spatial domain. In the second phase, a vector containing condensed information regarding the feature data previously extracted is characterized separately for each of the feature groups. In the third phase, a set of experiments are carried out to compare different combinations of features, SVM algorithms and Kernels. To do so, different feature selection techniques and parameter optimization methods are applied aided by already implemented tools prior to the classification step where the combinations of elements previously mentioned will be tested. The fourth and last phase is the evaluation of the results, which implies to go back to the classification phase and repeat the experiment varying any of the mentioned components.

power the distinction of some particular cases in which the temporal features are not descriptive enough. The goals of utilizing this set of temporal and spatial features are:

a) Keep the feature extraction process as well as the classification stage as simple as possible, to decrease the computational complexity.

b) Avoid the dilution of classification accuracy when combining decisions made by multiple classifiers [2]. Further details are discussed in section 3.3.1.

c) Evaluate the discriminative power of each of the feature groups separately and also combined.

d) To explore the degree of improvement the spatial features offer to the main temporal features.

Point *a)* will be assessed in the methodology while *b)*, *c)* and *d)* will be evaluated experimentally. Four groups of features will be used to describe the content of the sequences. For each group, a vector of defined size will be used to store the representation of the video clip content as numeric data. Each of the four vectors corresponds to a group of features, and each element in a vector is a single feature within a group as shown by table 3.1. The features are structured as it follows.

Temporal features. Extracted from the video sequence as a whole, taking into account the relationship between frames.

a) **Shot Structure**. Representing the scene structure of the sequence which is intrinsically related with the video genre [13], like music video (many short shots) to adult films (assumed to be composed of medium to large shots).

b) **Camera basic movements**. The appearance ratio of each type of camera motion is closely related to the genre of the video [17]. For instance, some sports have a considerably high percentage of horizontal translation while news segments have a lot of static camera scenes.

Spatial features. Extracted separately from a set of representative frames known as key frames. Are based in the relation between pixels withing a single frame.

c) **Skin detection**. Typical adult content detection feature. Based in the idea that adult films have a great ratio of skin exposure.

d) **Color histograms**. The range of colors present in a video give clues about the intention of the film. Horror films have dark color schemes restricted in range, while comedy films

[2] For instance, to combine two decisions made by different classifiers, one being positive and the other one being negative, the resulting score would be 50% when no weights are assigned. If rather than using two intermediates classifiers the features are concatenated, the result might be more solid that a random guess depending on the complementary properties of the features

have bright color schemes with wider range of variation [15, 16]. Generally speaking, adult films do not use colors to express emotions.

As it can be seen from the previous list, all of these features are closely related to cinematic principles (being skin detection the sole exception). This principles are well known in the film literature and derive to rules known as *Film Grammar* that are extensively used by directors [15]. Although we expect adult films to present a characteristic feature signature, the precise structure of such is not clearly elucidated. Here is when machine learning algorithms come into play, for they are well known to be good to deal with problems having a subjective evaluation, where the differences in the categories is not completely clear, or the goals are ill-defined [55]. By applying these machine learning techniques to the mentioned set of features it is possible to provide a high discrimination ratio for adult content video.

Feature category	Temporal (main) features		Spatial (auxiliary) features	
Feature group	**Shot structure**	**Camera basic operations**	**Skin detection**	**HSV Color histograms**
	Average shot length (in frames)	Static camera percentage	Average skin ratio	Averaged H bins (16 elements)
	Shot length Standard Deviation	Horizontal camera movement percentage	Skin ratio standard deviation	Averaged S bins (4 elements)
			Average score	Averaged V bins (4 elements)
		Z axis (zoom, translation) camera movement percentage		H bins standard deviation (16 elements)
		Mixed (not identified) motion percentage		S bins standard deviation (4 elements)
				V bins standard deviation (4 elements)
Vector size	2 elements	5 elements	3 elements	48 elements

Table 2: Group of features used in the framework. They can be basically categorized as either temporal or spatial. In the terminology used in this work, a group of features is corresponding to a visual attribute of the video content and it's represented by a vector. Within each group of features there are single features, which correspond to an element in the vector destined to represent specific information using a numeric value.

3.1.1 Shot structure

A shot is the basic unit inside a video sequence [56], and we define it as a temporal portion of the sequence containing an action happening at a continuous time with no interruptions, similarly as [57, 16]. Shot structure is inherently connected with the genre of the video [13, 24]. For instance, music videos are composed of many short shots, while dramatic movies have fewer and longer shots [17]. Our first hypothesis is that adult videos have medium-sized to large shots mainly because the follow-up of sexual activity does not require a frequent change of scenes. The type of shot transitions, also known as editing effects [3] are also expected to be characteristic of adult

[3] Shot transitions are editing effects used to mark the end of a shot and the start of the next one. The simplest one is known as cut, in which the scene changes abruptly when moving to one frame to the next one. The ones known as fades are gradual transitions done by a progressive illumination changes. When the illumination starts from being completely dark and increases until the scene is displayed, there is a fade-in, when the opposite happens, is known as fade-out.

content, since very few abrupt transitions are used while progressive ones are more common. To accurately detect and label shot transitions is a complex problem and it is out of the scope of this work. However this and some other interesting approaches to extend this project will be discussed in Chapter 6: Future work. An algorithm detecting the shot boundaries and extracting the first frame as a representative frame (key frame) of the whole shot is used to retrieve the shot structural information. A vector with two elements is created to store two pieces of data to represent this structure. The first one is the average shot length in frames and the second is the corresponding standard deviation. The details about the shot detector implementation can be found in section 3.2.1.

3.1.2 Camera basic motion operations

When dealing with high quality cinematographic films, it is reasonable to expect many camera basic operations clearly performed, such as zoom, horizontal translation, or tilt. On other hand for sports segments a dominance in panning (which is the technical name for horizontal camera rotation) will be clearly observed [17] since this movement is often required to follow the action of a match. In such sequences also a few zooms in the critic moments are expected. Following this idea, our second hypothesis is that adult films are low-budget productions in which the professional equipment to perform professional camera movements, such as camera dollies[4], is expendable. In addition to that, zoom operations are expected to be common in opposition to vertical and horizontal translation. Many of the scenes are expected to present a nearly static camera as well. Some others will be close-ups where objects move independently, leading to a mixed motion scheme. Again, the structure of this characteristic combination of camera motion is not clearly defined. Still adult films are assumed to contain a particular motion signature which can be used for discrimination by a machine learning algorithm despise its ill definition [55]. For every frame in the sequence, the algorithm to identify camera operations will assign a label corresponding to a particular motion category. The term motion category comes from the fact that some similar types of motion are grouped together under the same label. For instance, pan and horizontal translation are both labeled as *horizontal camera movement* without making a distinction among them. There are five possible values of such labels: $\{\mathtt{static}, \mathtt{horizontal}, \mathtt{vertical}, \mathtt{z-motion}$, or $\mathtt{mixed}\}$ being $\mathtt{mixed}$ the category of a not identified pattern, assumed to be caused by a great presence of local movements[5]. The vector characterizing the *camera motion feature group* will have five elements represented using numeric values. Each of these elements is regarded as a feature. The numeric values correspond to the ratio of frames labeled as a particular motion category over the total number of frames. There will be then five ratios for each of the motion categories in the same order as listed in the lines above. Although these features allow to identify a high number of true positives for most of the cases, additional features will be required to distinguish between some challenging samples, particularly home videos, which are also poor in professional camera movement as well.

Dissolve is when two scenes are mixed together by making the ending shot progressively more transparent while the starting shot decreases in transparency, creating the effect of the first scene dissolving into the second one.

[4] A dolly is a mobile platform on top of which a camera lies. Is used to move a camera in a given direction avoiding shakes and creating an effect of purity and smoothness in the translation

[5] For a full discussion of the implementation please refer to section 3.2.2

3.1.3 Skin detection

The first of the two groups of auxiliary features is based in skin detection. The aim of adding this group of features is to remove false positives classified as adult content because of the poor professional camera movements, i.e. home video. Most of adult films show undressed people during most of the sequence and therefore can be differentiated from home made videos by recognizing the skin regions in a representative set of frames. Skin's hue is in essence the same for every ethnicity and the variations in color are due concentrations in melanin and hemoglobin [36]. Therefore a detection method robust enough to handle differences in skin color can be implemented. The so called skin detection have been extensively applied from very simple methods recognizing a fixed range of colors as skin. Some estimate a likelihood of every pixel being a skin region [18], and more complex ones involve face [21] erotogenic parts [38, 6, 22, 2] detection as well as morphological operations and adaptive skin models. These methods are discussed in details in chapter 2. The same set of key frames derived from the shot structure extractor is used as input on the basis that each of them corresponds to a shot. Three very simple features will be computed from the information extracted by the skin detector. The first one is an average score based on votes. This is computed by dividing the number of key frames having more than 50% of pixels detected as skin, by the total number of key frames. The second one is the ratio of skin pixels over the totality of pixels per image, which is later averaged for every key frame. The third one is the standar deviation corresponding to the second feature. For a discussion of the technical details used to implement the skin detector please refer to section 3.2.3.

3.1.4 Color histograms

The color range of a sequence provides important clues about its genre. For instance, in movies, a brighter color scheme is likely to correspond to a comedy while darker and dimmed tones are used in horror movies [15, 16]. In addition to this, there exist theories in art literature pointing out the properties of certain color schemes to induce an emotional state in the spectator and to create an aesthetic look in the scene [58]. Another example are violent films where there is a high incidence of red tonalities in blood and explosions [32]. Adult films are expected to ignore such theories since they do not aim to communicate any emotion, and also to have a rather natural color appearance. Again we assume adult films have a characteristic color signature that is not clearly defined. To represent this information, a color histogram in the *HSV color space* is created for each key frame. This color space is selected given its properties to represent human visual perception [58] and isolate the luminance component (value) since luminance is not interesting for the color characterization. Since we are focusing in the hue, 16 bins are used for the H channel, while the S and V channels are represented by only 4 bins each. Each of the bins is averaged for all the key frames and stored in a 24 element vector. A second 24-element vector is created to store the corresponding standard deviations for each bin and is concatenated with the previous one. The final feature vector is a 48 element in length. Details about the implementation of the color histogram computation can be found in section 3.2.4

3.2 Feature extraction process

The extraction method for each of the four feature groups is quite complex *per se* and it is done separately (however, the spatial feature computation depends on the shot detector). While the temporal features can be extracted directly from a video sequence as a whole, the spatial features are extracted individually from a set of representative key frames, each of which corresponds to a shot in the sequence. Rather than classifying each of the key frames as an image and combining the decisions using a voted-based score as done by [10], we will fuse the feature information to create a single vector for each sequence. This avoids a lot of computations and it is expected to be robust enough to improve the discrimination provided by the temporal features. The full details about the fusion method for temporal and spatial features can be found in section 3.3.1.

3.2.1 Shot structure extraction

Figure 7: Shot detection false positives due to camera jerks. From the test sequence Western, a) Frame 2868 previous to the false detection, b) Frame 2883 where false positive is detected. The movements occurs in an interval of 15 frames (0.0625 seconds). The sudden movement in such a small period of time is the main cause of the erroneous detection.

Figure 8: Shot detection false positives due to sudden movements. The false detection is caused by the quick movement of many objects simultaneously, causing a sudden change in the distribution of the color features. Most of the greenish regions in the left image are gone in the right image, including the yellow subtitles. a) Frame 386 and b) Frame 394 from the test sequence Troll. There is a distance of 8 frames between them (0.33 seconds).

Figure 9: [Shot detection false positives due to illumination changes. a) Shows frame 389 and b) frame 402 (where the change is detected) from the test sequence Humus. The shadow cast over the cookware on the table causes a sudden color change. The HSV color histograms in this case will greatly differ triggering the false detection.

An algorithm implemented by Mears[6] based on the works by [59] is used to retrieve the shot structure. The basic idea is to compute a color histogram in the HSV color space and use it to measure the degree of change along time in the frames. The algorithm builds a time-varying feature matrix representing the current N and previous $N-1$ frames by computing the color histograms in the HSV color space and then applies singular value decomposition to factorize it. Then a rank value is calculated for a frame by analyzing the number of singular values exceeding a certain threshold. When the color features in a a pair of frames changes significantly a shot transition is declared. This is also applicable for gradual transition such as *fades* and *dissolves* in which changes occur progressively, by detecting the peak of such variations. The shot detector functions well although some false positives are detected due to quick movements in the scene and sudden changes of illumination. *Figures* 7 and 8 show shot boundary detection false positives, *Figure* 9 shows false positves.

3.2.2 Identifying camera operation

An implementation was carried on in Matlab based in the motion vectors mean value analysis in sub regions as done by [29], taking a set of square sub-regions which structure is similar to [30], and grouping the camera operations into categories based on the motion direction as in [14]. Figure 10 shows a diagram of the camera basic operation feature extraction process. First the optical flow is computed using an implementation [7] of the work presented by [60]. Rather than taking each pair of successive frames, the motion vectors are obtained for each 12th frame in the sequence, taking the idea of motion persistence as in [30] on the basis that any kind of identified motion must be consistent for at least 12 frames before being declared. This also will decrease significantly the calculations required which are critical considering the high

[6]Code available at http://vision.eecs.ucf.edu/reu-web/reu2009/BenMears/sourceCode/index.htm. Benjamin Mears.
[7]Code available at http://perception.inrialpes.fr/people/Chari/myweb/Software/, implemented by Visesh Uday Kumar Chari.

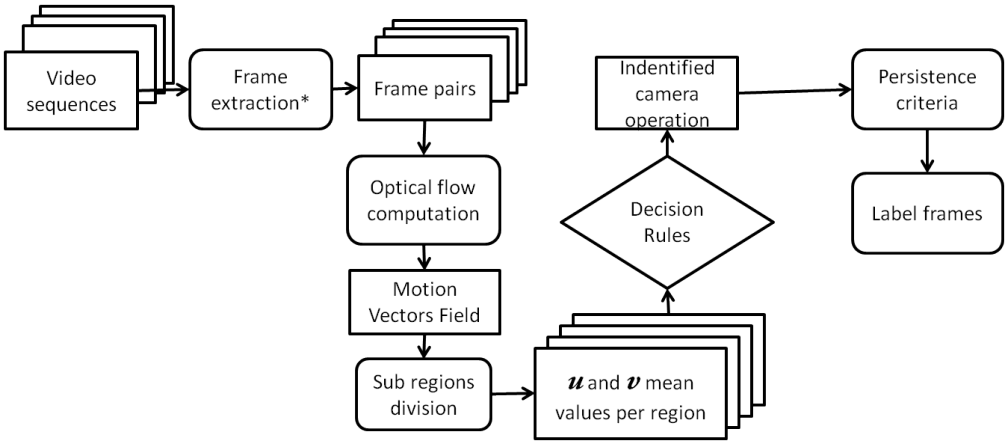

Figure 10: Camera basic operation identification diagram. This diagram summarizes the feature extraction process for this particular feature. At the end of the process (box 'label frames'), a list containing a label for each group of 12 frames is generated. This list is used to compute statistical information to characterize a vector representing the camera operation group of features as described in section 3.1.2.

computational expenses of the optical flow estimation. Next, the area of the frames is divided in 7 non-overlapping regions consisting of a central area, a horizontal and a vertical central axis forming a cross in the middle of the image, and four quadrants in each of the corners as shown in figure 11. The mean values of the horizontal and vertical component of the motion vectors, u and v respectively, are computed for each region in a similar way as done by [29]. Such values are examined in a sequential fashion as described below:

1. First the mean values are examined for every quadrant to determine if they are big enough to consider that the camera is not static. If most of them have a significant magnitude the verification continues. Otherwise the corresponding frames are labeled as $\mathtt{static}$.
2. Next, the central axes are examined. For zoom and z-translation movements, the magnitudes of the motion vectors are expected to fairly cancel each other during the computation of the mean along the vertical and horizontal central axis and therefore be smaller than a threshold. To better comprehend this notion, please refer to figure 12. Note that each of the central axes are divided in two blocks containing vectors pointing to opposite directions having similar magnitude. This property is responsible for the fair cancellation of the mean values. If the motion pattern does not match, the next point is evaluated. Otherwise the frames are labeled as $z-\mathtt{motion}$ and the verification process stops.
3. Next the regions I, II, III and IV, that we will call simply the quadrants, will be examined. Since in real world sequences a translation is rarely pure, a dominance ratio between horizontal and vertical components is calculated over the four quadrants to determine if there exists vertical or horizontal movement. The ratio is simply calculated by dividing the absolute values of u by v and vice versa. Given the fact that horizontal translation and panning are more common than vertical translation and tilt, the horizontal dominance is evaluated first. When at least 2

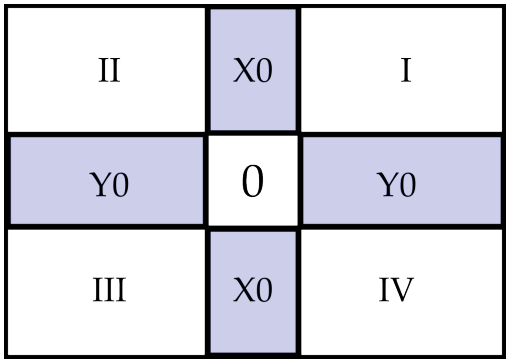

Figure 11: Image sub-regions. As the figure shows, the area of each frame is divided into 7 non-overlapping regions to identify camera basic operations using the mean values of the motion vectors. The central axes called X0 and Y0, are used to identify z-translation and zooms. The four quadrants, namely I, II, III and IV, are used to identify either horizontal or vertical movement. When the mean values in the four quadrants are below a threshold, the frames are labeled as $static$. If a clear pattern is not identified, the motion is declared as $mixed$.

quadrants exhibit a similar dominance mean value and they are above a given threshold, the corresponding motion is declared. The label used is $z-motion$.

4. If no pattern is found then the motion is assumed to be heterogeneous due to local motion caused by specific objects. The label then is $mixed$.
5. If the current group of frames has the same label as the previous one, then both entires are merged into a single one in the data structure.

This simple yet effective algorithm to classify the camera operation according to the plane of occurrence is reasonably fast, being optical flow computation the main limitation in terms of computational complexity. Modeling the motion using just three axis-based parameters is accurate enough for discriminative purposes in high-level classification according to [14]. Additionally to these three directions we add the $static$ and $mixed$ motion states. A visual representation of the typical motion vectors patterns found for the horizontal, vertical and z-motion categories is shown by figure 12. Finally, the data structure containing the relation between frames and the labels of the motion identified on them is used to compute the number of frames having each of the 5 motion category states. Each of these values is normalized by the total number of frames in the sequence and is stored in a vector of five elements that will be used as input for an SVM classifier. Please note that when summing up all the values in the characterized vector a perfect 1 will result.

3.2.3 Skin detection algorithm

The skin detection algorithm[8] is a quite simple one. Rather than pursuing a very high precision on the detection, we aim to get complementary information in the space domain to be added to the data extracted from the temporal domain (a diagram of the algorithm is shown in figure

[8]Implementation by Faizan. Code available at http://www.mathworks.com/matlabcentral/fileexchange/26849

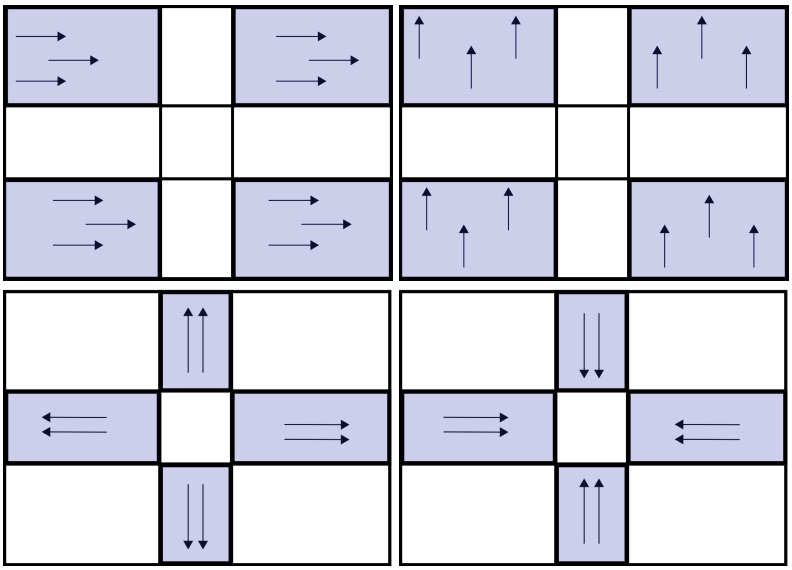

Figure 12: Camera motion typical patterns. The top row shows an schematic representation of the typical motion vectors pattern for horizontal (left) and vertical (right) camera movement. In these example there is a perfect motion to the right and up respectively. Bottom row shows the schematic patterns for zoom-in (left) and zoom out (right).

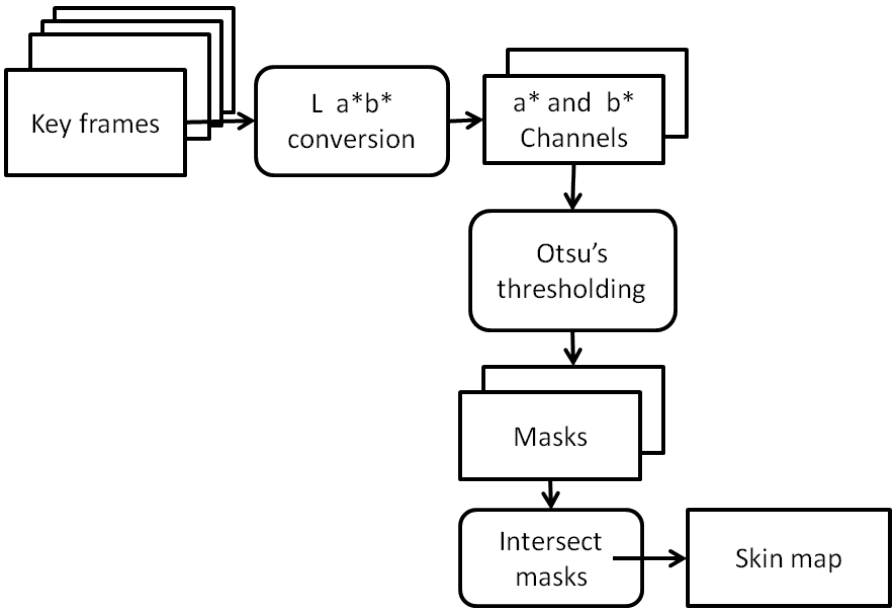

Figure 13: Skin detection algorithm diagram. This diagram summarizes the skin detection algorithm. The so called skin map is a binary mask where all the skin pixels are labeled as 1's while the non skin regions as 0's. All the masks for every keyframe are then used to computed statistical information about the skin regions that will be utilized to characterize a vector as described in section 3.1.3.

Figure 14: Generated skin map. Images on the left are the input, while the ones on the right are the resulting skin map. In the bottom row we see the method is even working with cartoons. Note that no morphological operations are applied to the masks.

13). Therefore the extraction process for this auxiliary feature must be as simple as possible to avoid computational expenses. In the shot structure extraction process described in section 3.2.1 a set of key frames, one per each shot, is generated to compute statistical information about the structure of the scenes. The same set of key frames will be used as inputs for the skin detector. Rather than classifying each image separately as done in the traditional approaches, we aim to simplify the process by avoiding the training and testing of an image-based classifier. The discriminative power will be evaluated for these features by themselves and when combined with the temporal features. In other words, instead of combining the outputs of the classifier in a vote-based score, which is known as decision fusion [54], we integrate the information of the extracted features by averaging the data in a scheme known in the literature as feature-level fusion [54]. Each of the key frames is first converted to the CIE L a*b* color space because of its correlation with visual perception and the property of separating the luminance channel from the chromatic ones. It is well known that human skin color is enclosed into a well defined range. An intuitive way to look at this property is to highlight the tendency of skin to present more reddish tonalities as opposite to greenish (represented by a* channel), and more yellowish tonalities opposing to blueish (b* channel). Therefore a fair discrimination can be achieved by thresholding the gray level images corresponding to each chromatic channel. The pixels with greater intensity will in theory correspond to red and yellow colors (the expected dominant colors for human skin) for the a* and b* channels respectively. An optimal threshold is automatically selected by maximizing the separability of the two classes (skin and non-skin) for both of the gray level

Figure 15: Skin map unsatisfactory results. On the left the original images, on the right the binary skin maps. The skin detector is based on the assumption than skin colors are reddish and yellowish. Since an automatic threshold is selected in terms of the separability of the gray level images for each channel, when red or yellow regions having high chroma appear in the image, the threshold will be determined taking those regions as the maxima and will wrongly assume those correspond to skin. In the top row, the highly intense red colors in the blouse and sheets are incorrectly assumed to be skin. This will affect the automatic thresholding process causing to miss the actual skin regions. This can be solved by dimming colors with high chroma in the upper range. The bottom row shows a table that has a very similar color to the skin of the subject in the image. To solve this problem adaptive color skin models and face detection can be used. However the complexity of the algorithm will increase substantially when handling such issues. The gain in accuracy of skin detection might not compensate for the increase in processing time.

images following Otsu method [61]. Then both masks are intersected, selecting only those areas detected as skin in both. The resulting binary image is a representation of the skin regions present in the image, this is known as skin map. Figure 14 shows an example of skin maps. The resulting map is straight-forward used to compute the skin statistical information that will be used to characterize the vector as described in section 3.1.3. No morphological operations are applied to the map. This method is very fast and reasonably accurate but presents some inconveniences. Those are shown in figure 15

3.2.4 HSV Color histograms computation

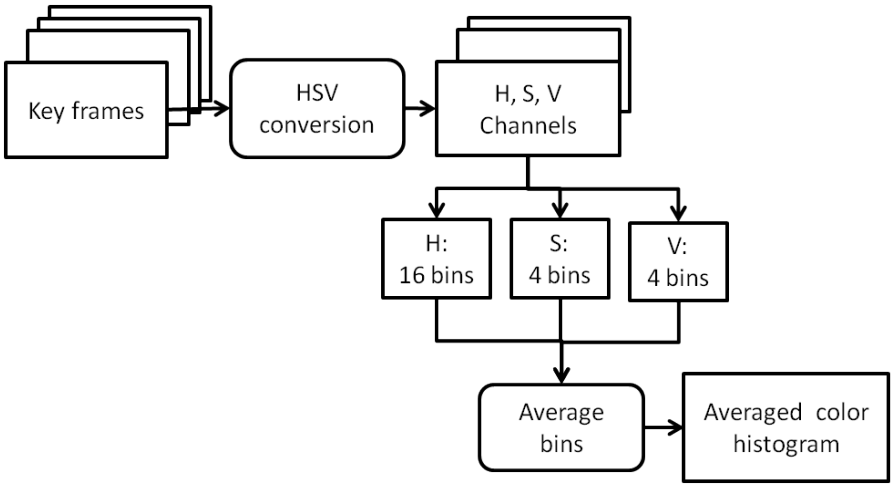

Figure 16: Color histogram computation diagram. This diagram shows the steps followed to compute the HSV color histograms. Each of the bins of the averaged color histogram and the corresponding standard deviation values are stored in a vector of 48 elements as described in section 3.1.4.

A diagram for the algorithm is shown in figure 16. This very simple algorithm just converts the image to the HSV color space, which is a very common choice for color histograms computation since it separates the brightness from the saturation and the hue. The hue in this case is considered as more important since we expect adult films to have a characteristic print of color schemes. So a histogram is created using 16 bins for Hue while the saturation and value (brightness) are represented only using 4 bin per each as in [23]. Each of the bins is averaged to produce the characteristic vector. This feature group is the simplest of all and is very quickly computed. The goal of using it is merely to add auxiliary discriminative information for the main features extracted in the temporal domain. The use of more sophisticated techniques for color characterization and the different methods to fuse these features to maximize discrimination are subjects of future work. This and some other interesting research directions are discussed in chapter 6.

3.3 Classification

As mentioned before, the objective of the current approach is to demonstrate that a high accuracy in discrimination of adult-content can be achieved by applying SVM classifiers altogether with

the set of features mentioned in section 3.1. To achieve this, we will apply and evaluate the performance of the more successful classifiers for visual recognition task and, in particular, for adult content identification reported by the literature. However, since our approach uses features that have not been utilized for this particular purpose so far, we expect the performance to be quite different as the reported by the related work. Moreover, rather than following the decision fusion normally applied in image classification, we follow the feature fusion scheme by extracting the spatial features from each of the key frames and averaging the corresponding numerical values. This way we use a joint feature representation for the whole set of key frames that will be processed by a single classifier in the final decision. Rather that implementing, improving or proposing new machine learning techniques or algorithms, we simply make use of tools that have those algorithms already implemented. These tools will be used to evaluate the performance of the different group of features by themselves and in different combinations as well as the existing machine learning algorithms. Different settings of three main elements impact significantly the the performance of the classifier [62], these are tested in the third phase of our approach. *1) The selected set of* ***features***, *2) the* ***SVM algorithm***, and *3)* ***Kernel*** *functions*. Prior to the tests, some *feature selection* heuristics will be applied as well as a tool for feature selection. Also a *parameter optimization* process aided by tools will be performed to determine the best parametric configuration of the SVM algorithm and Kernel. This section will discuss just the methodology and tools used. For a discussion about both the specific parameters and configurations used, as well as the obtained results, please refer to chapter 4.

Classification criteria

Of course what is considered offensive for a certain audience is quite subjective and depends on many factors, mainly on cultural background. For instance, in some countries such as Sweden, access to pornography is not restricted to any ages, while in some other cultures nudity is considered offensive. Rather than determining the degree of erotic content as done by [3], a simple criteria will be established. Any scene containing exposure of erotogenic parts, that is mainly female nipples and genitals of both genders, will be regarded as adult material. Needless to say that any other sequences containing the elements just mentioned plus sexual activity will fall into the same category.

3.3.1 Fusion method

As mentioned before, a single vector representing shot structure and camera basic operations can be extracted from the video sequence as a whole. For the spatial features this is not possible since there is a set of many key frames associated with a single sequence as the source of the information. The usual way to deal with this issue is to classify each keyframe as an image and compute a vote-based score based on the decisions as done by [10]. This is known as decision fusion [54, 63] and requires an image-based classifier, a specific image training set, individual classification per keyframe, and a method to combine the decisions. We avoid this scheme because of the obvious negative impact in computational complexity both in terms of memory and time. Instead we combine the data in a feature fusion scheme [54, 63] by averaging the numeric values representing the features extracted for every keyframe in a given sequence. At this point the feature vectors of both temporal and spatial information are of the same order. They can be

either used individually, or combined with each other by simple concatenation to be used as the inputs for a single SVM classifier that will take a unique and final decision. The advantages of this approach are *1)* ***Reduce the computation complexity***, and *2)* ***evaluate***, or track down, ***the most discriminative features*** rather than diluting the decision of the classifiers. This will lead to a more robust mechanism in which it is possible to explore the complementarity properties of different combinations of features to improve their correlation with the class information. To clarify the last point, imagine there is an adult sequence with body painting where the skin detection feature will be fooled. The camera motion information however will detect it as an adult film. If we were to combine the scores of two classifiers, one utilizing the skin features and another one the camera motion, the final conclusion would be that the sequence has 50% of probabilities of being pornographic. This score would either need to be passed to another classifier to make a final decision, or processed by a rule based model, requiring additional processing time. This would almost equal to a random guess. In this toy example, the feature with the greatest contribution will be 'diluted' in the one that does not contribute. In adition to the previous example, the temporal features are expected to generate the most abundant amount of discriminative information while the spatial ones are aimed to help to decrease the false positive ratio by adding new information bypassed by the main features. As mentioned before, the poorly professional camera movements in home videos are already considered as a challenge and therefore the spatial features are added to deal with those. The methodology used for selecting a subset of all the extracted features will be discussed in section 3.3.3

3.3.2 Classifiers

As extensively mentioned before, the implementation or improvement of machine learning algorithms is out of the scope of this project. For this reasons we selected a software package based in Java [9] that has many algorithms and tools conveniently implemented. Weka [10] makes possible to call the functions from Java code or use them directly with the included graphic user interface. Support Vector Machines have been widely used in recent years research for pattern classification. SVMs are considered a vanguardist tool for visual recognition tasks [10] and are known for being suitable for problems with relatively few samples but having large feature vectors [18, 56, 20]. We are particularly interested in the derived applications for recognition tasks, particularly those involving visual aspects such as video genre classification [17, 46, 47], and more specifically for adult content detection in video [10, 7, 53, 9]. In our approach we will compare two different SVM algorithms, the first one, LibSVM[11] [64], is referenced for applications such as content-based filtering [48], spam identification [65], video categorization [46], and adult content detection [8], visual concept categorization [49], and camera motion classification [31]. A special LibSVM wrapper[12] for integrating LibSVM to Weka is utilized. The second algorithm is known as Sequential Minimal Optimization for training an SVM, which will be referred to as SMO in the future lines, based in John Platt's algorithm [66] and implementing the improvements proposed by Keerthi [67]. The main contribution of SMO algorithm is to deal

[9] http://www.java.com/en/download/faq/whatis_java.xml
[10] http://www.cs.waikato.ac.nz/ml/weka/
[11] http://www.csie.ntu.edu.tw/ cjlin/libsvm/
[12] Implemented by Yasser EL-Manzalawy in 2005. URL: http://www.cs.iastate.edu/ yasser/wlsvm/

with the optimization problem that arises when an SVM is trained. This impacts in reducing the training time and the memory required which is the main drawback of SVMs in general [68]. As done by some other SVM training algorithms, SMO breaks the quadratic programming optimization problem into smaller problems, but unlike the rest of the algorithms, it chooses the smallest possible set of problems. This makes possible for the algorithm to handle very large training sets with an optimal use of both memory and processing time [66]. This algorithm is included in the default Weka's library and is widely referenced for more broad classification tasks such as content-based spam filtering [69], Internet traffic classification [70], gene micro array categorization for cancer diagnosis [71], text classification [68], and art painting categorization by artistic genre [72], to mention a few. For libSVM, we will compare the linear, polynomial, radial basis function (RBF), and sigmoid kernels. For SMO, the RBF, Pearson Universal, Polynomial and Normalized Polynomial kernels will be used.

3.3.3 Feature subset selection methods

As mentioned early, one of the objectives of the present work is to evaluate the discriminative properties of each of the feature groups by themselves or when combined with others. To select and evaluate a subset of features we make use of three approaches. *1)* Selection based on feature groups combinations, *2)* Manual selection based on estimated complementarity, and *3)* attributeSelection tool. The first two are manual, while the third one is quasi-automatic. The *first* heuristic aims to compare the contribution for each set of features and evidently consists on testing each group separately first. Then different combinations of features groups are used in pairs and triplets, and finally the four groups (a single set containing all the features). The second one, *manual selection*, is done by following the same reasonings of expected discriminatory properties of each of the features plus complementarity between them as discussed in section 3.1. In addition to this, a series of plots produced by Weka are examined to aid the decision [13][14]. The plots for all the features with the experimental data are available as an appendix in chapter 7. The third one, *attributeSelection*, is a flexible supervised feature filter with two parameters, the first one is an evaluator, that basically determines the criteria a set will be assessed with, and the second one is search method, which basically means the order in which the combinations will be explored. The results of all the experiments are discussed in chapter 4.

3.3.4 Parameter optimization methods and tools

Both the SVM algorithms and the Kernel functions are formulated with some parameters that need to be tuned-up in order to maximize the performance. This is usually a tedious process because the simplest way to do it is by brute force. Luckily Weka provides two useful functions that allows to make this brute-force process quasi-automatic. The first tool is called *CVParameterSelection*. This tool optimizes a set of parameters by trying different values specified as a range. The parameter producing the best accuracy is returned. *GridSearch* functions in a similar fashion

[13] For each individual feature, a plot of the distribution of the numeric values and their relation with both of the class labels is generated. These plots give an idea on the influence the numeric value of the feature has over the class label. The intuition would be, the more variable the distribution is, the more valuable the feature is.

[14] Also it is possible to generate an scatter plot combining each of the features against each other in a grid scheme to visualize the separability of classes such pairs would offer. The most separable the classes are for two given features, the more complementary they are. However, from the scatter plot is very hard to predict the complementarity of multiple features

except that it tests the combination of two parameters at a time. The first inconvenience regarding this two tools is that they are not completely reliable, meaning that the predicted outputs for a set of parameters does not always match with the actual ones. However these tools can be useful as a starting point. The given parameters will have to be tried in a real classification and some manual brute force exploring will be required.

The next chapter describes the experimental setup and the results.

4 Experiments and results

4.1 Experimental setup

A data set consisting 90 clips, 48 adult videos and 42 harmless clips, is used to validate our methodology. For each sequence in the dataset, four group of features are extracted separately: *1) Shot structure*, *2) Camera basic operations*, *3) Skin exposure* and *4) Color histograms*. The last two group of features are extracted using the key frame set derived from 1). A full discussion about the features properties can be found in section3.1, while the extraction methodologies in section 3.2. Being *camera motion* feature group the most demanding in computational complexity, the feature extraction process was done using two types of computers within a lab. *Type A* with an Intel Core i7 processor, 870 @ 2.93GHz, 2.93GHz, and 4GB RAM. *Type B* having Intel Core 2 Duo processor, E6750 @ 2.66GHz, 2.66GHz and 8GB RAM. After this a set of experiments are run in Weka using different combination of *a) Features*, *b) SVM classifiers* and *c) kernels*. Before every experiment is done all the kernel parameters are optimized using the methods described in 3.3.4. The features used are single feature groups followed by pairs, triplets and a quartet(every group), as well as the resulting sets from applying manual selection and attributeSelection tool in Weka. The compared SVM classifiers are libSVM, using linear, polynomial, Gaussian radial basis function (RBF), and sigmoid kernels, and SMO, utilizing RBF, Pearson Universal, Polynomial and Normalized Polynomial kernels. For each experiment a cross-fold validation with leave-one-out scheme is followed. In such scheme, the entire sample collection except one element is used as a training set, the remaining single element will be used as test set. The operation is repeated for every element of the collection to calculate the classification accuracy. The leave-one-out scheme is more realistic (less optimistic) than the ones randomly splitting the dataset in a fixed proportion, e.g. 60% training, 40% testing, but comes with a higher computational cost. Nevertheless, given then size of the dataset, this scheme is recommended. On the other hand, the cross validation can prevent the over-fitting problem [73] and it is also recommended when parameter optimization functions such as *grid search* are being performed on medium-sized problems [64]. To identify the missclasisfied items, the filter *AddID* that adds an aditional ID feature is applied to the data set. When the classifier is runned, another filter is set so the ID feature is ignored, since it could be used by the classifier to learn from. When this ID feature is not removed the accuracy gets inflated, meaning that a higher but fake value is achieved.

4.1.1 Data set

For some computer vision tasks, such as object recognition, public datasets are available, like the Flower[1] or the PASCAL Visual Object Classes[2] . It is fair to say that at the moment a standardized collection of video for evaluating adult-content detection applications does not exist.

[1]The flower data set was created by Maria-Elena Nilsback and Andrew Zisserman for object recognition purposes. Available at http://www.robots.ox.ac.uk/ vgg/data/flowers/

[2]Available at http://pascallin.ecs.soton.ac.uk/challenges/VOC/

Sub-genre	Number of clips	Length
Cartoon	7	36:07
Erotic	3	09:36
Home video	3	10:35
Movie	8	40:00
Music video	6	25:23
News	4	23:31
Race	3	14:32
Sports	8	49:15

Table 3: **Harmless dataset sub genres**. The objective of having this data set further divided into sub-genres is not to classify each of them but rather to create a data set containing a representative mixture of non-adult genres. The categories *a)Erotic* and *b)Home video* are specially challenging. Sub category *a)* because of the thin line between the high level concepts of erotic and pornographic, having both similar low level features for all of the extracted feature groups. Sub category *b)* has very poor professional camera movement and being this the strongest feature for adult content recognition, this is very likely to produce false positives.

Furthermore researchers are not always willing to share their own data because of copyrights issues. Another particular problem in the context of our methodology is that most of the existing datasets are simply inadequate for extracting shot structure since they use very short clips of around 10 or 20 seconds long. Therefore creating a dataset was mandatory. It consists of 48 adult videos ranging from 1 to 7 minutes in length, being the majority of clips around 5 minutes long, and 42 harmless videos downloaded from youtube[3] ranging from 2 to 9 minutes, being most of them around 5 minutes long as well. Table 8 and 9 in the Appendix 7 show the dataset length details. Every video has been processed using *ffmpeg*[4] to change the resolution to 320x240 and discard the audio. The total length of the adult set is 03:46:30 hours while the non-offensive set has 03:29:02 hours of duration. The non-offensive subset is further divided into several categories in order to have a representative mixture of different genres as shown by table 3. Note that the goal of the classifier is not to further identify these sub-categories. The first observation is the fair length in processed video with a total of 07:15:32, having more than 3 hour per category. Some approaches like [74] use 1 hour of recording per each of the five categories defined within the proposed approach and each sequence with a length of around 5 minutes.

4.1.2 Parameter optimization

The main parameter to be optimized for both SVMs classifiers used in the experiments is the parameter C, referenced as complexity parameter for SMO and cost for libSVM in Weka documentation. The cost parameter in libSVM controls the trade-off between the misclassification and over fitting. The wider is the acceptable margin of error, the more general the model might be. The more narrow the margin, the more specific the model is suiting the particular dataset, which might imply over-fitting and therefore convey to a model non-applicable to a different dataset. Similarly, C in SMO controls how smooth the margins are. This is related to the number of instances that will be used as support vector machines to build-up the boundaries that will separate the classes in the transformed space. The linear kernels can be optimized just in terms

[3] http://www.youtube.com
[4] Video coding platform. URL: http://ffmpeg.org/

of C, however some other kernels, such as the RBF or Sigmoid, are very sensitive to another parameter known as γ. In this case the optimal value of C depends on the value of γ and vice versa. Therefore a search method comparing simultaneously different combinations of values for this pair of parameters is required. *Grid search* will be used to pairwise evaluate only the most critical parameters for each classifier and Kernel. The parameter pairs are (C, γ) for the RBF and Sigmoid kernels, and $(C, degree)$ for the Polynomial Kernel in libSVM. The RBF kernel is used for both libSVM and SMO, while the Sigmoid kernel is just used for libSVM. However the implementation of the *Grid Search* method in Weka presents the major inconvenience of being unable to handle the classifiers' nested options. In the case of SMO, for the Person Universal (PUK), Polynomial and Normalized Polynomial kernels, the parameter optimization was focused only on the complexity parameter C using *CVParameterSelection* tool, given that SMO is very sensitive to this one. For the RBF kernel *Grid Search* was used to optimize (C, γ). The ω and σ parameters of PUK where not optimized.

4.1.3 Feature selection

After the best classifier and kernel in terms of accuracy is experimentally found, a set of tests are executed to compare different feature sets. The four feature groups combined are evaluated and then each of the feature groups individually followed by combinations in pairs and triplets. Finally two more sets based on manual and automatic feature selection are experimentally tested. The manual selection is described in section 3.3.3. For the automatic feature selection, Weka's *attributeSelection* filter is used, with *a) CfsSubsetEval* as the evaluator and *b) BestFirst* as the search method. *a)* stands for *Correlation-based feature selection for machine learning* [75] and basically considers just the predictive abilities of an specific feature by itself as well as the the redundancy with previously selected features. Setting *b)* Explores the space of features by hill-climbing in a greedy fashion plus a backtracking mechanism. Table 4 shows the selected and non-selected features.

Feature Group	**Selected features**	**Non-selected features**
Shot structure	Shot length standard deviation	Average shot length
Camera motion	Mixed motion ratio	
	Horizontal motion percentage	Static motion percentage
	Vertical motion percentage	Z-motion percentage
Skin detection	Average score	Skin percentage standard deviation
	Skin percentage mean	
Color histogram	Averaged bins: 9 19 20 21 23	Rest of the bins
	Standard deviation for bins: 20, 21	

Table 4: **Automatically selected feature set using *attributeSelection* filter**. There are a total of *13 features* derived from the 4 groups. The first observation is that surprisingly the standard deviation of shot length was selected rather than the mean. From the camera motion features *static motion* and *z-motion* are discarded. This makes perfect sense since *z-motion* is the most weakly identified operation because the zoom or z-translation has to be nearly pure (not combined with any other movement at all), meaning that the object of attention has to be precisely at the center of the screen, otherwise the movement will not be identified. Skin percentage (the ratio of skin pixels over all the pixels) standard deviation is expected to contribute less than the mean, therefore it is excluded. Finally just a few of the averaged bins of the color histograms (as well as a couple of corresponding standard deviations) were discriminative enough to be automatically selected.

4.2 Results

The experimental part involved the repetition of many tests as the results converged to a higher accuracy. This means that when a set of features, a classifier, or a kernel yielding to a high classification accuracy was discovered, the experiments had to be repeated to compare the variations. The criteria to assess the performance of the classifier is, in this order, *a) Accuracy*, *b) the number of used features*, given that selecting a few features with high discriminative rate might relegate the rest as not very useful and save processing time in the classification phase, *c) number of feature groups*, because it is possible to avoid computing the features of all of those feature groups not contributing, and finally *d) number of parameters to optimize*, because it requires time to estimate them in addition to the complexity added to the process. SMO achieved the highest accuracy with a value of **94.44%** when the *automatically selected* set of features are used altogether with the *Normalized Polynomial* kenel after parameter optimization. Being SMO the algorithm with best performance, a more exhaustive set of experiments was performed using that classifier. *LibSVM* achieved a slightly smaller maximum accuracy of **92.22%** using just the *camera motion* features together with either *RBF* of *linear kernel*. In this particular case, the linear kernel is considered to be superior to RBF since just the cost parameter C needs to be optimized. Being libSVM results less accurate than those of SMO, a more limited set of tests were conducted for libSVM. For *libSVM*, table 5 shows the parameter values, set of features, and kernel used for each experiment as well as the achieved accuracy, number of features, number of feature groups involved and number of parameters optimized. For *SMO*, table 6 shows the experiments for evaluating the feature sets, while table 7 shows the ones for assessing the kernel functions. In addition to a better accuracy, SMO has a superior performance in terms of time. The time taken for the classification phase is less than one minute per experiment. For the feature extraction process, the time is less than one minute per sample in the case of skin detection and color histogram computations, while shot detection takes around 3 minutes per sample. Camera motion identification is the most time consuming extraction process, taking around 2 hours per minute of video using a type B[5] computer . This is caused by the optical flow computation which is well known for being painfully slow. An alternative to solve this issue is proposed within section 6.1. Figures 18 19 show a comparison of SMO ROC curves corresponding to the results sumarized by the tables mentioned above. A comparison of the ROC curves for most successful combination of feature set and kernels for each SMO and libSVM classifiers are shown by figure17.

Existing solutions comparison

Although the adult-content detection for images has been fairly explored, not many solutions specifically focused on video exist among the literature. The lack of standardized databases mainly derived from copyright issues and the different measures used to represent the performance of the methodologies makes it difficult to establish a completely objective comparison criteria. Also it is important to keep in mind that in the existing solutions there are many variations in the used features, fusion strategies, used classifiers, among others, that might impact the comparability of the obtained data. We briefly mention the results of all of those solutions focusing in adult-content video detection. Endeshaw [12] identifies adult content by means of

[5]Computer type B having Intel Core 2 Duo processor, E6750 @ 2.66GHz, 2.66GHz and 8GB RAM.

Kernel	Feature group	Cost param	Degree param	γ param	Accuracy	**Features #**	**Feature Group #**	**Optimized Param #**
Linear	Cam	**31**	3	0	**92.22%**	5	1	1
Polynomial	Cam	1	**3**	**2**	90.00%	5	1	2
RBF	Cam	**40**	3	**2**	92.22%	5	1	2
Sigmoid	Cam	**16**	3	**-2**	63.33%	5	1	2
Polynomial	Auto Selected	16	3	2	76.67%	13	4	0
RBF	Auto Selected	**16**	3	**2**	53.33%	13	4	2
Sigmoid	Auto Selected	**16**	3	**-2**	38.89%	13	4	2

Table 5: textbfLibSVM classifier experiments results. The optimized parameters are shown in bold. Note that for the Polynomial Kernel and the *automatically selected* set of features none of the parameters are in bold since they were tunned-up manually because of of the unsatisfactory results obtained by *grid search* tool. The linear kernel could not be evaluated for the *automatically selected* set of features since Weka got frozen over days. As libSVM resulted in a slightly smaller accuracy than SMO for the compared test cases, a more limited set of experiments were executed using this algorithm. The four kernel functions were tested using *a)* the more discriminative feature group *Camera motion* and *b)* the *Automatically selected* feature set. For *a)* the results are slightly worse than those of SMO, but for *b)* the accuracy is significantly smaller. The rest of the settings (not specified in this table) are shown by table 13 in the Appendix.

Kernel	**Feature set**	**C param**	**E param**	**γ param**	**Accuracy**	**Optimized Param #**
NormPoly	Auto Selected	**8**	2	N/A	**94.44%**	1
Polynomial	Auto Selected	**3**	1	N/A	92.22%	1
RBF	Auto Selected	**5**	N/A	**1**	90.00%	2
PUK	Auto Selected	**1**	N/A	N/A	91.11%	1

Table 6: **SMO classifier kernel comparison experiments results**. All of the assessments were done using the *Automatically Selected* set of features shown by table 4 since it is the one yielding to a higher accuracy (when using SMO with a Normalized Polynomial kernel). This set comprises 13 features from all the 4 groups. Optimized parameters are shown in bold. These correspond to the complexity parameter C for every kernel and the combination (C, γ) for the RFB kernel. In the case of the Polynomial and Normalized polynomial kernels the parameter E, the exponent, was not optimized even though a couple of tests changing the value were performed. In the case of the Pearson Universal Kernel (PUK) the parameters ω and σ were not optimized and the value used for both is 1. The rest of the parameters and settings, used commonly for every test, are shown in table 14 in the Appendix

Feature group	Complexity param	Accuracy	Features #	Feature Group #
All groups	**3**	92.22%	58	4
Shot	**5**	55.56%	2	1
Cam	**7**	93.33%	5	1
Skin	**1**	80.00%	3	1
Color	**6**	76.67%	48	1
Shot+Cam	**7**	92.22%	7	2
Skin+Color	**2**	86.67%	51	2
Cam+Skin	**6**	87.78%	8	2
Cam+Color	**4**	87.78%	53	2
Shot+Cam+Color	**6**	90.00%	55	3
Shot+Cam+Skin	**2**	90.00%	10	3
Cam+Skin+Color	**2**	90.00%	56	3
Shot+Skin+Color	**2**	86.67%	53	3
Manual	**5**	91.11%	28	4
Auto Selected	**8**	**94.44%**	13	4

Table 7: **SMO classifier feature selection experiments results**. All of the experiments were done using the kernel showing the best performance: The *Normalized Polynomial Kernel* for which γ parameter is not required. The $exponent$ parameter value is 2 for every case. The rest of the settings are shown by table 14 in the Appendix.

repetitive motion detection and using a restricted set of videos reports a true positive probability greater than 85%. Jansohn [10] achieves an equal error rate of 6.04% combining BOVW[6] and motion histograms altogether with SVMs and decision fusion for the image-based features. Lee [7] employs just spatial features based on skin likelihood and HSV color histograms as well , achieving a maximum accuracy of 91.48%. Lopes [8] uses solely BOVW features based in SIFT and HueSIFT with a linear SVM obtaining an accuracy of 93.20%. Chang [9] uses skin, texture, and shape image-based features on keyframes and moments matching reporting a true positive ratio of 96.5%. Bouirouga2011 [18] exploits skin distribution features and SVMs using RBF kernel reporting a quite impressive accuracy of 97%. Finally, based on the accuracy of 94.44% achieved by our methodology, we can affirm it outperforms most of the existing frameworks while it is fairly close to those surpassing our possitive results. Taking into account the many possible refinements, we can ensure our approach is yet even more promising.

[6] Bad of visual words.

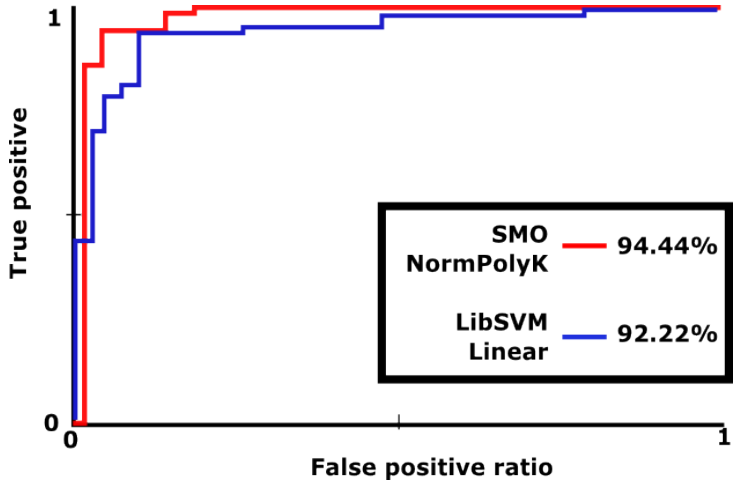

Figure 17: **SMO versus libSVM ROC comparison curves**. Comparison between the most successful combination of kernels and features for every classification algorithm. In the case of SMO the *automatically selected* feature set is used with the Normalized Polynomial kernel. For libSVM the camera motion feature group and linear kernel are used. Note the small difference between them of just 2.22%

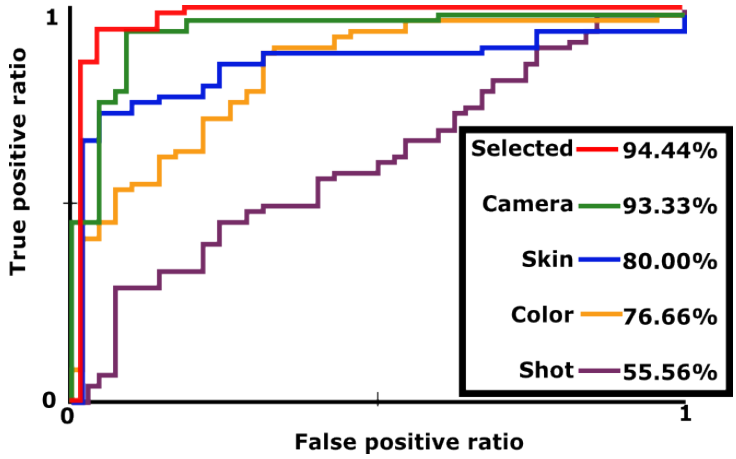

Figure 18: **SMO feature comparison ROC curves**. Comparison between each of the four feature groups individually plus the *automatically selected* feature set which is the one yielding the best accuracy. All of them using Normalized Polynomial kernel corresponding to the results shown by table 6.

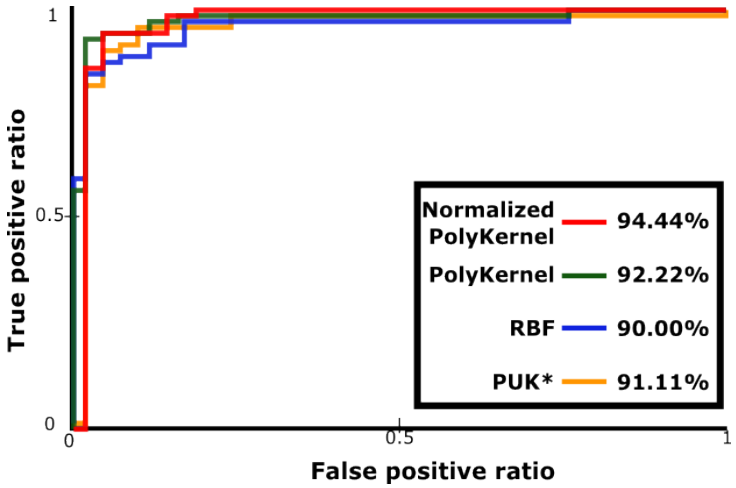

Figure 19: **SMO kernel comparison ROC curves**. Comparison between kernels using the *automatically selected* feature set. This curves correspond to the results shown by table 7

Figure 20: **Adult content detection false positives**. Sequences were incorrectly classified as adult-content being the true class harmless. This misclassification comes from the most successful approach using *automatically selected* feature set with SMO and Normalized Polynomial kernel. Both of the clips are part of the challenging categories, erotic (left) and home video (right). The sequence on the left has very similar characteristics to adult content for it has a *a)* big ratio of skin exposure and *b)* poorly professional camera movements. The one on the right exhibits just *b)*.

Figure 21: **Adult content detection false negatives**. Sequences were incorrectly classified as harmless being the true class adult-content. This misclassification occurs when using the most successful approach comprising the *automatically selected* feature set with SMO and Normalized Polynomial kernel. The sequence on the left has a camera nearly static most of the time, feature which was specifically removed from the *automatically selected* feature set. Also the undressing part comes nearly in the end of the sequence. The one in the middle is a clip extracted from and adult video using *ffmpeg*. Even though it is part of a pornographic video, the clip itself does not comply with the criteria for being considered as adult (technically speaking is harmless based on the content) and basically shows a completely dressed girl talking. The one on the right was originally part of the harmless dataset since it shows a girl performing some sensual dancing. However there are a couple of frames in which she exposes fully her erotogenic parts. Note that this last clip was difficult to classify even manually.

5 Discussion

The main objective of this methodology is to demonstrate that a high accuracy can be achieved by using temporal-based features aided by spatial-based features altogether with SVM classifiers. In a first phase four group of features are extracted *a) shot structure* and *b) camera motion* as the main features in the temporal domain, as well as *c) skin detection* and *d)HSV color histograms* as complementary features in the spatial domain. In the next phase the information extracted is used to characterize four vectors with different number of features each corresponding to a group. The third phase of the approach is to use different combinations of the features to evaluate the performance of two SVM algorithms, namely SMO and libSVM. In this process the parameters are optimized and different kernel functions assessed. Please refer to section 3 for more details. In accordance to the discussion presented in 3.3, we determined experimentally the best combination of three main elements impacting the performance of the classifier: the used set of features, the SVM algorithm, and the kernel function. Based on the results discussed in the previous chapter, the best combination found by our assessments is the *automatically selected* set of features[1], SMO SVM algorithm and the Normalized Polynomial kernel, producing an accuracy of **94.4%**. As it can be observed from the results obtained, the temporal features based on cinematic principles are highly discriminative for adult-content detection, being camera motion the most contributing feature group when used individually, yielding to an accuracy of **93.33%** when SMO algorithm is utilized altogheter with the Normalized Polynomial Kernel fuction. It is seen clearly that even though there exists an optimal combination $(\mathtt{features}, \mathtt{classifier}, \mathtt{kernel})$ the differences in accuracy when the parameters are properly tunned-up is not extremely divergent, being in some cases quite similar (e.g. when comparing the performance of libSVM and SMO using the camera motion feature group alone and different kernels). This metodologhy is a fair first step, having already quite good results, towards a very promising solution having many refinements and expansion posibilities.

SVM algorithms and kernels

The algorithm with the best performance in general is SMO, which in most of the cases outperforms libSVM by a tight difference of around 1%. As said before, the kernel producing a higher accuracy for SMO is the Normalized Polynomial kernel. In the case of libSVM the best performance is achieved when just the *camera motion* features are used altogheter with the linear kernel with an accuracy of 92.22%. However the tests applied for libSVM were not exhaustive enough to conclude which kernel is the best in general. This is due to the fact that libSVM performs very poorly in execution time when the number of features is more than 5. Taking into account this last detail, the fact that most of SMO kernels can be optimized in terms of a single

[1]This data set is generated by executing Weka's *attributeSelection* as discussed in section 4.1.3 and consists of a selection of 13 features from all the four groups based on the individual predictive properties of each. A table showing this feature set details is available 4

parameter while libSVM requires often 2 or even more simultaneously, and of course the better accuracy, we can soundly conclude that SMO general performance in our experiments was quite superior.

Features

The temporal based features are *a)* highly discriminative and *b)* fairly fast to compute[2]. Since they are extracted straight from the video sequence as a whole, no fusion method is required to characterize a single vector corresponding to a video sample. This is not the case of the spatial features, which need to be combined either using decision or feature fusion before mapping them to a single clip because they are extracted individually from each of the many key frames within a sequence. The more discriminative group of features is *camera motion* and the second one is *skin detection* with an accuracy of 80.00%[3] despise is considered by the literature as a poor feature. However the complementarity of the two strongest features is low since combined they produce just 87.78% in accuracy, which is considerably lower than the 93.33% achieved just by the camera motion feature group. As discussed in section 6.1, all of the features can be further refined while keeping the computational complexity from increasing dramatically.

Misclassified instances

From figure20 is clearly seen that the two false positives come from the challenging subcategories within the harmless dataset. Both sequences having poorly professional camera movements (which happen to be the most discriminative feature group) is the main cause of the misclassification. The false negatives shown by figure 21 were indeed very challenging adult sequences. One of them has an undressing sub sequence near to the end, another exposes female genitalia for a few seconds, and the third one is a clip extracted from an adult video that does not contain any nudity or sexual activity (in other words is harmless technically speaking although it was placed in the adult dataset because of its origins). This misclassification correspond almost exactly with our predictions done after choosing the feature groups and creating the datasets. There are a number of improvements that can be applied to our methodology to fairly decrease the number of wrong classifications. These are discussed in chapter 6: Future work.

[2]The camera motion identification, however, depends on optical flow computation which is well know for being painfully slow. There are some alternatives to obtain the motion vectors in a quick way but were not implemented since those are out of the scope of the project.

[3]Using SMO and Normalized Polynomial Kenel

6 Future work

Adult-content identification is a complex problem since it implies high level conceptual categorization starting from low level features. To achieve this, it is necessary to use techniques from different disciplines such as image processing, colorimetry, machine learning and specifically pattern recognition. Creating a framework that demands the solution of many challenging problems, such as editing effects identification, shot boundary detection, camera motion labeling, among others, leads to define a stop criteria since, unfortunately, solving each of those sub-problems is simply not feasible given the limited time. The feature extraction process was coded using Matlab[1]. To improve the execution time of the feature extraction algorithms is possible to use a programming language that runs faster than Matlab. As a result, this section suggests the different areas of improvement and future direction to explore the promising path this work leads to. The following list shows a summary of the possible improvements.

— *Feature extraction*.

- **Shot structure.** Identify editing effects within the shot boundary such as fade, dissolve, cuts or wipes [51].
- **Motion vectors extraction.** Make use of the MPEG coding scheme to extract the motion vectors directly instead of computing the optical flow for a pair of frames [30].
- **Camera motion.** Modify the algorithm to deal with complex zooms and z-translations where the object of interest is not in the precise center of the screen.
- **Skin detection.** Apply morphological operations such as `opening`. Integrate a fast face detector and add some extra features such as face-skin divided by body-skin and biggest skin patch connected. Make use of skin adaptive models.
- **Color histograms.** Find a compact and more knowledgeable way to characterize the color information like incorporating color variance.

— *Classifiers*.

- **Dataset.** Expand the dataset with more sequences, specially in the adult content set, to gain a richer sampling mixture.
- **Fusion strategies.** Experimentally compare decision fusion against feature fusion for the spatial features in terms of accuracy and processing time.
- **Classifier experiments.** Perform a more complete set of experiments for libSVM.

— *Further expansion*.

[1]MATLAB is a programming environment for algorithm development, data analysis, visualization, and numerical computation. URL: http://www.mathworks.se/products/matlab/

- **On line video streaming.** Explore the possibility to deal with on line video streams to be analyzed on-the-fly.
- **Database auto-update.** Research about possible methods of adding samples to the training database automatically when the classifier is used for real-life applications.

6.1 Feature extraction

The first category of improvements is within the feature extraction process. Rather than extracting just the key frames corresponding to each shot to retrieve the **shot structure**, it would be highly discriminative to label the shot transitions according to the editing effect used (i.e. fade, dissolve or cut) as done by [51]. Similarly as the camera motion, a percentage of each of those editing effects might be used for charactering a vector. This on the basis that *dissolves*, are very common in adult films while in home made videos (the main challenging subcategory) the *cuts* are more frequent. The **camera motion identification** algorithm can be improved firstly by replacing the optical flow computation, which is painfully slow, for a decent motion vector extractor working within the coding scheme of MPEG standard[2] as done by [30]. The second improvement would be to modify the algorithm to make it robust enough to deal with non-ideal zooms and z-translation movements. As discussed in section 4.1.3, z-motion percentage was automatically removed by the *attributeSelection* tool. This is because for this particular kind of motion the object of interest has to be exactly at the center of the screen, which means it can not be mixed with any other motion like panning or tilt. This hardly ever occurs in real-life video, and therefore the contribution of this feature is poor. The **skin detector** could be improved without impacting the computational time by adding a fast face detector, such as the one proposed by Viola and Jones [39]. The use of more clever features such as face-skin divided by total skin ratio, size of the biggest skin component in the image, or the use of adaptive skin color models are some of the possible refinements. The **color histogram** representation can be improved by obtaining a more compact feature vector (being 48 the size used currently in our approach). Most of the bins used as features within this vector simply do not contribute at all. Is also possible to add more clever features such as color variance.

6.2 Classifiers

The first improvement is in terms of the **dataset**. Even though the processing time of the video is quite fair with a length of around 3 and a half hours per category and a total duration of 07:15:32 hms, the number of samples is not as big as in some approaches, having just 48 adult clips and 42 harmless samples. In addition to this, it is possible to collect a wider range of adult films based on subcategories as done for the harmless dataset, this to reunite a wider variety of samples representative of a general adult content class. Regarding the **fusion strategies** used for the spatial features, it would be fair to experimentally compare the currently used scheme (feature fusion) against the typically used in the literature (decision fusion) within our approach.

[2] In the MPEG standard, the so called P-frames are predicted for the reference I-frames (which contains the richest amount of information and the lowest compression) by using the motion vectors and the error computed from the difference between the prediction and the actual frame. URL: http://mpeg.chiariglione.org/

For the **classifiers experiment** a more complete set of evaluation tests using libSVM would be desirable.

6.3 Further expansion

The most ambitious ideas for future research lay within this category. The first interesting approach would be to actually being able to deal with **on line video streams** in real time. For instance, analyze video sequences from pages like *youtube* while they are being loaded into the web browser. The second direction would be to find a way to add new samples automatically to **auto update the database**. Mechanisms such as user validation, and on-line maintenance of the database, among many other challenging problems would arise to deal with this.

Bibliography

[1] Ajay Divakaran. *Multimedia Content Analysis: Theory and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[2] Xuanjing Shen, Wei Wei, and Qingji Qian. The filtering of internet images based on detecting erotogenic-part. In *Proc. 3rd International Conference on Natural Computation (ICNC)*, 2007.

[3] T. Deselaers, L. Pimenidis, and H. Ney. Bag-of-visual-words models for adult image classification and filtering. In *Proc. 19th International Conference on Pattern Recognition(ICPR)*, pages 1 –4, 2008.

[4] J. Wang, J. Li, G. Wiederhold, and O. Firschein. System for screening objectionable images. Technical Report 1998-5, Stanford InfoLab, 1998.

[5] H.A. Rowley, Y. Jing, and S. Baluja. Large scale image-based adult-content filtering. In *Proc. International Conference on Computer Vision Theory and Applications*, 2006.

[6] Xiaoyin Wang, Changzhen Hu, and Shuping Yao. A breast detecting algorithm for adult image recognition. In *Proc. International Conference on Information Management, Innovation Management and Industrial Engineering.*, 2009.

[7] Hogyun Lee, Seungmin Lee, and Taekyong Nam. Implementation of high performance objectionable video classification system. In *Proc. 8th International Conference in Advanced Communication Technology (ICACT)*, 2006.

[8] A.P.B. Lopes, S.E.F. de Avila, A.N.A. Peixoto, R.S. Oliveira, M. de M. Coelho, and A. de A. Araujo. Nude detection in video using bag-of-visual-features. In *Proc. XXII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 224 –231, 2009.

[9] Chang-Yul Kim, Oh-Jin Kwon, Won-Gyu Kim, and Seok-Rim Choi. Automatic system for filtering obscene video. In *Proc. 10th International Conference on Advanced Communication Technology (ICACT)*, 2008.

[10] Christian Jansohn, Adrian Ulges, and Thomas M. Breuel. Detecting pornographic video content by combining image features with motion information. In *Proc. 17th ACM international conference on Multimedia*, 2009.

[11] N. Rea, G. Lacey, C. Lambe, and R. Dahyot. Multimodal periodicity analysis for illicit content detection in videos. In *Proc. 3rd European Conference on Visual Media Production(CVMP)*, pages 106 –114, 2006.

[12] T. Endeshaw, J. Garcia, and A. Jakobsson. Classification of indecent videos by low complexity repetitive motion detection. In *Proc. 37th IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1 –7, 2008.

[13] N. Vasconcelos and A. Lippman. A bayesian video modeling framework for shot segmentation and content characterization. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 59 –66, 1997.

[14] M.J. Roach, J.D. Mason, and M. Pawlewski. Video genre classification using dynamics. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[15] Z. Rasheed and M. Shah. Movie genre classification by exploiting audio-visual features of previews. In *Proc. 16th International Conference on Pattern Recognition*, 2002.

[16] Zeeshan Rasheed Yaser, Yaser Sheikh, and Mubarak Shah. Semantic film preview classification using low-level computable features. In *Proc. 3rd International Workshop on Multimedia Data and Document Engineering (MDDE-2003)*, 2003.

[17] Xun Yuan, Wei Lai, Tao Mei, Xian-Sheng Hua, Xiu-Qing Wu, and Shipeng Li. Automatic video genre categorization using hierarchical svm. In *Proc. IEEE International Conference on Image Processing*, pages 2905 –2908, 2006.

[18] Hajar Bouirouga, Sanaa El Fkihi, Abdeilah Jilbab, and Driss Aboutajdine. Comparison of performance between different svm kernels for the identification of adult video. *World Academy of Science, Engineering and Technology*, 2011.

[19] Holger Frohlich, Olivier Chapelle, and Bernhard Scholkopf. Feature selection for support vector machines by means of genetic algorithms. In *Proc. International journal on artificial intelligence tools*, pages 142–148. IEEE Computer Society, 2003.

[20] Kristin P. Bennett and Colin Campbell. Support vector machines: hype or hallelujah? *SIGKDD Explor. Newsletter*, 2000.

[21] Julian Stottinger, Allan Hanbury, Christian Liensberger, and Rehanullah Khan. Skin paths for contextual flagging adult videos. In *Advances in Visual Computing*. Springer Berlin / Heidelberg, 2009.

[22] Yue Wang, Jun Li, HeeLin Wang, and ZuJun Hou. Automatic nipple detection using shape and statistical skin color information. In *Advances in Multimedia Modeling*. Springer Berlin / Heidelberg, 2010.

[23] H. Lu and Y.-P. Tan. An effective post-refinement method for shot boundary detection. In *Proc. International Conference on Image Processing(ICIP)*, 2003.

[24] J. Baber, N. Afzulpurkar, M.N. Dailey, and M.Bakhtyar. Shot boundary detection from videos using entropy and local descriptor. In *Proc. 17th International Conference on Digital Signal Processing (DSP)*, pages 1 –6, 2011.

[25] Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2007.

[26] Chen Yinzi, Deng Yang, Guo Yonglei, Wang Wendong, Zou Yanming, and Wang Kongqiao. A temporal video segmentation and summary generation method based on shots' abrupt and gradual transition boundary detecting. In *Proc. 2nd International Conference on Communication Software and Networks (ICCSN)*, pages 271 –275, 2010.

[27] Hong Jiang Zhang, Jianhua Wu, Di Zhong, and Stephen W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 1997.

[28] A.M. Ferman, A.M. Tekalp, and R. Mehrotra. Robust color histogram descriptors for video segment retrieval and identification. *IEEE Transactions on Image Processing*, 2002.

[29] W. Xiong and J. C. Lee. Automatic dominant camera motion annotation for video retrieval. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 1997.

[30] Sangkeun Lee and Monson H. Hayes. Real-time camera motion classification for content-based indexing and retrieval using templates. In *Proc. IEEE International Conference in Acoustics, Speech, and Signal Processing (ICASSP)*, 2002.

[31] Lina Liu, Ru Zhang, and Ling Fan. Camera motion classification based on svm. In *Proc. 3rd International Congress on Image and Signal Processing (CISP)*, 2010.

[32] J. Nam, M. Alghoniemy, and A.H. Tewfik. Audio-visual content-based violent scene characterization. In *Proc. in Conference on International Image Processing (ICIP)*, 1998.

[33] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. 7th international joint conference on Artificial intelligence*, pages 674–679, 1981.

[34] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 1994.

[35] Y. Wang and R. S. Gaborski. Automatic video classification using holistic spatial features and optical flow. *Intl. Conf. on Image Processing, Computer Vision, and Pattern Recognition*, 2011.

[36] M.J. Jones and J.M. Rehg. Statistical color models with application to skin detection. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[37] J.A. Marcial-Basilio, G. Aguilar-Torres, G. Sanchez-Perez, L.K. Toscano-Medina, and H.M. Perez-Meana. Detection of pornographic digital images. *International journal of computers*, 2010.

[38] Piyabute Fuangkhon and Thitipong Tanprasert. Nipple detection for obscene pictures. In *Proc. 5th WSEAS international conference on Signal, speech and image processing*, 2005.

[39] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 2004.

[40] D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. 7th IEEE International Conference on Computer Vision*, 1999.

[41] J.-K. Kamarainen, V. Kyrki, and H. Kalviainen. Invariance properties of gabor filter-based features-overview and applications. *IEEE Transactions on Image Processing*, 2006.

[42] Xiaofeng Tong, L. Duan, C. Xu, Q. Tian, Hanqing Lu, J. Wang, and J.S. Jin. Periodicity detection of local motion. In *Proc. IEEE International Conference on Multimedia and Expo(ICME)*, pages 650 –653, 2005.

[43] Adrian Ulges, Christian Schulze, Daniel Keysers, and Thomas M. Breuel. A system that learns to tag videos by watching youtube. In *Proc. 6th international conference on Computer vision systems*, 2008.

[44] H. Fujiyoshi and A.J. Lipton. Real-time human motion analysis by image skeletonization. In *Proc. 4th IEEE Workshop on Applications of Computer Vision (WACV)*, pages 15 –21, 1998.

[45] R. Cutler and L.S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[46] Wei hao Lin. News video classification using svm-based multimodal classifiers and combination strategies. In *Proc. In ACM Multimedia, Juan-les-Pins*, pages 1–6. ACM Press, 2002.

[47] Vakkalanka Suresh, C. Mohan, R. Swamy, and B. Yegnanarayana. Content-based video classification using support vector machines. In *Neural Information Processing*. Springer Berlin - Heidelberg, 2004.

[48] Antonio da Luz Jr., Eduardo Valle, and Arnaldo de Albuquerque Araújo. Content-based spam filtering on video sharing social networks. *CoRR*, 2011.

[49] Koen E.A. van de Sande, Theo Gevers, and Cees G.M. Snoek. A comparison of color features for visual concept classification. In *Proc. International conference on Content-based image and video retrieval*, 2008.

[50] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[51] Ba Tu Truong and C. Dorai. Automatic genre identification for content-based video categorization. In *Proc. 15th International Conference on Pattern Recognition*, 2000.

[52] Yizhi Liu, Xiangdong Wang, Yongdong Zhang, and Sheng Tang. Fusing audio-words with visual features for pornographic video detection. In *Proc. IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1488 –1493, 2011.

[53] Sheng Tang, Jintao Li, Yongdong Zhang, Cheng Xie, Ming Li, Yizhi Liu, Xiufeng Hua, Yan-Tao Zheng, Jinhui Tang, and Tat-Seng Chua. Pornprobe: an lda-svm based pornography detection system. In *Proc. 17th ACM international conference on Multimedia*, 2009.

[54] Belur V. Dasarathy. *Decision Fusion*. IEEE Computer Society Press, 1994.

[55] Jude W. Shavlik and Thomas E. Deitterich. *Readings in Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 1991.

[56] D. Brezeale and D.J. Cook. Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2008.

[57] Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and S. Maybank. A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2011.

[58] Jianchao Wang, Bing Li, Weiming Hu, and Ou Wu. Horror video scene recognition via multiple-instance learning. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1325 –1328, 2011.

[59] W. Abd-Almageed. Online, simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing. In *Proc. 15th IEEE International Conference on Image Processing (ICIP)*, pages 3200 –3203, 2008.

[60] Thomas Brox, Andrruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision - ECCV 2004*. Springer Berlin / Heidelberg, 2004.

[61] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 1979.

[62] Shih-Wei Lin, Zne-Jung Lee, Shih-Chieh Chen, and Tsung-Yuan Tseng. Parameter determination of support vector machine and feature selection using simulated annealing approach. *Appl. Soft Comput.*, 2008.

[63] Pinaki Chowdhury, Sukhendu Das, Suranjana Samanta, and Utthara Mangai. A survey of decision fusion and feature fusion strategies for pattern classification. *IETE Technical Review*, 2010.

[64] Chih chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines, 2001.

[65] Fabricio Benevenuto, Tiago Rodrigues, Virgilio Almeida, Jussara Almeida, Chao Zhang, and Keith Ross. Identifying video spammers in online social networks. In *Proc. 4th international workshop on Adversarial information retrieval on the web*, 2008.

[66] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Advances in kernel methods - Support vector learning, 1998.

[67] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Comput.*, 2001.

[68] Soumen Chakrabarti, Shourya Roy, and Mahesh V. Soundalgekar. Fast and accurate text classification via multiple linear discriminant projections. *The VLDB Journal*, 2003.

[69] D. Sculley and Gabriel M. Wachman. Relaxed online svms for spam filtering. In *Proc. 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[70] Hyunchul Kim, KC Claffy, Marina Fomenkov, Dhiman Barman, Michalis Faloutsos, and KiYoung Lee. Internet traffic classification demystified: myths, caveats, and the best practices. In *Proc. ACM CoNEXT Conference*, 2008.

[71] Ji Zhu and Trevor Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 2004.

[72] M. Culjak, B. Mikus, K. Jez, and S. Hadjic. Classification of art paintings by genre. In *Proc. 34th International Convention MIPRO.*, pages 1634 –1639, 2011.

[73] C. W. Hsu, C. C. Chang, and C. J. Lin. *A practical guide to support vector classification*. Department of Computer Science and Information Engineering, National Taiwan University.

[74] Li-Qun Xu and Yongmin Li. Video classification using spatial-temporal features and pca. In *Proc. International Conference on Multimedia and Expo (ICME)*, 2003.

[75] M.A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

7 Appendix

Number of clips	Length range
32	From 4-6 minutes
11	From 1 to 4 minutes
6	From 6 to 7 minutes
Total number of clips	48
Total length	03:46:30 hours

Table 8: Adult-content dataset length details.

Number of clips	Length range
31	From 4-5 minutes
7	From 3 to 3:30 minutes
4	From 7 to 9 minutes
Total number of clips	42
Total length	03:29:02 hours

Table 9: Harmless dataset length details.

				Error		
Support vectors #	Kernel evaluation #	Kappa statistic	Mean absolute	Root mean squared	Relative absolute	Root relative squared
25	9785	0.8886	0.0897	0.2347	17.8147 %	46.524 %

Table 10: **SMO classification additional data**. This includes different error metrics as well. This information corresponds to the best results achieved using the *Automatically selected* set of features altogether with SMO and the Normalized Polynomial kernel function with an accuracy of 94.44%.

Listing 7.1: Main function of the camera motion identification algorithm based in [29, 30, 14])

```
clc;
clear all;

total = tic;


path='C:\My Documents\matlabo\datasets\input';
out='C:\My Documents\matlabo\results\Glabo\2011-05-05 divided_results_5min';

addpath('C:\My Documents\matlabo\Camera motion\opflow\brox');
addpath('C:\My Documents\matlabo\Utilities');
addpath('C:\My Documents\matlabo\Camera motion\mmread');

```

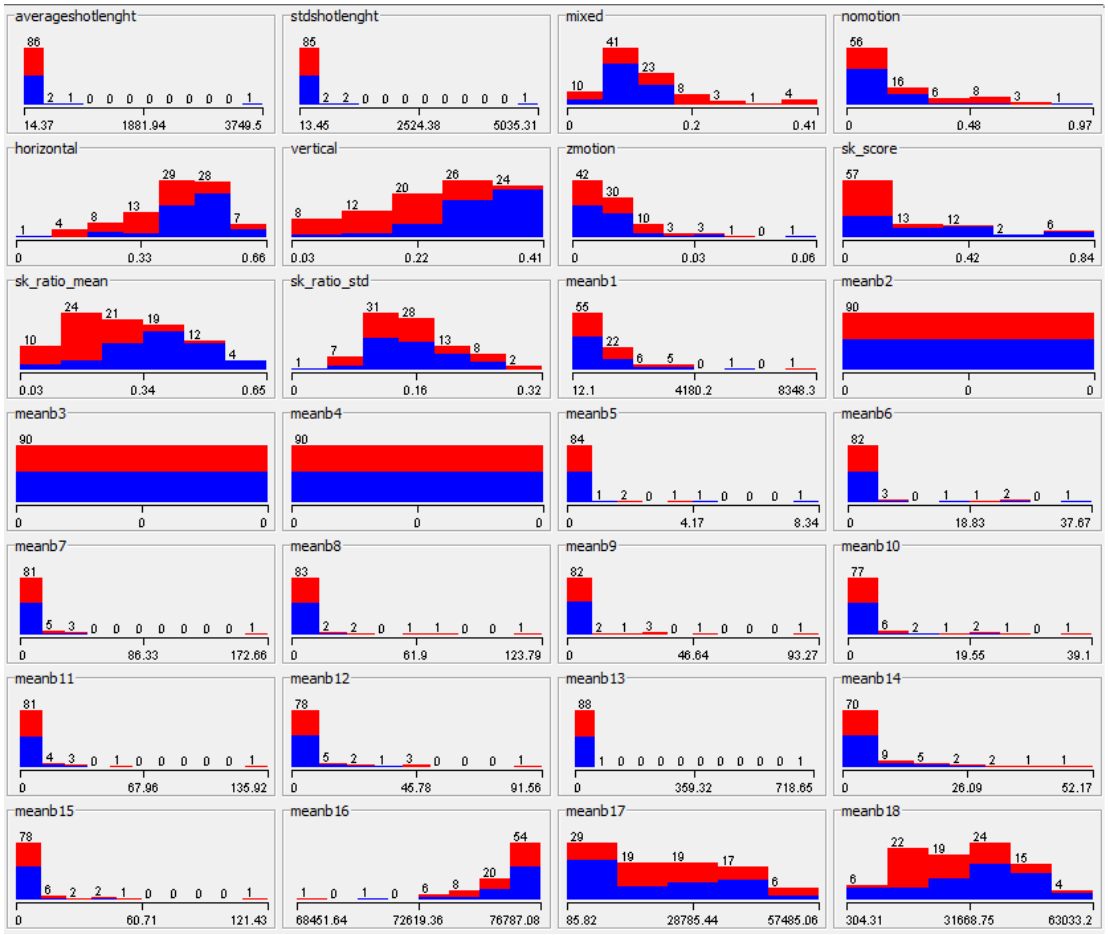

Figure 22: **Feature values and class contribution (part 1).** First part of the series of plots generated by each feature individually.

		Predicted	
		Adult	Non-offensive
Actual	*Adult*	45	3
	Non-offensive	2	40

Table 11: **SMO classifier Confusion matrix**. This information corresponds to the best results achieved using the *Automatically selected* set of features altogether with SMO and the Normalized Polynomial kernel function with an accuracy of 94.44%.

	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**
Adult video	0.938	0.048	0.957	0.938	0.947	0.967
Non-offensive	0.952	0.063	0.93	0.952	0.941	0.967
Weighted Average	0.944	0.055	0.945	0.944	0.944	0.967

Table 12: **SMO classifier detailed accuracy per class**. This information corresponds to the best results achieved using the *Automatically selected* set of features altogether with SMO and the Normalized Polynomial kernel function with an accuracy of 94.44%.

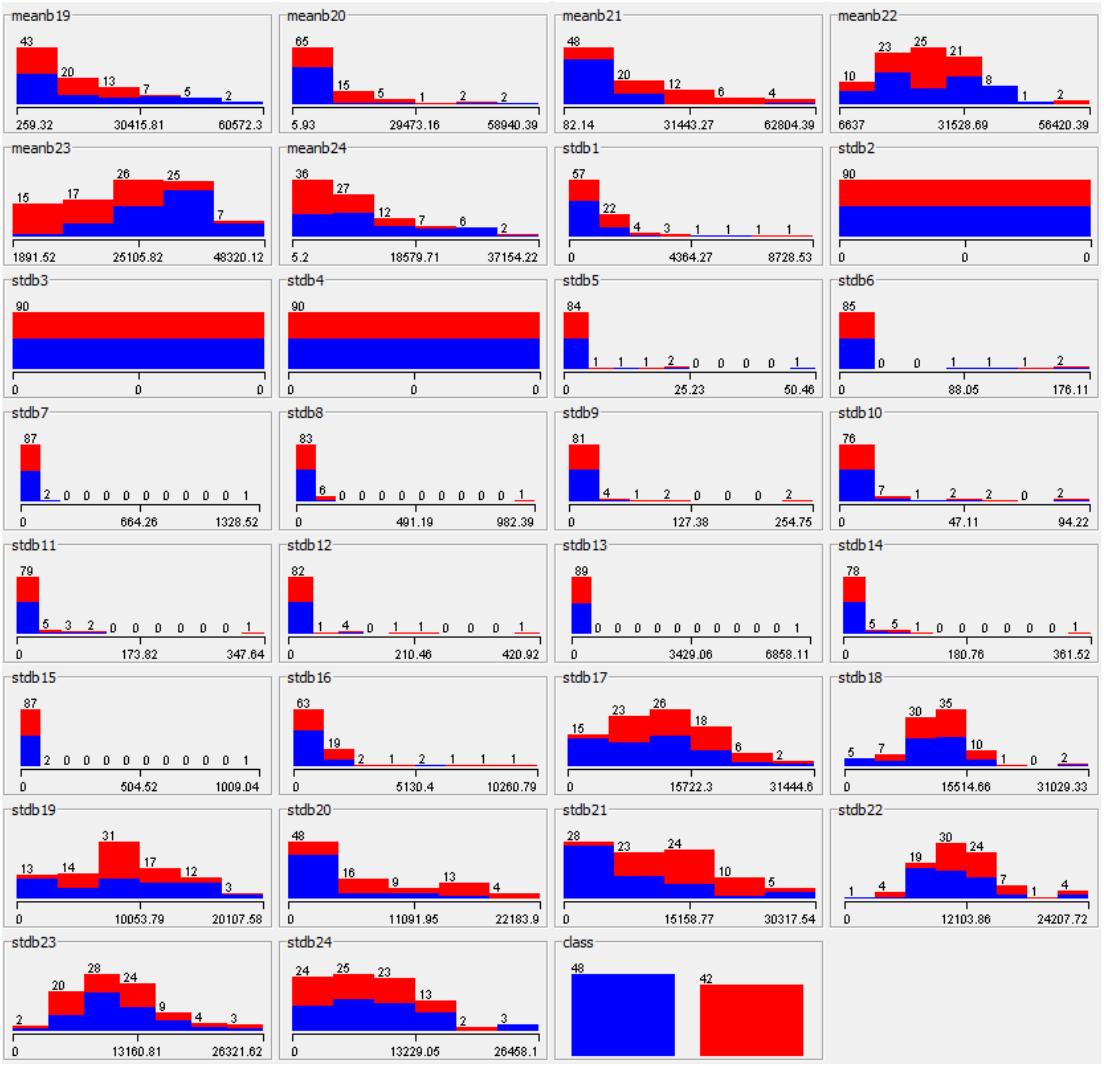

Figure 23: **Feature values and class contribution (part 2)**. Second part of the series of plots generated by each feature individually.

Property	Value
SVM TYPE	C-SVC
caheSize	40
coeaf0	0
debug	No
Replacing missing Values	Yes
eps	0.001
loss	0.1
Normalize	No
nu	0.5
Probability estimation (ROC curve)	Yes
Shrinking	True
Weights	No

Table 13: LibSVM classifier general settings. Table 5 shows the values of the parameters used for every libSVM experiment. Here, the rest of parameters and settings used are shown. This table applies for every libSVM test and contains all the unspecified configurations in the experiment results table.

Property	Value
Build Logistic Models (ROC)	Yes
ChecksTurnedOff	False
debug	No
Epsilon	1.0 E-12
Filter	Normalize training data
randomSeed	1
Tolerance	0.001
Cache Size	250007

Table 14: SMO classifier general settings. Tables 6 and 7 show the specific values of the parameters used for every SMO experiment to compare the feature sets and kernels respectively. Here the rest of the parameters values and settings used commonly for every test are shown.

```
-        9    *     <0.038894 0.241767 0.607615 0.409449 0.05114 0.275046 0.004901 0.061669 0.019775 0.181848 0.503208 0.107589 0.525768 > * X]
-    5.7094   *                    <0.200939 0.295693 0.372181 0.809522 0 0.107045 0 0.493341 0.0074 0.012325 0.742034 0.038422 0.065579 > * X]
-        9    *     <0.026953 0.358174 0.842689 0.642557 0.041148 0.312582 0.000246 0.066104 0.000299 0.008283 0.66742 0.003359 0.055374 > * X]
+    0.0183   *     <0.01501 0.134594 0.193783 0.044031 0.306843 0.450478 0.035126 0.041803 0.158963 0.288776 0.553459 0.684868 0.699218 > * X]
-    5.0632   *       <0.0297 0.406852 0.763055 0.714974 0.212501 0.528698 0.000441 0.312395 0.005526 0.003447 0.891723 0.05723 0.036938 > * X]
+    0.0679   *             <0.008691 0.088483 0.353369 0.054473 0.8758 0.807739 0 0.130451 0.322406 0.855634 0.117015 0.712157 0.266708 > * X]
-    3.3163   *               <0.121701 0.096525 0.318357 0.482356 0.076167 0.325152 0 0.155065 0.06495 0.341 0.292209 0.176008 0.392613 > * X]
+        9    *                  <0.050143 0.305428 0.635687 0.409375 0 0.320676 0 0.116043 0.037508 0.239232 0.458654 0.077523 0.111611 > * X]
+        9    *     <0.021045 0.443835 0.757916 0.711119 0.075524 0.260494 0.017372 0.173088 0.01567 0.040388 0.445097 0.061727 0.144609 > * X]
+    6.0377   *         <0.01164 0.241893 0.221026 0.291542 0.40596 0.572002 0.537401 0.567123 0.106624 0.066528 0.894524 0.317772 0.472 > * X]
+    3.4688   *           <0.023223 0.279483 0.510304 0.413171 0.109475 0.376346 0 0.242581 0.331113 0.499995 0.181394 0.88098 0.706308 > * X]
-     8.306   *    <0.028332 0.278817 0.693474 0.799985 0.063925 0.456756 0.002297 0.252815 0.010461 0.190274 0.649923 0.084214 0.762817 > * X]
-    1.6418   *                          <0.017188 0.24311 0.767535 0.915899 1 1 0.030654 0.285348 1 0.06871 0.621123 0.554338 0.149657 > * X]
-    1.1295   *                                                                <1 0 0 0 0 0 0 0 0.101631 0.067398 0.964695 0.108241 0 0 > * X]
+    4.2404   *            <0.001958 0.167562 0.857597 0.26162 0.190164 0.338803 0 0.137564 0.253543 0.636107 0.193532 0.583465 0.500815 > * X]
-    7.8363   *           <0.067716 0.281401 0.728653 0.793829 0.170468 0.458885 0 0.082234 0.003727 0.325826 0.313434 0.008792 0.402149 > * X]
-    7.4313   *    <0.022873 0.229877 0.810959 0.284107 0.226311 0.418987 0.011461 0.473387 0.035126 0.269959 0.357243 0.186305 0.571087 > * X]
+        9    *           <0.025021 0.411425 0.718291 0.812204 0 0.362839 0.001072 0.137741 0.056562 0.199459 0.788259 0.222449 0.455053 > * X]
+    2.6164   *           <0.007064 0.380486 0.689921 0.789653 0.309368 0.527967 0 0.334314 0.476162 0.773007 0.115096 0.695272 0.536291 > * X]
+        9    *      <0.009298 0.439008 0.649746 0.724954 0.01665 0.14014 0.002383 0.040434 0.003451 0.161954 0.466754 0.023722 0.606289 > * X]
+    5.9843   *             <0.010489 0.503446 0.609861 0.482064 0.02887 0.23723 0 0.40219 0.029387 0.115036 0.706556 0.071974 0.420038 > * X]
+        9    *    <0.050769 0.267204 0.735778 0.657822 0.307943 0.493381 0.026285 0.062877 0.009608 0.270676 0.534028 0.072539 0.955467 > * X]
-        9    *                  <0.041223 0.273982 0.682242 0.675649 0 0.243918 0.076337 0.055162 0.061795 0.30658 0.496375 0.661773 1 > * X]
+        9    *               <0.0348 0.362247 0.697197 0.90312 0.017294 0.207368 0 0.20683 0.156478 0.249938 0.708196 0.228703 0.355764 > * X]
-        9    *    <0.012525 0.401992 0.691411 0.819437 0.089336 0.346262 0.061061 0.277197 0.200218 0.159716 0.493369 0.350399 0.299523 > * X]
-    0.1639
```

Table 15: **SMO classifier Support vectors**. This data corresponds to the best results achieved using the *Automatically selected* set of features altogether with SMO and the Normalized Polynomial kernel function with an accuracy of 94.44%.

```
f_interval=12;
persistence_t= 12;
filelist=dir(path);
levels=9; %Name of pyramidal levels for the optical flow computation (raises precision
    and time)

%Create .mat files with the f_interval to append the info later
save([out '\motion_stats'], 'f_interval');
save([out '\motion_stats_vectors'], 'f_interval');
save([out '\camera_motion'], 'f_interval');
save([out '\time'], 'f_interval');

    for i=3:size(filelist, 1);%For each video in the folder
    loop = tic;
     [pathstr, fname, ext]=fileparts(filelist(i,1).name);

    [vid]= mmread([path '\' filelist(i,1).name], [], [], false, true);%Read video
    n_frames= vid.nrFramesTotal;
    [vid]= mmread([path '\' filelist(i,1).name], [1 f_interval:f_interval:n_frames],
        [], false, true);%Read video
    n_frames = size(vid.frames,2);

    camera_motion= zeros(n_frames-1,2);
    camera_motion(:,:)=-6; %6 does not correspond to any movement label therefore is
                    %used as a default value
%%
        fcur= vid.frames(1,1).cdata;
        fnext = vid.frames(1,2).cdata;

        [u, v] = optic_flow_brox(fcur, fnext, levels);%Compute optical flow
        [label]= labelMotion(u,v); %Identify and label motion

        camera_motion(1,:)=[f_interval label];

        cm_i=2;%Camera_motion index
        i_frame= f_interval;%
        %%
        for j=3:n_frames %Computes optical flow of n-1,n and n,n+1 and sums them up

        i_frame= i_frame+f_interval;
        fcur= vid.frames(1,j-1).cdata;
        fnext = vid.frames(1,j).cdata;

        [u, v] = optic_flow_brox(fcur, fnext, levels);%Compute optical flow
        [label]= labelMotion(u,v); %Identify and label motion

        %Labels: -1:Mixed  0: No motion 1: Horizontal, 2: Vertical, 3: Z-motion
            if camera_motion(cm_i-1,2)==label
            %Do nothing
            else
            camera_motion(cm_i,:)=[i_frame label];
            cm_i=cm_i+1;
            end
        end

    %% Marking final frame
        if camera_motion(cm_i-1,1)==i_frame %If last frame checked is already there
        %Do nothing
        else if camera_motion(cm_i-1,2)==label %If label matches
                camera_motion(cm_i-1,:) =[i_frame label]; %Overwrite with last frame
            else
                camera_motion(cm_i,:) =[i_frame label];%New rew for the new label
```

```
            end
75
        end
77
    camera_motion(any(camera_motion==-6,2),:)=[];%Delete garbage rows
79

81  %% Create array of statistical information about camera's operation
    motion_stats=[[-1; 0; 1; 2; 3] [0;0;0;0;0]];
83
  %For the first camera_motion row
85  frames= camera_motion(1,1)-1;
      if frames>=persistence_t
87    lb=camera_motion(1,2);
      index=find(motion_stats(:,1)==lb);
89    else
      index=find(motion_stats(:,1)==-1);
91    end
    motion_stats(index,2)=motion_stats(index,2)+ frames;
93  %% Rest of the frames
      for k=2:size(camera_motion)
95    frames= camera_motion(k,1)- camera_motion(k-1,1);

97        if frames>=persistence_t
          lb=camera_motion(k-1,2);
99        index=find(motion_stats(:,1)==lb);
          else
101       index=find(motion_stats(:,1)==-1);
          end
103   motion_stats(index,2)=motion_stats(index,2)+ frames;
      end
105
     %% Motion stats normalizer
107   total_frames= sum(motion_stats(:,2));
      ms_vector= motion_stats(:,2)/total_frames;
109   ms_vector=ms_vector';
      %% Save data
111   loop_time=secs2hms(toc(loop));

113   eval([fname '_ms_vector = ms_vector;']);
      eval([fname '_motion_stats = motion_stats;']);
115   eval([fname '_camera_motion = camera_motion;']);
      eval([fname '_time = loop_time;']);
117
      save([out '\motion_stats'], '-regexp', '_motion_stats', '-append');
119   save([out '\motion_stats_vectors'], '-regexp', 'ms_vector',  '-append');
      save([out '\camera_motion'], '-regexp', '_camera_motion', '-append');
121   save([out '\time'], '-regexp', '_time', '-append');


123


125   end%For each video

127
total_time_r= secs2hms(toc(total));
129 date_r=datestr(now);


131
save([out '\time'], '-regexp', '_r', '-append');
133 save([out '\variables']);
```

Listing 7.2: Function to identify and label the type of camera operation. Algorithm based in [29, 30, 14])

```
function [label]=labelMotion(u,v)

[x0,y0, z, i,ii, iii, iv] =subregions(u);%Get subregions indexes
%% Divide the motion vector matrix into subregions using the indexes

% u: horizontal component
X0_u=vertcat(u(x0(2,2,1):x0(1,2,1), x0(1,1,1):x0(3,1,1)), u(x0(2,2,2):x0(1,2,2), x0
    (1,1,2):x0(3,1,2)));
Y0_u=[u(y0(2,2,1):y0(1,2,1), y0(1,1,1):y0(3,1,1)) u(y0(2,2,2):y0(1,2,2), y0(1,1,2):y0
    (3,1,2))];
zero_u= u(z(2,2,1):z(1,2,1), z(1,1,1):z(3,1,1));
I_u=u(iv(2,2,1):iv(1,2,1), iv(1,1,1):iv(3,1,1));
II_u=u(iii(2,2,1):iii(1,2,1), iii(1,1,1):iii(3,1,1));
III_u=u(ii(2,2,1):ii(1,2,1), ii(1,1,1):ii(3,1,1));
IV_u=u(i(2,2,1):i(1,2,1), i(1,1,1):i(3,1,1));

% v: vertical component
X0_v=vertcat(v(x0(2,2,1):x0(1,2,1), x0(1,1,1):x0(3,1,1)), v(x0(2,2,2):x0(1,2,2), x0
    (1,1,2):x0(3,1,2)));
Y0_v=[v(y0(2,2,1):y0(1,2,1), y0(1,1,1):y0(3,1,1)) v(y0(2,2,2):y0(1,2,2), y0(1,1,2):y0
    (3,1,2))];
zero_v= v(z(2,2,1):z(1,2,1), z(1,1,1):z(3,1,1));
I_v=v(iv(2,2,1):iv(1,2,1), iv(1,1,1):iv(3,1,1));
II_v=v(iii(2,2,1):iii(1,2,1), iii(1,1,1):iii(3,1,1));
III_v=v(ii(2,2,1):ii(1,2,1), ii(1,1,1):ii(3,1,1));
IV_v=v(i(2,2,1):i(1,2,1), i(1,1,1):i(3,1,1));

%% Calculate mean and standard deviation for every zone
%X0,Y0,zero,I,II,III,IV
mean_u=[mean2(X0_u) mean2(Y0_u) mean2(zero_u) mean2(I_u) mean2(II_u) mean2(III_u) mean2
    (IV_u)];
mean_v=[mean2(X0_v) mean2(Y0_v) mean2(zero_v) mean2(I_v) mean2(II_v) mean2(III_v) mean2
    (IV_v)];

%Dominances
u_dom= mean_u./mean_v; %Horizontal over vertical ratiov_dom
v_dom= mean_v./mean_u; %Vertical over horizontal ratio


st=.4; %Threshold for declaring no global motion
dt= .7; %Threshold. 1 is perfect balance between vertical and horizontal components
mt= .3;
mt_y0= .6;
stdt= .9; %


%% Set of rules to identify motion

%Labels: -1:Mixed  0: No motion 1: Horizontal, 2: Vertical, 3: Z-motion
%X0,Y0,zero,I,II,III,IV
%ERROR

if scoreMeans(abs(mean_u),st) >= .5 && scoreMeans(abs(mean_v),st)>=.5
% Vertical and horizontal mean components smaller than threshold
    label=0; %No camera motion

  else if abs(mean_u(2))<= mt && abs(mean_v(1)) <= mt...
       ||  abs(mean_u(1))<= mt && abs(mean_v(2)) <= mt

        label= 3;   %Z motion
```

```
        else if scoreMeans(v_dom, dt) >= .5
             label=1;       %Horizontal - Pan or horizontal translation  mean(abs(u_dom
                 (4:7)))>mean(abs(v_dom(4:7)))
            else if scoreMeans(u_dom, dt)>= .5  %If v I?II?III?IV?0 (similar and small)
                 label= 2;
                else
                 label=-1;  %Mixed
                end
             end
        end
end
```

Listing 7.3: Function to evaluate sub-regions to compute a vote-based score. Algorithm based in [29, 30, 14])

```
function score=scoreMeans(mean,th)

    flags=0;
     for i=4:7

          if abs(mean(1,i))<= th
          flags=flags+1;
          end

     score=flags/4;
     end
```

Listing 7.4: Function to divide the video canvas in sub-regions. Algorithm based in [29, 30, 14])

```
function [X0,Y0, zero, I,II, III,IV] =subregions(mat)

%Each of the subregions
zero=zeros(4,2);
X0=zeros(4,2,2);
Y0=zeros(4,2,2);
I=zeros(4,2);
II=zeros(4,2);
III=zeros(4,2);
IV=zeros(4,2);

[h,w]= size(mat);


% Y0 area will be 1/4 of the total height localized in the verticalcenter
% of the canvas

% Two dimensions

%(:,1,:)-> x (horizontal)
%(:,2,:)-> y (vertical)

%(1,:,:)-> Upper left
%(2,:,:)-> Bottom left
%(3,:,:)-> Upper right
%(4,:,:)-> Bottom right


% Three dimensions
%(:,:,1)-> Left Y0,  Upper X0
%(:,:,2)-> right Y0, Bottom X0
```

```
32
34 % X0 area will be 1/6 of the total width localized in the horizontal center
   % of the canvas
36 % Y0 area will be 1/4 of the total width localized in the horizontal center
   % of the canvas
38
    % Y0 LEFT area
40      %Upper left
        Y0(1,1,1)=1;
42      Y0(1,2,1)=round(h/2+h/8); %Same as Y0(3,1,1)
        %Bottom left
44      Y0(2,1,1)=1;
        Y0(2,2,1)=floor(Y0(1,2,1)-(h/4));
46      %Upper Right
        Y0(3,1,1)=floor(w/2-w/12); %Bordering with X0(1,2,1) [-1]
48      Y0(3,2,1)=round(h/2+h/8); %Same as Y0(1,1,1)
        %Bottom right
50      Y0(4,1,1)=floor(w/2-w/12);
        Y0(4,2,1)=floor(Y0(1,2,1)-(h/4));
52
    % X0 UPPER area
54      %Upper left
        X0(1,1,1)=floor(w/2-w/12)+1;
56      X0(1,2,1)=h;
        %Bottom left
58       X0(2,1,1)= floor(w/2-w/12)+1;
         X0(2,2,1)= Y0(3,2,1)+1;
60      %&&&&&&&&&&&&&&&&&&&&&&&&
        %Upper Right
62      X0(3,1,1)=round(X0(1,1,1)+w/6);
        X0(3,2,1)=h;
64      %Bottom right
        X0(4,1,1)=round(X0(1,1,1)+w/6);
66      X0(4,2,1)=Y0(3,2,1)+1;

68  % Y0 RIGHT area
        %Upper left
70      Y0(1,1,2)=X0(4,1,1)+1;
        Y0(1,2,2)=Y0(1,2,1);
72      %Bottom left
        Y0(2,1,2)=X0(4,1,1)+1;
74      Y0(2,2,2)=Y0(4,2,1);
        %Upper Right
76      Y0(3,1,2)= w;
        Y0(3,2,2)=Y0(1,2,1);
78      %Bottom right
        Y0(4,1,2)= w;
80      Y0(4,2,2)=Y0(4,2,1);

82     % X0 BOTTOM area
        %Upper left
84      X0(1,1,2)=X0(1,1,1);
        X0(1,2,2)=Y0(4,2,1)-1;
86      %Bottom left
        X0(2,1,2)= X0(1,1,1);
88      X0(2,2,2)= 1;
        %&&&&&&&&&&&&&&&&&&&&&&&&
90      %Upper Right
        X0(3,1,2)=X0(4,1,1);
92      X0(3,2,2)=Y0(4,2,1)-1;
        %Bottom right
```

```
94          X0(4,1,2)=X0(4,1,1);
            X0(4,2,2)=1;
96
%Area zero (central area)
98      %Upper left
        zero(1,1)=X0(2,1,1);
100     zero(1,2)=Y0(3,2,1);
        %Bottom left
102     zero(2,1)=X0(2,1,1);
        zero(2,2)=Y0(4,2,1);
104     %Upper Right
        zero(3,1)=X0(4,1,2);
106     zero(3,2)=Y0(1,2,2);
        %Bottom right
108     zero(4,1)=X0(3,1,2);
        zero(4,2)=Y0(2,2,2);
110
%Area I (upper right)
112     %Upper left
        I(1,1)=X0(3,1,1)+1;
114     I(1,2)=h;
        %Bottom left
116     I(2,1)=X0(3,1,1)+1;
        I(2,2)=Y0(1,2,2)+1;
118     %Upper Right
        I(3,1)=w;
120     I(3,2)=h;
        %Bottom right
122     I(4,1)=w;
        I(4,2)=Y0(3,2,2)+1;
124
%Area II (upper left)
126     %Upper left
        II(1,1)=1;
128     II(1,2)=h;
        %Bottom left
130     II(2,1)=1;
        II(2,2)=Y0(1,2,1)+1;
132     %Upper Right
        II(3,1)=X0(1,1,1)-1;
134     II(3,2)=h;
        %Bottom right
136     II(4,1)=X0(2,1,1)-1;
        II(4,2)=Y0(3,2,1)+1;
138
%Area III (bottom left)
140     %Upper left
        III(1,1)=1;
142     III(1,2)=Y0(2,2,1)-1;
        %Bottom left
144     III(2,1)=1;
        III(2,2)=1;
146     %Upper Right
        III(3,1)=X0(1,1,2)-1;
148     III(3,2)=Y0(4,2,1)-1;
        %Bottomright
150     III(4,1)=X0(1,1,2)-1;
        III(4,2)=1;
152
%Area IV (bottom right)
154     %Upper left
        IV(1,1)=X0(3,1,2)+1;
```

```
156     IV(1,2)=Y0(2,2,1)-1;
        %Bottom left
158     IV(2,1)=X0(3,1,2)+1;
        IV(2,2)=1;
160     %Upper Right
        IV(3,1)=w;
162     IV(3,2)=X0(3,2,2);
        %Bottom right
164     IV(4,1)=w;
        IV(4,2)=1;
```