Technical Report
CVL-TR-5

Contextual Skin Segmentation For Content Filtering

Rehan Ullah Khan

Computer Vision Lab
Institute of Computer Aided Automation
Vienna University of Technology

May 26, 2011

Abstract

Skin color detection is a popular and useful technique because of its wide range of applications both in human computer interaction and content based analysis. Applications such as detecting and tracking of human body parts, face detection and recognition, naked people detection and people retrieval in multimedia databases all benefit from skin detection. In images, skin-color lies within a small region (red, yellow and brown) of the color spectrum, regardless of the ethnicity of the person within an image. One major problem limiting the robustness of color-based skin detection is varying lighting conditions, resulting in the same skin area appearing as two different colors under two different illumination sources. Even under constant illumination, skin-color may vary between individuals. Objects having skin-like colors in our daily life also pose challenges for skin detection algorithms. While the detection of faces, hands, or just skin is one of the easier tasks for humans, this is a difficult and challenging task in computer vision. For face detection, face recognition and hand detection, a robust skin detection will assist further processing steps.

The objective of this study is to arrive at an improved and robust skin detection methods for varying lighting conditions and unconstrained environments. We also focus on the usage of contextual information for skin detection and the evaluation of color spaces and skin color modeling techniques, which will help in the selection of the best combinations for robust skin detection. As a simple and fast method, we introduce two new static filters based on two chrominance components in the IHLS and CIELAB color spaces. Using classifiers and color spaces, we show that cylindrical color spaces outperform other color spaces, the absence of the illuminance component decreases performance,

an appropriate skin color modeling approach selection is important for pixel based skin classification, and that the tree based classifiers (Random forest and J48) are well suited to pixel based skin detection. With the usage of color constancy algorithms, it is found that when using classifiers, skin classification can be improved. As fusion of different color spaces for skin detection, the non-perfect correlation between the color space channels is exploited by learning weights based on an optimization for a particular color space channel using the mathematical financial model of Markowitz. With the graph cut approach, we propose a concept for processing arbitrary images using the universal seed, thereby providing a basis for general skin segmentation, exploiting the spatial relationship among the neighboring skin pixels. We present a systematic approach for skin segmentation with graph cuts by using local skin information, the universal seed based skin segmentation and skin augmentation using an off-line learned model, thus providing a basis for merging spatial and non-spatial data. We propose real-time skin detection using the multiple model approach for videos taking advantage of the contextual information for varying illumination circumstances and a variety of skin colors from person to person. As an application of skin detection, we present the usage of skin detection for flagging videos as potentially objectionable due to sexual content of an adult nature. The skin paths are introduced which provide summarization of a video in the form of a path in a skin-face plot, allowing potentially objectionable segments of videos to be found.

The comprehensive skin detection study presented should, in combination with other cues, enable robust face detection, hand detection and blocking objectionable content in unconstrained environments.

Contents

Chapter 1

Introduction

Color based skin detection is a popular and useful technique because of its wide range of applications both in human computer interaction and content based analysis: Applications such as detecting and tracking of human body parts [3], face detection [16], naked people detection and people retrieval in multimedia databases [17], all benefit from skin detection [51]. Also, color based skin detection gains attention in contributing to blocking objectionable image or video content on the Internet automatically [103]. Besides its usage in computer related technologies, skin color plays an important role for humans and human-human relations: It can serve as an indication about the well-being of a person, one can trace a person's ethnic origin and age from the skin color and it gives an indication whether somebody was exposed to the sun for a longer time [101].

The daily life of individuals is surrounded by computers assisting them in all kinds of activities. Here, one crucial aspect is the interaction between human and computer (user interfaces). Such user interfaces include speech input, face recognition, facial expression interpretation, hand gesture recognition and large scale body language recognition systems [76]. These systems involve several processing steps, initiated by video acquisition. For such systems, the approaches aim to be as robust as possible and being independent of different skin tones, dynamic backgrounds and variable lighting. While the detection of faces, hands, or just skin is one of the easier tasks for humans, this is a difficult and challenging task in computer vision [51]. For face detection, face recognition and hand detection, a robust skin detection will assist further processing steps.

The emphasis of the work presented in this report is on computer vision methods for analysis and detection of skin color in unconstrained environments with and without contextual information. Figure 1.1 presents an overview of the different approaches investigated in this work. The static approaches are those which are either based on static color boundaries or on off-line training. As static approaches for skin detection, thresholding, skin color classification, the universal seed and the financial model of Markowitz are used and important results are derived. The effect of five color constancy algorithms is investigated for thresholding and skin color classification. Contrary to static approaches, the dynamic models take advantage of the local skin information and are introduced to adjust for changing lighting conditions and different skin colors. Based on the dynamic model, a systematic approach is introduced which uses local seeds, the universal seed and classifier probability integration. For real-time requirement of the

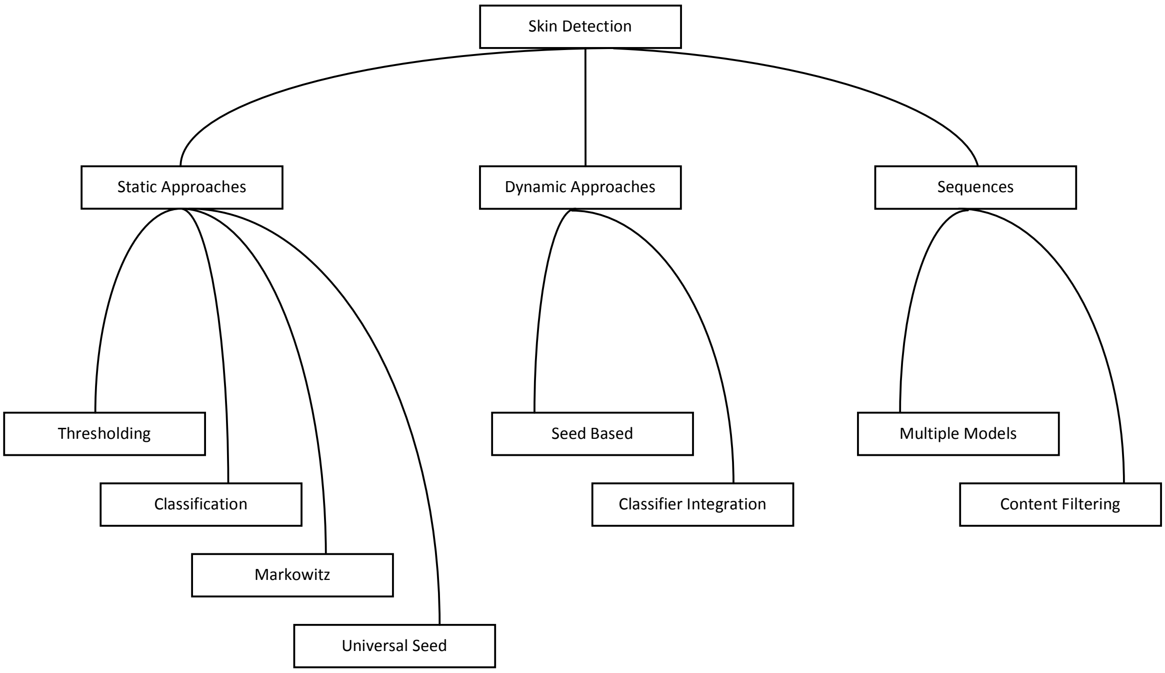


Figure 1.1: Skin detection approaches discussed in this thesis.

applications, skin detection in videos is considered using the multiple model approach. Finally, the usage of skin detection is demonstrated for objectionable content filtering in videos.

1.1 Skin Color Detection

In images, skin color lies within a small region (red, yellow and brown) of the color spectrum regardless of the ethnicity of the person. Skin has little texture; except for extremely hairy subjects, which are rare [28]. Although it covers a small region within the color spectrum, it also includes other, easily identifiable non-skin objects that have skin-like color, for example wood, brown colored walls, hair, sand and other cluttered backgrounds.

Regarding color based skin detection, the advantage of using color over grayscale is due to the extra dimensions of color, i.e., two objects of similar gray tones might be very different in a color space [43]. A color feature is pixel based requiring no spatial context, therefore it is orientation and size invariant and fast to process. One major problem limiting the robustness of color-based skin detection is varying lighting conditions, resulting in the same skin area appearing as two different colors under two different illumination sources. Even under constant illumination, skin color may vary between individuals. Objects having skin-like colors in our daily life also pose challenges for skin detection algorithms. In summary, skin color predominantly depends on the scene context and integration of such information is useful for robust skin detection.

The primary objective of skin detection or classification is building a decision rule that will differentiate between skin and non-skin pixels. The most widely used approach to identifying skin colored pixels involves creating a static skin filter, a volume into which most skin pixels would fall in a given color space [109]. There is a set of techniques which estimate the distribution of skin color by a training phase. These methods are referred to as non-parametric skin models [50]. Finally, other methods include parametric skin distribution models, such as the Gaussian skin color model [116].

Following Kakumanu et al. [51], the major difficulties in skin color detection are caused by various effects:

- *Illumination circumstances*: Any change in the lighting of a scene changes color and intensity of a surface's color and therefore changes the skin color present in the image. This is known as the color constancy problem and is the most challenging one in skin detection.

- *Camera characteristics*: The color distribution of a picture is highly dependent on the sensitivity and the intrinsic parameters of the capturing device.

- *Ethnicity*: The great variety of skin color from person to person and between ethnic groups challenges the classification approaches. There are different techniques available that address this problem.

- *Individual characteristics*: Age, sex and body parts affect the skin color appearance. Detecting context (such as face or other body parts) might overcome some of these problems.

- *Other factors*: Makeup, hairstyle, glasses, sweat, background colors, and motion influence the skin color. Skin detection based on face detection helps to overcome these problems.

An approach for reliably detecting skin has therefore to be stable against noise, artifacts and very flexible against varying lighting conditions. One solution to this problem is using contextual information for skin detection.

1.2 Skin Color and Context

In this thesis, we take advantage of the contextual information using faces in videos. Such contextual information is not only useful for robust skin detection for different skin tones but also for adjusting to the changing lighting conditions. To better understand the importance of context when classifying skin, an on-line poll was created (by Christian Liensberger[1]), where people were asked to rate fragments of images as containing skin or not. Participants were reached by publishing the poll in several web portals where there is a broad distribution of visitors coming from different parts of the world. 403 people participated from six continents, rating 18338 skin/non-skin fragments. The poll consisted

[1] http://www.liensberger.it/

of a set of random frames (skin/non-skin) from the videos in dataset DS1 (Section 1.5.1). To remove scene context, the frames were cut into small fragments (see Figure 1.2). Every participant was presented with a random set of fragments and asked whether each fragment contained skin and the amount of visible skin.

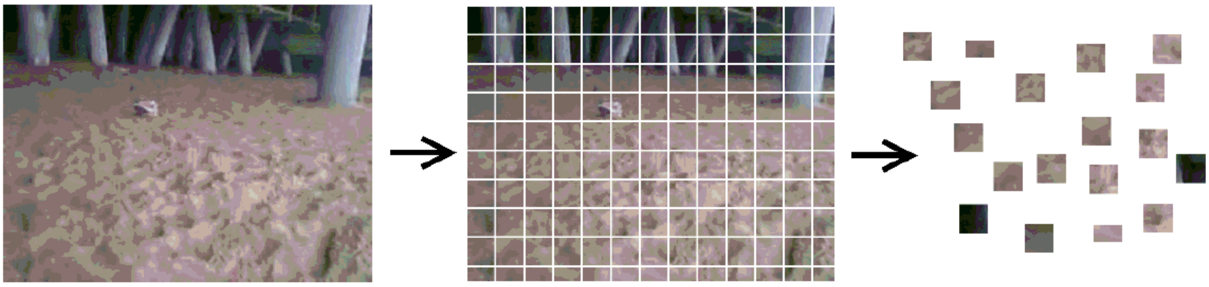


Figure 1.2: Images cut into fragments to remove the context (source: [66]).

From the results, it was concluded that even humans are not able to properly detect skin without scene context. Skin colored materials like sand or wood are likely to be misinterpreted. It was observed that without context, humans take decisions based on color [66]. In some scenarios, with skin-color like materials, humans fail completely. This demonstrates that context is a strong cue for reliable skin detection for humans and should therefore be considered for skin detection algorithms.

1.3 Skin Color and Skin Structure

The apparent skin color is due to light falling on skin and the physical structure of skin. The absorption and reflection of light and as a result the apparent skin color is due to several layers. The outermost section of human skin is the stratum corneum (see Figure 1.3), which is a stratified structure having a thickness of approximately 0.01-0.02 mm [2]. The stratum corneum is composed mainly of dead cells, called corneocytes, embedded in a particular lipid matrix [104]. Light absorption is low in this tissue, with the amount of transmitted light being relatively uniform in the visible region of the light spectrum [4]. The epidermis is a 0.027-0.15mm thick structure composed of four layers (stratum basale, stratum spinosum, stratum granulosum and stratum lucidum) [2]. The epidermis propagates and absorbs light. The absorption property comes mostly from a natural pigment (or chromophore), melanin. There are two types of melanin, the red/yellow phaeomelanin and a brown/black eumelanin [4]. The skin color is mostly associated with eumelanin and the ratio between the concentration of phaeomelanin and eumelanin present in human skin varies from individual to individual, with much overlap between skin types [4].

The dermis is a 0.6-3mm thick structure which also propagates and absorbs light and can be divided into two layers (Figure 1.3): the papillary dermis and the reticular dermis [2]. These layers are primarily composed of dense, irregular connective tissue with nerves and blood vessels. The hypodermis is an adipose tissue characterized by a negligible

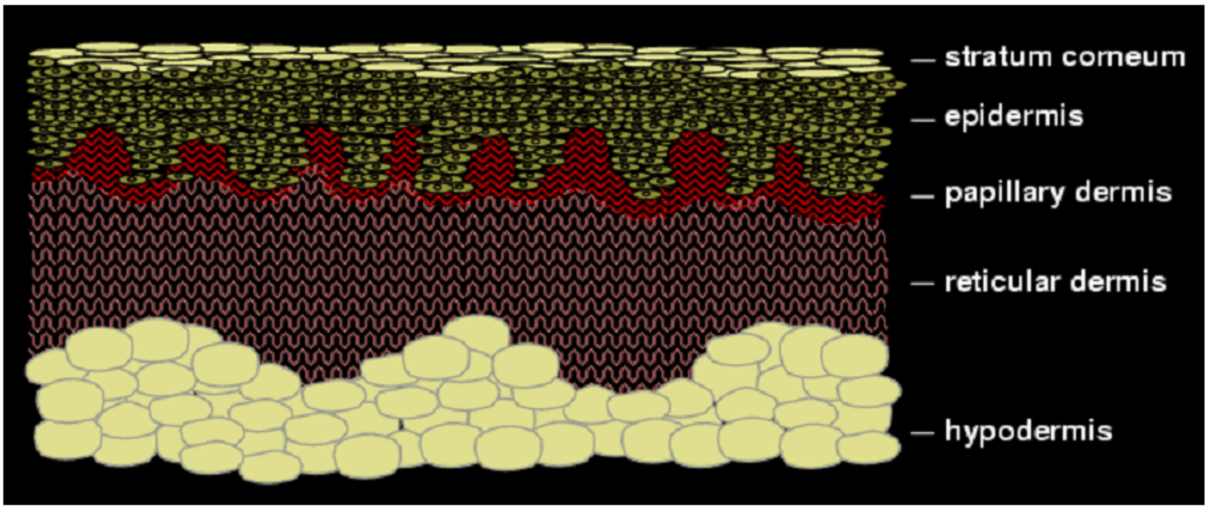


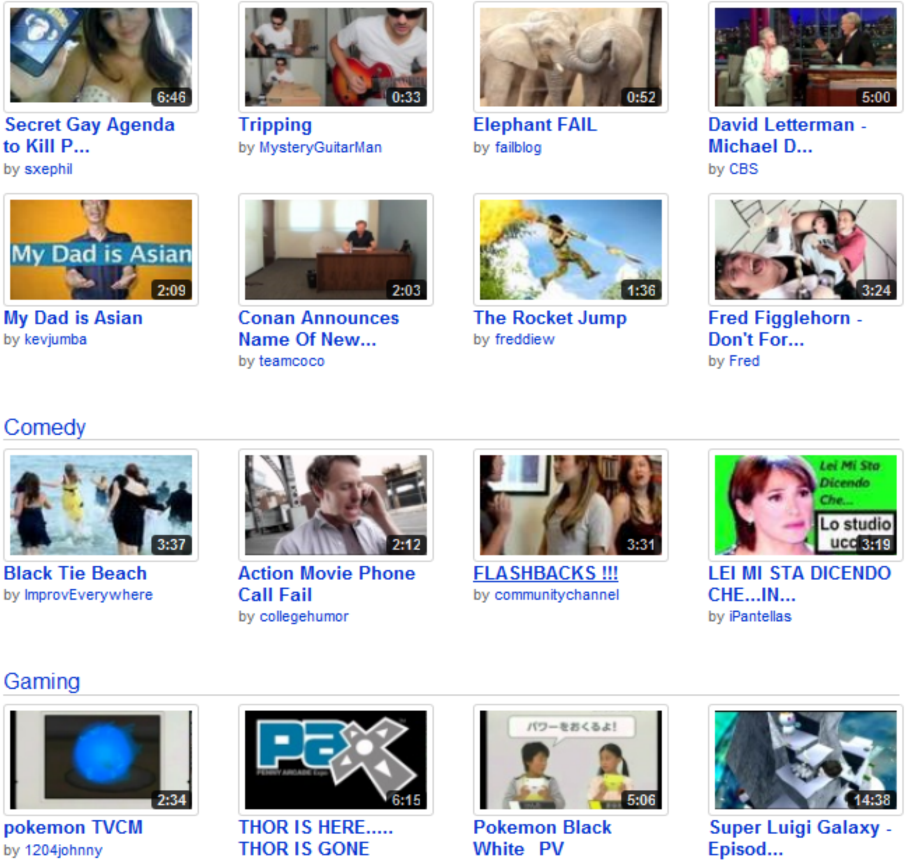
Figure 1.3: Coss-section of skin and the subcutaneous tissue. Soruce [4].

absorption of light in the visible region of the spectrum [12]. The hypodermis presents significant deposits of white fat, whose cells are grouped together forming clusters. Due to the presence of these white fat deposits, most of the visible light that reaches this tissue is reflected back to the upper layers [26].

1.4 Skin Color and Content Filtering

For videos, we demonstrate the usage of skin color detection for filtering of objectionable content. User generated content has become very popular in the last decade and has significantly changed the way we consume media [20]. With the international success of several *Web 2.0* websites (platforms that concentrate on the interaction aspect of the Internet), the amount of publicly available content from private sources is vast and still growing rapidly.

The video content uploaded per day poses a challenge for the operating companies to manually classify every submitted video as appropriate or objectionable. The predominant methods to overcome this problem are to block contents based on keyword matching that categorizes user generated tags or comments [103]. Additionally, connected URLs can be used to check the context of origin to trap these websites [63]. This does not hold true for websites like YouTube that allow uploading of videos. The uploaded videos are not always labeled by (valid) keywords for the content they contain (see for example Figure 1.4). As no reliable automated process exists, the platforms rely on their user community: Users flag videos and the administrators may remove the videos flagged as objectionable. This method is rather slow and does not guarantee that inappropriate videos are immediately withdrawn from circulation. A possible solution for rapid detection of objectionable content is a system that detects such content as soon as it is uploaded. As a completely automated system is not feasible at present, a system that flags potentially objectionable content for subsequent judgment by a human is a good compromise. Such a system has



Secret Gay Agenda to Kill P...
by sxephil

Tripping
by MysteryGuitarMan

Elephant FAIL
by failblog

David Letterman - Michael D...
by CBS

My Dad is Asian
by kevjumba

Conan Announces Name Of New...
by teamcoco

The Rocket Jump
by freddiew

Fred Figglehorn - Don't For...
by Fred

Comedy

Black Tie Beach
by ImprovEverywhere

Action Movie Phone Call Fail
by collegehumor

FLASHBACKS !!!
by communitychannel

LEI MI STA DICENDO CHE...IN...
by iPantellas

Gaming

pokemon TVCM
by 1204johnny

THOR IS HERE..... THOR IS GONE

Pokemon Black White PV

Super Luigi Galaxy - Episod...

Figure 1.4: Most popular videos from youtube.com on September 02, 2010. The uploaded videos are not always labeled by (valid) keywords for the content they contain.

two important parameters: the number of harmless videos flagged as potentially objectionable (false positive rate), and the number of objectionable videos not flagged (false negative rate). In the context of precision and recall of a classification application, these two parameters present a trade-off. For a very low false negative rate, a larger amount of human effort will be needed to examine the larger number of false positives. These parameters should be adjustable by the end-users depending on the local laws (some regions have stricter restrictions on objectionable content) and the amount of human effort available. A further enhancement to reduce the amount of time required by the human judges is to flag only the segments of videos containing the potentially objectionable material, removing the need to watch the whole video, or search the video manually.

One reason why videos may be considered objectionable is due to explicit sexual content. Such videos are often characterized by a large amount of skin being visible in the frame, so a commonly used component for their detection is a skin detector [63, 122]. However, this characteristic is also satisfied by frames not considered as objectionable, most importantly close-ups of faces e.g. in interviews. We show that contextual based skin detection is useful in such scenarios.

1.5 Experiments

Experimental evaluation is presented for the approaches developed in the course of this thesis. The experiments are not centralized and are given with the corresponding approach. Due to number of algorithms covered, the evaluation/results are presented immediately after the presentation of each algorithm instead of in a single chapter. In the following, the datasets and the evaluation measures used are therefore explained.

1.5.1 Datasets

For the evaluation of skin detection approaches developed, we use two datasets. The DataSet (DS1) was created mainly by Christian Liensberger, spanning 25 YouTube videos, chosen by an Internet service provider. The second DataSet (DS2) is provided by Sigal et al. [94].

DS1

15 videos have been provided by an Internet service provider that requires a skin detection application for their on-line platform. Their aim was to choose challenging videos with near skin-color backgrounds. Pink and brown backgrounds such as beaches, sand, cork boards or similar are included to provide variation and adding an extra challenge (see Figure 1.5). We added 10 videos to encounter additional challenges such as a larger variety of skin-colors, especially different skin-colors in one frame. Most of the sequences also contain scenes with multiple people and/or multiple visible body parts and scene shots both indoors and outdoors, with steady or moving camera. The lighting varies from natural light to directional stage lighting. Sequences contain shadows and minor occlusions. Collected sequences vary in length from 100 frames to 500 frames. They also contain data errors and are generally of poor quality, varying size and frame rate. For all of the videos ground truth has been generated.

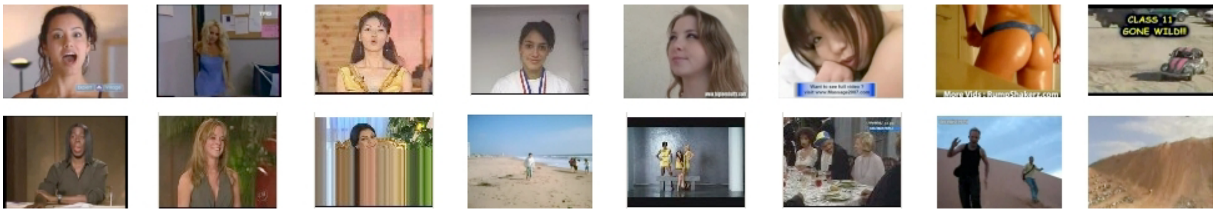


Figure 1.5: Example frames from the annotated video dataset DS1.

The authors [66] of the dataset used Adobe Flash[2] for the ground truth generation because it allowed to output a binary ground truth video with a per pixel precision, which was easier to process than using the XML that Viper GT[3] produced. Examples of the annotated ground truth can be seen in Figure 1.6. White pixels indicate annotated

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

[3] http://viper-toolkit.sourceforge.net/

skin. Non-skin facial features like eyes, eyebrows or similar are left out if they are visible. Because of the poor quality of the videos this was not always possible. For skin detection experiments, we have used 8,991 images from these video sequences that are represented in the following three sets.

Skin-only: This is a set of (5,817) images, in which every image contains some skin.

Non-skin: This is a set of (3,174) images, in which every image is without skin.

Hybrid: This is the full set of (8,991) images, containing both skin-only and non-skin images.

In experiments, unless specifically specified, when we use DS1, we mean the DS1 hybrid set.

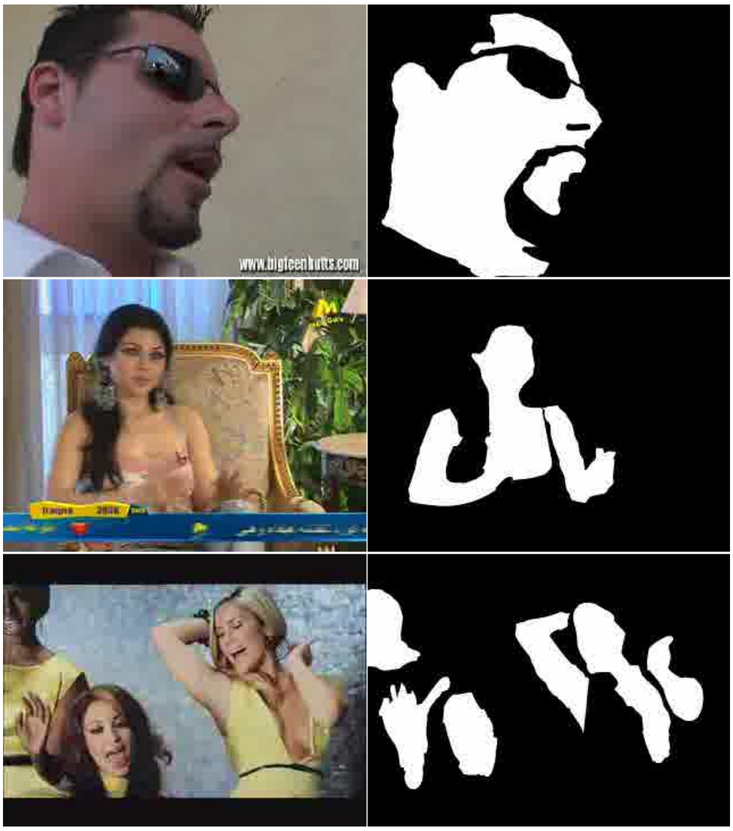


Figure 1.6: Example frames and their per pixel basis annotation for the dataset DS1.

DS2

This dataset was used in [94] and contains a set of 21 high quality video sequences from nine popular movies. The sequences span a wide range of environmental conditions. People of different ethnicity and various skin tones are represented. Some scenes contain multiple people and/or multiple visible body parts. Collected sequences contain scenes shot both indoors and outdoors, with static and moving camera. The lighting varies from natural light to directional stage lighting. Some sequences contain shadows and minor occlusions. Collected sequences vary in length from 50 to 350 frames. Figure 1.7 shows example frames from the collected sequences.

The sequences are hand-labeled by the author of the dataset to provide ground truth data for evaluation. Every fifth frame of the sequences is labeled. For each labeled frame, the human operator created one binary image mask for skin regions and one for non-skin regions (background). Boundaries between skin regions and background, as well as regions that had no clearly distinguishable membership in either class are not included in the masks and are considered "don't care" regions. Figure 1.8 shows one example frame and its ground-truth labeling. There are 720 images in this dataset and every image contains some skin, therefore, we do not divide it into further sets.

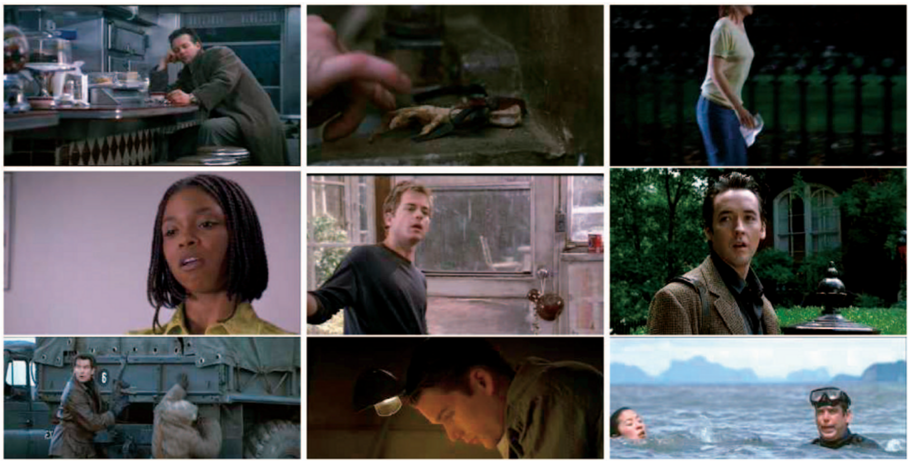
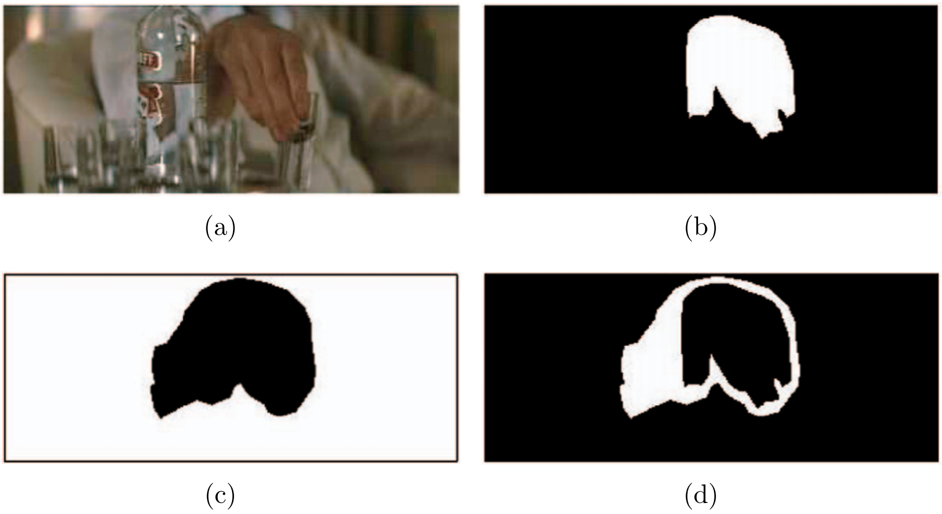


Figure 1.7: Example frames from dataset (DS2) consisting of 21 video sequences. (Source: [94]).



(a)

(b)

(c)

(d)

Figure 1.8: Example of a labeled ground truth frame for the dataset (DS2). (a) Original, mask images for (b) skin, (c) background and (d) don't care regions. (Source: [94]).

1.5.2 Evaluation Measures

Performance evaluation for the approaches developed is based on F-measure and or specificity. The F-measure is calculated by evenly weighting precision and recall:

$$\mathrm{F-measure} = 2\left(\frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}\right) \tag{1.1}$$

where,

$$\mathrm{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}, \quad \mathrm{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Specificity is defined as the true negative rate:

$$\mathrm{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}} \tag{1.2}$$

For images (DS1 skin-only, DS1 hybrid and DS2), the evaluation is based on F-measure and Specificity, as all the images contain some skin. For images without skin (DS1 non-skin), the F-measure is not defined (as the precision is zero or undefined and recall is undefined because every detection is a false positive) and therefore, we use specificity as the only evaluation measure for this set.

10-fold Cross Validation: For pixel based skin color classification (using classifiers) in Chapter 3, we use 10-fold cross validation on datasets DS1 and DS2. In 10-fold cross validation, out of ten partitions of the data, nine are used for training the model and one is used for testing and the process is repeated ten times. For classifiers (and color spaces) comparison, this strategy is adopted because all observations are used for both training and validation and each observation is used for validation exactly once giving a valid performance comparisons for the classifiers.

Training/Test Sets: In an evaluation based on training and test sets, we use separate smaller training sets and evaluate the developed approaches on every image of the test set. With the exception of classifiers (also evaluated on the basis of 10-fold cross validation), all the developed approaches in this thesis will be based on training/test set evaluation.

Per Image: For per image evaluation, we calculate F-measure and specificity for each image of the dataset and report the mean.

Per Pixel: For per pixel evaluation, we calculate true positives, false positives, true negatives and false negatives for all pixels in the whole dataset and then apply F-measure/specificity formulae.

1.6 Contributions

- As a simple and fast method, we introduce two new static filters based on two chrominance components in IHLS and CIELAB color spaces.

- We investigate and evaluate (1) the effect of color space transformation on skin detection performance and finding the appropriate color space for skin detection,

(2) the role of the illuminance component of a color space, and finally (3) the appropriate pixel based skin color modeling technique (classifier based). We present a comprehensive evaluation on color space and skin color modeling techniques which helps in the selection of the best combinations for skin detection. We show that (a) the cylindrical color spaces outperform other color spaces, (b) the absence of the illuminance component decreases performance, (c) an appropriate skin color modeling approach selection is important for pixel based skin classification and that the tree based classifiers (Random Forest and J48) are well suited to pixel based skin detection.
Publications:
"Color Based Skin Classification". Submitted to Pattern Recognition Letters.
"Skin Detection: A Random Forest Approach". Presented at International Conference on Image Processing (ICIP 2010). [56]

- We investigate the use of color constancy algorithms for skin detection and find that when using classifiers, skin classification can be improved with the introduction of lighting correction.

- We linearly merge a large number of color space channels from various color spaces, representing it as a fusion process for skin detection. The aim of fusing different color space channels is to achieve invariance against varying imaging and illumination conditions. The non-perfect correlation between the color spaces is exploited by learning weights based on an optimization for a particular color space channel using the mathematical financial model of Markowitz. The weight learning process develops a color weighted model using positive training data only.
Publication:
"Color Space Channel Fusion for Skin Detection". Submitted to: Journal of Image and Vision Computing.

- We present the idea of a highly adaptive universal seed, thereby exploiting the positive training data only. We model the skin segmentation as a minimum cut problem on a graph defined by the image color characteristics. The prior graph cuts based approaches for skin segmentation do not provide general skin detection when the information of foreground or background seeds is not available. We propose a concept for processing arbitrary images; using a universal seed to overcome the potential lack of successful seed detections thereby providing basis for general skin segmentation exploiting the spatial relationship among the neighboring skin pixels, thereby providing more accurate and stable skin blobs.
Publication:
"Universal Seed Skin Segmentation". Presented at International Symposium on Visual Computing (ISVC 2010). [58]

- We present a systematic approach for skin segmentation with graph cuts by using local skin information, universal skin information and skin augmentation using the off-line learned models. The skin segmentation process starts by exploiting the local skin information of detected faces. The detected faces are used as foreground seeds

for calculating the foreground weights of the graph. If local skin information is not available, we opt for the universal seed. To increase the robustness we learn external off-line models. The learned models are used to augment the universal seed for skin segmentation when no local information is available from the image.
Publications:
"Augmentation of Skin Segmentation". Presented at International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV 2010). [57]
"Weighted Skin Color Segmentation and Detection Using Graph Cuts". Presented at Computer Vision Winter Workshop (CVWW 2010). [55]

- We propose a straightforward skin detection method for videos where real-time processing is the main concern. To overcome varying illumination circumstances and a variety of skin colors, we introduce a multiple model approach which can be carried out independently per model. The color models are initiated by skin detection based on face detection and adapted in real time.
  Publication:
  "An Adaptive Multiple Model Approach for Fast Content-Based Skin Detection in On-Line Videos". Presented at ACM Multimedia 2008 workshop. [59]

- As an application of skin detection, we consider the flagging of uploaded videos as potentially objectionable due to sexual content of an adult nature. We introduce to this task two uses of contextual information in the form of detected faces. The first is to use a combination of different face detectors to adjust the parameters of the skin detection model. The second is through the summarization of a video in the form of a path in a skin-face plot. This plot allows potentially objectionable segments of videos to be found, while ignoring segments containing close-ups of faces. The proposed approach runs in real-time.
  Publication:
  "Skin Paths for Contextual Flagging Adult Videos". Presented at International Symposium on Visual Computing (ISVC 2009). [103]

1.7 Outline and Contents of Thesis

This thesis is organized in seven chapters. In Chapter 2, the theoretical background of color spaces, classifiers and color constancy algorithms used in this thesis are introduced and the state-of-the-art in skin detection is reviewed. Chapter 3 discusses static skin filters, color based skin classification and the effect of color constancy algorithms on these approaches. In Chapter 4, feasibility of the financial mathematical model of Markowitz for color based skin detection is demonstrated. Chapter 5 introduces the universal seed and discusses the usage of contextual information (dynamic models) and a systematic approach for skin detection, integrating classifier probabilities for robust skin detection. Chapter 6 discusses skin detection in videos in real-time using the multiple model approach and illustrates the usage of skin detection for blocking objectionable content. The last chapter draws conclusions and suggests further directions.
The remainder of this section summarises and outlines the following 6 chapters.

State Of The Art

Chapter 2 introduces the background of this thesis and reviews related work that is concerned with the approaches used in general and the skin detection in particular. It consists of five sections: (1) An overview of the classifiers used as skin color modeling techniques. These classifiers are used in the derivation of important results in Chapter 3 for skin color modeling and detection. Moreover, we present the idea of merging classifier weights with the spatial structure of graph cuts for robust skin detection in Chapter 5. (2) A brief introduction to the color spaces used for different approaches in this thesis. (3) An overview of five color constancy algorithms used in Chapter 3, namely Gray-Edge, Gray-World, max-RGB, Shades of Gray, Gray-Edge and Bayesian color constancy. (4) A review of skin detection methods for computer vision, focusing on the usage of color spaces as well as (5) The skin color modeling techniques.

Static Skin Segmentation and Skin Classification

Chapter 3 is concerned with static approaches and color based skin classification. Two new static skin filters for IHLS and CIELAB color spaces are introduced and compared to the other state-of-the-art static filters. The effect of five color constancy algorithms is studied for the static skin filter in the YCbCr color space. The approaches used based on off-line training (classifiers) are evaluated for skin segmentation with different color spaces. Moreover, their effect on performance in the presence of 3D color spaces and 2D color spaces is studied and important results are derived. The effect of color constancy algorithms is reported for the Random forest classifier.

Markowitz Model For Skin Detection

In Chapter 4, the mathematical financial model of Markowitz [68] is introduced and its feasibility for skin detection is demonstrated. As a usage of the Markowitz model for training based on positive data only, weights for different color space channels are learned and linearly merged, representing it as a fusion process for skin detection. With the Markowitz model, the non-perfect correlation between different color space channels is exploited based on an optimization for a particular color space channel.

Seed Based Approaches

In Chapter 5, a novel idea of off-line training based on the universal seed is introduced, which is based on training on positive data only. A local seed based approach is illustrated as well as how it can improve skin segmentation. Size reduction and static filters used with the seed improve skin detection. It is shown how a seed can be propagated from one scene to other scenes for contextual skin segmentation. An external weighted model, learned off-line, can be used to augment the universal seed weights, in the case when no local seeds are available. A comprehensive skin segmentation approach using the local skin information, global skin information and classifier (J48) is presented. The non-spatial weights from the classifier are augmented with the spatial structure of the graph cut approach.

Video Sequences and Content Filtering

In Chapter 6, we port skin detection in still images to videos where real-time performance is a principal concern. A multiple model approach using the YCbCr color space is presented. This approach not only takes advantage of the contextual information in terms of faces but also uses multiple models, while at the same time satisfying the real-time demands. The last section considers the flagging of uploaded videos as potentially objectionable due to content of an adult nature. Such videos are characterized by the presence of a large amount of skin, although other scenes, such as close-ups of faces, also satisfy this criterion. Two uses of contextual information are introduced. The first is to use a combination of different face detectors to adjust the parameters of the skin detection model. The second is through the summarization of a video in the form of a path in a skin-face plot. This plot allows potentially objectionable segments of videos to be found, while ignoring segments containing close-ups of faces.

Conclusions
Chapter 7 summarizes the main contributions of this thesis and further research directions are suggested.

Chapter 2

State Of The Art

This chapter introduces the background of this thesis and reviews related work that is concerned with the approaches for skin detection. Since classifiers are used for the derivation of important results and for performance comparison with the developed approaches, an introduction to classifiers as skin color modeling techniques is given in Section 2.1. An introduction to color spaces used is given in Section 2.2. An overview of five color constancy algorithms used, namely Gray-Edge, Gray-World, max-RGB, Shades of Gray, Gray-Edge and Bayesian color constancy is given in Section 2.3. The state-of-the-art in color based skin modeling and detection for computer vision is reviewed in Section 2.4, focusing on color spaces used for this task. Section 2.5 reviews skin detection methods with emphasis on skin color modeling techniques.

2.1 Classifiers and Skin Classification

We have used 8 classifiers for pixel based skin classification namely Naive Bayes, Bayesian Network, J48, Random Forest, Multilayer Perceptron, RBF Network, SVM and multi class AdaBoost (Adaboost.M1). In this section, we present the basic theory behind each of these classifiers. They are used in Chapter 3 for color based skin classification.

2.1.1 Naive Bayes

The Naive Bayes classifier is a specification of Bayes inference with a naive assumption of independence [64]. It is a simple probabilistic classifier providing maximum a posteriori probability for each testing instance. In text categorization, the Naive Bayes classifier is used for document classification. The test document is given the class label with maximum a posteriori probability. The naive assumption in Bayes statistics is simply that the contribution of a word in all documents of one class is statistically independent or unrelated to all other words in the documents of the same class. Given a set of variables, $X = \{x_1, x_2, ..., x_d\}$, the posterior probability for the event c_j among a set of possible outcomes $C = c_1, c_2...c_d$ using Bayes' rule is:

$$p(c_j|x_1, x_2, ..., x_d) \alpha p(x_1, x_2, ..., x_d|c_j)p(c_j) \tag{2.1}$$

where $p(c_j|x_1, x_2, ..., x_d)$ is the posterior probability of class membership, i.e., the probability that X belongs to c_j. Since Naive Bayes assumes that the conditional probabilities of the independent variables are statistically independent, we can decompose the likelihood to a product of terms:

$$p(c_j|X) \alpha p(c_j) \prod_{k=1}^{d} p(x_k|c_j) \tag{2.2}$$

Finally, using Bayes' rule above, we label a new case X with a class level c_j that achieves the highest posterior probability.

2.1.2 Bayesian Network

A Bayesian network is also called a belief network and a directed acyclic graphical model. It is a representation for random variables and conditional independences within these random variables. The conditional independences are represented by Directed Acyclic Graph (DAG). More formally, a Bayesian network $B =< N, A, \theta >$ is a DAG $< N, A >$ with a conditional probability distribution for every node (collectively θ for all nodes). A node $n \epsilon N$ in the graph G represents some random variable, and each edge or each arc $a \epsilon A$ between nodes shows a probabilistic dependency. For learning Bayesian networks from specific datasets, data attributes are represented by nodes [23].

In a Bayesian network, the learner does not distinguish the class variable from the attribute variables in data (unsupervised learning) [31]. As such, a network (or a set of networks) are created that "best describes" the probability distribution of the training data. The problem of learning a Bayesian network can be stated as: Given a training set $D = \{u_1, ..., u_N\}$ of instances of U, find a network B that best matches D. Heuristic search techniques are used to find the best candidate in the space of possible networks. The search process relies on a scoring function that assesses the merits of each candidate network [31]. If we assume that for training, a Bayesian network B encodes a distribution $P_B(A_1, ..., A_n)$ from the training dataset with C classes, then for testing, a classifier based on B returns the label c that maximizes the posterior probability $P_B(c|a_1, ..., a_n)$. The network B can also be used to find out updated knowledge of the state of a subset of variables when other variables (the evidence variables) are observed. This process of computing the posterior distribution of variables given evidence is called probabilistic inference [23].

2.1.3 Decision Tree (J48)

J48 is an open source Java implementation of the Quinlan's [82] C4.5 decision tree algorithm. Decision tree algorithms begin with a set of cases, or examples, and create a tree data structure that can be used to classify new cases. Each case is described by a set of attributes (or features) which can have numeric or symbolic values. Associated with each training case is a class label. Each internal node of a decision tree contains a test, the result of which is used to decide what branch to follow from that node.

C4.5 creates decision trees from labeled training data using the concept of information entropy [82]. It splits the data into smaller subsets based on the fact that each attribute of the data can be used to make a decision. When an attribute is chosen for splitting the data, C4.5 examines the normalized information gain that results from the splits produced by the chosen attribute. For decision making, C4.5 selects an attribute resulting in the highest normalized information gain. C4.5 recurses on the smaller subsets and splitting stops if all instances in a subset belong to the same class. A leaf node is created in the decision tree for identifying the class.

Compared to other decision tree algorithms, C4.5 handle both continuous, discrete attributes and attributes with differing costs. It uses pruning after trees creation. For pruning, C4.5 removes branches that do not contribute in classification by replacing them with leaf nodes.

2.1.4 Random Forest

The popularity of tree classifiers is their intuitive appeal and easy training procedures. However, there is no classical decision tree approach to increase both classification and generalization accuracy. For this purpose the random forest was introduced by Tin Ho [44]: The random forest is an ensemble of tree predictors such that each tree depends on the values of a random vector. To classify a new object from an input vector, the input vector is presented to each of the trees in the forest. Each tree gives a classification, and we say the tree “votes” for that class. The forest chooses the classification having the most votes.

For growing trees, if the number of cases in the training set is N, sample N cases at random, but with replacement, from the original data. This sample will be the training set for growing the tree. If there are M input variables, a number $m << M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing. Each tree is grown to the largest extent possible. There is no pruning. With the increase in tree count the generalization error converges to a limit. In practice as few as 10 trees present competitive results [11].

2.1.5 Multilayer Perceptron

A multilayer perceptron is a feedforward artificial neural network model which maps sets of input data onto a set of appropriate output. It is different from the standard linear perceptron in that it uses two or more layers of neurons with nonlinear activation functions. Compared to the standard linear perceptron, it can distinguish data that is not linearly separable, or separable by a hyperplane [6]. What makes a multilayer perceptron different is that each neuron uses a nonlinear activation function which was developed to model the frequency of action potentials, or firing, of biological neurons in the brain [6]. In multilayer perceptron each node in one layer connects with a certain weight to every node in the following layer. Learning occurs in the perceptron by changing connection weights (through backpropagation [85]) after each training cycle, based on the amount of error in the output compared to the expected result.

2.1.6 RBF Network

A Radial Basis Function (RBF) network is a neural network that uses radial basis functions as activation functions and is a linear combination of radial basis functions [117]. An RBF network generally consists of three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer. The output, $\varphi : \mathbb{R}^n \to \mathbb{R}$ is given by [18]:

$$\varphi(X) = \sum_{i=1}^{N} \alpha_i \rho(||\mathrm{X} - c_i||) \tag{2.3}$$

where N is the number of neurons in the hidden layer, c_i is the center vector for neuron i, and α_i are the weights of the linear output neurons. In the basic form all inputs are connected to each hidden neuron. RBF networks are universal approximators on a compact subset of $\mathbb{R}^n$. This means that an RBF network with enough hidden neurons can approximate any continuous function with arbitrary precision [18].

2.1.7 SVM

A Support Vector Machine (SVM) finds the optimal hyperplane for inter-class separation. SVM constructs hyperplanes in a high dimensional space and a good inter-class separation is achieved by the hyperplane (optimal hyperplane) that has the largest distance to the nearest training data points of any class (called functional margin) [25, 108]. For training data, SVM finds the separating hyperplane in such a way that the data points with similar labels fall on the same side. For training data (X, Y) where $x_i \in \mathbb{R}$ and $y_i \in \{+1, -1\}$ are the corresponding labels with $1 \leq i \leq N$, the classifier finds the hyperplane parameters w and b such that:

$$y_i \left(w x_i + b \right) > 0, \quad i = 1, ..., N \tag{2.4}$$

The data is linearly separable if a hyperplane satisfying Equation 2.4 exists. If the data set is not linearly separable, then the SVM introduces a parameter C to control the number of misclassified points. The parameter C penalizes sample misclassification in proportion to the classification boundary distance [25]. The user has to chose a value of C to fix the penalty for misclassification. Also, for data that is not linearly separable, the SVM uses mapping of the data to the higher dimensional space.

2.1.8 Adaboost.M1

Adaptive Boosting is a meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance [30]. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers but less susceptible to the overfitting problem than most learning algorithms.

Given a set of labeled training examples $(x_1, y_1), ..., (x_m, y_m)$ where x_i is contained in some domain or instance space X, and y_i is from the label set Y. Y can be a binary label set or a multi label set. AdaBoost calls a weak classifier (based on weak hypothesis) repeatedly in a series of rounds $t = 1...T$ and a distribution of weights is updated that indicates the importance of examples in the dataset for the classification. On each round, the weights of each incorrectly classified example are increased, so that the new classifier focuses more on those examples.

The AdaBoost.M1 is simply the multi-class AdaBoost. The multi-class case requires the accuracy of the weak hypothesis greater than $\frac{1}{2}$. This condition in the multi-class is stronger than that in the binary classification cases.

2.2 Color Spaces

In this section, we describe color spaces that are used in experiments during the course of the thesis. One usage of color spaces is exhibited in classifiers and color space comparison (Chapter 3). Also, using the Markowitz model (Chapter 4), different color space channels are weighted for their role in skin detection. An important property of some color spaces is the perceptual uniformity. Perceptual uniformity means that a small perturbation to a component value is approximately equally perceptible across the range of that value [109].

RGB is a color space originated from CRT (or similar) display applications, when it was convenient to describe color as a combination of three colored rays (red, green and blue) [109]. There exists a very high correlation between the three channels, namely R, G and B. This color space suffers from significant perceptual non-uniformity and the mixing of chrominance and luminance data [109]. Therefore, these points make RGB not a very favorable choice for skin detection, color analysis and color based recognition algorithms.

Normalized RGB is a linear representation that is easily obtained from the RGB values by a simple normalization procedure:

$$nr = \frac{R}{R+G+B} \tag{2.5}$$
$$ng = \frac{G}{R+G+B}$$
$$nb = \frac{B}{R+G+B}$$

As can be seen, the sum of the three normalized components is 1 ($nr + ng + nb = 1$), and thus the third component can be omitted. The remaining components are called pure colors, for the dependence of nr and ng on the brightness of the source RGB color is diminished by the normalization [109]. A remarkable property of this representation is that for matte surfaces, while ignoring ambient light, normalized RGB is invariant (under certain assumptions) to changes of surface orientation relatively to the light source [109]. Thus one usage of normalization can be to compare two images taken under variations of illumination provided that the color temperature remains the same as the illumination output varies.

Cylindrical Color Spaces (HSI, HSV, IHLS) HSI describes color with intuitive values, based on the artist's idea of tint, saturation and tone. Hue defines the dominant color (such as red, green, purple and yellow) of an area, saturation measures the colorfulness of an area in proportion to its brightness. The intensity, lightness or value is related to the color luminance. The intuitiveness of the color space components and explicit discrimination between luminance and chrominance properties made it popular for skin detection. An interesting property is that Hue is invariant to highlights at white light sources, and also, for matte surfaces, to ambient light and surface orientation relative to the light source [109]. The color channels of HSI are represented as H (represented in degrees) for hue, S for saturation and I for intensity. H, S and I are defined as:

$$H = \arctan(\frac{\beta}{\alpha}) \tag{2.6}$$

$$S = \sqrt{\alpha^2 + \beta^2} \tag{2.7}$$

$$I = (R + G + B)/3 \tag{2.8}$$

where $\alpha = R - \frac{1}{2}(G + B)$ and $\beta = \frac{\sqrt{3}}{2}(G - B)$.

HSV (Hue, Saturation, Value) is obtained by nonlinear transformation of RGB and can be referred to as being a perceptually uniform color space due to its similarity to the human perception of color. Hue describes pure color, saturation gives a measure of the degree to which a pure color is diluted by white Light and value represents brightness along the grey axis (e.g. white to black) [54]. The hue color channel of HSV color is represented in degrees. The saturation S and V are defined as:

$$S = \frac{max(R, G, B) - min(R, G, B)}{max(R, G, B)} \tag{2.9}$$

$$V = max(R, G, B) \tag{2.10}$$

The Improved Hue, Luminance and Saturation (IHLS) color space is introduced in [42]. It is obtained by first placing an achromatic axis through all the grey ($R = G = B$) points in the RGB color cube. Then the coordinates of each point are specified in terms of

- Position on the achromatic axis (brightness)
- Distance from the axis (saturation)
- Angle with respect to pure red (hue)

The IHLS model is improved with respect to the similar color spaces (HLS, HSV, etc.) by removing the normalization of the saturation by the brightness. This has the following advantages:

- The saturation of achromatic pixels is always low
- The saturation is independent of the brightness function used.

One may therefore, choose any function of R, G and B to calculate the brightness. The color channels of IHLS are represented as iH (represented in degrees) for hue, iS for saturation and iY for intensity. iH is defined as the trignometric angle. iS and iY are defined as:

$$iS = max(R, G, B) - min(R, G, B) \tag{2.11}$$

$$iY = 0.2125R + 0.7154G + 0.0721B \tag{2.12}$$

CIELAB was created to serve as a device independent model. It describes visible color spectrum of the human eye and is thus a perceptually uniform color space [87]. CIELAB is based on Opponent-Colors theory, which assumes that eye perceives color as the pairs of opposites which are

- Light-dark
- Red-green
- Yellow-blue

The level of light or dark is indicated by the L value, redness or greenness by the a value and the b value indicates yellowness or blueness. CIELAB is an chromatic value color space due to the fact that the red/green and yellow/blue opponent channels are computed as differences of lightness transformations of cone responses [87].

YCbCr is an encoded non-linear RGB signal, commonly used by European television studios and for image compression work, such as JPEG (Joint Photographic Experts Group) and MPEG (Moving Picture Experts Group). Color is represented by luma and two color difference values. Luma is computed from non-linear RGB constructed as a weighted sum of the RGB values, and the two color difference values Cr and Cb are formed by subtracting luma from RGB red and blue components. For 24 bit color depth, the following values apply:

$$\begin{gathered} Y = (0.299 * (R - G)) + G + (0.114 * (B - G)) \\ Cb = (0.564 * (B - Y)) + 128 \\ Cr = (0.713 * (R - Y)) + 128 \end{gathered} \tag{2.13}$$

The favorable property of this color space for skin color detection is the stable separation of luminance, chrominance, and its fast conversion from RGB.

The **Opponent Color Space** aims to mimic the color processing in the human retina [7]. The opponent color channels red-green RG and yellow-blue YB are defined as:

$$RG = R - G \tag{2.14}$$

$$YB = (2B - R + G)/4 \tag{2.15}$$

The color models discussed are selected for experiments in this thesis, because they are commonly used in color image precessing. They contain both variant and invariant properties with reference to imaging conditions. RGB, CIE L and SV are sensitive to shadows, shading, illumination and highlights and the nr, ng and CIE ab are invariant to shadows, shading and illumination intensity [99, 98]. The opponent color components RG and YB are invariant to highlights, assuming a white light source [99]. The transformation simplicity and explicit separation of luminance and chrominance components makes YCbCr attractive for skin color modeling [109]. The unnormalized saturation of the iHLS color space gives a better distribution in the color space.

2.3 Color Constancy

The effect of lighting correction for skin detection is studied for static skin filters and classifiers in Chapter 3. In this section, we present an overview of the color constancy algorithms used.

Color constancy is the ability of the human vision system to resolve object colors in a scene independently of the illuminant. In other words, the color constancy problem can be defined as the ability to estimate the unknown light of a scene from an image/photograph. More formally, assume that an image f is composed of [107]:

$$f(x) = \int_w e(\lambda)c(\lambda)s(x,\lambda)d\lambda \tag{2.16}$$

where $e(\lambda)$ is the color spectrum of the light source, $s(x,\lambda)$ the reflectance of the surface and $c(\lambda)$ is the camera sensitivity function. Further, w and x are the visible spectrum and the spatial coordinates respectively. The goal of color constancy is to estimate the light source color $e(\lambda)$:

$$e = \int_w e(\lambda)c(\lambda)d\lambda \tag{2.17}$$

Different color constancy algorithms provide different estimation of e. In the following, we discuss Gray-edge hypothesis, Gray-world hypothesis, max-RGB, Shades-of-gray and Bayesian color constancy.

2.3.1 The Gray-edge Hypothesis

Van de Weijer et al. [107] propose the Gray-Edge hypothesis, which can be stated as "the average of the reflectance differences in a scene is achromatic":

$$\frac{\int |s_x^\sigma(x,\lambda)|dx}{\int dx} = g(\lambda) = k \tag{2.18}$$

where s is the reflectance and the subscript x indicates the spatial derivative at scale σ. With the Gray-Edge assumption, the light source color can be computed from the average color derivative in the image given by:

$$\frac{\int |f_x(x)|dx}{\int dx} = \int_w e(\lambda)\left(\frac{\int |s_x(x,\lambda)|dx}{\int dx}\right)c(\lambda)d\lambda = k\int_w e(\lambda)c(\lambda)d\lambda = ke \tag{2.19}$$

where $|f_x(x)| = (|R_x(x)|, |G_x(x)|, |B_x(x)|)^T$. Equation 2.19 starts with color image derivative for e estimation. The Gray-Edge hypothesis originates from the observation that the color derivative distribution of images forms a relatively regular, ellipsoid like shape, of which the long axis coincides with the light source color [107]. The Gray-Edge hypothesis can be adapted to incorporate the Minkowski norm:

$$\left(\frac{\int |f_x^\sigma(x)|^p dx}{\int dx}\right)^{\frac{1}{p}} = ke \tag{2.20}$$

Color constancy based on this equation assumes that the p-th Minkowski norm of the derivative of the reflectance in a scene is achromatic. There are two special cases [107]. For $p = 1$, the illuminant is derived by a normal averaging operation over the derivatives of the channels. For $p = \infty$, the illuminant is computed from the maximum derivative in the scene. There is a resemblance between the color constancy derivation in the Gray-World and Gray-Edge hypothesis. Both methods can be combined in a single framework of color constancy methods based on low-level image features. An advantage of such color constancy methods is that they are based on low computational demanding operations and the methods do not require an image database taken under a known light source for calibration as is necessary for more complex color constancy methods such as color gamut mapping, and color by correlation [107].

2.3.2 Gray-World Hypothesis

Buchsbaum [14] proposes the Gray-World hypothesis which assumes that the average reflectance in a scene is achromatic. In the original work, the hypothesis is used to derive that the average reflectance for the short-wave, middle-wave and long-wave regions is equal. The achromatic reflectance of a scene is:

$$\frac{\int s(\lambda, x)dx}{\int dx} = g(\lambda) = k \tag{2.21}$$

Equation 2.18 shows average reflectance differences, while Equation 2.21 shows average reflectance. Buchsbaum [14], for example, needed to make further assumptions on the basis functions for the camera sensitivities, the surface reflectances, and the light source spectra. The constant k is between 0 for no reflectance (black) and 1 for total reflectance (white) of the incident light, and the integral is over the domain of the scene. For such a scene with achromatic reflectance, it holds that the reflected color is equal to the color of the light source [107], since

$$\frac{\int f(x)dx}{\int dx} = k \int_w e(\lambda)c(\lambda)d\lambda = ke \tag{2.22}$$

From implementation point of view, a simple method of Gray-World Assumption enforcement would be to find the average values of the image's R, G, and B color components and use their average to determine an overall Gray value for the image. An alternative choice proposed by [5] is the general surface reflectance DataBase based Gray-World (DB Gray-World) algorithm. The general reflectance database attempts to characterize a wide range of surfaces, the mean of which is used to estimate the illuminant. The Gray-World algorithm produces good results when different colors are present in an image and the image is viewed under a single uniform illuminant.

2.3.3 max-RGB/White-Patch

max-RGB is based on the assumption that the reflectance which is achieved for each of the three channels is equal [107]:

$$\max_x f(x) = ke \tag{2.23}$$

where the max operation is executed on the separate channels:

$$\max_x f(x) = \Big(\max_x R(x), \max_x G(x), \max_x B(x)\Big) \tag{2.24}$$

This method is also explained as being derived from the white-patch hypothesis with the assumption that there is a white patch in the scene. Since a white patch reflects all the incident light, its position in the image can be found by searching for the maximum RGB values [107]. However, the max-RGB method does not require the maxima of the separate channels to be on the same location, hence it also obtains correct illuminant estimation results when the maximum reflectance is equal for the three channels.

2.3.4 Shades of Gray

The Gray-World and the max-RGB algorithm are two different instantiations of a more general color constancy algorithm based on the Minkowski norm. A Shades of Gray is computed by [107] [27]:

$$\left(\frac{\int (f(x))^p dx}{\int dx}\right)^{\frac{1}{p}} = ke \tag{2.25}$$

For $p = 1$, the equation is equal to the Gray-World assumption. For $p = \infty$, it is equal to color constancy by max-RGB. Finlayson and Trezzi [27] investigated the performance of the illuminant estimation as a function of the Minkowski norm and found that the best results are obtained with a Minkowski norm with $p = 6$.

2.3.5 Bayesian Color Constancy

For Bayesian color constancy, we use the approach of Gehler et al. [34]. In the Bayesian color constancy approach, the observed image pixels are modeled with a probabilistic generative model, decomposing them as the product of unknown surface reflectances with an unknown illuminant. Using Bayes rule, a posterior for the illuminant is obtained, and from this the estimate with minimum risk is extracted.

The algorithm for estimating the illuminant has two parts [34, 84]: (1) Discretizing the set of all illuminants on a fine grid and computing likelihood (2) Selecting an illuminant which minimizes the risk. The likelihood of the observed image data $\mathbf{Y}$ for a given illuminant ℓ is [84]:

$$p(\mathbf{Y}|\ell) = \int_{\mathbf{X}} \left(\prod_i p(y(i)|\ell, x(i))\right) p(\mathbf{X}) d\mathbf{X} \tag{2.26}$$

where $x = (x_r, x_g, x_b)$ is the reflectance, with each channel ranging from zero to one. $\ell = (\ell_r, \ell_g, \ell_b)$ is the power of light in each channel, and $p(\mathbf{X})$ is the reflectance distribution. $y(i)$ is pixel with reflectance $x(i)$ for ith pixel. The posterior probability for ℓ is:

$$p(\ell|\mathbf{Y}) \propto \left(\int_{\mathbf{X}} \left(\prod_i p(y(i)|\ell, x(i))\right) p(\mathbf{X}) d\mathbf{X}\right) p(\ell) \tag{2.27}$$

The next step is finding an estimate of ℓ with minimum risk. Gehler et al. [34] use a grid over all the admissible illuminants, computing the posterior mean in a single loop over all those illuminants in the grid. For estimation of the parameters for reflectance prior $p(\mathbf{X})$ and the illumination prior $p(\ell)$, training data consisting of images with known illuminant color is used. During testing, the likelihood for all training illuminants is computed and a likelihood-weighted average in chromaticity space is taken.

2.4 Color Spaces and Skin color

In this section, skin color modeling methods are reviewed with emphasis on color spaces for skin detection. To model and classify skin color properly, the choice of the appropriate color space is crucial. In Yang et al. [113], it is observed that for skin colors, intensity is more likely to change than chrominance. Therefore, many approaches disregard the intensity information in their detection process. Additionally, Yang et al. showed that clusters in normalized RGB are an appropriate model for skin color and therefore many successful approaches rely on this color space e.g. [90, 13, 15]. Still, the normalized RGB color space suffers from instability with dark colors. The HS* color spaces are regarded as good measurement for skin detection, and are widely used in the scenarios of skin detection. Examples are found in [13, 32, 33]. Perceptually uniform color spaces like the CIELAB, CIELUV are used for skin detection e.g. in [16]. Orthogonal color spaces like YCbCr, YCgCr, YIQ, YUV, YES form components as independent as possible. YCbCr is one of the most successful color spaces for skin detection and used in e.g. [112, 46].

In [88], 845 images (more than 18.6 million pixels with manually annotated ground truth) are used for comparison of nine color spaces. With nine color spaces, the absence or presence of the illuminance component and the skin color modeling approaches are compared for indoor and outdoor scenarios with varying modeling parameters. The performance evaluation is based on Receiver Operating Characteristic (ROC). From the experimental results it is concluded that: (a) Color space transformations do affect performance in certain instances, (b) Performance decreases with the absence of the illuminance component, (c) The skin color modeling technique has greater impact on skin detection performance than color space transformation. The best performance is reported for indoor images by transforming pixel colors to the HSI or SCT (Spherical Coordinate Transform) color spaces, keeping the illuminance component and histogram based skin color modeling.

Shin et al. [93] evaluated the performance of nine color spaces. The RGB color space is used as a baseline for performance measure. The University of Oulu Physics-Based Face database (UOPB) [69] and the face dataset (AR) at Purdue [71] are used for the skin samples. For non-skin pixels the University of Washington's content-based image retrieval dataset is used. The skin and non-skin separability was evaluated using four metrics, two based on the scatter matrix and the other two were histogram based. The skin and non-skin separability was highest in the RGB color space (or absence of color space transformation) according to three of four separability metrics. Disregarding the illuminance component was found to have a negative effect on the performance and significantly worsen the separability in three of four metrics as well.

Four color spaces are evaluated in [1], using 200 images from the ViBE video dataset

[119]. It is shown that for every color space there exists an optimum skin detection scheme such that the overall performance is the same, i.e., the skin and non-skin separability is independent of color space transformation that is invertible. The claim is demonstrated for color spaces where invertible transformations exist. Experimentally, with RGB, HSV, and YCbCr color spaces using over 200 skin images, it is shown that the performance of all three color spaces is the same. CbCr components of the YCbCr space showed a lower performance than the other three color spaces because it is not invertible back to the other color spaces.

Nine color spaces are compared in [105] for skin color detection. Skin distribution is modeled as a single Gaussian and a mixture of Gaussians. Images of faces of Asians and Caucasians were captured under slowly varying illumination conditions using a single camera. For color space comparisons, images downloaded from the Internet were used. True Positive (TP) and True Negative (TN) rates are used as evaluation measures. It is reported that the normalized color spaces, especially the normalized T-S space yields best results and is the recommended color space for skin detection. Regarding the skin color modeling technique, it made no difference whether the distribution was modeled with a single Gaussian or with a mixture of Gaussians. For the choice of color spaces that are not illumination normalized, the mixture of Gaussians performed better than the single Gaussian.

Zarit et al. [121] compared five color spaces. For testing and training, different images downloaded from a variety of sources, including frames from movies and television are used, covering a wide range of skin tones, environments, and lighting conditions. It is reported that HSV combined with the lookup table method has higher skin detection performance using the percentage skin correct measure. In case of the lookup table method, color space transformation affects the overall skin detection performance whereas, in the case of the Bayesian approach, the skin detection performance is independent of the choice of a color space.

In [72], color based skin detection is compared using seventeen color spaces. The problem of varying illumination is addressed using the University of Oulu Physics-Based Face dataset [69], containing face images recorded under 16 different illumination/camera calibration conditions and four different cameras. The RGB skin pixels were transformed into 17 color spaces for determining the range of skin color in the different color spaces and the overlapping percentage between different skin groups (Asian and Caucasian). The overlapping extends from 50% to 80% with only one illuminant color, depending on the camera and the color space while for all illuminant colors, the overlapping is up to 98.8%. It is deduced from the skin locus (range of possible skin colors), that none of the color spaces offer invariance for illumination color change and that the skin locus is a useful tool for increasing tolerance for illumination color changes and non-uniformity. Regarding the overlap between the data of different skin color groups and size of skin locus in color space, the I1I2I3 color space [35] outperformed every color space. It is also demonstrated that for most of the color spaces, the skin locus may be modeled by one or two functions of up to quadratic order only.

According to [36, 38, 10], a single color space may limit the performance of the skin color filter and better performance can be achieved using two or more color spaces. Using the most distinct invariant color coordinates of different color spaces increases the performance under certain conditions. The combination of different color spaces increases the reliability of the classification results. It eliminates false positives since the combination stabilizes the area that is used for skin detection.

Table 2.1 summarizes the work done with emphasis on color spaces for skin detection. In all these approaches, the color spaces are used independently.

Table 2.1: Skin detection approaches with emphasis on color spaces.

Paper	Dataset	# Images	Color Spaces	Methods	Best
Schmugge et al.[88]	UOPB, AR	845	9	Hist./ND.	HSI/SCT
Shin et al.[93]	UOPB+AR	6,112	9	Scatter/Hist.	RGB
Albiol et al.[1]	ViBE	200	4	Hist.	RGB, HSV, YCbCr
Terrillon and Akamatsu[105]	Manual	300	9	Gaussian	TSL
Zarit et al.[121]	Manual	112	5	Hist./Bayesian	HSV
Martinkauppi et al.[72]	UOPB	4,000	17	Hist.	I1I2I3

2.5 Skin Color Modeling

In this section, skin detection work is reviewed with emphasis on skin color modeling techniques. Skin color modeling can be viewed as a two class problem: skin-pixel vs. non-skin pixel classification. For skin color, classification is modeled using different techniques. The most basic of all these approaches is explicit color space thresholding. Non-parametric skin modeling methods estimate skin color distribution from the training data without using an explicit model of the skin color [39]. The parametric skin color modeling estimates skin color distribution from the training data by making inferences about the parameters of an explicit model of the distribution. Illumination adaptation (dynamic) approaches adapt to the changing lighting conditions. Table 2.2 summarizes non-parametric, parametric and dynamic approaches.

2.5.1 Color Space Thresholding

Human skin color can be approximated in a well defined cluster given a color space, if the recording conditions for the images remain consistent (illumination controlled environment)[113]. Based on this idea, one method is to build a static skin classifier. A static skin classifier defines explicitly (using a number of rules) the boundaries the skin cluster has in a color space. Single or multiple ranges of threshold values for each color space component are created and the image pixel values falling within these range(s) for all the chosen color components are defined as skin pixels. The advantage of this method is the simplicity of skin detection rules and the computational efficiency because it is pixel based. The main difficulty achieving high recognition rates with this method is the need to find both a good color space and adequate decision rules empirically [109]. Generally, the TP rate is high but at the same time due to the large boundary of the static filter, the FP rate is also high.

Chai and Ngan [21] exploit the spatial distribution of human skin color in images. A static skin filter is derived and uses the chrominance components of the image for skin pixel detection. It is assumed that the different skin colors that are perceived in the image cannot be differentiated by the chrominance information of the corresponding image region and therefore, skin color can be represented by the static values of Cb and Cr component of the YCbCr color space. The ranges for the static filters are found by testing on a large number of images and then tuning the corresponding values in case of violations. The final values reported are,

$$Cb_{max} = 127, \quad Cb_{min} = 77, \quad Cr_{max} = 173, \quad Cr_{min} = 133 \tag{2.28}$$

A pixel is skin, if it lies between these values.

Peer et al. [80] advocate the usage of the RGB color space for face detection. They specifically deal with the problem of varying illumination and compensate for lighting correction using the Gray World algorithm and Color by Correlation technique. Classification of skin color is performed by heuristic rules taking into account two different illumination conditions: uniform daylight and lateral illumination. A filter for uniform daylight illumination:

$$\begin{gathered} R > 95, G > 40, B > 20 \qquad (2.29) \\ (Max\{R, G, B\} - min\{R, G, B\}) > 15 \\ |R - G| > 15, R > G, R > B \end{gathered}$$

A filter for daylight lateral illumination (flashlight):

$$\begin{gathered} R > 220, G > 210, B > 170 \qquad (2.30) \\ |R - G| \leq 15, B < R, B < G \end{gathered}$$

A static filter for the normalized RGB color space is reported in [40]. The paper describes a new constructive induction algorithm for creating adequate attributes to constitute the skin map. Using a simple set of operators and the three normalized RGB components, a model for skin detection is presented with a combination of different rules. The Restricted Covering Algorithm (RCA) is used for selective learning during the training phase. RCA is based on selection of single well defined separable rules. RCA performs its search in parallel for finding a single set of rules. There are different combinations of rules reported and the highest precision and accuracy is reported for the following rule:

$$\frac{nr}{ng} > 1.185, \frac{nr.nb}{(nr + ng + nb)^2} > 0.107, \frac{nr.ng}{(nr + ng + nb)^2} > 0.112 \tag{2.31}$$

where nr, ng and nb correspond to normalized coordinates. Skin detection is used as a cue for face detection in [92] using the HSI color space. A binary skin map is generated for oriental face detection for locating multiple faces in natural scenes. A clustering-based splitting algorithm is used to separate facial and non-facial regions in the skin color map.

The HSI color space is favored because of its stable behavior in non-uniform lighting conditions. Skin is segmented in the HSI color space based on the following rules:

$$\begin{aligned}
&I > 40 \\
&\ 13 < S < 110, 0^\circ < H < 28^\circ \quad \text{Or} \quad 332^\circ < H < 360^\circ \\
&\text{Or} \\
&I > 40 \\
&\ 13 < S < 75,\ 309^\circ < H < 331^\circ
\end{aligned} \tag{2.32}$$

A static skin filter in the HSV color space is reported in [106]. The authors argue that skin color can be accurately characterized by hue and saturation. The thresholds used in the HSV color space for skin segmentation are:

$$\begin{aligned}
&V \geq 40 \\
&0.2 < S < 0.6; \\
&0^\circ < H < 25^\circ \quad \text{or} \quad 335^\circ < H < 360^\circ
\end{aligned} \tag{2.33}$$

The V component filters out dark colors. The range of saturation S excludes pure red or dark red colors. The hue H and saturation S account for slightly varying lighting conditions.

2.5.2 Non-Parametric Skin Color Modeling

Non-parametric methods construct a Skin Probability Map (SPM) which assigns a probability to each of the values of a discretized color space [10]. Several skin detection systems [95, 121, 96, 50, 10, 28] are based on a histogram based approach for skin pixel classification. Color histograms are stable object representations unaffected by occlusion, changes in the view, and can be used to differentiate a large number of objects [113]. In the histogram based skin color modeling technique, the color space is quantized into a number of bins, each corresponding to a particular range of color component value. The histogram is referred to as 2D or 3D depending on whether two or three components of the color space are used. The 2D or 3D histogram are also referred to as the the Lookup Table (LUT). The number of occurrences of a particular color, based on training data is stored into each bin followed by a normalization process. The histogram bin counts are converted into a probability distribution, $P(c)$ as follows:

$$P(c) = \frac{count(c)}{T} \tag{2.34}$$

where $count(c)$ is the count in the histogram bin, c is color associated with the $count(c)$ and T is the total count obtained by summing the counts in all the histogram bins.

Jones and Rehg [50] constructed a 3D RGB histogram model from 18,696 web images. It was observed that 77% of the possible RGB colors are not encountered and most of the histogram was empty. Only 10% of the total pixels encountered were skin pixels.

Table 2.2: Summary of skin color modeling approaches. (CS: color space, DM: dynamic method, TP: true positives, n/a: not available)

	Paper	Dataset	# Images	CS	Method	DM	TP
Non Parametric	Jones and Rehg[50]	Compaq	18,696	RGB	Bayes	–	90
	Brown et al.[13]	Compaq+Manual	19,196	TSL	SOM	–	78
Parametric	Jones and Rehg[50]	Compaq	18,696	RGB	GMM	–	90
	Jedynak et al.[49]	Compaq	18,696	RGB	MaxEnt	–	82.9
	Lee and Yoo[62]	Compaq	18,696	Xyz	Ellip. Model	–	90
		Compaq	18,696	YCbCr	SGM	–	90
		Compaq	18,696	Xyz	GMM	–	90
	Sebe et al.[90]	Compaq	18,696	RGB	BN	–	98.32
	Phung et al.[81]	ECU	4,000	RGB	Bayes	–	88.9
		ECU	4,000	RGB	MLP	–	88.5
		ECU	4,000	YCbCr	SGM	–	88
		ECU	4,000	YCbCr	GMM	–	85.2
	Yang and Ahuja[116]	Michi. Face DS	200	CIELUV	GMM	–	n/a
	Seow et al.[91]	Manual	410	RGB	NN	–	n/a
	Greenspan et al.[41]	Manual	682	RGB	GMM	–	n/a
Dynamic	Peer et al.[80]	Manual	40	YUV	Ellip. Model	GW	n/a
	Hsu et al.[46]	HH1,Champion	66,227	YCbCr	SGM	WP	96
	Nayak and Chaudhri[77]	Manual	n/a	RGB	Condensation	NN	n/a
	Kakumanu et al.[52]	Manual	326	RGB	Thresh.	NN	n/a
	Stoerring et al.[102]	UOPB	4,000	RGB	Thresh.	Skin Locus	n/a
	Cho et al.[24]	Manual	379	HSV	Thresh.	Adapt. Thresh.	n/a
	Yang et al.[113]	n/a	n/a	RGB	SGM	SGM Adapt.	n/a
	Oliver et al.[78]	n/a	n/a	RGB	GMM	GMM Adapt.	n/a
	Soriano et al.[97]	UOPB	4,000	RGB	Skin Locus	Hist. Adapt.	n/a
	Sigal et al.[94]	Manual	720	HSV	Bayes	HMM Adapt.	86.8

Jones and Rehg also computed skin and non-skin histograms. Given skin and non-skin histograms, the class conditional probability was defined as:

$$P(c|skin) = \frac{s(c)}{T_s}, \qquad P(c|non-skin) = \frac{n(c)}{T_n} \tag{2.35}$$

where $s(c)$ is the pixel count in the color c-bin of the skin histogram and $n(c)$ is the pixel count in the color c-bin of the non-skin histogram. T_s and T_n represent the total counts in the skin and non-skin histogram bins. From the generic skin and non-skin histograms, it is concluded that there is a reasonable separation between skin and non-skin classes. It can be used to build fast and accurate skin classifiers even for images collected from unconstrained imaging environments such as web images, if the training dataset is sufficiently large [50]. With the class conditional probabilities of skin and non-skin color models, a skin classifier can be built using Bayes Maximum Likelihood (ML) approach [109]. Using this, a given image pixel can be classified as skin, if

$$\frac{P(c|skin)}{P(c|non - skin)} \geq \theta \tag{2.36}$$

where θ is a threshold value which can be adjusted to trade-off between true positives and false positives.

Brown et al. [13] proposed a self organizing map (SOM) based skin detection approach. They trained two SOMs, skin-only and skin + non-skin with a dataset of about 500 manually labeled images. The performance was evaluated on the authors' training/test images set and the Compaq skin database [50]. Several color spaces were tested with the SOM detector. The performance of the SOM based skin detection was observed to be almost independent of the color space. The SOM performance on the 500 images dataset is reported to be marginally better than a Gaussian mixture model, while for the Compaq database the SOM performance is inferior to the RGB histograms. In favor of SOM, the authors argue that the SOM based method requires less resources and can be implemented for real-time applications using SOM hardware.

2.5.3 Parametric Skin Color Modeling

The need for a more compact skin model representation for certain applications along with the ability to generalize and interpolate the training data stimulates the development of parametric skin distribution models [109]. In the following, such parametric skin color modeling approaches are reviewed.

Single Gaussian Model (SGM) The skin color of different individuals occupies a well defined cluster in a given color space, if the images are recorded under controlled lighting conditions. Hence, under certain lighting conditions, the skin-color distribution of different individuals can be modeled by a multivariate normal (Gaussian) distribution in normalized color space [51, 113, 114]. As such an Elliptical Gaussian joint probability distribution function is used to model the skin-color distribution:

$$p(c) = \frac{1}{(2\pi)^{1/2}|\sum|^{1/2}} \exp\left[-\frac{1}{2}(c-\mu)^T \sum (c-\mu)\right] \tag{2.37}$$

where μ and $\sum$ are the mean vector and the covariance matrix respectively, and c is the color vector. $\sum$ and μ are estimated over all the color samples from the training data using Bayes maximum likelihood (ML) estimation. A certain threshold obtained from the training data is used to compare the probability obtained [60, 115].

Gaussian Mixture Models (GMM) Yang and Ahuja [116] showed that though skin color clusters in a small region in a color space, different modes co-exist within this cluster and hence cannot be effectively modeled by a single Gaussian distribution. This is also true for skin images recorded in uncontrolled and varying illumination circumstances. Hence Yang and Ahuja suggest using the Gaussian mixture model. A Gaussian mixture density function is obtained by adding individual Gaussians:

$$p(c) = \sum_{i=1}^{N} w_i \frac{1}{(2\pi)^{1/2}|\sum_i|^{1/2}} \exp\left[-\frac{1}{2}(c-\mu_i)^T \sum_i (c-\mu_i)\right] \tag{2.38}$$

where c is a color vector and $\sum_i$, μ_i are the covariance matrix and the mean. The number of Gaussians are represented by N and the w_i is the weight factor which is the contribution of ith Gaussian. Expectation-maximization (EM) is used to estimate the parameters ($\sum_i$, μ_i, and w_i) from the training data [116]. Yang and Ahuja used two Gaussians in the CIELUV color space using a face database. In [41], the authors also use a two component GMM, one component for capturing the distribution of the normal light skin color and the other for capturing the distribution of the more highlighted regions on the skin. Lee and Yoo [62] use six Gaussians for skin classification. Four Gaussian components are used in [47]. For skin segmentation, Gaussian mixture models are also used in [50, 16, 73, 78].

Lee and Yoo [62] proposed an elliptical boundary model using a threshold value (chosen empirically), for classifying pixel as skin or non-skin. Though the elliptical boundary model out performed GMM, its limitation is its application to binary classification problems.

Multi Layer Perceptron (MLP) A three layered feed forward neural network for skin classification is used in [22]. In [53], the YCbCr color space is used for skin classification. Only the Cb and Cr components of the YCbCr are used with the MLP. A similar color space with MLP skin classification is presented in [81]. Using the RGB color space with an MLP is presented in [86]. A dataset of web images are used and a three layer MLP is trained with further fine tuning using a Gaussian model. A three layered MLP is used in [91] with the RGB color space extracting skin regions and also using it for interpolating the skin regions in the 3D color cube.

Maximum Entropy (MaxEnt) Classifier MaxEnt finds its application in areas related to speech recognition, audio-visual speech analysis and natural language processing domains. The MaxEnt model for skin detection is used by Jedynak et al. [49] using the Compaq dataset. The feature set consists of the color of the pixel and the two neighboring adjacent pixels. The Bethe tree approximation [83] is used for parameter update and the Gibbs sampler algorithm [19] is used for determination of the probability of skin for individual pixels. Due to the complex set of parameters, one of the drawbacks of MaxEnt is extended training times.

Bayesian Network (BN) Sebe et al. [90] use a BN, proposing a new method for learning the structure of the BN with labeled and unlabeled data. In the case of learning with labeled and unlabeled data for skin detection using Bayesian networks, the authors suggest to start Naive Bayes (NB) and Tree-Augmented Naive Bayes (TAN) classifiers and learn only with the available labeled data. Then test whether the model is correct by learning with the unlabeled data. If the results are not satisfactory, then Stochastic Structure Search (SSS) is advised. If none of the methods using the unlabeled data improve performance over the supervised TAN or NB then either unlabeled data can be discarded, or labeling some of the unlabeled data using the active learning methodology can be done. With a training data of 600 labeled + 54,000 unlabeled samples from the Compaq dataset, the proposed approach achieves detection rates of 95.82% for 5% false positives and 98.32% for 10% false positives.

2.5.4 Dynamic Approaches

Though human skin color is clustered in a small region in a given color space, the skin color of the same person under varying lighting conditions differs. Even under consistent illumination circumstances, background, shadow and reflections influence skin-color distribution. Furthermore, if a person is moving, the apparent skin color changes as the person's position relative to the camera or light changes. This effect is more pronounced in video sequences where the skin color in the consecutive frames slightly differs from the previous frames due to lighting effects, reflections and camera position. Human vision can adapt to the changing lighting conditions. This ability of humans to retain a stable visual representation of the objects color is known as color constancy. Illumination adaptation approaches are used for skin detection.

Lighting correction for skin detection is demonstrated in [80]. The authors eliminate the influence of non-standard illumination from images by applying Gray-World color constancy and Color by Correlation. On a dataset of 40 images, it is shown that skin color detection performance can be improved by applying the lighting correction algorithm. It is concluded that face detection based on skin detection with lighting correction is more robust and that the installation "15 Seconds of Fame" [80] can be exhibited almost anywhere. The white patch based skin color constancy is used in [46]. The white patch based color corrected images are transformed into the YCbCr color space. An elliptical boundary model with Mahalanobis distance is used to detect skin pixels. The authors report 96% skin detection rate on a face dataset. The dataset consists of frontal and non-frontal faces, recorded in different lighting conditions with varying background. Nayak and Chaudhuri [77] use a Neural network (NN) based color constancy algorithm for skin detection. RGB components of the skin color are used as the input to NN. For training, the authors use human palm images recorded under varying illumination conditions. It is concluded that NN can predict the illuminant parameters and NN based lighting corrected palm images can be tracked precisely using conditional density propagation (Condensation [48]) in a variety of varying illuminations and different background scenarios. Kakumanu et al. [52] also use an NN for skin color constancy. For skin detection, a simple thresholding technique is used for the NN based lighting corrected images. Störring et al.[100, 101, 102] use a physics-based model for color constancy for skin detection. With this model the authors describe the expected area for the skin chromaticities under assumed illuminations. They use the term skin locus which is defined as the knowledge of skin pixels defining an area in chromaticity space. Once the skin locus is defined, simple thresholding can be applied for classifying skin from non-skin pixels for a range of different illumination and white balancing conditions.

An adaptive skin color filter is proposed by Cho et al. [24] for the detection of skin color regions in color images. The method consists of two stages. First, a thresholding box in the HSV color space is updated adaptively using a color histogram. During the second stage, color vectors inside the thresholding box are classified into skin color vectors and background color vectors. The approach is tested on 379 images obtained from the Internet. It is argued that the previous methods adopt a fixed threshold scheme and therefore they are useful only in a restricted (i.e., controlled) environment. The proposed method is applicable to images in more general situations since it is can adaptively adjust thresholds and electively separate skin color regions from similar background color regions.

Yang et al. [113] use a GMM for skin modeling in the *rg* space proposing an adaptive approach for face tracking based on skin color. A linear combination of the parameters based on maximum likelihood approach are used to approximate the mean and covariance of the model for varying illumination. The authors recommend this adaptive model for varying indoor illumination conditions. Zhu et al. [123] use the EM algorithm for training an adaptive GMM, where four Gaussian components are used to model the background and one to model hand color. An adaptive GMM in *rgb* space is used in [78] for changing illumination conditions using a mixture model obtained off-line with the EM algorithm. An incremental EM technique is used for dynamically updating GMM parameters. Similarly, McKenna et al. [73] use a GMM in the HSV color space using stochastic equations [61] for parameter update. For removing outliers, log-likelihood measurements are used. For the position and the spatial extent of skin regions, Schwerdt and Crowley [89] use the first-order moments of the skin color histogram in *rg* space. For robust tracking, the skin histogram is weighted using a Gaussian function. The mean and covariance of a new frame are updated using the previous frame.

Similar to [100, 101, 102], Soriano et al. [97] use skin locus based adaptive skin color modeling. The process includes skin pixel extraction from the tracked area within the skin locus which is defined in a normalized r and g chromaticity box. The histogram of these pixels and the histogram of the whole image is used to update the ratio histogram followed by the back projection of the histogram for defining the search space for an incoming frame.

Skin color-based video segmentation under time-varying illumination is described in [94]. The authors use an explicit second order Markov model for predicting an evolving skin color (HSV) histogram over time. The current segmentation and predictions of the Markov model is used to dynamically update color histograms. They parametrize the evolution of the skin-color distribution at each frame by translation, scaling, and rotation in color space with subsequent changes in geometric parameterization of the distribution propagated by warping and re-sampling of the histogram. They estimate the parameters of the discrete-time dynamic Markov model by Maximum Likelihood Estimation evolving over time.

2.6 Summary

This chapter introduced the background of this thesis and reviewed related work that is concerned with skin detection. The eight classifiers (NaiveBayes, Bayesian network, J48, Random Forest, RBF network, multilayer perceptron, Adaboost.M1 and SVM) are introduced which are used as skin color modeling techniques in the derivation of important results in Chapter 3. A brief introduction to the color spaces used for various approaches in this thesis is given. An overview of five color constancy algorithms is presented, namely Gray-Edge, Gray-World, max-RGB, Shades of Gray, Gray-Edge and Bayesian color constancy. We also presented an extensive survey of the up-to-date techniques for skin detection using color information. A good skin classifier must be able to discriminate between skin and non-skin pixels for a wide range of people with different skin types and be able to perform well under different illumination conditions. Obtaining robust color representations against varied illumination conditions is a major challenge. However, the application of color constancy techniques for skin-color modeling proved to improve performance.

Chapter 3

Static Skin Segmentation and Skin Classification

This chapter deals with static skin filters (Section 3.1) and color based skin classification (Section 3.2). Two new static skin filters for IHLS and CIELAB color spaces are introduced and compared to the state-of-the-art static filters. The effect of five color constancy algorithms (Gray-Edge, Gray-World, max-RGB, Shades of Gray, Gray-Edge and Bayesian color constancy) is studied for the static skin filter in the YCbCr color space. The approaches used based on off-line training (8 classifiers) are evaluated for skin segmentation with different color spaces. Moreover, their effect on performance in the presence of 3D color spaces and 2D color spaces is studied and results are derived. The effect of color constancy algorithms is reported for the Random forest classifier.

3.1 Static Skin Filters

The advantage of static filters is the simplicity of skin detection rules. This results in the construction of a classifier which is computationally favorable [109]. For the static filters, we need to find both a good color space and adequate decision rules empirically. Generally, the true positive rate can be increased by tuning but at the same time the false positive rate is also affected [55, 57, 59]. We introduce two new static skin filters in the IHLS and CIELAB color spaces. The two new static filters and four state-of-the-art static filters in YCbCr, HSI, RGB and normalized RGB color spaces are evaluated on the two datasets DS1 and DS2, on the basis of F-measure.

For the IHLS color space, we built a static filter from the skin distribution in Weka [45] and refined the corresponding values on test images. Finally, the following rules are used:

$$iH_{max} = 50,\ iH_{min} = 0,\ iS_{max} = 0.9,\ iS_{min} = 0.1 \tag{3.1}$$

where iH_{max} and iH_{min} are the upper and lower boundary values for the hue component, iS_{max} and iS_{min} are the upper and lower boundary values for the saturation component of the IHLS color space. For CIELAB, we also built a static filter using the same procedure as that of the IHLS color space and finally using the following rules:

$$a_{max} = 14,\ a_{min} = 2,\ b_{max} = 18,\ b_{min} = 0.7 \tag{3.2}$$

For YCbCr, the static filter reported in [21] is used. For HSI, the static filter of [92] is used. For the RGB color space, the static filter reported in [80] is used. For normalized RGB, static filter of [40] is used.

3.1.1 Evaluation

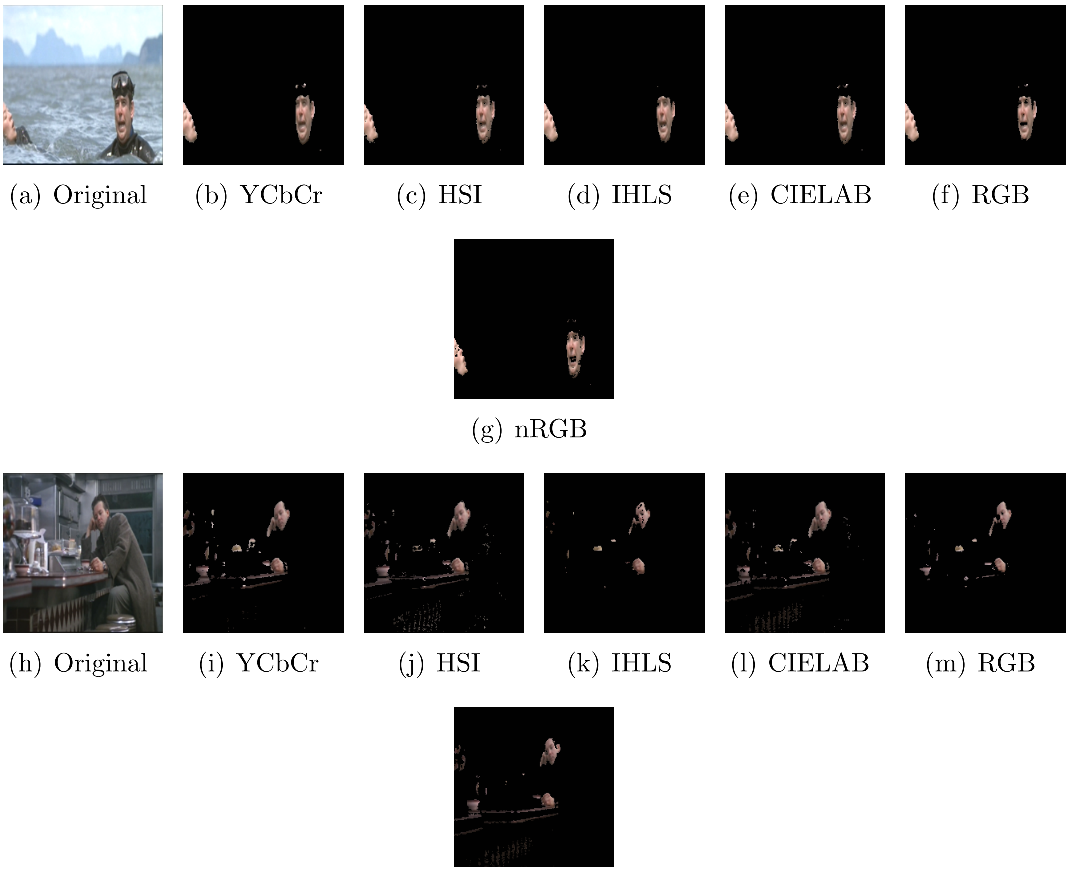
Figure 3.1 shows the output of these static skin filters on two images. Figure 3.1 (second row) shows an actor from a movie scene with items having skin like colors. The static filter in the IHLS color space (Figure 3.1(k)) reports fewer false positives compared to the other five static filters. Figure 3.2 shows the F-measure for static filters of the six color spaces for datasets DS1 and DS2. For DS1, The highest F-measure of 0.50 is reported by the CIELAB static filter and the lowest F-measure of 0.43 by the normalized RGB. The second highest F-measure of 0.49 for DS1 is reported by the HSI and the RGB static filters. YCbCr achieves an F-measure of 0.45. For the dataset DS2, the highest F-measure of 0.50 is reported by the normalized RGB static filter and the lowest F-measure of 0.33 by YCbCr. The second highest F-measure of 0.49 is reported by the HSI static filter. IHLS, CIELAB and RGB achieve F-measures of 0.38, 0.37 and 0.45 respectively.

The two datasets DS1 and DS2 represent skin colors in different lighting conditions, resulting in different skin locus (skin color ranges in a color space) for the corresponding dataset (and color space). Since, the static filters use static boundaries, any shift of skin color ranges from the static boundaries will result in varying performance. Therefore, the F-measure rankings of the color spaces are different for the datasets DS1 and DS2.

3.1.2 Effect of Lighting Correction

We show the effect of color constancy algorithms on static skin filter using Gray-Edge, Gray-World, max-RGB, Shades of Gray, Gray-Edge and Bayesian color constancy (described in Section 2.3). For this purpose, we select YCbCr as the color space and its static filter because of its transformation simplicity, explicit separation of luminance and chrominance and its wide usage for skin detection [109]. Figure 3.3 shows the output of the color constancy algorithms used. Figure 3.4 shows two examples of the result of skin detection after applying Gray-World color constancy. Figure 3.4(a)(d) shows original images, Figure 3.4(b)(e) shows skin detection using the static filter in the YCbCr color space and Figure 3.4(c)(f) shows improved skin detection using the same filter after applying the Gray-World algorithm. This result clearly shows the benefit of using lighting correction for skin detection. Color constancy can however negatively affect the results. As shown in Figure 3.5(c), the application of lighting correction has resulted in reporting pixels that do not belong to skin as skin pixels and thus increasing false positives. Figure 3.5(f) reports a scenario where the application of a lighting correction algorithm decreased true positives.

Figure 3.6 shows F-measure for DS1 skin-only set (Section 1.5.1) and DS2. We select DS1 skin-only set because we are interested, specifically in the impact of color constancy


(a) Original (b) YCbCr (c) HSI (d) IHLS (e) CIELAB (f) RGB

(g) nRGB

(h) Original (i) YCbCr (j) HSI (k) IHLS (l) CIELAB (m) RGB

(n) nRGB

Figure 3.1: Example results of skin detection using static skin filters in different color spaces. Black shows non-skin.

algorithms on skin pixels. For DS1, we find that the F-measure of 0.58 without using lighting correction is decreased to 0.55 by using the Gray-Edge algorithm. Gray-World also reports decreased performance with an F-measure of 0.46 while in the case of max-RGB, the skin detection performance is increased to 0.60. Shades-of-Gray reports decreased performance of 0.53 and Bayesian reports a slight increase with F-measure of 0.59 over the original uncorrected result.

For DS2 (Figure 3.6), by applying lighting correction, the results are improved in all the cases. We find that the F-measure is slightly increased by using the Gray-Edge algorithm and max-RGB. Gray-World reports increased performance with F-measure of 0.41. Shades-of-Gray reports an increased performance of 0.40 and Bayesian color constancy reports an increase of 0.35 over the original uncorrected result.

From the results, it can be concluded that the lighting correction before skin detection using static filters can improve skin detection performance. At the same time, lighting correction can have negative effects on the results. This is due to the fact that color constancy algorithms produces a compact representation of the skin locus, but the skin locus can also be shifted and deviated in the chromaticity space. Since the static filters use static boundaries, we therefore, get different results by applying color constancy

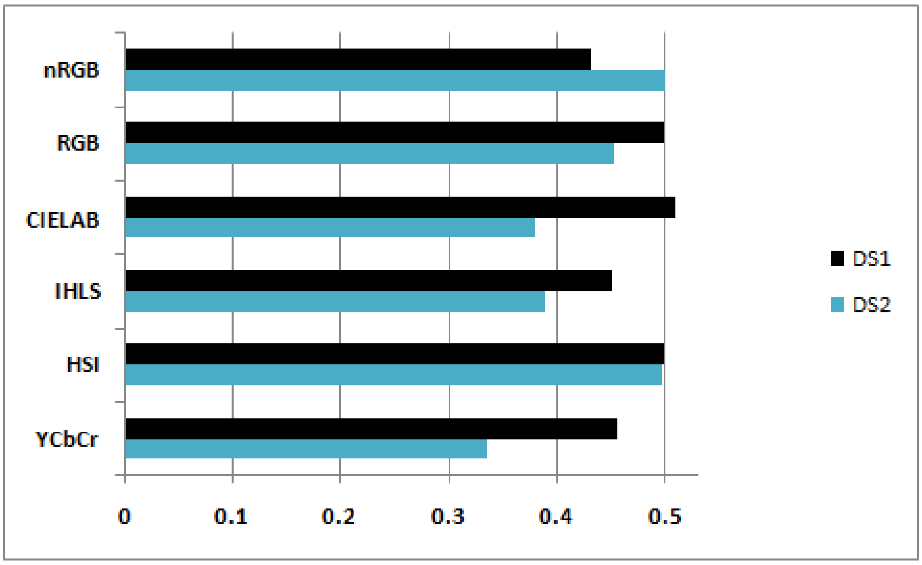


Figure 3.2: F-measure for DS1 and DS2 based on static skin filters in six color spaces.

algorithms.

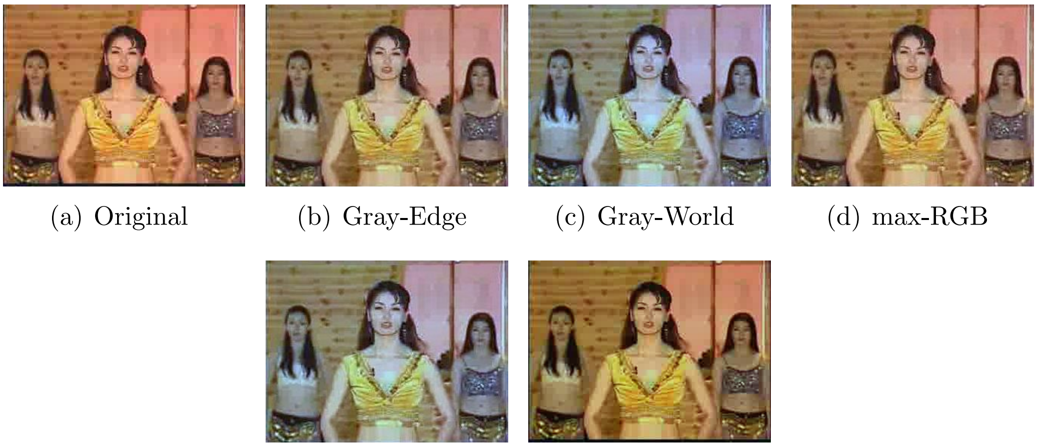
3.2 Color Pixel Classification

In this section, we deal with color based skin classification. As such, we investigate and evaluate (1) the effect of color space transformation on skin detection performance and finding the appropriate color space for skin detection, (2) the role of the illuminance component of a color space, (3) the appropriate pixel based skin color modeling technique, and finally (4) the effect of color constancy algorithms on color based skin classification. The importance of this evaluation is a comprehensive color space and skin color modeling technique that will help in the selection of the best combinations for skin detection.

It is commonly assumed that variation in skin colors occurs more in intensity than in chrominance and that robustness in skin detection can be achieved by dropping the illuminance component and using chrominance components only [88]. In our representation, when we use all the components of a color space, we refer to it as a 3D color space, while a 2D color space is without the illuminance component: L of IHLS, I of HSI, G of RGB, nG of nRGB, Y of YCbCr, and L of CIELAB are the illuminance components.

3.2.1 Skin Color Modeling

We use eight skin color modeling techniques (classifiers, Section 2.1) for pixel based skin classification: AdaBoost, Bayesian network, J48, Multilayer Perceptron, Naive Bayesian, Random Forest, RBF network and SVM, being all commonly preferable choices for clas-


(a) Original (b) Gray-Edge (c) Gray-World (d) max-RGB

(e) Shades-of-Gray (f) Bayesian

Figure 3.3: The outputs of different color constancy algorithms. (a) Original frame. (b)-(f): Results of applying the indicated algorithm.

sification problems. For each pixel in every color space, a feature vector is created by using all the three color channels in case of 3D color or using two color channels as the feature vector for the 2D color spaces. For the skin color modeling techniques considered, the performance is affected by the parameter settings. The parameters affect precision and recall in such a way that if changing a parameter increases precision, the recall is decreased and vice versa, but the over-all F-measure still remains the same. The F-measure reported is based on 10-fold cross validation. A detailed discussion on the eight classifiers used is presented in Section 2.1. We discuss the parameters for each classifier below.

AdaBoost: We use the multi-class case which requires the accuracy of the weak hypothesis greater than 0.5. During the training by 10-fold cross validation, using Decision Stump as the base classifier, the weight threshold of 100 and the number of iterations of 10 reports overall best performance in the case of 3D and 2D color spaces.

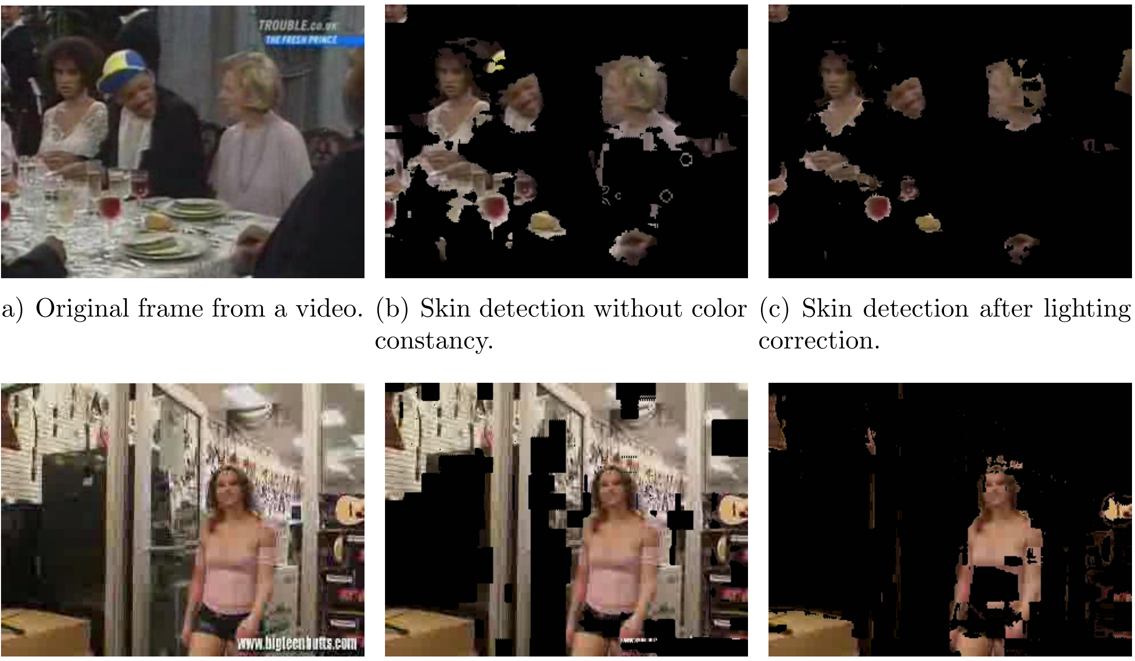
Bayesian Network (BayesNet): During the training by 10-fold cross validation, we obtain a high F-measure by setting the estimator parameter to 0.5 and using a hill climbing algorithm as the searching algorithm.

J48: The confidence factor and the minimum number of instances per leaf has an effect on the performance of skin classification. We achieve the highest performance with a confidence factor of 0.25 and minimum instances per leaf to be 2.

Multilayer Perceptron (MLP): The optimum performance is obtained by setting the learning rate (the amount the weights are updated) to 0.3 and the momentum applied to the weights during updating to 0.2.

Naive Bayesian (NaiveBayes): We find that using supervised discretization (to convert numeric attributes to nominal ones) and kernel estimation (for numeric attributes) rather than a normal distribution increases performance.

Random Forest: The most important parameter is the number of trees grown for the classification. We find that the highest F-measure is reported for 10 trees grown. Using less than 10 trees decreases overall performance and greater than 10 does not increase performance but rather converges to a stable performance. Also, limiting the depth of



(a) Original frame from a video. (b) Skin detection without color constancy. (c) Skin detection after lighting correction.

(d) Original frame from a video. (e) Skin detection without color constancy. (f) Skin detection after lighting correction.

Figure 3.4: Skin detection can be improved by first applying lighting correction using color constancy algorithms. For skin detection, a static filter in the YCbCr color space is used.

the trees grown decreases performance.

RBF Network (RBF): We find that the performance is independent of ridge value for the linear regression of RBF. The performance is however affected by the number of clusters selected with optimum performance for number of clusters being 2 for the problem at hand.

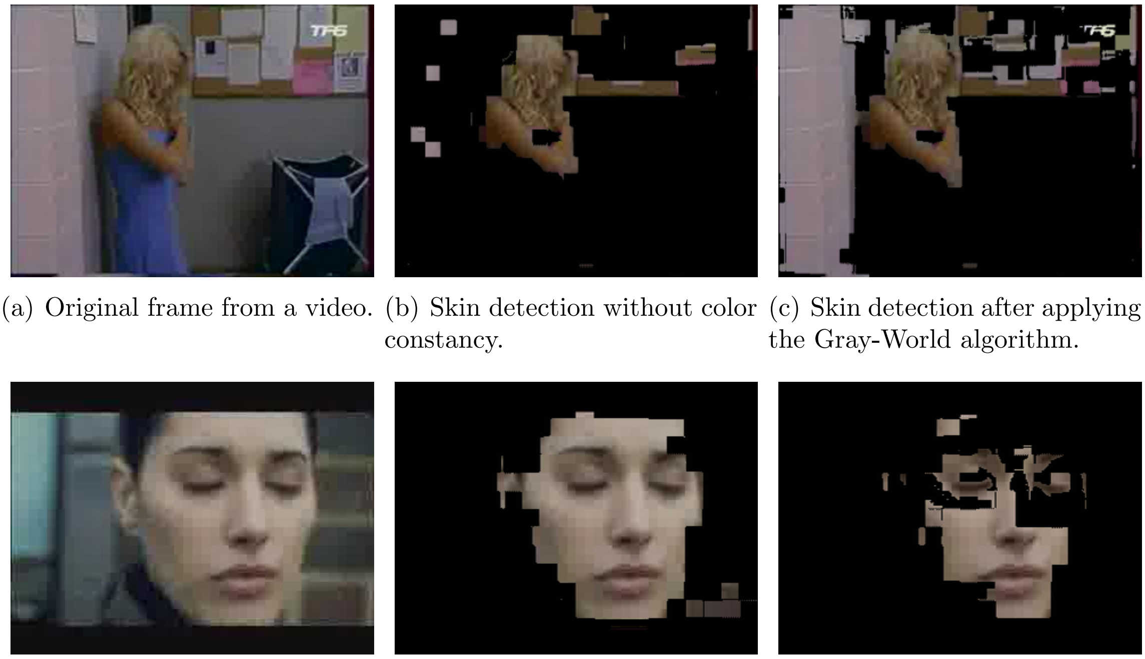
SVM: During the training by 10-fold cross validation, the performance is increased by using the polynomial kernel. The complexity parameter C was found out to yield maximum performance with $C = 1$ and the tolerance parameter of 0.9. For the complexity parameter, values below $C = 1$ decreased the overall performance, while for any increase above 1, we attained a stable performance close to that of $C = 1$.

3.2.2 Effect of Color Space Transformation

We investigate if a color space transformation improves skin and non-skin separability. As such we examine it by F-measure, comparing color spaces (RGB as a base line).

For Dataset DS1

Refer to Table 3.1, Figure 3.7 and Figure 3.8 for the effect of color space transformation on skin performance. In Table 3.1 and Figure 3.7, using RGB as a base line, the IHLS color space improved in all the eight skin modeling approaches. The HSI color space also



(a) Original frame from a video. (b) Skin detection without color constancy. (c) Skin detection after applying the Gray-World algorithm.

(d) Original frame from a video. (e) Skin detection without color constancy. (f) Skin detection after applying the Gray-World algorithm.

Figure 3.5: Color constancy can decrease skin detection performance in some cases. For skin detection, a static filter in the YCbCr color space is used.

improved in all cases except a slight decrease for the MLP and a significant decrease in the case of RBF. The worst performing of all the color spaces independent of the skin color modeling came out to be nRGB. YCbCr significantly improved in almost all the skin color modeling approaches. CIELAB also showed improved performance in all but for a slight decrease in the NaiveBayes case.

Since, we are using the RGB as a base line, a comparison of 2D color spaces with the 2D RGB can be seen in Table 3.2. The comparison of 2D and 3D color spaces is given in Section 3.2.3. For 2D color modeling compared to 2D RGB, the IHLS shows improvement in all the skin color modeling approaches and follows the trend of 3D color spaces. HSI follows the same trend of the IHLS color space significantly improving in all the cases except for the RBF network. nRGB shows significant improvement over 2D RGB in Bayesian network, J48 and Random Forest. YCbCr significantly worsened in case of RBF and slightly in case of SVM. CIELAB compared to RGB, shows significant improvement in 7 skin color modeling approaches with a slight performance decrease in the NaiveBayes case (Table 3.2).

What we experience is that independent of the skin color modeling, nRGB (represented in black in Figure 3.7 and Figure 3.8) shows to be unsuitable for robust skin classification. This is contrary to prior experiments in the literature, but can be explained with the very

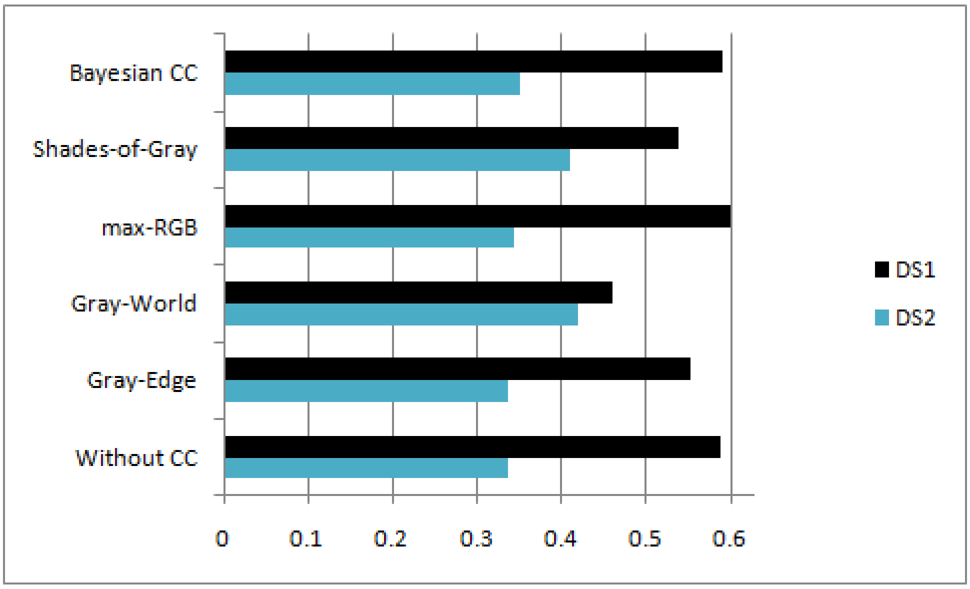


Figure 3.6: Static filter of YCbCr with color constancy (CC) on DS1 (skin-only) and DS2.

Table 3.1: F-measure for 3D color spaces with different skin color modeling approaches using DS1.

Classifier	IHLS	HSI	RGB	nRGB	YCbCr	CIELAB
AdaBoost	0.519	0.510	0.217	0.190	0.408	0.541
BayesianNet	0.615	0.587	0.300	0.545	0.521	0.542
J48	0.673	0.665	0.645	0.518	0.671	0.660
MLP	0.595	0.575	0.593	0.427	0.592	0.593
NaiveBayes	0.363	0.593	0.220	0.210	0.435	0.212
Random Forest	**0.740**	**0.733**	**0.706**	**0.632**	**0.702**	**0.724**
RBF	0.347	0.247	0.360	0.319	0.384	0.470
SVM	0.538	0.536	0.319	0.280	0.330	0.538

noisy dataset of on-line videos, which shows many dark colors where nRGB becomes unstable. IHLS (represented in blue in Figure 3.7 and Figure 3.8) reports over-all highest performance, outperforming other color spaces. Finally, we see that the color spaces improve more with the tree-based skin modeling (Random Forest and J48) than with other approaches.

For Dataset DS2

Refer to Table 3.3, Figure 3.9 and Figure 3.10 for the effect of color space transformation on skin performance. In Table 3.3 and Figure 3.9, we find that compared to RGB, the IHLS color space improved in all the eight skin modeling approaches except MLP. The HSI color space shows decreased performance in the case of AdaBoost, MLP and the RBF network. The worst performing of all the color spaces independent of the skin color modeling came out to be nRGB. YCbCr significantly improved in all the skin color modeling approaches with the exception of MLP and Random Forest. CIELAB also

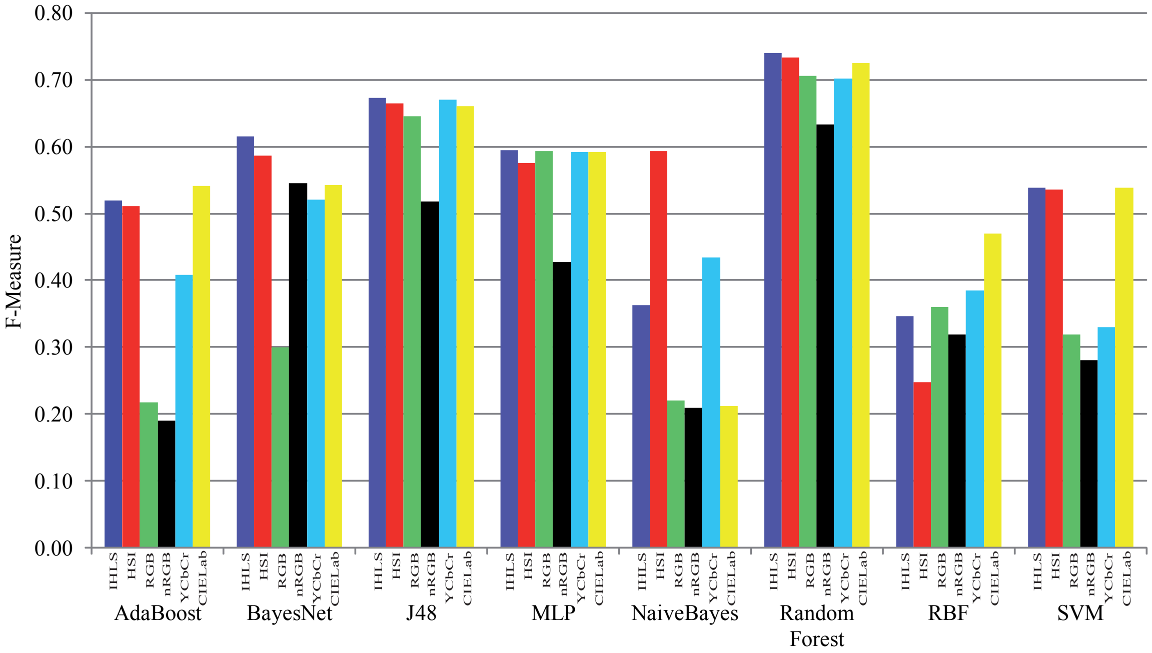


Figure 3.7: F-measure for skin color modeling approaches with 3D color spaces for DS1.

Table 3.2: F-measure for 2D color spaces (without the illuminance component) for eight color modeling approaches using DS1.

Classifier	IHLS	HSI	RGB	nRGB	YCbCr	CIELAB
AdaBoost	0.519	0.410	0.210	0.199	0.390	0.279
BayesNet	0.611	0.553	0.220	0.401	0.500	0.491
J48	0.633	0.550	0.466	0.510	**0.615**	0.571
MLP	0.557	0.520	0.412	0.400	0.568	0.471
NaiveBayes	0.308	0.554	0.220	0.220	0.375	0.174
Random Forest	**0.636**	**0.666**	**0.547**	**0.632**	0.607	**0.654**
RBF	0.381	0.165	0.358	0.269	0.177	0.446
SVM	0.520	0.530	0.317	0.270	0.300	0.393

showed improved performance in all but for a slight decrease in case of MLP.

For 2D color modeling (see Table 3.4), compared to 2D RGB, the IHLS shows improvement in all the skin color modeling approaches with the exception of MLP and NaiveBayes. HSI shows improvement in the case of BayesNet, J48, Random Forest and SVM. nRGB shows significant improvement over 2D RGB in the Bayesian network, J48, NaiveBayes and Random Forest. YCbCr significantly worsened in the case of MLP, NaiveBayes, RBF and SVM. CIELAB compared to RGB, shows significant improvement in 6 skin color modeling approaches with a slight performance decrease in case of AdaBoost and MLP (Table 3.4). A comparison of 2D and 3D color spaces is given in Section 3.2.3.

nRGB (represented in black in Figure 3.9 and Figure 3.10) does not show as bad per-

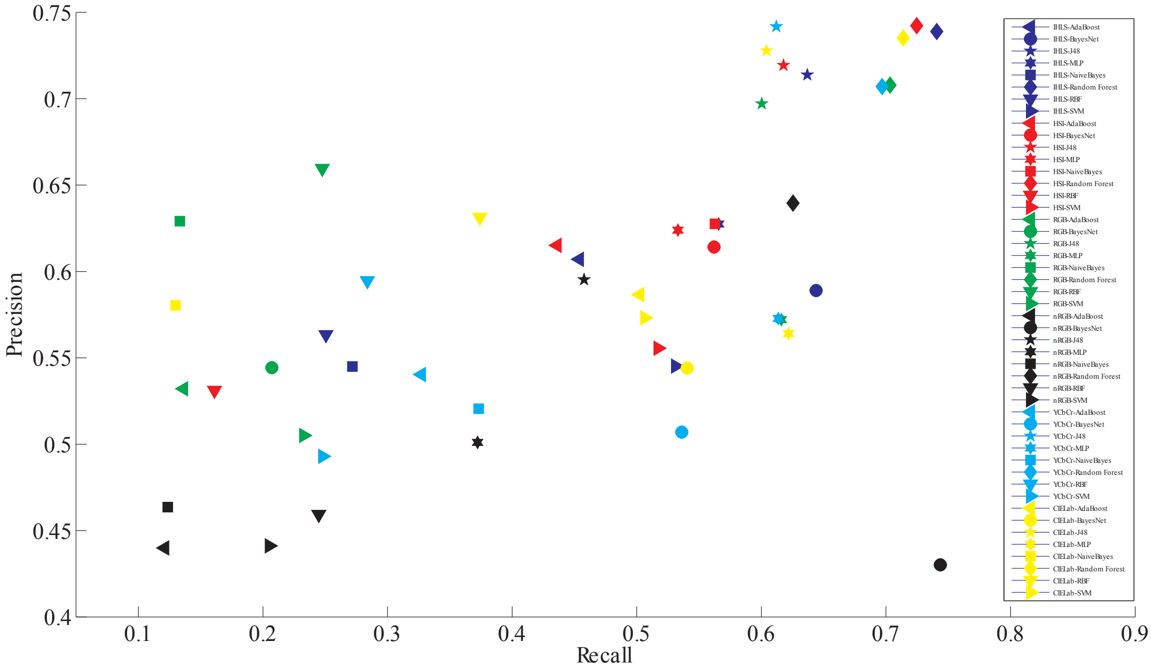


Figure 3.8: Precision and Recall space for skin-color modeling methods and 3D color spaces for DS1. Random Forest (diamond shaped) and J48 (five pointed stars) exhibit dominant clusters at the extreme top right and thus show overall good performance. nRGB (black data points) has the worst performance.

formance as in the case of DS1. IHLS (represented in blue in Figure 3.9 and Figure 3.10) reports overall highest averaged performance, outperforming other color spaces with the second being the CIELAB color space. Finally, as in case of DS1, with DS2, we see that the color spaces improve more with the tree-based skin modeling (Random Forest and J48) than with other approaches.

3.2.3 Role of Illuminance Component (2D vs 3D)

In the following, we evaluate the effect of removing the illuminance component of a color space on skin detection performance.

For Dataset DS1

The F-measure values for 2D color spaces i.e. without the illuminance component are shown in Table 3.2. The difference of the F-measure for the 2D and 3D colors is shown in Table 3.5, reported for each color and classifier combination. The difference is computed as the 3D F-measure minus the 2D F-measure multiplied by 100. We find that in almost all cases the 3D color spaces perform better than 2D color spaces. There is one case (IHLS with AdaBoost, see Table 3.5) where the performance stays constant. There are 4 cases (negative values in Table 3.5) where the performance of the 3D color space is slightly less than 2D. There are 43 cases (positive values in Table 3.5) where the performance of the 3D

Table 3.3: F-measure for 3D color spaces with different skin color modeling approaches using DS2.

Classifier	IHLS	HSI	RGB	nRGB	YCbCr	CIELAB
AdaBoost	0.473	0.276	0.284	0.250	0.361	0.116
BayesianNet	0.578	0.541	0.321	0.548	0.439	0.613
J48	0.681	0.684	0.662	0.626	0.701	0.701
MLP	0.566	0.501	0.635	0.569	0.627	0.622
NaiveBayes	0.466	0.312	0.255	0.408	0.427	0.454
Random Forest	**0.745**	**0.741**	**0.724**	**0.705**	**0.722**	**0.762**
RBF	0.467	0.262	0.389	0.343	0.491	0.529
SVM	0.503	0.471	0.323	0.218	0.385	0.599

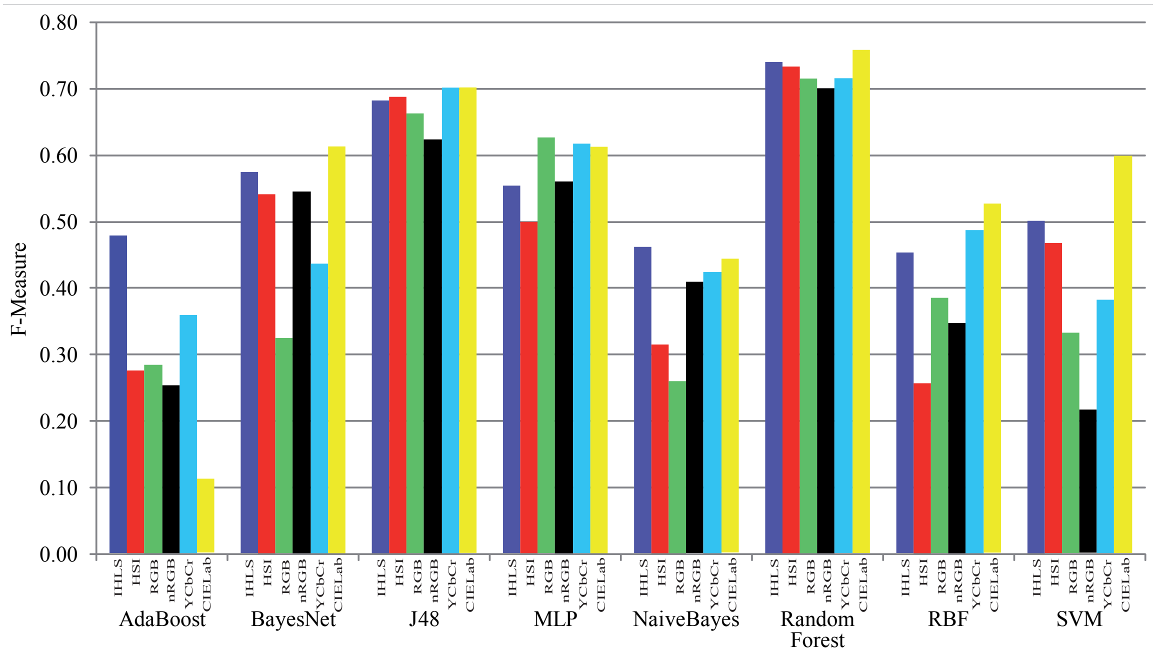


Figure 3.9: F-measure for skin color modeling approaches with 3D color spaces for DS2.

color space is higher than 2D. We represent significant improvement as a value greater than 10. There are a total of 11 cases (values with asterisk in Table 3.5) where the performance of 3D color space is significantly higher than their 2D counterpart. The difference of F-measure is also related to the color space transformation. We find (Table 3.5) that the effect of the illuminance component is more dominant in the case of CIELAB, YCbCr and HSI color spaces. We argue that the illuminance component adds more information for skin and non-skin separability in the case of CIELAB, YCbCr and HSI color spaces than the nRGB and IHLS color spaces.

For Dataset DS2

Table 3.4 shows F-measure values for 2D color spaces. The difference of the F-measure for the 2D and 3D colors is shown in Table 3.6, reported for each color and classifier

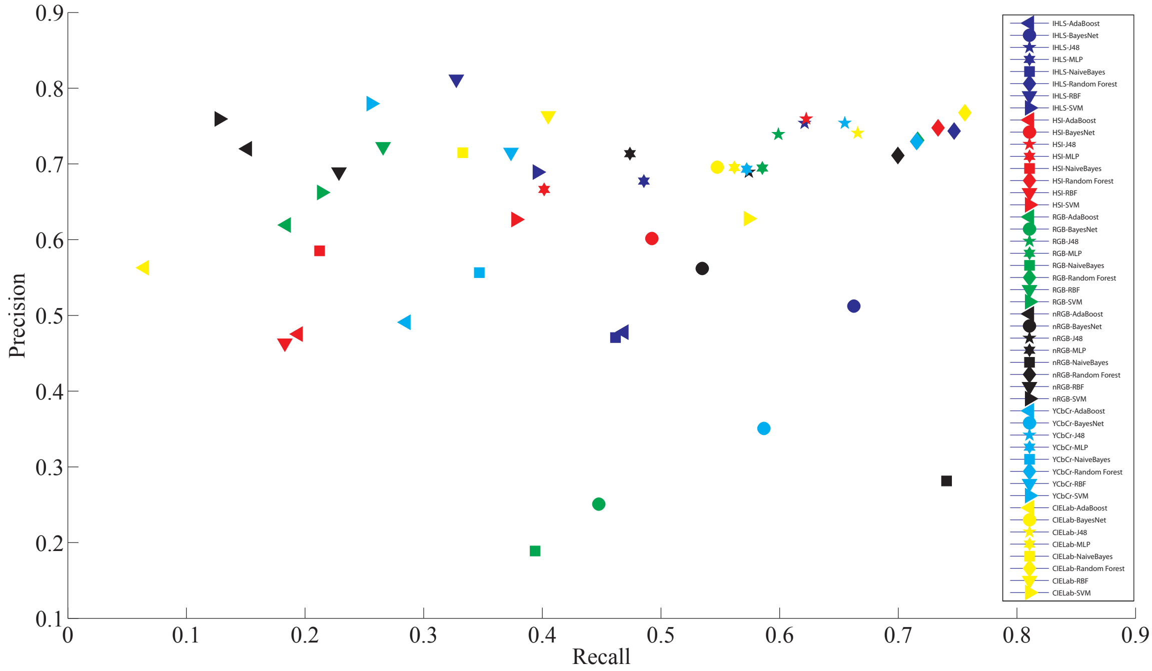


Figure 3.10: Precision and Recall space for skin-color modeling methods and 3D color spaces for DS2. Similar to DS1, Random Forest (diamond shaped) and J48 (five pointed stars) exhibit dominant clusters at the extreme top right and thus show overall good performance.

combination. The difference is computed as for DS1. There are 2 cases (negative values in Table 3.6) where the performance of the 3D color space is slightly less than 2D. There are 46 cases (positive values in Table 3.6) where the performance of 3D color space is higher than 2D. There are total of 16 cases (values with asterisk in Table 3.6) where the performance of 3D color space is significantly higher (greater than 10) than their 2D counterpart. We also find that the effect of the illuminance component is more dominant in the case of CIELAB, YCbCr and IHLS color spaces.

3.2.4 Role of Skin Color Modeling

How does skin-color modeling affect the skin detection performance? What is the best combination of color space and skin color modeling approach? For answers and the role investigation, we consider the 3D color spaces.

Regarding DS1, for the visual interpretation of results see Figures 3.7 and 3.8. Figure 3.8 is the precision and recall space for the combination of 3D color spaces and skin color modeling techniques. Any data values falling towards the top right corner are the good combinations. We see that the Random Forest (diamond shaped data points in Figure 3.8) has a well defined cluster at the extreme top right. For F-measure (see Figure 3.7), the Random Forest dominates other classifiers. Of most importance is the combination of Random Forest and IHLS color space outperforming all possible combinations. The second best combination with the Random Forest is that of HSI color space. The other

Table 3.4: F-measure for 2D color spaces (with out the illuminance component) for eight color modeling approaches based on DS2.

Classifier	IHLS	HSI	RGB	nRGB	YCbCr	CIELAB
AdaBoost	0.480	0.257	0.259	0.213	0.334	0.103
BayesNet	0.413	0.311	0.292	0.332	0.391	0.413
J48	0.540	0.588	0.228	0.593	0.560	0.567
MLP	**0.549**	0.481	**0.602**	0.527	**0.571**	0.534
NaiveBayes	0.284	0.220	0.292	0.389	0.227	0.423
Random Forest	0.531	**0.657**	0.415	**0.670**	0.539	**0.670**
RBF	0.418	0.186	0.351	0.327	0.333	0.501
SVM	0.462	0.438	0.312	0.204	0.227	0.418

Table 3.5: Difference of F-Measure (multiplied by 100) in the 3D color space compared to the 2D color space using DS1. An asterisk represents a significant performance difference (greater than 10) between the 3D and 2D color spaces.

Classifier	IHLS	HSI	RGB	nRGB	YCbCr	CIELAB
AdaBoost	0.00	10.04*	0.70	-0.90	1.76	26.20*
BayesNet	0.46	3.38	7.98	14.42*	2.08	5.16
J48	4.04	11.49*	17.94*	0.75	5.53	8.94
MLP	3.76	5.44	18.13*	2.72	2.44	12.05*
NaiveBayes	5.45	3.89	-0.03	-1.00	6.00	3.80
Random Forest	10.39*	6.73	15.84*	0.08	9.46	7.06
RBF	-3.45	8.26	0.24	5.08	20.76*	2.42
SVM	1.81	0.56	0.12	1.00	2.95	14.51*

dominant cluster (data points with five pointed stars in Figure 3.8) is exhibited by the J48 which belongs to the same class of tree based classifiers as the Random Forest. In the case of J48 the highest F-measure is reported by the IHLS and YCbCr color spaces.

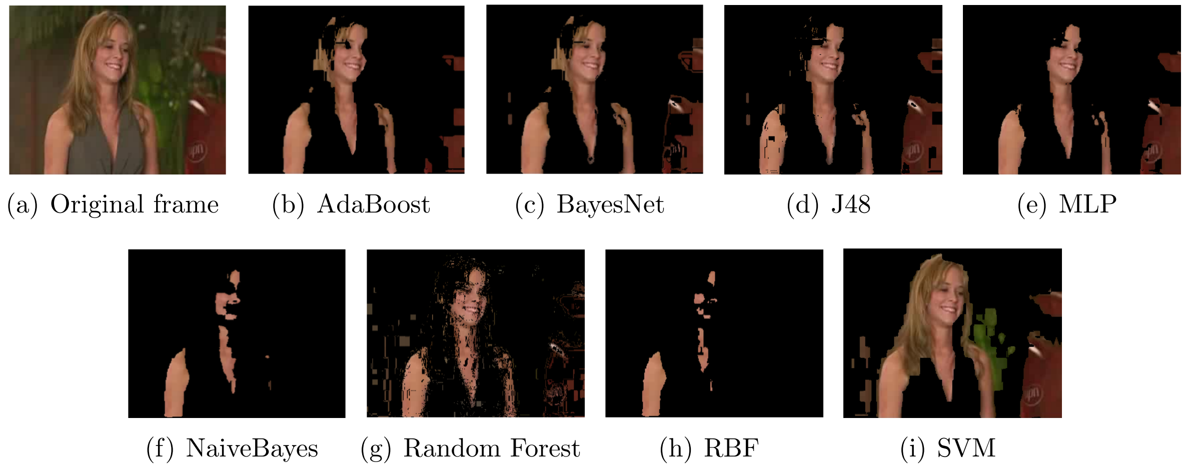
For dataset DS2, the visual interpretation of results can be seen in Figures 3.9 and 3.10. As in the case of DS1, for DS2, the Random Forest (diamond shaped data points in Figure 3.10) dominates other classifiers. Of most importance is the combination of Random Forest and CIELAB color space, outperforming all possible combinations. The second best combination with the Random Forest is that of the IHLS color space. Similar to DS1, the second dominant cluster (data points with five pointed stars in Figure 3.10) is exhibited by the J48 . In case of J48 the highest F-measure is reported by CIELAB and YCbCr color spaces.

For both DS1 and DS2, regarding AdaBoost and MLP, the combination with cylindrical color spaces like HSI and IHLS achieves state of the art results. Opposed to our expectations, even after an extensive parameter tuning, we were not able to achieve comparable results with SVM, although there is a significant boost in performance using cylindrical color spaces. Figures 3.11 and 3.12 show sample skin detection on an image from dataset DS1 and DS2 respectively, using eight skin-color modeling approaches in the IHLS color space.

From the eight skin color modeling approaches on two datasets DS1 and DS2, we find

Table 3.6: Difference of F-Measure (multiplied by 100) in the 3D color space compared to the 2D color space using DS2. An asterisk represents a significant performance difference (greater than 10) between the 3D and 2D color spaces.

Classifier	**IHLS**	**HSI**	**RGB**	**nRGB**	**YCbCr**	**CIELAB**
AdaBoost	-0.68	1.85	2.54	3.70	2.64	1.31
BayesNet	16.48*	23.08*	2.99	21.56*	4.78	20.00*
J48	14.12*	9.64	43.34*	3.35	14.14*	13.38*
MLP	1.64	2.03	3.34	4.24	5.61	8.73
NaiveBayes	18.17*	9.14	-3.70	1.89	20.00*	3.09
Random Forest	21.42*	8.32	30.83*	3.50	18.38*	9.14
RBF	4.92	7.58	3.79	1.64	15.77*	2.82
SVM	4.08	3.36	1.11	1.45	15.77*	18.11*



(a) Original frame (b) AdaBoost (c) BayesNet (d) J48 (e) MLP

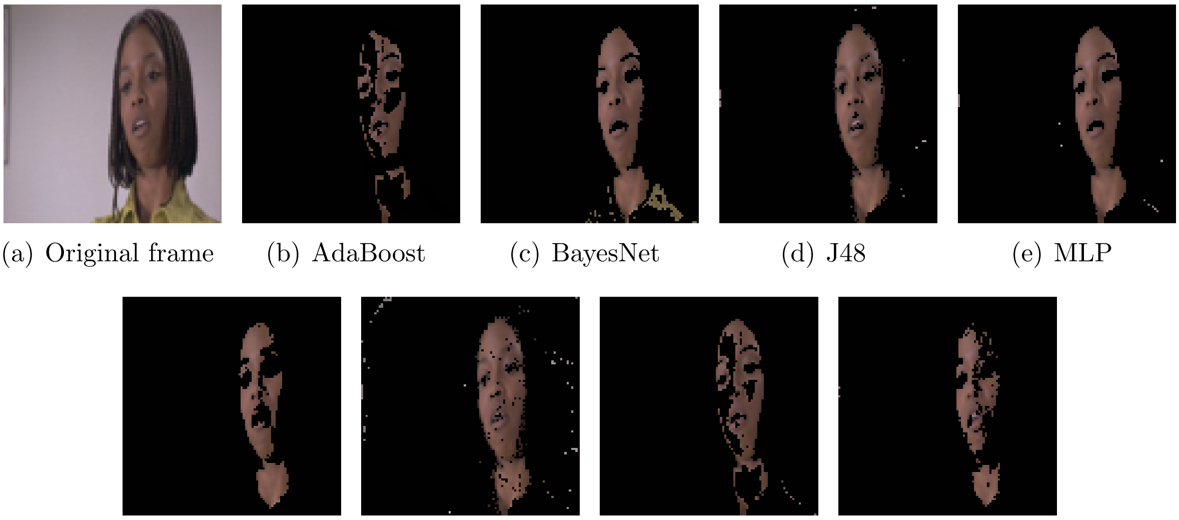
(f) NaiveBayes (g) Random Forest (h) RBF (i) SVM

Figure 3.11: Skin detection using different classifiers in the IHLS color space for DS1. Non-skin is shown as black.

that the choice of skin color modeling approach does make a significant difference for skin color modeling. Also, its effect on the performance is greater than the effect due to the presence or the absence of the illuminance component and that skin color modeling has a greater impact than the color space transformation. With the cylindrical color spaces, the tree based classifiers (Random forest and J48) outperform other combinations and are well suited to pixel based skin detection.

3.2.5 Effect of Lighting Correction

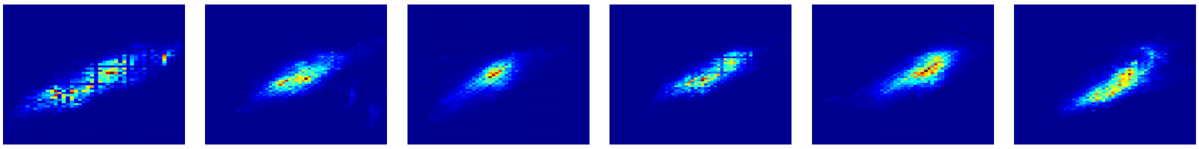
In this section, we show the effect of color constancy algorithms (Gray-Edge, Gray-World, max-RGB, Shades of Gray, Gray-Edge and Bayesian color constancy) for color based skin classification. For this purpose, we fix YCbCr as the color space because of its wide usage for skin detection [109] and Random forest as the classifier because of its overall increased classification performance.



(a) Original frame (b) AdaBoost (c) BayesNet (d) J48 (e) MLP

(f) NaiveBayes (g) Random Forest (h) RBF (i) SVM

Figure 3.12: Skin detection using different classifiers in the the IHLS color space for DS2. Non-skin is shown as black.



(a) Without Color constancy (b) Gray-Edge (c) Gray-World (d) max-RGB (e) Shades-of-Gray (f) Bayesian

Figure 3.13: Skin spread for color constancy using DS1.

Figure 3.13 displays the skin locus for DS1 in the YCbCr color space for different color constancy algorithms. It is a 2-dimensional histogram with Cb component on x-axis and the Cr component on Y-axis. Figure 3.13(a) shows skin locus without applying lighting correction. Figure 3.13(b-f) reports skin locus after applying Gray-Edge, Gray-World, max-RGB, Shades-of-Gray and Bayesian color constancy respectively. It can be seen that skin locus in Figure 3.13(b-f) is much smaller and compact when compared with Figure 3.13(a). We argue that this will increase classification when used with classifiers. Figure 3.14 shows Accuracy, Precision, Recall and F-measure for DS1 with and without lighting correction using the Random forest classifier. It can be seen that in every case Accuracy, Precision, Recall and F-measure is increased compared to the uncorrected scenario. Regarding the F-measure, compared to an F-measure of 0.70 (without lighting correction), we get an increased classification of 0.79 for Gray-Edge, 0.76 for Gray-World, 0.75 for max-RGB, 0.75 for Shades-of-Gray and a slight increase of 0.71 for Bayesian color constancy. This increase in performance is supported by the compact skin locus of Figure 3.13(b-f) for corresponding color constancy algorithms.

Figure 3.15 displays the skin locus for DS2 in the YCbCr color space. Figure 3.15(a)

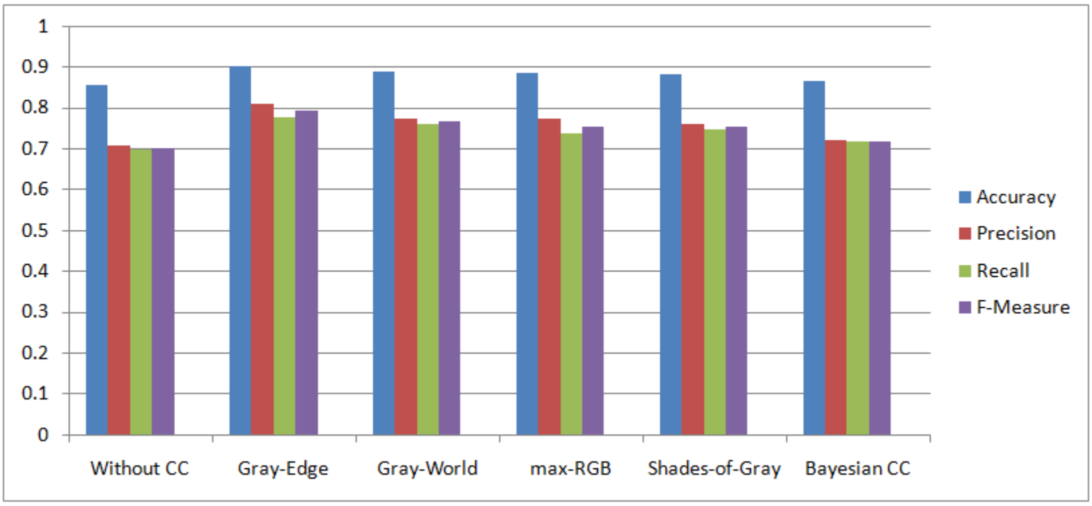


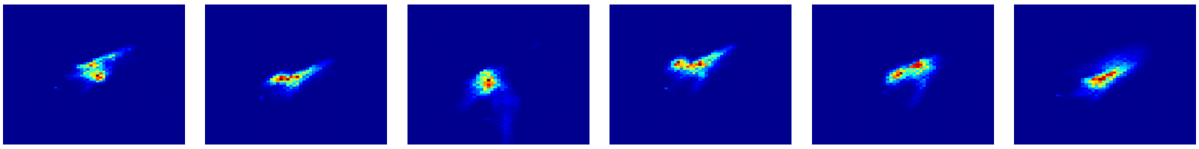
Figure 3.14: Results of the Random forest classifier for color constancy (CC) on DS1.

shows skin locus without applying lighting correction. Figure 3.15(b-f) reports skin locus after applying Gray-Edge, Gray-World, max-RGB, Shades-of-Gray and Bayesian color constancy respectively. As opposed to DS1, the skin locus in Figure 3.15(b-f) is much deviated and shifted though compact in some cases when compared with Figure 3.15(a). This deviation and shift has an effect on the overall classification results as well. Figure 3.16 shows Accuracy, Precision, Recall and F-measure for DS2 with and without lighting correction using the Random forest classifier. We find that the skin locus in Figure 3.16(b) for Gray-Edge is much different than the original one and rather appears to be a flip in chromaticity space. As a result, we get decreased classification with F-measure of 0.69 for the Gray-Edge algorithm. Figure 3.15(c) shows the Gray-World skin locus which is compact compared to the uncorrected case, reporting an increase in classification with F-measure of 0.74. Regarding max-RGB, the skin locus resembles that of Gray-Edge and thus we get almost identical results for Accuracy, Precision, Recall and F-measure. The Shades-of-Gray gives a compact locus with an increased performance, having F-measure of 0.78. Bayesian color constancy also reports an increase in performance with an F-measure of 0.76 compared to 0.72 (without lighting correction).

From the results, it can be concluded that the lighting correction for skin classification can improve performance. At the same time, lighting correction can have negative effect on the results. This is due to the fact that color constancy algorithms produce a compact representation of skin locus (skin color ranges in a color space) but the skin locus can also be shifted and deviated in the chromaticity space, resulting in varying performance.

3.3 Summary

In this chapter, we discussed skin detection using static skin filters and classifiers. As a simple and fast method, we introduced two new static filters based on two chrominance



(a) Without Color constancy (b) Gray-Edge (c) Gray-World (d) max-RGB (e) Shades-of-Gray (f) Bayesian

Figure 3.15: Skin spread for color constancy using DS2.

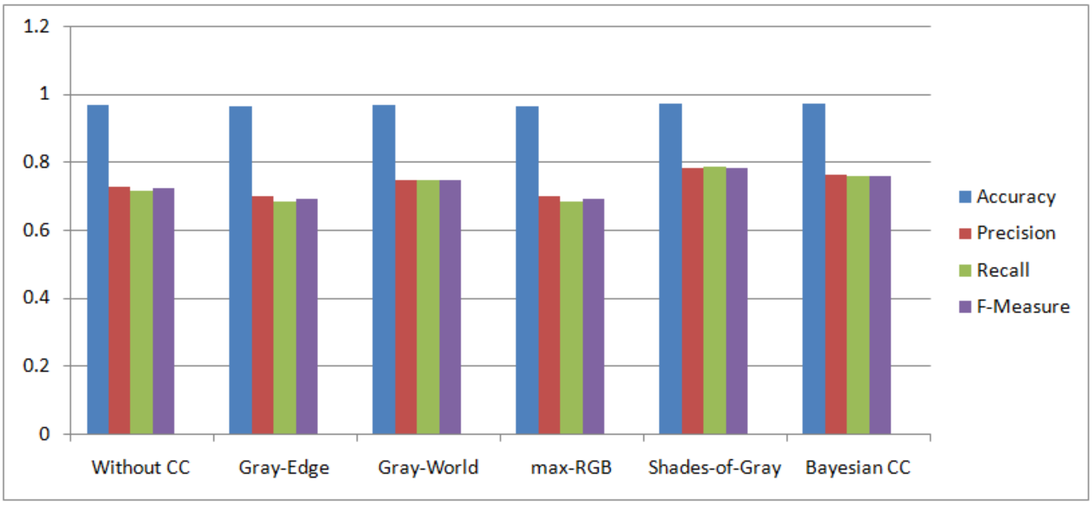


Figure 3.16: Results of the Random forest classifier for color constancy (CC) on DS2.

components in the IHLS and CIELAB color spaces. We showed that the skin detection performance can be improved with the usage of color constancy algorithms. We also investigated and evaluated (1) the effect of color space transformation on skin detection performance and finding the appropriate color space for skin detection, (2) the role of the illuminance component of a color space, (3) the appropriate pixel-based skin color modeling technique and (4) the effect of lighting correction for pixel based skin classification.

We presented a comprehensive evaluation on color space and skin color modeling techniques which will help in the selection of the best combinations for skin detection. We showed that the cylindrical color spaces outperform other color spaces and that the absence of the illuminance component decreases performance. From the eight skin color modeling approaches on two datasets DS1 and DS2, we found that the choice of skin color modeling approach does make a significant difference for skin color modeling. Also, its effect on the performance is greater than the effect due to the presence or the absence of the illuminance component and that skin color modeling has a greater impact than color space transformation. With the cylindrical color spaces, the tree based classifiers (Random forest and J48) outperform other combinations and are well suited to pixel based skin detection. Regarding lighting correction, since color constancy algorithms produce a compact representation of the skin locus, skin classification performance is improved. At the same time, shifting and deviation of the skin locus in chromaticity space will result in varying performance.

Chapter 4

Markowitz Model For Skin Detection

In this chapter, we introduce the mathematical financial model of Markowitz [68] and use it for color based skin detection. We linearly merge different color space channels, representing it as a "fusion" process for skin detection. The non-perfect correlation between the color spaces is exploited by learning weights based on an optimization for a particular color space channel. As such, we demonstrate the usage of fusion of color space channels for training based on positive data only and investigate the role of various color spaces (color channels) for color based skin detection.

4.1 Markowitz Learned Model

In financial markets, the assets in an investment portfolio cannot be selected individually, each on their own merits. It is important to consider how each asset changes price relative to how every other asset changes price in the portfolio. Since investment is a trade off between risk and return, the assets with higher return are riskier.

The Markowitz modern Portfolio Theory (MPT) is related to investment which tries to maximize return and minimize risk with the proportionate selection of different assets based on the mathematical formulation of the concept of diversification in investment. The aim of proportionate selective investment collection is to have lower risk than the individual assets. The MPT models the return of assets as normally distributed random variables. The risk is modeled as the standard deviation and the portfolio as the weighted combination of assets. The return portfolio is the weighted combination of the asset returns. With the combination of different assets whose returns are not correlated, the objective of MPT is to reduce the total variance of the portfolio. For the amount of risk, MPT describes how to select a portfolio with the highest possible return.

For the mathematical formulation of MPT, given sets of observations of the same quantity expressed in the same unit but their method of obtaining is different, with the only knowledge that the probability distribution of the observations is a unimodal function. How does one combine the output of these methods in order to obtain the most accurate measurement of the process? In general the observations can be represented as [99]:

$$u = \mu_u \pm \sigma_u \tag{4.1}$$

where μ is the mean value and σ is the fluctuation of the quantity u. With the mathematical model for efficient portfolio selection of Markowitz, in general, N different observations can be fused using the following weighting scheme:

$$\mu = \sum_{i=1}^{N} x_i \mu_i \tag{4.2}$$

where μ_i is the average return value of a particular method i and x_i is the weight assigned to that method. μ is the total return for all the quantities involved. For weight optimization, the constraints imposed are:

$$\sum_{i=1}^{N} x_i = 1 \tag{4.3}$$

$$-1 \leq x_i \leq 1, \quad \text{and} \quad i = 1, \ldots, N. \tag{4.4}$$

The Markowitz model finds the set of portfolios that provides minimum risk for all the possible returns. In general, the Markowitz model involves maximizing the expected return or minimizing the variance. According to Equation 4.2, the expected estimate of quantity μ from a large set of N quantities involved is the weighted sum of the expected estimate of the individual quantities involved. For computing the variance of the whole set of quantities, we need the covariances between the individual quantities as well. The variance for several combined observations is

$$V = \sum_{i=1}^{N} x_i^2 v_i + \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j c_{ij}, \quad \text{and} \quad i \neq j \tag{4.5}$$

$$c_{ij} = \sigma_i \sigma_j \rho_{ij}, \quad v_i = \sigma_i^2 \tag{4.6}$$

where v_i is the variance, ρ_{ij} is the correlation between the quantities involved and σ_i is the fluctuation or standard deviation. Covariance between the quantities i and j is represented by c_{ij}. The Markowitz model involves minimizing

$$\sigma = \sqrt{\sum_{i=1}^{N} x_i^2 \sigma_i^2 + \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j \rho_{ij} \sigma_i \sigma_j} \tag{4.7}$$

the standard deviation under the constraints given in Equations 4.3 and 4.4 for a given expected estimate or maximizing the expected estimate of a given standard deviation σ. Constraint (4.3) is for full allocation of the resources. The search space for the solution of Equation 4.7 is limited by the constraint of Equation 4.4. The objective function in Equation 4.7 is quadratic with linear constraints, solved by linear programming.

A Markowitz model weights different observations taking into account the non-perfect correlation between the observations involved and the individual performance. For different quantities involved, the efficient frontier (see Figure 4.1 and Figure 4.2) obtained with the Markowitz model provides mean-standard deviation pairs corresponding to different weighting pairs. At every point on the efficient frontier is the set of weights corresponding to the total number of quantities involved. The weights on the optimal frontier give the optimal combination of the quantities involved. The selection of the final set of weights from the infinite set of weights from the efficient frontier given by the Markowitz model is generally problem dependent. The optimal weights are those that have the highest signal to noise ratio or those providing the highest ratio between return and risk. The optimal set of weights, known as the *risky weights* can be obtained from the optimal frontier by maximizing the objective function W,

$$W = \frac{\mu}{\sigma} \tag{4.8}$$

where μ represents the mean and σ is the standard deviation for a particular set of weights generated.

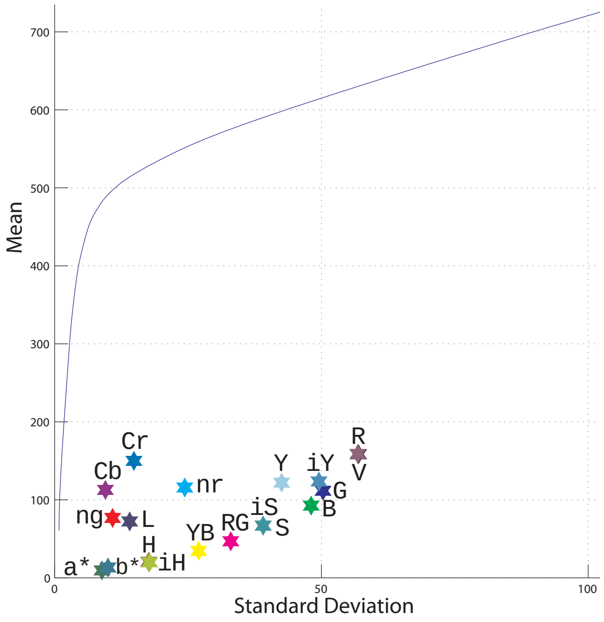


Figure 4.1: Mean-Standard deviation space and the corresponding placement of 19 color channels in this space for DS1.

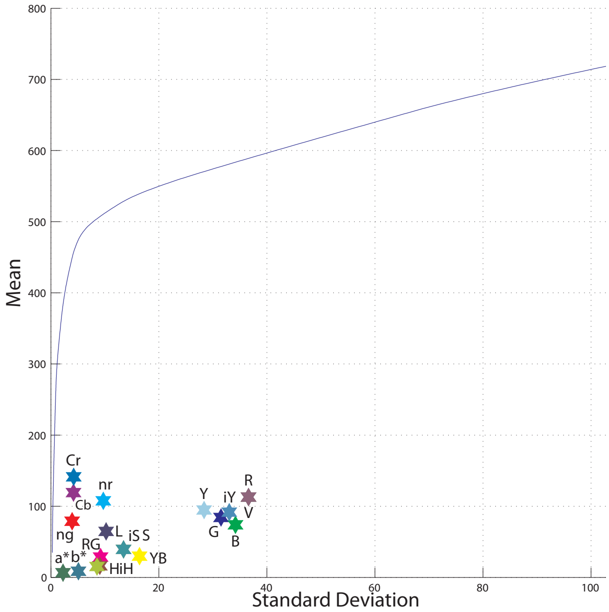


Figure 4.2: Mean-Standard deviation space and the corresponding placement of 19 color channels in this space for DS2.

4.2 Markowitz Model For Color

In this section, we apply the Markowitz model to color based skin detection and explain the fusion operation.

4.2.1 Color and MPT

For color based detection in different imaging conditions, the choice of a color space is essential as it induces equivalence classes for the detection algorithm [99, 98]. We aim to exploit fusion based relative integrative correlation of color spaces by the Markowitz model. For color based feature observations related to a particular object in different color spaces, the data thus combined should represent the same quantity with boosting and diminishing behavior due to the inherent properties of color spaces. For the skin detection scenario, in Equation 4.1, μ is the average value of positive skin samples and σ is the standard deviation for that set of samples in a particular color space channel. Regarding color based features for a training/testing sample, the return values in Equation 4.2 are the pixel values in a particular color space channel. For color based training/testing scenario, it is the aggregated value of different color space channels. For skin detection in images where the region of human skin color is defined, minimizing the standard deviation in Equation 4.7 will increase the concentration towards the trained color of skin samples in a color space.

The performance of the color feature detectors is based on repeatability and discriminative power. Repeatability is concerned with the invariant behavior under uncontrolled viewing conditions, such as varying illumination, shading and highlights. However, there is a trade-off between repeatability and distinctiveness. For the skin detection task which is subject to different viewing conditions, the selection of color models invariant to varying illumination should provide discriminative power for the skin segmentation algorithm. Therefore, to weight color channels for a proper balance between color invariance and discriminative power, the skin clustering space and correlation between the color channels has to be taken into account. For the skin detection problem where different color space channels are the quantities involved, the efficient frontier obtained with the Markowitz model provides mean-standard deviation pairs, representing different weightings for different color space channels.

The Markowitz model [68] based weighting scheme has been proposed in [99, 98] for color space channel selection and fusion based on the positive samples only. We extend [99], where 12 color channels were used to learn a model from a single image and the detection was shown on another, not generalizable to further images. We concentrate on post fusion classification rules for flagging a pixel as skin or non-skin by learning a model on representative skin samples. From the linearly fused data, we propose an indicator image which is used in a decision threshold for flagging a pixel as skin or non-skin. For the decision threshold, we use the model obtained from the training data. For performance tuning, we introduce parameter K. In total, seven color spaces (19 color channels) are weighted for their role in the skin detection scenario, opting for a complete skin detection system where performance can be tuned for applications.

4.2.2 Algorithm: Color Space Channel Fusion

For the training part of the fusion of multiple color channels, the positive training data (RGB) is transformed into (1) Normalized red nr and normalized green ng channels of the normalized RGB color space (2) Red-green RG and yellow-blue YB channels of the opponent color space (3) L, a^* and b^* channels of the CIELAB color space (4) Y, Cb and Cr channels of the YCbCr color space (5) H, S and V channels of HSV color space and (6) iH, iS and iY channels of IHLS color space. Thus, including R, G and B channels of the RGB color space, 19 color channels are processed. For training on the skin samples, following steps are performed.

1. For all the color channels of the training samples consisting of only skin pixels, the mean is estimated.

2. The standard deviation is computed.

3. The correlation between the color channels is calculated.

4. The standard deviation and correlation is used to calculate the covariance between the color channels.

5. Using the obtained results to select the optimal weightings of the color planes using the Markowitz model. We denote as w_i where $i = 1 \dots 19$, the weights obtained through the model for 19 color channels of six color spaces. The obtained weights are multiplied with the respective color channel of the training data per pixel and all the channels are added.

   $$tc = \sum_{i=1}^{N} w_i c_i \tag{4.9}$$

   where tc is the aggregated data of the training samples, c_i is the respective color channel and $N = 19$. The mean E_{train} and standard deviation σ_{train} of the aggregated training data tc are calculated per pixel.

For the test image for skin detection:

1. Convert the test image to all the described color planes.

2. The appropriate weights obtained from the training phase are multiplied with the color channels and all the color channels are added to get a gray value image.

   $$g(x, y) = \sum_{i=1}^{N} w_i \bar{c}_i \tag{4.10}$$

   where $g(x, y)$ is the gray value image, (x, y) are the image coordinates, $\bar{c}$ is the color channel and w_i is the weight for that color channel. N represents the total number of color channels which is 19.

3. From the gray value image $g(x, y)$, the indicator image is obtained by:

   $$I_{nd}(x, y) = |g(x, y) - E_{train}| \tag{4.11}$$

   where I_{nd} is the indicator image, (x, y) are the image coordinates and E_{train} is the mean of aggregated data obtained from training phase. For the indicator images, if the pixel value is closer to zero, it will correspond more to being a skin pixel. With reference to E_{train}, a value closer to this value and classified as skin will have higher probability of actually being skin.

4. The pixel values in the image are labeled as skin or non-skin from the indicator image according to the following decision rule:

$$sn(x,y) = \begin{cases} s(1) & \text{if } I_{nd}(x,y) < (\sigma_{train} + K) \\ ns(0) & \text{Otherwise} \end{cases} \tag{4.12}$$

   where $sn(x,y)$ is the binary image representing skin/non-skin, with pixel values set to 1 for skin and 0 for non-skin and I_{nd} is the indicator image. The skin detection performance (precision and recall) is controlled through the parameter K. A value of K greater than 0 expands the skin decision boundary, while a value of K less than 0 produces tighter bounds for skin decision. The optimal value of K can be experimentally determined, as shown in Section 4.2.6.

4.2.3 Color Space Channels Correlation

We use separate training sets for DS1 and DS2 in order to study the weight distribution for skin representations in varying imaging conditions. For the fusion of color space channels, Tables 4.1 and 4.2 show correlation coefficients between different color space channels for the training sets of DS1 and DS2 respectively. Negative correlation is exhibited for example by the red channel of the RGB in relation to the normalized red channel, H of HSV, Cb of YCbCr and iH channel of the IHLS color space for both DS1 and DS2 training data. Strong positive correlation is exhibited by Cr of YCbCr and RG of the opponent color space. Similarly, V of HSV and R of the RGB color space are positively correlated in both the training datasets.

4.2.4 Incremental Training and Weights for DS1

Table 4.3 shows weights obtained for a particular size of data. The first row shows weights for 321000 pixels and as we increase the size of data, the weights are slightly varied in the next rows. The last row of the Table 4.3 shows weights for the total representative training data. The weights in the last row do not variate a lot with reference to the first row of the table. If completely new training data is introduced which covers skin samples in different lighting conditions, the weights distribution will be affected. In our experimental setup, the weights in the last row of Table 4.3 are used for experiments and for skin detection examples regarding dataset DS1.

4.2.5 Mean-Standard Deviation Space and Weights

The optimal frontier calculated from the training sets is illustrated in Figures 4.1 and 4.2 for DS1 and DS2 respectively. Figures 4.1 and 4.2 show that both Cb and Cr have low standard deviation and higher mean in comparison to other color channels and are therefore assigned higher positive weights in Tables 4.3 and 4.4 for DS1 and DS2 respectively. This corresponds to the literature [112, 46] about the successful skin detection using Cb and Cr. We also demonstrate that these two color channels have the major effect on skin detection performance and thus high positive weights. The R and G channels of the RGB color space have high standard deviation in the mean-standard deviation space and are highly correlated. Therefore, for both DS1 and DS2, the weights assigned to these color channels are close to zero and their overall effect is reduced. This diminishes especially the role of the R channel for skin detection. The nr channel gets a positive weight of 0.35 for DS1 and 0.06 for DS2. However, the effect of ng channel is reduced by giving it a lower weight for both DS1 and DS2. The effect of RG channel of the opponent color space is boosted by assigning it a higher negative weight compared to the YB channel. Channels B and a^* achieve negative weights for both DS1 and DS2. For DS1, b^* has negative weight while for DS2, it is close to zero. For both DS1 and DS2, in Figures 4.1 and 4.2, H, S, V of HSV and iH, iS, iY of IHLS reside very close to each other in the Mean-Standard deviation space and therefore their weights are almost identical. S and iS for both DS1 and DS2 have perfect correlation of 1 in Table 4.1 and 4.2 and placed at exactly the same place in Figures 4.1 and 4.2.

Tables 4.3 and 4.4 reveal similarities between the weights for DS1 and DS2. For both datasets, the effect of R, G and B channels is reduced. RG has high negative weights for both datasets. Cb and Cr achieve the maximum positive weight for both datasets. The color channels ng, a^*, H, S, V, Y, iS and iY get almost identical weights for these datasets.

4.2.6 Performance Parameter

E_{train} in Equation 4.11 obtained from the training data for DS1 is 275.93, for DS2 is 196.01. σ_{train} in Equation 4.12 for DS1 is 18.41 and for DS2 is 1.29. The skin detection performance (precision and recall) is controlled through the parameter K in Equation 4.12. As shown in Figure 4.3 for DS1 and Figure 4.4 for DS2, starting from the negative value ($K = -15$), the performance increases. The maximum F-Score is achieved using $K = 2$ for DS1 and $K = -1$ for DS2. Any increase beyond these values of K decreases performance. All the experimental evaluation and skin detection examples are based on these values of K for DS1 and DS2.

4.2.7 Skin Detection in Images

Figure 4.5 shows successful skin detection examples using the proposed approach. The first column shows the original images, the second column shows the indicator images. For easy visualization the grey levels of the indicator images are inverted. As can be seen the skin portion is prominent in these indicator images compared to other regions. These prominent areas represented are closer to E_{train} and more probable to be classified as skin. The third column shows the skin detected images based on thresholding in Equation 4.12.

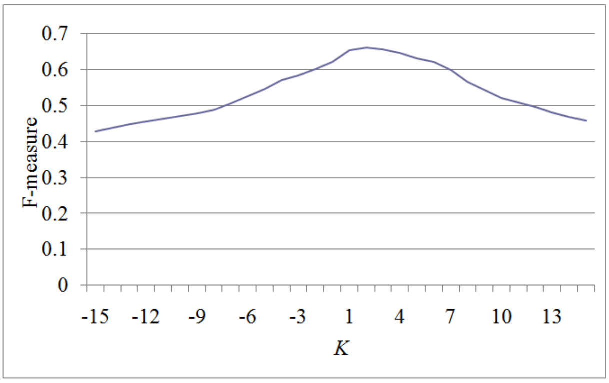


Figure 4.3: Skin detection performance is controlled through parameter K. $K = 2$ is used for all the comparative experiments regarding DS1.

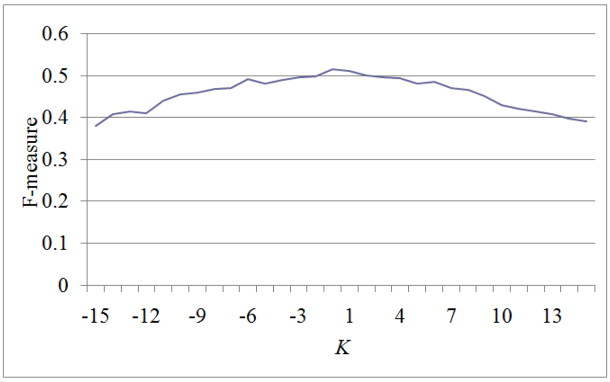


Figure 4.4: K as a parameter controls the skin detection peformance. $K = -1$ is used for all the comparative experiments regarding DS2.

Figure 4.6 reports cases where skin detection fails or non-skin pixels are reported as skin pixels. The fusion technique is trained only on positive images and therefore the non-skin pixels are detected as skin pixels in these images.

4.2.8 Performance Evaluation

For the fusion of color space approach, the evaluation is based on F-measure and specificity for datasets DS1 and DS2 on per image and per pixel. For dataset DS1, we use the three sets which are skin-only, non-skin and hybrid (Section 1.5.1). Since the proposed approach is based on positive training data, the specificity is considered as an evaluation measure to see the performance for non-skin pixels.

For performance comparison, we select classifiers (AdaBoost, BayesNet, NaiveBayes, RBF network and J48) that use positive and negative training data (Section 3.1). Following [79], boosting is the optimal detection method for the detection of skin color and faces. We select the Bayesian network [90] and neural network [81] based classifiers based on the reported best performance in [51]. Based on the independent feature model, we select the NaiveBayes. J48 is selected based on the superior performance in [56] for tree based classifier (Section 3.2). Since, the proposed technique is based on training on positive data only, we need skin samples. We aim to evaluate the approaches on unseen data. For training data, we choose (198) images from the Internet with a great variety of skin colors and lighting conditions. 118 images are used for DS1 and 80 are used for DS2. We use two different sets to study the weight distribution for skin representations in varying imaging conditions. For DS1, the total number of skin pixels (positive training data) is 2,113,703, the number of negative training samples is 6,948,697. For DS2, the total number of skin pixels is 437,202 and the number of negative training samples is 2,264,559. For all the classifiers, the 19 color channels are used as the feature vectors. In the following, we list the performance evaluation on the two datasets.

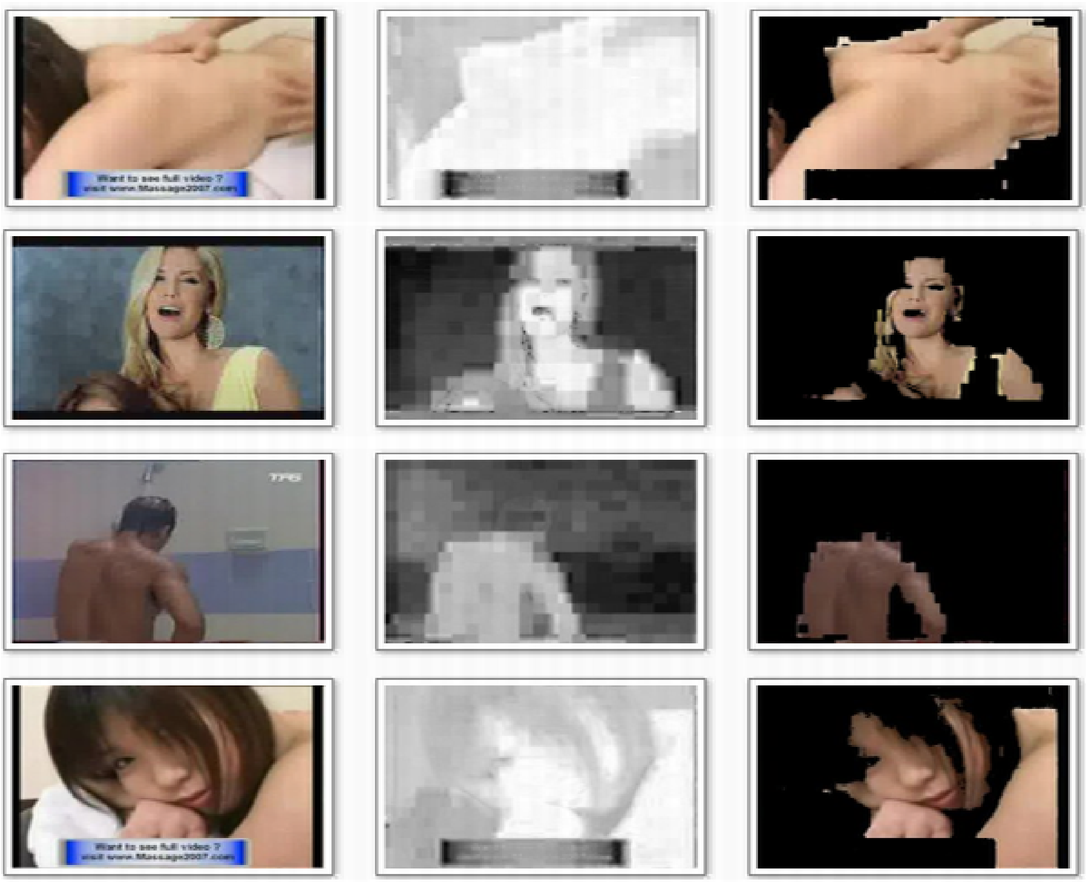


Figure 4.5: Skin detection based on weighting of color space channels. First column: original images. Second column: the indicator images. For easy visualization the indicator images are shown with grey levels inverted. Third column: skin detected images (black shows non-skin).

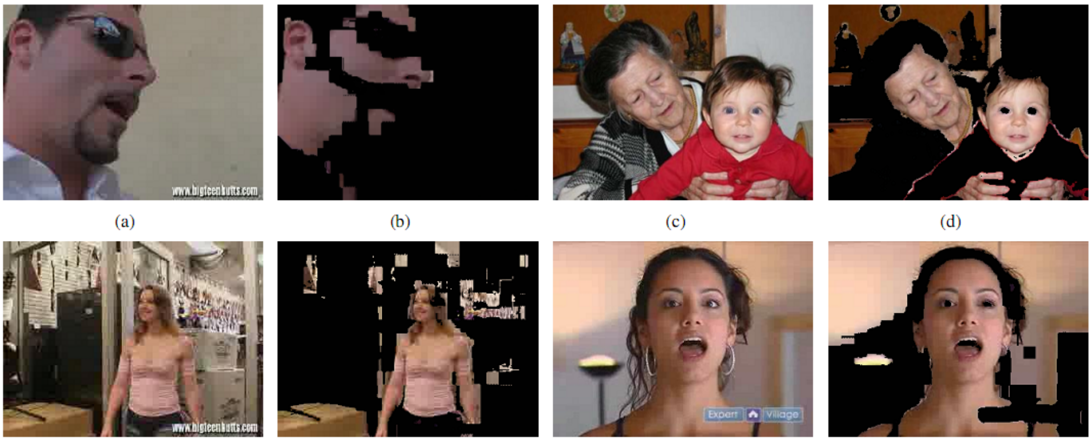


Figure 4.6: Skin detection scenarios (for fusion approach) where either skin is not properly detected or non-skin pixels are reported as skin.

For Dataset DS1 Skin-only

Figure 4.7 shows the F-measure and specificity per pixel for all the six approaches on the skin-only set. Mean and standard deviation for skin-only data with respect to specificity per image is displayed in Figure 4.8 and in Figure 4.9 with respect to F-measure per image. In Figure 4.7, high F-measure is reported for the fusion technique compared to AdaBoost, BayesNet, NaiveBayes, RBF and J48. The specificity in Figure 4.7 is not as high as AdaBoost, RBF network and J48, though it is greater than that of NaiveBayes and BayesNet. For the per image results, Figure 4.8 shows that the specificity mean of the fusion approach is lower than that of AdaBoost, RBF network and J48. In terms of increasing/decreasing trend in Figure 4.8 for skin-only specificity, the Fusion approach is almost identical to the BayesNet and NaiveBayes approaches. For the F-measure in Figure 4.9 for skin-only data, the fusion technique has higher mean (0.54) with a standard deviation of 0.19.

For Dataset DS1 Non-skin

If a skin detection system detects skin when applied to images containing skin, then it should detect nothing when applied to images containing no skin. We compare the six approaches based on specificity on the DS1 non-skin set. Figure 4.10 shows the specificity of the six techniques for non-skin images calculated per pixel. The specificity of the fusion technique (0.76) is higher than BayesNet (0.74) and less than NaiveBayes (0.78), RBF network (0.95), AdaBoost (0.88) and J48 (0.90). The comparably low specificity is due to the fact that the fusion technique is trained only on positive samples. Interestingly, the RBF network, which performs worse on skin-only images, has a high true negative rate for the non-skin images. Mean and standard deviation (per image) for the non-skin set with respect to specificity is displayed in Figure 4.8. Similar to the skin-only data, the specificity mean (0.68) of the fusion approach is lower than that of AdaBoost, RBF network and J48. The specificity mean of the NaiveBayes and the fusion approach is the same (0.77), with standard deviation of 0.32 for NaiveBayes and 0.24 for the fusion approach.

For Dataset DS1 Hybrid

For the hybrid set of images, we use the F-measure and specificity as the evaluation measure. Figure 4.11 shows F-measure and specificity calculated per pixel. Figure 4.8 reports mean and standard deviation for specificity and Figure 4.9 for F-measure on the hybrid data calculated per image. In Figure 4.11, it can be seen that the specificity of the fusion approach is higher than Naive Bayesian and Bayesian network approach and less than AdaBoost, RBF network and J48. Similarly, mean and standard deviation for specificity in Figure 4.8 shows that the specificity mean of fusion approach is lower than that of AdaBoost, RBF network and J48. Regarding precision and recall, Figure 4.11 shows that the fusion technique has higher F-measure having a higher mean of 0.37 (Figure 4.9) with a standard deviation of 0.26. The fusion technique provides increased classification performance of almost 4% to AdaBoost, 3.8% to Bayesian network, 18%

to Naive Bayesian and 26% to RBF network and decreased performance of almost 4% compared to J48. For combined skin-only and non-skin images the fusion technique outperforms other approaches in terms of precision and recall with the exception of J48.

For Dataset DS2

In dataset DS2, every image contains some skin, therefore, we use the F-measure and specificity as the evaluation measures. Figure 4.12 shows F-measure and specificity calculated per pixel. Figure 4.13 reports mean and standard deviation for specificity and F-measure calculated per image. In Figure 4.12, it can be seen that the specificity of the fusion approach is lower than all other approaches. Mean and standard deviation for specificity in Figure 4.13 show that the specificity mean of the fusion approach is also lower than AdaBoost, BayesNet, NaiveBayes, RBF network and J48. Regarding precision and recall, Figure 4.12 shows that the fusion technique has higher F-measure than AdaBoost, Bayesian network, and RBF and lower than NaiveBayes and J48. In Figure 4.13, the F-measure mean of DS2 is higher than all other approaches except J48. The fusion technique provides increased classification performance of almost 6% to AdaBoost, 1% to Bayesian network and 10% to RBF network and decreased performance of almost 3.3% compared to NaiveBayes and 15% to J48.

4.3 Summary

In this chapter, we introduced the mathematical financial model of Markowitz and applied it to color based skin detection. We exploited the non-perfect correlation between the color spaces by learning weights based on an optimization for a particular color space channel using the financial model of Markowitz. As such, (1) we demonstrated the usage of Markowitz model for training based on positive data only, and (2) we studied the role of different color spaces (color channels) for color based skin detection. We conclude that this approach can be extended to any color based object detection. With the higher weights for Cb and Cr color channels, we demonstrated that the proposed approach supports the empirical results that Cb and Cr are the preferred color channels for skin detection. We showed that even using less data for training, the proposed approach achieves high true detection rate compared to other approaches.

The fusion of color space channels approach can be efficiently used to learn weights for the contextual based skin detection based on faces. The positive data for training can be obtained from the face area returned by the face detector. The calculation of weights takes about 6 milliseconds in Matlab. The time consuming operation is the color space conversion. For real-time either the number of color spaces could be reduced or with a binary implementation of the color space conversion, real-time learning of weights and thereby skin detection can be achieved.

Table 4.1: Correlation coefficients for 19 color channels using training data for DS1.

	R	G	B	nr	ng	RG	YB	L	a^*	b^*	H	S	V	Y	Cb	Cr	iH	iS	iY
R	1																		
G	0.82	1																	
B	0.72	0.95	1																
nr	-0.11	-0.59	-0.67	1															
ng	0.09	0.54	0.48	-0.84	1														
RG	0.48	-0.11	-0.21	0.69	-0.67	1													
YB	0.49	0.88	0.95	-0.81	0.63	-0.49	1												
L	0.89	0.96	0.91	-0.52	0.46	0.06	0.79	1											
a^*	0.08	-0.47	-0.46	0.87	-0.95	0.85	-0.67	-0.34	1										
b^*	0.04	-0.34	-0.56	0.84	-0.44	0.59	-0.68	-0.28	0.56	1									
H	-0.09	0.17	0.14	-0.33	0.40	-0.40	0.25	0.08	-0.43	-0.14	1								
S	0.56	0.03	-0.16	0.63	-0.46	0.92	-0.42	0.18	0.67	0.70	-0.19	1							
V	1.00	0.82	0.73	-0.13	0.09	0.47	0.50	0.89	0.07	0.03	-0.05	0.56	1						
Y	0.91	0.98	0.92	-0.46	0.4	0.07	0.8	0.98	-0.3	-0.25	0.19	0.09	0.91	1					
Cb	-0.55	-0.16	0.12	-0.49	0.17	-0.72	0.33	-0.26	-0.38	-0.76	0.14	-0.89	-0.54	-0.27	1				
Cr	0.50	-0.09	-0.21	0.69	-0.63	1.0	-0.49	0.08	0.82	0.63	-0.39	0.94	0.49	0.1	-0.76	1			
iH	-0.08	0.17	0.14	-0.33	0.4	-0.4	0.25	0.09	-0.43	-0.13	0.99	-0.19	-0.05	0.09	0.13	-0.39	1		
iS	0.56	0.03	-0.16	0.63	-0.46	0.92	-0.42	0.18	0.67	0.7	-0.19	1.0	0.56	0.19	-0.89	0.94	-0.19	1	
iY	0.91	0.98	0.92	-0.46	0.4	0.07	0.8	0.98	-0.3	-0.25	0.09	0.19	0.91	1.0	-0.27	0.1	0.09	0.19	1

Table 4.2: Correlation coefficients for 19 color channels using training data for DS2.

	R	G	B	nr	ng	RG	YB	L	a^*	b^*	H	S	V	Y	Cb	Cr	iH	iS	iY
R	1																		
G	0.97	1																	
B	0.93	0.97	1																
nr	-0.38	-0.53	-0.63	1															
ng	0.03	0.12	0.02	0.03	1														
RG	0.64	0.45	0.36	0.31	-0.29	1													
YB	0.88	0.95	0.99	-0.70	0.07	0.23	1												
L	0.97	0.97	0.94	-0.45	0.16	0.53	0.90	1											
a^*	0.18	-0.01	0.03	0.45	-0.66	0.74	-0.08	0.07	1										
b^*	-0.33	-0.44	-0.63	0.84	0.20	0.21	-0.68	-0.41	0.06	1									
H	-0.39	-0.36	-0.49	0.22	0.26	-0.31	-0.47	-0.43	-0.51	-0.61	1								
S	0.39	0.21	0.01	0.56	0.02	0.83	-0.10	0.29	0.43	0.66	0.19	1							
V	1.00	0.97	0.93	-0.38	0.03	0.64	0.88	0.97	0.18	-0.33	-0.39	0.39	1						
Y	0.99	1.00	0.97	-0.49	0.08	0.51	0.94	0.98	0.06	-0.43	-0.39	0.25	0.99	1					
Cb	-0.11	0.03	0.26	-0.62	-0.23	-0.53	0.34	-0.03	-0.12	-0.86	-0.47	-0.91	-0.11	-0.01	1				
Cr	0.60	0.40	0.28	0.38	-0.22	0.99	0.16	0.49	0.68	0.32	-0.20	0.89	0.60	0.46	-0.64	1			
iH	-0.38	-0.36	-0.51	0.38	0.56	-0.29	-0.49	-0.38	-0.52	0.68	0.90	0.24	-0.38	-0.39	-0.53	-0.18	1		
iS	0.39	0.21	0.01	0.56	0.02	0.83	-0.10	0.29	0.43	0.66	0.19	1.0	0.39	0.25	-0.91	0.89	0.24	1	
iY	0.99	1.00	0.97	-0.49	0.08	0.51	0.94	0.98	0.06	-0.43	-0.39	0.25	0.99	1.0	0.01	0.46	-0.39	0.25	1

Table 4.3: Weights obtained for the color space channels corresponding to different sizes in number of pixels of the training samples for DS1. The training size is the number of pixels of the training data shown incrementally. The weights shown in the last row are used for the experiments and the skin detection examples.

R	G	B	nr	ng	RG	YB	L	a^*	b^*	H	S	V	Y	Cb	Cr	iH	iS	iY	Training Size
0.00	0.04	-0.06	0.35	0.30	-0.66	-0.09	0.09	-0.25	-0.44	0.00	0.01	-0.01	0.03	0.68	0.99	0.00	0.01	0.02	321000
0.01	0.07	-0.09	0.33	0.27	-0.61	-0.13	0.06	-0.32	-0.47	0.00	0.03	-0.02	0.05	0.75	1.00	0.00	0.03	0.03	642000
0.01	0.06	-0.08	0.40	0.33	-0.71	-0.13	0.10	-0.30	-0.55	0.00	0.02	-0.02	0.05	0.76	1.00	0.00	0.02	0.03	963000
0.02	0.05	-0.07	0.38	0.30	-0.74	-0.13	0.10	-0.31	-0.50	0.00	0.03	-0.01	0.04	0.79	1.00	0.00	0.03	0.03	1284000
0.02	0.05	-0.07	0.40	0.30	-0.75	-0.14	0.12	-0.31	-0.51	0.00	0.03	-0.01	0.04	0.78	1.00	0.00	0.03	0.03	1605000
0.01	0.08	-0.11	0.35	0.08	-0.98	-0.19	0.16	-0.26	-0.29	0.02	0.02	-0.01	0.06	1.00	1.00	0.01	0.02	0.04	2113703

Table 4.4: Weights obtained for different color space channels corresponding to total training samples for DS2.

R	G	B	nr	ng	RG	YB	L	a^*	b^*	H	S	V	Y	Cb	Cr	iH	iS	iY	Training Size
-0.01	0.01	-0.02	0.06	0.05	-0.42	-0.03	0.05	-0.25	0.03	0.07	0.02	-0.01	0.01	0.59	0.89	-0.07	0.02	0.02	437202

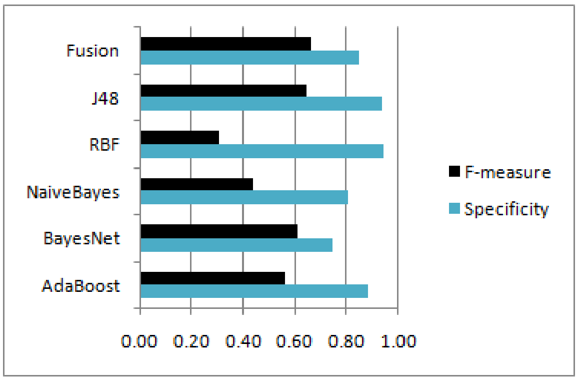


Figure 4.7: Fusion: F-measure and specificity for the DS1 skin-only set. The fusion of color space channels has higher precision and recall and thus high F-measure, outperforming other approaches which require negative and positive training data. The values reported are calculated per pixel.

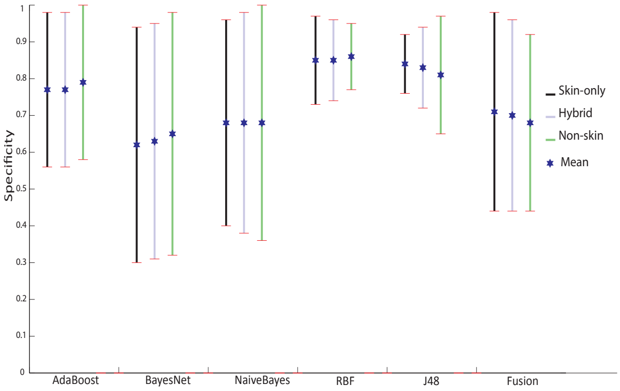


Figure 4.8: Fusion: Specificity mean and standard deviation for DS1 skin-only, hybrid and non-skin sets. The values reported are per image.

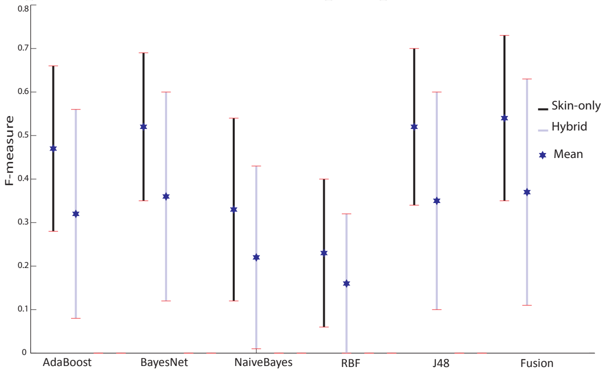


Figure 4.9: Fusion: F-measure mean and standard deviation for DS1 skin-only and hybrid sets. The values reported are per image.

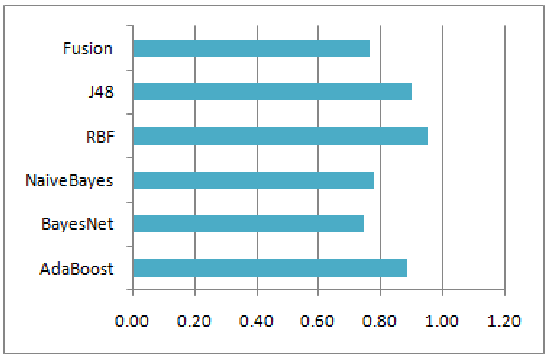


Figure 4.10: Fusion: Specificity for the non-skin images from DS1. Since the fusion technique is trained on positive data only, the true negative rate is not as high as the F-measure given in Figure 4.7.

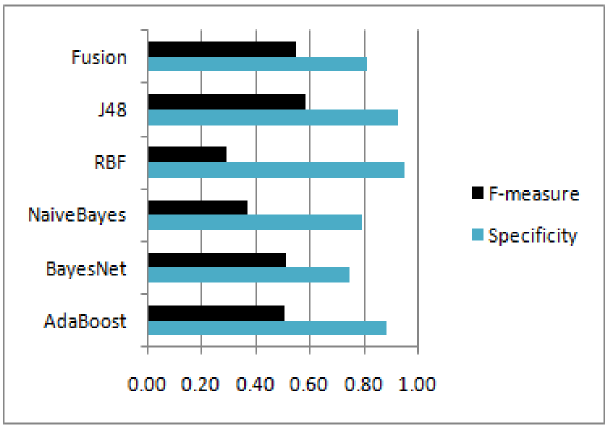


Figure 4.11: Fusion: F-measure and specificity for the DS1 hybrid images. The fusion technique outperforms other approaches in terms of precision and recall with the exception of the tree based classifier (J48). The values are calculated per pixel.

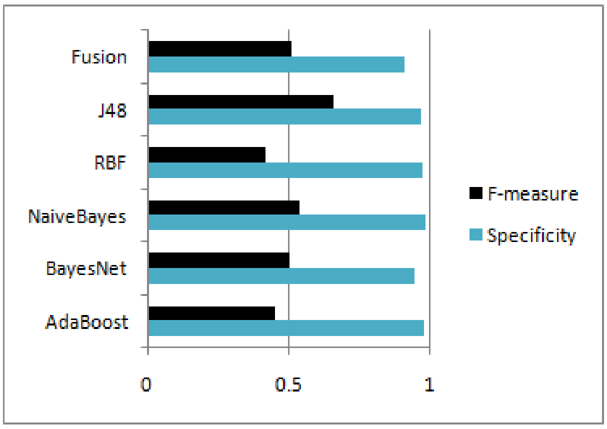


Figure 4.12: Fusion: F-measure and specificity for DS2. The fusion technique for DS2 outperforms other approaches in terms of precision and recall with the exception of the tree based classifier (J48) and NaiveBayes. The values are calculated per pixel.

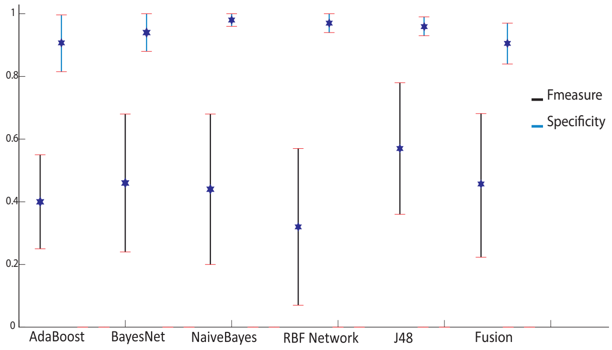


Figure 4.13: Fusion: Mean and standard deviation of specificity and F-measure for dataset DS2. The values reported are per image.

Chapter 5

Seed Based Approaches

In this chapter, for training based on positive data only, we present a novel idea of the universal seed. With the universal seed, we propose a concept for processing arbitrary images; to overcome the potential lack of successful seed detections, thereby providing basis for general skin segmentation. It exploits the spatial relationship among the neighboring skin pixels, providing more accurate and stable skin blobs. For taking advantage of the contextual information for skin detection, we present skin detection based on local seeds (face detection). Local seed based skin segmentation can be improved with seed filtering. Such contextual information is useful for updating the skin model for different skin colors and varying illumination conditions. We introduce a systematic approach for skin segmentation with graph cuts by using local skin information (local seeds from the image), universal skin information (universal seed) and skin augmentation using off-line learned model. For the systematic approach, when no local seeds are available, an external model is integrated into the existing setup of the universal seed approach for robust skin detection.

5.1 Universal Seed Approach

For general skin detection based on *graph cuts*, we introduce the universal seed concept. The universal seed based skin segmentation process can be described through a block diagram, shown in Figure 5.1. Two types of weights have to be adjusted; the neighborhood weights and the background/foreground weights. A graph is constructed given the proper adjustment of the neighborhood weights and background/foreground weights. A graph cut technique of [9] is used to segment the skin.

5.1.1 Graphs and Graph Cut

A directed weighted graph $G = (V, E)$ is made up of a set of nodes V and a set of directed edges E that connect them [9]. The nodes can represent pixels or voxels. A graph can contain some additional nodes called terminals which can correspond to the set of labels that can be assigned to pixels. We deal with graphs having two terminals. In the case of two terminals, the terminals are usually called the source, s, and the sink, t. In Figure

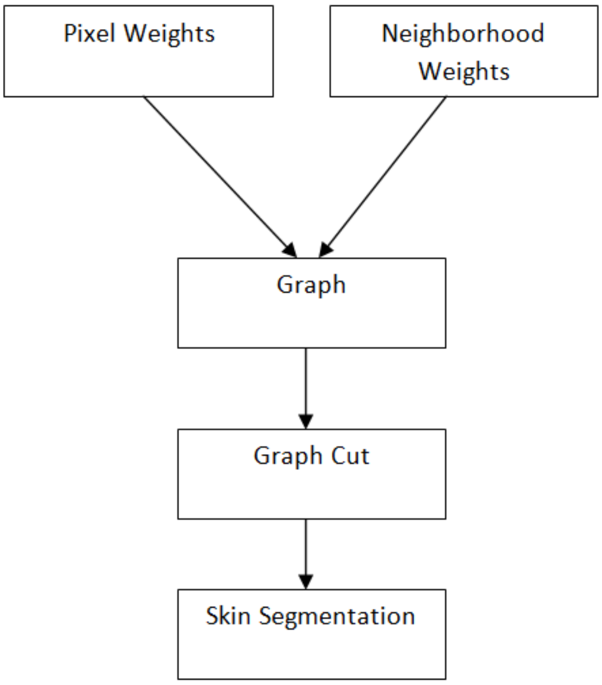


Figure 5.1: Block diagram showing the overview of the skin segmentation based on seeds. Skin segmentation is affected by algorithms for the neighborhood relation between pixels and also by the algorithms for setting weights for the background and foreground pixels.

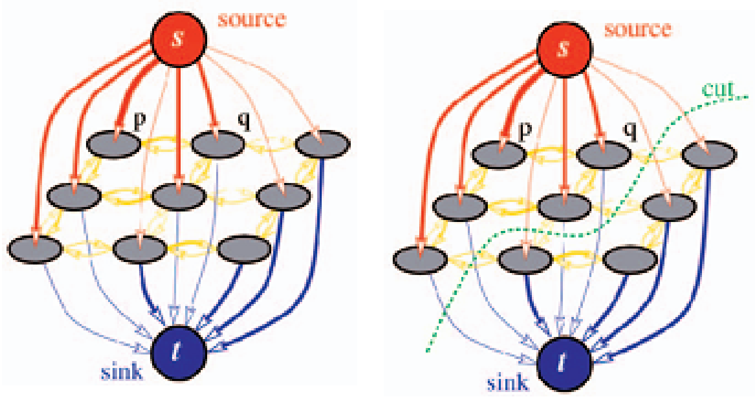
5.2(a), a simple example of a two terminal graph is shown that is used to minimize the energy on a 3×3 image having two labels.

Some weight or cost is assigned to every edge. A cost of a given directed edge (p, q) differs from the cost of the reverse edge (q, p). Generally speaking, there are n-links edges and t-links edges. N-links are used to connect pairs of neighborhood pixels or voxels representing a connectivity in the image. The cost of n-links corresponds to a penalty for discontinuity between the pixels. These costs are derived from the pixel interaction term $V_{p,q}$ in the energy [9]. T-links connect pixels with terminals (labels). The cost of a t-link connecting a pixel and a terminal corresponds to a penalty for assigning the corresponding label to the pixel. This cost is derived from the data term D_p in the energy [9].

Min-Cut and Max-Flow

An s/t cut C on a graph with two terminals divides the graph into two disjoint subsets S and T such that the source s is in S and the sink t is in T. We refer to s/t cuts simply as *cuts*. Figure 5.2(b) shows an example of a cut. The cost of a cut $C = \{S, T\}$ is defined as the sum of the costs of edges (p, q) where $p \epsilon S$ and $q \epsilon T$ and the minimum cut problem on a graph is to find a cut that has the minimum cost among all cuts [9].

The solution to the minimum s/t cut problem is finding a maximum flow from the source s to the sink t. The maximum flow is assumed to be the maximum "amount of water" that can be sent from the source to the sink by interpreting graph edges as directed



(a) A graph G.

(b) A cut on G

Figure 5.2: Example of a directed graph. Weight values are represented by the line thickness. (Source: [9]).

"pipes" with capacities equal to edge weights [9]. Any cut assigns pixels (nodes) to labels (terminals). If edge weights are appropriately set based on parameters of an energy, a minimum cost cut will correspond to a labeling with the minimum value of this energy.

Algorithms for Min-Cut/Max-Flow

There are polynomial algorithms for solving min-cut/max-flow problems on graphs (directed,weighted) with two terminals. Most of the algorithms belong to one of the following two groups [9]:

- Goldberg-Tarjan style push-relabel methods [37].
- Ford-Fulkerson style augmenting paths [29].

Standard augmenting paths-based algorithms work by pushing flow along non saturated paths from the source to the sink until the maximum flow in the graph G is reached while Push-relabel algorithms use a different approach and do not maintain a valid flow during the operation [9]. Instead, the algorithms maintain a labeling of nodes estimating a lower bound on the distance to the sink along non-saturated edges. The algorithms attempt to push excess flows toward nodes with smaller estimated distance to the sink.

We use the min-cut/max-flow algorithm presented in [9] because of its real-time performance. This method belongs to the group of algorithms based on augmenting paths. Augmenting path-based methods build a new breadth-first search tree for source s and sink t paths as soon as all paths of a given length are exhausted. This means that for building a breadth-first search tree, most of image pixels have to be scanned and this thus is a computationally expensive operation. Also rebuilding a search tree on graphs makes standard augmenting path techniques perform poorly [9].

For improvement over the original Augmenting path-based methods, the new min-cut/max-flow algorithm [9] initially builds two search trees, one from the source and the other from the sink. Also the source and the sink trees (initially built) are reused and

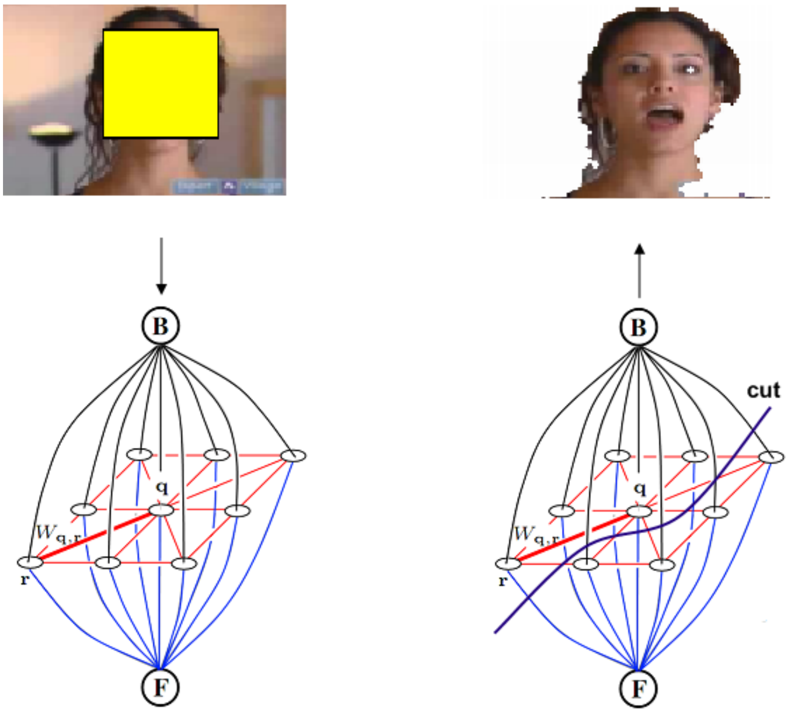


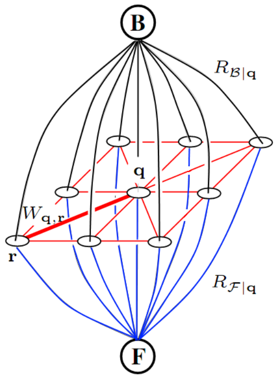
Figure 5.3: A graph cut for skin segmentation. A mask represents the object terminal and the whole image itself represents the background terminal. Note that hair is detected as skin because the seed covers the hair portion.

never built from scratch. The algorithm works several times faster than any of the other methods, making near real-time performance possible.

5.1.2 Graph Representing the Skin Image

Figure 5.3 summarizes the basic skin segmentation paradigm we are following. The skin segmentation is based on some seed from the image on which skin segmentation is to be applied. A graph is constructed whose nodes represent pixels and whose edges represent the weights. The min-cut/max-flow algorithm presented in [9] is used for graph cut. Later on, the concept in Figure 5.3 is extended to skin segmentation without the need for a local seed from the image (universal seed based skin detection).

For the skin segmentation problem, we have two classes named “skin” and “non-skin”. Therefore, we represent these as foreground $\mathcal{F}$ node which means “skin” and the background $\mathcal{B}$ node meaning “non-skin”. These terminal nodes are connected to the non-terminal nodes through edges. The general framework for building the graph is depicted in Figure 5.4. The graph is shown for a 9 pixel image and an 8-point neighborhood N. The neighborhood weights ($W_{q,r}$), foreground/background penalties ($R_{\mathcal{F}|q}$, $R_{\mathcal{B}|q}$) and λ are discussed as follows.


B
$R_{\mathcal{B}|\mathbf{q}}$
q
$W_{\mathbf{q},\mathbf{r}}$
r
$R_{\mathcal{F}|\mathbf{q}}$
F

edge	cost	region
$\{\mathbf{q},\mathbf{r}\}$	$W_{\mathbf{q},\mathbf{r}}$	$\{\mathbf{q},\mathbf{r}\}\in\mathcal{N}$
$\{\mathbf{q},F\}$	$\lambda\, R_{\mathcal{F}\|\mathbf{q}}$	$\forall\mathbf{q}$
$\{\mathbf{q},B\}$	$\lambda\, R_{\mathcal{B}\|\mathbf{q}}$	$\forall\mathbf{q}$

Figure 5.4: Left: Graph representation for a 9 pixel image. Right: Table defining the costs of graph edges. λ is a constant described in the text (source: [75]).

5.1.3 Neighborhood Weights

The edge weights of neighborhood N are encoded in matrix $W_{q,r}$, which is not symmetric as in [74]. The neighborhood size and density has a profound effect on the computation times. For the problem of skin segmentation we control the size and density of the neighborhood through two parameters. These parameters are window size and sampling rate. We use a neighborhood window of size 21×21. For skin segmentation, we use a sampling rate of 0.3. This means that we only select at random 30% of all the pixels in the window. This has two benefits. Firstly, by using only a fraction of pixels we are reducing the computational demands and secondly, only a fraction of pixels allows the use of larger windows and at the same time preserves the spatial relationship between the neighboring pixels. We show the calculation of weight matrix $W_{q,r}$ as follows:
For a grayscale image, the boundary penalties can be calculated as follows [8]:

$$W_{q,r} \propto e^{-\frac{||I_q - I_r||^2}{2\sigma^2}} \cdot \frac{1}{||q,r||} \tag{5.1}$$

where I_q and I_r are the intensities at point q and point r, $||q-r||$ is the distance between these points and σ is a parameter. For skin detection in color images, this function is modified to take color into account as follows:

$$W_{q,r} = e^{-\frac{||c_q - c_r||^2}{\sigma_1}} \cdot \frac{1}{||q-r||} \tag{5.2}$$

where c_q and c_r are the RGB vectors of points at the position q and r respectively. σ_1 is a parameter, for which a value of $\sigma_1 = 0.02$ is used. The boundary penalty of Equation 5.2 is good only for images having no texture. For taking into account the texture in the segmentation process, the neighborhood penalty of two pixels is defined as follows:

$$W_{q,r} = \left(e^{-\frac{g(q,r)^2}{\sigma_2}}\right)^2 \tag{5.3}$$

where σ_2 is a parameter. For skin segmentation we used $\sigma_2 = 0.08$ and

$$g(q,r) = p_b(q) + \max_{s \epsilon \mathcal{L}_{q,r}} p_b(s) \tag{5.4}$$

where $p_b(q)$ is the combined boundary probability explained below and

$$\mathcal{L}_{q,r} = \{x\epsilon\mathbb{R}^2 : x = q + k(r - q), k\epsilon(0, 1)\} \tag{5.5}$$

is a set of points on a line from the point q (exclusive) to the point r (inclusive).

Boundary Estimation

Combined boundary probability is used for selecting weights for the neighborhood pixels in the graph. Therefore, for the skin segmentation task, boundary detection is necessary. On the other hand, detecting true boundaries is a difficult task, as it should work for images of human-made environments and for natural images [75]. The basic emphasis is on detecting boundaries where different textured regions change and not on local changes inside one texture. Edge detectors generally have large responses inside a textured area. Therefore, in order to correctly detect the boundaries, the color changes and texturedness of the regions have to be considered. The brightness, color, and texture gradients introduced by [67, 70] are used.

Color and Brightness Gradient

We explain the brightness and color gradient calculation for boundary detection with reference to the CIELAB color space, as this color space is perceptually uniform. Given an image, at every pixel location in the image a circle of radius r is created. The circle of radius r is divided along the diameter at the orientation θ. A difference between the contents of the disc halves is calculated and a large difference indicates a discontinuity in the image along the disc diameter.

Different orientations can be taken into account. We select 8 orientations, i.e. every 45 degrees, the radius for the L channel $r_L = d/100$, for the a channel $r_a = d/200$ and for the b channel $r_b = d/200$. d is the length of the diagonal of the image in pixels as used by [74] and [70]. The division of a circle along its diameter creates two resulting disc halves. Let histograms g_i and h_i describe the half-disc regions. Then histograms g_i and h_i are compared using the χ^2 histogram difference operator:

$$\chi^2(g, h) = \frac{1}{2}\sum_{i}^{N_b} \frac{(g_i - h_i)^2}{g_i + h_i} \tag{5.6}$$

where N_b represents the total number of bins in histograms g_i and h_i. We selected $N_b = 32$. Since we select 8 orientations, for each pixel there are as many numbers as the orientations. The gradient at every pixel p is calculated as the maximum number in the corresponding orientation. More details can be found in [70]. Finally a gradient for every channel, i.e. $G^L(x, y)$, $G^a(x, y)$, $G^b(x, y)$ is calculated.

Texture Gradient

Here we are interested in capturing the variation in intensities in a local neighborhood i.e. gradient computation in the texton domain rather than the gradient relation to the surface orientation. As in [74], we also take into account the usage of an oriented filter

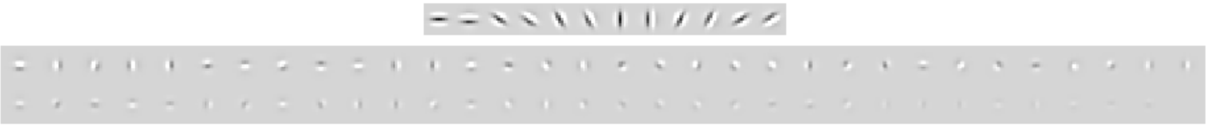


Figure 5.5: Top: Filter bank. Bottom: Universal textons. (source: [74])

bank, shown at the top of Figure 5.5. The filters are based on rotated copies of a Gaussian derivative and its Hilbert transform and contain even and odd-symmetric filters [74]. An image I is convolved with this bank of filters. Similar to [74] we use 24 filters. These filters contain odd and even filters with 6 orientations and 2 scales as can be seen at the top of Figure 5.5. After the convolution operation every pixel now contains a feature vector. Each feature vector has responses to all the filters in the filter bank.

The K-means clustering technique is used to cluster the pixels. As a result of clustering, the *textons* are the dominant K clusters. Universal textons (generalized textons computed from training images) can also be used as shown in the bottom of Figure 5.5. Then an assignment operation assigns each pixel to the closest universal texton. The image greyscale range, which is 0-255, is now scaled to the range $[0, 1-K]$, where K represents the number of textons. The strategy of half discs used for brightness and color gradient calculation is also used for the texture gradient. Here we use 6 orientations for θ. The number of histogram bins used here is basically the number of textons for Equation 5.6. Finally, a texture gradient $G^T(x, y)$ is calculated.

Boundary Detection

For boundary detection, the brightness/color gradients and texture gradients are combined. The merging process of color and texture gradients is aimed at obtaining a single value. A vector comprising of brightness, color, and texture gradients is defined as:

$$X(x, y) = [1, G^L(x, y), G^a(x, y), G^b(x, y), G^T(x, y)]^\top \tag{5.7}$$

where $G^L(x, y), G^a(x, y), G^b(x, y)$ are the color and brightness gradient and $G^T(x, y)$ is the texture gradient. A sigmoid function is used for deciding whether a pixel at position (x, y) is a boundary pixel or not as:

$$p_b(x, y) = \frac{1}{1 + e^{-\mathrm{x}^\top \mathbf{b}}} \tag{5.8}$$

where $p_b(x, y)$ represents the final boundary probability. The constant vector $\mathbf{b}$ contains weights for each of the partial gradients. $\mathrm{x} = (1, 0, 0, 0, 0)^\top$ and $\mathrm{x}^\top \mathbf{b} = b_1$, in the case when there is no boundary change, and "1" at the beginning of the vector x is used to control the weight through the b_1 in the vector $\mathbf{b}$. For obtaining the weights in $\mathbf{b}$, information from all the gradients has to be combined in an optimal way as explained in [70]. We use $\mathbf{b}$ which is provided with the source code by the authors [70].

The resulting values for p_b are between 0 and 1 inclusive. The 0 value for p_b in Equation 5.8 indicates absence of a boundary, while 1 indicates a boundary with maximum

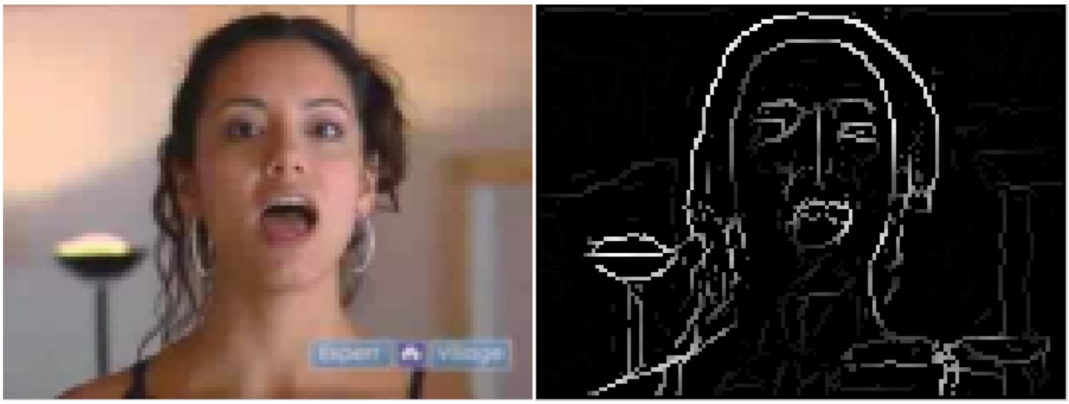
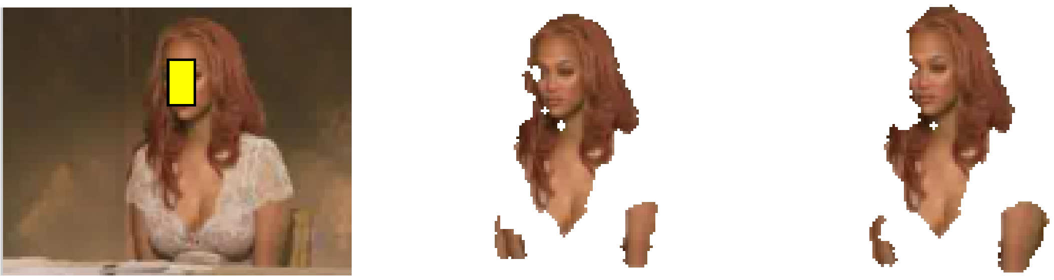


Figure 5.6: Combined boundary probability using color + texture gradient. White points stand for high, black for low boundary probability. Left: original image. Right: boundary probability image.



(a) Original image

(b) Result of segmentation using color.

(c) Result of segmentation using color plus texture.

Figure 5.7: Comparison of using color vs color plus texture for the weight function in the graph.

confidence. Figure 5.6 shows the combined boundary probability. In Figure 5.6 (right image), the white points show high while black points show low boundary probability. The boundary detection scheme is used for taking texture into account instead of color only for the neighborhood weight function. An example of the effect of only using color and the combined color plus texture is shown in Figure 5.7. Figure 5.7(a) shows the original image, Figure 5.7(b) shows the result of color only and Figure 5.7(c) shows the resultant skin segmentation based on color plus texture information. It is shown in Figure 5.7(c) that the part of the hair that was detected as skin in (b) is suppressed.

5.1.4 Foreground/Background (Pixel) Weights

With the pixel being connected to terminal nodes, we can incorporate the information provided by the local or universal seed/template. With this setup, we can incorporate the penalty for each pixel being foreground or background based on [74]. The regional penalty of a pixel as being foreground $\mathcal{F}$ or background $\mathcal{B}$ can be defined as:

$$\begin{aligned} R_{\mathcal{F}|q} &= -\ln p(\mathcal{B}|c_q) \\ R_{\mathcal{B}|q} &= -\ln p(\mathcal{F}|c_q) \end{aligned} \tag{5.9}$$

where $c_q = (c_L, c_a, c_b)^T$ stands for a vector in $\mathbb{R}^3$ of L*a*b* values at the pixel q. To compute the posterior probabilities in Equation 5.9, Bayes theorem is used as follows:

$$p_{(\mathcal{B}|c_q)} = \frac{p(c_q|\mathcal{B})p(\mathcal{B})}{p(c_q)} = \frac{p(c_q|\mathcal{B})p(\mathcal{B})}{p(\mathcal{B})p(c_q|\mathcal{B}) + p(\mathcal{F})p(c_q|\mathcal{F})} \tag{5.10}$$

We first demonstrate it on $p(\mathcal{B}|c_q)$, for $p(\mathcal{F}|c_q)$ the steps are similar. Initially, we do not know a priori the probabilities $p(\mathcal{F})$ and $p(\mathcal{B})$. Thus, we fix them to $p(\mathcal{F}) = p(\mathcal{B}) = 1/2$ as is also reported in [74]. After this assumption the Equation 5.10 reduces to

$$p_{(\mathcal{B}|c_q)} = \frac{p(c_q|\mathcal{B})}{p(c_q|\mathcal{B}) + p(c_q|\mathcal{F})} \tag{5.11}$$

where the foreground prior probability is given by:

$$p(c_q|\mathcal{F}) = f^L_{c_L}.f^a_{c_a}.f^b_{c_b} \tag{5.12}$$

and the background prior probability is:

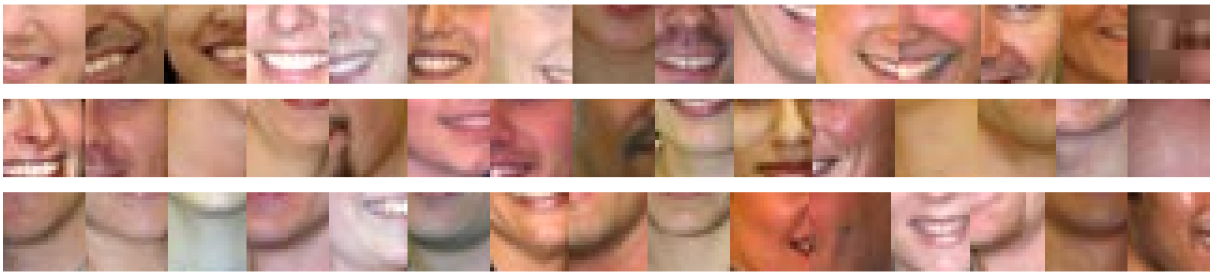
$$p(c_q|\mathcal{B}) = b^L_{c_L}.b^a_{c_a}.b^b_{c_b} \tag{5.13}$$

where $f_i^{\{L,a,b\}}$ and $b_i^{\{L,a,b\}}$ represent the foreground and the background histogram of each color channel separately at the ith bin. For smoothing, we use one-dimensional Gaussians:

$$\bar{f}_i = \frac{1}{G}\sum\nolimits_{j=1}^{N} f_j e^{-\frac{(j-i)^2}{2\sigma^2}} \tag{5.14}$$

where G is the normalization factor enforcing $\sum_{i=1}^{N}\bar{f}_i = 1$. The number of histogram bins, $N = 64$. We select $\sigma = 1$. λ in Figure 5.4 is set to 1,000 and controls the importance of penalties for foreground and background against the neighborhood weights.

For the skin detection scenario, the prior probabilities for "skin" and "non-skin" are calculated from the histograms for "skin" and "non-skin". The "skin" histogram can be computed from all the pixels in the seed/template patch provided by a user interaction or an automatic seed/template detection (for example, face detection). In the next section, we show that with the universal seed concept, we are not restricted to local seeds for skin detection. For the "non-skin" histogram, we do not know the colors of the background and have no information about the template patch of the background. Therefore, we compute the background histogram from all the image pixels, as suggested in [74]. The criteria is based on the assumption that the histogram computed from all pixels includes information on all colors ("skin" and "non-skin") in the image. Therefore, since $\sum_{i=1}^{N}\bar{b}_i$=1, the probability $p(c_q|\mathcal{B})$ gives smaller values than $p(c_q|\mathcal{F})$ for the "skin" colors present in the template. Thus, pixels more similar to the template are assigned in the graph more strongly to the "skin" than to the "non-skin" node.



(a)

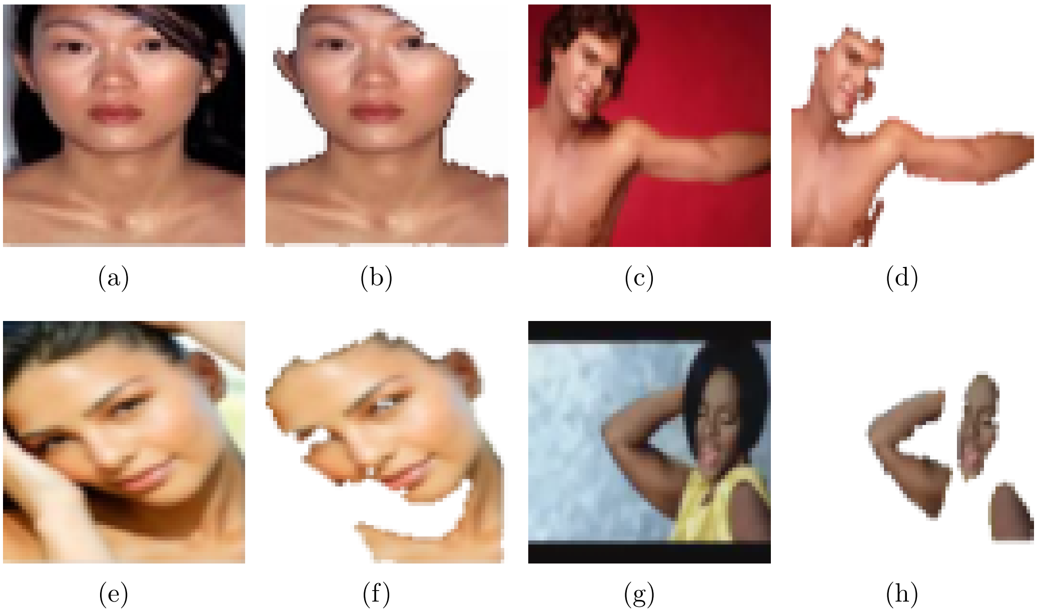
Figure 5.8: Skin samples used for universal seed.

5.1.5 Universal Seed As Static Filter

A local skin patch from the image can be used as a seed to detect skin in an image. One solution for obtaining local seeds from an image is to use a face detector. A face detector is normally followed by post filtering steps for the removal of non-skin portion from the face for using it as the local skin seed. In case of failure of face detection, the local seed based skin segmentation will fail. We propose a concept for processing arbitrary images; using a universal seed to overcome the potential lack of successful seed detections thereby providing a basis for using static foreground weights based skin segmentation. With the universal seed, the objective is to produce a seed/template that is as general as possible and can be used as a skin filter. We base the segmentation process on positive training data samples only, exploiting the spatial relationship between the neighborhood skin pixels. For the universal seed, different skin tones are collected, see Figure 5.8. These positive skin samples cover different ethnicities in different lighting conditions. For the universal seed we do not use the negative (non-skin) portion of the image. Since there could be infinite background/negative training data, the objective is taking a skin/non-skin decision based on representative skin samples. For the skin scenario, this makes sense as the skin covers a well defined area in a color space. We denote these positive representative skin samples as the universal seed. The universal seed is highly adaptive. For adding a new skin patch under different lighting conditions, we simply have to merge it with skin patches and recalculate the foreground histogram. The foreground histogram in Equation 5.12 for a new image is calculated based on this seed. The background histogram is calculated from the whole image.

5.1.6 Experiments

For the universal seed approach, the evaluation is based on F-measure and specificity for datasets DS1 (hybrid set, Section 1.5.1) and DS2 on per image and per pixel basis. Since the universal seed is based on positive training data, the specificity is considered as an evaluation measure to see the performance of the universal seed approach for untrained backgrounds. The total number of features for skin pixels (universal seed) is 2,113,703. For performance comparison, we select classifiers (AdaBoost, BayesNet, NaiveBayes, RBF network and J48) that use positive and negative training data and the YCbCr static filter


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

(e) (f) (g) (h)

Figure 5.9: Universal seed skin segmentation: successful skin detection.

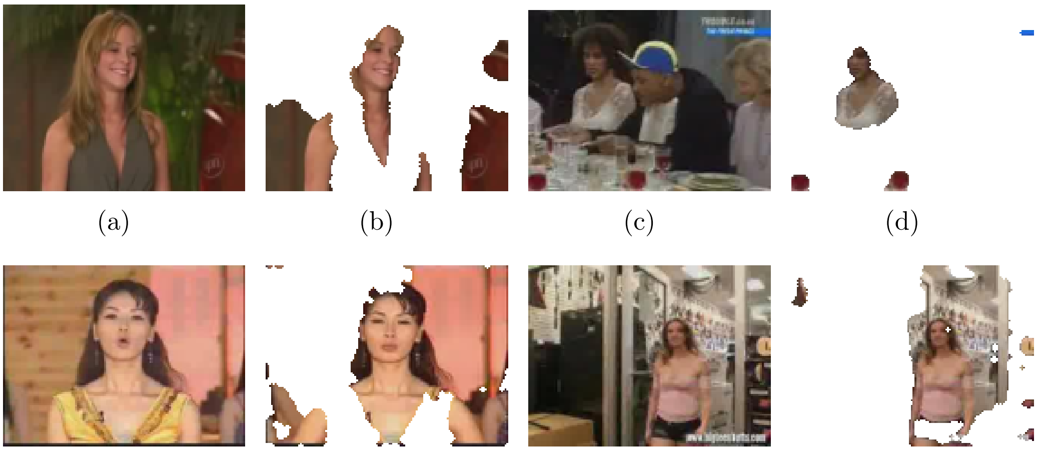
(Section 3.1).

Example results of the proposed approach using the universal seed are visualized in Figure 5.9. Figure 5.10 reports cases where skin is missed or false detections are considered as skin. In the following, the experimental results for datasets DS1 and DS2 are presented and discussed.

For Dataset DS1 Hybrid

Figure 5.11 shows specificity and F-measure for different approaches calculated per pixel. Figure 5.12 reports mean and standard deviation for specificity and F-measure calculated per image. In Figure 5.11, it can be seen that the specificity of the universal seed approach is higher than the YCbCr static approach and less than AdaBoost, BayesNet, NaiveBayes, RBF network and J48. Similarly, the mean and standard deviation for specificity in Figure 5.12 show that the specificity mean (0.54) of the universal seed approach is lower than that of AdaBoost (0.77), BayesNet (0.63), NaiveBayes (0.68), RBF network (0.85) and J48 (0.83). The specificity mean of the universal seed is higher than that of the YCbCr static approach (0.41). In terms of increasing/decreasing trends in Figure 5.12 for specificity, the universal seed approach is similar to BayesNet and NaiveBayes approaches. The mean and standard deviation for F-measure in Figure 5.12 show that the universal seed mean (0.37) is lower than J48 (0.52) and higher than AdaBoost (0.32), BayesNet (0.36), NaiveBayes (0.22), YCbCr static (0.25) and RBF network (0.16).

Regarding F-measure (precision and recall), Figure 5.11 shows that the universal seed approach has a higher F-measure (0.54) compared to other approaches with the exception of the tree based classifier J48. The universal seed approach provides increased classifi-



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

(e) (f) (g) (h)

Figure 5.10: Universal seed skin segmentation: cases where skin is not properly segmented.

cation performance of almost 4% to AdaBoost, 3% to Bayesian network, 18% to Naive Bayesian, 12% to YCbCr static filter, 26% to RBF network and decreased performance of almost 5% compared to J48. For the test data set the universal seed approach outperforms other approaches in terms of precision and recall with the exception of J48. For the skin detection scenario, the J48 simple rule based decision classification generalizes well for simple feature based classification with high F-score, outperforming the universal seed approach. Since the universal seed approach is based on training on positive data only, we get comparably low values for specificity.

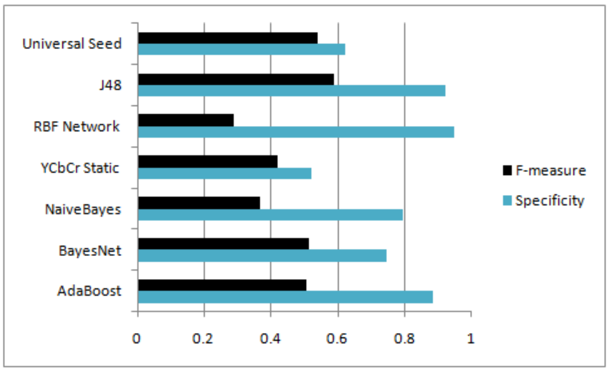


Figure 5.11: Universal seed: Specificity and F-measure for the evaluated approaches using DS1 Hybrid images. The values reported are per pixel.

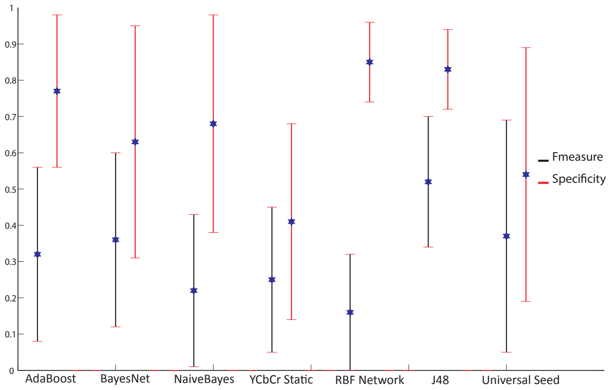


Figure 5.12: Universal seed: Mean and standard deviations for F-measure and specificity per image using DS1 hybrid images.

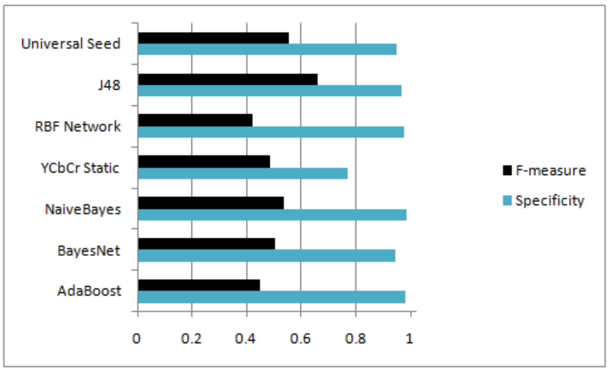


Figure 5.13: Universal seed: Specificity and F-measure for the evaluated approaches using dataset DS2. The values reported are per pixel.

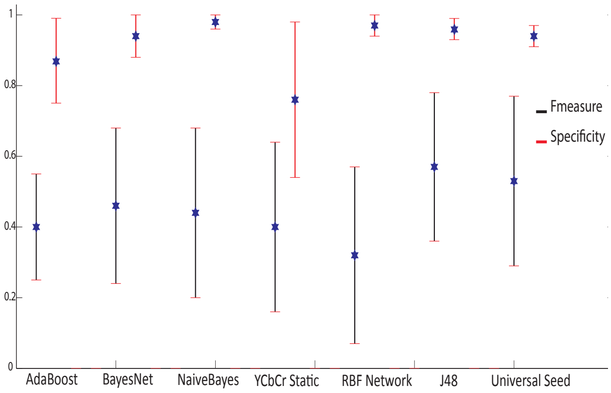


Figure 5.14: Universal seed: Mean and standard deviations for F-measure and specificity per image using dataset DS2.

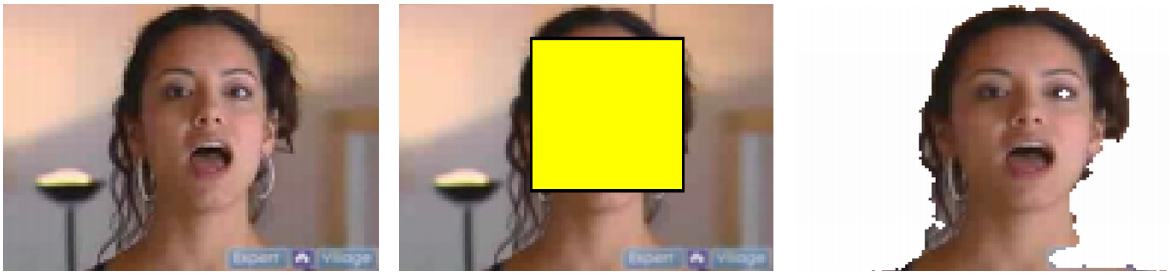
For Dataset DS2

For the evaluated approaches, specificity and F-measure calculated per pixel is shown in Figure 5.13. Figure 5.14 reports mean and standard deviation for specificity and F-measure calculated per image. In Figure 5.13, it can be seen that the specificity of the universal seed approach is higher than the YCbCr static and BayesNet and less than AdaBoost, NaiveBayes, RBF network and J48. The mean and standard deviation for specificity in Figure 5.14 show that the specificity mean (0.94) of the universal seed approach is lower than that of NaiveBayes, RBF network and J48 and higher than AdaBoost, BayesNet and YCbCr static filter with smaller standard deviations in comparison to the statistics of dataset DS1.

In terms of increasing/decreasing trend in Figure 5.14 for specificity, the universal seed approach is similar to RBF network, and J48. The mean and standard deviation for F-measure in Figure 5.14 shows that the universal seed mean (0.538) is lower than J48 (0.57) and higher than AdaBoost, BayesNet, NaiveBayes, YCbCr static and RBF network. Regarding precision and recall, Figure 5.13 shows that the universal seed approach has higher F-measure (0.55) compared to other approaches with the exception of the tree based classifier J48 with F-measure of 0.66. Based on F-measure, the universal seed approach provides increased classification performance compared to AdaBoost, Bayesian network, Naive Bayesian, YCbCr static, RBF network, though it is trained on skin data only. Similar to DS1, since the universal seed approach is based on training on positive data only, we get comparably low values for specificity.

5.2 Local Seeds (Contextual Information)

With the universal seed, we do not take advantage of the contextual information present in the scene. In this section, we introduce the usage of contextual information in terms of faces for increasing skin detection performance. The detected faces are used as foreground seeds for calculating the foreground weights of the graph. Such an approach takes



(a) Original image

(b) Detected face used as mask and overlayed on original image.

(c) Result of a skin segmentation based on face as a seed. The hair is also detected as skin because the mask includes this information.

Figure 5.15: Face detection used as a seed/template for skin segmentation.

advantage of learning the skin distribution in changing lighting conditions and changing skin color from person to person. For local seed generation, we use the Viola and Jones [111] face detector.

5.2.1 Face As Local Seed

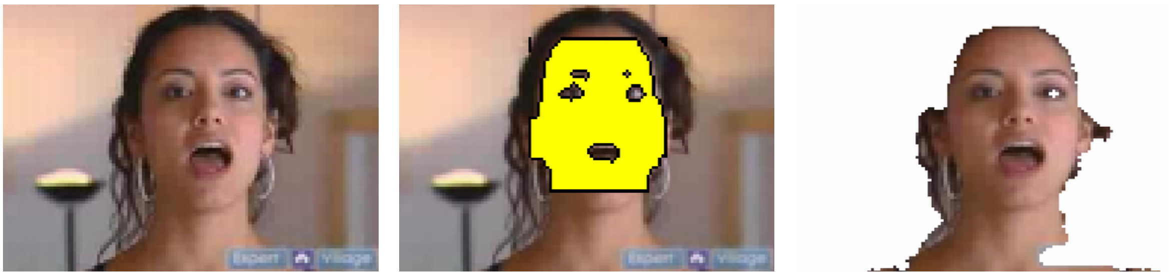
The face detector of Viola and Jones [111] finds faces and returns a rectangular area representing the face. The histogram for foreground (section 5.1.4) is calculated from the face area. The histogram for background is calculated from the whole image including the face area. Based on these histograms, the weights for foreground and background are calculated and assigned to the edges in a graph. Finally a graph cut algorithm segments the image into skin and non-skin areas, see for example Figure 5.15. Figure 5.15(a) shows the original image. In Figure 5.15(b) the detected face seed is overlayed on the face to show the area covered by the seed. Finally, Figure 5.15(c) shows the result of skin segmentation using graph cuts. Note the hair detected as skin, because the seed included this information. For skin detection such a strategy will increase false positives. In the following, we present strategies to reduce false positives.

5.2.2 Seed Filtering

We need to get rid of the non-skin information added by the seed. We call such a strategy where non-skin information is removed "seed filtering". We present two techniques for removing false information added by the seed to the graph weights. First using a static skin filter, and second by reducing the seed dimensions i.e. width and height.

Seed Filtering Using Static Filter

A static filter in the YCbCr color space (Section 3.1) is used for filtering out non-skin information. Skin is detected in the rectangular area (returned by the face detector) using the YCbCr static filter. Only those pixels which are reported as skin by the static filter



(a) Original image

(b) Static skin color model used to filter the seed/mask. The areas of hair are filtered out by the static filter.

(c) Result of a skin segmentation based on the static filter applied to filter out non-skin information.

Figure 5.16: Face detection used as a seed/template for skin segmentation while applying static skin filter to filter non-skin areas.

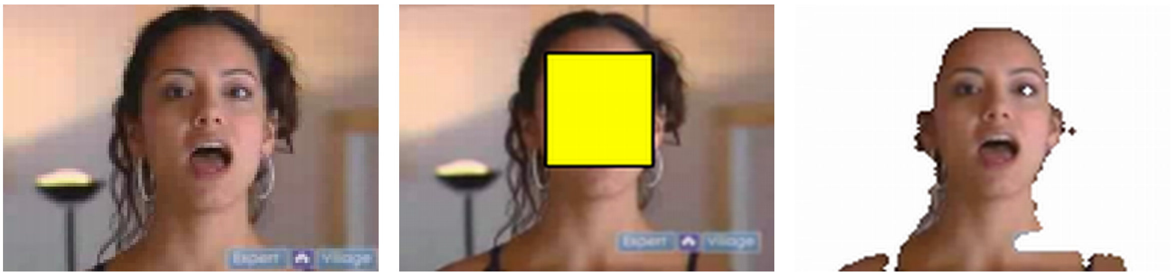
are included in the seed. The rest of the pixels are masked out. With this approach we get a seed which contains only skin pixels. Since, the values of the static filter apply to a very broad range of illumination circumstances and a wide range of skin colors, such a filter could be used with confidence. Figure 5.16 shows the usage of a static filter to block hair from being counted as skin. Figure 5.16(a) shows the original image, (b) shows the filtered seed overlayed on the image and (c) shows the result of skin segmentation based on this seed. The result in Figure 5.16(c) clearly shows the benefit of using a static filter at the initial phase of skin segmentation to block non-skin information from the seed.

Seed Size Adjustment

A simple strategy to block the unwanted information from being included in the seed is to reduce the dimensions of the rectangular area returned by the face detector. We have done experiments as to how much will be the accurate reduction in the size of width and height to remove the unwanted information. Experimentally 25% width reduction and 20% height reduction removes the hair and the background information. An interesting property of our seed based skin segmentation using graph cuts is that only a representative skin information is needed to accurately segment skin. So we can reduce the rectangular area returned by a face detector to 50% by width and 50% by height. Figure 5.17(b) shows the reduced size by 25/20% of the actual face seed. Skin segmentation is improved and can be seen in the Figure 5.17(c). See Figure 5.15 for comparison with the full mask used as a seed.

5.2.3 Multiple Seeds

An interesting property of the seed based skin segmentation is the incorporation of multiple seeds in the segmentation process. If more than one face is detected by the face detector, the detected faces are used as multiple seeds. The histogram for the foreground

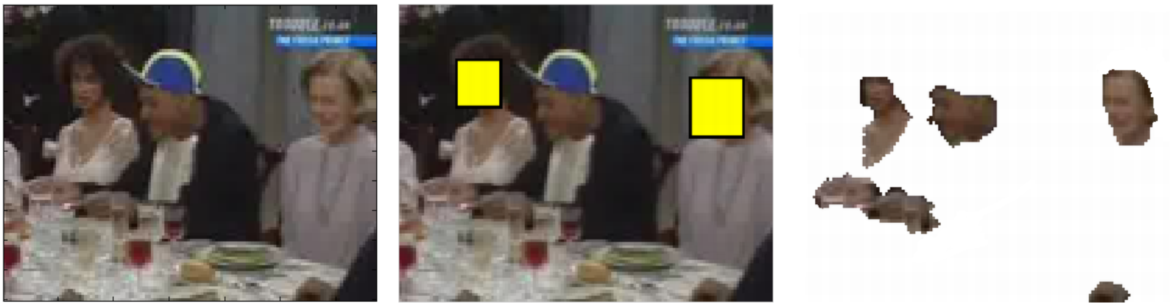


(a) Original image

(b) Seed size adjusted and overlayed on the original image.

(c) Result of skin segmentation using reduced sized seed.

Figure 5.17: Reducing the dimensions of a seed removes most of the non-skin information (background and hair) improving skin segmentation.



(a) Original image

(b) Two detected faces used as masks and overlayed on an original image.

(c) Result of skin segmentation using two faces as seeds.

Figure 5.18: Multiple seeds are useful to track different skin tones.

is constructed from multiple seeds. This broadens the skin detection/segmentation window.

Using multiple faces as multiple seeds, enables the detection of skin in different lighting conditions and for people with different skin tones. Figure 5.18(a) shows people with different skin tones, (b) shows multiple seeds overlayed on the original image and finally, (c) shows skin detected for different people with different skin tones.

5.2.4 Seed Propagation

The seed based approach is useful for segmentation and for the detection of skin in images with a characteristic a priori known property defined through a seed/template patch. An important property of videos is the correlation between frames in a given scene. As video contains frames and in a specific scene, the incoming frames are similar in characteristics and lighting conditions and thus a seed from one frame in a sequence can be used to detect skin in other frames. Figure 5.19 shows how a seed obtained from one frame of a video can be used to detect skin in the subsequent frames. Figure 5.19(a) shows one of the original frame from a video scene used for seed generation, (b) shows the further incoming frames in this particular scene and (c) shows skin segmentation using the seed from (a).

The same seed patch can also be used to detect skin in other images in very different scenes/videos and even slightly different lighting conditions. Figure 5.20(a) shows a frame from video and the seed overlayed on the image and (b) is the detected skin based on this seed. The same seed from the Figure 5.20(a) is used to detect skin in other different video frames, see Figures 5.20(c) and (e) and their detected skin in (d) and (f).

In the experimental setup of Section 5.3, we do not use seed propagation. We take advantage of the universal seed for skin segmentation in the absence of local seeds from an image.

5.3 Classifier Probabilities Integration (A Systematic Approach)

In this section, we aim to introduce an external off-line learned model and integrate it into the seed (local and universal) based approach for improving the overall skin segmentation performance. We call such an integration an "augmentation". For weights augmentation, we use a classifier (J48) as an external model. With the classifiers, we get skin and non-skin probabilities for each pixel. Any classifier can be used to augment the weights for seed-based skin segmentation. As such, we assume that a classifier provides robustness due to its training on positive and negative data. At this point, we present a systematic approach for skin segmentation with graph cuts by using local skin information, universal skin information and off-line model integration (J48 classifier).

The skin segmentation process starts by exploiting the local skin information of detected faces. The detected faces are used as foreground seeds for calculating the foreground weights of the graph. If local skin information (based on face detection) is not available, we opt for the universal seed. To increase robustness, we learn a decision tree based classifier (J48). The learned model is used to augment the universal seed for skin segmentation when no local information (based on face detection) is available from the image. With the weight integration technique, we improve the overall skin segmentation performance.

The systematic skin segmentation approach is summarized in a block diagram in Figure 5.21. If a face (or faces) is detected, we compute the foreground histogram based on the face template. If a face is not detected we load a universal seed and compute the foreground histogram based on this universal seed. The background histogram is calculated from the whole image. The foreground and background histograms are used to create foreground and background weights. The foreground/background weights and the neighborhood weights are used to create a graph. If the universal seed is used, the foreground weights are integrated with the weights given by the probabilities of the decision tree based classifier (J48). The weight integration schemes are described in the following section.

5.3.1 Weight Augmentation Using Decision Trees (J48)

For the J48 classifier, the leaf nodes contain class labels instead of tests. In classification mode, when a test case (which has no label) reaches a leaf node, it is classified using the label stored there. The foreground probabilities are estimated based on the decision functions throughout the visited internal nodes. We train the classifier based on annotated pixel values in the YCbCr color space.

The skin probabilities $p_{skin-j48}$ and non-skin probabilities $p_{non-skin-j48}$ are obtained from the J48 classifier. Then using the classifier skin probabilities, we obtain the skin and non-skin weights as follows:

$$FJ = -\ln(p_{skin-j48})\lambda \tag{5.15}$$

$$BJ = -\ln(p_{non-skin-j48})\lambda \tag{5.16}$$

where FJ and BJ are the foreground and background weights of the J48. λ is set to 1,000 as in cases of universal seed and the seed based approach. It controls the importance of penalties for foreground and background against the neighborhood weights. Similarly, foreground and background weights from the universal seed are obtained representing them as FU and BU for foreground and background respectively.

With the setup of graph cuts, the weights from the classifier alone can be used to improve overall segmentation. Figure 5.22 shows a comparison of skin detection using the J48 alone and the integration of J48 probabilities into the graph cuts approach. Figure 5.22(b) and (e) show the result of skin detection using the classifier voting for pixels alone. Figure 5.22(c) and (f) show the result of skin detection by calculating the weights based on the probabilities of the classifier and passing them to the graph cuts approach. More stable and accurate blobs are constructed and reported as skin. Also false positives are suppressed.

There are a number of possibilities for merging weights from the universal seed and the J48 classifier. We aim to merge the segmentation capabilities of the universal seed and that of the classifier for improving skin segmentation. As such, we choose the best possible combination with a set of preliminary experiments on representative data gathered from the datasets. Due to the extended time required for testing all the combinations on both datasets DS1 and DS2, we use a smaller representative set of 400 images from DS1 and 200 images from DS2.

Figure 5.23 shows the results of different possibilities compared based on F-measure. FJ and BJ represent the foreground and background weights of the J48, while FU and BU are the foreground and background weights for the universal seed. The first scenario includes adding the foreground weights of the universal seed and J48, keeping the background weights of the universal seed only. This possibility achieves the lowest F-measure. In the second case, we add the foreground and background weights from both the universal seed and J48, achieving higher F-measure compared to the first scenario. In the third scenario, we keep the universal seed weights for foreground and use the J48 weights for background. This combination outperforms the first and second combination. The fourth possibility is using J48 weights as the foreground weights and the universal seed weights as the background weights. This setup outperforms all other combinations.

In the fifth combination, we set a test that if the J48 weights for foreground are higher, then they will be used as the foreground weights and the background weights are fixed to that of the universal seed. This reports lower F-measure compared to the fourth scenario. In the sixth combination, if the J48 weights for foreground are higher then they will be used as the foreground weights and if J48 weights for background are higher then they will also be used as the background weights. With this, we achieve a second best performing combination. In the last combination, if foreground weights of J48 are higher, they will be used and if the background weights of universal seed are higher, the J48 weights will be used. We do not get an improvement of results with this scenario. For all the experiments, we use the fourth combination which out-performs all other combinations.

5.3.2 Experiments

We evaluate on the basis of F-measure (per pixel):

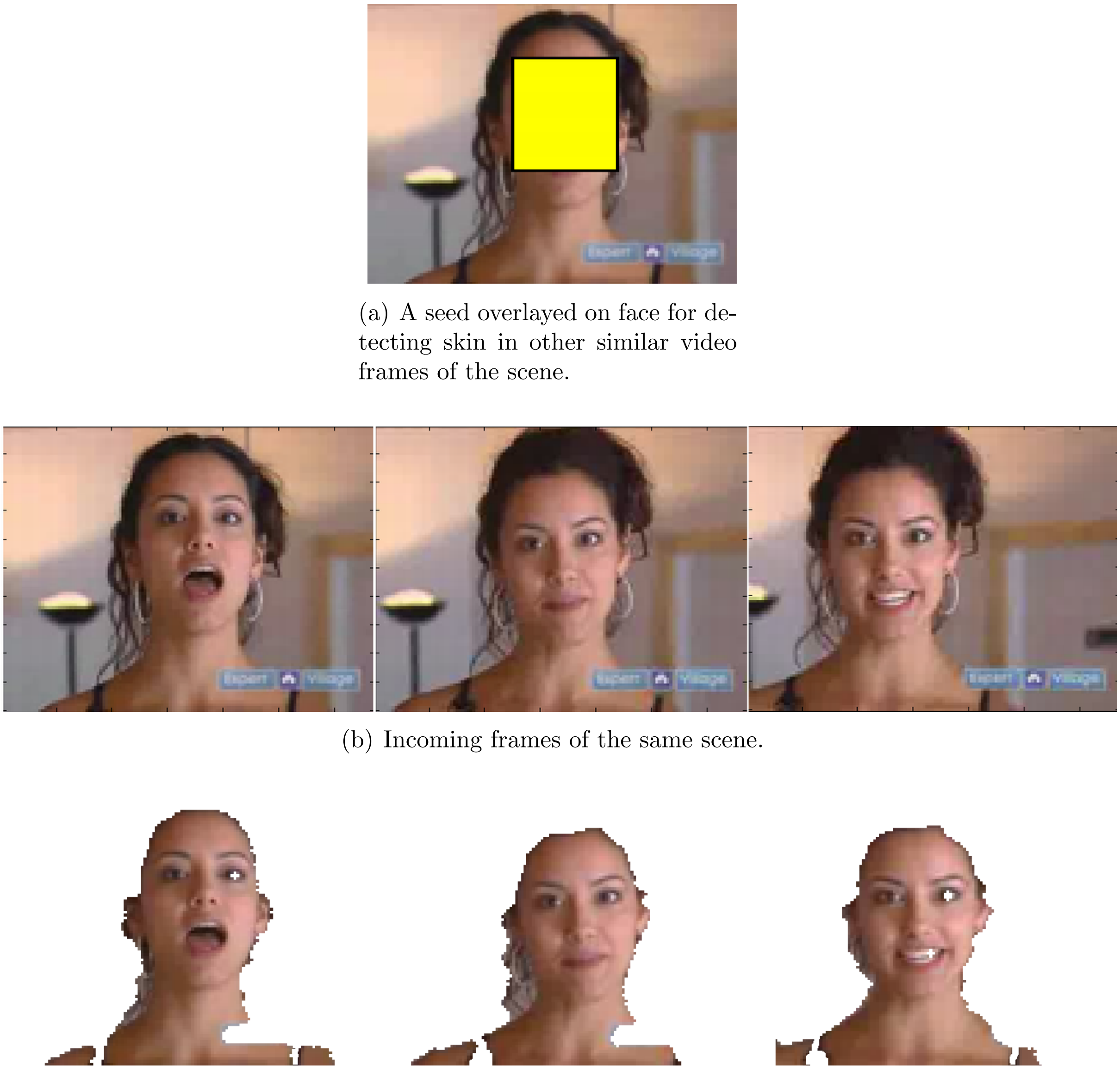
- The universal seed only approach.
- The universal seed plus face (if a face is detected, it is used as seed otherwise the universal seed is used).
- J48 only.
- The systematic approach where we combine the local seed (face based), the universal seed and the J48.

Figure 5.24 shows a comparison of skin segmentation with and without the systematic approach described. The first column of Figure 5.24 contains original images. The second column shows skin detection using the universal seed only approach. The third column shows an improved skin segmentation using the combination of J48 and the universal seed. The areas of skin which were missed by the universal seed are detected by the described approach. False positives are also suppressed by the background weights of the J48.

Figure 5.25 shows the F-measure for the four approaches on datasets DS1 and DS2. For DS1, the universal seed achieves an F-measure of 0.54. With the usage of faces, we get an increased F-measure of 0.59. With J48 alone, we get an F-measure of 0.59. With the systematic approach, we get an increased F-measure of 0.68. For DS2 in Figure 5.25, we get an F-measure of 0.55 for the universal seed. We get an increased F-measure of 0.64 with the usage of faces. With J48 alone, we get an F-measure of 0.66. With the systematic approach, we achieve an increased F-measure of 0.71. For both DS1 and DS2, the systematic approach achieves the highest F-measure (with the exception of 10-fold cross validation in Section 3.2.1) of all the approaches developed in this thesis.

5.4 Summary

In this chapter, for training based on positive data only, we presented the idea of the universal seed. With the universal seed, we proposed a concept for processing arbitrary images; to overcome the potential lack of successful seed detections thereby providing a basis for general skin segmentation. With the local seed based approach (based on detected faces), we provided robust skin detection for different skin tones and varying illumination and showed that local seed based skin segmentation can be improved with seed filtering. We introduced a systematic approach for skin segmentation with graph cuts by using local skin information (detected faces), universal skin information (universal seed) and skin augmentation using an off-line learned model. For the systematic approach, when no local seed is available, skin segmentation can be improved by integrating external models into the existing setup of the universal seed approach. For merging the universal seed and classifier, we found that using foreground weights of the J48 classifier and background weights of the universal seed outperformed all the other possible combinations.

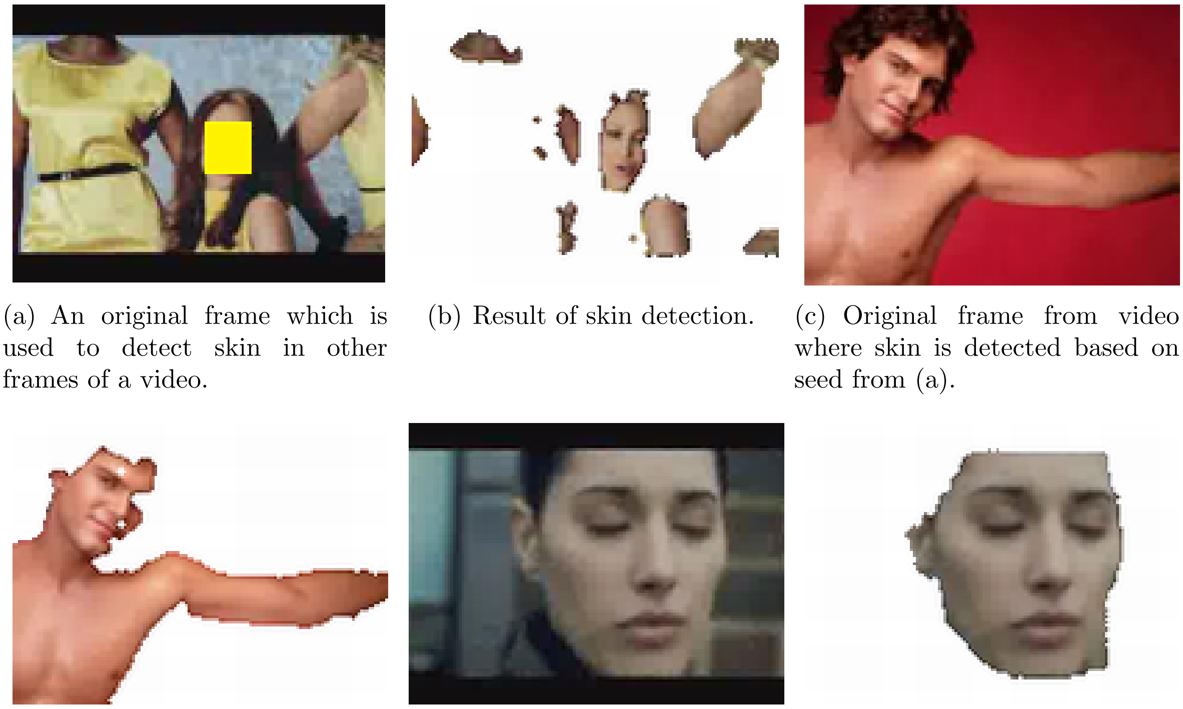


(a) A seed overlayed on face for detecting skin in other similar video frames of the scene.

(b) Incoming frames of the same scene.

(c) Results of skin segmentation using the seed from (a).

Figure 5.19: A seed from one frame can be used to detect skin in other frames having similar characteristics.



(a) An original frame which is used to detect skin in other frames of a video.

(b) Result of skin detection.

(c) Original frame from video where skin is detected based on seed from (a).

(d) Skin detected based on seed from (a).

(e) Another frame of a video where skin is detected based on seed from (a).

(f) Skin detected based on seed from (a).

Figure 5.20: Using a seed from a specific frame, skin can be detected even in different scenes and videos using the seed based approach. The seed from (a) is used to detect skin in frames (c) and (e) from two different videos. Note that skin is detected even there is a difference of context and lighting conditions.

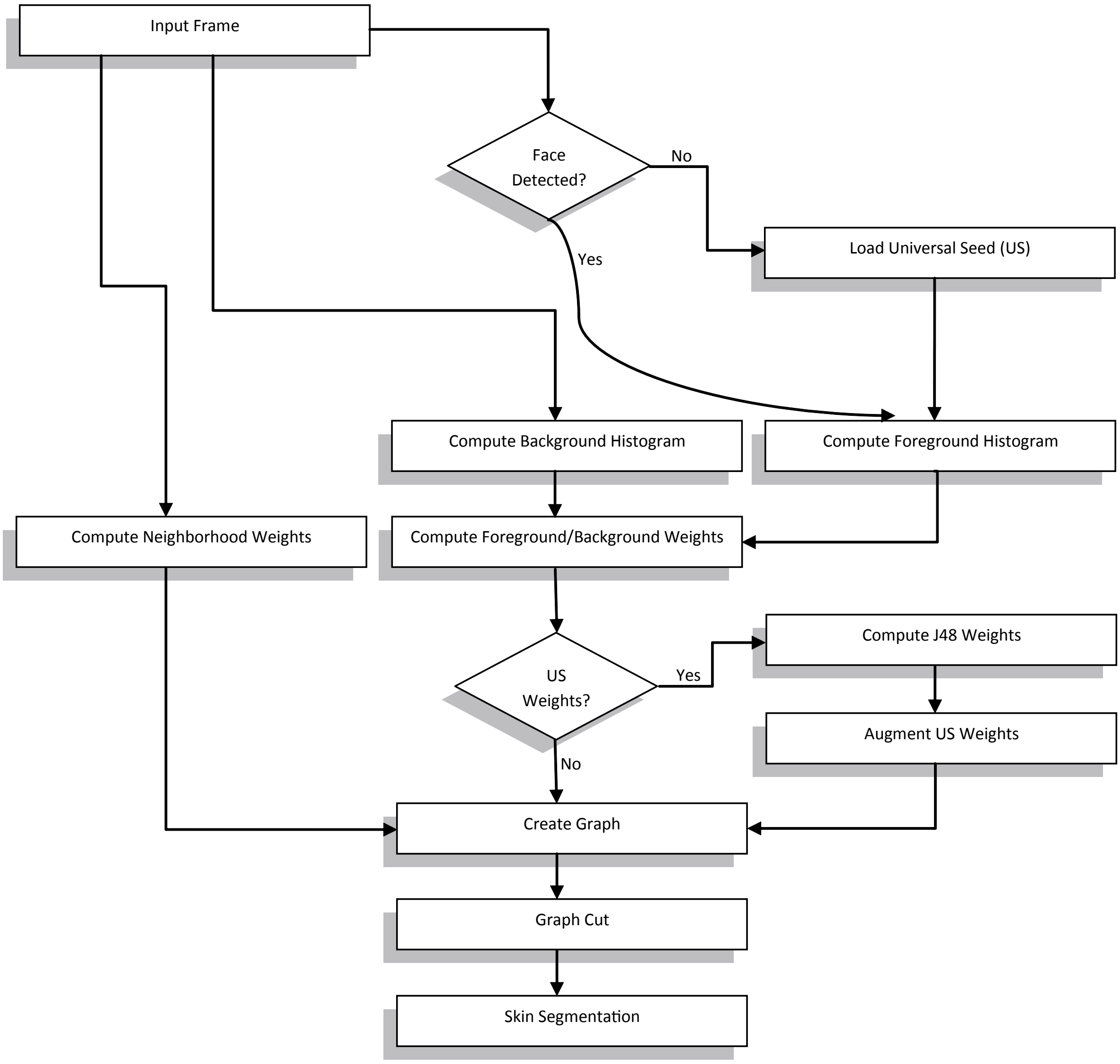
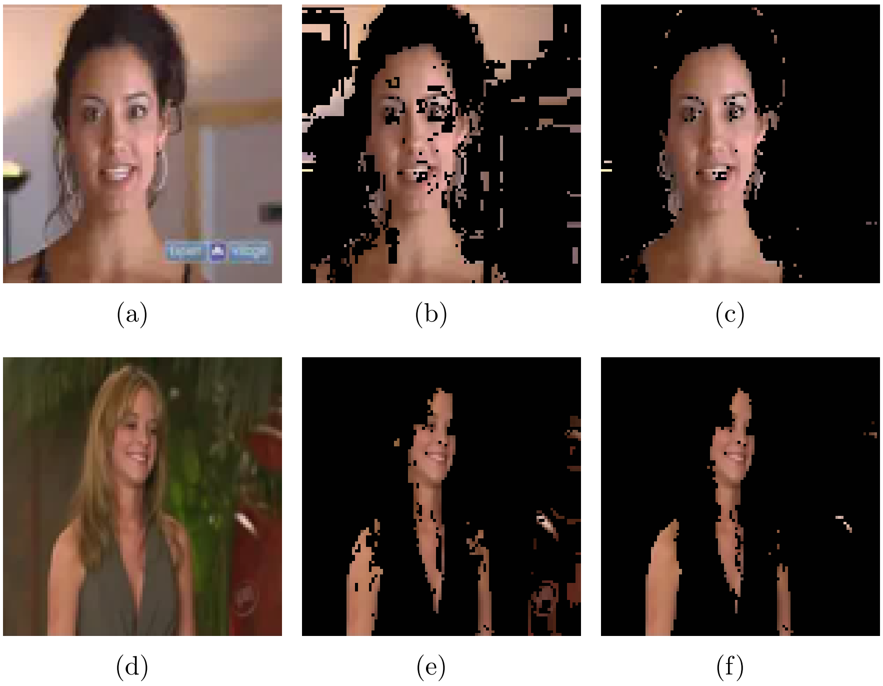


Figure 5.21: Systematic approach to skin detection. Merging spatial and non-spatial data.


(a) (b) (c)

(d) (e) (f)

Figure 5.22: Comparison of J48 classification alone vs J48 in graph cut approach. First column: Original images. Second column: J48 alone. Third column: J48 with graph cut approach.

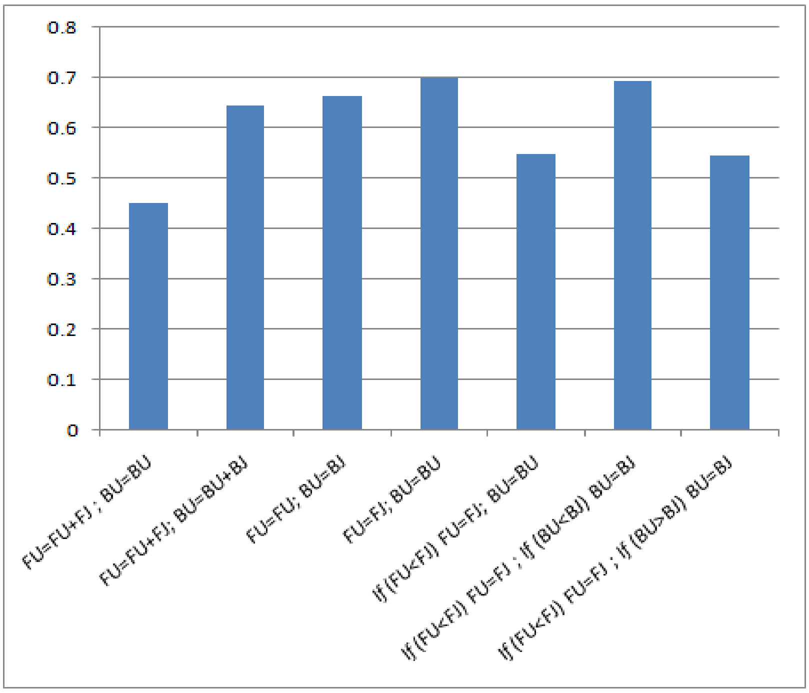


Figure 5.23: Different possible combinations of spatial and non-spatial data.

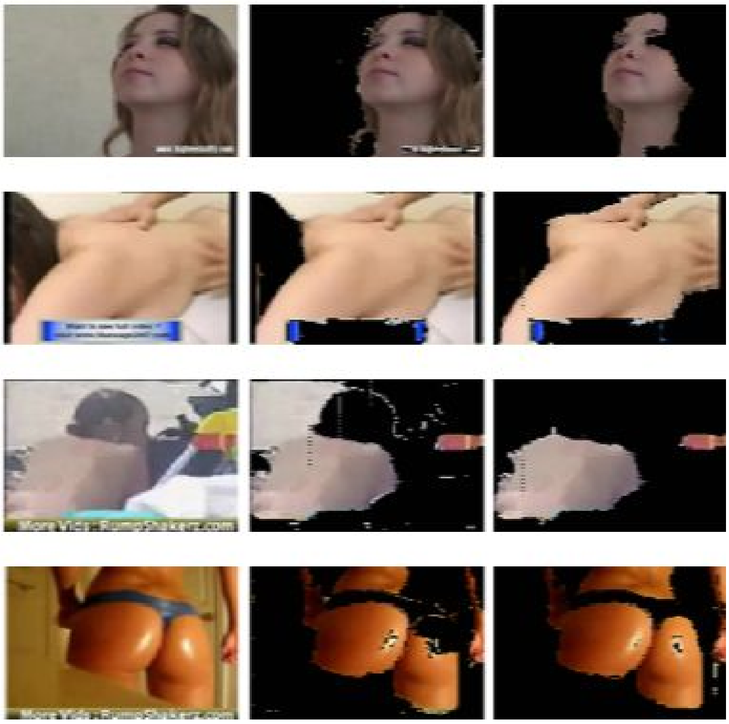


Figure 5.24: First column: original images. Second column: Universal seed based skin segmentation. Third column: Skin segmentation using the systematic approach.

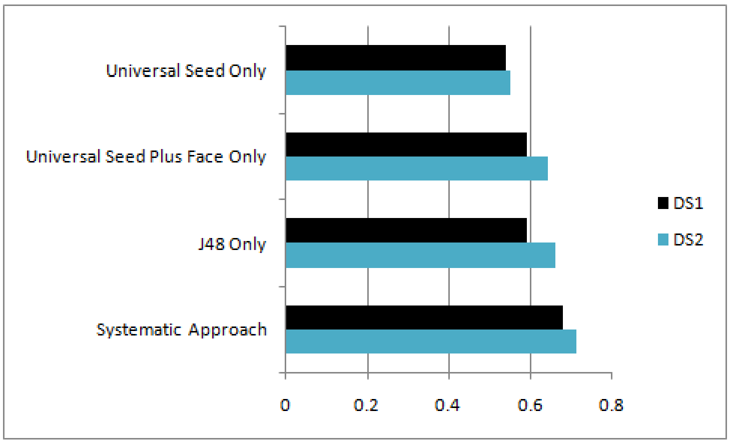


Figure 5.25: F-measure (per pixel) for datasets DS1 and DS2 using the systematic approach.

Chapter 6

Video Sequences and Content Filtering

In this chapter, we port skin detection in still images to videos where real-time performance is a principal concern. In videos, since there is a correlation of one frame to other frames in a particular scene, we can take advantage of the contextual information in one frame to be used for other frames of the scene. For real-time skin detection, we introduce the multiple model approach. We take advantage of the contextual information in terms of faces. With such a strategy, we not only achieve real-time performance but also address the problem of changing lighting conditions in videos by adapting the skin-color model according to reliably detected faces. This approach also solves the problem of multiple people with different skin tones. As an application, we show the usage of real-time skin detection for flagging explicit content.

6.1 Multiple Model Approach

Figure 6.1 gives an overview of the main steps of the multiple model approach. The multiple model approach is based on exploiting the local skin information from faces detected in frames of a video of a particular scene. Prior to any face detection, the static YCbCr skin filter (Section 3.1) is applied for skin detection. The favorable property of the YCbCr color space for skin color detection is the stable separation of luminance, chrominance, and its fast conversion from RGB [109]. These points make it suitable for our real-time skin detection. After a successful face detection, a new model is created for the skin being detected based on this updated model. Due to its real-time performance, we use the Viola-Jones [110] face detector for providing parameters for updating: Any detected face introduces a new skin-color model, which allows the detection of skin of different color and under different lighting conditions even within one single frame (see Figure 6.2). We can do face detection and tracking and color conversion in parallel on the input frame. With this data, we can build our skin model and propagate it to adjust to skin-color variations and illumination changes for a robust skin color classification.

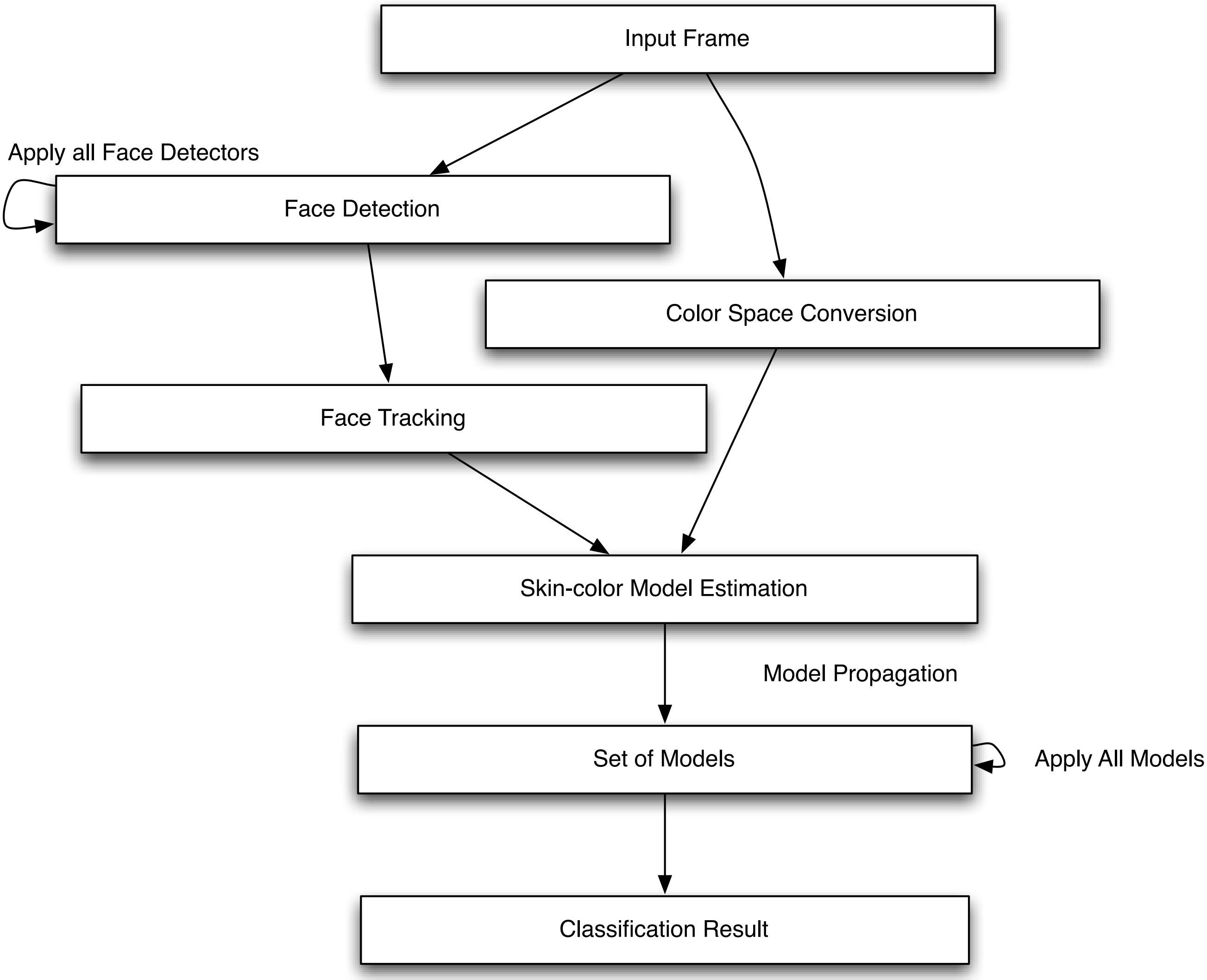
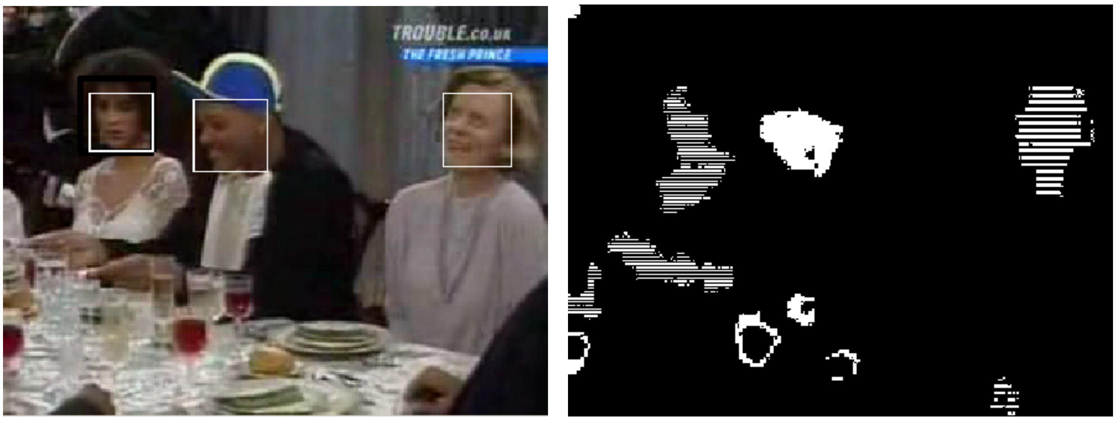


Figure 6.1: Overview of the multiple model approach.



(a) An example frame with three detected faces (three models).

(b) Shadings of white indicate the three applied models.

Figure 6.2: Multiple model skin detection in a frame of an on-line video.

6.1.1 Adaptive Skin Color Modeling

Before any successful face detection, the static skin color model is applied to the frames for skin detection. If a face is detected, the face region is extracted for further processing. From time to time due to the nature of the face detection algorithm, certain parts of the video that are not true faces are reported as faces. This might happen since certain parts of the image, due to compression and artifacts, look like faces to the algorithm. This problem is overcome by the face confidence algorithm. If a face detection time span is under a certain time threshold, it is discarded as noise and not used for the skin detection. For face confidence, we use a simple tracker and confidence check for reasons of achieving real-time performance. Every detected area A_n in the frame n is regarded as a face if:

$$(A_n \cap A_{n-1}) \wedge (A_n \cap A_{n+3}) \geq 0.5 \tag{6.1}$$

The parameters $n + 3$ and 0.5 are chosen after parameter training.

The key problem here is to remove (filtering) the hair and background information from the returned faces. We discard pixels in the detected regions which are most likely non-skin pixels based on the static model and by face area reduction (Section 5.2.2). For face area reduction, we use 25% width reduction and 20% height reduction. After removing the non-skin information from the detected faces, mean is calculated over the skin pixels for updating the model in order to account for real-time performance.

6.1.2 Dynamic Skin Parameters

After filtering, the average obtained is used to model the new boundary values. A comprehensive testing was carried out to calculate the dynamic Cr and dynamic Cb values. We use dCr for dynamic Cr (mean Cr) and dCb for dynamic Cb (mean Cb). It was found out that dCr has more sensitivity (effect on skin detection with value variation) than dCb value, for representing the skin areas. The following boundary gives true positives while minimizing the false positives, and is based on extensive experimentation. They coincide with the values presented in [112].

$$
\begin{aligned}
dCr_{max} &= dCr + 9 \\
dCr_{min} &= dCr - 3 \\
dCb_{max} &= dCb + 10 \\
dCb_{min} &= dCb - 10
\end{aligned}
\tag{6.2}
$$

Equation 6.2 constitutes the boundary values for the adaptive skin filter. These values are used for subsequent frames if no further faces are detected. These values will be discarded and reverted back to the static filter when a scene change is encountered.

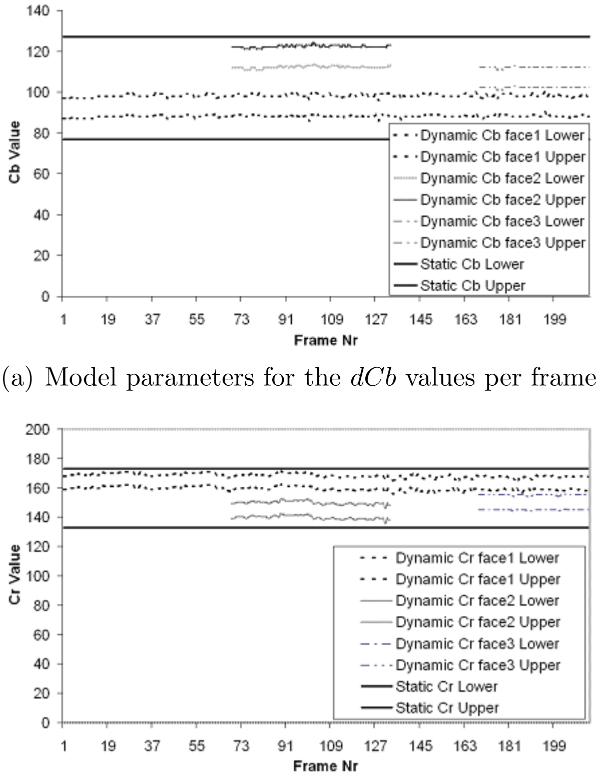
6.1.3 Multiple Models

We capture skin information from a reliably detected face for using it to adapt our model parameters for the skin-color detection. At a time when more than one detected face exists in a frame and face information is properly extracted, we use multiple adapted models. Our multiple model approach makes it possible to filter out skin for multiple people with different skin tones and reduce its false positives. In Figure 6.2(a), three faces are detected which are indicated by white rectangles. Based on this data, three models are estimated and applied (see Figure 6.2(b)). Black indicates non-skin pixels. For every detected region, the shading shows for which model the most pixels apply to.

Scenes having multiple faces at a particular time contribute multiple models to the detection mechanism. A separate model is used for each separate face as shown in Figure 6.2. The development of different models can be seen in Figure 6.3: The upper and lower borders for dCb and dCr values of each applied model per frame are shown. Figure 6.3(a) shows the model parameters for the dCb values and Figure 6.3(b) for dCr values.

The thresholds/boundaries are determined after face size reduction and taking the average of the skin area of the face, providing dynamic parameters i.e. dCb and dCr. These dynamic parameters are expanded using Equation 6.2 to obtain the final dynamic boundary values. Note that the adaptive model just covers a small part of the static one. This multiple adaptive model approach is therefore more precise than the static one in terms of detecting true skin pixels and decreasing false positives (see Figure 6.4).

The models can be processed independently. At any particular time there could be n skin models present in the scene. This can be carried out in parallel per frame. When all faces are lost, the last accumulative model which existed will be used as a default filter for subsequent frames in a particular scene. This strategy is adopted because within a same scene there might be chances that faces are occluded, rotated, not detected in the subsequent frames or lost due to the limitations of the face detection algorithm. This approach overcomes problems of skin-color variance of multiple people with different skin tones. If we find a scene change, we initialize the model parameters to the static model.



(a) Model parameters for the dCb values per frame

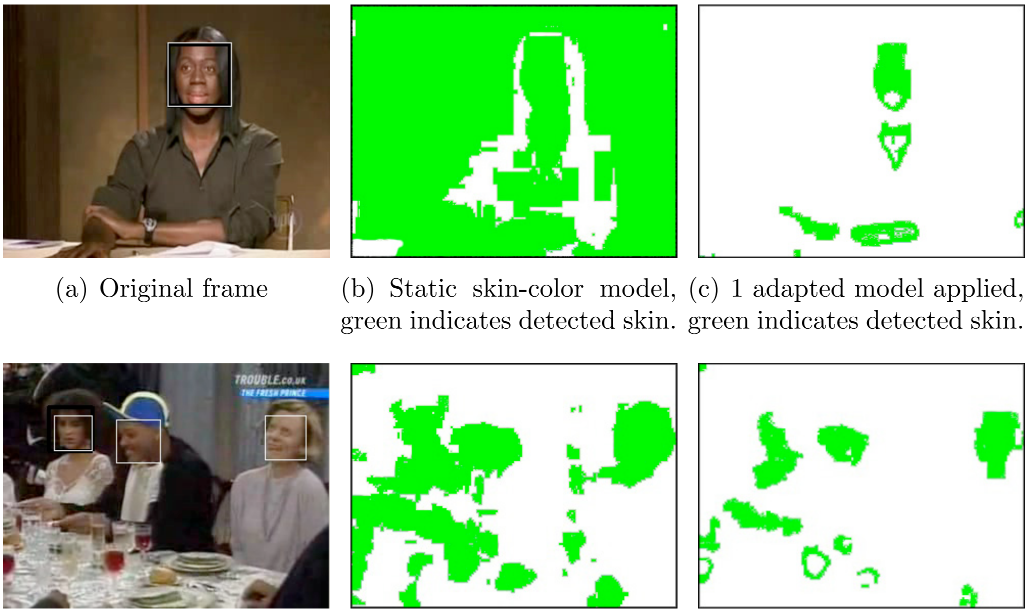
(b) Model parameters for the dCr values per frame

Figure 6.3: The temporal development of models over one complete video clip. The thresholds/boundaries are obtained by scaling up and down the dCb and dCr. The black bold lines shows the YCbCr static filter.

6.1.4 Scene Change

We tested color histogram differences [118] and edge change ratio [120] scene change detectors. We find that color histogram differences performed well for the detection of valid scene changes on our datasets compared to the edge change ratio scene detector. More effective cut detectors are available [65], but this is not the focus of this chapter.

The color histogram-based shot boundary detection algorithm is the most reliable variant of histogram-based detection algorithms [65]. It is based on the principle that the color content in an image does not show rapid transition within but across shots. Therefore, hard cuts and other short-lasting transitions can be detected as single peaks in the time series of the differences between color histograms of contiguous frames or of frames a certain distance d apart from each other. During the course of execution, we initialize the model parameters to the static model by scene change detection. This strategy is adopted because there is a chance that the dynamic model can be overwhelmed by noise and bias towards particular skin color/lighting conditions in a particular scene and will not be valid for the other scenes.



(a) Original frame

(b) Static skin-color model, green indicates detected skin.

(c) 1 adapted model applied, green indicates detected skin.

(d) Original frame

(e) Static skin-color model, green indicates detected skin.

(f) 3 adapted models applied, green indicates detected skin.

Figure 6.4: Comparison of static skin filter and the dynamic multiple model.

6.1.5 Experiments

For skin detection performance comparison, we use the YCbCr static skin filter. The evaluation is based on datasets DS1 and DS2. Since DS1 and DS2 contain images from video sequences and are numbered in the same sequence as in the videos, they are used here for evaluation. For visual interpretation, Figure 6.4 shows an example of improved skin detection using the multiple model approach compared to the static model. Figure 6.5 shows the F-measure calculated on a per pixel basis for the static and dynamic multiple model approaches. It can be seen that the usage of the dynamic multiple model approach provides an increased classification performance compared to the static approach alone. The contextual information makes the approach more precise, reducing the number of false positives. From Figure 6.5, the dynamic multiple model approach provides increased classification performance of almost 10% compared to the static filter alone on DS1 and 12% compared to the static filter alone on DS2, at the same time satisfying the real-time requirement.

6.2 Objectionable Content Filtering

One reason why videos may be considered objectionable is due to explicit sexual content. Such videos are characterized by a large amount of skin visible in frames. Skin detection can be of profound importance in the detection of explicit content. Such an application of skin detection is useful not only for on-line video providers such as Google Videos and YouTube but also as a parental guard for home users by restricting the exposure

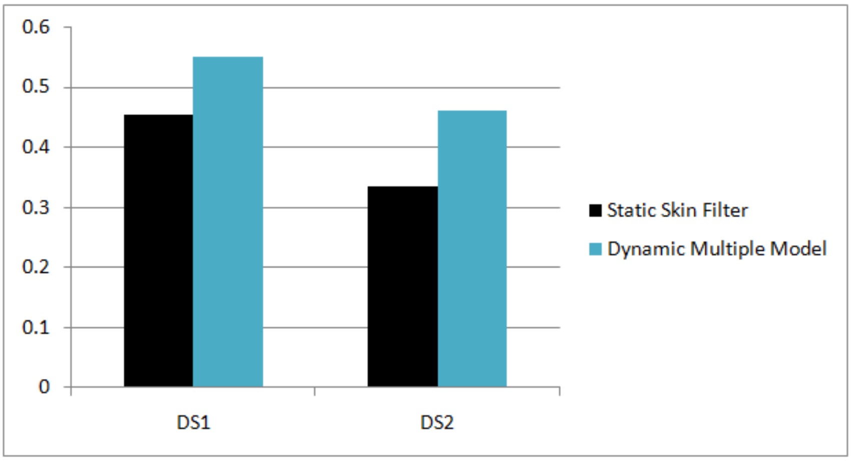


Figure 6.5: F-measure for dynamic multiple model and static skin filter for datasets DS1 and DS2.

of children to pornographic contents. As such, we show the usage of skin detection for flagging explicit content.

For objectionable content filtering, as opposed to the dynamic multiple model approach, we use a profile face detector and frontal face detector in parallel for more robust face detection. We track faces in videos and when the face is lost by the tracker (Equation 6.1), a possible re-detection of the face gives a second very similar skin-color model. We do not discard any model we create in the course of the video. This is an assumption which works well for rather short on-line video clips with a limited number of persons, but does not hold for long movies with a grand variation of illumination circumstances and actors. We also note that this assumption best fits explicit content where the lighting and background mostly remains the same.

6.2.1 Skin Graphs

The multiple model approach can be used to build skin graphs. The skin graph shows skin pixels detected per frame. Figure 6.6 shows the skin graph for an example video. Per frame, the number of detected skin pixels is shown. For peaks in this graph, we extract corresponding frames which are shown above the graph. This leads to a scene classification which could be used for the final video classification and categorization done by humans. Presence of strong skin peaks in a video could be marked for manual classification. Probably merged with other key-frame extraction algorithms, these key-frames give a hint towards the nature of a specific video for a final human classification for conspicuous video clips.

The major problem of classification based on skin graphs is that e.g. portrait shots like interviews and news do have a large amount of skin present in the scene, which makes a decision based on skin pixel count difficult. In the form of “skin paths”, we present a possible solution to this problem.

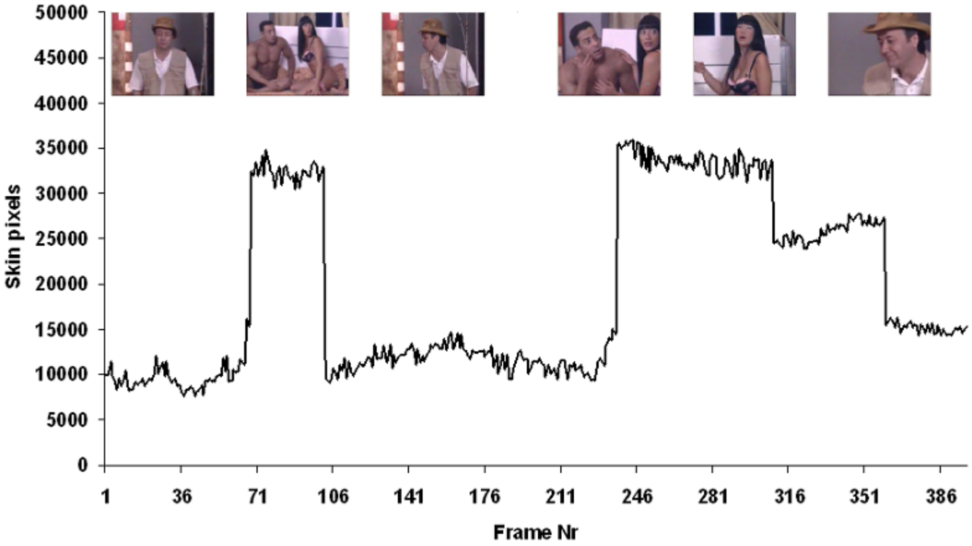


Figure 6.6: Number of detected skin-color pixels per frame (skin graph). For large changes in the graph, we extract a key-frame giving a compact representation of the video clip.

6.2.2 Skin Paths

After a successful face detection, we overcome the problem related to the skin graphs by comparing the amount of skin inside the face region and the whole frame. This measure gives an idea about the scale of the people in a shot. By plotting this measure against the overall skin detection, we are able to describe the property of the given frame meaningfully compared to the skin graph. Videos differ heavily from scene to scene and from shot to shot. To get an idea of a video, we have to provide a compact representation for our detection method.

We introduce skin paths, which average the described measure over a fixed number of frames and give an intuitive idea of the character of a given video. In Figure 6.7, the skin path for video number 9 from dataset DS1 and the related frames are shown. On the x-axis, the mean quotient of the skin color area inside a tracked face and the skin coverage of a fixed number of frames (80) is given. The y-axis represents the mean total skin color coverage in these frames. The path starts at the very left, as in the beginning there is no face detected and much sand is detected as skin. Following the path, more faces are detected giving a better idea about the amount of skin present in the scene. We show in Section 6.2.3 that there are certain areas in the skin graph correlated with properties and content of the videos. From the information of the skin paths, we can categorize the nature of videos reliably by the position of the data points of the graphs. Additionally, there is a trend that data points with $x = 0$ provide more unreliable results as there are none or few faces detected. This gives a confidence measurement for the skin detection itself.

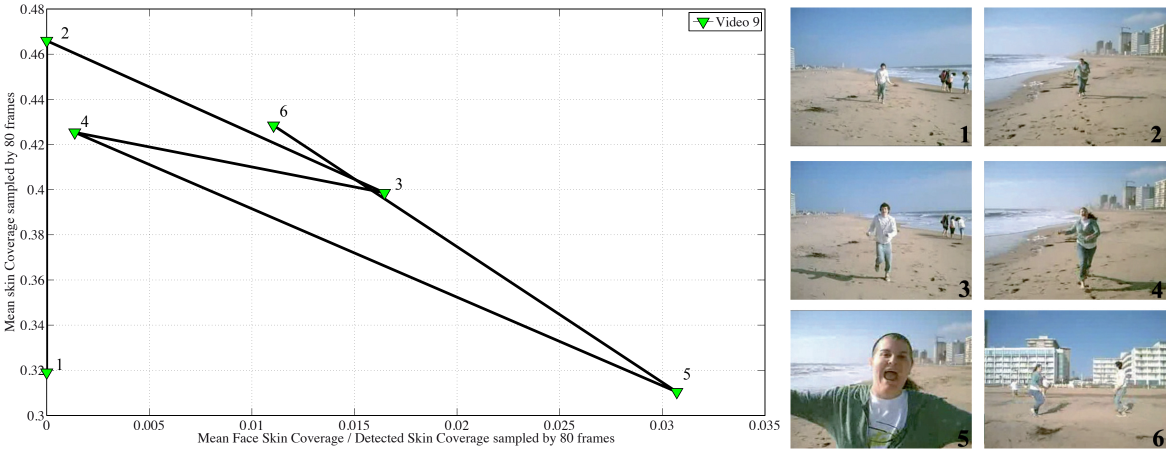


Figure 6.7: Skin path for the classification of Video 9 from DS1 and the corresponding key frames. The detection and incorporation of facial skin makes the results more reliable.

6.2.3 Experiments

In this section, we show the feasibility of adult content filtering using skin detection, on a large dataset containing the most popular on-line videos (at particular time). In the following, we describe the videos used in more detail and evaluate the approach in a large scale experiment.

Datasets

The first dataset is similar to DS1. However, DS1 contains selected images extracted from 25 videos and the dataset we use in this chapter is the complete set of unaltered 25 videos. The second dataset consists of 200 publicly available videos. To provide an objective collection of videos, we chose the 100 most popular videos from YouTube[1]. As there are not more videos available in this category, we additionally gathered 50 videos “being watched right now” which are not in the previous category. For the adult material, we chose the 50 most popular videos from YouPorn[2], as this platform provides explicit adult material only and is publicly available. We want to make sure that our classification is not biased by the two different data sources. There is a probability that the two classes of video material differ e.g. in frame rate, size, video quality or noise level just because of the two platforms they are downloaded from. Such criteria would nullify any classification success. Therefore, we chose 10 videos of the adult material where we encounter rather extended non-adult scenes and deleted the adult scenes out of them. Finally, the second data set consists of 160 videos with non-adult material (100 most popular, 50 being watched, 10 edited adult material) and 40 videos with explicit content.

[1] `http://www.youtube.com`

[2] `http://www.youporn.com`

Character of Video	Classification Rule	Classification Accuracy
Adult Material	$x < 0.08, y > 0.55$	0.85
Suspicious	$x < 0.08, y > 0.43$	-
Portrait	$x \gg 0.08, y < 0.25$	1

Table 6.1: The 3 main characters of videos that can be extracted from the skin paths reliably and their classification performance on the annotated data set.

Adult Video Classification and Detection

In Figure 6.8, the skin paths for the first dataset of 25 videos are given. As is shown by the red paths, adult material tends to have few and small faces compared to a large amount of skin present. This intuitive criteria is well suited for classification of videos in the skin path diagram: We make the assertion that the skin path of suspicious video material enters the area defined by $x < 0.08$ and $y > 0.55$. We classify our whole dataset correctly with two false positive detections of videos. These two contain desert shots without faces present. With this classification technique, we can detect adult material reliably with the tendency to get false positive detections but very few false negative ones. Additionally, the distance to the upper left corner gives an idea of the character of the scene: Video 8 and 21 sporting women in bikinis have a path beginning below $x = 0.5$, $y = 0$ (and therefore near adult material) and smoothly adapt themselves towards the lower right (towards the unsuspicious space). Videos without much non-facial skin visible (e.g. Interviews) have skin paths significantly towards the bottom. The main areas are summarized in Table 6.1. Adult material is the zone where we encounter adult material to be flagged, suspicious videos tend to show persons in full with lots of visible skin as they appear e.g. in sports clips. 5 videos are classified as suspicious material. They are intuitively correct as they are beach, dance and massage scenes. We separate videos with portrait shots as there are in news, interviews and most of the “webcam” video messages robustly into the portrait area with a classification accuracy (correctly classified videos divided by total videos) of 1. We show in the next section that these values hold for arbitrary on-line videos as well.

Flagging Adult On-line Videos

We repeat the experiment on the 200 on-line videos. As can be seen in Figure 6.9, adult material has again a strong trend towards the upper left corner. We apply the classification rules defined for adult material and reach an accuracy of 0.91. The two false negatives are both classified as suspicious character, which can be seen in the second line of Table 6.2. The reason for this wrong classification is in the nature of the two videos: The first one is of explicit adult material, but almost no skin and no face is visible as the actors are dressed fully. The other false positive contains a couple that apparently takes the video with their web-cam. In contrast to all other adult videos, the actors appear rather small in the image. Although the skin color is detected precisely, the amount of skin is not enough for our adult material classification rule. 54 videos are classified as portrait

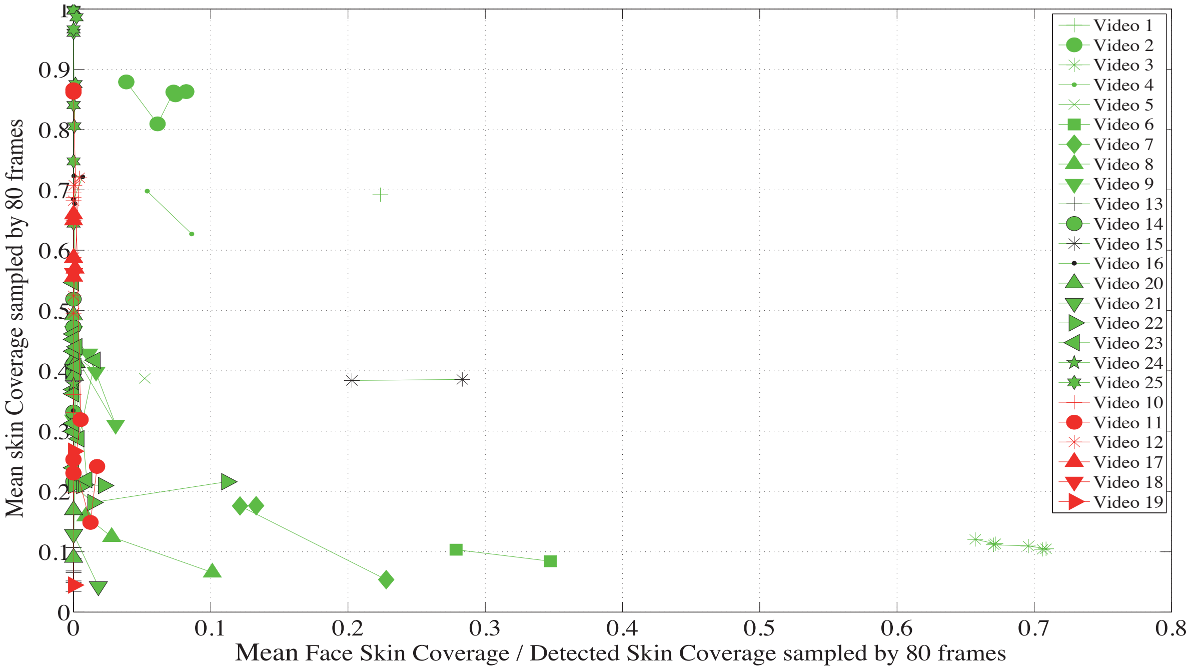


Figure 6.8: Skin paths of the relation between facial skin pixels and other skin pixel drawn onto the overall skin coverage. Red indicates adult material, green unsuspicious video material.

Classification Rule	Classification Accuracy	# False Negatives	# False Positives
$x < 0.08, y > 0.55$	0.91	2	15
$x < 0.08, y > 0.43$	0.82	0	37

Table 6.2: Classification accuracy and absolute number of false negatives and false positives. The first line shows flagging performance of adult material, second line for both adult and suspicious material.

videos. They all contain portrait shots. We do not encounter false positive detections of tracked faces, although we do not know precisely how many were missed.

6.3 Summary

In this chapter, we presented a real-time practical approach for skin detection in videos. Instead of using solely color information, we integrated the available contextual information through face detection and combined face tracking. By using a combination of face detectors and an adaptive multiple model approach to dynamically adapt skin color decision rules, we are able to significantly reduce the number of false positive detections. The classification results become more reliable compared to static color threshold based

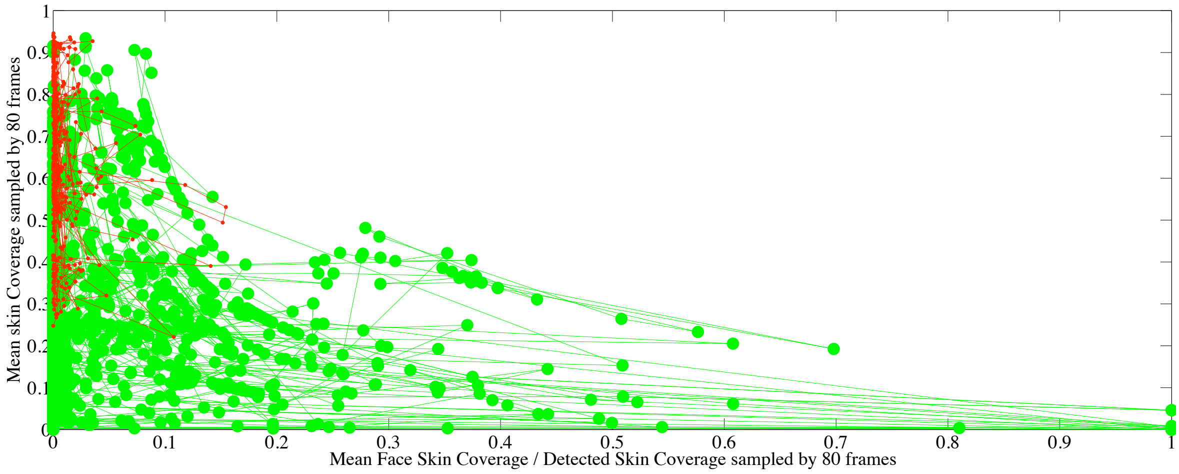


Figure 6.9: Skin paths of the relation between facial skin pixels and other skin pixel drawn onto the overall skin coverage. Red indicates adult material, green unsuspicious video material.

approaches and can be carried out in parallel. For objectionable content filtering, we introduced skin paths as a compact and powerful representations of videos. With the reliable features from facial and non-facial skin, we successfully classified on-line videos. The approach is computationally inexpensive and the number of false negatives is very low, providing a reliable system for flagging adult material.

Chapter 7

Conclusions

In this thesis, we have explored skin detection methods in general, focusing on the usage of contextual information in particular. As a simple and fast method, we introduced two new static filters based on two chrominance components in IHLS and CIELAB color spaces. Using classifiers and color spaces, we showed that (1) the cylindrical color spaces outperform other color spaces, (2) the absence of the illuminance component decreases performance, (3) an appropriate skin color modeling approach selection is important for pixel based skin classification and that the tree based classifiers (Random forest and J48) are well suited to pixel based skin detection. We investigated the use of color constancy algorithms for skin detection and found that when using classifiers, skin classification can be improved with the introduction of lighting correction. As "fusion" of different color spaces for skin detection, the non-perfect correlation between the color space channels is exploited by learning weights based on an optimization for a particular color space channel using the mathematical financial model of Markowitz. With graph cut approach, we proposed a concept for processing arbitrary images using the "universal seed", thereby providing basis for general skin segmentation, exploiting the spatial relationship among the neighboring skin pixels. We presented a "systematic" approach for skin segmentation with graph cuts by using local skin information, the universal seed based skin segmentation and skin augmentation using an off-line learned model, thus providing a basis for merging spatial and non-spatial data. We proposed real-time skin detection using a "multiple model" approach for videos taking advantage of the contextual information for varying illumination circumstances and a variety of skin colors from person to person. We presented the usage of skin detection for flagging videos as potentially objectionable due to sexual content of an adult nature. The "skin path" provides summarization of a video in the form of a path in a skin-face plot, allowing potentially objectionable segments of videos to be found.

The comprehensive skin detection study presented should, in combination with other cues, enable robust face detection, hand detection and blocking objectionable content in unconstrained environments.

Dataset	Static Filter**	Classifier*	Classifier**	Universal Seed**	Markowitz**	Systematic**	Multiple Model**
DS1	0.51	0.730	0.62	0.55	0.55	**0.68**	0.55
DS2	0.50	0.735	0.66	0.54	0.51	**0.71**	0.45

Table 7.1: Result overview: Maximum of approaches in this thesis. (*) indicates 10-fold cross validation, where as (**) indicates evaluation based on training/test sets. Based on training and test sets, the systematic approach provides the maximum performance compared to all the approaches on both datasets.

7.1 Result Overview

In this section, an overview of the results obtained in this thesis is presented. Table 7.1 summarizes and reports the maxima of different approaches.

Static Filters: On DS1, the maximum F-measure of 0.509 is reported by the CIELAB static filter. On DS2, the maximum F-measure of 0.50 is reported by the normalized-RGB static filter.

Classification: With classifiers, we have two modes of evaluation, 10-fold cross validation and training/test sets:

10-fold Cross Validation: For the dataset DS1, the maximum F-measure of 0.73 is achieved by the IHLS color space with the random forest and with the J48, the IHLS color space achieves the second highest F-measure of 0.68. For the dataset DS2, the maximum F-measure of 0.735 is achieved by the CIELAB color space with random forest and with J48, CIELAB achieves the second highest F-measure of 0.70. In classification mode, it is observed that dropping an illuminance component decreases performance.

Training/test sets: Based on training and test sets, the Random forest achieves an F-measure (pixel based) of 0.62 on DS1 and 0.66 on DS2. This is the highest performance we get with classifiers.

Universal Seed: Based on training and test sets, the universal seed achieves an F-measure (pixel based) of 0.55 on DS1 and 0.539 on DS2.

Markowitz Color Fusion: For the fusion of color space channels approach, we get an F-measure of 0.55 for DS1 and 0.509 for DS2. By using 19 color channels, our objective was to see the role of different color channels in skin detection. We argue that increasing the number of color channels makes the model complex and thus good performance can be achieved by decreasing the number of color channels.

Systematic Approach: Using the systematic approach for skin detection (integration of face detection, universal seed and classifier), we get an F-measure of 0.68 for DS1 and 0.71 for DS2. For both DS1 and DS2, this is the highest F-measure achieved of the all the approaches discussed in this thesis based on training/test sets.

Multiple Model Approach: The purpose of such an approach is robust skin detection in videos. The F-measure of 0.45 of the YCbCr static filter is increased to 0.55 for DS1 and F-measure of 0.33 is increased to 0.45 for DS2, using this strategy.

7.2 Recommendations

The following recommendations are presented for a skin detection user. If real-time skin detection is the main concern, then static filters with contextual information are useful, as demonstrated using the multiple models approach. The performance of these static filters can be increased by using color constancy for a specific constrained environment. If on the other hand, there is no real-time limitation, then we recommend the systematic approach (detailed in Section 5.3), which uses contextual skin information, global skin information and integration of external classifiers, achieving maximum performance. If the objective is to visually display the skin detection results, we recommend using the graph cut approach because it takes into account the spatial neighborhood of the pixels and produces more connected blobs of skin. This approach can be used with the probabilities from any external learned model as demonstrated, producing more stable and connected blobs with the J48 classifier results. If a skin detection system has to be deployed for a specific purpose under a constrained environment (for example, the availability of less labeled data), then we recommend universal seed and Markowitz color fusion approaches which can be trained with only positive training data, requiring no time consuming training. If a skin detection system has to be constructed based solely on classifiers (availability of sufficient labeled data), we recommend J48 and Random forest and we observe that J48 produces more connected blobs than the Random forest. For using skin detection as a cue for blocking objectionable content, we recommend skin graph and skin path approaches, giving a compact representation of videos.

7.3 Future Perspectives

Regarding static filters, an adaptation to the changing environment using local skin information in terms of faces is presented. If there are no faces, we do not adapt to changing lighting conditions and one future work will be applicable to images in general situation capable of adaptively adjusting the skin detection thresholds. Regarding the usage of general color constancy algorithms, one future direction will be to come up with a more targeted color constancy algorithm for skin i.e. an approach which is trained on human subjects under different lighting conditions and can estimate skin color under varying illumination conditions and appropriately adjust skin detection parameters.

Although we have demonstrated a real-time skin detection system using static skin filter, a more robust system will be to extend the graph cut based approach for real-time. In videos, one can exploit segmentation from one image to other images and thus we do not need to compute all of the computationally expensive operations for consecutive frames. For videos, we presented an approach for blocking objectionable content, however, for still images the approach needs more work as there is only one image available on which to take a decision.

Bibliography

[1] Alberto Albiol, Luis Torres, and Edward J. Delp. Optimum color spaces for skin detection. In *Proceedings of the ICIP*, pages 122–124, 2001.

[2] R. R. Anderson and J. A. Parrish. The optics of human skin. *The Journal of investigative dermatology*, 77(1):13–19, July 1981.

[3] Antonis A. Argyros and Manolis I.A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *ECCV*, pages 368–379, 2004.

[4] Gladimir V. G. Baranoski and Aravind Krishnaswamy. Light interaction with human skin: from believable images to predictable models. In *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses*, pages 1–80, New York, NY, USA, 2008. ACM.

[5] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms. I: Methodology and experiments with synthesized data. *Image Processing, IEEE Transactions on*, 11(9):972–984, 2002.

[6] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1 edition, January 1996.

[7] Silvio Borer and Sabine Suesstrunk. Opponent color space motivated by retinal processing. In *International Conference on Color in Graphics, Imaging and Vision (CGIV)*, pages 187–189, 2002.

[8] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV 2001*, volume 1, pages 105–112 vol.1, 2001.

[9] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

[10] J. Brand and J.S. Mason. A comparative assessment of three approaches to pixel-level human skin-detection. In *ICPR*, volume 1, pages 1056–1059, 2000.

[11] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[12] J. D. Bronzino. *Biomedical Engineering Fundamentals (The Biomedical Engineering Handbook, Third Edition).* CRC Press, 3rd edition, April 2006.

[13] D. Brown, I. Craw, and J. Lewthwaite. A SOM based approach to skin detection with application in real time systems. In *BMVC'01*, pages 491–500, 2001.

[14] G Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310:1 – 26, 1980.

[15] T.S. Caetano and D.A.C. Barone. A probabilistic model for the human skin color. In *International Conference on Image Analysis*, pages 279–283, 2001.

[16] J. Cai and A. Goshtasby. Detecting human faces in color images. *Image and Vision Computing*, 18:63–75, 1999.

[17] Liang-Liang Cao, Xue-Long Li, Neng-Hai Yu, and Zheng-Kai Liu. Naked people retrieval based on adaboost learning. In *International Conference on Machine Learning and Cybernetics*, pages 1133–1138, 2002.

[18] Mats Carlin. Radial basis function networks and nonlinear data modelling. In *Neural Networks and their Applications*, volume 1, pages 623–633, 1992.

[19] George Casella and Edward I. George. Explaining the Gibbs Sampler. *The American Statistician*, 46(3):167–174, 1992.

[20] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Int. Conf. Internet Measurement*, pages 1–14, 2007.

[21] D. Chai and K.N. Ngan. Locating facial region of a head-and-shoulders color image. In *Int. Conf. Automatic Face and Gesture Recognition*, pages 124–129, 1998.

[22] C. Chen and S.-P. Chiang. Detection of human faces in colour images. *Vision, Image and Signal Processing, IEE Proceedings*, 144(6):384 –388, dec. 1997.

[23] Jie Cheng and Russell Greiner. Comparing bayesian network classifiers. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 101–110, San Francisco, CA, 1999. Morgan Kaufmann.

[24] Kyung-Min Cho, Jeong-Hun Jang, and Ki-Sang Hong. Adaptive skin-color filter. *Pattern Recognition*, 34(5):1067 – 1073, 2001.

[25] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cedric Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[26] Motonori Doi and Shoji Tominaga. Spectral estimation of human skin color using the kubelka-munk theory. *Color Imaging VIII: Processing, Hardcopy, and Applications*, 5008(1):221–228, 2003.

[27] Graham D. Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color Imaging Conference*, pages 37–41, 2004.

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

[29] L. R. Ford and D. R. Fulkerson. *Flows in Networks.* Princeton University Press, 1962.

[30] Y. Freund and R. Schapire. A short introduction to boosting. *Artificial Intelligence*, 14(5):771–780, 1999.

[31] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Mach. Learn.*, 29:131–163, November 1997.

[32] Zhouyu Fu, Jinfeng Yang, Weiming Hu, and Tieniu Tan. Mixture clustering using multidimensional histograms for skin detection. In *ICPR*, pages 549–552, Washington, DC, USA, 2004.

[33] C. Garcia and G. Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Transactions on Multimedia*, 1(3):264–277, Sep 1999.

[34] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 06 2008.

[35] T. Gevers and A. Smeulders. Color based object recognition. *Pattern Recognition*, 32:453–464, 1997.

[36] Moheb R. Girgis, Tarek M. Mahmoud, and Tarek Abd-El-Hafeez. An approach to image extraction and accurate skin detection from web pages. In *Proceedings of World Academy of Science, Engineering and Technology*, pages 367–375, 2007.

[37] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, October 1988.

[38] G. Gomez, M. Sanchez, and Luis Enrique Sucar. On selecting an appropriate colour space for skin detection. In *MICAI '02: Proceedings of the Second Mexican International Conference on Artificial Intelligence*, pages 69–78, London, UK, 2002. Springer-Verlag.

[39] Giovani Gomez. On selecting colour components for skin detection. *Pattern Recognition, International Conference on*, 2:961 – 964, 2002.

[40] Giovani Gomez and Eduardo F. Morales. Automatic feature construction and a simple rule induction algorithm for skin detection. In *ICML*, pages 31–38, 2002.

[41] Hayit Greenspan, Jacob Goldberger, and Itay Eshet. Mixture model for face-color modeling and segmentation. *Pattern Recognition Letters*, 22(14):1525 – 1536, 2001.

[42] Allan Hanbury. A 3d-polar coordinate colour representation well adapted to image analysis. In *SCIA*, pages 804–811, 2003.

[43] E. Hjelmas and B. K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, pages 236–274, September 2001.

[44] Tin Kam Ho. Random decision forests. In *ICDAR*, pages 278–282, 1995.

[45] G. Holmes, A. Donkin, and I.H. Witten. Weka: a machine learning workbench. In *Second Australian and New Zealand Conference on Information Systems*, pages 357–361, nov-2 dec 1994.

[46] R.L. Hsu, M. Abdel-Mottaleb, and A.K. Jain. Face detection in color images. *PAMI*, 24:696–706, 2002.

[47] Quan Huynh-Thu, Mitsuhiko Meguro, and Masahide Kaneko. Skin-color extraction in images with complex background and varying illumination. *Applications of Computer Vision, IEEE Workshop*, pages 280–290, 2002.

[48] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[49] Bruno Jedynak, Huicheng Zheng, and Mohamed Daoudi. Statistical models for skin detection. In *Computer Vision and Pattern Recognition Workshop*, volume 8, pages 92–92, jun. 2003.

[50] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *IJCV*, 46(1):81–96, 2002.

[51] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *PR*, 40(3):1106–1122, 2007.

[52] P. Kakumanu, S. Makrogiannis, R. Bryll, S. Panchanathan, and N. Bourbakis. Image chromatic adaptation using ANNs for skin color adaptation. In *IEEE International Conference on Tools with Artificial Intelligence*, pages 478 – 485, nov. 2004.

[53] J. Karlekar and U.B. Desai. Finding faces in color images using wavelet transform. In *International Conference on Image Analysis and Processing*, pages 1085 –1088, 1999.

[54] W. Kelly, A. Donnellan, and D. Molloy. Screening for objectionable images: A review of skin detection techniques. In *Machine Vision and Image Processing Conference, 2008. IMVIP '08. International*, pages 151 –158, 2008.

[55] Rehanullah Khan, A. Hanbury, and J. Stöttinger. Weighted skin color segmentation and detection using graph cuts. In *Proceedings of the 15th Computer Vision Winter Workshop*, pages 60–68, February 2010.

[56] Rehanullah Khan, Allan Hanbury, and Julian Stoettinger. Skin detection: A random forest approach. In *ICIP*, pages 4613 – 4616, 2010.

[57] Rehanullah Khan, Allan Hanbury, and Julian Stöttinger. Augmentation of skin segmentation. In *International Conference on Image Processing, Computer Vision, and Pattern Recognition*, pages 473–479, 2010.

[58] Rehanullah Khan, Allan Hanbury, and Julian Stöttinger. Universal seed skin segmentation. In *International Symposium on Visual Computing*, pages 75–84, 2010.

[59] Rehanullah Khan, Julian Stöttinger, and Martin Kampel. An adaptive multiple model approach for fast content-based skin detection in on-line videos. In *ACM MM, AREA workshop*, pages 89–96, 2008.

[60] Sang-Hoon Kim, Nam-Kyu Kim, Sang Chul Ahn, and Hyoung-Gon Kim. Object oriented face detection using range and color information. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 76 –81, apr. 1998.

[61] Peter E. Kloeden and Eckhard. Platen. *Numerical solution of stochastic differential equations (Stochastic Modelling and Applied Probability)*. Springer-Verlag, Berlin, New York, 1992.

[62] Jae Young Lee and Suk Yoo. An elliptical boundary model for skin color detection. In *ISST*, pages 579–584, 2002.

[63] Jiann-Shu Lee, Yung-Ming Kuo, Pau-Choo Chung, and E-Liang Chen. Naked image detection based on adaptive and extensible skin color model. *PR*, 40(8):2261–2270, 2007.

[64] David D. Lewis. Naive (Bayes) at forty: The independence assumption in Information Retrieval. In *ECML*, pages 4–15, 1998.

[65] Rainer Lienhart. Comparison of Automatic Shot Boundary Detection Algorithms. In *Proc. IS&T/SPIE Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 290–301, 1999.

[66] Christian Liensberger, Julian Stöttinger, and Martin Kampel. Color-based and context-aware skin detection for online video annotation. In *MMSP*, pages 1–6, 2009.

[67] Jitendra Malik, Serge Belongie, Thomas K. Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[68] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[69] Elzbieta Marszalec, Birgitta Martinkauppi, Maricor Soriano, and Matti Pietikainen. Physics-based face database for color research. *Journal of Electronic Imaging*, 9(1):32–38, 2000.

[70] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.

[71] Aleix Martínez and R. Benavente. The ar face database. Technical Report 24, Computer Vision Center, Bellatera, Jun 1998.

[72] J. Birgitta Martinkauppi, Maricor N. Soriano, and Mika V. Laaksonen. Behavior of skin color under varying illumination seen by different cameras at different color spaces. *Machine Vision Applications in Industrial Inspection*, 4301(1):102–112, 2001.

[73] Stephen J. McKenna, Shaogang Gong, and Yogesh Raja. Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition*, 31(12):1883 – 1892, 1998.

[74] Branislav Micusík and Allan Hanbury. Steerable semi-automatic segmentation of textured images. In *SCIA*, pages 35–44, 2005.

[75] Branislav Micusík and Allan Hanbury. Supervised texture detection in images. In *CAIP*, pages 441–448, 2005.

[76] Thomas B. Moeslund. Computer vision based motion capture of body language. PhD Thesis, Alborg University, Denmark, 2003.

[77] A. Nayak and S. Chaudhuri. Self-induced color correction for skin tracking under varying illumination. In *ICIP*, volume 3, pages 1009–1012, sep. 2003.

[78] N. Oliver, A.P. Pentland, and F. Berard. Lafter: lips and face real time tracker. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 123–129, 17-19 Jun 1997.

[79] Vladimir Pavlovic. Boosted detection of objects and attributes. In *CVPR*, pages 1–8, 2001.

[80] Peter Peer, Jure Kovac, and Franc Solina. Human skin colour clustering for face detection. In *EUROCON*, pages 144–148, vol.2, 2003.

[81] S. L. Phung, D. Chai, and A. Bouzerdoum. A universal and robust human skin color model using neural networks. In *IJCNN*, pages 2844–2849, 2001.

[82] Ross J. Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., 1993.

[83] P. De Los Rios, S. Lise, and A. Pelizzola. Bethe approximation for self-interacting lattice trees. *EPL (Europhysics Letters)*, 53(2):176, 2001.

[84] Charles R. Rosenberg, Thomas P. Minka, and Alok Ladsariya. Bayesian color constancy with non-gaussian models. In *NIPS*, pages 1–8, 2003.

[85] D.E. Rumelhart, G.E. Hintont, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[86] Hichem Sahbi and Nozha Boujemaa. Coarse to fine face detection based on skin color adaption. In *ECCV '02: Proceedings of the International ECCV 2002 Workshop Copenhagen on Biometric Authentication*, pages 112–120, London, UK, 2002. Springer-Verlag.

[87] Janos Schanda. *Colorimetry: Understanding the CIE System.* Wiley, Inc., 2007.

[88] Stephen J. Schmugge, Sriram Jayaram, Min C. Shin, and Leonid V. Tsap. Objective evaluation of approaches of skin detection using ROC analysis. *Computer Vision and Image Understanding*, 108(1-2):41 – 51, 2007.

[89] K. Schwerdt and J.L. Crowley. Robust face tracking using color. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 90 –95, 2000.

[90] Nicu Sebe, Ira Cohen, Thomas S. Huang, and Theo Gevers. Skin detection: A Bayesian network approach. In *ICPR*, pages 903–906, 2004.

[91] Ming-Jung Seow, D. Valaparla, and V.K. Asari. Neural network based skin color model for face detection. In *Applied Imagery Pattern Recognition Workshop*, pages 141 – 145, oct. 2003.

[92] Hsieh I. Sheen, Fan K. Chin, and Chiunhsiun Lin. A statistic approach to the detection of human faces in color nature scene. *PR*, 35(7):1583 – 1596, 2002.

[93] Min C. Shin, Kyong I. Chang, and Leonid V. Tsap. Does colorspace transformation make any difference on skin detection? In *In IEEE Workshop on Applications of Computer Vision*, pages 275–279, 2002.

[94] L. Sigal, S. Sclaroff, and V. Athitsos. Skin color-based video segmentation under time-varying illumination. *PAMI*, 26(7):862–877, July 2004.

[95] Leonid Sigal, Stan Sclaroff, and Vassilis Athitsos. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In *CVPR*, pages 152–159, 2000.

[96] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen. Skin detection in video under changing illumination conditions. In *ICPR*, volume 1, pages 839–842, 2000.

[97] Maricor Soriano, Birgitta Martinkauppi, Sami Huovinen, and Mika Laaksonen. Adaptive skin color modeling using the skin locus for selecting training pixels. *Pattern Recognition*, 36(3):681 – 690, 2003.

[98] Harro Stokman and Theo Gevers. Selection and fusion of color models for feature detection. In *Proceedings of the CVPR*, pages 560–565, Washington, DC, USA, 2005. IEEE Computer Society.

[99] Harro Stokman and Theo Gevers. Selection and fusion of color models for image feature detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):371–381, 2007.

[100] M. Störring, H.J. Andersen, and E. Granum. Estimation of the illuminant colour from human skin colour. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 64–69, 2000.

[101] Moritz Störring. Phd thesis: Computer vision and human skin color. PhD Thesis, Alborg University, Denmark, 2004.

[102] Moritz Störring, Tomás Kočka, Hans J. Andersen, and Erik Granum. Tracking regions of human skin through illumination changes. *Pattern Recogn. Lett.*, 24(11):1715–1723, 2003.

[103] Julian Stöttinger, Allan Hanbury, Christian Liensberger, and Rehanullah Khan. Skin paths for contextual flagging adult videos. In *International Symposium on Visual Computing*, pages 303–314, 2009.

[104] Priva Talreja, Gerald Kasting, Nancy Kleene, William Pickens, and Tsuo-Feng Wang. Visualization of the lipid barrier and measurement of lipid pathlength in human stratum corneum. *Journal of the American Association of Pharmaceutical Scientists*, 3:48–56, 2001.

[105] Jean-Christophe Terrillon and Shigeru Akamatsu. Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images. In *Proceedings of the 12th Conference on Vision Interface*, pages 180–187, 2000.

[106] Sofia Tsekeridou and Ioannis Pitas. Facial feature extraction in frontal views using biometric analogies. In *EUSIPCO*, pages 315–318, 1998.

[107] J. van de Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Processing*, 16(9):2207–2214, 2007.

[108] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer, 1995.

[109] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreev. A survey on pixel-based skin color detection techniques. In *GraphiCon*, pages 85–92, 2003.

[110] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition*, 1:I–511–I–518, 2001.

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

[112] K.W. Wong, K.M. Lam, and W.C. Siu. A robust scheme for live detection of human faces in color images. *Signal Processing: Image Communication*, 18(2):103–114, 2003.

[113] Jie Yang, Weier Lu, and Alex Waibel. Skin-color modeling and adaptation. In *ACCV*, pages 687–694, 1997.

[114] Jie Yang and A. Waibel. A real-time face tracker. In *IEEE Workshop on Applications of Computer Vision (WACV '96)*, pages 142–147, Washington, DC, USA, 1996. IEEE Computer Society.

[115] Ming Yang and Narendra Ahuja. Detecting human faces in color images. In *ICIP*, volume 1, pages 127 –130, October 1998.

[116] Minghsuan Yang and Narendra Ahuja. Gaussian mixture model for human skin color and its application in image and video databases. In *SPIE*, pages 458–466, 1999.

[117] Paul Yee and Simon Haykin. A dynamic regularized radial basis function network for nonlinear, nonstationary time series prediction. *IEEE Transactions on Signal Processing*, 47:2503–2521, 1998.

[118] Boon-Lock Yeo and Bede Liu. Rapid scene analysis on compressed video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 5(6):533 –544, December 1995.

[119] Jau yuen Chen, Cuneyt Taskiran, Alberto Albiol, Charles Bouman, and Edward Delp. Vibe: A video indexing and browsing environment. In *Proceedings of the SPIE Conference on Multimedia Storage and Archiving Systems IV*, pages 148–164, 1997.

[120] Ramin Zabih, Justin Miller, and Kevin Mai. A feature-based algorithm for detecting and classifying scene breaks. In *ACM international conference on Multimedia*, pages 189–200, New York, NY, USA, 1995. ACM.

[121] Benjamin D. Zarit, Boaz J. Super, and Francis K. H. Quek. Comparison of five color models in skin pixel classification. In *RATFG-RTS '99: Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 58–63, Washington, DC, USA, 1999. IEEE Computer Society.

[122] Huicheng Zheng, Mohamed Daoudi, and Bruno Jedynak. Blocking adult images based on statistical skin detection. *ELCVIA*, 4(2):1–14, 2004.

[123] Xiaojin Zhu, Jie Yang, and A. Waibel. Segmenting hands of arbitrary color. In *IEEE International Conference on Automatic Face and Gesture Recognition.*, pages 446 –453, 2000.