TU Kaiserslautern & DFKI
Image Understanding and Pattern Recognition
Prof. Dr. Thomas Breuel

Diplomarbeit

Statistical Classification of Image Content for Visual Information Filtering

Christian Jansohn

Februar, 2009

Betreuer:
Christian Schulze
&
Prof. Dr. Thomas Breuel

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge und Grafiken, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Kaiserslautern, im Februar 2009

Christian Jansohn

Acknowledgments

I wish to express my gratitude to all the people who helped and supported me with this thesis. I would like to thank Thomas Breuel and everyone at IUPR, especially Christian Schulze and Adrian Ulges, who constantly gave me useful tips and helped to shape this work into what it is now. I would also like to thank all other members at IUPR who provided help and answered my questions, especially Damian Borth, Markus Goldstein, and Matthias Reif. Finally, I would like to thank my parents for their continuous support.

Abstract

An increasing number of freely accessible adult content websites arose recently, displaying a wide variety of different offensive images and videos. Since many users do not want to be confronted with such material, automatic tools to detect and filter these images and videos are needed. Additionally, tools are required to protect children from accessing offensive websites. This thesis presents approaches for both classification of offensive images and videos. For the first, two different approaches are presented and evaluated on a variety of different datasets showing real world Web content. One traditional method is based on detecting and describing skin areas, while the other uses the popular bag-of-visual-words model. Video classification is based on keyframes and additional motion features, including periodicity detection. Evaluation of these techniques is done on offensive Web videos and inoffensive YouTube videos. The results show that the bag-of-visual-words approach is better suited for classifying offensive material than traditional skin features. Also, combining keyframe classification with additional motion features improves the performance of detecting offensive videos. Overall, a classification accuracy of 99% on images, and 94% on videos is reached.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This work deals with the classification of offensive material based on visual information. Hereby, offensive material include all kinds of adult-content, or pornographic images and videos. The focus lies on just the visual informations, so no additional text, or audio features are included in the classification process. A system which is able to block these kinds of images and videos would be beneficial, since more and more offensive material shows up and can be accessed by everyone, even by accident. A more detailed description and motivation for this work is presented in the following section. The successive sections cover previously published work on this topic, divided into the classification on images, and videos, as well as a section about skin detection, since it plays an important part in this task.

For the classification of offensive images, two methods are presented. One is based on detecting *skin* areas in an image and using simple features to describe these areas. The second approach uses the *bag-of-visual-words* method, which became recently popular and showed already good performance for image classification. Both are evaluated on a previously proposed dataset, downloaded offensive images, and inoffensive images from different sources, including Flickr, and normal Web images. Video classification is divided into keyframe based classification, and the incorporation of additional motion features. These features are evaluated on offensive Web videos and YouTube videos for the opponent class.

The further chapters are organized as follows: the second chapter covers theoretical background of the underlying methods, as well as the used classifiers: the *decision tree*, and the *support vector machine* (SVM). Chapter three presents the different approaches, while in chapter four the used datasets are explained. Also the experiments and their results are described in that chapter. Chapter five is a conclusion of this work and an outlook of possible future work is given there as well.

1.1 Motivation

The ever increasing amount of Web traffic makes it possible to find much different pornographic material in the Internet. Nowadays there is a huge offer of different, freely available, pornographic or offensive images and videos. This becomes an ever rising problem, since many users do not want to be confronted with such material. It is even more important to guard children from offensive material since it is very easy to accidentally visit websites that host these. Therefore, a system which is able to detect offensive images might be very beneficial.

One of the first problems which arise, is the definition of which image might be regarded as offensive and which not. The definition mainly depends on the culture, the country and its laws, and one's personal views. For example, in Germany it is common to see naked female breasts on cover of magazines, and in TV commercials. However more strict

and religious cultures may find this offending. Figure 1.1 shows some images that may be offensive depending on the culture's view. The first image (Figure 1.1(a)) shows a woman with partly transparent clothes under a shower. In most European countries this image would not be regarded as an offensive image by most adults. But parents might not want their young children to see such an image. The second image (Figure 1.1(b)) displays a bare breasted woman. Again this image may not be offensive in everybody's view. The third image (Figure 1.1(c)) shows people having sex, which clearly states it as an adult image which should not be viewed by children. In this work, a common sense view is applied regarding the definition of offensive. An image is therefore offensive (if not stated otherwise), if it contains (partly) naked sexual organs including female breasts, or if it shows sexual actions. Inoffensive images are all images, that are not offensive. Because the definition of offensiveness is not trivial, a possible requirement for a system that blocks these images is flexibility. It should be able to be tuned for different personal or cultural needs.

Most of the existing systems that deal with Web content filtering are based on one of the following approaches: black- , or whitelists, keyword scanning, and rating systems [13]. Blacklists are lists of websites that contain unwanted material and therefore are blocked. This is an efficient way to block material from these sites, but a big disadvantage is the amount of work that is needed to keep these lists up to date. Everyday new websites come up and it is nearly impossible to keep track over all the websites. Whitelists, in contrast, contain only websites that are known to be clean. The same disadvantage as for the blacklists applies, too. It is nearly impossible to get a list of all available websites. One might only rely on the pages he has visited so far, but this might restrain him from finding new interesting websites. Keyword scanning is a technique, that scans a website for certain keywords. If one keyword is found, the site is blocked. The advantage of this method is in its simplicity. Offensive websites, for example, might contain the word "breast". So a website is regarded as offensive, if the word "breast" is found. This might also apply on websites that deal with breast cancer. Also such a mechanism can easily be fooled by swapping characters in the word. "Braest", for example, could not be recognized anymore. A rating system allows either the owner of a website, frequent visitors of the site, or an independent third party to rate the page's content. The rating then can be used for filtering purposes. However, there are various disadvantages of this approach. First, such rating information may not be available for all websites. Second, it is not assured that this information may be always reliable, especially if the information is user generated or given by the owner. There exist also more sophisticated text based classifiers of offensive websites. They basically try to learn frequent occurring words on websites with different classifiers like support vector machines, neural networks, or nearest-neighbor [13, 21, 27, 47]. A general disadvantage of these methods is, that there are not able to filter websites, that only contain images. Since the actual images are what has to be blocked, a system that can recognize offensive images only based on the visual information would be of great benefit.

Another reason for a system that just operates on images, is that other possible applications exist that are not just for blocking images from websites. One possible application could be to help police forces to find offensive images on hard disks. Since the available storage capacity on disks increased drastically over the last years, more and more pictures may be stored locally. Some of the offensive images might be illegal and a tool that quickly scans a hard drive and finds all possible offensive images might be beneficial, since not all images have to be searched manually.

Video streams also benefited form the increasing Web traffic. More online portals arise where videos can be viewed over the Internet, for example YouTube[1]. YouTube allows users to upload their own videos and to share them with everybody else. However, it is not wanted that users upload offensive videos as YouTube explains in their Community Guidelines[2]. The

[1] http://www.youtube.com
[2] http://www.youtube.com/t/community_guidelines

(a) (b) (c)

Figure 1.1: *Samples of images that might be offensive depending on the personal view: a) might be regarded as offensive only in very strict cultures, b) might be regarded as offensive in some cultures, c) should be regarded as offensive in most cultures*

portal uses a community based approach to find videos that contain unwanted scenes. Users can report films that, in their view, violate the guidelines. It is possible to set a flag which tells that a video may be inappropriate for an underage person to view. However, this is not a real age certification, since everybody with an YouTube account can view these videos. A system that automatically detects videos that show sex and nudity might therefore lead to a better recognition of offensive material on such online portals. Additionally, there exist also similar portals that focus on offensive video streams. Again, there is not much protection from these sites and a system which can detect offensive videos might be beneficial to protect children.

Videos can be seen as a set of images. Using this, the detection of offensive films could be done by extracting meaningful frames out of the video stream. Then classify these keyframes and if they are offensive, label the video as offensive, too. However, video streams also include other information which further might be used to improve the classification process. Possible features might be evolved out of the motion or the audio stream.

The goal of this work is to present a system that is able to classify images into offensive and inoffensive ones. The system is afterwards extended to video data. The following sections present some recent work on the detection of adult images and the detection of adult videos.

1.2 Related Work

This section presents some of the approaches for the detection of offensive visual material that already exist. It is divided into three parts. In the first part methods for the classification of offensive images are presented. Many of these methods share the same general approach whose first step consists of some skin detection technique. Because there exist several methods of detecting skin, in the second part some comparative results of previous works, are presented. The last part covers additional methods for the classification of offensive videos.

1.2.1 Classification of Offensive Images

Since most of the offensive images show naked people or parts of naked people, an intuitive approach would be to find skin areas in an image and do a classification of the resulting skin area. Skin itself consists mainly of melanin and blood and because of that the dominating colors are red, brown, and yellow [34]. Therefore skin color could be easily distinguished from other materials. However, some matters complicate the detection of human skin. First,

the skin color may share a common source, but it is also very spread around different races. So there are many combinations from very dark, over brown, red and yellow, to very bright. Second, different illumination sources may change the appearance of skin color and may lead to even blueish or greenish tones. Also the skin area might be desaturated or shadowed, which leads to a much darker or brighter appearance. Finally, there exist various materials like rock, wood, fur of animals, or some kinds of metal whose color resembles the color of human skin. This may lead to areas in the background of images, which may be classified as skin. Some methods that try to cope with these problems and distinguish images between offensive and not are presented in this section. The following section describes some other results regarding skin detection, mainly the underlying methods and choices of color space that also play an important role.

The first approach for the classification of offensive images was presented by Forsyth and Fleck in 1996 [17]. This approach aims to find naked people in pictures by first extracting large skin regions and then try to match the skin regions to cylindrical shapes. These shapes are grouped to form human limbs which are further grouped to form bodies. To solve the problem of detecting skin in an image, the authors transform the R, G, B color values into the log-opponent values I, R_g, B_y. The idea behind this transformation, is to make the R_g and B_y values independent from the intensity value which is represented with the green color value. The skin filter itself consists of a set of rules that mark pixels as skin, whose color values are in a certain, manually defined range. Because skin is usually soft it has only little texture. So areas are additionally marked as skin, if their texture amplitude is small. Due to the skin's reflectance, small gaps within the skin regions may occur. To avoid these gaps, the output of the skin filter is expanded so that adjacent regions are included, if the deviation of their color values is small. The skin regions are then analyzed to find cylindrical shapes, since most parts of the human body can be represented in that way. To achieve this, first, edge detection is performed to find sets of connected edge curves. A set is denoted cylindrical, if a straight axe can be put through it. The cylinders that are found are afterwards grouped into possible human limbs by using a set grouping rules. For example, two cylinders may form a limb, if their axes intersect and the average width of both is similar. Further rules are used to find girdles, thighs or spines. If a body can be found in this way, the picture is regarded as being offensive. For experiments 565 pictures that contain naked people are used and 4,289 pictures with different motifs. The people were mostly Caucasians, some were Africans or Asians. The authors claim to achieve a recall rate of 43% with a precision of 57%. The skin filter achieved a recall rate of 79% and a precision of 48%. So the whole algorithm could successfully extract 43% of the test images but only 4% of the control images.

Another approach was presented by Ze Wang et al. in [45] and called WIPE$_{TM}$ (Wavelet Image Pornography Elimination). This method consists of five steps which include icon detection, graph/photo detection, color histogram analysis, texture analysis and shape matching. The icon filter checks if the length of any side of an image is small. If this is the case the image is categorized as benign. Images that pass the icon filter are rescaled so that the longest side has 256 pixels. This is mainly done to save computation time in the successive steps. The second step aims to detect whether an image is a photograph or a graph. If it is a graph, it is likely that it is not offensive. To do this, the image is partitioned into blocks where each block is classified as graph or photo. If the percentage of one class exceeds the other, the whole image is classified as this class. The classification is based on the analysis of wavelet coefficients in high frequency bands, because artificial images tend to have sharper edges than photographs. The authors claim that the RGB color histograms of offensive images have a different color distribution than inoffensive images. For the analysis the RGB color space is partitioned into 512 bins. Also a color range that resembles human skin color

was defined manually. With that color range a weight for each color in the histogram is set, expressing its probability of belonging to skin. By summing over the whole histogram, a weighted amount of human body colors can be obtained. Finally, a threshold is used to decide to which class the image belongs. The following step analyzes the texture of the areas in which skin color was detected. The histogram of high frequency bands of the wavelet transform is analyzed and if an area contains many high frequencies it is classified as not being skin.

The final step analyzes the shape of the extracted skin regions. First, an edge map of the image is constructed by applying a wavelet transform to construct an edge map for vertical, horizontal and diagonal edges each. The three edge maps are combined to form the final edge map. To describe the shapes in the edge map the normalized central moments up to order five are computed and the seven translation, rotation and scale invariant moments defined by Hu [19] are used as well, giving a feature vector of length 28. The feature vectors of the training images are stored in a database. To classify new images, the Euclidean distance to the existing feature vectors is calculated. If the vector is close to an existing one, the image is classified as offensive. For training, 500 offensive images were downloaded from the internet and 8,000 benign image were used from various sources. Testing was performed on 1,076 objectionable and 10,809 benign images, achieving a true positive rate of 96% with a false positive rate of 9%.

Jones' and Rehg's work [22] is focused on the detection of human skin. They claim that skin color is separable in the color space and that a model for skin detection that is based on the color distribution can be built if a training set of sufficient size is used. The model they built is based on 1 billion labeled training pixels gathered from the internet. The decision the classifier takes is only based on the RGB values of a pixel and is based on Bayes rule. The probability of a pixel belonging to skin $P(skin \mid rgb)$ is estimated with histograms over the color space.

Histograms with size of 32 bins lead to a classification rate of 80% with a false positive rate of 8.5% or 90% classification rate with a false positive rate of 14.2%. Most of the false positives are from wood, rock or copper colored materials while highly saturated or shadowed skin also fails. They also compared the histogram based model to a mixture of 16 Gaussian functions to represent the skin color region. The histogram model achieved slightly better results.

After skin detection, simple features that are based on the skin areas are used to classify offensive images. These features include:

- percentage of pixels detected as skin
- average probability of skin pixels
- size in pixels of the largest connected component
- number of connected components
- size of the image

A neural network classifier was trained on 5,453 pornographic images and 5,226 non-pornographic images. The classifier achieved 85.8% correct detections and 7.5% false positives on gathered images from web crawls. Most of the false positives were portrait images. Finally additional text-based features were combined into the classifier. These features are just a list of objectionable words that are found on a website matched against the image label. The final classifier achieved a classification rate of 93.9% with a false positive rate of 8.0%.

Harvey and Smith present another approach for finding skin color in [9]. They compared five different color spaces for this: RGB, HSV, normalized RGB, Log opponent, and comprehensive. For the training 140 images were used. If necessary, the pixels are first transformed into the target color space. The resulting skin color cluster is then normalized using Principal Component Analysis (PCA) [37]. This forms a cluster of skin color points in the whole color space, which is centered around the origin. To determine if a pixel belongs to skin, it first has to be transformed into the right color space. Afterwards it has to be projected into the transformed skin color space in which the distance to the origin is calculated. If it is lower than a given threshold θ, the pixel is denoted as skin. The results for the different color spaces show, that the choice of color space is not critical regarding the classification performance. However, the approach has a general drawback, since the transformation with PCA takes some computational time. To speed this up, a lookup table is proposed, which is similar to Jones approach [22]. Histograms are used to create a likelihood score for a color belonging to skin. The authors also mention another problem, the occurrence of isolated pixels which are associated with the background although they belong to skin and vice versa. To remove these pixels, a region growing algorithm is used.

For classification three features are used which try to capture the skin areas. The features are: the ratio of the skin area to the whole image area, the ratio of the area of the largest skin segment to the whole image area, and the number of segments in the picture. A k-nearest neighbor classifier is used which achieved around 55% correct classifications. To improve these results, the integration of additional text features is proposed. These features are for example tags (meta information), title, or descriptive commentary for the image. The reason for this approach is to be able to better distinguish between images that are actual pornographic and images that are used for educational purposes. With the help of textual features the filtering rate could be improved to 70%.

The work of Harvey and Smith was further improved in [5]. Since they used little data for classification in their first paper, they extended their data and also used more classes which include the following: pornography, nudity, people, portraits, graphical images, and miscellaneous. Each category contained around 1800 pictures. To detect skin in the images, they use the approach from their first paper [9]. For classification, four different classifiers are compared: generalized linear model, k-nearest neighbor, multi-layer perceptron (MLP) and support vector machine (SVM). Five features are used which include: the fractional area of the largest skin blob, the number of skin segments, the number of colors in the image, the fractional area of the largest skin blob, and the fractional area that is accounted for by a face. The results show that the MLP worked best.

Zheng et al. [51] focus on shape-based detection of offensive images. The first step is the skin detection, where a multi-Bayes classifier is used. This classifier aims to capture the varying illumination conditions in the image scenes. A k-means clustering is performed to group the sample images into different clusters according to the average brightness and average chromaticity. For each cluster a skin probability map (SPM) is built following the approach of [22], with additional values for chromaticity and illumination. A pixel is denoted skin if $P(Skin \mid R, G, B, L, T) \geq \theta$, where L is the average brightness and T the average chromaticity. After skin detection, the regions are refined, using morphological operations. For the following shape-based classifier different features are used to capture the shape of the skin region, which was found in the first step. Three simple shape descriptors are eccentricity (the length ratio between the major and minor axes of the object), compactness (the ratio between the object's boundary and the object's area), and rectangularity (the ratio of the object's area to the area of the object's bounding box). Further descriptors are the seven normal moment invariants (Hu's moments) which are invariant to translation, scale and rotation and the Zernike moments. For the actual classification of shapes the authors compared the following as weak classifiers: Decision Stump, C4.5, SVM and MLP, which are

boosted with AdaBoost to improve the results. The use of a boosted C4.5 classifier achieved the best results with a true positive rate of 89.2% and a false positive rate of 15.3%.

The authors present further research results in [49] which result in a system they call Image Guarder. They focused again on a skin detection followed by a classification of the skin areas. The skin color detection works similar to their first approach, but the chromaticity is omitted for the skin model and the illumination is presented only by three levels: dark, normal, and bright. The texture of areas which match skin color is validated afterwards with a first order statistic texture descriptor. The local variance is used to measure the smoothness of the region in a moving window. If the variance is below a threshold the region is classified as skin. Only if a skin region is detected, the image will be processed further. Otherwise it will be classified as benign. In the following step different features for color, texture and shape are used for classifying adult images. Color features include the mean of skin color probability and variance, texture features include texture contrast and coarseness. The shape is described by skin region area, the region edge intensity and the Zernike moments [46]. The classification is performed by a SVM. The results achieve a precision of 76.5% for adult and 95% for benign images, performing slightly better than a C4.5 classifier.

Another approach was proposed by Rowley et al. in [35] which is used in Google's adult-content filtering mechanism being integrated in Google's search engine. Since a basic requirement for the use of a filter in a search engine is speed, they focused mainly on the processing speed of the approach which is the main reason they did not use shape descriptors. Also many of the following features are computed in a region of interest (ROI), which is a rectangle centered in the image, the size of 1/6 of the original image dimensions on all sides. Two kinds of features are distinguished: skin-dependent features and skin-independent features. To get the skin region in the ROI, the skin color model from Jones and Rehg [22] is used. That model is combined with Bayes rule to construct a skin probability map using a prior of 20% for skin color. For refinement of the map, morphological operations erosion followed by dilation are performed, which is done to reduce noise and gaps. The mean and standard deviation of the skin map in the ROI are then used as features. In the next step, connected component analysis is performed on a binarized version of the skin map, where the probability of skin is higher than 50%. The used features include the number of connected components, mean and standard deviation of the skin map within the connected components and the compactness of the connected components. These features aim to capture the compactness of body parts compared to the background of images that may contain skin color. After connected component analysis, the texture of the skin regions is analyzed. To get an approximation of the texture the Canny edge detector is applied to the gray-scale image. The following ratios are used as features. The first one gives a measure of the skin texture:

$$\frac{\text{number of edge pixels in connected component in ROI}}{\text{number of pixels in connected component in ROI}} \tag{1.1}$$

while the second one is a measure of how much of the image texture can be attributed to skin-colored pixels:

$$\frac{\text{number of edge pixels in connected component in ROI}}{\text{number of edge pixels}} \tag{1.2}$$

Since some materials that are recognized as skin have long, straight edges (e.g. wooden doors, bricks), the last skin-dependant feature is the number of distinct lines, that are found in the image.

The skin-independant features are used to describe properties of pornographic images, that cannot be covered with color information. The first set of three features describe general

image shape and size information. First, images that are smaller than 10×10 pixels the following features are not computed since these pictures are too small. A flag is used to express this case. The other two features are log of the number of pixels in the ROI, and the aspect ratio of the image. A useful indicator for classification between offensive and inoffensive images could be the detection of artificial images like graphs. The entropy of the intensity histogram can be used to express this:

$$-\sum_{i=0}^{255} P(i) \cdot \log_2(P(i)) \tag{1.3}$$

where $P(i)$ is the fraction of pixels in the image with intensity i. If an image has fewer distinct intensities present, it is more likely that this picture is an artificial one. The next set of features is used to describe the clutter in the image. An edge detection is used for this again. The features are the fraction of pixels in the ROI that belong to an edge, and the fraction of edge pixels of the whole image that are in the ROI. Finally a face detection is performed on the image, since it is a good indication for the presence of a person in an image. The number of faces in the picture and the fraction of skin pixels of the image that belong to the largest face are used as features. The classifier they used is a support vector machine. The training set consisted of 812 pornographic and 16,488 non-pornographic pictures. The test set had the size of 1,331 adult and 50,629 non-adult pictures. All of this images were gathered with Web downloads. The classification achieved a true positive rate of 50% with a false positive rate of 10%, or 90% true positives with 35% of false positives.

Arentz and Olstad presented an approach for classification of whole websites into offensive and inoffensive ones [3]. The decision is based only on the pictures contained on the site, not taking additional text features into account. The authors use the assumption that a website only displays pictures that are from one class. Therefore a website is classified as offensive, if the number of pornographic images $\Omega > \frac{N}{2}$, where N is the total number of images on the website. The authors show that under their assumption, the probability to misclassify a website is rather low, since $X > \frac{N}{2}$ images have to be wrongly classified. The classification of single images is mainly based on skin detection with shape description of skin areas. First, the RGB pixel values are transformed into YCbCr color space. A skin color range can be defined which represents the skin color and in the initial filtering process all pixels outside this color range are rejected. The remaining pixels are grouped into connected components. For each component the color histograms for Cb and Cr values are calculated. Also, a robust second-order texture descriptor is calculated, to describe the smoothness of each component. The next step aims to describe the component's shape which is done by a clockwise tracing of the outer borders of the component and storing the distance to the centroid. Afterwards a Fourier Transformation is performed on the normalized distance array and the first 28 coefficients are kept. All four descriptors are combined to a single feature vector, which are presented to a genetic algorithm for training. For the training 575 non-offensive images were used and 365 offensive ones were gathered by searching the web for English female names. The classifier achieved 92.1% correct classification rate on the training images and 89.4% on the evaluation set, which included 500 offensive and 800 inoffensice pictures. Portrait images are hard to classify correctly since they are similar to adult images. Of 20 websites (10 for each category) all were classified correctly.

Yoo introduced an intelligent adult image retrieval and rating system (AIRS) in [48] which uses a database that stores many offensive and inoffensive images. To decide whether a new image is pornographic or not, the ten most similar pictures are retrieved from the database. If the majority of these pictures is offensive, the new image will be classified as offensive as well. AIRS consists of three layers. The first is the query processing layer, the

second the indexing layer, and the third the model database layer. In the query processing layer the MPEG-7 descriptors [30] are extracted for the query image. The descriptors include the edge histogram descriptor, the color layout descriptor, and the homogenous texture descriptor. The distance between the new feature vector and the ones in the database is calculated. The indexing layer extracts the ten most similar images from the database, and a simple majority rule determines the class of the new image. The database layer contains four different kinds of pictures: pictures with naked female breasts, pictures with male or female genitals, pictures with explicit sexual actions, and inoffensive pictures. The system achieves a true positive rate of 99.25% with a false positive rate of 23%.

Kim et al. present an approach that can distinguish between five classes from inoffensive to offensive images [25]. They claim that in different cultures different kinds of images are perceived as being offensive, therefore systems are needed that can easily adapt to the required level of detection. The authors built a dataset with about 1700 pictures from each of the following classes: swim suit images (can be regarded as offensive in very strict cultures), topless images (can be regarded offensive in school environments), nude images (are likely regarded as offensive in many cultures), sex images (are regarded as offensive in most cultures), and normal images. For feature description, the MPEG-7 descriptors are applied which include Color Layout, Color Structure, Edge Histograms, Homogenous Texture, and Region Shape. Combinations of these descriptors are tested as well. The classification is performed by a neural network classifier, where the input layer consists of as many nodes as the descriptors dimension and the output layer consists of five nodes, one for each class. The network contains also two hidden layers with 50 nodes each. The results show that the Color Layout descriptor performs best for single descriptors, while the combination of Homogenous Texture with Color Layout works best for the combinations. The authors claim that the Color Layout may be the best feature for adult vs. normal image classification and the Homogenous Texture may be the best feature for swimsuit vs. topless image classification.

The most recent approach is presented by Deselaers et al. in [12]. In opposition to the other approaches which are mainly based on skin detection, they use a bag-of-visual-words (BOVW) model. These models are adapted from text classification in which a document can be described by the number of different words that are contained in the text. The same idea is adapted for images which can be represented by a number of patches that are then described with some kind of local descriptor. The first step of that approach is to create a vocabulary for the specific task that is built out of a training set of images. The authors use image patches which are extracted around difference-of-Gaussian interest points. These patches are then transformed by PCA and the first 30 coefficients are used for description of the particular patch. The main purpose of this is to reduce the dimensionality. The authors also claim that the direct use of the patches is more appropriate than the use of SIFT features, since color information is included. The creation of the vocabulary is done by a training of Gaussian mixture models which is able to capture frequently occurring patterns in the training data. Since an image is represented by a set of local features and each feature is described by an identifier of the closest Gaussian density, a histogram over all identifiers for each image is created. This histogram can be used as a feature vector for the following classification. Support vector machines and log-linear models are compared for the classification task. To get a better comparison to other approaches, they test their methods on the dataset that was presented in [25]. Following the idea of creating a system that is able to distinguish between different classes of offensive images, they also adapt a filtering rule. This rule allows to define which categories are regarded as offensive and the system filters the images belonging to these categories. The results show that the system can distinguish between the pornographic and inoffensive images without many misclassifications, while the classification of other categories is harder. Also the support vector machine performed

slightly better than the log-linear model. The authors compared the system on images that were downloaded from the Web. These images were harder to classify than the images of the dataset.

Table 1.1: *Performance of different adult image detecting systems reported by their authors in the given paper. The best results achieved the BOVW approach with a high true positive and the lowest false positive rate.*

Method	TP	FP
Skin detection + geometrical features [17]	42.7%	4.2%
WIPE$_{TM}$ [45]	96%	9%
Bayesian SPM with NN [22]	85.8%	7.5%
Bayesian SPM with NN and text [22]	93.9%	8%
Bayesian SPM with SVM [35]	50% 90%	10% 35%
Image Guarder [49]	76.5%	5%
AIRS [48]	**99.25**%	23%
Bag-of-visual-words with PCA and SVM [12]	**99.02**%	**0.05**%

In summary, most of the approaches just use skin detection, followed by shape description and a classification that is based on features, that are extracted in the steps. While these methods seem to work well, the most recent one using a bag-of-visual-words outperforms the skin detection based methods according to the authors. An overview of the performance can be found in Table 1.1. However, these rates are reported by the authors themselves and therefore cannot be directly compared to each other. This is one major problem regarding the research on classification of adult images. Because there does not exist some kind of standard dataset which allows researchers to compare their systems. Each of the authors use their own data which they mostly downloaded from the Internet. Unfortunately, there does not exist a survey which compares the different methods on the same data.

1.2.2 Comparison of Different Skin Detection Techniques

The previous section presented some of the existing methods for classifying offensive images. While most are based on finding human skin in the image, there exist different methods to do so. Skin has some properties that should make it possible to distinguish skin color from the color of other materials. Some researchers try to transform the pixel values into another color space, hoping that the separability between skin and non-skin increases. Because skin has a high reflectance its color may appear different under varying lightning conditions [2]. To cope with this, another idea is to drop the illumination component. This topic has been discussed in the literature. Since it may be vital for the success in detecting offensive images, the results from other researchers are presented in this section. Three surveys are mentioned which use different metrics to evaluate the use of color spaces under different methods.

Shin et al.[36] evaluated the following color spaces: RGB, normalized RGB, CIEXYZ, CIELAB, HSI, SCT, YCbCr, YIQ, and YUV. For each color space the illumination component was dropped in additional experiments. They claim that a transformation into a color space other then RGB is a pre-processing step whose goal is to increase the separability between skin and non-skin pixels while at the same time the separability between different skin tones should be decreased. The experiments were performed on a dataset of 805 images. Images that contain skin are from the AR and the UOPB face datasets while images with no

skin pixels are from the University of Washington's content-based image retrieval database. Two different kinds of measures are used. One is based on a scatter matrix and the other one on histogram comparison. The scatter matrix based metric include the scatterness within and between clusters. The histogram comparison calculates the intersection between skin and non-skin pixel color histogram and the histogram χ^2 error. The results show that the RGB color space performed best in most of the experiments. Further results show that the dropping of the illumination component did not improve the separability but it decreased the separability in 3 out of 4 measurements.

Another comparison of different skin color detection methods was performed by Vezhnevets et al. [42]. In contrast to Shin's work, they compared different color spaces and different detection methods simultaneously. The color spaces include RGB, normalized RGB, HSV, TSL, and YCbCr. The goal of each skin model is to define a point in the color space, whether it belongs to skin color or not. The methods are distinguished into three different kinds: explicitly defined skin regions, nonparametric skin distribution modeling, and parametric skin distribution modeling. The explicit defined skin region defines a color range in the used color space. Every pixel that falls into that range is denoted skin. This model can be expressed via a set of simple rules. An advantage of this method is its simplicity and its speed. However, it can be extremely difficult to find the right color range and the right color space. The nonparametric methods try to built a skin model out of a given training set but without the use of an explicit model. Jones and Rehg's skin model falls into this category, where the probability distribution of a pixel being skin is estimated with histograms. Advantages of these models are the speed and the independence from the shape of the skin color distribution in the particular color space. A disadvantage is the required storage space which can be quite large. Also does the result highly depend on the training data, since the ability to generalize is not given. The parametric models try to fit an explicit distribution on the training data to define the skin color probability. The skin color distribution for example, can be modeled with a mixture of Gaussians that can describe rather complex shapes. An advantage of these methods is the lower required space, but the goodness can depend on the used color space. Also the training consumes more time than the nonparametric models.

An overview of the performance of the different methods can be found in Table 1.2. The best performance was achieved with a Bayes SPM in RGB color space and the Maximum Entropy Model in RGB color space. The authors also found out, that parametric modeling methods are better suited if the training data is limited, since their ability to interpolate and generalize training data. Non-parametric methods are less dependent on skin cluster shapes and are therefore more promising for large target datasets. Finally the authors state, that the evaluation of a color space regarding its performance on skin detection, cannot be made in general, but does highly depend on the underlying method.

Another comparative study was presented in [1]. The authors compared many different color spaces in use for skin detection with lookup tables similar to the Bayes SPM. In their study they evaluated the color spaces on 2,284 downloaded offensive images from the Web from different categories like: indoor and outdoor shots, single and multiple persons in the image, professional and amateur shots. The HSV color space performed best with a true positive rate of 93% and a false positive rate of 5.8%, directly followed by RGB with 91% true positives and 5.9% false positives.

To summarize the results of the surveys, the choice of color space is not crucial for the performance of a skin detector. The RGB color space can be used, which has an advantage because it does not need additional time for transformation and it show good performance in already used experiments.

Table 1.2: *Performance of different skin detectors reported by the authors from [42]. According to their evaluation, the Bayes SPM, and the Maximum Entropy Model in RGB color space achieve the best combination of true and false positives.*

Method	TP	FP
Bayes SPM in RGB	**80**% **90**%	**8.5**% **14.2**%
Maximum Entropy Model in RGB	**80**%	**8**%
Gaussian Mixture Models	80% 90%	$\sim$ 9.5% $\sim$ 15.5%
SOM in TS	78%	32%
Elliptical boundary model in CIE-xy	90%	20.9%
Single Gaussian in CbCr	90%	33.3%
Gaussian Mixture in IQ	90%	30.0%
Thresholding of I axis in YIQ	**94.7**%	30.2%

1.2.3 Detection of Offensive Videos

While there already exist numerous of methods for classification of offensive images, there are only few methods for detecting adult videos so far. A video basically can be regarded as a set of images that have an additional time dimension. Therefore the techniques that exist for image classification can also be applied on the video data what is actually done in most of the approaches.

The approach of Lee et al. [26] compares two kinds of features which are based on color information of keyframes. The keyframes are extracted at regular intervals. The first feature is just based on one keyframe. A SPM is calculated in RGB color space, using a Gaussian Mixture Model. The map is downscaled to a 40 $\times$ 40 pixel resolution and then used as a feature vector for classification with a SVM. The keyframes used for the training process were selected manually out of the total amount of extracted keyframes for both offensive an inoffensive videos. The second feature is based on a group of keyframes. For each frame in the group a HSV color histogram with 256 bins is computed. The final feature is the accumulated and normalized histogram over all frames. For this feature a SVM is used again for the classification task. Finally both features are combined and used with a linear discriminant function for classification. The system achieved an accuracy of 91% on the test data and performed better than each of the single features. Also the group of frames feature achieved better results than the feature which is based on one keyframe. The data that was used for training and testing consisted of different video files from different genres.

The method in [24] by Kim et al. is based on a shape detection of skin areas in video frames, that are not title frames or shot boundaries. To detect title frames, a color histogram of the frame is computed. If a certain color range is high, the frame is regarded as a title frame and not used for the following steps. A shot boundary is the boundary between shots of the video. There exist three different kinds: hard cuts, fade in/out, and dissolve. The authors use a simple color difference histogram between two adjacent frames. This is a simple method based on the idea that the color distribution within one shot does not change as much as across shots. If a frame is detected as shot boundary, it is rejected. In the next step, the global motion is estimated. Global motion is the motion that evolves out of camera movement like rotation, zoom, and tilt. The estimation of motion vectors is done with a nearest neighbor approach within 16 $\times$ 16 macroblocks. If vectors are similar

to one belonging to global motion, the frame is regarded as containing global motion. These frames are rejected as well. The remaining frames are segmented into areas with similar color. These areas are analyzed if they resemble skin color. They use a manually defined color range for deciding which pixel belongs to skin. For each segment that contains skin, the normalized central moments are calculated to describe the segment's shape. The classification is done by calculating the weighted Euclidean distance to sample moments from a training database that contains both offensive and inoffensive sample videos. The evaluation was performed on 2,275 inoffensive and 980 offensive videos. A true positive rate of 96.5% and a false positive rate of 31.5% were achieved.

In [33] a method is presented by Rea et al. which uses additional motion and audio features as well as skin color detection. The idea is to estimate the foreground by combining skin detection and the localization of local foreground movement. First, a skin map is built for each frame, using the Jones' method [22]. The second step extracts the MPEG motion vectors. The vectors that belong to the global background movement are compensated. The remaining vectors are clustered with k-means and such segmenting the frame into areas that belong to the foreground and those that do not. The result from the skin map and the motion segmentation are combined. As additional feature the periodicity in the audio stream is proposed. The authors claim that obscene scenes can be recognized by recurring sounds and therefore are distinguishable if periodicity can be found in the audio signal. Periodicity detection is performed by locating local maxima and minima in the autocorrelation function of the audio energy. The measure is the area between the lines through the maxima and minima. For obscene videos the area should be larger than for the inoffensive videos. This method is also proposed for detecting periodicity in motion. The methods were tested on a sample movie but no evaluated on a larger dataset so there are no classification results given.

The presented methods are mostly based on color information in selected keyframes. Only one approach tries to use other informations like the audio stream and motion. This method, however, is not evaluated on a larger dataset. Especially in this area of research, there is still room for improvement which is indicated by the few existing approaches. As for the problem of classifying offensive images, there does not exist some kind of standard dataset, which can be used by every researcher to get comparable results.

Chapter 2

Background

This chapter covers the background of the used techniques. It should give additional information to understand underlying methods to our approach which, is explained in detail in the following chapter. The first section covers the construction of a *skin probability map* (SPM) as it was presented by Jones [22]. The second section introduces the *bag-of-visual-words* approach and additionally covers basics for the descriptors that will be used later on. In the successive sections the classifiers being used are presented: the *decision tree* and the *support vector machine* (SVM).

2.1 Bayesian Skin Probability Map

The initial step of one of the approaches is based on detecting skin inside an image. Some of the existing methods were already described in the previous chapter. Comparing studies showed that the approach of Jones [22] achieved good detection rates with low computational costs. This method is also used in systems that are designed to classify offensive images, for example in [35, 51, 20]. This section covers backgrounds on these histograms, because they are used in the presented approach as well.

The main idea is to estimate a skin color probability distribution with histograms in RGB color space. For each pixel with value rgb, the goal is to get the probability of that color value belonging to skin color $P(\text{skin}|rgb)$. This probability distribution can be estimated by Bayes rule:

$$P(\text{skin} \mid rgb) = \frac{P(rgb \mid \text{skin})P(\text{skin})}{P(rgb \mid \text{skin})P(\text{skin}) + P(rgb \mid \neg\text{skin})P(\neg\text{skin})} \tag{2.1}$$

where $P(rgb \mid \text{skin})$ denotes the distribution of rgb values of skin pixels, and $P(rgb \mid \neg\text{skin})$ denotes the distribution of rgb values of non-skin pixels. Both can be estimated with histograms from a given training set with labeled skin and non-skin pixels in the following way:

$$P(rgb \mid \text{skin}) = \frac{s[rgb]}{T_s} \tag{2.2}$$

$$P(rgb \mid \neg\text{skin}) = \frac{n[rgb]}{T_n} \tag{2.3}$$

where $s[rgb]$ denotes the number of skin pixel counts in the histogram bin associated with the RGB value rgb and T_s the total number of skin pixels. Analogous $n[rgb]$ is the number of non-skin pixel counts of colors rgb and T_n the total number of non-skin pixels. The number of bins is 32 per channel.

The prior probabilities $P(\text{skin})$ and $P(\neg\text{skin})$ can be computed directly from the dataset by the whole number of skin and non-skin pixels encountered. However, it is not necessary to use a prior, if one only is interested in what probability is higher. For that purpose the ratio of the posteriors can be used:

$$\frac{P(skin \mid rgb)}{P(\neg skin \mid rgb)} = \frac{P(rgb \mid skin)P(skin)}{P(rgb \mid \neg skin)P(\neg skin)} \tag{2.4}$$

which then can be compared to a threshold $0 \leq \theta \leq 1$:

$$\frac{P(rgb \mid skin)}{P(rgb \mid \neg skin)} \geq \theta \tag{2.5}$$

Here, a pixel is regarded as belonging to skin, if the ratio exceeds the threshold. The choice of this threshold can be estimated as a trade-off between costs of false positives and false negatives:

$$\theta = \frac{c_p(P(\neg\text{skin}))}{c_n(P(\text{skin}))} \tag{2.6}$$

where c_p denotes the costs of false positives and c_n denotes the costs of false negatives. This shows that the priors are not needed, since the costs can be altered to get the same results with different priors. The threshold θ can be evaluated out of a labeled training dataset of skin and non-skin pixels and set to meet the required classification rates.

2.2 Bag-of-visual-words

The *bag-of-visual-words* method became more popular in computer vision tasks, including object recognition [11], visual categorization [10, 28], and even the classification of adult images [12]. The idea is roughly based on models for text classification, where a text is categorized by the occurrence of certain words, which are captured in a *vocabulary*. Analogous to a specific document type which might contain certain frequent occurring words, images that show similar scenes might contain frequent occurring local areas that look alike. These local patches are referred to as *visual words*. The idea is to learn a *vocabulary* of *visual words* out of a set of images that fall into the same category, where each of the local patches is extracted and described with some kind of local descriptor. Since all images of the same category should have a similar distribution of visual words, a histogram of the occurring patches can be used as a feature vector for classification. The focus on local image information should make the system more robust against partial occlusion, deformation, and clutter.

The bag-of-visual-words method mainly consists of the following steps:

- Extraction and description of local image patches
- Creation of the *vocabulary* or *codebook*
- Construction of a *bag of visual words*
- Classification of images

For sampling of local image patches, two techniques are mostly used in recent work: sampling around a set of interest (salient) points, or sampling on a regular grid. One prominent method to detect salient points is the Harris-Laplace detector [31] which responds to regions that cover corners. Therefore homogenous regions in the background are often ignored when creating a vocabulary. While salient points may be advantageous for object recognition, a regular sampling might benefit scene classification [41, 28].

Another important decision is the descriptor to use for the local patches. The descriptor should capture enough information about the patch, making it discriminative enough on category level. However, it should also be invariant against variations like image transformations, and lighting variations. A descriptor which is frequently used [10, 50, 28] is the Scale Invariant Feature Transform (SIFT) [29]. It describes the shape of a region using histograms of local gradients in a 4×4 pixel neighborhood around the sample point. 8 bins are used for the histograms, leaving a 128 dimensional vector. The descriptor is invariant to image transformations translation, rotation, and scaling and also to light intensity changes. However, a disadvantage might be, that no color information is included. To cope with this disadvantage some modifications came up which include additional color informations. One possibility is to compute the SIFT features for each color channel instead of only a gray image. Another possibility is to concatenate color histograms to the SIFT features. The different advantages and disadvantages of such features are discussed in [41]. Instead of using some descriptor for the local patches, some approaches just use the gray-scaled image patches [28] themselves. Since the dimensionality of a whole patch might get very high, some dimensionality reduction might be applied, for example with Principal Component Analysis (PCA) [11].

After the local patches are extracted and described, the next step is to create a vocabulary. The vocabulary is basically a set of visual words, which occur in the sample images. Since the patches do not exactly occur in the images, similar patches have to be grouped together, to form a single visual word. One prominent technique is to cluster the patches into k clusters with *k-means* [14], where k is the size of the codebook. Each visual word is then represented by the mean of its cluster. In [12] another method for creating the vocabulary is described, which is based on the training algorithm for unsupervised learning of Gaussian Mixture Models.

With the existing vocabulary the bag of visual words can be created for each image. In this step, basically a histogram is created that counts the occurrence of each visual word in an image. This is done by extracting and describing the local patches in the same way as for the construction of the codebook. For each patch, its visual word is determined which can be done by i.e. calculating the Euclidean distance to each cluster center. This creates a k dimensional feature vector.

The feature vectors can then be used as input for any classifier. Various different classifiers were already tested in some papers, for example nearest neighbor, naive bayes, gaussian mixture models, log linear models, and support vector machines (SVM). Especially the support vector machine became popular and achieved good classification results with the χ^2 distance Kernel [50].

2.2.1 Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) expresses a discrete input signal as a sum of cosine functions with different frequencies and amplitudes [23]. In recent years it became quite popular for image compression, and it is used for example, in the JPEG standard. Here, the image is expressed by the low frequency components while the high frequencies are discarded, losing only little information. Therefore, the DCT can also be used to reduce the dimensionality of a local image patch. Only the low frequencies are used to describe the patch, and therefore the descriptor is discriminative enough.

The 1D-DCT of a signal $x = x_0, ..., x_{N-1}$ of length N is defined as:

$$C_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \tag{2.7}$$

for $k = 0, .., N-1$. Since images are 2 dimensional signals, the DCT has to be extended to a second dimension. The 2D-DCT for a $N \times M$ dimensional input signal I is defined as

follows:

$$C_{k,l} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I_{x,y} \cos\left[\frac{\pi}{N}\left(x+\frac{1}{2}\right)k\right] \cos\left[\frac{\pi}{M}\left(y+\frac{1}{2}\right)l\right] \tag{2.8}$$

A nice property of the DCT is its separability, which means that the 1-D DCT can be performed first, on the rows and then on the columns, saving computational time. Therefore the equation 2.8 can be expressed as:

$$C_{k,l} = \sum_{x=0}^{N-1} \cos\left[\frac{\pi}{N}\left(x+\frac{1}{2}\right)k\right] \sum_{y=0}^{M-1} I_{x,y} \cos\left[\frac{\pi}{N}\left(y+\frac{1}{2}\right)l\right] \tag{2.9}$$

With this adaption the number of operations can be lowered from $O(N^2)$ to $O(N \log N)$.

2.2.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a technique to calculate the most meaningful basis to a dataset [37]. It can also be used to reduce the dimensionality of data. The principal components of n samples x_i in an m-dimensional space can be calculated in the following way. First, the mean value μ over all samples x_i is subtracted from each sample: $y_i = x_i - \mu$ and stored in a matrix Y. Then the covariance matrix $C = \frac{1}{n} YY^T$ is calculated. The diagonal values in the covariance matrix store the variance of particular measurement types, while the values of the diagonal axis express the covariance between measurement types. Now the covariance matrix has to be diagonalized, leaving the eigenvectors and eigenvalues of C. Since the largest eigenvalue corresponds to the eigenvector with the largest variance which is the first principal component. So the eigenvectors are sorted according to their eigenvalues, leaving a new orthogonal basis to the original data. The PCA can also be calculated with single value decomposition (SVD).

The original signals s_i can then be expressed in as a linear combination of the principal components:

$$s_i = \sum_{k=1}^{m} \alpha_{i_k} p^k \tag{2.10}$$

where p^k denotes the k-th principal component and α_{i_k} is the correlation coefficient for the k-th component for the i-th signal. Since the principal components are ordered according to their influence in the data, higher coefficients tend to get small. Therefore, for dimensionality reduction, a signal can be expressed by using only the first $l < m$ principal components, instead of all of them. This is done by using the following steps: First, out of a set of sample signals a PCA model is learned in the way it is described above. Then a new sample is transformed into the new basis, leaving the coefficients $\alpha_{i_1}, ..., \alpha_{i_l}$. These coefficients are used as a new signal, which has a lower dimension than the original one.

2.2.3 SURF

SURF (Speeded Up Robust Features) [4] is a similar technique for interest point detection and description like SIFT. Its goal is the *detection* and *description* of local points in an image. An important requirement for the detector is, that it finds the same points under different viewing conditions. The descriptor should be robust to noise, detection errors and invariant to image transformations, while also being distinctive enough. SURF focuses hereby on scale and rotation invariance. The detector is based on the Hessian matrix and uses integral images to get an approximation with high processing speed. The detector is based on Haar-wavelet-responses which are also approximated with integral images. Integral images were first proposed by Viola in [43]. They can be used as a fast implementation of

box convolution filters. The integral image $I_\Sigma(x, y)$ displays the sum over all pixels from the origin to the point (x, y): $I_\Sigma(x, y) = \sum_{i \leq x} \sum_{j \leq y} I(i, j)$.

The detector is based on calculating the determinant of the Hessian matrix at different locations and scales. Usually, the entries in a Hessian matrix are the partial derivatives of a function f. For an image the derivatives can be expressed by a second order normalized Gaussian filter, which is dependent on the position (x, y) and a scale factor σ. In SURF, the second order Gaussian derivatives are approximated by using box filters which can be calculated with integral images at low computational costs. Different scales are evaluated by using larger box filters instead of scaling the image down. A local interest point corresponds to maxima of the determinant of the approximated Hessian matrix.

The descriptor describes the intensity distribution within a scale dependent neighborhood around the local interest point. Gradients in the x and y directions are found by using Haar wavelet responses. These wavelets can be calculated with the integral images and are therefore fast to compute. A local rectangular patch around the interest point is defined with the size at the particular scale at which the point was found and the orientation of the point. Then each patch is divided into 4×4 sub-regions. In each of these regions the Haar wavelets are calculated at regular intervals for x and y directions, denoted d_x and d_y. The features for each sub-region are the sum of d_x and d_y and the sum of absolute values $|d_x|$ and $|d_y|$. As a whole, the SURF feature vector for a local image patch consists out of 64 values.

An extension of the the SURF feature vector is SURF-128. It further splits the sums of d_x and $|d_x|$ according to $d_y < 0$ and $d_y \geq 0$, and vice versa. Therefore the number of features is doubled, which creates more distinctive features, but may be slower to compare. One important characteristic of the SURF descriptor is, that it does not include color information, since it focuses on the intensity distribution of the patches.

2.3 Decision Trees

Decision trees are classifiers that partition the feature space into subspaces, where each of the subspaces should correspond to one class [14]. A simple example for a decision tree is shown in Figure 2.1. Here two classes are distinguished by a two-dimensional feature vector. A non-terminal node represents a *rule* (or *split*) which is of the form $x < v$ or $x \geq v$, where x is the tested feature value, and v a threshold value. If the test at a node is true, the rule at the following node is evaluated. In the sample tree, the left child is the successor if the rule did apply, and the right child otherwise. If a terminal node is reached, the final decision is made according to the label at that node. Because the splits have the described form, the feature space is partitioned into rectangular subsets. The feature space to this sample tree is shown in Figure 2.2. Each class is represented by ten samples. The dashed lines show the rule boundaries for each of the splits.

The construction of a decision tree is the process of building the tree on a set of training samples. This process can be regarded as a recursive process, where the feature space is continuously split into subsets. So there are basically two important questions to answer: Which are good splits of the feature space and when decide to stop splitting. The following steps are explained for *Classification and Regression Trees* (CART) [6] which are used as classifiers later.

Good splits are splits that separate the space into smaller subsets which are "purer" than the parent set. A measure is defined which expresses how pure a node is: the *impurity* $i(t)$ of a node t. It has its maximum value if each class is represented with an equal number of samples in the resulting subset and it is zero if only samples of one class are represented in the subset. Therefore the quality of a split s at node t can be defined as the decrease in

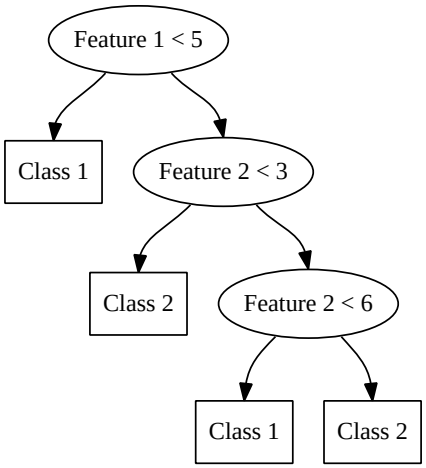


Figure 2.1: *Decision tree for a two-class separation problem in a two-dimensional feature space. Some sample points in the feature space can be found in Figure 2.2 with the decision rules given by the tests in the nodes of the tree.*

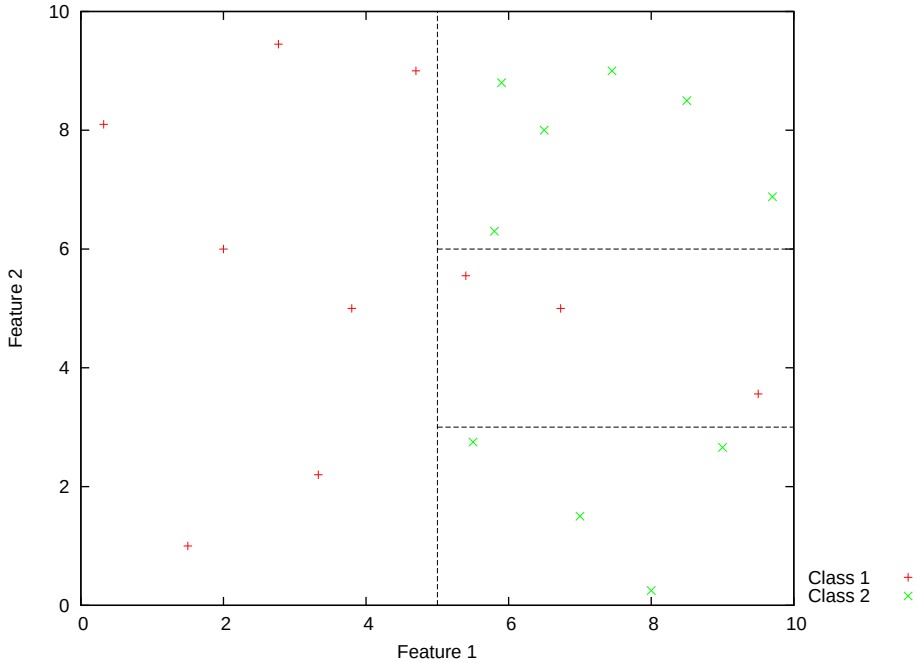


Figure 2.2: *Two-dimensional feature space with ten samples of each of the two classes. The dashed lines show the decision boundaries given by the tree in Figure 2.1.*

impurity:

$$\triangle i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{2.11}$$

where t_L and t_R are the left and right descendant nodes, $i(t_L)$ and $i(t_R)$ their impurities, and p_L, p_R the fractions of samples at the particular nodes. A possible measure for the impurity is the *entropy*. The construction of a tree starts with the root node where all possible splits are evaluated over all possible features. The one with maximum decrease in impurity is chosen. This process continues for the child nodes until the impurity at a node cannot be decreased further and that node is then declared a leaf. Now a class label has to be assigned to that node. A class assignment rule as follows is used: $j^*(t) = j$ if $p(j|t) = \max_i p(i|t)$, where $p(i|t)$ denotes the fraction of samples of class i in the subset of node t. So the class label of the class with the most samples in the subset which belongs to the terminal node is used as classification result. The fraction $p(j|t)$ of samples can be used to express a probability score for the classification result.

By this, a tree can be constructed with no impurity at each terminal node. However, this is not what one actually wants, because this tree would resemble a look-up table which is specialized on the training data and might not be able to generalize to test data. One way to select an appropriate time to stop splitting is to use a stopping rule. A small threshold $\beta > 0$ is introduced and the splitting at node t is stopped, if $\max_{s \in S} \triangle i(s,t) < \beta$, where S denotes all possible splits at node t. So if the impurity at a node cannot be lowered by a minimal, previously defined amount, the set is not split up further.

A more appropriate method to get a final tree is *pruning*. Pruning is the process of making a tree smaller, after it has been fully grown. Two terminal nodes are fused together, if the impurity rises only little, if the according split is not used. Both nodes are then deleted and the parent node is declared a leaf.

CART uses *minimal cost-complexity pruning*. A *cost-complexity* parameter α is introduced, which compares the tree size to its misclassification rate in the following way: $R_\alpha(T) = R(T) + \alpha|T|$, where $R(T)$ denotes the misclassification rate of the tree and $|T|$ the size of the tree. α is used to punish large subtrees that do not improve the classification rate much, according to their size. For the actual pruning, a sub-branch T_t of the tree is found for which $R_\alpha(T_t)$ is bigger than the misclassification rate at node t, which is the root of the sub-branch. This branch is cut away, leaving t as a terminal node. The process is continued until no branch can be found for which the pruning can be applied.

After training, a decision tree can be applied to new data samples in the way it was described above. An additional nice property of decision trees is, that simple rules are defined which can be easily interpreted by the user. This is in direct contrast to black box classifiers, like artificial neural networks, where no such rules can be extracted. A rule can be extracted out of the tree by combining all tests from the root to a terminal node.

2.4 Support Vector Machines

Support vector machines (SVMs) are based on a linear discriminant function. They aim to get a better generalization by maximizing the *margin*, the distance from the separating *hyperplane* to the closest data samples. Additionally, for non-separable cases, a *kernel* is used, that transforms the sample points in another space, where the separability is higher.

Considering a binary classification task with n-dimensional feature vectors x_i and class labels $y_i = \pm 1$, a linear discriminant function has the following form:

$$f(x) = \text{sign}(w \cdot x + b) \tag{2.12}$$

A correct classification can be achieved if

$$y_i(w \cdot x_i + b) > 0 \ \forall i \tag{2.13}$$

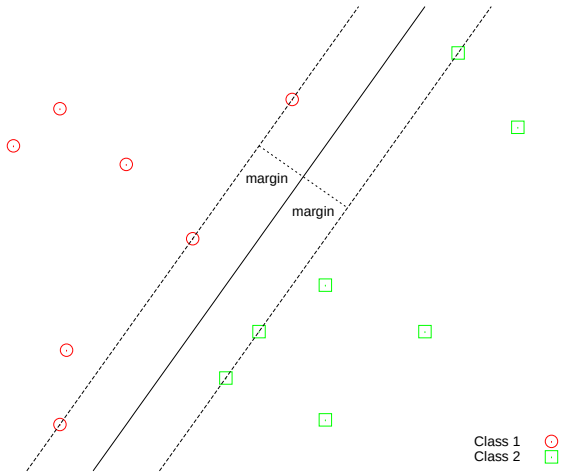


Figure 2.3: *Sample points from two different classes are separated by the separating hyperplane indicted by the solid line. Canonical hyperplanes (dashed lines) are parallel to the separating hyperplane and go through the closest sample points, the support vectors. The margin is given by the distance of a canonical hyperplane to the separating one.*

The separating hyperplane is given by $w \cdot x + b = 0$. For the closest points to that hyperplane, *canonical hyperplanes* can be defined, such that $w \cdot x + b = 1$ holds for all points on the positive side and $w \cdot x + b = -1$ for all points on the negative side. These points are called the *support vectors*. The margin is then given by a projection of these points to the normal vector of the hyperplane $w/||w||_2$. Therefore, the margin is $1/||w||_2$. Figure 2.3 shows the separating hyperplane, the canonical hyperplanes, and the margin for some sample points. The goal now is to maximize the margin by minimizing $\frac{1}{2}||w||_2^2$ subject to the constraints defined in equation 2.13. The learning task then can be reduced to a minimization of the primal Lagrangian [7, 8].

Since applying a linear discriminant function benefits of linear separable data, which might not be the case in most of the applications, a function is introduced, which maps the features into a higher dimensional space. This space might be of infinite dimension. However, it is not needed to know the exact transformation, because it can be implicitly defined by a *kernel* $K(x_i, x_j)$. A more detailed description of classification with kernels and SVMs can be found in [7].

There are many possible choices for such a kernel function. One prominent choice is the *Radial Basis Function* (RBF) Kernel which is defined as follows:

$$K(x_i, x_j) = e^{-\gamma||x_i - x_j||^2}, \gamma > 0 \tag{2.14}$$

Another kernel is the χ^2-distance Kernel which we will use for classifying histograms of local image patches. Previous results showed its good performance for this application [50].

$$K(x_i, x_j) = e^{d_{\chi^2}(x_i, x_j)/\gamma} \tag{2.15}$$

with

$$d_{\chi^2}(x_i, x_j) = \sum_{k=1}^{n} \frac{(x_{ik} - x_{jk})^2}{x_{ik} + x_{jk}} \tag{2.16}$$

For the actual training process, the steps from [18] are followed. Training is actually a

optimization problem, which has the following form:

$$\min \ \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{l}\xi_i$$

subject to the constraints defined in equation 2.13, where l denotes the number of training samples, and $C > 0$ is a penalty parameter of the error term ξ. The actual training process starts with scaling of data into the range of $[-1, 1]$, or $[0, 1]$. Then a grid search over the parameters C and γ of the SVM is performed, using cross-validation. γ is a kernel parameter, and C the penalty term defined above. For v-fold cross-validation, the training set is split into v subsets of equal size. The classifier is trained on $v-1$ of these sets, while the remaining one is used for testing. During the grid search different parameter combinations are tested and the best performing one is used for final training, for which the whole training set is used.

Chapter 3

Approach

This chapter deals with the approaches that were used to classify offensive images and videos. It is divided into methods that are used for image classification and methods which are used for video classification. Since videos can be regarded as a set of images, some of the techniques that are used on images, are used for video classification as well. Additionally, motion information is used to further improve the separability between offensive and inoffensive material.

The main goal of the methods is to built a system that classifies offensive images or videos. Therefore, the most important requirement is to get good classification results. Usually, one wants to maximize the correct classifications and minimize the misclassifications. However, there are other requirements that may be of need, depending on the actual task. One goal may be to lower the computational effort which is needed to perform the classification. Depending on the task, this might become an important requirement, too. For example, in use for content based Web filtering, the processing speed for a new image has to be very fast, because the user wants to access a website directly and not to wait some additional time until a result shows up. In order to find all offensive images on a hard drive, the duration of the process is not that important. Further requirements, like usability might exist as well. However, the main focus is on classification performance, while leaving the processing speed as secondary.

All of the approaches are based on a general processing pipeline for classification tasks presented by Duda et al. in [14]. A slightly adapted pipeline is shown in Figure 3.1. The first step consists of some *preprocessing* methods. These steps are usually performed to make the data more suitable for later classification. Methods to make the data more similar, for example by scaling all images to a fixed resolution, or the transformation into a more appropriate color space, might be included in this step. A goal of preprocessing is to increase the separability between the classes of the dataset.

The successive step is the *feature extraction*. Instead of using whole data samples, i.e. whole images, features are extracted out of the samples for the classification task. A feature should capture distinctive information in a way, such that the values for samples from the same category are similar and different for samples of different categories. Furthermore, features should be invariant to transformations such as translation, rotation and scaling. Feature based representations are also preferred because they tend to have a lower dimensionality than the whole data samples, which saves computational effort. The features are stored in vectors that are fed to a classifier. A classifier is *trained* on a training set which consists of samples of the data that should be classified later. The choice of the training data and the number of samples influence the performance of the final classifier. The final classifier is *evaluated* on a separate evaluation set after the training is completed to measure its performance.

The following subsections describe the realization of the first two steps for the different approaches, while the training and testing are described in more detail in the following chapter.

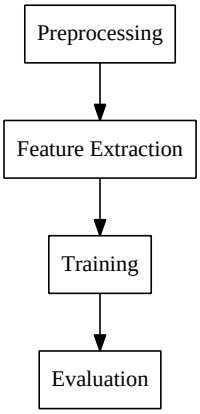


Figure 3.1: *The general pipeline for image classification consists of four successive steps. First, the data is preprocessed for the later steps. Then, features are extracted and used to train classifier. In the final step, this classifier is evaluated.*

3.1 Methods for Classifying Offensive Images

The literature presented mainly two different techniques that were used for detecting offensive images: skin color based methods and methods that were based on local patches. The first one was used by most researchers while the latter one was recently presented by Deselaers in [12]. However, the visual words approach outperformed the skin color based approaches. Therefore, the skin color based approach is just based on simple features and mainly used here for comparing the visual words approach to skin color based methods.

The same preprocessing step is used for all images before any features are extracted. Each image which has a larger height than 250 pixels is downscaled such that the height is 250 pixels while the aspect ratio is preserved. This is done to save computation time if an image is too large.

The second reason is the regular sampling method used for the bag-of-visual-words approach. If an image is much larger than the others, an extracted local patch contains a much smaller region than it would in a smaller image. Since the local patches are clustered according to their similarity, using images with varying sizes might ruin the idea of getting patches that correspond to certain reoccurring patterns in images. Therefore, images which are too large are scaled down to get rid of this problem.

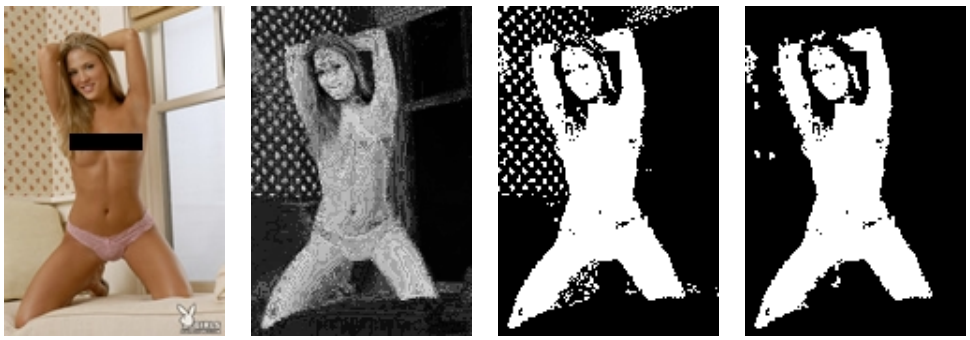
Since two different methods for the classification task are presented, both methods are fused to see if the performance can be improved by combining the results of the two approaches. This is done by a *late fusion* stage, meaning that the classifier output of the two methods is combined instead of combining features before the actual classification.

The section is divided as follows: First, the approach based on skin detection is described. The second section covers the bag-of-visual-words approach, while the last section is about the fusion of the results from the previously described methods.

3.1.1 Features Based on Skin Detection

The first approach is based on skin detection with simple features. The idea is to implement a baseline system that follows closely the existing methods for offensive image classification.

Basically, the following steps are used: First, a skin probability map (SPM) is constructed. Then the map is binarized leaving either pixels that belong to skin or pixels that do not. Afterwards, a step is included to remove noise. Finally, five features are evolved out of these steps that are used for classification.



(a) original image (b) skin probability map (c) binarized SPM (d) binarized SPM after noise removal

Figure 3.2: *Three steps are applied to detect the skin areas in an image. First, a skin probability map is constructed by using color histograms. The resulting map is thresholded to get a binarized version. In the last step, noise is removed.*

Figure 3.2 shows the outcome of the main steps. The first image (Figure 3.2(a)) shows the original image. This image contains skin as well as other materials in the background which resemble skin color. First, the SPM of the original image is created using color histograms in RGB color space. The same histograms are used as were presented by Jones in [22]. A more detailed description was already given in section 2.1. The probability of each pixel being skin is displayed with a brightness value in the skin probability map. For high probabilities the pixel appears brighter. If a pixel is white, the probability equals 1 and if a pixel is black, the probability is 0. The result for the sample image can be found in Figure 3.2(b). The brightest parts are the body and some parts of the tiles in the background. Some parts of wood in the background as well as the hair have a lower probability of belonging to skin.

In the next step a binarized version of the SPM is constructed. To do this, a global threshold θ is used. If the value of a pixel exceeds this threshold, the pixel is marked as skin and the pixel is set as white. If the probability is below this threshold, the pixel gets black. θ is estimated in a way that $\theta = \frac{\mu_1+\mu_2}{2}$ holds, where μ_1 is the mean gray value of skin pixels and μ_2 the mean gray value of non-skin pixels. The threshold is estimated in this way, because in overshadowed or too much illuminated images the probability of belonging to skin changes. Samples for this can be found in Figure 3.3 and Figure 3.4. The first figure shows an image with a rather dark tone. Therefore the SPM (Figure 3.3(b)) contains lower probabilities of skin indicated by the fact, that there are less bright spots. Especially if compared to Figure 3.2(b). The same fact can be found in Figure 3.4(b) where the original image is too much illuminated. However, both binarized maps capture the skin regions in a rather good way.

The last step is performed to reduce noise in the skin images. Figure 3.2(c) shows the binarized SPM where much noise can be found in the background. Materials like wood, tiles, wallpapers, or sheets share the same color range as skin. These materials are often contained in backgrounds of offensive images. Noise removal is done in two steps: first, the connected components in a 4-pixel neighborhood are computed. Then all but the ten biggest connected components are removed. The result can be seen in Figure 3.2(d). Most of the small points in the background are removed. The number ten was chosen to capture skin

parts that are broken, either because the people are partly dressed or because some parts are overshadowed or too much illuminated. Since skin has a high reflectance this might often be the case. This phenomena can be seen in Figure 3.3. Some areas of the body appear too dark or too bright. In the SPM of this image, the areas are black, leaving holes in the skin image. Finally five features are calculated during this process. These features include the following:

- Skin probability ratio
- Skin ratio before noise removal
- Number of connected components
- Skin ratio after connected components
- Skin ratio of the largest connected component

The skin probability ratio is calculated in the following way:

$$SPR = \frac{\sum_{x,y}^{N,M} p(x,y)}{NM} \tag{3.1}$$

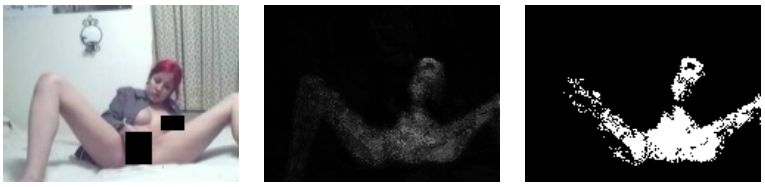
where $p(x,y)$ is the probability of being skin of the pixel x,y and the resolution of the image is $N \times M$. The skin ratio before noise removal is the sum over all skin pixels in the binarized SPM divided by the pixel count:

$$SR = \frac{\sum_{x,y}^{N,M} s(x,y)}{NM} \tag{3.2}$$

where $s(x,y)$ denotes a skin pixel at position x,y in the binarized SPM. The number of connected components are just the number of connected components found in the skin image. The skin ratio after component removal is calculated in the same way as the skin ratio only that just the ten biggest components are taken into account. The size ratio of the largest connected component is the number of pixels in the biggest component divided by the image size. Most features are normalized by the total number of pixels to get values that are independent of the image size.

(a) original image (b) SPM (c) binarized SPM

Figure 3.3: *A dark image where the probability of the skin regions is not very high, as can be seen in the skin probability map. By using an adaptive global threshold, the binarized SPM covers almost the whole skin area.*



(a) original image (b) SPM (c) binarized SPM

Figure 3.4: *A Bright image that is too much illuminated and because of that the probability of skin is low for the actual skin pixels. The binarized SPM manages to cover the whole skin region, since an adaptive threshold is utilized.*

3.1.2 Bag-of-visual-words

The second method is based on the *bag-of-visual-words* method that was generally described in section 2.2. This section describes how the steps of this method are performed and why things were done this way.

Sampling Methods of Local Image Patches

First, a codebook has to be created. Therefore the extraction and description of local image patches have to be executed. Two different methods to extract the local patches are distinguished. The first one is a regular sampling of 14×14 patches every 5 pixels, which will be referred to as the regular sampling method. The second one samples rectangular patches with different sizes at different steps. A basic stepsize b is introduced with $b = N/40$, where N denotes the height in a $N \times M$ dimensional image. For the sampling rectangular patches of size $p \times p$ are extracted every s steps, where $p = \alpha b$ and $s = \beta b$. The sampling is repeated, starting with $\alpha, \beta = 1$ until $\alpha = 4$ and $\beta = 2.5$, where α is increased in every new run by 0.5 and β is only increased every second run by 0.5. This method will be referred to as very dense sampling. Grid sampling was chosen, because it led to a better performance in classifying whole scenes [41, 28]. Unlike sampling around salient points, the regions in the background are used as well which might lead to a better discrimination between images of different categories.

Very dense sampling is used for the SURF and ColorSURF descriptors to express different scales as well, because SURF can handle different scales. For the PCA and DCT descriptors only the regular sampling is used, because PCA needs samples of the same dimension and we wanted to compare both descriptors to each other.

Table 3.1: *Overview of the descriptors for local patches*

Descriptor	Sampling	# Features	Color Information?
DCT	regular	78	yes
PCA	regular	30	yes
SURF	very dense	128	no
ColorSURF	very dense	256	yes

Different Descriptors and Construction of the Vocabulary

Four different descriptors are used to describe the local patches: DCT, PCA, SURF, and ColorSURF. A short overview of some properties of the descriptors can be found in Table 3.1. For the DCT and PCA descriptors, actually only the image patches are used. The

two transformations are only applied for dimensionality reduction. Instead of the widely used SIFT descriptor, we use SURF, since it is faster to extract and the performance is similar. Because SURF is based on the gray scale image, no color information is put into this descriptor. Since color may be an important source of information for the classification of offensive images, the idea of concatenating color histograms to the SIFT features has been applied to SURF as well. This descriptor will be referred to as ColorSURF.

- **DCT**: For the DCT descriptor, the local patches are extracted with the regular sampling method. The same transformation and description method as in [40] is used. The patches are transformed into the YUV color space in the following way:

$$YUV : \begin{cases} Y = 0.30R + 0.59G + 0.11B \\ U = -0.15R - 0.29G + 0.44B \\ V = 0.62R - 0.52G - 0.10B \end{cases} \tag{3.3}$$

  with R, G, B being the color channels of the original RGB color space. The transformation expresses the color with one *luminance* component (Y) for intensity changes and two *chrominance* components (UV) for color changes. For each channel the DCT is calculated. From the resulting coefficients the first 36 coefficients are extracted in a zigzag pattern from the luminance channel and the first 21 coefficients are obtained from each of the chrominance channels in the same way. As a whole, each image patch is described with a 78 dimensional vector.

- **PCA**: For the PCA descriptor, the local patches are extracted with the regular sampling method. The patches are not transformed into another color space. A PCA model is learned from all the patches and then used for dimensionality reduction of the patches by using the first 30 correlation coefficients (see section 2.2.2) after applying PCA on the patches.

- **SURF**: The patches that are described with SURF are sampled with the very dense sampling method, since the SURF descriptor allows to describe patches with different scales. For each patch the SURF-128 feature vector is computed. For more details see section 2.2.3 or [4].

- **ColorSURF**: Since one disadvantage of SURF is, that it does not include color information, the idea presented by van de Sande in [41] are applied to modify SURF. Color histograms in HSV color space are concatenated to the SURF-128 feature vector. The HSV values are calculated from RGB in the following way:

$$HSV : \begin{cases} H = \frac{1}{2\pi} \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-B)(G-B))}} \\ S = \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)} \\ V = \max(R,G,B) \end{cases} \tag{3.4}$$

  HSV describes colors as points in a cylinder around a central "brightness" axis. The angle is expressed by H and denotes the color's *hue*, while its *saturation* is expressed with S, which is the distance from the central axis to the point. V (the *value*) gives the color's intensity value which is displayed as the height on the central axis.
  For histogram creation, each local patch is divided into 4 equally sized blocks. For each block a color histogram for the H and S channel is calculated, using 8 bins for H and 4 bins for S. Intensity is not used, because it is already expressed in the SURF descriptor. The histograms are concatenated with the SURF features, leaving a 256 dimensional vector.

After describing each of the local patches with the particular descriptor, the vocabulary is learned by clustering the features for the patches with k-means. Each visual word is represented by the mean vector of its cluster. As a whole 2000 of these visual words are used for each codebook.

Histogram Construction

After the creation of the codebook, the patches for each image are extracted and a histogram $H(I)$ of the occurrence of each word is created. This is done by representing each patch i_p with the cluster number $c \in \{1, ..., C\}$ that corresponds to the patch in the following way:

$$c(i_p) = \arg\min_c d(i_p, v_c) \tag{3.5}$$

where d denotes the Euclidean distance, and v_c the mean vector of cluster c. The bin $H_c(I)$ is constructed in the following way for each cluster:

$$H_c(I) = \sum_{p=1}^{P_I} \delta(c, c(i_p)) \tag{3.6}$$

where

$$\delta(x, y) = \begin{cases} 1, & x = y \\ 0, & otherwise \end{cases} \tag{3.7}$$

and P_I denotes the number of local patches in the image. The final histogram is a 2000 dimensional feature vector and captures the frequency of how often each visual word occurs in an image. As classifier for the bag-of-visual-words features a SVM is used with a χ^2-distance Kernel.

3.1.3 Fusion of Results

So far two different methods are used: a simple method based on skin features and the bag-of-visual-words approach. Each of these methods is used with a classifier, getting classification scores after applying these methods. These scores indicate the probability of an image I being offensive $P_m(o|I)$, where m denotes one of the methods. The idea behind the fusion is to improve the classification result by combining the results in an appropriate way.

There are some possibilities to fuse the classification results. One would be to use the maximum of the scores. Another possibility would be to use the product or the sum of both scores. A more general method, the *weighted sum* of the classification results of the two approaches is used. An advantage of this fusion is, that the influence of a particular method can be seen by the weight for it. The final classification score is given by:

$$P(o|I) = \sum_{m=1}^{2} w_m P_m(o|I) \tag{3.8}$$

where $w_m \in [0, 1]$. Hereby the weights are learned from an additional validation set in the following way: First, the classifiers are trained on a training set. Afterwards the classification results of the previously trained classifiers are created for the samples in the training set. Then different combinations of weights are tested and the pair with the best performance on the validation data is picked. Finally, the performance is evaluated on a test set, like the methods without fusion.

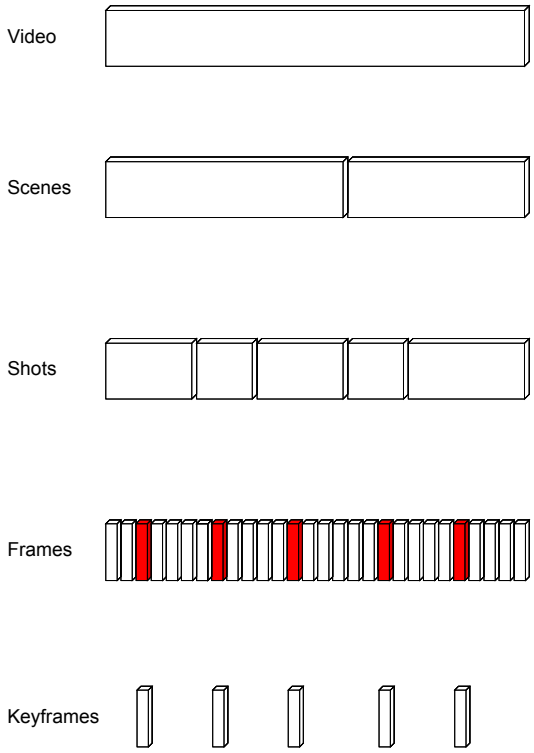


Figure 3.5: *A video can be separated into scenes, which have a plot specific meaning. Scenes are divided into shots according to camera position. Each shot consists of a number of frames. To get a meaningful image representation of a shot, a keyframe is extracted.*

3.2 Classification of Offensive Videos

The classification of offensive videos is divided into two categories: one approach is based on only the frames, the other one uses additional motion features out of the video stream. Basically, videos can be regarded as sets of images with an additional time stamp that defines a sequence for these images. Larger videos can be divided into *scenes* which can be further divided into *shots*. A scene captures footage with a similar plot specific meaning, e.g. a talk between two persons in a restaurant. A shot is the separation according to camera position, e.g. the close-up of one of the two person during the talk. Each shot consists of a set of *frames*. Because one second of video footage consists of 25 frames, the frames during one shot are usually pretty similar. Therefore, a shot can be expressed with a so called *keyframe*, that captures the displayed information. A visualization of the segmentation of videos can be found in Figure 3.5. Additional motion features can be exploited, which describe the motion between frames. The hope is that the classification performance can be improved by using additional motion features to the frames based ones.

As for the classification of images, the classification process for videos consists of the same four steps as shown in Figure 3.1. The preprocessing step in the process of the classification of videos is a scaling to a 320×240 resolution. This size was chosen for the same reasons

as for scaling images to a similar size: to increase processing speed and to get similar sized local patches for keyframe classification. The following step is the extraction of keyframes for which a simple method is proposed. Because the available offensive video material consists mainly of short sample clips which were obtained by a Web download (see section 4.1.6) and are mainly between 20 and 30 seconds long, methods for shot boundary detection are not needed for this data. Therefore, the frames are regularly extracted every two seconds, which corresponds to every 50 frames.

Further parts of this section are divided as follows: the next section covers the classification of videos based on keyframes, while the following section covers the methods which are based on additional motion features. A late fusion of the classification results is performed as well. The method is the same as was presented in the previous section on classification of offensive images.

3.2.1 Classification of Offensive Videos Based on Keyframes

Keyframes are basically images. Therefore, the methods which were presented for classification of offensive images can be applied to them in the following way: After the keyframes have been extracted for a video, they are classified by one of the presented approaches. Since these were described in section 3.1 this part is not repeated here. After the classification, a result is needed for the whole video and not just for its keyframes so the scores have to be fused to get a final result.

Ideally, all keyframes of a video should correspond to the same class, although the classification scores might differ slightly. A reasonable choice for a fusion is to use the *maximum* score off all the keyframes, since a video should be regarded as offensive, if one of its keyframes is offensive. However, due to misclassifications, this might increase the false positive rate by a large amount. So a more stable fusion is needed, which is robust to misclassifications of single keyframes.

One assumption is that an offensive video contains mostly offensive keyframes, while an inoffensive video contains mostly inoffensive keyframes. Based on this assumption, the *average* score of all keyframes is a reasonable choice for a fusion rule. Therefore, let X be a video with keyframes x_i with $0 \leq i \leq N$, and $P(o|x_i)$ the classification score of keyframe x_i being offensive. This classification result can be obtained from one of image classification methods. The final result is calculated in the following way:

$$P(o|X) = \frac{\sum_{i=0}^{N} P(o|x_i)}{N} \tag{3.9}$$

as it was presented by Ulges et al. in [39]. This fusion rule can be compared to the sum rule in classifier combination. Since the final score is an average over all frames, misclassifications of some frames are compensated. However, if only some frames of an offensive video are offensive, this method might fail. But this should actually be not the case in most of the videos.

3.2.2 Classification of Offensive Videos Based on Motion Features

So far the classification of whole videos is achieved by splitting the video into a set of keyframes, which are classified with the already presented methods. Afterwards a final result is obtained by fusing the scores of all keyframes. However, videos store more information than just the frames. One additional information which can be exploited, are the occurring motion signals. It is distinguished between two different motion descriptors that have been used before. One is based on motion histograms, and the other one tries to find periodic patterns in the motion signals.

A basic assumption is that the occurring motion signals differ for offensive and inoffensive videos. The motion histograms cover the motion patterns in local blocks. The periodicity detection tries to find periodic motion in the video stream which might correspond to scenes that show people having sex. Each of these features might not be discriminative enough. Hopefully, a late fusion of classification results of motion and keyframe features can improve the classification performance.

The motion vectors for both methods are extracted from the XViD[1] encoded MPEG video. Generally, motion is distinguished into *global* and *local* motion. Global motion denotes all the camera related motion, like camera zooming, rotation, and panning, while local motion involves the motion which corresponds to objects. A simple method to estimate the global motion is utilized, which is basically presented by Pilu in [32]. An affine velocity model is fitted using RANSAC [16]. The local motion can be obtained by subtracting the global motion from the whole motion field.

Motion Histograms

This approach follows the motion features presented by Ulges et al. in [40]. The idea is to describe the motion signal via histograms which are constructed for local regions. By constructing histograms for different regions, the descriptor can cover the occurrence of a motion pattern in a frame as well as the kind of motion which occurs.

Each frame is divided into 4×3 regular blocks to define the different regions. A motion histogram over all frames in the video is then constructed for each of these blocks. The size of the histograms is 7 bins each for motion vectors in x- and y-direction respectively. All motion vectors are clipped to $[-20, 20] \times [-20, 20]$. The final feature vector is obtained by concatenating all histograms for the blocks, leaving a 588-dimensional vector.

Periodicity Detection

The idea of using periodicity detection for offensive videos was already proposed by Rea et al. in [33]. Periodic motion signals might more frequently occur in offensive videos, particularly for scenes showing sexual activities. To detect periodic signals, the *autocorrelation function* (ACF) is used as was proposed in [33] and by Tong et al. in [38].

The ACF for a discrete signal s of length N is defined as follows:

$$ACF_s(\tau) = \frac{1}{N} \sum_{i=1}^{N} s(i) \cdot s(i + \tau) \tag{3.10}$$

It can express how similar a signal is to itself at different *lags* τ [44]. The autocorrelation can be regarded as a convolution, and therefore the *Fast Fourier Transform* (FFT) can be used to avoid the quadratic calculation. This is achieved by calculating a dot product in the frequency domain. Since the ACF of a periodic signal is also periodic with the same periodicity, the ACF can be used to estimate the periodicity of a signal. Because the ACF measures the self-similarity, the beginning of a period is indicated by a local maximum in the ACF. Therefore, periodicity in a signal can be detected by finding local maxima in its autocorrelation function and measuring the distance between them.

The following steps are performed to detect periodic patterns in motion signal. First, the extracted motion vectors are compensated of global motion as was described above. From the resulting signals, the mean motion signals of a $N \times M$ sized frame are calculated separately for the x- and y-direction over the whole video.

$$\overline{dx} = \frac{1}{NM} \sum dx(x, y) \tag{3.11}$$

[1] http://www.xvid.org

$$\overline{dy} = \frac{1}{NM} \sum dy(x, y) \tag{3.12}$$

where dx, and dy are the local motion vectors for x- and y-directions at position (x, y). The following steps are calculated for both $\overline{dx}$, and $\overline{dy}$. However, the formulas are only given for $\overline{dx}$.

First, the mean of the mean motion signal is subtracted from each value:

$$s(t) = \overline{dx}(t) - \text{mean}(\overline{dx}) \tag{3.13}$$

Afterwards the signal is normalized in a way that every value falls into $[-1, 1]$. Afterwards the autocorrelation function ACF_s for the normalized signal s is calculated. The local maxima m_i, $i = 1, ..., k$ of ACF_s are found using a sliding window over the function. Then a a signal l is calculated, which stores the differences of the maxima: $l_j = m_{j+1} - m_j$, where $j = 1, ..., k - 1$. Now the periodicity can be estimated by calculating the mean of l:

$$p = \frac{1}{k-1} \sum_{j=1}^{k-1} l_j \tag{3.14}$$

which is used as one component of the final feature vector. Additionally, the variance of l is calculated as well:

$$v = \sqrt{\frac{1}{k-1} \sum_{j=1}^{k-1} (l_j - p)^2} \tag{3.15}$$

We also use features based on [33], where the surface between the curve through the local maxima and the curve through the local minima is used. The local minima are also found by a sliding window. The surface is calculated in the following way:

$$a = \text{area}(f_{max}) - \text{area}(f_{min}) \tag{3.16}$$

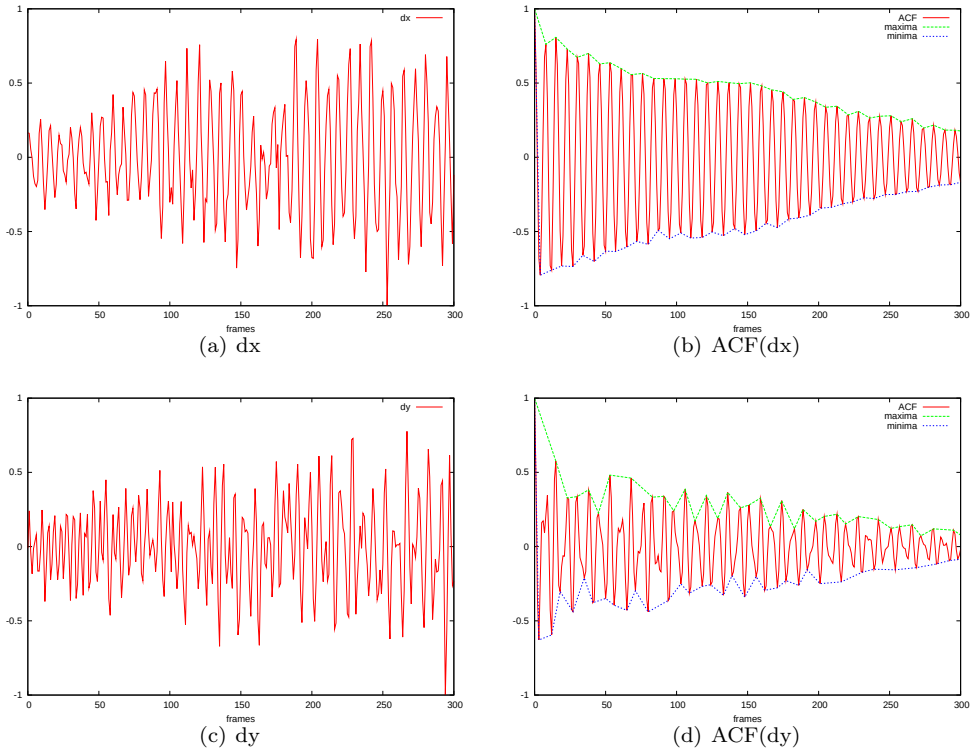
where area(f) is the area under the function f, f_{max}, and f_{min} denote the line through the local maxima and minima of the ACF of signal s. The final feature vector is constructed by calculating p, v, and a for both directions and storing them in the same vector.

A visualization of the periodicity detection is given in Figure 3.6, and Figure 3.7. Figure 3.6 shows the mean motion signals and their ACF for an offensive sample video, that shows people having sex. In Figure 3.6(a) the mean motion signal in x-direction is displayed. This signal shows a periodic pattern, which can also be seen in the according ACF (Figure 3.6(b)). In both signals the peaks have the same distance from each other. Also the area between the line through the local maxima and the line through the local minima is large which indicates also a strong periodicity. The same observation can be made for the mean motion signal of the y-direction (Figure 3.6(c)). However, for this signal, the periodicity is not as strong as for the x-direction, which results an a smaller area between the two enclosing lines (Figure 3.6(d)). Also the peaks in the ACF have a larger distance compared to the other one.

In contrast, Figure 3.7 shows the same graphs for an inoffensive YouTube video. The mean motion signals for both directions (Figure 3.7(a) and Figure 3.7(c)) show only very little periodic patterns. Both ACFs (Figure 3.7(b) and Figure 3.7(d)) show no periodicity and the local maxima have a rather large distance from each other. The area between the lines through the extrema is small.

These two figures give an impression of how the method works. For periodic signals, the signal autocorrelation function shows peaks at the same periods as the source signal. For offensive videos the distance between the peaks is expected to be small, and therefore showing a high periodicity. Also the area encapsulated by the lines through the local maxima

and minima should be large for offensive videos, whereas for inoffensive videos the opposite should hold.



(a) dx (b) ACF(dx)

(c) dy (d) ACF(dy)

Figure 3.6: *This plot shows the mean motion vectors and their ACFs for an offensive video, that shows people having sex. The short distance between the peaks of the ACF and the large area between the lines through the extrema show, that there is a strong periodic pattern in both motion signals.*

Since the videos may be long regarding the number of frames contained, we also test a slightly different method for periodicity detection. The presented features are only extracted for a sliding window over the mean motion signals. A classification score is generated for the window, and the final classification result is obtained by fusing the results of all windows. This is similar to the classification method for videos which only use the keyframes as input.

The sliding window uses a step size of one second and a window size of three seconds, which corresponds to 25 frames for step size and 75 frames for window size. For the fusion, the following methods are compared: a simple *maximum* vote, an *average* vote, and a combination of both. Let $P(o|w_i)$ denote the classification score of sliding window w_i with $i \in [1, W]$, then the maximum vote is given by:

$$P_{\max}(o|X) = \max_i P(o|w_i) \tag{3.17}$$

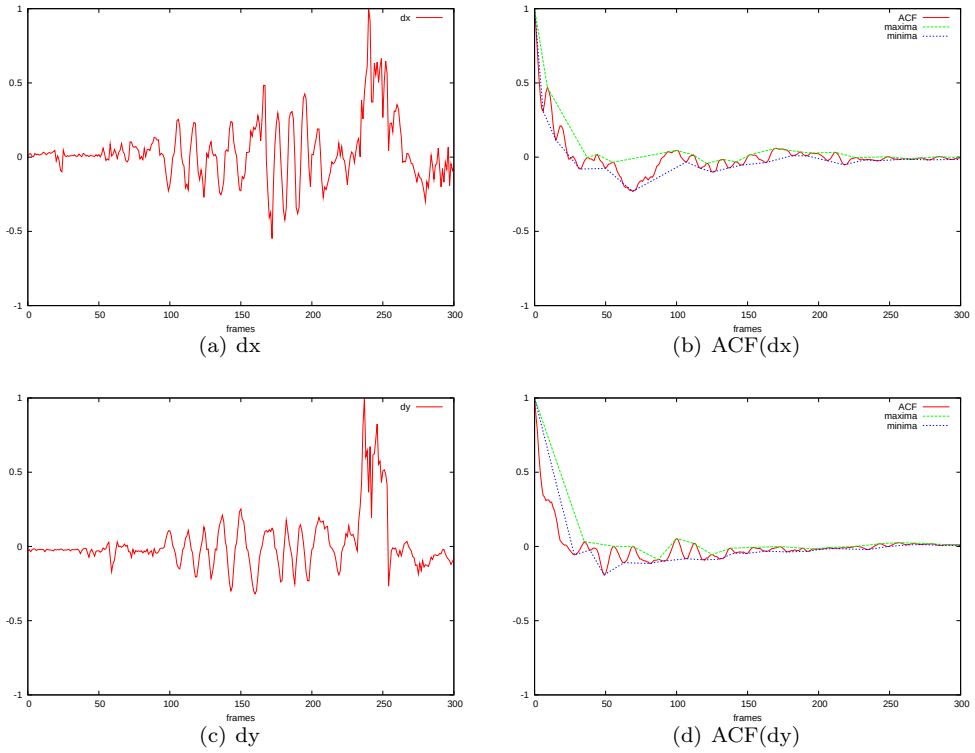
The average vote is calculated as follows:

$$P_{\text{avg}}(o|X) = \frac{1}{W} \sum_{i=1}^{W} P(o|w_i) \tag{3.18}$$

and the combination is obtained by:

$$P_{\text{avgmax}}(o|X) = P_{AVG}(o|X) + P_{MAX}(o|X) \tag{3.19}$$

In the following text, this method will be referred to as PeriodicityWin, while the periodicity detection on the whole video will be referred to as periodicity detection.



(a) dx (b) ACF(dx)

(c) dy (d) ACF(dy)

Figure 3.7: *This plot shows the mean motion signals and their ACFs for an inoffensive video. Both mean motion signals show no periodic pattern which is also indicated by the ACFs. The distance between the peaks is not regular, and the area between the lines through the extrema is low.*

Chapter 4

Experiments

This chapter contains an overview of the performed experiments as well as an overview of the datasets on which the experiments were performed.

4.1 Datasets

The task of finding the right datasets is not an easy one. Since the classifier is trained on a certain dataset, the performance depends on how well the data is picked. It is important to find a dataset which represents the real data in a good way to get reliable results for the classification. It is also important to get enough data, because a classifier needs a certain amount of training samples to perform well on test data. Another problem that occurs during the process of finding a dataset is the need of labeling. Each image has to get at least one label, declaring whether this image is offensive or not. The labeling process, however, is tedious work and it is beneficial if this step can be skipped, for example by using an existing dataset.

For the task of classifying offensive images, we need to get both offensive and inoffensive images. Offensive images can be gathered by a crawl through the Internet. There exist large amounts of websites showing adult content of many kinds. But a wide range of images is not offensive which makes it difficult to gather a representative set of these images. Therefore we decided to focus on two likely scenarios in which a filter of adult images may be of use. One scenario is an adult content filter for web browsers which is a likely application to guard children from the increasing amount of adult material in the Internet. The other scenario is the search for pornographic images on a hard drive which might be performed by police forces to detect illegal material on a suspects computer. Therefore we decided to use the following sources for our training and test data for inoffensive images: the Corel Image Database, Flickr[1], and images gathered from a crawl through the Web.

An additional problem is to get comparable results to other publications on the specific topic. The best way to get comparable results is to have some kind of a standard dataset which can be used to create the results. Regarding the classification of offensive images, however, such a dataset does not really exist, since every researcher uses his own data. A good comparison between systems is therefore difficult, because one cannot compare the results directly and one has to rely on the published results. There exists one exception, which has been used in two publications ([25] and [12]) and therefore, the presented systems are tested on this data as well.

A short overview of the datasets used for image classification can be found in Table 4.1 while a more thorough description will be presented in the following sections. For video

[1] http://www.flickr.com

classification, only offensive Web videos are taken into account. The opponent class is given by YouTube videos which cover a wide range of inoffensive footage. An overview of the two video datasets is given in Table 4.2.

Table 4.1: *Overview of the datasets for offensive image classification*

Dataset	size (number of pictures)
Standard dataset	8,510
Offensive images from the Web	4,248
Corel Image DB	4,198
Flickr images	2,000
Inoffensive images from the Web	2,752

Table 4.2: *Overview of the datasets for offensive video classification*

Dataset	number of videos	number of keyframes
Offensive videos from the Web	932	11,612
YouTube videos	2,663	25,660

4.1.1 Standard Dataset for Detection of Offensive Images

This dataset was introduced by Kim in [25] and later used by Deselaers in [12]. The purpose of this dataset was to train a classifier that performs well with different filtering rules. Varying filtering rules are needed, because different cultures have different views on what may be offensive and what may not. This dataset divides the images into the following five categories:

- inoffensive images: images that mostly contain landscape and nature scenes (D)
- images with lightly dressed people, for example people wearing swimsuits or underwear (AA)
- images with topless people (AB)
- images with naked people (CA)
- pornographic images, images that show people having sex (CB)

Each of these classes contains 1,702 pictures, giving a total of 8,510 pictures. Some sample images of each of these categories can be found in Figure 4.1. It was decided to use this dataset for various reasons. First, it contains images that are labeled for five different categories. The labeling of images is tedious work and having access to already labeled data can save a large amount of time. Second, this dataset has been used in two other publications, which allows to get a better comparability to these approaches. To test the performance of the classifier, we perform four experiments with images from this dataset. In the first experiment only the pornographic images from class CB are used as offensive images. In the second experiment the images containing naked people (class CA) are added to the offensive images, and so on. The final experiment regards images from all categories AA, AB, CA, CB as offensive. Each set of offensive images is tested against the inoffensive

images of class D. These experiments are denoted as "CB-D", "CA-CB-D", "AB-CA-CB-D", and "ALL-D" in the following sections respectively.

A disadvantage of this dataset, however, may be that it is not representing the real world data. The offensive images have mostly rather high quality compared to images downloaded from the Web. The class of inoffensive images mostly contains images with nature shots where no human being appears. Regarding the application as a Web content filter this may not be the most appropriate dataset to choose.

(a) AA

(b) AB

(c) CA

(d) CB

(e) D

Figure 4.1: *Images from the standard dataset: a) a lightly dressed person, b) a topless person, c) naked people, d) people having sex, and e) an inoffensive image*

4.1.2 Offensive Images from the Web

One possible use of a system that can detect offensive pictures is to block these kind of pictures in a Web browser. Because the performance of a trained system is best when applied to the same domain as trained on, 4,248 adult content images were gather by a random crawl over pornographic websites and manually labeled afterwards. The websites were freely accessible and not password protected, therefore these images can be viewed by everyone. WGET was used to recursively download all images on a website. Images were regarded as offensive if:

- they display partly naked people, for example a naked female breast
- they display naked people
- they display people having sex

The pictures have a wide range of different resolutions from very high to very low. The image quality has also a very wide range since some are professional shots while others are made by amateurs. Some examples are shown in Figure 4.2. This dataset is used for offensive images that are tested against the datasets with inoffensive images, that are presented in the following subsections. These experiments are denoted "XXX-Corel", if images from the Corel Image DB form the opponent class, "XXX-Flickr", if the inoffensive images are taken from Flickr, and "XXX-Web Images", if the inoffensive Web images are for the opponent class.

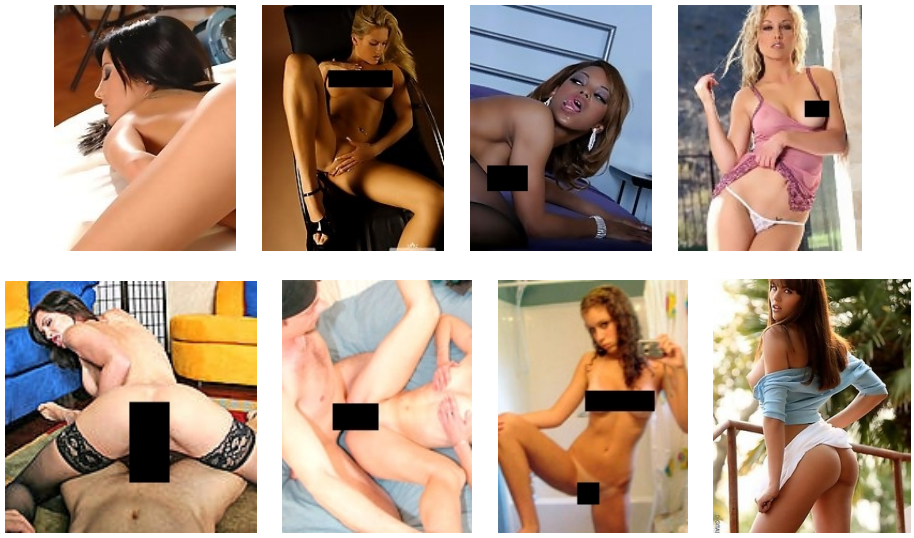


Figure 4.2: *Samples of offensive images which were downloaded from the Web. They show a wide variety of (partly) naked people and people having sex in various scenes.*

4.1.3 Corel Image Database

To get a dataset with images that are not offensive, we used 4,198 randomly selected images from the Corel Image database. These images display a large amount of inoffensive scenes including the following:

- various animals like dogs, horses, penguins, tigers
- landscapes and cities of countries and continents like Africa, Alaska, Australia, Holland, Hawaii, Italy
- sceneries like beach, desert, forest, rock formations, ruins, sunset, tropical islands
- people from various countries
- artificial images
- food like barbecue, and fruits

The images in this set are not representative of common web images, because they share the same resolution. Also they have a high quality which is not guaranteed in common web images. This dataset is used nevertheless, because it contains a huge amount of different motifs that are not offensive. Also many materials can be found that are similar to skin color like rocks, sunsets, and the fur of some animals. However, these images should be easier to separate form offensive image than inoffensive images from the Web.

4.1.4 Flickr Dataset

Another set of inoffensive images was gathered by downloading from Flickr[2]. Flickr is an online portal that allows people to upload and share pictures they have taken themselves. The images can be annotated with tags, allowing other people to find pictures easily. This makes it possible to get a large amount of images that are not offensive. Another advantage

[2] http://www.flickr.com

Figure 4.3: *Sample images from the Corel Image DB which show different landscape shots, people, and animals*

is that the images are made by normal users and not from professional photographers like the Corel Image Database. These images give a better representation to images that can be found on websites (for example web blogs or other personal homepages). These kinds of images can also be found in private image collections, since they display scenes from holidays or social events like concerts and weddings. They are the same pictures people put onto Flickr. Pictures with the following themes and tags were downloaded from Flickr:

- animals (bird, cat, dog, insects, etc.)
- events (concert, party,wedding)
- landscape (skyline, skyscraper)
- nature (beach, forest, hills, lake, mountain, river, etc.)
- people (face, person, portrait)
- sports (American football, baseball, basketball, bowling, hockey, soccer, etc.)

There is a concentration on images that contain people (events, people, sports) because these may be the problematic cases when used with a skin detector. However, this choice should make these images harder to separate from offensive image than images form the Corel Image DB. The other images were chosen because many of these images exist on local harddrives and in the Web. As a whole 2,000 pictures were gathered from Flickr. Some examples are given in Figure 4.4.

4.1.5 Inoffensive Images from the Web

The last dataset with inoffensive images should represent common inoffensive Web images. However, gathering images from the Internet that are not offensive but are displayed on frequently visited websites is not easy. One problem is to define what usual pictures in the Internet are. Alexa's top 500 visited websites list[3] is used to get the sites that are visited most frequently. Since the list also contains websites with offensive content, we manually deleted these sites from the list. A similar download method like for the download of offensive images was executed. The so gathered images had to be checked if they contained any offensive ones which were removed. Images that were smaller than 40×40 pixels were

[3] http://www.alexa.com/site/ds/top_500

Figure 4.4: *Sample images downloaded from Flickr which show people in different scenes that are not offensive.*

also removed because they were mostly small icons or thumbnails. As a whole 2,752 pictures were gathered in this way. Examples are shown in Figure 4.5.

From all datasets with inoffensive images, this should be the most representative regarding the application of Web content filtering. However, this should also be the most difficult one. Many of the pictures in this dataset have only a small resolution and low quality. Also artificial images are included and many pictures that contain people.

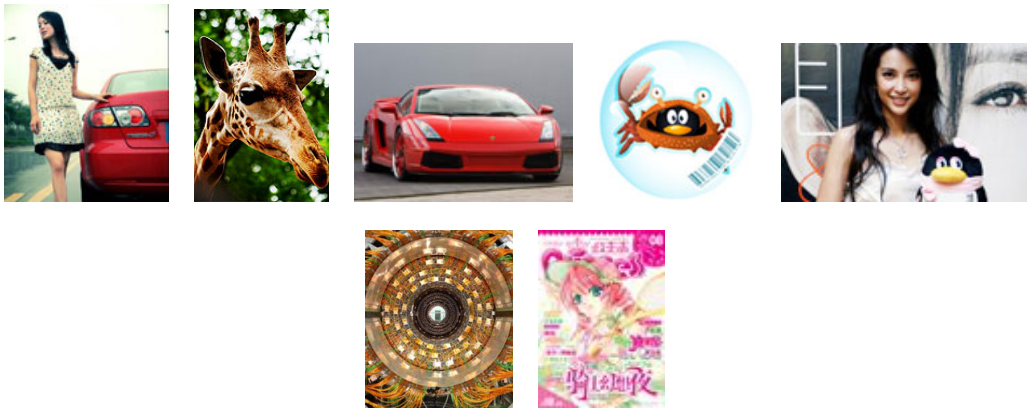


Figure 4.5: *Samples of inoffensive images which were downloaded from the web*

4.1.6 Offensive Videos from the Web

Offensive video material was also gathered by a random crawl over the web. The same websites were used as for the download of offensive images. 932 adult content videos were acquired this way. Most of these videos are small snippets of larger films and therefore just contain one scene or even just one shot. The run-time of these videos ranges between ten and thirty seconds. This is a big advantage, since methods to separate the videos into shots do not have to be applied. Also, these videos should be representative for a large amount of

pornographic films since a wide range of material is covered from amateur videos to videos with a more professional production. Also the videos contain different scenes from sexual actions to just naked people. All videos were scaled to a 320 × 240 resolution for reasons presented in section 3.2. Since some of the used methods are performed on images, the first step is keyframe extraction. A keyframe is a frame out of the video which displays a meaningful shot out of a scene. Because only short snippets are available a regular extraction of the keyframes is applied at every 50 frames. This corresponds to an extraction every two seconds. As a whole 11,612 keyframes were extracted for the offensive video dataset. Some sample keyframes are shown in Figure 4.6.

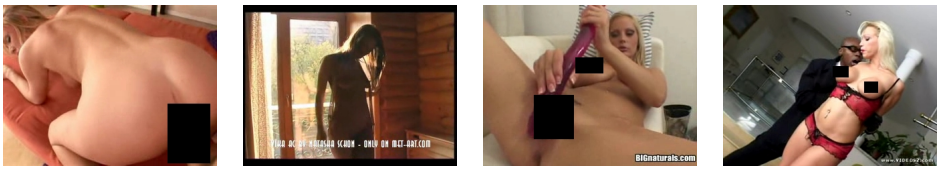


Figure 4.6: *Some keyframes from offensive videos which were downloaded from the Web*

4.1.7 YouTube Videos

To get a large variety of non-offensive videos, the online video portal YouTube was used as source. On YouTube one can be (almost) sure to get lots inoffensive videos which are already labeled. Just like the images from Flickr, YouTube videos are tagged. Videos with the following tags were chosen to be downloaded:

- animals (cats, dogs)
- events (concert, dancing, demonstration, interview, singing, talkshow)
- nature (beach, desert, flower, mountain, hiking)
- people (hand, two people)
- sports (basketball, golf, sailing, soccer)

Again tags were chosen, that either relate to social events, and sports, and therefore contain people, or videos that show nature shots. One serious disadvantage of community based labeling is the problem, that many videos are insufficiently or even wrongly labeled. Therefore a video with the tag “Dog” might not even contain a dog at all. However, it is possible to get a large amount of videos with different content. Another disadvantage is the amount of videos in poor quality. Many of the videos are home made and therefore have poor quality. Other videos are taken from TV shows which usually have good quality. This might complicate feature extraction since many videos are therefore noisy. The videos are also much longer than the offensive video snippets from the offensive websites. Therefore, snippets ranging from 10 to 20 seconds were randomly sampled out of the YouTube videos. Like the offensive videos, all videos are scaled to a fixed resolution of 320 × 240 pixels. Keyframes were extracted in the same way as for the offensive videos. As a whole 2,663 videos, and 25,660 keyframes were gathered. Some samples can be found in figure Figure 4.7.

Figure 4.7: *Some keyframes from YouTube videos*

4.2 Experiments for Classification of Offensive Images

This section deals with the experiments that were executed to see how well the detection of offensive images works. First, the measure of how the classification results are compared is presented. Then, the results for the skin ratio features are presented which are regarded as a baseline system. The results of a decision tree and a SVM as classifier with these features were compared. The following experiments deal with the performance of the bag-of-visual-words approach, where the different descriptors for the local patches were compared. Also an experiment is shown, where the number of training samples are evaluated. Finally, the results are fused to see if the performance can be improved.

Several experiments were performed to measure the performance of the approaches that were presented in the previous section. Because one wants to compare the results, an appropriate measure of a classifiers performance is required. The classification rate (the number of correctly classified samples divided by the total number of samples) is not an appropriate measure because it does not relate the number of true and false positives. Table Table 4.3 shows a general confusion matrix for the classification of offensive images to show the relation of these numbers. The true positives (TP) are the number of correctly classified offensive images and the true negatives (TN) are the number of correctly classified inoffensive images. The false negatives (FN) are the number of inoffensive images which are classified as offensive ones and the false positives (FP) are the number of offensive images which are classified as inoffensive. Since one wants a measure that relates both true and false negatives, the equal error rate is used. The equal error rate is the point where the *false positive rate (fp)* equals the *false negative rate (fn)*:

$$fp = \frac{FP}{FP + TN} \tag{4.1}$$

$$fn = \frac{FN}{FN + TP} \tag{4.2}$$

Therefore, the false positive rate gives the fraction of incorrectly classified inoffensive images and the total number of inoffensive images. The false negative rate is the number of offensive images that are classified as inoffensive divided by the whole number of offensive image. The equal error rate is the point where both of these rates balance out. So if two classifiers are compared, the one with the smaller equal error rate is preferred.

Table 4.3: *Confusion Matrix for classification of offensive images*

true class	*predicted class*	
	Offensive	Inoffensive
Offensive	TP	FN
Inoffensive	FP	TN

Each of the presented approaches is tested on seven different stets which are built on the mentioned datasets. The first four experiments are performed on the standard dataset to see how well the approach works compared to other existing approaches. The following experiments are performed on the downloaded offensive image material with each of the inoffensive image datasets as inoffensive material. These experiment should tell how well the classification process can work in the real world.

4.2.1 Skin Segmentation Approach

Skin features with decision tree classifier

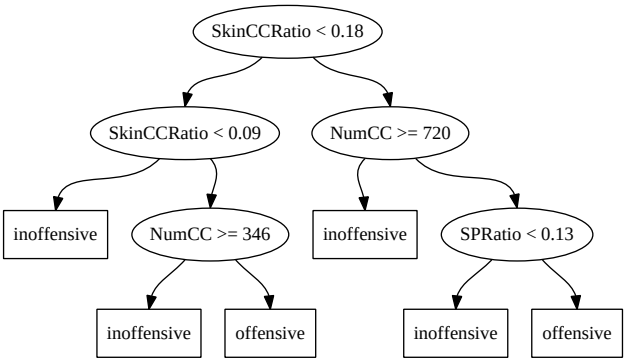
The first experiments were performed with the skin features that were presented in section 3.1.1. For each of the sets a decision tree is built on data from the training set and evaluated on a test set. The training set contained 1,000 images from each class. Due to the different sizes of the datasets, the test set contained 1,000 images per class for the Corel, Flickr and Web images, and 500 inoffensive images of the standard dataset. The results can be found in Table 4.4. The measure is the equal error rate which was presented in the previous section.

Table 4.4: *Equal error rates for experiments on different datasets using skin features and decision tree classifier*

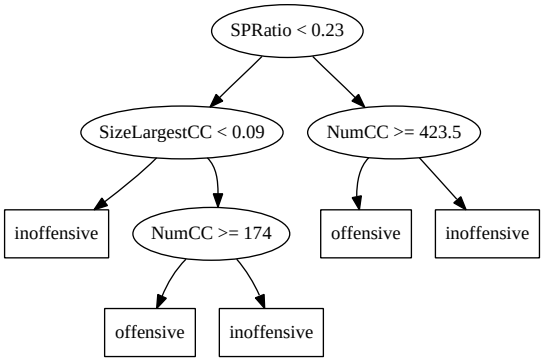
Dataset	EER
CB-D	0.0501
CA-CB-D	0.0720
AB-CA-CB-D	0.0641
ALL-D	0.0650
XXX-Corel	0.0890
XXX-Flickr	0.1085
XXX-Web Images	0.1450

The results show the best performance on the standard dataset with only the pornographic images regarded as offensive and the nature shots as inoffensive. The least performance was achieved by classifying offensive images from the Web against inoffensive images from the Web. These results are not surprising, since the first set was expected to be the easiest separation task while the last one was expected to be the hardest one. The Corel images are a little bit better to distinguish from offensive images than the images from Flickr. The performance of experiments on the standard dataset is better than on the other datasets. This further supports the assessment that the standard dataset's images are not much representable of real world data.

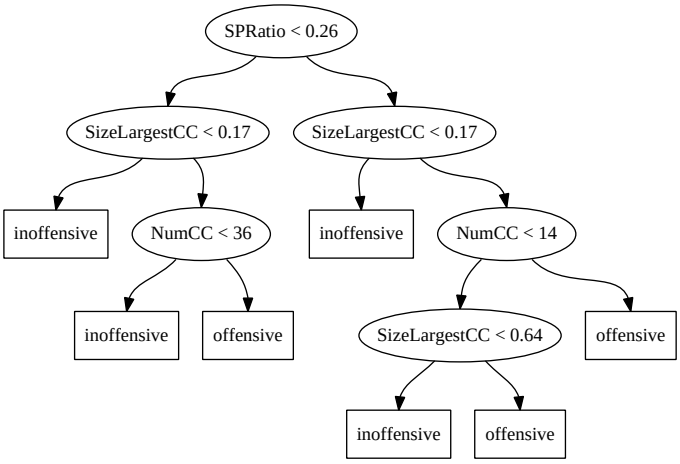
An advantage of a decision tree classifier is that separation rules can be easily created and that these rules can be viewed and interpreted. Therefore, some of the trained trees are shown in Figure 4.8. The features are as follows: `NumCC` is the number of connected components, `SizeLargestCC` is the size ratio of the largest connected component, `SkinCCRatio` is the ratio of skin pixels after removing all but the ten biggest connected components, and `SPRatio` is the skin probability ratio. All of the trees are pruned further than the ones that are used for the classification to be able to display them. The decision rule is shown inside the node. If the test is true for the given feature value, the left descendant node shows the next rule, the right descendant node otherwise. The terminal nodes are displayed with a rectangle and show the assigned class label.



(a) Trained decision tree on CB-D



(b) Trained decision tree on XXX-Flickr



(c) Trained decision tree on XXX-Web Images

Figure 4.8: *Some trained decision trees with skin features on different datasets*

The tree trained on the standard dataset Figure 4.8(a) uses the number of skin pixels after connected component analysis in the first rule. If this value exceeds a threshold, the number of connected components is analyzed. If the number of connected components is high, the image is regarded as inoffensive. This may be reasonable, since in an image with a high number of skin components, it may be more likely that these originate from some kind of background and not from human skin. The final feature in this branch is the skin probability over the whole image. If this probability is high, the image is classified as offensive, otherwise not. If the initial check for the amount of skin after the component removal is below the threshold, the image is more likely to be inoffensive. Only if the number of connected components is low, the image may be classified as offensive. This rule should capture the whole skin areas which should consist of few connected components.

Figure 4.8(b) shows a trained decision tree on offensive Web images and Flickr images. Here the most important feature is the skin probability over the whole image. This can be explained with the fact that the inoffensive images from Flickr also contain amounts of skin. So the probability of skin over the whole image is more important than just the size of the largest skin component. A portrait image, for example, has a huge skin component, but cannot be regarded as offensive. Another difference to the first tree, is that a higher number of connected components leads to a classification as an offensive image. A reasonable explanation for this is that also partly dressed people are shown in the offensive images. This leads to a higher number of connected skin components. Large parts of skin might also be separated by shadows or too much illuminated which may occur more frequently in these images, since many amateur shots are included. Even another explanation is that the Web images show more skin color like background than the pictures from the standard dataset, especially in offensive images. The remaining rules are also reasonable. If the largest skin component is small, the image is more likely to be inoffensive.

The last tree (Figure 4.8(c)) is trained on offensive and inoffensive Web images. Here the skin probability is the most prominent feature again. Just like in the second tree, another important feature is the size of the largest skin component. If this is high, the image is more likely to be offensive. Interesting, however, is the much lower threshold by rules with the number of connected components. This can be explained with the fact, that the inoffensive Web images are usually of smaller size than the inoffensive images from other sources. Therefore, the total number of components is also lower. The rules themselves do not differ much from the previous tree.

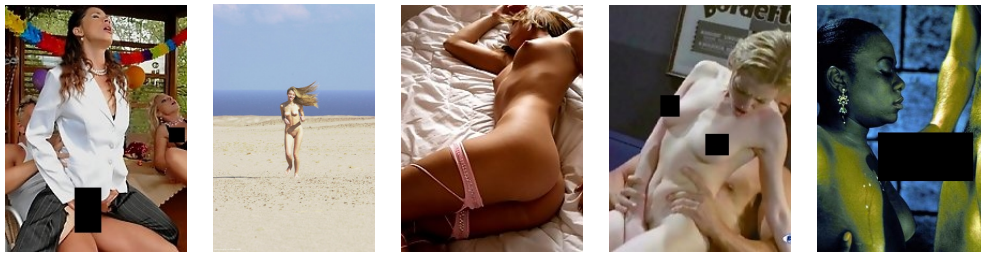
In summary, the tree trained on the standard dataset uses different rules than the ones trained on Web images and Flickr images. It is interesting but also reasonable, that both trees which use the Web images as offensive class have pretty similar rules and differ only in the threshold values. Another interesting fact is that the skin ratio (the number of pixels denoted as skin before connected component removal) is not taken into account for the first rules at all. Actually this is positive, since this feature also contains a lot of noise from the background and the removal of components is performed to reduce this noise. The assurance that this feature is not used in the first rules tells that it is not very important. However, it might be used for further rules, which are not displayed in the pruned trees.

To conclude the experiments from the decision tree classification, it is important to see which pictures are misclassified. Samples of wrongly classified offensive images can be found in Figure 4.9. These samples show some of the typical reasons why the classification fails. One reason is that although sexual content is shown, there is not much skin presented in the image (Figure 4.9(a)). The same problem appears if a naked person is present, but displayed only small (Figure 4.9(b)). In these cases the skin detection itself works, but the amount of skin is too small, causing the classifier to fail. Another problem appears when the skin detection itself fails. Reasons for this may be due to overshadowing of skin regions (Figure 4.9(c)), too much illumination which lets the skin appear as being white

(Figure 4.9(d)), or due to illumination sources which alter the skin color into other colors, for example yellow (Figure 4.9(e)). Although the decision trees have rules that should cope with some of these problems, these rules fail in extreme cases.

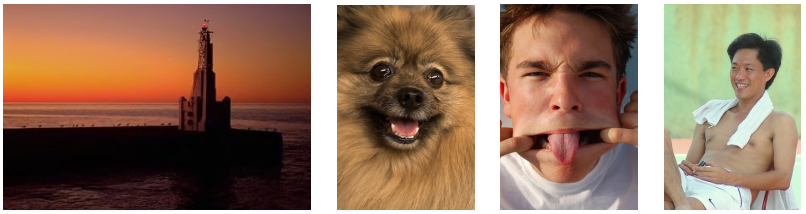
Typical misclassified inoffensive images are shown in Figure 4.10. The reason why these images are regarded as offensive is because large areas in these images are detected as skin. Common examples are: sunsets (Figure 4.10(a)), fur of animals (Figure 4.10(b)), rocks, wood, some metals, etc. The second reason for false positives is the occurrence of large skin regions, although the image is not offensive. These cases include portrait images (Figure 4.10(c)) and partly dressed people (Figure 4.10(d)).

Concluding the results achieved with skin features used for classification with a decision tree led to results that are similar to the results of existing approaches like the approach of Jones [22], and the approach of Rowley [35]. The presented method gets similar classification results as well as similar problems for the misclassifications. The results show, that a baseline system was built with roughly the same performance as other existing techniques that are also based on skin detection.



(a) (b) (c) (d) (e)

Figure 4.9: *Examples of typical offensive images that are misclassified by the skin ratio features: a) people being mostly dressed, b) the displayed person is too small compared to image size, c) the image is overshadowed, d) the image is too much illuminated, and e) the image is illuminated by a yellowish tone*



(a) (b) (c) (d)

Figure 4.10: *Examples of typical offensive images that are misclassified by the skin ratio features: a) a sunset, b) animal fur, c) a portrait image, and d) an image with much skin although it is not offensive*

Skin Features with SVM Classifier

In additional experiments we compare the performance of a SVM classifier to the performance of a decision tree classifier in use with the skin features. Because the underlying methods differ, the features and the experiments have to be adapted for the SVM. First, the number of connected components is normalized for all features so that it falls into the range between 0 and 1 like the other features. This step is needed because the kernel function

represents a distance measure. If one value in the feature vector dominates the others, the distance measure will not work correctly. The second adaption is needed because the SVM is a probabilistic classifier and not deterministic like the decision tree. If a decision tree is trained and tested several times on the same training and test sets, the classification results are always the same. If the classification is performed using a SVM the results, however, may differ. To get reliable results, we perform the experiments in the following way. For each experiment, the dataset is randomly split into a training set which contains 1,000 samples and a test set which contains 500 samples. The experiments are repeated ten times. Table 4.5 shows the mean equal error rate and the variance of the equal error rate over ten experiments on each dataset. The SVM performs better on the standard dataset while the decision tree performs better on the downloaded images. The performance of the classifiers on downloaded offensive and Corel and Flickr images are roughly the same. This can be explained with a different distribution of the features in the feature space of the different sets. This may lead to a better separation with a certain classifier. Regarding the results from these experiments, the SVM performs better on data that was categorized to be easier to separate while the decision tree performs better on real world data.

Concluding the classification with features which are based on skin detection in an image, we managed to implement a system that classifies offensive images. The previous results were not improved, which was not the goal. This system should be regarded as a baseline system. Recent publications by Deselaers [12] showed that a bag-of-visual-word approach leads to better results than just using skin color information.

Table 4.5: *Performance of SVM and decision tree on different datasets measured with mean and variance of the equal error rate over ten runs. The SVM performs better on the standard dataset, while the decision tree shows a better performance on the experiments with offensive Web images.*

n = 10	SVM		DT	
Dataset	μ-EER	σ-EER	μ-EER	σ-EER
CB-D	**0.0266**	0.0003	0.0501	0.0026
CA-CB-D	**0.0452**	0.0007	0.0795	0.0004
AB-CA-CB-D	**0.0512**	0.0003	0.0580	0.0004
ALL-D	**0.0467**	0.0005	0.0532	0.0002
XXX-Corel	**0.0568**	0.0002	**0.0568**	0.0008
XXX-Flickr	0.0908	0.0006	**0.0874**	0.0012
XXX-Web Images	0.1790	0.0019	**0.1297**	0.0019

4.2.2 Bag-of-visual-words Approach

The experiments using the bag-of-visual-words approach aim for two things: First, to see how much the performance can be improved to the skin detection based features and second, which descriptor of the local features achieves the best performance. As classifier a SVM is used, therefore setup is similar to the one used for classification with SVM and skin features. Each experiment is performed ten times on the sets with randomly selected training sets with 1,000 images and test sets containing 500 images. The codebooks for the different descriptors are learned from all available images. Therefore, the codebooks are not specialized on the different datasets. The mean and variance of the equal error rate are used again to measure the performance.

The results on the standard dataset can be found in Table 4.6. For all descriptors the CB-D experiments achieved the best results, while ALL-D was the hardest to separate.

Table 4.6: *Classification performance of different local feature descriptors on the standard dataset, measured with mean and variance of the equal error rate over ten runs. The DCT descriptor shows the best performance on all of the four experiments.*

n = 10	DCT		PCA		SURF		CSURF	
Dataset	μ	σ	μ	σ	μ	σ	μ	σ
CB-D	**0.0118**	0.0001	0.0146	0.0004	0.0358	0.0003	0.0196	0.0003
CA-CB-D	**0.0340**	0.0005	0.0363	0.0002	0.0618	0.0005	0.0396	0.0002
AB-CA-CB-D	**0.0362**	0.0004	0.0378	0.0004	0.0567	0.0009	0.0436	0.0003
ALL-D	**0.0448**	0.0015	0.0439	0.0003	0.0612	0.0003	0.0510	0.0003

Table 4.7: *Classification performance of different local feature descriptors on downloaded offensive images, measures with mean and variance of the equal error rate over ten runs. Again, the DCT descriptor shows the best performance on all experiments.*

n = 10	DCT		SURF		CSURF	
Datasets	μ	σ	μ	σ	μ	σ
XXX-Corel	**0.0274**	0.0005	0.0649	0.0007	0.0514	0.0001
XXX-Flickr	**0.0608**	0.0010	0.0822	0.0008	0.0685	0.0005
XXX-Web images	**0.0635**	0.0007	0.1008	0.0008	0.1110	0.0006

Overall the best performance was achieved by the DCT descriptor and the PCA descriptor was just marginally worse. ColorSURF was better than SURF which is reasonable since SURF does not include any color information. As a whole the bag-of-visual-words approach leads to better results than the skin detection approach. Of all descriptors only SURF performed worse than the skin features. This leads to the conclusion that color is an important information for classifying offensive images and should not be omitted.

The results for the classification of the downloaded offensive images (Table 4.7) are similar. The PCA descriptor is missing, since its performance on the standard dataset was almost the same as the DCT descriptor. The ColorSURF descriptor performs better than the SURF descriptor, except for the experiment with the inoffensive Web images. A possible explanation is the occurrence of gray scale or artificial images in this dataset. Also the low image quality might play an important role for the worse performance of the ColorSURF descriptor on this data.

Concluding these experiments, the bag-of-visual-words approach shows a clear improvement over the skin detection approaches. From all descriptors, the DCT descriptor performs best.

Number of training samples

One problem that frequently arises during the training of a classifier is the question of how many training samples are needed. To measure the influence of the number of training samples, the following experiment was conducted. An increasing number of training samples is used to train a SVM with the DCT features. The DCT descriptor is used, because it was performing best. Images for the offensive class are taken from the Web images while the inoffensive images are from the Corel set. The choice for the inoffensive pictures was made, because this set contains more images than the other sets. Sample numbers are 5, 10, 15,

25, 50, 100, 250, 500, 1000, 2000 per class. Each training per sample count is repeated five times with a newly, randomly generated set. Testing is performed on a set containing 500 images from each class. A plot with the equal error rates and the mean equal error rate can be found in Figure 4.11. It shows that the equal error rate decreases while the number of training samples increases. There is a significant increase in the classification performance from an error rate of 30% to 5% between 10 and 200 samples. The results can be improved to an equal error rate of 2.5% by using 20 times more training samples. Since we used 2000 samples for each training process the results should be accurate. However, adding more samples might give a slight improvement.

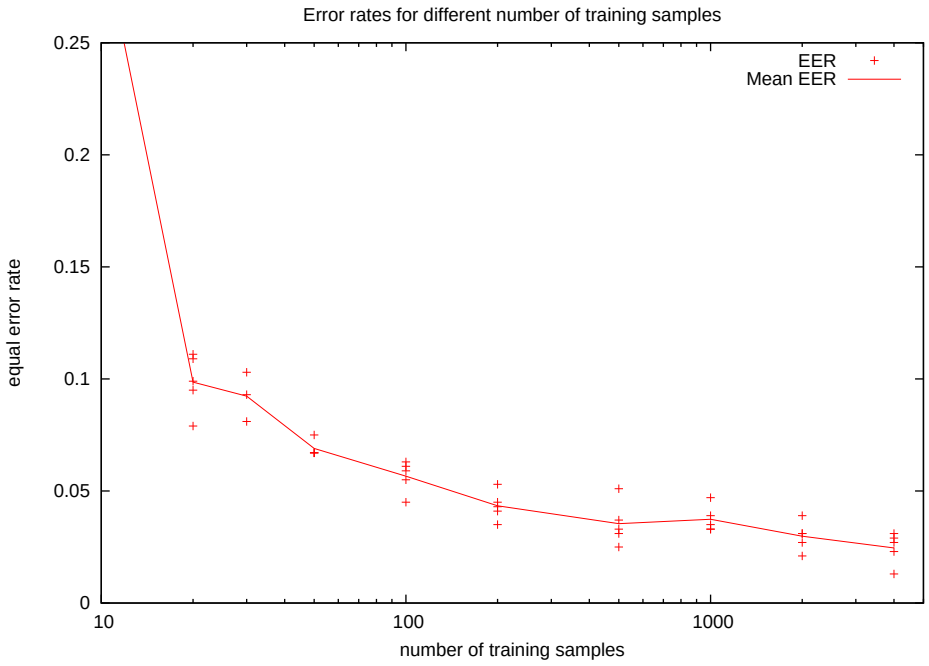


Figure 4.11: *The plot shows equal error rates for different numbers of training samples for the bag-of-visual-words approach with the DCT descriptor.*

4.2.3 Late Fusion of Results

In another experiment a fusion of the skin detection approach and the bag-of-visual-word approach with the DCT descriptor is investigated. For creating scores with both methods, a SVM classifier is used. The method for the late fusion has already been described in section 3.1.3. Because we learn weights for each feature, an additional validation set is needed to estimate the final performance. The dataset is split into a training set which contains 1,000 samples, a validation set with 500 samples, and a test set with 500 samples. Each experiment is performed ten times with newly randomly created training, validation, and test sets. The measure is again mean and variance of the equal error rate. The results are given in Table 4.8. It can be seen that the fusion achieves an improvement in all experiments over the previously existing results. The mean weights for each experiment are shown in table Table 4.9. For most of the datasets, the weight is bigger for the DCT descriptor than for the skin features, indicating that its influence on the final result is higher. This is reasonable, since the performance of the DCT features by themselves was better than the skin features alone. However, the skin features are not neglected and contribute to the final classification result.

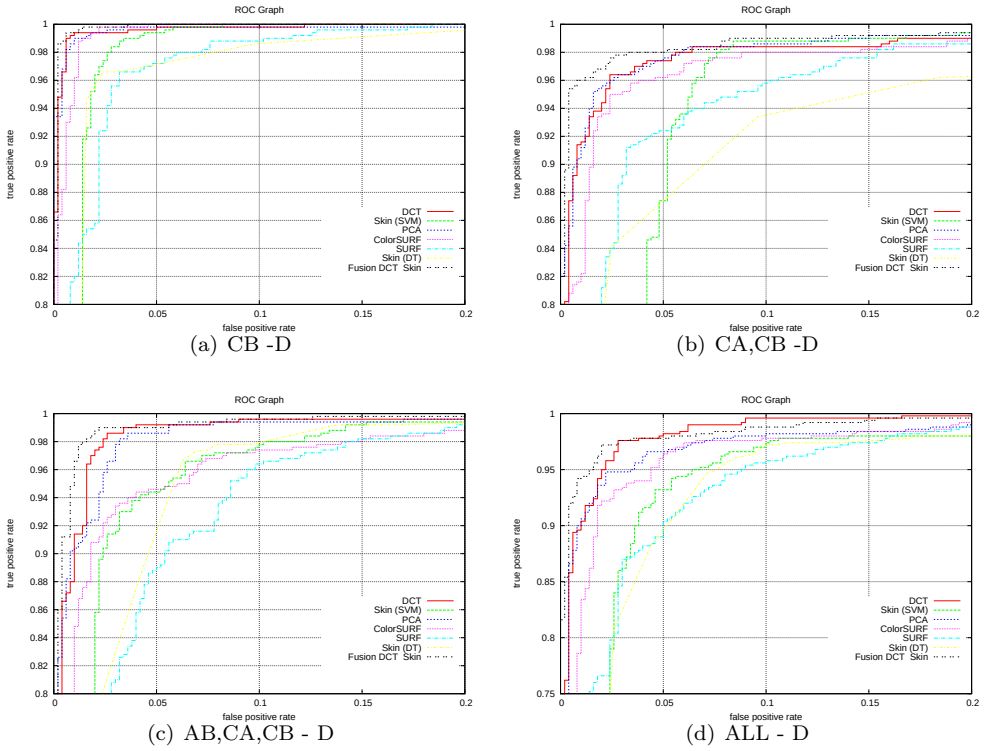
Table 4.8: *Classification performance of fusing DCT and skin features, measured by mean and variance of the equal error rate over ten experiments. On each dataset, the fusion performed better than the classification with single features.*

n = 10	DCT		Skin		Fusion	
Dataset	μ-EER	σ-EER	μ-EER	σ-EER	μ-EER	σ-EER
CB-D	0.0106	0.0000	0.0266	0.0003	**0.0088**	0.0000
CA-CB-D	0.0380	0.0002	0.0452	0.0007	**0.0288**	0.0003
AB-CA-CB-D	0.0412	0.0009	0.0512	0.0003	**0.0324**	0.0007
ALL-D	0.0468	0.0012	0.0467	0.0005	**0.0312**	0.0003
XXX-Corel	0.0274	0.0005	0.0568	0.0002	**0.0238**	0.0003
XXX-Flickr	0.0608	0.0010	0.0908	0.0006	**0.0574**	0.0008
XXX-Web Images	0.0635	0.0007	0.1790	0.0019	**0.0620**	0.0005

Table 4.9: *The mean of the learned weights of the late fusion of DCT and skin features. The weight for the DCT descriptor is larger than the weight for the skin feature in most of the experiments.*

	Fusion weights	
Dataset	w_1	w_2
CB-D	0.65	0.35
CA-CB-D	0.55	0.45
AB-CA-CB-D	0.585	0.415
ALL-D	0.49	0.51
XXX-Corel	0.515	0.485
XXX-Flickr	0.695	0.305
XXX-Web Images	0.72	0.28

To conclude the results of the classification of offensive images, *Receiver Operating Characteristics* (ROC) curves are presented. ROC curves plot the *false positive rate* against the *true positive rate* [15]. They are used to compare different classifiers. Since the perfect classifier has only true positives and no false positives, it is represented by a line between (0,1) and (1,1). The diagonal line between (0,0) and (1,1) can be seen as a classifier that randomly guesses so it generates the same amount of true and false positives. A classifier performs better than another, if its curve is closer to the line of the perfect classifier. ROC curves can be generated out of the classification results in the following way. Both the decision tree and the SVM create a score for each instance of belonging to offensive images. A threshold is then utilized to assign a label. For example if the score is higher than 0.5 the image is offensive. With different values for this threshold, different points in the ROC space can be created.



(a) CB -D (b) CA,CB -D

(c) AB,CA,CB - D (d) ALL - D

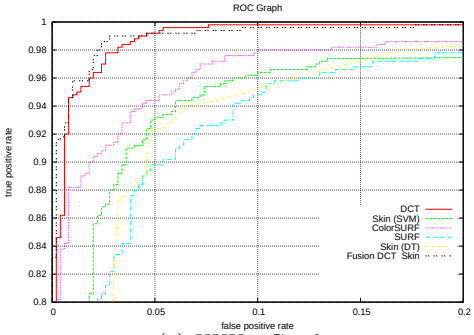
Figure 4.12: *ROC curves for different descriptors on the standard dataset*

Figure 4.12 shows the ROC graphs of all experiments performed on the standard dataset for each descriptor. The graphs basically show the same results as the tables before. From the single descriptors, the DCT descriptor shows the best performance. The skin features with both classifiers and the SURF features show the worst performance. The fusion of DCT and skin feature gives an improvement to all classification results. The same can be seen for the experiments on the downloaded offensive images in Figure 4.13.

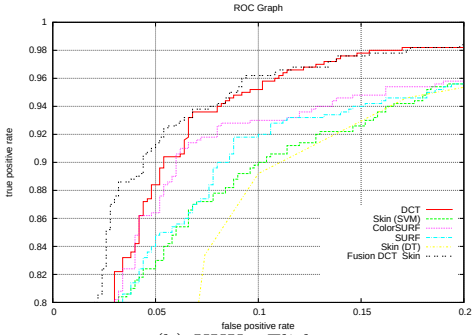
In summary the experiments showed, that both the skin detection and the bag-of-visual-words approach can cope with previously existing approaches. The performance can further be improved if a fusion of both techniques is used. However, the performance is highly dependent on the data. Artificially created sets were much easier to classify than the real world data.

4.3 Classification of Offensive Videos

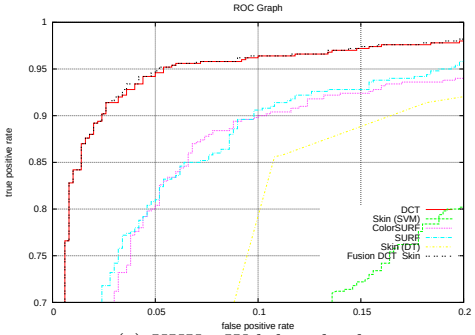
This section deals with the classification of offensive videos. The experiments are divided into experiments that are based on keyframes and experiments that are based on motion features. Finally, it is investigated if a fusion of both levels improves the classification performance. To create sets for training, validation and testing, the whole dataset was split into five sets. Each sets contains roughly 200 videos of both classes. It was not possible to create sets of the exact same size, since the videos do not have the same length. Since the dataset for offensive videos contains small samples out of larger films, it was assured that all samples from one film are in the same set. A SVM was used as classifier again. Every experiment was repeated five times on new randomly created combinations of the sets. Three sets were combined for the training set and one set was used for testing and



(a) XXX - Corel



(b) XXX - Flickr



(c) XXX - Webdownload

Figure 4.13: *ROC curves for different descriptors on offensive images of the Web*

validation. The performance was measured by mean and variance of the equal error rate.

4.3.1 Classification Based on Keyframes

The first experiments on the video data were performed on the keyframes. Since the DCT descriptor showed the best performance for the bag-of-visual-words approach, we use only this descriptor for these experiments. Additionally, the performance of skin features was evaluated using also the SVM classifier. Table 4.10 shows the mean equal error rate and the variance of the equal error rate over five runs. The visual words approach clearly outperforms the skin features. Compared to the previously taken experiments on offensive images, the results are worse for both methods.

There are several reasonable explanations for this behavior. First, one of the most prominent reasons for misclassification of offensive keyframes is that the persons in the videos are often partly dressed (Figure 4.14(a), Figure 4.14(b)). Therefore skin color based features are of limited use. Another reason are poor lighting conditions, e.g the video is too dark (Figure 4.14(c)) or too bright. Also some videos have very poor quality which also complicates the detection of skin (Figure 4.14(d)). Basically, these are the same reasons as for the failure of classification of offensive images. However, extreme cases apply more often for the video data. This can also be compared to the performance on Web images, where the recognition rate was less than for the standard dataset.

Misclassification of inoffensive keyframes has also similar reasons as misclassification of inoffensive images. The main reason is the occurrence of large skin patches although they are not offensive, e.g. a face (Figure 4.15(a)), or hands (Figure 4.15(b)). Furthermore, images that contain many skin-like colors are often misclassified. The fire in Figure 4.15(c) is one example. Like for the offensive images, the poor quality might be a reason. One

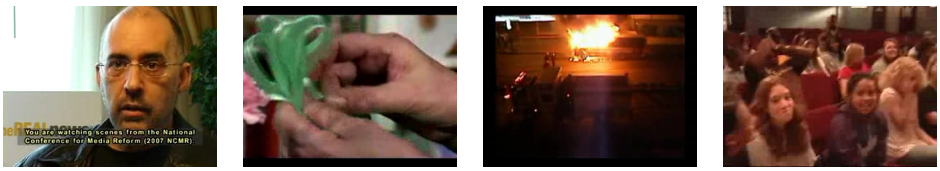
keyframe of a video with poor quality (e.g. poor illumination, or poor resolution) is shown Figure 4.15(d) were a reddish tone is added to the image.

Table 4.10: *This table shows the mean and variance of the equal error rates over 5 runs for keyframe-based offensive video classification with DCT and Skin features. The DCT achieved better results than the skin features.*

Descriptor	μ-EER	σ-EER
DCT	**0.0988**	0.0021
Skin (SVM)	0.1835	0.0064

(a) (b) (c) (d)

Figure 4.14: *Some usually misclassified keyframes from offensive videos: a) and b) exposed skin only visible in small amounts, c) the frame is too dark, and d) the frame is too bright and the overall quality is low.*



(a) (b) (c) (d)

Figure 4.15: *Some usually misclassified keyframes from inoffensive videos: a) a face, b) hands, c) an explosion, and d) many persons. All show large areas with skin like color.*

4.3.2 Experiments for Periodicity Detection on a Small Dataset

The periodicity features are first tested on a subset of the whole data. Their goal is to detect periodic motion patterns in offensive videos which should correspond to scenes where people have sex. Because not all offensive videos show sex scenes, a specialized dataset is used for these experiments. The set consists of 161 offensive videos which show only sex scenes and 200 randomly selected YouTube videos. Each set is split up into training and test sets with a 75:25 ratio. Although these experiments are primarily performed to evaluate the periodicity features, the other approaches are applied as well to get comparable results.

A decision tree classifier is used for the PeriodicityWin features (see 3.2.2) to see the influence of the different features in the classification process. Figure 4.16 shows the resulting tree, which was built on features extracted from 2,002 offensive, and 1,994 inoffensive video windows. In the tree, `per_x` denotes the estimated periodicity for the x-direction, `var_x` its variance, and `area_x` the area between the line through the ACFs local maxima and the line through the ACFs local minima. `per_y`, `var_y`, and `area_y` denote the same for

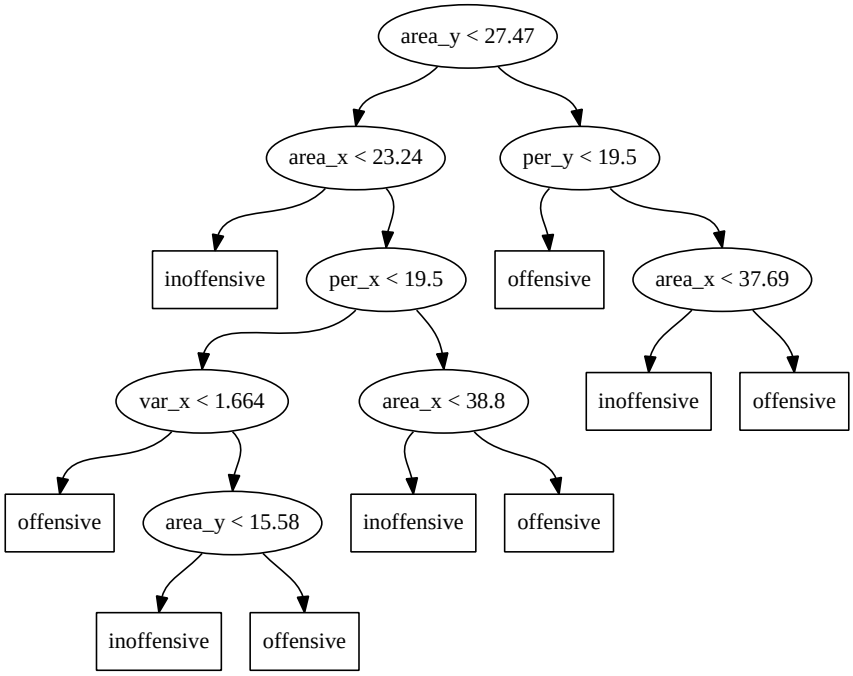


Figure 4.16: *A decision tree for the PeriodicityWin features which has been trained on a small sample set. Sliding windows in offensive videos are recognized, if the area between the lines through the local maxima and minima in the ACF of the mean motion signals is big, and if the motion signals have short periods with little variance.*

the y-direction. The first split is done according to the `area_y` which marks it as the most important feature. The second split is either done with the area feature of the x-direction, or the periodicity of the mean y-motion. Basically, the rules generated are as expected. Sliding windows in offensive videos are recognized as having large areas between the lines through their ACFs extrema and short periods with little variance.

Table 4.11 shows the classification results using different features and classifiers. A support vector machine is used for the DCT and Motion Histogram descriptors as well as for the skin and periodicity features. The PeriodicityWin features are classified on the window level with the decision tree displayed in Figure 4.16 and fused for the video level utilizing the three different methods presented in the approach section. `PeriodicityWin-MAX` denotes the *maximum* vote, `PeriodicityWin-AVG` the *average* vote, and `PeriodicityWin-AVGMAX` the *combination* of both votes.

The best performance is shown by the DCT descriptor with the bag-of-visual words approach. Its performance is even better than the performance on the whole dataset which was presented in the previous section. The same can be seen for the skin detection method whose performance is still worse than the DCT descriptor, but the classification rate on the reduced dataset is better than for the whole set. This can be explained by the fact, that the offensive videos contain only sex scenes. Because of this, there is more skin color present in the frames, which makes the separation using color features easier.

The second best performance is achieved by the motion histograms. Using the periodicity detection on the whole video is clearly the worst approach regarding its equal error rate. Periodicity detection on sliding windows performs generally better than this method. The average and maximum plus average vote of the fusion techniques show the best performance which is an improvement to the normal periodicity detection, but still worse than the motion

Table 4.11: *Classification performance of different descriptors for the specialized dataset measured with the equal error rate. The best performance is achieved by the DCT descriptor. The periodicity show the worst equal error rate, while the PeriodicityWin features perform better. Of all motion related descriptors are the motion histograms performing best.*

Descriptor	EER
DCT	**0.0325**
Skin (SVM)	0.1325
Motion Histogram	0.078
Periodicity	0.21
PeriodicityWin-MAX	0.15
PeriodicityWin-AVG	0.11
PeriodicityWin-AVGMAX	0.11

histograms or even the DCT descriptor.

Reasons for the poor performance of the periodicity features include that the features are not robust enough against small breaks in the motion signal. Large camera movement, motions in other areas of the frames which do not belong to the actual periodic movement, are some events the periodicity feature cannot cope with. Therefore, the periodicity detection is noisy for complete videos. Using the sliding window approach should be more robust against these cases, which can be seen by the improved equal error rate. The average vote shows the best performance for the PeriodicityWin features. It is more stable against periodic motions occurring in windows of inoffensive videos. Applying a maximum vote increases the false positive rate since a video is classified as offensive, if a single window is labeled offensive. The combination of both votes did not improve the equal error rate further.

However, the performance of the PeriodicityWin feature is still worse than using motion histograms. One reason for this is that the periodicity features cannot reliably detect periodic patterns, if the motion only occurs in a small area of the frame. If the camera is zoomed out, this is often the case. Another problem is the extraction of motion vectors itself. The quality of the extracted motion vectors highly depends on the image quality [32]. Some of the videos in the dataset, however, have very low quality. Further, the extraction of motion vectors works better for textured image regions. Skin areas, in contrast, are usually smooth and have little texture, which complicates a reliable motion vector extraction since most movement occurs in the skin areas. The motion histograms capture the different occurring motion patterns in a better way. They express different motion patterns at different positions. Therefore, they should be more redundant to noise.

4.3.3 Classification of Offensive Videos Based on Motion Features

The previous section presented the results of motion related features on a smaller dataset to prove that the approaches work. While the motion histogram performed better than the PeriodicityWin features, the PeriodicityWin features showed, that they are able to capture periodic motion patterns. This section covers experiments of motion features on the larger dataset which are performed in the same way as for the smaller dataset. The PeriodicityWin features use a decision tree classifier for the video windows and afterwards one of the three fusion votes that were presented earlier. The remaining features are classified with a SVM.

Table 4.12 shows the classification results, where motion histograms show the best performance. Periodicity detection on the whole video, performs badly, while the PeriodicityWin features show an improvement to the normal periodicity detection. Reasons for this behavior were already presented in the previous section. Additional reasons evolve out of the com-

position of the datasets themselves. First, not all offensive videos contain sex scenes, and therefore there are no typical periodic patterns present. YouTube videos may have periodic motion, for example in dancing people in music videos, the rhythmically waving of arms in a concert video, or the regular ball motion in a basketball video. For these reasons, the periodicity detection on a whole video is of no use in classifying offensive videos.

For the PeriodicityWin features, the average vote proved to be the best fusion rule. This is the same result as for the smaller dataset. While using windows for periodicity detection is still an improvement over using the complete video, the overall performance is pretty low, compared to the motion histograms. The histograms features themselves, still perform worse than the DCT features based on the keyframes.

ROC curves of all used approaches are shown in Figure 4.17. The curves show the same result as the equal error rates. Using a keyframe based classification with local patches described with DCT, shows the best performance of the single features. Periodicity detection on the whole video and the maximum vote for fusing periodicity windows perform worst. The curve for PeriodicityWin-MAX contains only few sample points, because each final score corresponds to one terminal node in the decision tree. Since there are not much terminal nodes present, the curve is not so fine grained as the other ones.

Table 4.12: *This table shows the classification results of motion related features over five runs. The measure is the mean and variance of the equal error rate. Motion histograms are performing best.*

Descriptor	μ-EER	σ-EER
Motion histograms	**0.1252**	0.0104
Periodicity	0.3785	0.0062
PeriodicityWin-AVG	0.2833	0.0017
PeriodicityWin-MAX	0.3449	0.006
PeriodicityWin-AVGMAX	0.3045	0.0056

4.3.4 Late Fusion of Results for Video Classification

The final experiments investigate the use of a late classification of previously created results. To do this, the same method is used as for the late fusion for image classification. During each of the five performed runs, a model is trained on the training set and classification scores are created for each sample in the validation set. These scores are used to learn the weights for the summation. The final fusion results are created for the test set, by applying the models first, and fusing them afterwards with a weighted sum, according to the learned weights. Since the whole dataset was partitioned into five sets, three of these are used for training, one for validation, and one for testing.

Table 4.13 gives the final results for five runs. Both keyframe based approaches are fused with motion related features, and the fusion of motion histograms with each of the PeriodicityWin features is evaluated as well. The overall best result is achieved by the fusion of DCT and motion histograms, which even outperforms the previously best performance by using only the DCT descriptor. Fusing DCT with PeriodicityWin did not improve the results significantly. It is still better than using only the periodicity detection, but only slightly better than the DCT and only by using the average vote for the window scores. The improvement for both other combinations can be neglected.

Similar results can be seen for fusing the skin features with motion features. Using motion histograms gives a serious improvement over using only the skin features. However,

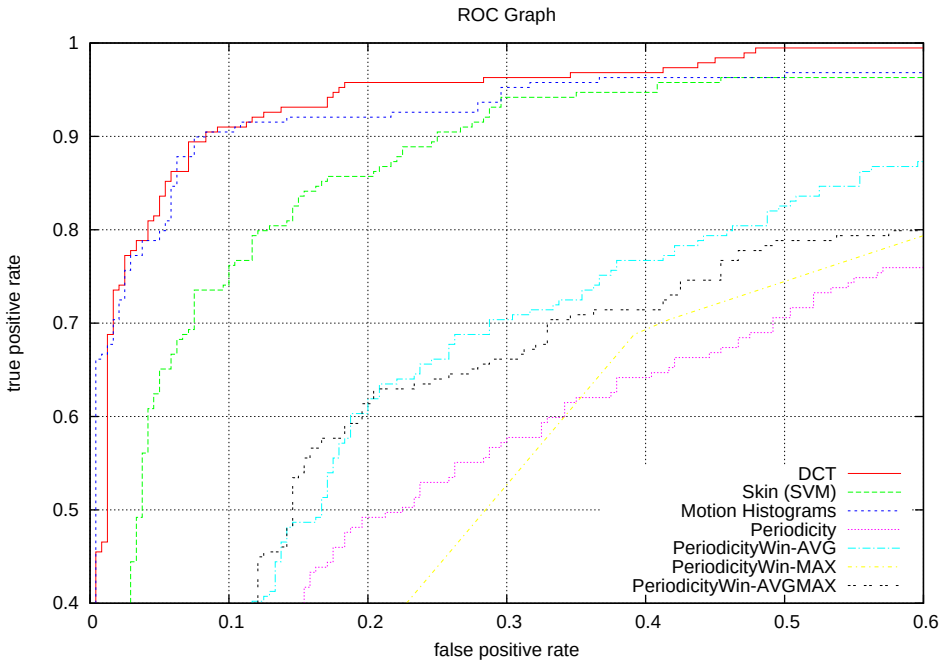


Figure 4.17: *The ROC curves show the best performance of the DCT descriptor on the keyframe level, for the classification of offensive videos.*

the improvement is not as strong as using only motion histograms without fusion. Fusion with PeriodicityWin performs worse still, and only the average vote leads to better results than the single skin features. The fusion of motion histograms with PeriodicityWin features did not achieve an improvement either. As for the single descriptors, Figure 4.18 shows ROC curves for all evaluated fusions. The plot shows the superior performance of fusing DCT with motion histograms while the other fusions show only little improvement, if any.

Table 4.13: *This table shows the results for the late fusion of different descriptors for classification of offensive videos measured by mean and variance of the equal error rate over five runs. The fusion of DCT and motion histograms is an improvement to both single descriptors.*

Fusion		μ-EER	σ-EER	w_1	w_2
DCT	Motion Histograms	**0.0604**	0.0027	0.54	0.46
DCT	PeriodicityWin-AVG	0.0856	0.0018	0.45	0.55
DCT	PeriodicityWin-MAX	0.0945	0.0032	0.45	0.55
DCT	PeriodicityWin-AVGMAX	0.0936	0.003	0.45	0.55
Skin	Motion Histograms	**0.1097**	0.0119	0.59	0.41
Skin	PeriodicityWin-AVG	0.1743	0.0076	0.56	0.44
Skin	PeriodicityWin-MAX	0.2002	0.0252	0.56	0.44
Skin	PeriodicityWin-AVGMAX	0.1833	0.0168	0.56	0.44
Motion Hist.	PeriodicityWin-AVG	0.1087	0.0054	0.45	0.55
Motion Hist.	PeriodicityWin-MAX	0.1093	0.005	0.45	0.55
Motion Hist.	PeriodicityWin-AVGMAX	0.1186	0.0076	0.45	0.55

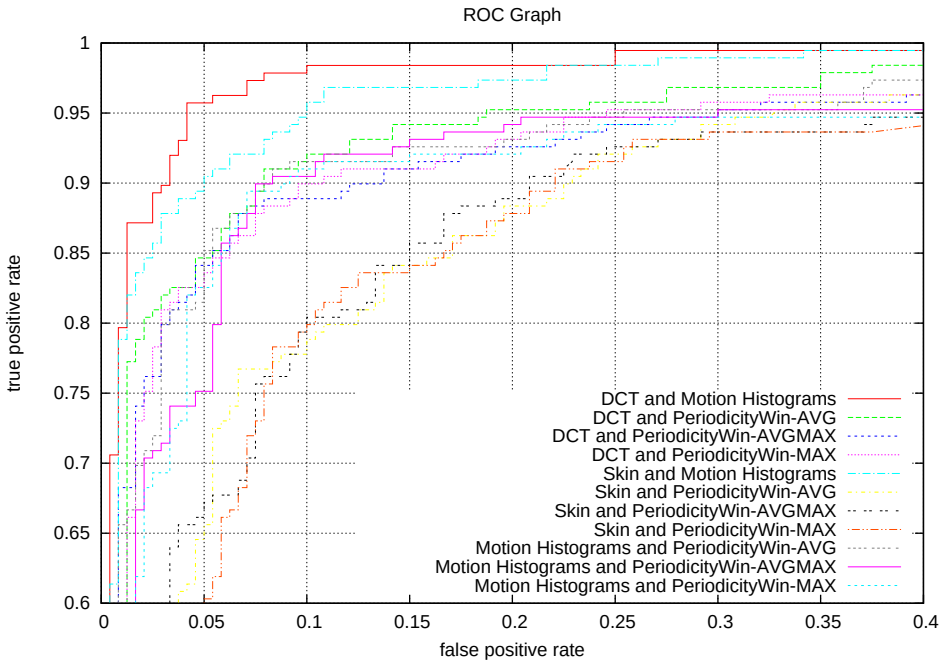


Figure 4.18: *This plot shows ROC curves for the fusion of different descriptors for classifying offensive videos.*

Chapter 5

Conclusion

In the final part of this work a short summary of the results is presented and an outlook of possible future work is given. In previous chapters, a variety of approaches for adult content classification were developed and tested on different data. Also, samples and reasons for misclassifications were given. In the following section, these results are summarized and an interpretation of the most important insights is illustrated. The second part covers some basic ideas of improving the approaches and other possible future work in the area of classifying offensive material.

5.1 Summary of Methods and Results

In this work, several methods for classifying offensive images and videos were presented. For image classification, two approaches were shown: one is based on skin detection, the other one uses the bag-of-visual-words method where different descriptors for local image patches were compared. The latter showed a better performance for both the standard dataset and the Web images, while the first one can be seen as a baseline system with a similar performance to the state-of-the-art. Among the descriptors for image patches, the DCT descriptor proved to be the best achieving an equal error rate of 0.01. SURF showed the poorest performance, because it does not include color information. Also, the classification results heavily depend on the data which they should separate. On the dataset which was presented by Kim [25] better results could be achieved than for real world Web images. Video classification has been divided into keyframe based classification and classification based on motion information. Results show that using additional motion information helps to improve the classification performance. This section presents the most important results and gives explanations for these.

First, color information is of high importance for the classification of offensive material. The SURF descriptor did not achieve the classification results of the other methods, which use color. While this result was already mentioned in [12], it is nevertheless important. One characteristic of offensive images is the presence of skin which cannot be expressed by descriptors which only use gray scale. For video classification, the motion features alone could not achieve the performance of the keyframe based methods, which use color information. This is a further indication that color is the most important feature for classification of offensive material.

Second, both the skin based method and the bag-of-visual-words approach show drawbacks if skin appears in an unusual fashion. Offensive images are usually misclassified if too little skin is present in the image, or it cannot be recognized. Reasons for this include too much or too little illumination, an overshadowing of the skin area, or simply the fact that the persons wear too many clothes. On the other side, inoffensive images are usually labeled

offensive, if they show objects whose color resembles skin color, or if they contain large areas of skin, although the image is not offensive. The latter is often the case for portrait images, images that show many people, or images that show partly dressed people in an inoffensive way. Both approaches have similar misclassifications, although the bag-of-visual-words approach shows fewer misclassifications.

Third, the image quality is important for accurate classification. The classification performance on the standard dataset was higher than on Web images. Web images tend to have a lower quality regarding resolution and compression than the images from the standard set. Wrongly illuminated images are more often present in the Web images, because many images are made by amateurs. Also these images show much more variety in the displayed scenes. Image quality also is responsible for decreasing classification results for keyframes of offensive and YouTube videos compared to classification results on offensive images.

Another result is that combining color and motion information helps to improve classification results on offensive video data. Periodicity detection works if a sliding window approach is applied. Motion histograms achieve a better result, because they are able to combine information about the location and the characteristics of motion. Also they are more robust to noise. However, the combination of keyframe based techniques and motion histogram show a serious improvement, which indicates that motion information can be exploited to get better results.

5.2 Future Work

While the presented methods already achieved good results, there are still some issues, which were addressed in the previous section. The periodicity detection was very prone to noise. Extracting these features in a sliding window improved the results, but still had some problems. Using another method than the ACF to find periodic patterns might improve this approach. One possibility is the combination of periodograms and ACF presented by Vlachos et al. in [44]. Also it might be interesting to evaluate the periodicity detection on another dataset which has a better quality (resolution, compression rate) than the ones used for this thesis, since the low quality is one reason for the occurring problems.

An interesting adaption to the periodicity detection would be to use only motion from skin colored areas. Because the classification with skin features achieved the worst results for these videos this might not be very promising. For another dataset, however, an improvement could be achieved. Another idea is to test the ability of finding scenes which show sexual actions in larger movies. So far only short video clips from offensive websites were separated from YouTube clips. The short offensive videos were mostly parts of larger films. However, detecting sex scenes in larger films to filter or block them might be beneficial.

Incorporating additional, non-visual features might also improve the classification results significantly. Examples for these are text features from web pages, tags of images, audio streams in videos, etc. While this work is focused on visual information, other clues could also be used to increase classification performance. The text features might be frequent occurring words on offensive websites. Image tags contain meta information about the image which if might give information about the displayed content. The audio stream of sex scenes might, like the motion contain periodic patterns which if detected can help to identify these kind of scenes.

Improving the results for image classification depends on the image material itself. One of the most frequently occurring cases for false positives are portrait images. Adding a face detection might get rid of this problem. This might also help to detect the number of occurring persons in an image which might be a further feature for classification. Another problem is the low image quality of Web images which complicates the extraction of useful features. However, the most promising way to improve classification performance, is to

fuse different approaches for a final result. Combining both skin and DCT achieved an improvement, as well as combining image and motion features.

Of great benefit to this research area would be to have some representative dataset every researcher could use. So far most researchers use their own data to train and test their methods. Because of this, comparative results cannot be created easily. Therefore, a very thorough evaluation has been performed on various datasets to get the results for this thesis. The dataset, which was used in two previous papers, proved not to be representative enough of real Web images. This is an important requirement, because the dataset should contain samples of the final field of application. Problems might arise, since offensive data is often copy right protected, or is even illegal in some countries.

Bibliography

[1] A. Abadpour and S. Kasaei. Pixel-based skin detection for pornography filtering, 2005.

[2] Hans J. Andersen and Erik Granum. Skin colour detection under changing lighting conditions. In *7th Symposium on Intelligent Robotics Systems*, pages 187–195, 1999.

[3] Will Archer Arentz and Bjørn Olstad. Classifying offensive sites based on image content. *Computer Vision and Image Understanding*, 94(1-3):295–310, 2004.

[4] Herbert Bay, Tinne Tuytelaars, Van Gool, and L. Surf: Speeded up robust features. In *9th European Conference on Computer Vision*, Graz Austria, May 2006.

[5] A. Bosson, G. Cawley, Y. Chan, and R. Harvey. Nonretrieval: blocking pornographic images, 2002.

[6] Leo Breiman, Jerome H. Freidman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group, 1984.

[7] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[8] Colin Campbell. Algorithmic approaches to training support vector machines: A survey. In *In Proceedings of ESANN2000*, pages 27–36. D-Facto Publications, 2000.

[9] Y. Chan, R. Harvey, and D. Smith. Building systems to block pornography, 1999.

[10] Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, and Gabriela Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[11] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Discriminative training for object recognition using image patches. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 157–162, Washington, DC, USA, 2005. IEEE Computer Society.

[12] Thomas Deselaers, Lexi Pimenidis, and Hermann Ney. Bag-of-visual-words models for adult image classification and filtering. Tampa, Florida, USA, 08/12/2008 2008.

[13] R. Du, R. Safavi-Naini, and W. Susilo. Web filtering using text classification. In *Networks, 2003. ICON2003. The 11th IEEE International Conference on*, pages 325–330, 2003.

[14] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

[15] Tom Fawcett. Roc graphs: Notes and practical considerations for data mining researchers, 2003.

[16] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. pages 726–740, 1987.

[17] Margaret M. Fleck, David A. Forsyth, and Chris Bregler. Finding naked people. In *ECCV (2)*, pages 593–602, 1996.

[18] C. W. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification. Technical report, Taipei, 2003.

[19] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IEEE Transactions on*, 8(2):179–187, 1962.

[20] Zhiwei Jiang, Min Yao, and Wensheng Yi. Filtering objectionable image based on image content. In *PRICAI*, pages 1027–1031, 2006.

[21] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. pages 137–142. Springer Verlag, 1998.

[22] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, January 2002.

[23] Syaed Ali Khayam. The discrete cosine transform (dct): Theory and application. Technical report, Michigan State University, 2003.

[24] Chang-Yul Kim, Oh-Jin Kwon, Won-Gyu Kim, and Seok-Rim Choi. Automatic system for filtering obscene video. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, volume 2, pages 1435–1438, 2008.

[25] Wonil Kim, Han-Ku Lee, Jinman Park, and Kyoungro Yoon. Multi class adult image classification using neural networks. In *Canadian Conference on AI*, pages 222–226, 2005.

[26] Hogyun Lee, Seungmin Lee, and Taekyong Nam. Implementation of high performance objectionable video classification system. *ICACT*, pages 959–962, 2006.

[27] Pui Y. Lee, Siu C. Hui, and Alvis Cheuk M. Fong. Neural networks for web content filtering. *IEEE Intelligent Systems*, 17(5):48–57, 2002.

[28] Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[29] David G. Lowe. Object recognition from local scale-invariant features. pages 1150–1157, 1999.

[30] B. S. Manjunath, Jens rainer Ohm, Vinod V. Vasudevan, and Akio Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:703–715, 2001.

[31] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.

[32] Mauriziu Pilu. On using raw mpeg motion vector to determine global camera movement. Technical report, Hewlett Packard Laboratories Bristol, 1997.

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

[34] Hazel Rosetti. *Colour: Why the world isn't grey.* Princeton University Press, Princeton, NJ, 1983.

[35] Henry A. Rowley, Yushi Jing, and Shumeet Baluja. Large scale image-based adult-content filtering. In *VISAPP (1)*, pages 290–296. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2006.

[36] M. Shin, K. Chang, and L. Tsap. Does colorspace transformation make any difference on skin detection.

[37] Jonathon Shlens. A tutorial on principal component analysis, 2005.

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

[39] Adrian Ulges, Christian Schulze, Daniel Keysers, and Thomas Breuel. Identifying relevant frames in weakly labeled videos for training concept detectors. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 9–16, New York, NY, USA, 2008. ACM.

[40] Adrian Ulges, Christian Schulze, Daniel Keysers, and Thomas Breuel. A system that learns to tag videos by watching youtube. In *International Conference on Computer Vision Systems*, 2008.

[41] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek. A comparison of color features for visual concept classification. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 141–150, New York, NY, USA, 2008. ACM.

[42] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques, 2003.

[43] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.

[44] Michail Vlachos, Philip S. Yu, and Vittorio Castelli. On periodicity detection and structural periodic similarity. In *SDM*, 2005.

[45] James Z. Wang, Gio Wiederhold, and Oscar Firschein. System for screening objectionable images using daubechies' wavelets and color histograms. In *IDMS '97: Proceedings of the 4th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services*, pages 20–30, London, UK, 1997. Springer-Verlag.

[46] Chong-Yaw Wee, Raveendran Paramesran, and Fumiaki Takeda. New computational methods for full and subset zernike moments. *Inf. Sci. Inf. Comput. Sci.*, 159(3-4):203–220, 2004.

[47] Yiming Yang and Se An Slattery. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18:219–241, 2002.

[48] Seong-Joon Yoo. Intelligent multimedia information retrieval for identifying and rating adult images. In *KES*, pages 164–170, 2004.

[49] Wei Zeng, Wen Gao, Tao Zhang, and Yang Liu. Image guarder: An intelligent detector for adult images. *Asian Conference on Computer Vision*, pages 198–203, 2004.

[50] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vision*, 73(2):213–238, 2007.

[51] Qing-Fang Zheng, Wei Zeng, Gao Wen, and Wei-Qiang Wang. Shape-based adult image detection. In *ICIG '04: Proceedings of the Third International Conference on Image and Graphics*, pages 150–153, Washington, DC, USA, 2004. IEEE Computer Society.