Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computer Vision and Image Understanding 94 (2004) 295–310
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Classifying offensive sites based on image content

Will Archer Arentz* and Bjørn Olstad

Department of Computer and Information Science, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway

Received 1 December 2002; accepted 29 October 2003

Abstract

This paper proposes a method for helping to identify adult web sites by using the image-content as means of detecting erotic material. The image content is classified by investigating probable skin-regions, and extracting their feature vectors. These feature vectors are based on color-, texture-, contour-, placement-, and relative size-information for a given region. The importance of the different elements in the feature vector is determined by a genetic algorithm. For each picture, the algorithm gives the probability that a certain picture has erotic content. By mapping all the images in a web site, and running the image-based classifier on the whole collection, we were able to set up a histogram of images with regards to the log-likelihood of erotic content for each image. Hence giving a good overview of the web site's content and at the same time leaving room for errors in the image-based classifier.

The algorithm proved to be quite successful in our tests where all 20 sites where classified correctly. The image-based classifier is able to properly identify 89% of the evaluation images at an average processing speed of 11 images per second.

Although this experiment focused on classifying adult web sites, small alterations to the system can be done, enabling classification of other kinds of images and web sites.
© 2003 Elsevier Inc. All rights reserved.

Keywords: Object recognition; Erotica/pornography; Internet; Web site classification; Color; Image retrieval

* Corresponding author.
E-mail addresses: willa@idi.ntnu.no (W.A. Arentz), Bjorn.Olstad@fast.no (B. Olstad).

1077-3142/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2003.10.007

1. Introduction

The recent development of multimedia availability on the Internet has enabled Internet users around the world to access enormous amounts of video-, picture-, and audio-files. However, this has also lead to new business-opportunities for the adult-industry. Furthermore, due to this industry's aggressive marketing, it has become very difficult to avoid stepping into an adult site from time to time, be it through web-surfing, spam-mail, or search results from a web search-engine. This may be very upsetting to some people, especially parents with Internet-surfing children. A system preventing the browser to enter such sites has been requested on several occasions. This problem is especially evident in web search engines, where a company often has some responsibility for the quality of the search results.

The solution is to use a content-based image-retrieval algorithm to identify naked or scantily dressed people in images. This is not a new idea; Forsyth and Fleck [8] managed to create a fully automated system for detecting such pictures. However, their system was much too slow to be used as part of the data gathering process in a search engine, where speed is of the essence. Currently existing systems use advanced text-analysis and linguistics to determine if a web-page is offensive or not. Unfortunately, some adult sites just have a listing of un-suspicious-looking file-names, while other sites have all the text incorporated into bitmaps displayed on the web page. Hence, image understanding is the only way to determine what kind of site it actually is.

When dealing with detection of nudity, it is important to have a good method of recognizing skin regions. Much work has been done in this regard.

Detection of human skin and the effect different cameras, light-settings, human race, and color spaces has on the recognition process is discussed in [2,4,10,12,13,19,20,24]. Furthermore, [8,17] also use texture information as a component in the skin detection. As a third component in object recognition, many have looked at the shape of the object as a last stage in the information gathering process [1,7,11,15,18]. Veltkamp and Hagedoorn [23] have written a survey of different state-of-the-art shape matching methods.

With a set of features describing the image, such as color, texture, and shape for segmented objects, it is possible to build a joint feature vector, where all the information is stored. And thereafter, using a training-method for weighting the importance of each element in the vector. Using a genetic algorithm (GA) is one way to do this, and several people have attempted to use GA in similar feature vector based image retrieval schemes [5,6,16,21,22].

Classifying Internet sites is currently being done by several search engines. The most famous is perhaps Google, where PageRank [14] is used to classify the importance of web pages and web sites based on the hyperlink information. Even though there has been much work done on classifying documents, little published work has been done in labeling the actual web sites. This is what we were aiming to do, based on the content of images on the site.

In the next section of this paper, we explain the design of the content-based image retrieval algorithm, while the site classification is discussed in Section 3. Furthermore, Section 4 reviews the results of our experiments, before some ideas about future work are revealed in Section 5, followed by a final conclusion in Section 6.

2. The image content examiner (ICE)

Identifying regions with skin in the images is the key element in identifying nudity. Also, since we want fast processing, filtering out all non-skin regions can significantly reduce the amount of data processed. Although there is a lot of nudity on the web, most of the images do not belong to a this category, and would hence be trivially rejected as non-offensive.

According to Shin et al. [19] YCbCr is the best 2D (CbCr) color space for skin detection of the nine tested. However, it is argued that the RGB color space gives the best performance in 3D, and surpasses the performance of CbCr. Nevertheless, since we wish to have a fast system, a smaller data representation implies faster processing. Thus, we choose CbCr to be the color representation in our system.

The initial filter, which removes non-skin pixels, is simply based on rejecting pixels outside a given range of Cb and Cr in the CbCr color space. The pixel range was determined by generating a color histogram for the Cb and Cr color components, using a database of 500 skin-dominant images. As can be seen from Fig. 2, Cb values from 78 to 135 and Cr values from 85 to 185 have very high readings, thus suggesting the use of these values as initial filter thresholds. We choose to use a rather large range at this stage, allowing some non-skin pixels to pass through the filter, since we wish to accept as much of the probable skin pixels as possible.

The pixels that remain after the initial filtering are grouped into connected regions and labeled, using a simple recursive labeling algorithm. Generally, recursive algorithms are considered slow, but the use of pointers in an efficient implementation made this process quite fast.

The labels are stored in the array L, and the array P contains the number of pixels in a given region or object. This array is sorted so that the N largest objects may be chosen and further analyzed. In our experiments we set $N_{\max} = 10$.

Furthermore, an overview of the image processing system can be found in Fig. 1, while the training process is depicted in Fig. 4.

2.1. The composite object feature vector

The next stage in the analysis is to extract a feature vector for each of the objects. The feature vector is composed of information about the color, texture, and shape of the object. In addition some information about placement and size is included. The composition of the color feature, is simply two histograms of Cb and Cr pixel-values.

Next, the texture feature is extracted based on the complexity curve [3]. This method has the advantage of being very computationally efficient, and Baheerathan et al. shows that the method gives a robust second-order description of the textures. The actual implementation is done by calculating the difference between a pixel and

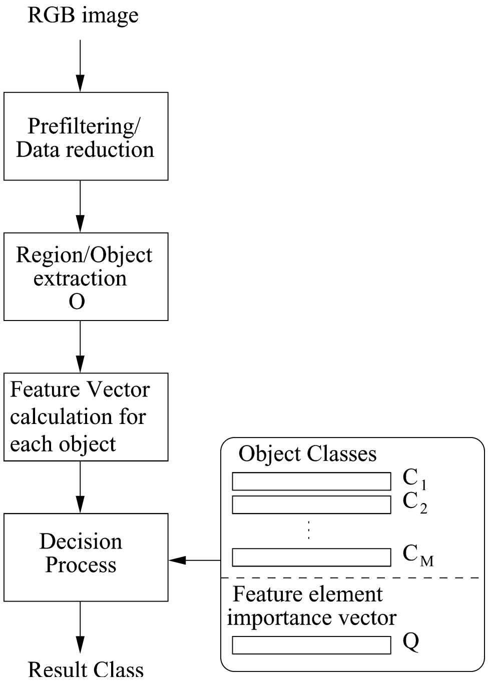

Fig. 1. An overview of the image analysis process.

its right neighbor in the luminance picture, and entering these values into a histogram. This histogram is then entered into the composite feature vector. High intensity in the lower part of this histogram would therefore, suggest smooth transitions, while high intensity in the higher areas would suggest a noisy picture. Both the color and texture section of the feature vector is normalized by taking the feature element and dividing it by the total number of pixels in the object.

The third piece of information extracted from an object is a description of the object's shape. This task is accomplished by clockwise tracing the outer borders of the object and storing the distance from the object's centroid. The sequence of distances are stored in an array and normalized so that the sum of distances to the center is constant, before finally applying the Fast Fourier Transform. The 28 most significant energy-coefficients from the Fourier transform are then added to the object's total feature vector.

The three final components stored in the feature vector are the relative x- and y-positioning of the object's centroid, and the relative size of the object. This produces a feature vector of 801 real number-elements for each object, as can be seen in Fig. 3.

2.2. Training by genetic algorithms

The preceding chapters explained how to calculate a set of feature vectors for some skin-dominant objects extracted from an image. Even though the composite

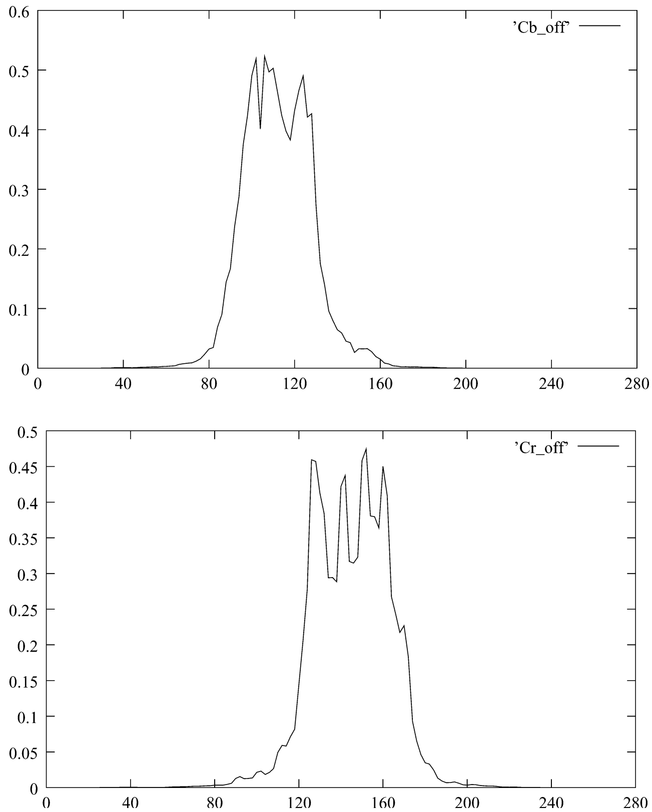

Fig. 2. Histograms of the pixels in the Cb (top) and Cr (bottom) color space.

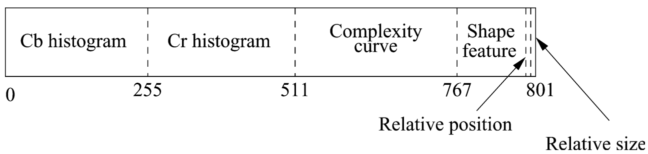

Fig. 3. The composition of the feature vector.

feature vector contains much information, we know little about the importance of the different elements. Hence, it is necessary to find a weighting scheme for this vector.

Furthermore, a set of objects classes **C** should be defined as reference, so that the retrieval process can be done by classifying which class each object belongs to. Then the probability that an image is offensive, can be calculated by accumulating the probabilities of the object-classes to which the image's objects belong. In the experiments 30 object classes were used.

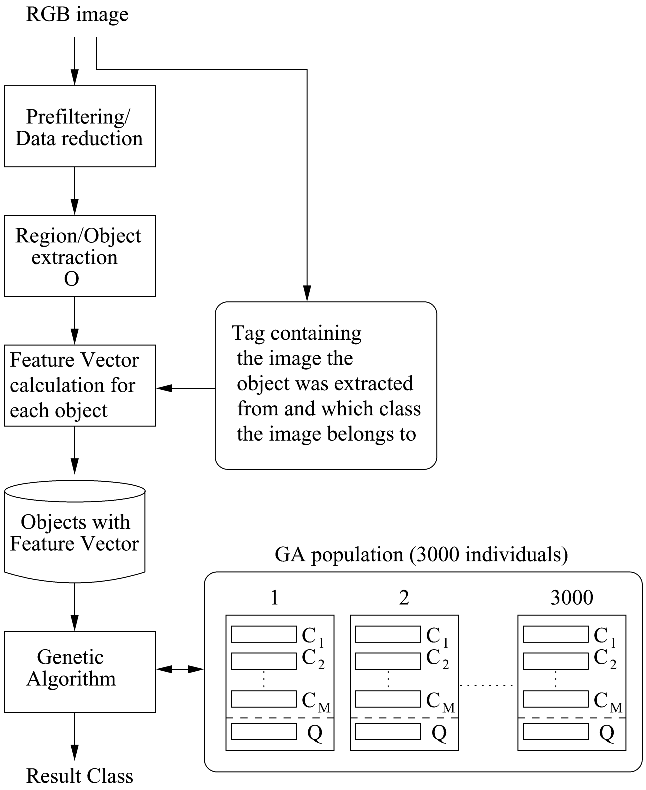

Fig. 4. An overview of the training process.

Choosing the object classes could of course be done by some kind of un-supervised clustering algorithm like for example c-means clustering. However, in an attempt to find a more optimal distribution in the feature space, it was decided to use a genetic algorithm to find these classes, as well as finding the importance of the different elements in the feature vector. This last part was done by defining a vector Q with the same dimension as the feature-vector for an object O. The Q-vector is initialized with random values from 0.0 (not important) to 1.0 (most important). This vector is multiplied with the object's feature vector when calculating the distance between the object class C_j and the object O^k. This can be written as a function

$$d(O^k, C_j, Q) = \sum_i ((O_i^k - C_{ji})Q_i)^2, \tag{1}$$

where $d(O^k, C_{j_k}, Q)$ is minimized with respect to the class C_j, so that the object O^k is assigned to the class which minimizes the distance between object and class. We write this as

$$C_{j_k}(O^k) = \operatorname*{argmin}_{C_j \in \mathbf{C}} d(O^k, C_j, Q). \tag{2}$$

A set of objects $O^1 \cdots O^k \cdots O^N$ is derived from an image I, and when processing images on a fully trained system, an object O^k is mapped to an object class $C_{j_k}(O^k)$ where j_k is the class number matching the kth object.

To decide whether an image is offensive or not it is sensible to calculate a statistical probability that an image is offensive based on the gathered information. In this regard A represents the offensiveness, such that the probability that a given object class C_j represents offensive material given the image I is calculated as in (3). The training objects are either from an offensive image or from a non-offensive image. All images are manually classified into the two groups before the training process.

$$P(A|C_j) = \frac{\text{number of offensive images with objects belonging to } C_j}{\text{number of images with objects belonging to class } C_j} \tag{3}$$

We define the probability of offensiveness in an object as in Eq. (4), where the right hand side is the probability defined in Eq. (3). In the manner shown in Eq. (2), the distances from an object to all the M object-classes are calculated to find the class with the shortest distance. This class is then chosen to represent the object. We set the probability of the object being offensive equal to the probability of the class, to which the object is assigned, being offensive. This is expressed as

$$P(A|O^k) = P(A|C_{j_k}(O^k)). \tag{4}$$

By applying Bayes formula, we calculate the probability that an image is offensive as shown in Eq. (5). Since $P(C_{j_1}(O^1), \ldots, C_{j_N}(O^N))$ is the probability that a set of classes $C_{j_1} \ldots C_{j_N}$ occur in an image, the denominator is constant and can therefore be neglected retaining proportionality.

$$\begin{aligned} P(A|O^1, \ldots, O^N) &= P(A|C_{j_1}(O^1), \ldots, C_{j_N}(O^N)) \\ &= \frac{P(C_{j_1}(O^1), \ldots, C_{j_N}(O^N)|A) \cdot P(A)}{P(C_{j_1}(O^1), \ldots, C_{j_N}(O^N))} \\ &\propto P(C_{j_1}(O^1), \ldots, C_{j_N}(O^N)|A) \cdot P(A) \end{aligned} \tag{5}$$

The objects extracted in the preprocessing are mutually exclusive. It is hence reasonable to assume that the object classes are independent. We will use this assumption for finding the discrimination function. Based on this we can write

$$P(C_{j_1}(O^1), \ldots, C_{j_N}(O^N)|A) = \prod_{k=1}^{N} P(C_{j_k}(O^k)|A), \tag{6}$$

which implies

$$\log(P(C_{j_1}(O^1), \ldots, C_{j_N}(O^N)|A)) = \sum_{k=1}^{N} \log(P(C_{j_k}(O^k)|A)). \tag{7}$$

$P(A)$ is the probability of offensiveness which would be 0.5 if the amount of offensive data equal the amount of non-offensive. A large investigation of the ratio of offensive versus non-offensive data on the Internet, may provide a reasonable value for $P(A)$. Nevertheless, the value is a constant and since we ultimately only are interested in the proportional equality we remove this term from the equation. Hence, we can join Eqs. (5) and (7), which gives us the log-likelihood of an image being offensive as

$$\log(P(A|C_{j_1}(O^1),\ldots,C_{j_N}(O^N))) = \sum_{k=1}^{N} \log(P(C_{j_k}(O^k)|A)) + K, \tag{8}$$

where K is a constant.

Given a set of objects extracted from an image and a set of object classes, we can now use (8) to calculate a value describing the offensiveness of the image. However, there is still a matter of setting the threshold for what values should be accepted as offensive. This value β separates the two content types, and can be found by calculating $\log[P(A|Image)] = \log[P(A|C_{j_1}(O^1),\ldots,C_{j_N}(O^N))]$ for a large set of manually classified offensive and non-offensive images. An initial value for β is chosen, and for all $\log(P(A|Image)) > \beta$ the images are marked as offensive. Then the number of erroneous classifications ϵ are counted. Different values of β are then tried out to minimize this error.

Both the vector Q and all object classes in C are found by using a genetic algorithm that attempts to maximize the ability to separate between offensive and non-offensive objects. As mentioned earlier the Q-vector as well as the C-vectors all consists of 801 real value elements. Since 30 classes are used in the experiment, this means that 24,831 values must be set by the genetic algorithm. This is a very large amount of values and other approaches were considered to replace the GA. In this regard one other method was attempted, namely simulated annealing [9]. Unfortunately this approach was too slow to be useful.

The first step of the GA is to set the initial values of the each individual. Since each individual consists of 24,831 values and we use 3000 individuals in our population, 74,493,000 elements are initially set to random values between 1.0 and 0.0. A large population is chosen because the versatile distribution provides some quite fit individuals, in the sense that they achieve very high initial fitness. This seems more powerful in reaching high fitness than running a high number of generations. Furthermore, a multi-point crossover method is chosen to improve the handling of the large data set. The crossover sections are set by running through the whole set of elements in an individual, marking the elements as a crossover start- or stop-point with a probability of 5%. Mutations are performed on an element with a probability of 0.5%. Upon a mutation the element is set to a new random value. The number of generations run is 50, although the the last eight generations provide little improvement in fitness. We used tournament selection as the method for selecting individuals in our GA implementation. The best individual from the previous generation is automatically forwarded to the next generation, while the rest of the population is set by tournament selection of the 50% best individuals from the previous generation.

The GA fitness calculation is the most crucial part of the training process. The value returned from the fitness function measures the ability of the image processing method to separate between offensive and non-offensive images. Arguments to the fitness function are the Q-vector and the class vectors. These are first determined by the GA, before being evaluated by the fitness function. Further arguments to the fitness function are a selection of offensive and non-offensive training images. As shown in

fitness($I_{offensive}$[],$I_{non-offensive}$[],C[][],Q[])
begin
  $O_{offensive}$:= *all objects extracted from* $I_{offensive}$
  $O_{non-offensive}$:= *all objects extracted from* $I_{non-offensive}$
  for *all objects O find the closes class* C_j *so that*
    $C_{j_k}(O^k)$ *denote the class* C_j *containing the object* O^k
  for *each class* C_j *make a list* L_{C_j} *containing*
    all objects belonging to the class C_j
  for *all lists*
    for *all elements in* L_{C_j}
      replace all objects O with a reference to the image they were
      extracted from I
      calculate $P(A|C_j) = \frac{\text{number of offensive images in } L_{C_j}}{\text{total number of images in } L_{C_j}}$
  for *all images* $I[l]$
    extract all objects for I
    $P(A)_{I[l]} := \sum_{k=1}^{N} log(P(C_{j_k}(O^k)|A)$
  set a value $\beta \in 0 \ldots 1$
  while *did not investigate all possible* β
    $\epsilon := 0$
    for *all* $P(A)_{I[l]} < \beta$
      if *any image I[l] is offensive*
        $\epsilon = \epsilon + 1$
    for *all* $P(A)_{I[l]} \geq \beta$
      if *any image I[l] is non-offensive*
        $\epsilon = \epsilon + 1$
    if $\epsilon < \epsilon_{opt}$ **then** $\epsilon_{opt} = \epsilon$
  return *total number of images* $- \epsilon_{opt}$
end

Fig. 5. The genetic algorithms fitness function.

Fig. 5, every fitness calculation involves loading a large set of objects from a training-image into memory, while labeling the objects as offensive or non-offensive. Following this, the closest matching class for each object is found using the function given in Eq. (2). Hence, each object class ended up with a set of object members, where (3) is applied to calculate the probability that the class represents offensive material. Finally, as a step in evaluating the performance and calculating the fitness; the images in the training set are classified as offensive or non-offensive utilizing the probabilities found above. That is, for each image all objects were loaded and their belonging class is determined. When all objects with their matching classes are known, (8) is used to find the total log-likelihood of an image being offensive. At last, β is determined and the total error rate counted. This process results in a number quantifying the classification ability, which in the end is used as the GAs fitness value.

3. Web site classification

The error analysis was based on the hypothesis that a site contains images of only one type. Either all the images are offensive or they are all non-offensive. This seems to be a fair assumption for virtually all offensive sites.

N	p=0.05	p=0.1	p=0.2	p=0.3	p=0.4
2	0.0025	0.0100	0.0400	0.0900	0.1600
3	1.11E-3	6.11E-3	0.0326	0.0840	0.1597
5	2.04E-4	2.17E-3	0.0215	0.0768	0.1784
10	2.75E-6	1.46E-4	6.34E-3	0.0473	0.1662
20	5.37E-10	7.09E-7	5.63E-4	0.0171	0.1275
50	5.54E-21	1.07E-13	4.92E-7	9.33E-4	0.0573
100	3.74E-39	6.32E-25	5.17E-12	9.03E-6	0.0168

Fig. 6. The probability that a site is wrongly classified as a function of N and p.

The developed image retrieval system is used to classify web sites after crawling all images on a given site. Next, all N images on the site are run through the ICE and classified into offensive or non-offensive. If we let Ω denote the number of offensive images on a site, then if $\Omega > \frac{N}{2}$ we assume that the site is an offensive site. This process repeats for all sites to be investigated.

Given the probability that an image is wrongly classified (p) and the total number of images on the site (N), one can use the binomial distribution (9) to find the probability that the whole site is falsely classified.

$$P(X = x) = b(x; n, p) = \binom{n}{x} p^x (1-p)^{n-x} \tag{9}$$

Since we assume that a site only can contain one type of images it follows that a site with N images is misclassified only if $X > \frac{N}{2}$ single images are falsely classified. Thus, we can write the probability that the site is wrongly classified as

$$P\left(X > \frac{N}{2}\right) = \sum_{x=\frac{N}{2}+1}^{N} b(x; N, p) = \sum_{x=\frac{N}{2}+1}^{N} \binom{N}{x} p^x (1-p)^{N-x}. \tag{10}$$

Eq. (10) can be used to build a table that expresses the probability of wrongly classifying a site, given the number of images on the site and the probability of wrongly classifying and image. This is depicted in Fig. 6 for the values $N = 2, 3, 5, 10, 20, 50, 100$ and $p = 0.05, 0.1, 0.2, 0.3, 0.4$.

As shown in the table, the probability of a site being falsely classified is very small, even with a quite high probability of wrongly classifying a single image. For example, a site with 20 images and a 30% probability of falsely classifying an image has a mere 1.7% probability of being incorrectly classified.

4. Experimental results

4.1. The image processing

When doing the experiments on the image retrieval system, the operation was divided into two parts; the training and the evaluation. Both parts indicating something about the performance of the system. The evaluation results were counted as

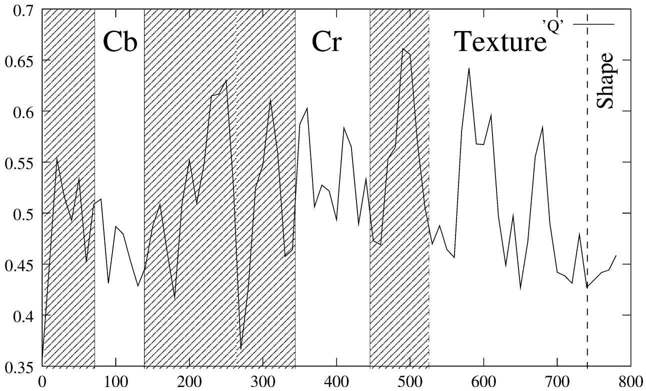

Fig. 7. The Q-vector describing the importance of the elements in the feature vector. The relative importance of the feature elements are texture 58%, Cr 25%, Cb 11% and shape 6%.

the official performance of the system. These results were usually just a few percent lower than the results from the training set. This indicates that the training to a large extent was able to generalize the findings.

The images used in the training were all gathered using AllTheWeb's[1] image-index. As many as 575 non-offensive images were randomly picked from the index, and 365 offensive images were selected by searching on English female names. As a restriction, no images with less than 200 pixels in each dimension were accepted.

The training process was run with a GA population of 3000 over 50 generations. The resulting Q-vector is shown in Fig. 7, and the separation level was found to be $\beta = 0.22$. The shaded areas represent the colors blocked by the initial filter, thus no information is stored in these areas. Like all values in the Q-vector these values are also set randomly during the GA initialization, and they are not excluded from any part of the GA process. For this reason they appear as in the figure and not as zeros. In an optimized version of the algorithm these areas would be removed to reduce computation. However, if the initial filtering is changed to a general segmentation, perhaps to identify other kinds of images, the shaded areas would become an active part of the feature vector.

It can be argued that the color information could be removed from the feature vector, since the initial filtering blocks out everything but the skin colors. Unfortunately, the initial filter accepts a greater range of colors, than that limited to skin. This is done to ensure that virtually all skin will be accepted by the filter. Thus, the later automatically trained process can decide whether certain colors represent offensiveness or not.

When examining Fig. 7 closely, we see that the Cr values are given a higher weight than Cb values. This is because Cr measures the red-green color ratio, whereas Cb

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

Type of images	Number of images	Images wrongly classified
Portraits	268	26.5%
Offensive	87	4.6%
Non-Offensive	25	12%

Fig. 8. The ability to classify different types of images.

measures the relationship between green and blue. Since skin color mostly is composed of red, it seems natural that Cr is more important.

Furthermore, the weights in the texture section shows that the lower gradients are given most importance. This means that smoother transitions are preferred, which corresponds to the nature of skin. The *Q*-vector also show a peak in the higher part of the texture section. This is perhaps because values in this range are used to determine that the object is not offensive since such large gradients do not occur in skin. Finally, the shape feature seems to have been given little importance.

The training resulted in 92.1% correct classifications of the images, while the evaluation on an independent data set suggested that 89.4% of the images were classified correctly. The evaluation data set was found in the same manner as the training set, and consisted of 500 offensive and 800 non-offensive images. However, by running the image classification on a dataset solely composed of portrait images, only 73.5% of the images were classified as non-offensive, in other words 26.5% of the 268 images in the dataset was falsely classified. This error is mainly due to the fact that the nature of portrait images is very similar to that of nude photos, with large regions of skin. As can be seen from Fig. 8, other kinds of images experience different kinds of error rates. The errors can probably be reduced at the cost of speed by replacing the initial skin filter, with a more complex image segmentation scheme.

The speed of the processing proved to be very satisfactory. On a single CPU system, an average of 11 images per second were processed. That equals 660 images per minute or 39,600 per hour. In comparison Forsyth and Fleck [8], operated with a processing rate of 10 images per hour but since their paper was published in 1996 it should be noted that the increased computational speed of modern computers also contributed to the increased processing rate experienced with our method. Needless to say, with faster processors or on a multiple CPU system processing speeds can be improved even further.

4.2. The site classification

A number of different sites were chosen at random and each site was crawled, downloading all images. As much as 20 of these sites were chosen for the experiment in Fig. 10. From the fact that none of the sites contained error-rates close to or above 50% we can conclude that all sites were classified correctly. The average image classification error rate for offensive sites were 14.1%, as compared to 9.8% for non-offensive.

Since a site containing N images, where Ω is the number of images determined by the image processing algorithm to be offensive, then as long as $\Omega \neq N - \Omega$, a choice can be made. If we assume that a site consists solely of one class of images, as we did in our work, then as long as the error rate of the image processing is less than 50% the site will be classified correctly. Note that sites are not always solely composed of one class of images, thus making it harder or sometimes impossible to do a correct classification. Fortunately, these sites are the exceptions.

The histograms showing the log-likelihood of the images being offensive, for two example sites, are shown in Fig. 9. The threshold β separates the two classes into offensive on the left and not offensive on the right. It appears quite clear that the sites contain different types of images, and that they can be classified into different classes.

As mentioned earlier, some types of images, like portrait photos, often experience high error rates. This error rate may seem appreciable, but it is far from being 50%, whereas a conclusive choice cannot be made. Moreover, as Internet sites usually consist of several images, the error rate becomes quite acceptable. As Fig. 6 shows, with a classification error rate of 30%, a site consisting of 10 images has a mere 4.7%

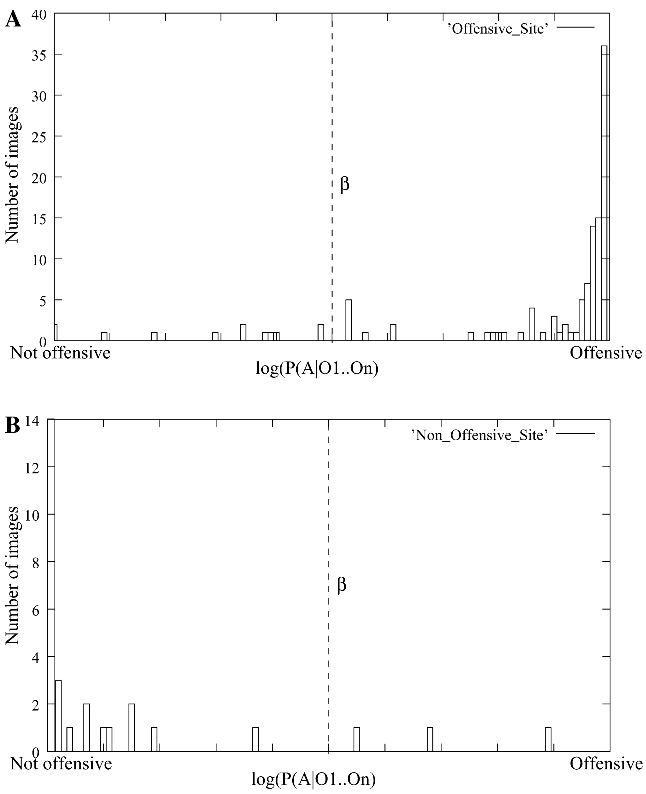

Fig. 9. Histograms of the log-likelihood of the images being offensive compared to the threshold β. (A) shows a histogram composed from an offensive site with 116 images, while (B) shows a histogram from a non-offensive site with 30 images.

Site Number	Images on site	Images classified as offensive	Images classified as non-offensive	Images wrongly classified
Offensive 1	87	83	4	4.6%
Offensive 2	112	87	25	22.3%
Offensive 3	78	62	16	20.5%
Offensive 4	81	68	13	16.0%
Offensive 5	88	76	12	13.6%
Offensive 6	87	68	19	21.8%
Offensive 7	82	77	5	6.1%
Offensive 8	81	70	11	13.5%
Offensive 9	81	72	9	11.1%
Offensive 10	84	74	10	11.9%
Non-Offensive 1	239	29	210	12.1%
Non-Offensive 2	144	26	118	18.0%
Non-Offensive 3	7	1	6	14.3%
Non-Offensive 4	97	2	95	2.0%
Non-Offensive 5	19	1	18	5.3%
Non-Offensive 6	70	2	68	2.9%
Non-Offensive 7	91	18	73	19.8%
Non-Offensive 8	57	6	51	10.5%
Non-Offensive 9	14	0	14	0.0%
Non-Offensive 10	53	7	46	13.2%

Fig. 10. The classified sites.

probability of being wrongly classified as a whole. This is, of course, assuming that offensive and non-offensive material is not mixed on the same site.

5. Further work

Much weight has been put into making the implementation as fast as possible. In this regard, some choices were made that may have had impact on the accuracy of the image retrieval process. Currently, we are working on a Bayesian approach for segmenting skin. The results are promising although the method is quite slow.

Further studies should be done on other methods for segmenting the image, as well as other methods of extracting features. In addition, further investigation on how to improve the training would be interesting. One can imagine using support vector machines in defining the regions that the objects belong to, or experimenting with different initializations of the GA. Perhaps positive values could be be sampled from a normal distribution with mean 0, and used as as initialization for the GA. Also it would be interesting to restrict the data manipulation to only work with the most significant bits, and thus reducing the amount of data to investigated. Furthermore, the unused values of the feature vector should be removed to reduce the size of the solution space. Some effort into adjusting the many parameters may also prove beneficial.

The use of textual information as an addition to the proposed system is an interesting topic. It is likely that one could use text as a manner of trivially rejecting the use of the image based algorithms to improve speed, as well as using it as a guide in the decision-making process. A study making use of existing methods for text analysis and classification would be highly relevant.

It would be interesting to apply this method to other datasets. Other types of images could be used by replacing the initial skin filter with a general segmentation. The training would then attempt to find typical objects represented in the training images. Moreover, the dataset could also be textual documents, where the feature vector is a vector of words. In this case, the system could be trained to search for a special document content, and thus Internet sites containing this content could be identified. Here, GA could be used to find the composition of words that identifies the requested document class.

6. Conclusion

The method proposed performs very well under the condition that the web sites investigated contain several images. All the 20 sites tested were correctly classified, and the image-retrieval part of the system was able to properly identify 89% of the images in the evaluation data set composed of 500 offensive and 800 non-offensive images. However, the system may wrongly classify sites with very few images.

Although the performance could be further improved by supplementing with text analysis, the system processed an average of 11 images per second, and is therefore fast enough to be used as a site classifier in a web search engine.

Acknowledgments

We would like to thank John Reidar Mathiassen, and several anonymous reviewers for their useful comments.

References

[1] P. Alshuth, T. Hermes, C. Klauck, J. Kreyss, M. Roper, Iris—image retrieval for images and videos, 1996.

[2] E. Angelopoulou, Understanding the color of human skin, in: Proc. SPIE Conf. on Human Vision and Electronic Imaging VI (SPIE) 2001, SPIE Vol. 4299, SPIE Press, May 2001, pp. 243–251.

[3] S. Baheerathan, F. Albregtsen, H.E. Danielsen, New texture features based on the complexity curve, Pattern Recogn. (1999) 605–618.

[4] L. Bretzner, I. Laptev, T. Lindeberg, Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering, 2002.

[5] S. Brumby, N. Harvey, S. Perkins, R. Porter, J. Szymanski, J. Theiler, J. Bloch, A genetic algorithm for combining new and existing image processing tools for multispectral imagery, 2000.

[6] S. Brumby, J. Theiler, S. Perkins, N. Harvey, J. Szymanski, J. Bloch, M. Mitchell, Investigation of image feature extraction by a genetic algorithm, 1999.

[7] S. Cohen, L. Guibas, Shape-based image retrieval using geometric hashing, 1997.
[8] D. Forsyth, M. Fleck, Identifying nude pictures, 1996.
[9] S. Geman, D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images, IEEE Tans. Part. Anal. Mach. Intell. 6 (1984) 721–741.
[10] R.-L. Hsu, M. Abdel-Mottaleb, A. Jain, Face detection in color images, IEEE Trans. Patt. Anal. Mach. Intell. 24 (2002) 696–706.
[11] A. Jain, A. Vailaya, Image retrieval using color and shape, 1996.
[12] B. Martinkauppi, M. Soriano, M. Laaksonen, Behavior of skin color under varying illumination seen by different cameras in different color spaces, in: M.A. Hunt (Ed.), Proc. SPIE Machine Vision in Industrial Inspection IX, Vol. 4301, January 19–26, San Jose, California, pp. 102–103.
[13] H. Kruppa, M.A. Bauer, B. Schiele, Skin Patch Detection in Real-World Images, DAGM Symposium 2002, Zurich, Switzerland, to appear.
[14] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford Digital Library Technologies Project, 1998.
[15] J.S. Park, D.H. Chang, 2-d invariant descriptors for shape-based image retrieval, Pattern Recognition Letters (2001).
[16] R. Poli, Genetic programming for image analysis, in: J.R. Koza, D.E. Goldberg, D.B. Fogel, R.L. Riolo (Eds.), Genetic Programming 1996: Proceedings of the First Annual Conference, Stanford University, CA, USA, 28–31 1996, MIT Press, pp. 363–368.
[17] J.M. Rehg, K.P. Murphy, P.W. Fieguth, Vision-based speaker detection using Bayesian networks, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Ft. Collins, Colorado, 1999, pp. 110–116.
[18] M. Safar, C. Shahabi, X. Sun, Image retrieval by shape: a comparative study, in: IEEE Internat. Conf. on Multimedia and Expo (I), 2000, pp. 141–144.
[19] M.C. Shin, K.I. Chang, L.V. Tsap, Does colorspace transformation make any difference on skin detection? IEEE Workshop on Applications of Computer Vision, Orlando, FL, December 2002, pp. 275–279.
[20] M. Störring, H.J. Andersen, E. Granum, Skin colour detection under changing lighting conditions, in: H. Araujo, J. Dias (Eds.), 7th Internat. Symp.on Intelligent Robotic Systems, Coimbra, Portugal, 1999, pp. 187–195.
[21] J. Szymanski, S. Brumby, P. Pope, D. Eads, D. Esch-Mosher, M. Galassi, N. Harvey, H. McCulloch, S. Perkins, R. Porter, J. Theiler, A. Young, J. Bloch, N. David, Feature extraction from multiple data sources using genetic programming, 2002.
[22] W.A. Tackett, Genetic programming for feature discovery and image discrimination, in: S. Forrest (Ed.), Proc. 5th Internat. Conf. on Genetic Algorithms, ICGA-93, University of Illinois at Urbana-Champaign, 17–21 1993, Morgan Kaufmann, pp. 303–309.
[23] R. Veltkamp, M. Hagedoorn, State-of-the-art in shape matching, 1999.
[24] J. Yang, W. Lu, A. Waibel, Skin-color modeling and adaptation, in: ACCV, vol. 2, 1998, pp. 687–694.