Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2016

Automatic Detection and Classification of Permanent and Non-Permanent Skin Marks

Armand Moulis

LINKÖPING
UNIVERSITY

Master of Science Thesis in Electrical Engineering

Automatic Detection and Classification of Permanent and Non-Permanent Skin Marks

Armand Moulis

LiTH-ISY-EX--YY/NNNN--SE

Supervisor: **Martin Danelljan, ISY, Linköpings universitet**

Examiner: **Lasse Alfredsson, ISY, Linköpings universitet**

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2016 Armand Moulis

Sammanfattning

När forensiker försöker identifiera förövaren till ett brott använder de individuella ansiktsmärken när de jämför den misstänkta med förövaren. Ansiktsmärken används ofta vid identifikation och de lokaliseras idag manuellt. För att skynda på denna process, är det önskvärt att detektera ansiktsmärken automatiskt. Detta examensarbete beskriver en metod för att automatisk detektera och separera permanenta och icke-permanenta märken. Den använder en snabb radial symmetri algoritm som en huvuddel i detektorn. Efter att kandidater av ansiktsmärken har tagits fram tas alla falska detektioner bort utifrån deras storlek, form och hårinnehåll. Resultaten visar att detektorn har en god känslighet men dålig precision. Eliminationsmetoderna av falska detektioner analyserades och olika kännetecken användes till klassificeraren. Det kan fastställas att färgen på ansiktsmärkena har en större betydelse än formen när det gäller att sortera dem i permanenta och icke-permanenta märken.

Abstract

When forensic examiners try to identify the perpetrator of a felony, they use individual facial marks when comparing the suspect with the perpetrator. Facial marks are often used for identification and they are nowadays found manually. To speed up this process, it is desired to detect interesting facial marks automatically. This master thesis describes a method to automatically detect and separate permanent and non-permanent marks. It uses a fast radial symmetry algorithm as a core element in the mark detector. After candidate skin mark extraction, the false detections are removed depending on there size, shape and number of hair pixels. The classification of the skin marks are done with a support vector machine and the different features are examined. The results shows that the facial mark detector has a good recall value while the precision is poor. The elimination methods of false detection was analysed as well as the different features for the classifier. One can conclude that the color of facial marks is more relevant than the structure when classifying them into permanent and non-permanent marks.

Acknowledgments

I would like to thank the supervisors at NFC in Linköping for their support and guidance during this master thesis. I also want to thank Lasse Alfredsson and Martin Danelljan at Linköpings university for their help during the master thesis process. A special thanks goes to my fiancée for her unconditional love and support through my work.

Linköping, February 28, 2017
Armand Moulis

Contents

Notation

Mathematical expression

Notation	Meaning
$*$	Convolution
$\bullet$	Dot product
$\|\|a\|\|$	Norm of a vector
$\hat{a}$	Normalized vector
a^T	Transpose of a vector
$\text{round}(x)$	Rounds x to nearest integer

Abbreviations

Abbreviation	Meaning
NFC	National Forensic Centre
RPPVSM	Relatively Permanent Pigmented or Vascular Skin Marks
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
LoG	Laplacian of Gaussian
RGB	Red Green Blue
HSV	Hue Saturation Value
LRSR	Light Random Sprays Retinex
FRS	Fast Radial Symmetry
RBF	Radial Basis Function

1 Introduction

Recently, the advancements in image analysis and computer vision provide many tools for forensics. One of the most promising tools are automated person identification which can help judicial system. The person identification is dependent on good facial features such as skin marks. This is the reason why this master thesis will investigate how to best detect and classify facial skin marks.

1.1 Background

The work of systematically recording physical measurements for law enforcement was introduced by Alphonse Bertillon as early as in the 19th century. He developed the Bertillonage system since he believed that each person could be uniquely identified by a set of measurements [1]. This system was however outdated quickly due to the rapid advancements in technology.

Today, the amount of video surveillance cameras, security cameras and cellphone cameras increases rapidly and there exist millions of devises capable of catching perpetrators in the act. The videos and still images can be used as evidence for identification during trials where forensic experts evaluate the strength of evidence whether if the suspect is the same person as the one caught on camera.

One common method of evaluating whether the perpetrator and the suspect are the same person is to compare facial features such as eyes, nose, mouth, scars, and other facial marks. This is nowadays done manually [48] by the forensic examiners, and in order to evaluate the strength of the results, a likelihood ratio [34] from Bayes rule is calculated. The likelihood ratio is estimated from two hypotheses, where the numerator gives the probability to achieve the results if the perpetrator and the suspect are the same person. The denominator gives the probability to achieve the results if the perpetrator is another man.

Facial features are divided into two groups: class and individual characteristics [45]. The class characteristics includes traits which put individuals into larger groups. Some of these features are e.g. hair and eye color, overall facial shape and size of the ears. The class characteristics does not suffice to identify unique individuals. Individual characteristics are traits that are unique to an individual, for example the number and location of facial skin marks.

Facial skin marks are any salient skin region that appears on the face. The most common facial marks are moles, pockmarks, freckles, scars, and acne. Some of these marks are not permanent, e.g. acne usually heals without leaving any permanent marks, while scares and moles remain the whole life [33]. Skin marks which can be used for identification need to be relatively permanent, common and also be observable without any special imaging or medical equipment. These relatively permanent marks usually occur due to increased pigmentation or vascular proliferation. Therefore, these kind of facial skin marks are called "relatively permanent pigmented or vascular skin marks (RPPVSM)". [35]

This master thesis will separate facial skin marks into two classes: permanent and non-permanent facial marks. Some examples of the two types can be seen in fig. 1.1. Which class a facial skin mark belong to is decided by the forensics at National Forensic Centre (NFC) i Sweden. NFC is currently running a project where an automatic facial recognition system can be used to extract statistics from a database of facial images. The main advantages of using such a method are that the likelihood ratio can be calculated based on statistics, and that the risk for human bias in the decision process is diminished.

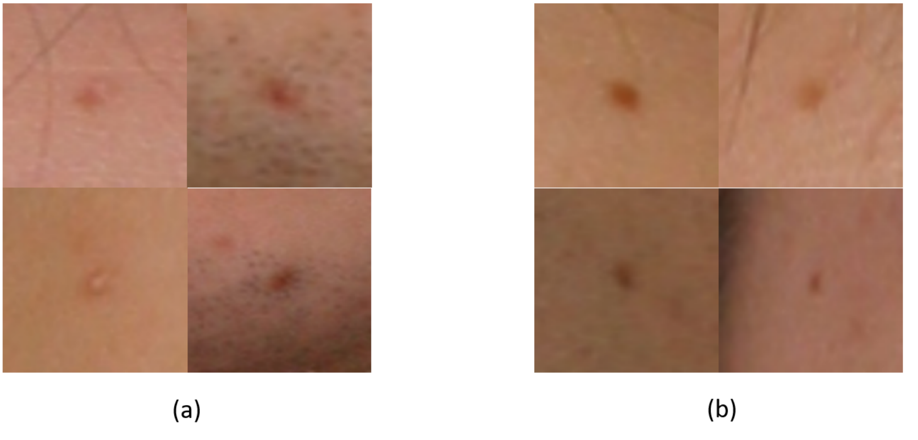

Figure 1.1: *Examples of facials marks: (a) non-permanent, (b) permanent*

This master thesis was started due to the need of combining the automatically calculated likelihood ratio value with the evidential value derived from the frequency of facial marks in certain regions of the face. The NFC is supporting this work by providing guidance and practical help.

1.2 Related work

A line of research relevant to this master thesis is the work by Vorder Bruegge et al. [33] which proposed a fully automatic multiscale facial mark system. It detects facial marks which are stable across the RGB-channels and different scales. These scales are called Gaussian pyramid and consist of low-pass filtered and subsampled images of the original image. This method to detect permanent marks are also used by Nisha Srinivas et al. [46] who tries to separate identical twins with an automatic multiscale facial mark detector. This method does not try to separate permanent and non-permanent facial mark rather tries to detect the more permanent marks.

Another option considered when looking for facial marks are object detection and object classification. The research on object detection and object classification is a wide and relevant field. Some of the things researchers have tried to detect and classify are faces [2], pedestrians [14] and vehicles [17]. These examples use descriptive features based on histogram of oriented gradients (HOG) and local binary patterns (LBP). Face detectors also uses Haar-like features [53]. These three sets of features all describes the shape and structure of the searched object.

Taeg Sang Cho et al.[11] proposed a method using a Support Vector Machine (SVM) as a classifier to separate true and false mole candidates. They used a gist-descriptor as descriptive features. The gist-descriptor is designed to describe texture patterns over space. Read more about the gist-descriptor in the work of Antonio Torralba et al.[49].

Another work using classifier are the work from Arfika Nurhudatiana et al. [36]. They tried to detect and separate RPPVSM from non-RPPVSM on back torsos. They tried out three different classifiers which include a SVM, neural network and a binary decision tree. As input, the classifier was given the same set of features which included contrast, shape, size, texture, and color. Tim K. Lee et al.[27] also used the same kind of features but does not use a trained classifier to separate true and false moles on back torsos. They use unsupervised algorithm to classify the mole candidates.

When it comes to the detection of potential skin mark, there often involves some kind thresholding of an edge enhanced images. Using Laplacian of Gaussian (LoG) kernel as edge enhancement is popular method [19, 38]. After the edge enhancement of a image, the skin marks are highlighted and can then be segmented with different thresholding methods.

1.3 Motivation

Many researchers [11, 36, 27] tried to separate skin marks and they used a fixed set of features to do this. Arfika Nurhudatiana et al. compared different classifiers but there has been little work on comparing different set of features to separate permanent and non-permanent skin marks. This is why this thesis work will focus on comparing different features as input to a supervised classier. Since the facial marks have a circular shape and mostly vary in color it would be wise to use colors maps as features.

This master thesis will look at a recently used and interesting method to highlight the skin marks, instead of the common LoG kernel. The algorithm is called fast radial symmetry (FRS) [46, 33] and it highlights radially symmetrical regions and suppresses regions that are asymmetrical. This is ideal when one is looking for circular objects which is perfect since facial marks are often circular. The FRS is expected to be more suitable for detecting skin marks compared to previous approaches, and is therefore investigated in this thesis.

The challenge with detecting skin marks, especially in the face, is that there are many other structures which can be mistaken as facial marks, e.g. nostrils, facial hair. Facial hair in the form of stubble can complicate the problems, as its appearance may be similar to a facial mark. The main challenge of this work is to find characteristic features for the permanent and non-permanent skin marks. They differ little in shape and structure but differ more in color. This master thesis will try to overcome these challenges.

1.4 Aim

The aim of this master thesis is to develop a method for creating a large data base with facial images and the location of facial marks. Such a database would provide better statistics for the evidential value in forensic facial image comparison examinations. The algorithm should detect facial marks automatically from a color image and then separate them into a permanent and non-permanent group.

1.5 Problem specification

From a single facial RGB-image en face, facial marks should be detected and classified as a permanent or non-permanent mark. This task can be divided into five smaller tasks. These tasks will be described more in detail in later chapters.

Task 1: Pre-processing The image can be illuminated unevenly and rotated which can cause difficulties when detection potential facial marks. Thus, the image has to be geometrically and photometrically normalized.

Task 2: Candidate detection The actual detection of potential marks is done with the help of radial symmetry in the image. The algorithm will search for areas which contains edges that have a circular shape.

Task 3: Post-processing Among the potential facial marks, there can be many false detections such as nostrils, facial hair, pupils et cetera. The false detection has to be eliminated and will be done with a hair removal method, blob identifier, size eliminator and face segmentation.

Task 4: Classification When the marks have been detected, they have to be sorted into the two classes, permanent and non-permanent. This is done by calculating different descriptive features. These features are used to train a supervised support vector machine. With the trained classifier, the facial mark can be sorted.

Task 5: Feature selection The major task in this master thesis is to compare and evaluate different descriptive features. This is done by choosing different sets of features for the classifier and evaluate the performance of the classifier for each set.

1.6 Scope

In general, when working with image, the quality of the images is crucial for the results. Low resolution and badly illuminated images taken from different angles can cause analytical difficulties. Therefore, this thesis assumes images which are high resolute, well illuminated, taken en face and in RGB-colors.

Also, this master thesis will focus on a comparison between different sets of features for the classifier instead of examining different ways of detecting facial marks. This is due to the little work done regarding feature selection.

The classifier will only be a binary classifier because no non-facial marks has been collected as labelled data during the thesis work due to lack of resources.

1.7 Thesis outline

This chapter describes the aim and problem specification of this master thesis. In Chapter 2, gives an insight in theory behind the methods used in the algorithm. Chapter 3 describes the pipeline of the algorithm and the implementation of the theory used in it. The results from the algorithm can be studied in Chapter 4 and an discussion about the result and methods used is found in Chapter 5. Finally, Chapter 6 consist of a conclusion of the master thesis and ideas for future work within the same scope.

2 Theory

This chapter will describe the underlying theory about the methods and algorithms used during in the automatic facial mark algorithm.

2.1 Facial landmarks

To process a facial image, it is useful to know where different parts of the face are located, e.g. mouth and eyes. These parts can be pinpointed with points called landmarks. With these landmarks, it is possible to create a unique mask for each face and produce a grid with different regions in the face. The landmarks are extracted by using an implementation based on Vahid Kazemi et al. [22]. It uses sate of the art algorithms for face alignment where cascade of regression functions is crucial for its success. The estimated shape of the face is updated by regressing the shape parameters based on normalized features from the image. The parameters are updated until they converge.

From this algorithm, 68 landmarks are extracted where the eyes, mouth, nose and chin are marked. From these, a mask generated where the nostrils, eyes, throat and background are cut out.

2.2 Image normalization

In order to get a reliable and uniform result in the algorithm, the facial images have to be normalized. There are two kinds of normalization applied on the image, geometric normalization and photometric normalization.

2.2.1 Geometric normalization

The geometric normalization consists of rotation of the image such that the line between the pupils is aligned with the bottom of the image. The rotations angle is calculated with the help of the landmarks at the corners of the eyes. An angle is calculated by averaging an angle from the right and left eye.

Rotation of an image is done by using a affine transformation with homogeneous coordinates [28]. Assume that a point is described as (x, y) in Cartesian coordinates. Then, the point can be transformed into homogeneous coordinates such that the point is described as $(x, y, 1)$. Thanks to this coordinate system, rotation can be expressed as a simple matrix multiplication as in eq. (2.1) where $(x', y', 1)$ are the rotated coordinates for a point. ϕ is the angle which each point is rotated counter clockwise.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.1}$$

The geometric normalization also includes a resize of the image such that the interpupillary distance is 500 pixels. A resizing factor is calculated by taking the fraction between 500 and the number of pixels between the pupils.

2.2.2 Photometric normalization

The photometric normalization is performed by using tone mapping operator based on the work of Nikola Banic et al.[3]. It uses a Light Random Sprays Retinex (LRSR) which is an improvement of the Random Sprays Retinex (RSR)[40]. All tone mapping operator transform pixel intensities depending on its surrounding. The RSR uses a random selection of pixels around the current pixel which decreases computations costs, sampling noise and dependency. The calculations are done on the intensity image of each RGB colour channels. An example of the output from the image normalization can be seen in fig. 3.4.

2.3 Face detection

An important component in the algorithm is the bounding box of the face in each image. It is found by using an OpenCV [9] implementation of object detection by Paul Viola et al. [53]. This face detection algorithm was chosen since it has equivalent positive result as other methods [43, 47]. In addition, it is much faster the other detector. The algorithm from Paul Viola et al. take advantage of three different parts.

The first is part is a new image representation which allows Haar features from each image to be calculated rapidly. The speed is achieved by using integral images instead of the original image.

The second part is the extraction of the most important features through AdaBoosting. It creates a strong classifier by combining the strength of weaker classifiers. A weak classifier is the best threshold for a feature which separates the faces from the non-faces.

The third part is a cascade decision which reduces the computations costs by rejecting potential bounding boxes for the face. A simple classifier is used to determine if the bounding boxes are promising candidates before a more complex classifier is engaged. This is repeated until all classifiers has been passed or if one of the returns a negative result. All bounding boxes which has returned a negative result is rejected immediately.

2.4 Segmentation

When searching for facial mark, hair lines and hair can cause false detection. Therefore, the image has to be segmented so that only skin area is regarded during the search of facial marks. Since interactive segmentation methods are more and more popular [8], it should be beneficial to choose an interactive segmentation method. Carsten Rother et al.[41] compared several popular interactive segmentation methods and also presented their own method, GrabCut. They concluded that GrabCut performs as well as GraphCut [8] with fewer user interactions.

Thus, the segmentation method used for the algorithm is GrabCut which uses Gaussian Mixture Model (GMM) for a color image. GrabCut needs a GMM for a known foreground and one for a known background. The known foreground used e.g. is the cheeks and forehead, is extracted with the help of the landmarks.

After creating GMM:s, an energy function is created so that its minimum correspond to a good segmentation which depends on the given foreground and background. The function is minimized iteratively until a converged segmentation is produced.

The segmentation mask is used to improve the mask created from the landmarks. Now using the improved mask, a well segmented image can be searched for facial marks.

2.5 Fast Radial Symmetry

There are many ways to extract interesting points or marks. One way is to look at the radial symmetry in the image. This method has been used by several researcher [46, 30, 33, 42]. It seems to be a reliable method since the point is to detect small circular shapes, which is what Jan Schier et al.[42] did when they tried to count yeast colonies. This is why the actual mark detector uses an algorithm called Fast Radial Symmetry (FRS) and it was created by Gareth Loy et al.[30].

For each point, p, in an image, the contribution of radial symmetry at radius r is calculated by producing an orientation projection image O_n and a magnitude projection image M_n. n is a specific radius. These images needs to know the so called positively-affecting pixel, $p_+(p)$, and negatively-affecting pixel, $p_-(p)$. To find theses affecting pixels the gradient, g, of the image is needed and it is calculated using a 3x3-Sobel kernel. Since the gradient computations are discrete, it is necessary to average the image with a 3x3 Gaussian kernel to remove sharp edges.

$$p_+(p) = g(p) + \text{round}\frac{g(p)}{||g(p)||}n \tag{2.2}$$

$$p_-(p) = g(p) - \text{round}\frac{g(p)}{||g(p)||}n \tag{2.3}$$

To retrieve the nearest integer the operation $round$ is used. The O_n and M_n are then updated according to eqs. (2.4) to (2.7)

$$O_n(p_+) = O_n(p_+) + 1 \tag{2.4}$$

$$O_n(p_-) = O_n(p_-) - 1 \tag{2.5}$$

$$M_n(p_+) = M_n(p_+) + ||g(p)|| \tag{2.6}$$

$$M_n(p_-) = M_n(p_-) - ||g(p)|| \tag{2.7}$$

The radial symmetry contribution at radius n depends on F_n and A_n which is defined as

$$F_n = \frac{M_n(p)}{k_n}\left(\frac{|\tilde{O}_n(p)|}{k_n}\right)^\alpha \tag{2.8}$$

$$\tilde{O}_n(p) = \begin{cases} O_n(p) & \text{if } O_n(p) < k_n \\ 0 & \text{else} \end{cases} \tag{2.9}$$

A_n is a Gaussian kernel with different size depending on n, α is radial strictness parameter and k_n is a scaling factor. α is set to 2 and k_n to 9.9 since Gareth Loy et al. deemed suitable for most applications.

The final radial symmetry image S_n is calculated

$$S_n = F_n * A_n \tag{2.10}$$

This was a calculation for radius n and it desirables to use multiple radii to detect point larger than n. It is not necessary to use a continuous spectrum of radii according to Gareth Loy et al. The average of radial symmetry images, S, are then calculated as in eq. (2.11). The image S is highlighting radial symmetrical regions and suppressing regions that are asymmetrical.

$$S = \frac{1}{N}\sum_{n=1}^{N} S_n \tag{2.11}$$

2.6 Candidate elimination

Since many facial mark candidates may be false positives, they have to be discovered and excluded. Vorder Bruegge et al. [33] used three elimination methods which seemed intuitive. Size, shape and presence of hair should be good indicators if the candidate is a false detection or not. Each detected candidate is given a 30x30 area which is processed through three eliminators.

2.6.1 Blob selection

Facial marks are often blob-shaped which is why the first eliminator uses a simple blob detector from OpenCV. It creates a thresholded images with connective pixels and does this with different threshold values. Each image created with a fixed threshold is put together into image which is the union of all the thresholded images. If the union does not contain a blob-shaped object, the candidate is eliminated. A blob-shaped object is defined by its circularity, inertia and convexity.

2.6.2 Hair elimination

The second eliminator uses a hair removal algorithm by Tim Lee et al. [26]. The algorithm smooth out hair pixels with closing operations using the three different structuring elements. The suggested structuring elements by Tim Lee et al. is larger than the one used in this implementation since their hair-structures where wider. Thus, the smaller structuring elements T_0, T_{45} and T_{90} were used.

$$T_0 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad T_{45} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad T_{90} = (T_0)^T$$

The closed image is generated by applying each structure element on each color channel as (2.12), where G is the closed image, M is the image of a mark, $T_x = [T_0, T_{45}, T_{90}]$ and C the RGB-channels. This means that M_c is a gray image of a mark where the structuring elements detect thin and small edges.

$$G_c = |M_c - \max_x(M_c * T_x)| \tag{2.12}$$

$\max_x(M_c * T_x)$ means that the largest pixel value from the structuring elements are pick for that color channel. In the end, the union of G_c is calculated and to get a hair mask, the union is binary thresholded with h_{hair}. If a region contains more than a certain amount of hair pixels, it will be excluded.

2.6.3 Size elimination

The third eliminator removes candidates depending on their size. If the candidate has an area smaller than 20 pixels or an area larger than 1000 pixels, it is eliminated. The thresholds were chosen because all annotated marks is within this interval, see fig. 2.1.

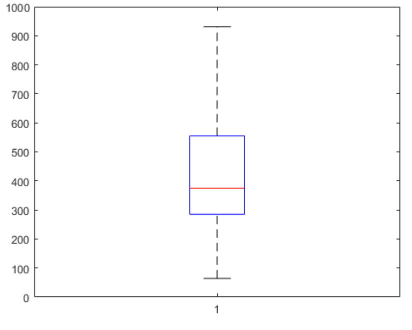

Figure 2.1: *The distribution of areas from the annotated facial marks.*

2.7 Machine learning

Machine learning is a very popular way of determining the future or sorting object into groups, e.g. whether forecast and spam filtering. The field of machine learning is growing quickly and new and more accurate methods are developed constantly. The principle is to use data to predict the outcome. The data can be somewhat incomprehensible when the dimension is getting large and abstract. This is when computers can ease the prediction by analyzing and finding patterns in the data.

Machine learning is divided into three groups [5].

- Supervised learning
  The system has access to labeled data from which it can find patterns and structures
- Unsupervised learning
  The system does not have access to labeled data.
- Reinforcement learning
  The system learns from feedback given to it in form of rewards and punishments.

A learned system can in turn be divided into two groups.

- Classification
  The system tries to determine the class which an object belong to, e.g. spam filtering
- Regression
  The system tries to predict value from an input, e.g. predicting the temperature

This master thesis will only focus on supervised learning and classification since the desired output is permanent or non-permanent mark and there are labeled facial mark are available.

2.7.1 Supervised learning

Supervised learning is when one try to find a function g that maps $X \to \Omega$. X is a vector with N samples with M descriptive features eq. (2.13), see section 2.8. A binary classifier usually has $\Omega = \{-1, 1\}$ which it is in this case. The function g takes a set of parameters $\omega = \{\omega_1 \dots \omega_K\}$ to speed up the classification of a new sample. To train the classifier, it needs training data which is a set of samples, X, paired with a label, Y. The labels has the same values as Ω.

$$X = \{x_1 \dots x_N\} \quad \text{where} \quad x_i = \{f_{i1} \dots f_{iM}\} \tag{2.13}$$

The choice of descriptive feature a crucial for the performance of the classifier. Avrim L. Bluma et al. [6] point out importance of finding relevant and strong features. It is very easy to access to huge amount of low-quality data on the Internet. It is not the number of features that decide the performance of a classifier but rather the relevance of feature and samples.

To illustrate how a learning method works, we jump right to a specific learning method called Support vector machine (SVM) [12], see section 2.7.2 . There exist several other learning methods such as decision tree [29], nearest neighbor [23], neural network [25] and many more. This master thesis will use SVM since it is simple to use and it performed best [36] compared to decision tree and nearest neighbor when classifying RPPVSM and non-RPPVSM.

2.7.2 Support vector machine

The principle behind SVM is to separate classes with a simple line (2D) or hyper plane (higher dimension). The line and plane can be described by its normal which has the parameters $\omega = \{\omega_1 \dots \omega_K\}$ and fulfill the equation of the plane eq. (2.14) where x is a point on the plane and b describes the distance from origin.

$$\omega^T \bullet x + b = 0 \tag{2.14}$$

The challenge now is to find the best ω which separate the classes with the largest margin. The first attempt is a linear SVM.

Linear SVM

Vapnik et al.[51] developed the linear SVM and it works like follows. Given a set of N samples, $X = \{x_1 \dots x_N\}$, one want to find the normal vector $\omega = \{\omega_1 \dots \omega_K\}$ of the hyperplane which separates the two classes with the largest margin. By setting up a set of equations eq. (2.15) where x_s is one of the samples closes to the hyperplane, so called support vector. z_p is a point on the hyperplane (not a sample), ϵ is the perpendicular distance between the hyperplane and x_s and b determines the offset of the hyperplane from the origin along w.

$$\begin{cases} \omega^T \bullet x_s + b = 1 \\ x_s = z_p + \epsilon\hat{\omega} \end{cases} \tag{2.15}$$

$$\begin{aligned} &\omega^T \bullet (z_p + \epsilon\hat{\omega}) + b = 1 \iff \\ &\epsilon\omega^T \bullet \hat{\omega} + \omega^T \bullet z_p + b = 1 \iff \\ &\backslash\omega^T \bullet \hat{\omega} = \|\omega\| \text{ and } \omega^T \bullet z_p + b = 0\backslash \\ &\epsilon\|\omega\| = 1 \end{aligned} \tag{2.16}$$

After some manipulation, see eq. (2.16), one notice that the best margin ϵ is achieved by minimizing $\|\omega\|$ which is the same as minimizing $\|\omega\|^2$. When finding the maximal ϵ, no samples may reside within the margin which can be expressed as eq. (2.17) where y_i is the class for each sample. This give the best hyperplane for the classifier where the classes are linearly separable fig. 2.2.

$$y_i(\omega^T \bullet x_i + b) \geq 1 \tag{2.17}$$

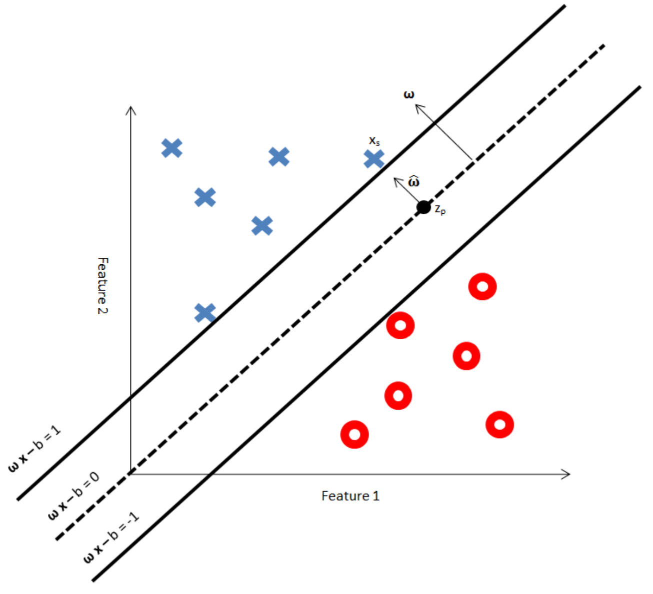

Figure 2.2: *Linearly separable classes.*

Soft margin SVM

All classification problems are not always linearly separable. In this case, eq. (2.17) does not hold true for all samples. This is solved by introducing a penalty ζ [12] for each sample on the wrong side of the hyperplane. This type of SVM is called a soft margin SVM. In this type, one try to solve eq. (2.18) where C is a parameters set before optimization.

$$\underset{\omega,b,\zeta}{\arg\min}\left(||\omega||^2 + C\sum_i \zeta_i\right) \tag{2.18}$$

under the condition eq. (2.19)

$$y_i(\omega^T \bullet x_i + b) \geq 1 - \zeta \quad \zeta \geq 0 \tag{2.19}$$

Large values on C results in a greater penalization for wrongly classified samples. This parameter makes a tradeoff between having a large margin and allowing samples to be on the wrong side of the hyperplane.

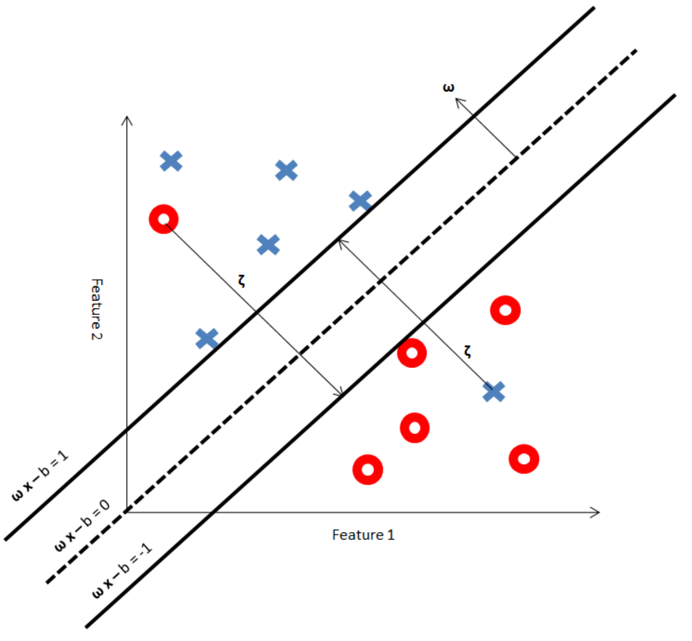

Figure 2.3: *Classes separated by a soft margin SVM.*

Non-linear SVM

Things are not always as simple as the case in figs. 2.2 and 2.3. The classes are often not linearly separable at all. Fig. 2.4 illustrate the so called XOR problem [54]. This classification problem requires a non-linear SVM. Boster et al. [7] presented a way to solve the XOR problem by mapping the samples onto a higher dimension. This is done by using kernels, $k(x_i, x_j)$, of different types [54]. The following are popularly used kernels:

- Polynomial: $k(x_i, x_j) = (x_i \bullet x_j + 1)^d$
- Radial Basis Function (RBF): $k(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2$
- Sigmoid: $k(x_i, x_j) = \kappa x_i \bullet x_j + c$

where γ, d, κ and c are parameters set by the user. This master thesis will only use the RBF kernel since it is easy to tune and the polynomial kernel has more hyperparameters which influence the complexity of the model [21]. The γ parameter defines how far the influence of a sample reaches, a small value meaning 'far' and vice versa. Read more about kernels in [52]

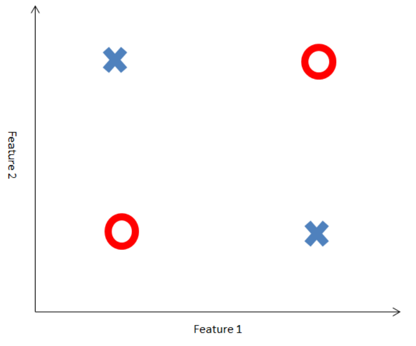

Figure 2.4: *XOR problem with two classes.*

2.7.3 Overfitting

A large number of parameters makes it possible to produce overly complicated boundaries if the training data is used for validation. This together with usage of training data as validation data can produce a problem in machine learning known as overfitting. In fig. 2.5 one can see that the red curve is a overfitted boundary while the green curve separates the two classes more generally. Overfitting occurs when the classifier tries to include outliers or wrongly labeled samples within the classifier boundary. To avoid this, one should use a subset of all the samples as testing data which will indicate if the classifier is overtrained.

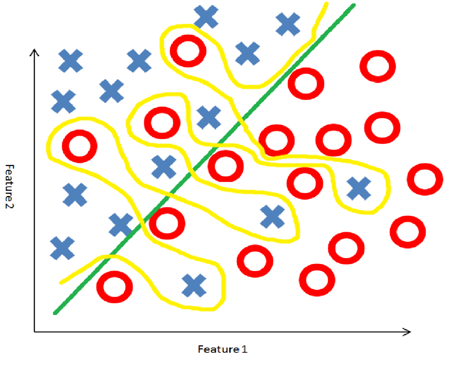

Figure 2.5: *Example of a overfitted boundary (yellow) and a more general boundary (green).*

2.8 Features descriptors

A feature descriptor extract information about patterns in an image, in this case a facial mark. This information can consist of colors in the image, edges for distinguishing light and dark areas, the texture of a surface and the direction of movement. The feature descriptors HOG, section 2.8.1, and LBP, section 2.8.2 are common descriptors when working with detection of objects [14, 17, 2] which is why these feature are used. Since facial marks mostly differ in color rather than shape [36], it would be wise to use features which are based on the color of the skin marks. RGB and HSV, section 2.8.3, are primitive color-mapping but has been used a feature descriptors before [36]. It would be even better to use more colors than just RGB and HSV color space. Color names, section 2.8.4, are linguistic color labels given to a single pixel [50]. By using even more colors to describe the facial marks it should result in better classification results.

2.8.1 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) was introduced by Dalal and Triggs [13] and it showed that it outperformed the current feature descriptors at that time.

The main idea of HOG is that a local object can be characterized by the edge directions of the object. The implementation of the descriptor is dividing each image into cells containing 4x4 pixels each. The orientation and magnitude of the gradient vectors is then calculated in each cell. The gradient was calculated using a simple 1-D $\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$ Sobel kernel, eq. (2.20), without any Gaussian filtering beforehand since it only reduced the performance of the descriptor. The gradient vectors are then sorted into nine different bins ranging from $0-180°$. This results in a histogram from each cell which is what is used as a descriptor. For better invariance to illumination, the descriptor vector should be normalized. This is done by grouping four cells into blocks. The cells in each block are concatenated, creating a vector, v, with the length 36. This vector is then normalized as in eq. (2.21). Here, ϵ is a small constant.

$$\nabla I = I * \begin{bmatrix}-1 & 0 & 1\end{bmatrix} \tag{2.20}$$

$$v_{norm} = \frac{v}{\sqrt{||v||^2 + \epsilon^2}} \tag{2.21}$$

Dalal and Triggs also showed that the performance of the descriptors increased even further if the block steps was made such that the blocks overlaps 50%. This overlapping can be observed in fig. 2.6. The window size which Navneet Dalal et al. used was 128x64 but since facial marks are more or less cylindrical, a window size of 48x48 was used in the master thesis.

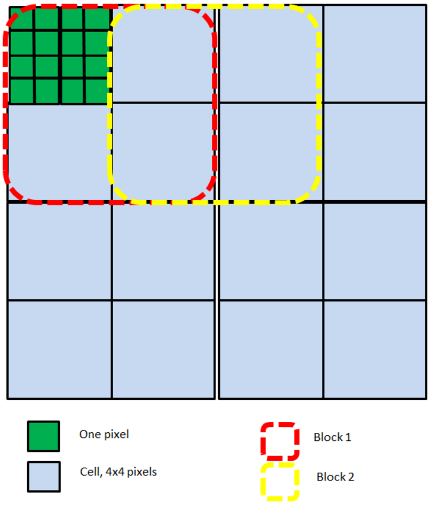

Figure 2.6: Schematic picture of the implementation of the HOG descriptor.

2.8.2 Local Binary Patterns

Local Binary Patterns (LBP) was developed by Timo Ojala et al. [37] by improving the work of Li Wang et al. [18]. Li Wang et al. introduced a texture analysis method so-called texture unit. From a 3x3 pixel area, the pixels surrounding the central pixel was given either the value 0, 1 or 2. Each 3x3 pixel area would thus be given 1 out of 6561 possible texture unit. The distribution of texture units over an image was called a texture spectrum.

Timo Ojala et al. reduced the number of possible texture unit by making a binary version of the texture unit. Each surrounding pixel, p_s can would instead receive 0 or 1 depending on the value of the central pixel, p_c. The value is decided by eq. (2.22). By using a binary version of the texture unit, the number of possible texture unit is instead 256.

$$f(p_s) = \begin{cases} 1 & \text{if } p_s \geq p_c \\ 0 & \text{else} \end{cases} \tag{2.22}$$

When each surrounding pixel has been given a value, fig. 2.7 (b), the 3x3 pixel area has a binary code, e.g. 00100010, which corresponds to 34 as decimal, eq. (2.23).

$$LBP(p_c) = \sum_{k=0}^{7} f(p)2^k \tag{2.23}$$

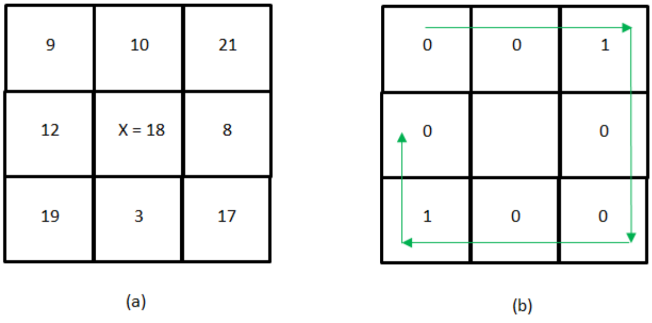

Figure 2.7: Schematic picture of the implementation of the LBP descriptor. (a) Original 3x3 pixel area with x as central pixel (b) 3x3 pixel area with binary values for the surrounding pixels.

The LBP for each pixel is binned in the corresponding decimal value. This results in a 256 long vector. This vector is used a descriptive feature for the classifier.

2.8.3 RGB and HSV

RBG is the intuitive feature of choice if one want information about the color of the object. It is possible to extract much information from the color channels but this master thesis will only use the mean, $\bar{p}$ eq. (2.24), and the standard deviation, p_σ eq. (2.25), from each color channel. These values are put into a vector and are used to train the classifier.

$$\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{2.24}$$

$$p_\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i - \bar{p})^2} \tag{2.25}$$

Similarly, the mean and standard deviation are extracted from the Hue, Saturation, and Value (HSV) color space [10]. HSV color space is a common cylindrical-coordinate representation of the RGB color space. It was developed to be more intuitive color representation than the RGB color space.

2.8.4 Color names

We use color names to describe our surrounding every day without thinking about it. It becomes however a challenge for computers to detect certain object with a specific color attribute, e.g. a red car. In computer vision, color names are used in search engines to retrieve demanded object with a certain color. To use color names in computer vision, the RGB color space has to be mapped to different colors. This has mainly been done by letting test subject label color chips [16]. The colors are to be chosen from a set of colors, usually black, blue, brown, gray, green, orange, pink, purple, red, white and yellow. These colors are the basic colors of the English language. The color mapping is derived from the labeled color chips.

The problem with the color chip method is that the color chips are under ideal lighting on a color neutral background. This is not the case with real-world images which is why Joost van de Weijer et al. [50] have investigate the use of color names in images from real-world applications. They used a large data set of labeled real-world images and used probabilistic latent semantic analysis (PLSA) [20] to model the data. This model tries to find the "meaning" of the words i a document. The model has also been used in computer vision where images take the role of documents and pixels the role of word [4]. The "meaning" of the pixels are in this case the color. Joost van de Weijer et al. showed that color names learned from real-world image outperform color chips which is why this trained color mapping is used in this master thesis.

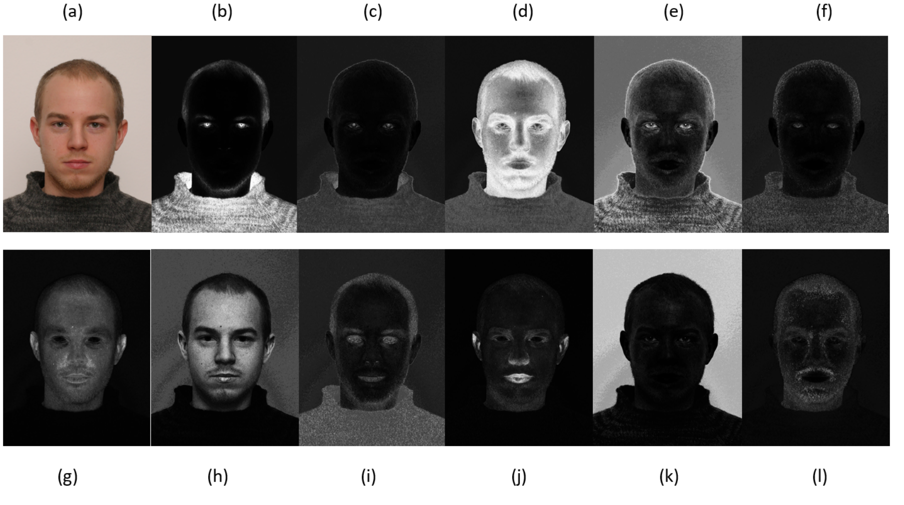

Figure 2.8: *Different color channels: (a) RGB, (b) black, (c) blue, (d) brown, (e) gray, (f) green, (g) orange, (h) pink, (i) purple, (j) red, (k) white, (l) yellow*

Like the RGB and HSV color space, the mean and standard deviation is extracted from each of the 11 color names: black, blue, brown, gray, green, orange, pink, purple, red, white and yellow.

3 Method

This chapter will describe the pipeline of the algorithm developed during this master thesis. The different parts of the algorithm are viewed with a more implementation focused view.

3.1 Overview

An overview of the algorithm is presented in figure 3.1. The algorithm start of with pre-processing an input image. This step makes sure that the image is normalized and that all necessary sub-parts are generated. Then a facial mask is generated in the segmentation step. Now the algorithm have everything in order to detect the skin mark candidates. All the candidates are then post-processed to eliminate false detections. Finally, the remaining skin marks are classified as a permanent or non-permanent skin mark.

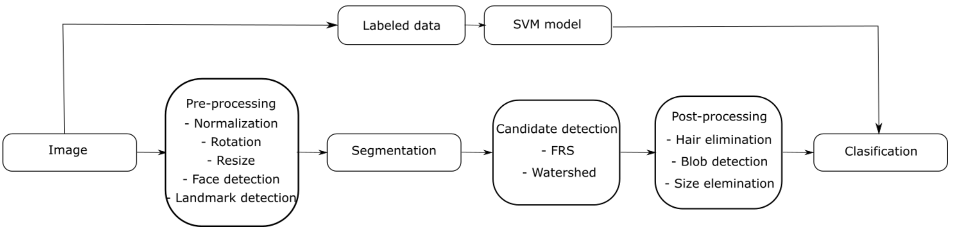

Figure 3.1: *Overview of the algorithm*

3.2 Data and annotation

To evaluate the algorithm, a set of 106 images of faces en face were acquired from SCface database [15] and FRGC database [39]. These images where collected at University of Zagreb and University of Notre Dame respectively. The purpose of the databases is to provide data to develop and improve face recognition algorithms. The images where taken in under controlled conditions indoor in a studio setting indoor with a high-quality photo camera. Figure 3.2 shows an example of the images from the databases.

Figure 3.2: *One example of the images used to evaluate the algorithm*

Each image was examined the supervisors at NFC who labeled facial skin mark of interest. Each mark was given either the label permanent or non-permanent according to NFC definitions. This resulted in 506 marks where 353 were labeled as permanent and 153 non-permanent.

3.3 Pre-processing

When the algorithm is given a RGB image, denoted I, it first detects the location of the face with the help of the face detector in OpenCV. It surrounds the face with a bounding box. With the bounding box, the facial landmarks can be detected by using the algorithm from Dlib. This landmark algorithm was chosen since it converges faster than other state of the art methods [22].

Figure 3.3: *Image with the 64 landmarks shown as blue dots*

With the landmarks, it is possible to begin the normalization process, see section 2.2. First, the image is photometric normalized using LRSR algorithm. This tone mapping operator is fast and had a good implementation available in C++. It performs on pair with the best tone mapping operator of today [3]. Photometric normalization is vital since the visibility of facial marks can be affected by varying illumination of the image.

Second, the image is rescaled such that the interpupillary distance is 500 pixels. This resizes the images to approximately 2100x2800 and the interpolation method used is cubic interpolation. The landmarks are also used to rotate the image so that the eyes are level. The rotation and resizing of the image is called geometric normalization and is necessary remove the effect of the distance and tilt of the camera. In fig. 3.4 one can see the result from the image normalization.

Figure 3.4: *Image after photometric and geometric normalization*

The last part of the pre-processing step is to segment out areas, see section 2.4, which can cause false detections such as facial hair, nostrils, pupils etc. This is done by generating a binary mask. To segment out areas with skin, the implementation of GrabCut in OpenCV was used since it has proven to perform as well or better than many other user interactive foreground extraction methods [41]. From the skin mask, the eyes, nostrils, mouth and throat are cut out using elliptical shapes around the landmarks marking these regions. To expand these holes, a morphological erosion algorithm was performed on the image with a 3x3-kernel containing ones. The resulting mask can be observed in fig. 3.5.

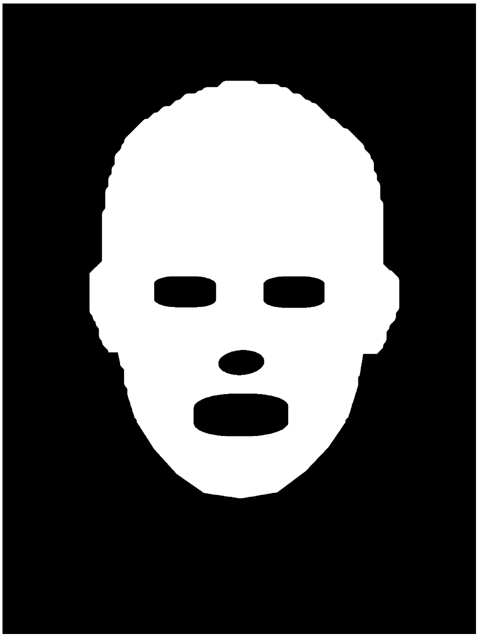

Figure 3.5: *Image of the facial mask*

3.4 Candidate detection

The pre-processed image, denoted I_{pre}, can now be used to search for facial skin mark candidates. This is done with the help of FRS algorithm, see section 2.5. It highlights circular shapes which can be more easily detected. The algorithm makes calculations with different radii, N, and the ones used are $N = \{1, 3, 5, 7, 9, 11, 13, 15\}$. These radii were used since 75% of facial marks had an area smaller than 600 pixels, see fig. 2.1. The Gaussian kernel A_n size increased from 3x3 to 7x7 depending on the radius r.

Below, fig. 3.6, the resulting FRS image is presented. It is hard to see the facial marks since the image contains positive and negative values. By taking the absolute value of the image, the marks appear more prominent, see fig. 3.7.

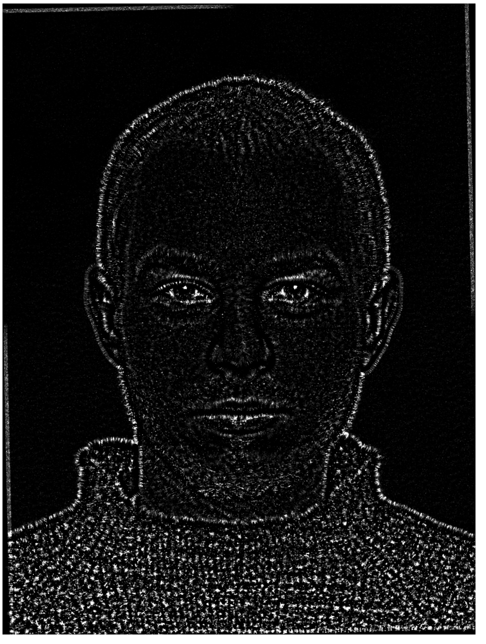

Figure 3.6: *FRS image*

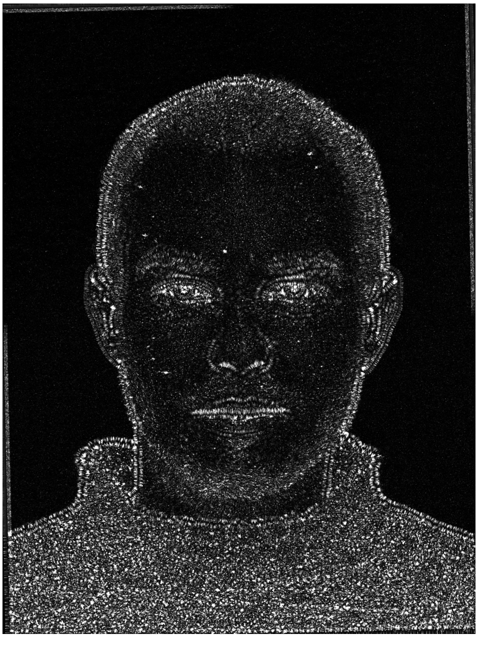

Figure 3.7: *Absolute value of FRS image*

At this point, an FRS-image with points of interest has been acquired. From this image, a binary threshold was applied with the threshold h_{FRS}, see eq. (3.1).

$$I(p) = \begin{cases} 1 & \text{if } I(p) \geq h_{FRS} \max(I) \\ 0 & \text{else} \end{cases} \tag{3.1}$$

This results in a binary image which is used in the watershed algorithm described by Fernand Meyer [32]. The use of watershed is good since it can find the contour of uneven marks as long as the pixels approximately have the same intensity value. The watershed algorithm is applied on a gray image of the face. The output from this is a set of bonding boxes containing facial marks candidates.

The h_{FRS} is the only parameter which is varied in the candidate detector. It is varied to examine the performance of the detector by looking at the recall and precision value of the detector.

3.5 Post-processing

After candidate detection, the false detection was eliminated by using three methods. The first method finds candidates which contains a blob. The blob detector in OpenCV was used and it was given the three parameters: inertiaRatio, convexity and circularity. The method also allows to sort out all candidates which contains more than one blob. These candidates was eliminated.

The second part removes candidates which contain to many hair pixels. The h_{hair} threshold, see section 2.6.2, was set to 0.02 and candidates containing more than 10% hair pixels was excluded.

The parameters from the first and second eliminator was chosen such that the number of false detection was reduced while preserving true detections. This was done by examining a few images with different parameter settings.

The third and last method simply removed all candidates which had a larger area than 1000 pixels. This value was chosen since no annotated marks had a area larger than that, see fig. 2.1.

3.6 Classification

When a set of facial marks has been acquired through the skin mark detector, they have to be separated into permanent and non-permanent marks. This is done with a non-linear SVM with a RBF-kernel, see section 2.7. It was trained with different sets of feature descriptors, table 3.1. Each set was trained with one part training data and evaluated with one part of test data.

The parameters C and γ was optimized by first training the classifier with a crude range of values. The C-value and γ that gave the best accuracy from the crude grid of values was located. Next a finer grid search was performed in the region of the best pair of C and γ. From this finer grid, the best parameters for the specific set of features could be picked out. Each grid of parameters contained 20x20 pair of parameters. The low number of pairs was chosen to reduce the computation time when searching for the best parameter pair.

When in comes to the feature descriptors, see section 2.8, the LBP-features has no variable parameters which is used in this master thesis. The HOG-features needs a couple of parameters. The window size was set to 48x48 pixels, block size was 8x8, block stride was 4x4, cell size 4x4 and 9 bins.

The different set of features used to train the classifier can be seen in table 3.1. Here, COLOR means the 11 color names described in section 2.8.4.

Table 3.1: *Sets of feature descriptors to be evaluated*

Set	Features
1	RGB
2	HSV
3	COLOR
4	HOG
5	LBP
6	HOG + RGB
7	HOG + HSV
8	HOG + COLOR
9	LBP + RGB
10	LBP + HSV
11	LBP + COLOR
12	RGB + HSV + COLOR

3.7 Implementation details

The algorithm was implemented in Visual Studio 2013 using the OpenCV 3.0.0 [9] for most image processing. The landmark detection algorithm comes from an open source library Dlib 18.18 [24]. When it comes to the graphs and bar-diagrams in this master thesis, Matlab [31] was used since it is easy to produce good looking graphs.

4 Experiments

This Chapter first describes the experiment to evaluate the algorithm and then presents the results.

4.1 Evaluation measures

To be able to compare the results from the different feature descriptors with each other, it is crucial to have some kind of evaluation measurement. The most common measurements for binary classifiers are based on the confusion matrix [44].

The confusion matrix displays the result from a classifier consist of four values:

- True positive (TB)
  The samples which were labeled as class one has also been predicted to belong to class one.
- True negative (TN)
  The samples which were labeled as class two has also been predicted to belong to class two.
- False positive (FP)
  Samples that were incorrectly assigned to class one.
- False negative (FN)
  Samples that were incorrectly assigned to class two.

	Predicted true	Predicted false
True positivve	True positive (TP)	False negative (FN)
True negative	False negative (FP)	True negative (TN)

Figure 4.1: *Confusion matrix*

From the confusion matrix, a collections of performance measurement can be calculated. This master thesis will use three values: accuracy eq. (4.1), precision eq. (4.2) and recall eq. (4.3). Accuracy show the overall effectiveness of the classifier. Precision show class agreement of the data labels with the positive labels given by the classifier. Recall show the effectiveness of a classifier to identify positive labels

$$\text{Accuracy [\%]} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.1}$$

$$\text{Precision [\%]} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.2}$$

$$\text{Recall [\%]} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.3}$$

4.2 Experiment setup

The experiment was set such that the image set was processed by the algorithm with 11 different thresholds values, h_{FRS}, for the FRS-image. The h_{FRS} ranged from 0.05 to 0.15. The output was compared to the ground truth. A correct detection was defined as all detections which overlaps with an annotated mark. This definition has been chosen since some of the detections can be very small. Also, since candidates with a area larger than 1000 pixels has been eliminated, no overly large candidates can give correct detections.

The h_{FRS}-value which gives the best recall value was used to evaluate the elimination process of the candidates. This was done by calculating the precision and recall values before the different elimination steps. The results is displayed in fig. 4.3.

To evaluate how the elimination process is working, the recall and precision values after each elimination step. These results can be observed in fig. 4.5.

To evaluate the facial mark classifier, a cross validation of the 506 annotated mark were performed. 100 marks was chosen at random to be used as test marks while the remaining marks was used for training the SVM, see fig. 4.2. This was repeated until all the marks had been used as test marks.

In order to find the best set of descriptive features from table 3.1, the classifier was trained for each set of features. The parameters C and γ was optimized for each set.

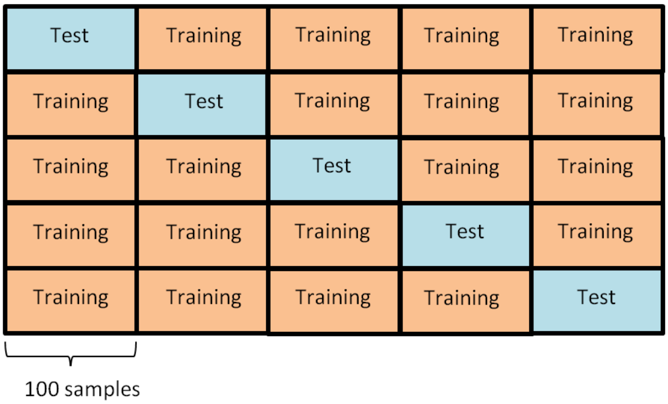

Figure 4.2: *Cross validation*

4.3 Results

This section will present the results from experiment described above. The result is divided into two parts: Detector and Classifier.

4.3.1 Detector

Here the results from the facial mark detector is presented. In the fig. 4.3, the precision and recall for different h_{FRS}-value can be examined. The precision corresponds to the white bar and the recall corresponds to the black bar. Note that this is only the detections of facial mark and no classification between permanent and non-permanent marks.

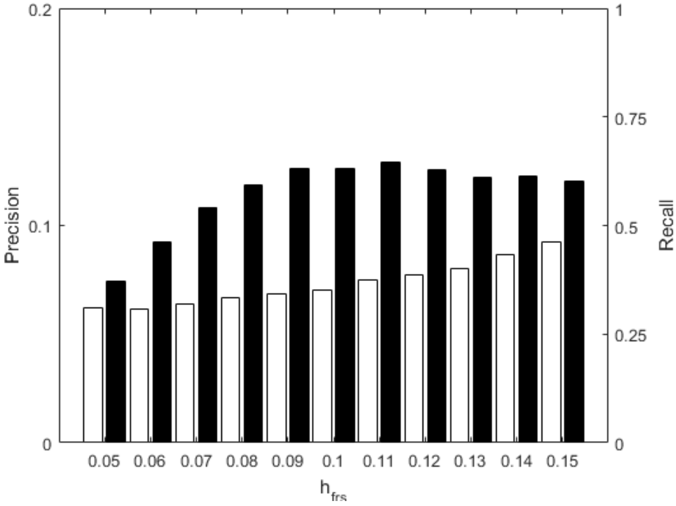

Figure 4.3: *Detection results from the algorithm with different h_{frs}-values. The white bars represents the precision value and the black bars represents recall value.*

As one can see, the precision increases with higher h_{FRS}-value without affecting the recall substantially. This is means that the number of candidates decreases with a growing h_{FRS}-value. Thus, a small h_{FRS}-value results in a large number of candidates while a larger value gives fewer candidates.

Note that the number of candidates found by the detector decreases with a larger h_{FRS}-value, see fig. 4.4. The size of the candidates also decreases which could explain the increase of recall in fig. 4.3. The algorithm finds different kind of candidates for small h_{FRS}-values.

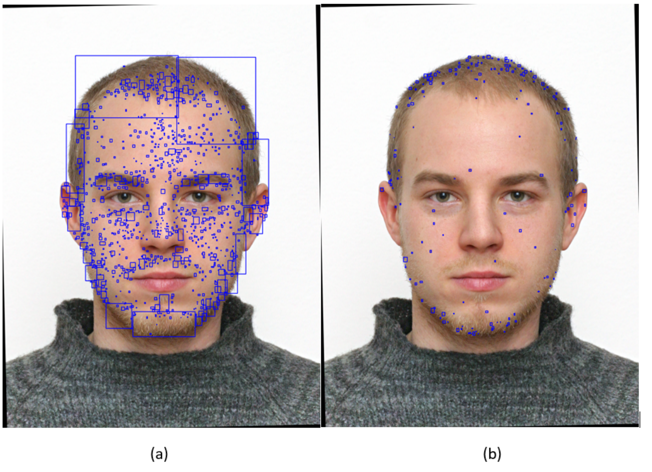

Figure 4.4: *Candidate detection before post-processing, (a) with small h_{hair}-value, (b) with large h_{hair}-value,*

In fig. 4.5, it is possible to see the effects of the different elimination steps. As before, the white bar represent the precision and the black bar represents the recall. The first pair is the result just after the candidate detection and the second pair is the result after the blob detector. Furthermore, the third pair is after the hair eliminator and the last pair is after the size eliminator.

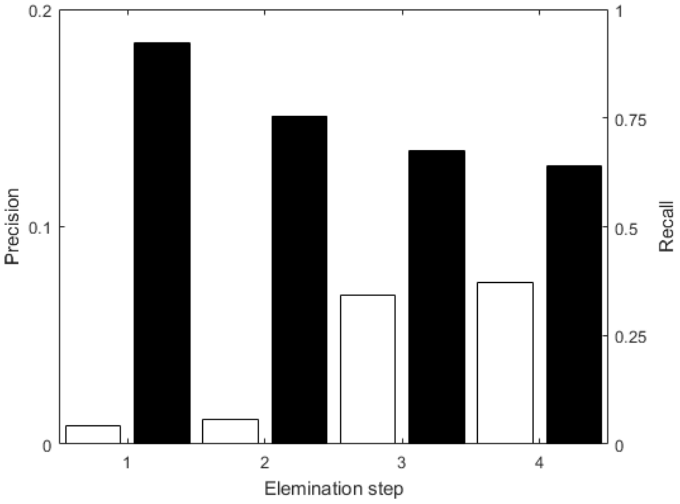

Figure 4.5: *Detection results from the algorithm after different candidate elimination steps. 1 = before elimination, 2 = after blob-elimination, 3 = after hair-elimination, 4 = after size-elimination. The white bars represents the precision value and the black bars represents recall value.*

It is obvious that the different eliminators are essential for the algorithm. The hair eliminator improves the precision the most while the blob detector worsens the recall the most.

In fig. 4.6 below one can observe all the candidates found by the detector before post-processing. The candidates are shown as blue bounding boxes and the annotated facial marks for this image is show as red bounding boxes. There are many false detections in the hair and the beard which is some of the problems identified in this master thesis.

After the post-processing, fig. 4.7, almost all the false detection has been eliminated but some of them remain. The remaining false detection are located in the beard and hair. Some bounding boxes are around skin marks which has not been deemed of interest by the forensics at NFC.

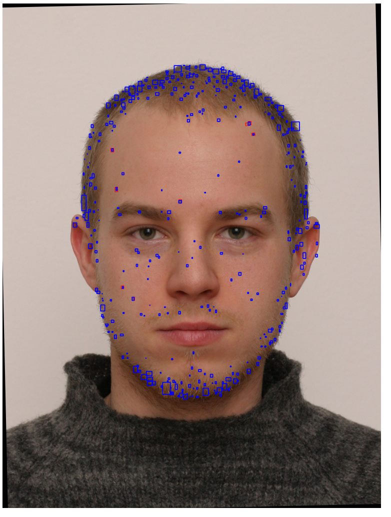

Figure 4.6: *An image of all potential facial marks. Each potential mark is shown as a blue box and all annotated facial mark is shown as a red box.*

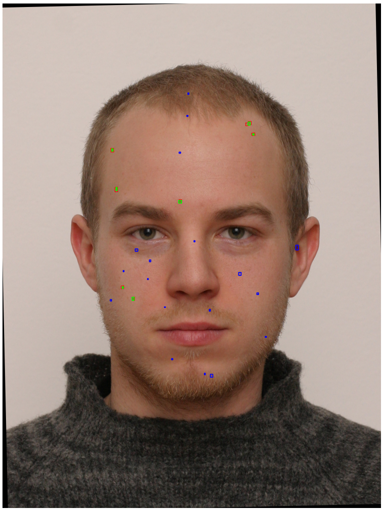

Figure 4.7: *An image of the final result of the detector. Green boxes denotes true detections, red boxes denotes annotated marks and blue boxes denotes false detections.*

4.3.2 Classifier

Here the results from the different feature sets for the classifier is presented. When looking at the different features separately, table 4.1, one can see that the HOG and LBP feature perform bad compared to the color based features. The RGB and the color names features has approximately the same accuracy which means they perform the best.

Table 4.1: *Confusion matrix for single features*

Feature set	RGB	HSV	COLOR	HOG	LBP
TP	336	334	335	348	313
TN	107	104	107	32	88
FP	17	19	18	5	40
FN	46	49	46	121	65
Accuracy	87,55	86,56	87,35	79,25	75,10

Now, if the color based features are added to the HOG and LBP features respectably, tables 4.2 and 4.3 it is clear that the color based features improve the accuracy for the structure based features. However, it is clear that the color names gives the best result to the classifier. It seams that the color names features is not effected as much by HOG and LBP features as the RGB features are.

Table 4.2: *Confusion matrix for HOG and color based features*

Feature set	HOG + RGB	HOG + HSV	HOG + COLOR
TP	326	341	332
TN	63	46	111
FP	27	12	21
FN	90	107	42
Accuracy	76,88	76,48	87,55

Table 4.3: *Confusion matrix for LBP and color based features*

Feature set	LBP + RGB	LBP + HSV	LBP + COLOR
TP	313	313	333
TN	89	88	106
FP	40	40	20
FN	64	65	47
Accuracy	79,45	79,25	86,76

If one combines the color based features table 4.4 the accuracy does not improve but stays at the same level as only using the color names.

Table 4.4: *Confusion matrix for color based features combined*

Feature set	RGB + HSV + COLOR
TP	336
TN	106
FP	17
FN	47
Accuracy	87,35

See appendix for an analysis of the SVM parameters C and γ.

5 Discussion

This section will discuss the results from the algorithm and the methods used to implement it. This section will also suggest future work and mention ethical perspective.

5.1 Result

As seen, the detector has problem with false detections which is a huge problem. The precision is not very good, not even over 10%, due to the many false detections. The precision does not increase faster than the decline of the recall with an increasing h_{frs}-value. This indicates that there are margins for improvement when it comes to the candidate detector.

Vorder Bruegge et al. [33] got a precision of 71% which is a lot better than the result from this master thesis. Taeg Sang Cho et al.[11] got a recall value of 84.7% which is also better than the results from this master thesis. One thing that should be noticed is that Vorder Bruegge et al. has been focusing to finding RPPVSM which is a wide definition of skin marks. This master thesis has tried to find the skin marks which has been of interest to the forensics at NFC. This could be one of the reasons to the large amount of false detections. The detector is probably detecting RPPVSM which has not been deemed of interest.

When looking at the elimination method used, they do improve the precision and recall values but maybe not to the extant which one was hoping for. There can be a great improvement in trying to find a optimal value for the h_{hair} threshold since this has not be investigated. Maybe there are better methods to bee used to eliminate the false detections than the ones used in this master thesis.

The classifier offers some indication to the importance of colors when it comes to separating permanent and non-permanent marks. The structural features alone perform all right but they do not improve the accuracy when they are combined with color based features. Simply put, they do not show complementary properties.

The reason why the color based features perform better is probably due to the small differences in structure between permanent and non-permanent skin marks, see fig. 1.1. One can see a slight difference in color, the permanent marks tend to be browner while the non-permanent marks tend to be more red.

It is hard to compare the performance of the classifier with other research since researchers using classifiers for face recognition mostly use the classifier to find matching set of skin marks. For example, a classifier is used to determine if a set of skin marks matches other set of skin marks. Also, Taeg Sang Cho et al. used classifiers to find moles but they only tried to separate moles and everything else. This should be easier than separating permanent and non-permanent skin marks. In other words, this master thesis has looked at a more difficult problem which is why it is hard to compare the results from this master thesis with other research.

5.2 Method

There are a lot to say about the methods used in the algorithm. The major problem with the algorithm is the elimination of candidates. They eliminate candidates which are true facial marks. The hair elimination part does improve the precision the best. The blob detector on the other hand is hardly improving the precision at the cost of recall loss. This means that the blob-detector is not contributing to the algorithm in a positive way.

Tim Lee et al. describes their algorithm well except when they are explaining how to calculate the hair mask for each color channel. It is not clear what the maximum from refers to. The algorithm in this work used the maximal pixel value between the different structuring elements.

The mark detector used in the algorithm was good at indicating the potential facial marks but the simple thresholding method to pin point them out was not optimal. It kept the pixels larger than a certain percent of the maximal value in the FRS image. This resulted in many unnecessary candidates which of course contributed to the high false detection rate.

Regarding the references used in these thesis, several of them uses FRS to detect point of interest which shows the actuality of the method. Many of the papers trying to detect facial marks uses a crude segmentation mask which does not follow the hairline and chin well. This algorithm uses a more precise segmentation method which reduces the areas which are processed.

5.3 Future work

A proposed remedy for the low precision on the detector is to improve the elimination of the false candidates. The hair-detector is to crude and may be combined with or replaced by a module which looks at the Fourier Transform of the candidates.

The scope of this master thesis did not allow to compare different skin mark detection methods. There are not many reports which compare a LoG-based with a FRS-based detector which would be interesting. Many face recognition algorithm uses multi scale methods to determine if a skin mark is permanent and not a classifier. Why is this is would be good research field.

It would be interesting to examine how different types of classifier would affect the performance of the classifier. Maybe would a random forest classifier perform better or even a nearest neighbor classifier. Also, one would like to make a multi class classifier where one class would be non-facial mark. Maybe would this be a solution to the low precision result from the detector.

5.4 Ethical perspective

As with all applications which can be used for surveillance of people, the integrity is at stake. Facial recognition algorithms using facial marks can be misused for malicious intent. They can also help the legal system to catch and convict criminals which is desirable outcome of this paper.

When it comes to the facial images, they are taken from an open source database which should only be used for academical research. There is no personal information attached to the images which makes them as anonymous as possible without corrupting the images.

6 Conclusion

The need to develop reliable face recognition algorithms has increased dramatically the last decades thanks to ever increasing developments in computer vision. Therefore, this master thesis has examined the possibilities to develop an algorithm which could detect facial marks and separate them into permanent and non-permanent marks. The separation was done with a classifier and the focus has been to evaluate which features are suitable for the classifier.

The algorithm consists of a preprocessing step where facial images have been normalized booth geometrically and photometrically. Then the skin mark candidates are found with the help of the radial symmetry in the image. Finally the false candidates are eliminated depending on their size, shape and hair content. The classification of the detected facial marks is done with a SVM classifier with a set of descriptive features.

To evaluate the algorithm, the precision and recall was calculated for different parameters in radial symmetry method. These performance values were also used to see the performance of the different elimination methods. For the classifier, the accuracy from the confusion matrix was used to evaluate the performance of different combinations of features. The feature examined was RGB, HSV, HOG, LBP and color names.

The results show that there exists a big problem with false detections of facial marks since the precision of the algorithm is very low. One can also say that the hair elimination process is increasing the precision the best. The classifier performs well and the result suggests that color based features are more important than structural features. No complementary properties could be found between the color based features and structural features.

There are many areas within this master theses which would be interesting to investigate further. One should first of all improve the precision of the detector by examining the elimination methods further or even try new methods. When it comes to the classifier, it would be interesting to look if the normalization of the feature affects the accuracy and how. The future work within facial recognition is endless and the use of automatic face recognition program will undoubtedly increase.

Appendix

Parameter optimization

This section shows the different accuracy for different pair of parameters set in the classifier. Each image shows the accuracy for on set of features evaluated in this master thesis.

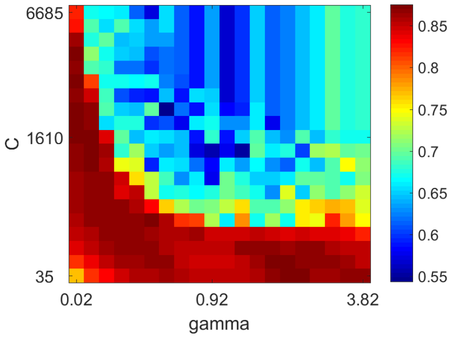

Figure A1: *Accuracy for RGB feature*

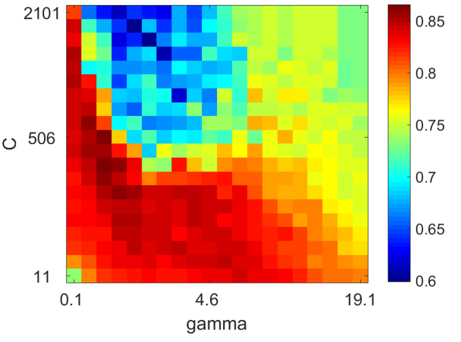

Figure A2: *Accuracy for HSV feature*

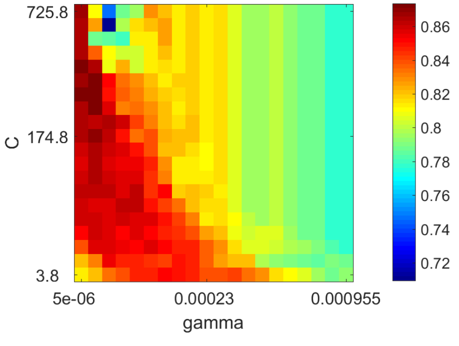

Figure A3: *Accuracy for COLOR feature*

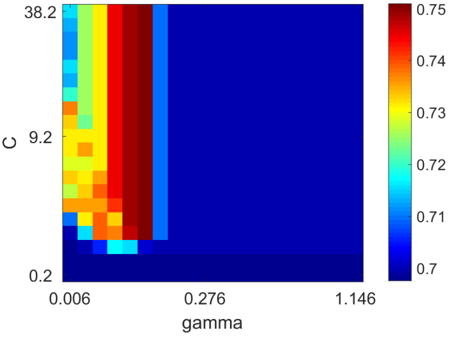

Figure A4: *Accuracy for HOG feature*

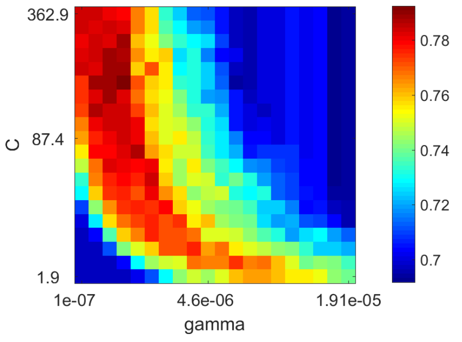

Figure A5: *Accuracy for LBP feature*

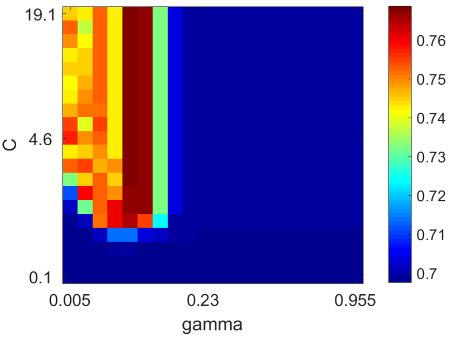

Figure A6: *Accuracy for HOG+RGB feature*

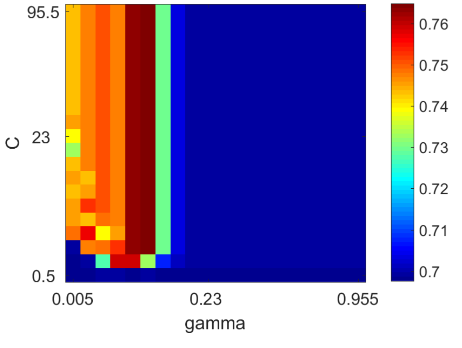

Figure A7: *Accuracy for HOG+HSV features*

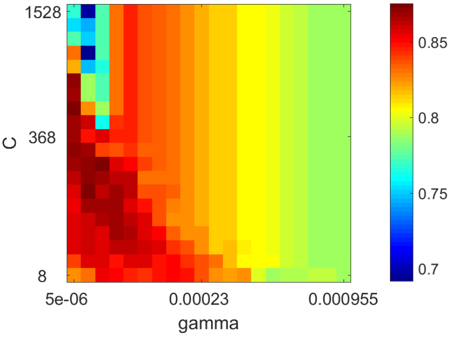

Figure A8: *Accuracy for HOG+COLOR features*

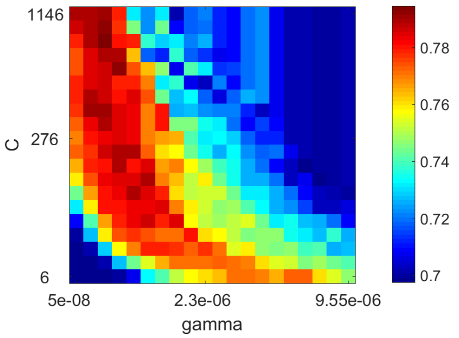

Figure A9: *Accuracy for LBP+RGB features*

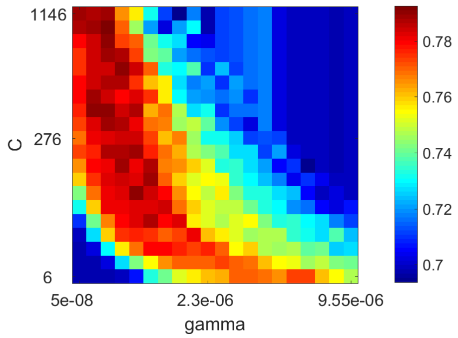

Figure A10: *Accuracy for LBP+HSV features*

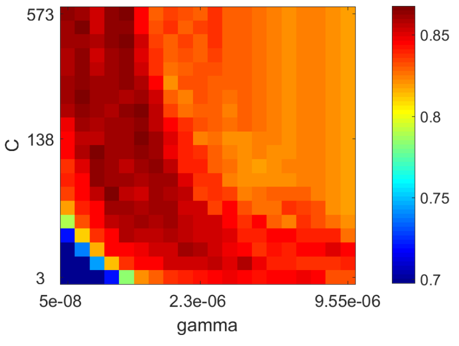

Figure A11: *Accuracy for LBP+RGB features*

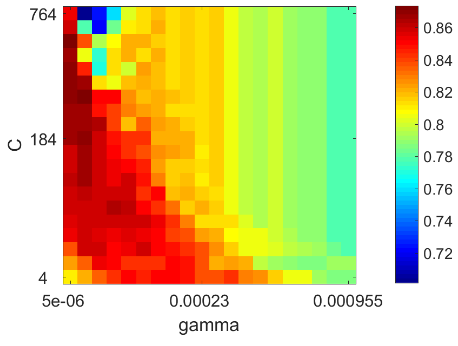

Figure A12: *Accuracy for HSV+RGB+color features*

Bibliography

[1] Bertillon A. Forensic facial analysis. identification anthropometrique: instructions signaletiques. France, Paris, 1885. Cited on page 1.

[2] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006. Cited on pages 3 and 18.

[3] Nikola Banić and Sven Lončarić. Color badger: a novel retinex-based local tone mapping operator. In *International Conference on Image and Signal Processing*, pages 400–408. Springer, 2014. Cited on pages 8 and 25.

[4] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M Blei, and Michael I Jordan. Matching words and pictures. *Journal of machine learning research*, 3(Feb):1107–1135, 2003. Cited on page 21.

[5] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006. Cited on page 13.

[6] Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1):245–271, 1997. Cited on page 13.

[7] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. Cited on page 16.

[8] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001. Cited on page 9.

[9] Gary Bradski et al. The opencv library. *Doctor Dobbs Journal*, 25(11):120–126, 2000. Cited on pages 8 and 30.

[10] Matthew P Carter. Computer graphics: principles and practice, 1997. Cited on page 20.

[11] Taeg Sang Cho, William T Freeman, and Hensin Tsao. A reliable skin mole localization scheme. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. Cited on pages 3, 4, and 41.

[12] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. Cited on pages 13 and 15.

[13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. Cited on page 18.

[14] Markus Enzweiler and Dariu M Gavrila. A multilevel mixture-of-experts framework for pedestrian classification. *IEEE Transactions on Image Processing*, 20(10):2967–2979, 2011. Cited on pages 3 and 18.

[15] Mislav Grgic, Kresimir Delac, and Sonja Grgic. Scface–surveillance cameras face database. *Multimedia tools and applications*, 51(3):863–879, 2011. Cited on page 24.

[16] Lewis D Griffin. Optimality of the basic colour categories for classification. *Journal of the Royal Society Interface*, 3(6):71–85, 2006. Cited on page 21.

[17] Feng Han, Ying Shan, Ryan Cekander, Harpreet S Sawhney, and Rakesh Kumar. A two-stage approach to people and vehicle detection with hog-based svm. In *Performance Metrics for Intelligent Systems 2006 Workshop*, pages 133–140. Citeseer, 2006. Cited on pages 3 and 18.

[18] Dong-Chen He and Li Wang. Texture features based on texture spectrum. *Pattern Recognition*, 24(5):391–399, 1991. Cited on page 19.

[19] Brian Heflin, Walter Scheirer, and Terrance E Boult. Detecting and classifying scars, marks, and tattoos found in the wild. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 31–38. IEEE, 2012. Cited on page 3.

[20] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999. Cited on page 21.

[21] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. Tech. rep. Department of Computer Science, National Taiwan University, 2003. Cited on page 16.

[22] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014. Cited on pages 7 and 24.

[23] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985. Cited on page 13.

[24] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. Cited on page 30.

[25] Teuvo Kohonen. An introduction to neural computing. *Neural networks*, 1 (1):3–16, 1988. Cited on page 13.

[26] Tim Lee, Vincent Ng, Richard Gallagher, Andrew Coldman, and David McLean. Dullrazor®: A software approach to hair removal from images. *Computers in biology and medicine*, 27(6):533–543, 1997. Cited on page 11.

[27] Tim K Lee, M Stella Atkins, Michael A King, Savio Lau, and David I McLean. Counting moles automatically from back images. *IEEE Transactions on Biomedical Engineering*, 52(11):1966–1969, 2005. Cited on pages 3 and 4.

[28] Hongbo Li, David Hestenes, and Alyn Rockwood. Generalized homogeneous coordinates for computational geometry. In *Geometric Computing with Clifford Algebras*, pages 27–59. Springer, 2001. Cited on page 8.

[29] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011. Cited on page 13.

[30] Gareth Loy and Alexander Zelinsky. Fast radial symmetry for detecting points of interest. *IEEE Transactions on pattern analysis and machine intelligence*, 25(8):959–973, 2003. Cited on page 10.

[31] MATLAB. *version 9.0.0 (R2016a)*. The MathWorks Inc., Natick, Massachusetts, 2016. Cited on page 30.

[32] Fernand Meyer. Color image segmentation. In *Image Processing and its Applications, 1992., International Conference on*, pages 303–306. IET, 1992. Cited on page 29.

[33] Richard W. Vorder Bruegge Ph.D. Nisha Srinivas M.Sc., Patrick J. Flynn Ph.D. Human identification using automatic and semi-automatically detected facial marks. *Journal of Forensic Sciences*, 61(S1):117–130, September 2015. Cited on pages 2, 3, 4, 10, 11, and 41.

[34] Anders Nordgaard, Ricky Ansell, Weine Drotz, and Lars Jaeger. Scale of conclusions for the value of evidence. *Law, probability and risk*, pages 1–24, 2011. Cited on page 1.

[35] Arfika Nurhudatiana, Adams Wai-Kin Kong, Keyan Matinpour, Siu-Yeung Cho, and Noah Craft. Fundamental statistics of relatively permanent pigmented or vascular skin marks for criminal and victim identification. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–6. IEEE, 2011. Cited on page 2.

[36] Arfika Nurhudatiana, Adams Wai-Kin Kong, Lisa Altieri, and Noah Craft. Automated identification of relatively permanent pigmented or vascular skin marks (rppvsm). In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2984–2988. IEEE, 2013. Cited on pages 3, 4, 14, and 18.

[37] Timo Ojala, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582–585. IEEE, 1994. Cited on page 19.

[38] Unsang Park and Anil K Jain. Face matching and retrieval using soft biometrics. *IEEE Transactions on Information Forensics and Security*, 5(3):406–415, 2010. Cited on page 3.

[39] P Jonathon Phillips, Patrick J Flynn, Todd Scruggs, Kevin W Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. Overview of the face recognition grand challenge. In *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, volume 1, pages 947–954. IEEE, 2005. Cited on page 24.

[40] Edoardo Provenzi, Massimo Fierro, Alessandro Rizzi, Luca De Carli, Davide Gadia, and Daniele Marini. Random spray retinex: A new retinex implementation to investigate the local properties of the model. *IEEE Transactions on Image Processing*, 16(1):162–171, 2007. Cited on page 8.

[41] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004. Cited on pages 9 and 26.

[42] Jan Schier and Bohumil Kovár. Automated counting of yeast colonies using the fast radial transform algorithm. In *Bioinformatics*, pages 22–27, 2011. Cited on page 10.

[43] Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 746–751. IEEE, 2000. Cited on page 8.

[44] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009. Cited on page 31.

[45] Nicole A Spaun and Richard W Vorder Bruegge. Forensic identification of people from images and video. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–4. IEEE, 2008. Cited on page 2.

[46] Nisha Srinivas, Gaurav Aggarwal, Patrick J Flynn, and Richard W Vorder Bruegge. Analysis of facial marks to distinguish between identical twins. *Information Forensics and Security, IEEE Transactions on*, 7(5):1536–1550, 2012. Cited on pages 3, 4, and 10.

[47] K-K Sung and Tomaso Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):39–51, 1998. Cited on page 8.

[48] Pedro Tome, Ruben Vera-Rodriguez, Julian Fierrez, and Javier Ortega-Garcia. Facial soft biometric features for forensic face recognition. *Forensic science international*, 257:271–284, 2015. Cited on page 1.

[49] Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006. Cited on page 3.

[50] Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009. Cited on pages 18 and 21.

[51] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, 24:774–780, 1963. Cited on page 14.

[52] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. A primer on kernel methods. *Kernel Methods in Computational Biology*, pages 35–70, 2004. Cited on page 16.

[53] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001. Cited on pages 3 and 8.

[54] Hwanjo Yu and Sungchul Kim. Svm tutorial—classification, regression and ranking. In *Handbook of Natural computing*, pages 479–506. Springer, 2012. Cited on page 16.