UNIVERSITAT POLITÈCNICA DE CATALUNYA
Universitat de Barcelona
Universitat Rovira i Virgili

MASTER THESIS

Semantic face segmentation from video streams in the wild

Author:
Deividas SKIPARIS

Academic Supervisor:
Dr. Sergio ESCALERA

Industry Supervisor:
Dr. Pascal LANDRY

A thesis submitted in fulfillment of the requirements
for the degree of Master of Artificial Intelligence

in the

Facultat d'Informàtica de Barcelona (FIB)
Facultat de Matemàtiques (UB)
Escola Tècnica Superior d'Enginyeria (URV)

June 16, 2017

Declaration of Authorship

I, Deividas SKIPARIS, declare that this thesis titled, "Semantic face segmentation from video streams in the wild" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Universitat Politècnica de Catalunya
Universitat de Barcelona
Universitat Rovira i Virgili

Abstract

Facultat d'Informàtica de Barcelona (FIB)
Facultat de Matemàtiques (UB)
Escola Tècnica Superior d'Enginyeria (URV)

Master of Artificial Intelligence

Semantic face segmentation from video streams in the wild

by Deividas SKIPARIS

Semantic segmentation of faces in the wild is a challenging task, despite the fact there has already been extensive amount of research in this field. Most of the more advanced approaches require high computational resources, which are not always available, e.g. in remote machines or mobile devices. This paper presents a non-deep approach to segment faces in video streams in the wild and to extract representative colors of each semantic region. This is achieved by a combination of hand-picked methods, fine-tuned to work in tandem to produce accurate color extractions. Firstly, the face is located using a HOG detector and fitted with facial landmarks for alignment and eye region localization. The located face is then subtracted from the background and later segmented into hair and skin regions, by using two bipartite graph cut stages for each operation separately. Colors from each of the semantic regions are then extracted with the help of histogram analysis. Temporal feature has been added to mitigate errors from isolated frame segmentations. The proposed method operates at 5fps on a standard CPU and achieves mean color extraction error of around 11% in CIELAB color space. It was shown, that 11% error, although noticeable, perceptually has very little difference, and is more than sufficient for the application purpose of extracting accurate colors of semantic face regions from video streams.

Acknowledgements

First of all, I would like to thank Imersivo SL for making this thesis possible, for providing with support and excellent conditions to work and study.

Moreover, very big thanks goes to Dr. Sergio ESCALERA for his guidance, endless hours of discussions and for always keeping up the motivation.

Lastly, I would like to express huge gratitude to my family for comprehensive support, for making my studies possible and for always believing in me.

THANK YOU...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image segmentation is an important part of computer vision field, which focuses on assigning each pixel in an image into meaningful clusters, which represent either a single object, collection of objects or some important part of an item. The main purpose of segmenting an image is to simplify the representation for a variety of purposes, such as scene recognition, object detection and recognition, navigation, etc.

Segmenting a human face into semantic categories, such as hair, eyes, lips, ears, nose, etc. is a challenging task and only a handful of approaches have demonstrated satisfactory results. This is made even more difficult in the wild, with various occlusions, busy backgrounds and unpredictable face angles. (Note: the term face segmentation in this paper refers to face division into semantic categories, not background subtraction around the face.)

Nevertheless, human face labeling can serve several different purposes, such as audience evaluation, emotion detection or even such novel applications as interactive mirror. Such applications highly depend on a robust and accurate segmentation of a human face. However, this is still a very little explored area of computer vision, despite the latest advances in computational hardware, neural networks and availability of annotated data. There exists only a handful of attempts to achieve this task, both in neural computing (NN) [1][2] as well as in more classical approaches [3][4][5]. Convolutional Neural Network (CNN) approach is favorable for accuracy, while other approaches are more rapid on the CPU hardware, when GPU is not available or not feasible.

Furthermore, there exists no prior attempts to perform semantic segmentation of faces in videos. Previous attempts of video segmentation focus on object of interest identification [6][7] or scenery segmentation [6][8][9], mostly in a semi-supervised fashion, meaning the algorithm is provided with information of what to expect in a video sequence, however the inference of possible object shape and boundaries are performed automatically.

This thesis was performed in collaboration with Imersivo SL in Barcelona in order to develop a software package, capable of accurately extracting colors of face components. This package will be deployed into a vision system, capable of classifying shoppers into 4 profiles, aka seasonal palettes, based on colors of their attributes. This information is used by fashion experts to recommend clothes and accessories, which match user profile. The system proposed in this project will be part of the system, which will play a role of automated fashion expert and attempt to mimic recommendations of those experts. Since it will be installed into an already existing system, some restrictions on methodology has to be taken into account. Firstly, the developed library has to be written in C++ in order to be installed alongside already existing code. Secondly, there exists hardware limitations, as the code will be deployed on local CPU machine. A requirement from Imersivo is that the code should

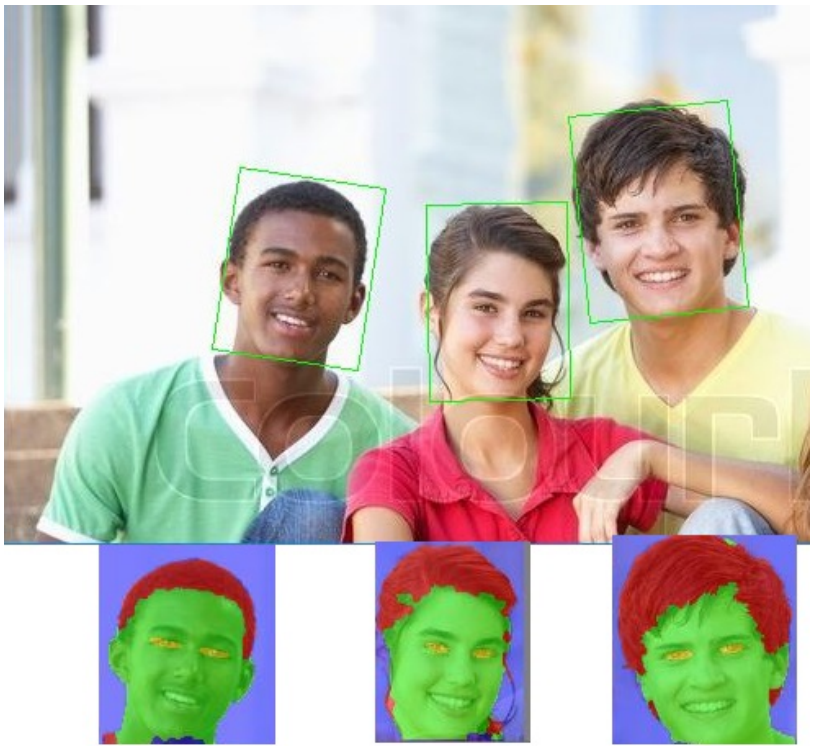

FIGURE 1.1: The proposed system for skin, hair and eyes segmentation

be capable of returning a result within 1 second on a standard up-to-date CPU with reasonable accuracy

This paper presents a non-deep approach for semantic face segmentation into 4 categories (hair, skin, eyes and background) in videos. A non-deep approach was chosen in order to adhere to hardware limitations of the customer. This system by design was made to infer accurate colors for each semantic region of faces of people standing in front of a camera. Such information can later be used in a variety of applications, like recommendation engines, fashion consulting or audience inspection. The proposed method is based on a multi-stage graphical model approach to achieve frame-wise segmentation, while color modeling is used to obtain time-based comparisons.

The pipeline starts with a face detection and 68 facial landmark fitting in an isolated frame. The video sequence segmenting is not capable of handling multiple faces, since multiple faces cannot be uniquely identified. Depending on the application, the most central face or largest face in the image can be used. A publicly available library *dlib* [10] is used for both of these tasks, which employs histogram of gradients with linear classifier for the former task and [11] implementation of head pose estimation for the latter. A two-stage bi-label graph cut method is then used to firstly subtract background and face and then segment the remained of the face into hair and skin. Gaussian color modeling is employed to calculate node weights, while relative node position defines edge weights. The difference between the two stages is that the first performs pixel-wise segmentation, while the latter - super-pixel-wise. Much of the region of interest (ROI) after face detection, is occupied by a face. While superior for segmentation tasks, super-pixelation of background can end up having too little samples to build an accurate predictive model, thus affecting the performance. The eyes-region is extracted using the fitted facial landmarks. Lastly,

pixel-wise color histograms are built from extracted regions in order to extract accurate representative colors.

The report is delivered in the following structure: Section 2 contains literature research, Section 3 introduces in detail the method used for semantic face segmentation, then Sections 4 and 5 explain testing setup and results achieved. It is concluded with identified shortcomings and proposed improvements.

Chapter 2

Related Work

2.1 Face Detection and Landmark Fitting

Face detection in the wild is a very typical research topic in computer vision, therefore there exists a huge number of papers. [12][13] have performed excellent surveys and comparisons between different approaches for this problem. One of the most referred to face detector is undoubtedly the Viola&Jones face detector [14]. The popularity and ubiquity of this classifier is enabled by open-source implementation in OpenCV library [15]. This detector uses integral image to calculate simple Haar-like features over gray images, using the scanning window approach. Together with boosted weak classifiers, this detector achieves real-time face detection. However, the achieved performance is only satisfactory, due to arbitrary poses [16]. While more complex features can be introduced to attempt to compensate, this would significantly increase the computation resources. Since then there has been significant advances using Deformable Part Models [17], which has established itself in the industry as a standard model for object detection. There exists a number of variations for this approach, such as deformable templates [18][19][20][21] or part-based models [22][18][23][24]. However, in general it provides superior performance over the HOG detector [25] as by design it is made to tackle huge intra-class variability of objects in the wild [26]. For components, which do not exhibit this variation, the performance of HOG and DPM is comparable. There has been a few attempts to make neural network based face detectors [27][28], however the results provided are questionable, due to datasets used for experimentations and it is unclear how these models would compare to other, more recent approaches.

Facial landmark detection has a diverse research history. Some of the older methods include Active Appearance models [18][29], multidimensional morphable models [30] and template tracking [31]. Due to heavy computational resources required for the above approaches, they have not gained popularity. Instead, supervised gradient descent [32] or ensemble of regression trees [11] provides a robust and very fast facial landmark fitting, reporting speeds of up to 1ms. As with a lot of computational vision tasks nowadays, CNNs [33][34] have been shown to provide a very good performance in a broad range of poses and virtually invariant to occlusions.

2.2 Segmentation

2.2.1 Object segmentation

Understanding the environment in photos and videos and dividing it into meaningful clusters of pixels is still a challenging task. Over the years, there has been a variety of methods explored to achieve scene segmentation. The clear majority of early

research on this topic is found on the use of classical features as SIFT [35] or HOG [25], with varying degree of complexity of implementations. SIFT/HOG features have been used with over-segmentation methods [36][37], however the approach is highly sensitive to the quality of over-segmentation, which can often miss important boundaries. Also, it was used in union with graph cuts [38][39] to achieve object detection and segmentation under complex background and invariant to occlusions. Further improvements to SIFT/HOG descriptors were done by [40], by combining them with Dense Scale and Rotation Invariant Descriptors (SID) and introducing Soft segmentation [41]. This even further improves segmentation ability under busy, moving background as well as occlusions. [42][43] use custom type feature descriptors, together with active shape fitting to detect object boundaries. However, these methods are limited to single objects per scene and suffer from false boundary detections. With introduction of Kinect and other depth sensors, image stereo information has become easily accessible. This has also benefited scene segmentation algorithms, as it provides additional and very important piece of knowledge. [44][45][46][47][48] all show depth images provide significant improvements over respective methods without it. Nevertheless, the most notable advances in semantic image segmentation are contributed by the application of convolutional neural networks (CNN). [48][49][50][51][52] have proven convolutional neural networks can be successfully trained end-to-end to generate pixel-wise probabilities.[53] employs recurrent feed-forward network to achieve semantic segmentation, however the results do not compare well to the accuracies obtained by the convolutional counterpart. [48] even further extends CNN with a usage of depth map even further, reportedly increasing the accuracy by 20% relative.

2.2.2 Semantic Face segmentation

In the most recent work regarding face labeling in the wild, the most common approach is by means of ANNs. Warrell and Prince [54] uses CRF with Adaboosted unary classifier and epitome priors, however this method suffers from poor occlusion detection. Kae et al. [5] improves on the CRF-based segmentation by replacing epitome priors with Restricted Boltzmann Machine priors. This provide a significant improvement due to inherent ability of RBM to model global shapes. Moreover [55] uses DPM to achieve hierarchical face parsing, then employs new deep learning strategy to achieve facial component labeling. Liu et al [1] also employs a face prior and uses it as an additional input into a multi-objective CNN. This significantly aids the regularization, allowing to build much smaller networks with comparable results. [2] demonstrates that fusing CNN with Dense CRF provides excellent segmentation results for a very broad range of applications, including face labeling. Facial segmentation in general deals at either super-pixel or pixel level, however [4] introduces a new approach to this problem, by considering patches of images instead, in a similar manner as SIFT does. According to the authors, patches are more discriminative than pixels and allow to build more accurate probability maps of classes. [56] employs exemplars of faces to achieve face part segmentation. The hair is not part of segmentation in the original approach, however authors provide a possible extension to achieve this. Chefler & Obodez [57] use color modelling with Baysian framework to achieve segmentation, however the approach provides less robustness as the previous methods due to background clutter and pose variations. Apart from a few deep approaches, the face segmentation is usually performed using graphical models. Since due to hardware limitations, deep methods cannot be

employed, this confirms that graph cut is a viable approach for this particular application.

2.2.3 Video Segmentation

Generally accepted definition of video segmentation is separating foreground objects from the background [58][59][7]. The main idea is spatio-temporal label propagation in a video, allowing for object tracking, gesture recognition. There are interactive video segmentation methods, which accept manual user labeling for some initial frames and attempt to propagate these labels [60][61][62]. However the obvious limitation of this approach is necessity for manual user input. A more robust technique to overcome this, is by the means of ranking object proposals [58][63][64][7]. This technique is based on finding similar recurring segments in a video and building a dynamic appearance model to achieve spatio-temporal segmentation [7]. This paper simplifies the definition of video segmentation, due to the nature and conditions of application, i.e. face segmentation of people standing in front of a camera. This allows to achieve spatio-temporal object localization with face detection techniques and use temporal segmentation as a self-validation mechanism.

2.3 Proposed system

The system proposed in the paper benefits from collection of above methods in order to achieve color extraction by segmentation. First of all, the system is equipped with a face detector, as it is vital for segmentation to localize the ROI. Since face is a highly-structured component, HOG face detector was employed due to having a good trade-off between accuracy and computational complexity. Moreover, the system has been equipped with facial landmark detector in order to allow correction for head pose variations prior to segmentation. The fitted landmarks will also be used to accurately segment the eyes-area without the need for segmentation algorithms. This will reduce system complexity and allow for quicker inference.

Segmentation will be a crucial part of the system, because accurate color extraction requires pixel-wise representation of the parts of interest. The inspiration for the proposed method has been taken from graph cut method, as it has already been successfully employed for complex segmentation tasks. The only issue with this approach is its high computational complexity $O(n^2m)$ **??** due to large number of pixels in each face ROI. Nevertheless, this effect is significantly mitigated by using SLIC image over-segmentation.

Lastly, temporal feature in this algorithm will be used as a way to self-check and prevent poor segmentation due to dynamic nature of faces. In the wild they can become occluded due to dynamic background, eyes are constantly blinking and faces change due to head pose. Because of this, selecting arbitrary frame for color extraction might return inaccurate results.

Chapter 3

Proposed System

The system presented in this paper consists of three main stages: Face-Background Segmentation, Skin-Hair-Eyes Segmentation and Color Extraction and Correction (Figure 3.1). In the first stage the face of a subject is detected and separated from the background. In the following stage the face is divided into 3 semantic regions (skin, hair and eyes) and finally colors of each of those regions are extracted and optionally corrected. The Color Correction is used by the segmentation algorithm to obtain the true colors of face regions, by compensating lighting, shadows and camera-specific settings. The Color Correction procedure presented as a separate package, since the evaluation of the two systems has been made separate.

3.1 Face Segmentation

In order to achieve semantic face segmentation, the algorithm performs the following steps

1. Face-Background segmentation
2. Skin-Hair-Eyes segmentation
3. Color Extraction

3.1.1 Face-Background segmentation

Face detection and Facial landmarks

The segmentation of any image starts with locating faces in the images. The program supports two face detection modes, namely 'All Faces' and 'Main Face Mode'. As the names suggest, according to the selected mode, the algorithm finds and segments all faces in an image or only the so called 'Main Face'. The latter could be set to either select the largest face in the image or select the face closest to the centre of the image. This is especially useful when the software is being used in a busy environment with other people in the background. For the further report, it is assumed only one face is detected as the same procedure applies for each face sequentially. Once the face has been found in the image, it is fitted with 68 facial landmarks (See Figure 3.2) using the ensemble of regression trees as presented by Kazemi et al [11], which exhibit state of the art performance among the classical CV approaches The implementation of face detection and facial landmark fitting was used as provided by dlib.net C++ library [10].

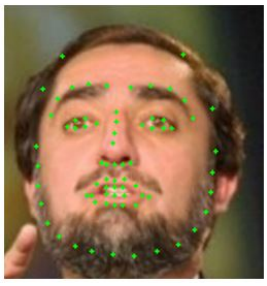

FIGURE 3.2: Fitted facial landmarks

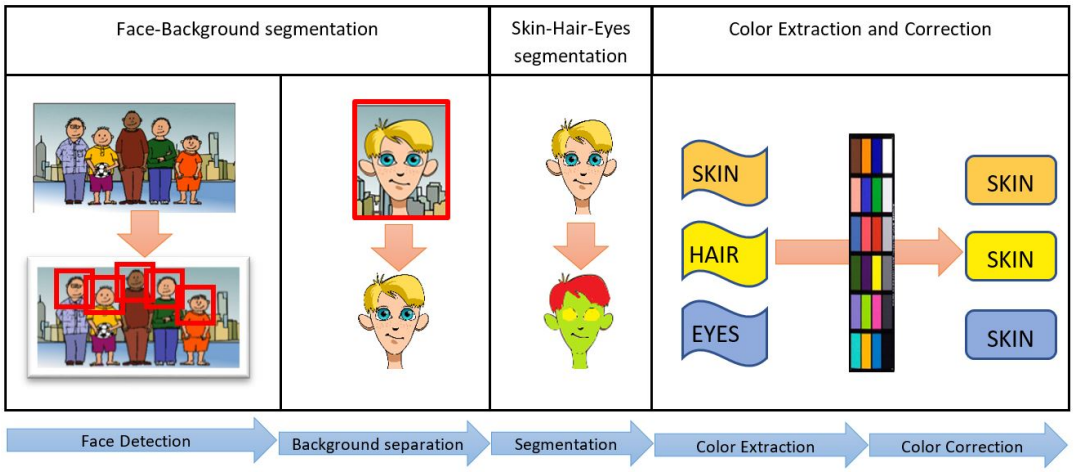

FIGURE 3.1: Proposed system functional pipeline

FIGURE 3.3: Bounding face rectangles. White - face area. Red - background area

Facial landmark fitting serves several purposes in the system. Firstly, it helps to understand the face pose and allows to discard the faces, which are in problematic poses. Secondly, fitting landmarks around the eyes revokes the need to perform eye-area segmentation, as the landmarks already provide very accurate eyes locations. Lastly, the facial landmarks allow to accurately calculate the size of the face, meaning precise face regions of interest (ROI) can be extracted for color modeling in later stages. Each face in the image is defined by two ROIs as shown in Figure 3.3. Face region – area of the image containing the face, as bounded by the white rectangle. Also Background Region – area of the image just outside the face, as bounded by the intersection of $BackgroundRegion \cap \sim FaceRegion$. The Background Region is simply a 1.5x scaled Face Region, both sharing the center-point of the bottom side and orientation (always up-right). The Face Region is a derivation from the facial landmarks calculated earlier. According to the fitted landmarks, this rectangular area is tailored in such way to always contain all of the face details.

Background Removal

After the face has been detected and fitted with facial landmarks, the first stage of segmentation can be performed. In this step the face is separated from the surrounding background (Figure 3.4). To perform this, Gaussian Mixture Models (GMM) are used to make pixel-wise differentiation whether each pixel in the face region belongs to the face or to the background. Two separate GMM models are built for background and face. The pixels used for training each model come from Face Region and Background Region as defined in previous section. These learned Gaussian distributions from the adaptive mixture model determine which class each pixel in the Face Region most likely belongs to, as described by Staufer et al [65]. From this pixel distribution, a graph is built, where each pixel represents a node and node weights are described by probabilities for belonging to either class. Then a min-cut max-flow algorithm divides this graph to give the most likely separation of foreground-background [66]. To implement the Staufer et al [65] method, GrabCut function from

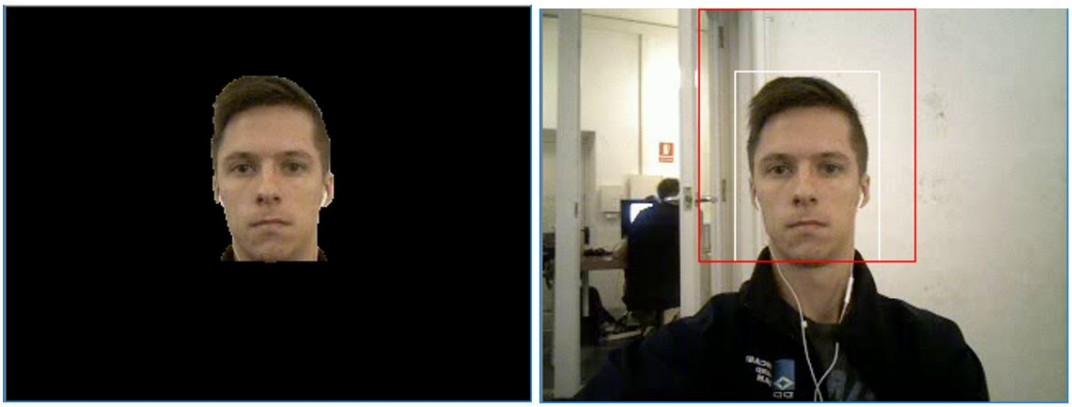

FIGURE 3.4: Face after background subtraction using bounding rectangles for color modeling

OPECV 2.4 computer vision library was used [15].

3.1.2 Skin-Hair-Eyes segmentation

Once the face is segmented from the background the next stage is dividing it into 3 semantic categories: Skin, Hair and Eyes. However, modeling eye-region with a color-based model is impractical as due its size, the region most likely will not exhibit any distinct color/texture characteristics. As the consequence, segmentation will most likely be inaccurate. Since the eyes region was already located when fitting facial landmarks, it will be extracted from there. A method, analogous to the one in FGD-BGD subtraction, was used for Skin-Hair segmentation, however in a much more controlled manner. It was initially considered to apply GrabCut as in the FGD-BGD subtraction to acquire Skin-Hair segmentation, however it was experimentally proven to be inferior by Graph Cut (See Figure 3.5.

FIGURE 3.5: Comparison of segmentation quality. Multi-class Graph-Cut vs. OpenCVs GrabCut

GMM and EM

The first step in this process is to build a model for predicting whether pixels in the Face Region belong to Skin or Hair. Exactly as per previous FGD-BGD segmentation, Gaussian mixtures together with Expectation Maximization algorithm are used to build a two separate GMM models for skin and hair. Upon merging the two models, the joint predictor is obtained, capable of distinguishing between the two semantic

categories of a face. Skin pixels for GMM_Skin are selected from the region of face, which is most often uncovered by beards, mustache or accessories. The region between 'under-eyes' and nose is identified as Definite Skin region and is used for this purpose. For learning the GMM_Hair model, hair pixels must be used, however their acquisition is not trivial. In order to have a robust process for obtaining a hair pixel sample, the following process was used (Graphical representation in Figure 3.6):

1. Take some fraction of pixels from the top of the head.
2. Calculate a N-bin/channel CIELAB color histograms for all the sampled pixels $(HIST_Forehead)$.
3. Calculate a N-bin/channel CIELAB histogram from the Definite Skin pixels $(HIST_Skin)$
4. Subtract the histograms – any non zero bins in the $HIST_Skin$ are removed from $HIST_Forehead$.
5. The pixels remaining in the histogram are treated as hair pixels.

FIGURE 3.6: Subtraction of histograms for hair color modeling

?? The selection of N bins per channels for histogram subtraction is justified experimentally in later section. Next step is to build a model, capable of predicting, to which Gaussian mixture some pixel most likely belongs. For this purpose, the built-in functionality of OpenCV [15], namely 'EM' method, was used. This implementation of OpenCV is based on iteratively building Gaussian mixtures from provided training pixels by employing expectation maximization method [67]. Two models for each semantic category were build and later merged to provide the prediction functionality for the later stages of segmentation.

Superpixelation

Once color models have been built, and the face is subtracted from the background, it can then be divided into superpixels using the SLIC method as proposed by Achanta et al [68] This very fast image division into small clusters gives a number of advantages:

1. Super-pixelation of an image captures most of the important boundaries in the image, therefore will provide a more accurate segmentation.
2. Number of superpixels is much smaller than the number of pixels, meaning the prediction time is dramatically decreased. The 100x150 Face Region is divided into 300 – 2500 superpixels, depending on the settings used. This means the prediction time of the model is reduced by at least 7 times. Time for superpixelating an image is negligible, compared to time saved when building GMM. Upon retrieving the superpixels (SP), each SP is then represented by the mean color of contained pixels. From this point, the Face Region becomes a list of mean SP colors.

Graph-Cut

Obtaining the list of superpixels for the face region and GMM model for Skin and Hair allows to construct a graph. The nodes in the graph are superpixels and this graph is divided into 3 regions to acquire Skin-Hair-Background segmentation of the Face Region. The background nodes have already been identified in the initial FGD-BGD segmentation, so are hard set and only included for simplicity of programming. To perform the most optimal graph division, the Graph Cut method was used. This method tries to find a way of dividing the graph, which minimizes the global energy, thus giving the most likely segmentation of an image [69]. However, Graph cut method is inherently binary, so when there are multiple labels, the process can become computationally very expensive. To overcome this, α-expansion minimization algorithm as suggested by Boykov et al [69] was used. α-expansion uses graph cut techniques from combinatorial optimization to obtain multi-label graph segmentations. The cuts of this move are so strong, they are guaranteed to be within the global minimum by a known factor. This means with only a few iterations the algorithm is capable of achieving a close to optimal segmentation. The energy to be minimized is by the Graph Cut algorithm is defined as per Equation 3.1

$$E(f) = \sum_{p \in P} D_p(i_p, f_p) + \sum_{p,q \in N} V_{p,q}(f_p, f_q) \tag{3.1}$$

The first term known as unary (or data) term, while the second one is pair-wise (or smoothness) term. Unary term is responsible for ensuring the current assigned labels are coherent with the observed probabilities. It penalizes the label f_p if it is different to what the GMM model has suggested. For this system, the unary cost (unr_cost) for a node s for a specific label *l* was defined as per Equation 3.2:

$$D_p(i_p, f_p) = -\sigma * \log(PROB(i_p, f_p) \tag{3.2}$$

Where $PROB(i_p, f_p)$ is probability of a superpixel i_p belonging to a class f_p and σ is a constant positive multiplier. The pair-wise term in the energy equation is responsible for ensuring the overall labeling of the graph is smooth, i.e. it penalizes two neighboring sites if their labels are too different. Regarding the smoothness term, there are 3 conditions to define it [70]:

1. $V(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$
2. $V(\alpha, \beta) = V(\beta, \alpha) \geq 0$
3. $V(\alpha, \beta) = V(\alpha, \gamma) + V(\beta, \gamma)$

The first two conditions suggest that pairwise cost for two different labels has to be more than zero, otherwise they are the same labels. The third point suggests a triangle rule, that a shortcut is always cheaper or similar than taking the whole path [70]. If all of the 3 conditions above are satisfied, the pair-wise term is said to be metric and can be used with alpha-expansion algorithm. Otherwise it is semi-metric as can only be used with alpha-beta-swap. For the face segmentation application, it can be assumed that each label (Skin-Hair-BGD) has equal opportunity of being next any of the other labels, thus $V(\alpha, \beta)$ for any two dissimilar labels can be set to some constant τ. To set the pair-wise cost in the system, the Equation 3.3 was used:

$$V(\alpha, \beta) = \begin{cases} \tau & \text{if } \alpha \neq \beta \\ 0 & \text{if } \alpha = \beta \end{cases} \tag{3.3}$$

With this setting, the 3rd condition of the smoothness term being metric is satisfied, thus the alpha-expansion application for this setting is justified. Note: By adjusting σ and τ it is possible to control how important is the data term over the smoothness term.

3.1.3 Color Extraction

Face region segmentation into 4 semantic categories (Skin-Hair-Eyes-Background) enables to extract color for each of the regions of interest. The application up to this point generated a binary mask representing each of these categories. Before performing color extraction from the regions, defined by the binary masks, it is important to erode the masks to ensure the edge pixels are not considered.

Kernels used for Skin/Hair and Eyes mask erosion. Skin/Hair = 4x4px; Eyes = 2x2px, both with central anchor points. For skin and hair masks, the local minimum operator using a 4x4 kernel was applied. Since eyes is a much smaller region, such large erosion would remove important details. Thus, the kernel was reduced to 2x2 square (Figure 3.7.

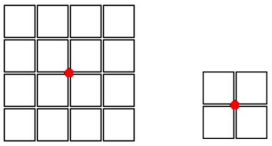

FIGURE 3.7: Kernels for skin/hair(left) and eyes(right) mask erosion

The difficulty with the hair of a person, is that the head hair and facial hair are likely to be of different color. Therefore, a color representing hair on the whole head, might be incorrectly calculated due to this reason. In order to avoid this issue, a simple heuristic method has been applied to the hair mask – only the hair above the eye-line has been considered as shown Figure 3.8.

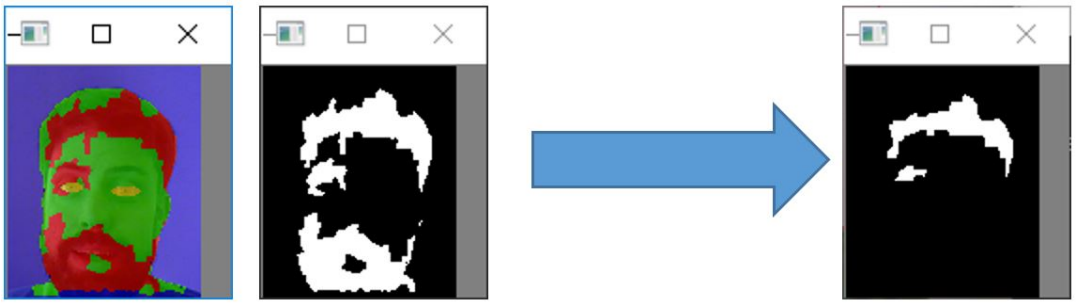

FIGURE 3.8: Effects of hair thresholding below eye-line

With the help of this simple heuristic, not only the facial hair is not taken into account, but also effects of poor segmentation are mitigated. Figure 3.8 above best illustrates this method. Using the modified masks, color histograms of N bins per channel were built for each of the regions and normalized using L1 norm. From these histograms, a weighted average of X largest bins was calculated to get a final color for a semantic category as per Equation 3.4.

$$color_c = \sum_{n=1}^{X} \omega_p^c * bin\_color \tag{3.4}$$

, where ω_p^c is the size of the Top-n largest bin for a semantic category c and bin_color is the representative color for that bin. Representative color for each bin is in the middle of D-dimensional range covered by the bin. To illustrate, Figure 3.9 shows a 1-D histogram with 4 bins, each bin being represented by the value

in the middle of the bin range. This method introduces error in the color calculation, which is inversely proportional to the number of bins and is equal to zero as the number of bins is equal to 256, such that $\lim_{n \to 256} ERROR = 0, n \in \mathbb{N}$

3.2 Temporal Segmentation

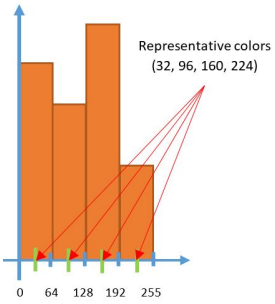

FIGURE 3.9: Histogram binning

Section 3.1 describes the segmentation procedure of a single image. However, the system is designed to be capable of processing multiple images from video streams and selecting best frames. Such analysis of multiple frames of a video stream over time is called temporal segmentation. Such temporal segmentation is achieved by segmenting multiple frames over some period of time and comparing the results between themselves in order to distinguish, if some of the frames have been incorrectly segmented (See Figure 3.10. The performance is significantly affected by the angle at which the person's head is positioned in relation to the camera, background clutter, rapid movements and other factors. Thus, it is expected they will be mitigated by applying temporal segmentation.

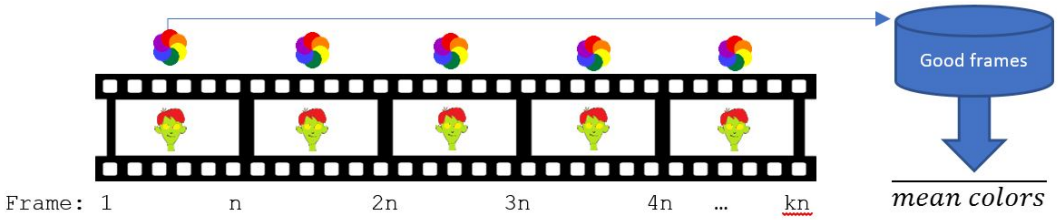

FIGURE 3.10: Temporal segmentation color extraction method

The procedure of temporal segmentation is as follows:

1. Video is loaded to the system
2. Every n^{th} frame is segmented. For all the temporal analysis tests for this work, n=50 was chosen. In a normal 24fps video, this is roughly one frame every 2 seconds. Such period of time is long enough to ensure that any adverse factors for segmentation have disappeared (if present), while still short enough to ensure the scene does not change significantly.
3. Color is obtained for each semantic category using the same process as described in Section 3.1.3
4. The extracted colors are saved into a time series vector.
5. The extracted colors from current frame are compared to colors in two previous time steps (current-1 and current-2).
6. If the results are within a pre-determined threshold, the values are saved as good samples.
7. After sufficient good samples have been collected (or after segmenting certain amount of frames), the mean of good samples is calculated to get the final color for each semantic category.

3.3 Color Correction

A very important part of recognizing the true colors of facial attributes of a person in front of a camera, is being able to diminish the variations due to external factors. A picture taken with the same camera parameters at different lighting conditions can have large variation in the color registered by the sensor of the camera. Therefore, it is necessary to perform some form of calibration in order to get the true color. There are two ways such calibration can be achieved. First, is calibrating the camera in such way, that the exterior conditions would be compensated by adjusting parameters of a camera being used. However, this method was not chosen due to the obvious difficulty, that he calibration is too complex to achieve and is camera specific. It is possible that even cameras of the same brand and model would give different results depending of the amount of time used and some internal discrepancies. This means every time a new camera is seen by the program, it would have to be recoded. This is very labor intensive and does not provide any advantage over other calibration methods. The second method of color calibration and the one chosen for the pipeline of this program is by adjusting how the software interprets colors, registered by a camera. This method can be applied to any camera and only needs to be performed upon significant change in exterior conditions. This functionality is achieved with the help of so called Color Calibration Chart (CCC). CCC is a physical board with a number of different known colors on it. These charts are mostly used in professional photography and are made for this purpose of calibrating high-end cameras. Such cameras have internal functionality of color calibration and CCCs are used to get true colors of a scene regardless of the conditions.

Nevertheless, the emphasis of this master project is color extraction through segmentation from a given media (pictures, recorded videos, live stream). While color correction is important step for Imersivo's application, it was decided to leave the Color Correction step out of the scope of this project, in order to have a more direct focus. Therefore, the color correction algorithm is not subjected to exhaustive testing to prove its efficiency. Instead, only examples of the code in application are provided in 4

3.3.1 Functional Architecture

The color correction for this system was performed by building a linear regression model. The model was trained onto the extracted color values, taken from an image of the Color Calibration Chart. Then new color values of skin, hair or eyes were predicted using the trained model. To extract real values of the CCC from the camera, firstly an image of this chart is taken. The user then manually points the 4 corners of the chart on the image using mouse, in order to specify its position. This fits a grid of on the chart, from which colors are sampled (See Figure 3.11).

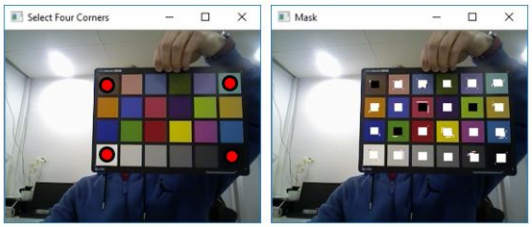

FIGURE 3.11: Reading color correction chart

The grids are fitted according to the size of the chart in the image, while also ensuring the colors are sampled only from the middle of each section of the chart. This way, any inaccuracies in grid fitting and close-edge shadows are mitigated. The pixels in each sampled area are averaged to give 24 sampled colors. Since each of

the sampled colors has a ground truth value associated to it, this enables to build a linear regression model. The built model, together with additional meta information is saved into a model file, which can be used in Face Segmentation module for rapid color correction.

Chapter 4

Experiments and Results

The following section introduces the testing procedure used for validating the system. Firstly, the testing data is described, followed by approach taken for system optimization. Lastly the results achieved are presented.

4.1 Data

In order to test the segmentation pipeline, videos from the YouTube personality dataset [71] was used. This dataset consists of a collection of "YouTube vloggers that explicitly show themselves in front of a webcam, talking about a variety of topics, including personal issues, politics, movies, books, etc." [71]. The dataset was designed to be used for predicting the apparent personality traits (the big 5) [72], however will serve very well for this project, as it will have people looking at the camera from a convenient distance, having various skin and eye colors, various hair colors and amounts, various genders and ages. 100 random videos have been selected from the dataset. The statistics of the videos are as presented in Table 4.1

Gender	**Age**	**Skin**	**Accessories**	**Hair Amount**	**Hair**	**Facial Hair**
Female (59)	20s (44)	Dark (15)	Glasses (7)	Little (10)	Dark (74)	Heavy (4)
Male (41)	30s (37)	Light (85)	Hat (10)	None (6)	Light (14)	Little (14)
	40s (15)		Head-band (1)	Normal (84)	Mixed (2)	None (77)
	50s (4)		None (82)		None (6)	Normal (5)
					Red (4)	

TABLE 4.1: Data Statistics

The data has been split into 0.9/0.1 training/validation lots by random sampling without replacement. The training data is used to find optimal parameters of the system, while validation data is used to confirm those parameters and calculate results.

4.1.1 Ground Truth Marker

Since there are no publicly available videos, which have skin, hair and eyes colors marked as ground truth, the videos have been manually labeled. A special "Ground Truth Marking (GTM)" tools has been created in order to aid in this process. The GTM tool (Figure 4.1) allows to mark pixels belonging to any semantic category in any selected frame in the video. Once all the desired frames in a video are marked up, the tool collects all the pixels from all the frames for each face part and calculates the mean color. It was decided that building a histogram of colors at this

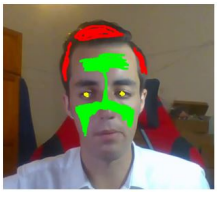

FIGURE 4.1: Ground Truth Maker process

point is redundant, since the person performing manual labeling
is capable of marking only pixels representative for each category.
The person performing the labeling should not select any of edge pixels due to possible discolorations.

4.1.2 Parameter search

The system of temporal face segmentation has 21 tunable parameters, which in one way or another can have influence on the semantic segmentation and/or color calibration. Out of those 21 parameters, 9 has been selected as having the most importance for the whole pipeline, thus it is vital to ensure the optimal values are found. For this purpose, the exploration of parameters has been performed. The parameters shortlisted for optimization are displayed in 4.3

Since the number of exploratory iterations grows exponentially with the number of parameters and number of their possible values, it would have been very inefficient to perform tests for all possible combinations of parameters. Instead, to reduce the training time, the parameters have been divided into 5 distinct categories and tested separately (Table 4.2).

Group1	**Group2**	**Group3,4,5**
Unary	hair_sample_size	top_N_colors_C
Pair-wise	hair_TH	TH_C
SP_amount	hist_bin	
	hist_bins_2	

TABLE 4.2: Parameter search groups

These three categories have very little or no influence on parameters in other groups, while being closely related to the ones in the same group. Groups 3,4 and 5 have the same parameters varied for skin, hair and eyes independently, assuming the values for other two categories does not have any influence on it. Such way, the number of iterations are reduced from $432 * 10^6$ to a much more manageable 640. For testing, the Intel NUC model NUC5i5RYK, with Intel® Core™ i5-5250U Processor (3M Cache, up to 2.70 GHz) with 16GB RAM computer is used.

No	System Part	Parameter	Description	Values Explored
1	Face Segmentation	Unary	Unary term in graph cut algorithm	1, 3, 5, 7, 9, 11
2		Pair-wise	Pairwise term in graph cut algorithm	1, 6, 11, 16, 21, 26, 31, 36, 41
3		SP_amount	Number representing the size of superpixels. The smaller the number, the larger are the superpixels	350, 700, 1050, 1400, 1750
4		hair_sample_size	The ratio of P/TP for hair color modeling, where P – no of pixels from the top of the head, TP – total pixels in the face area. The larger the ration, the more pixels are considered.	0.05, 0.1, 0.15, 0.2, 0.25
5		hair_TH	Threshold for ratio of R/hair_sample_size, where R – number of pixels remaining in the hair model after histogram subtraction. Used to determine if a person is bold. If ratio < hair_TH – the person is bald.	0.05, 0.15, 0.25, 0.35
6	Color Extraction	hist_bin	Number of bins in the histogram, used for extracting color.	16, 32, 64, 128
7		hist_bins_2	Number of bins in the histogram, used for building hair model.	16, 32, 64, 128
8		top_N_colors_C	Number of most occurring colors to be extracted from histograms for color calculation. C – semantic category.	1, 2, 3, 4, 5
9	Temporal Analysis	TH_C	Threshold used for comparing similarity of segmentation results with previous time steps. If $TH < \|\|color^0 - \vec{color}^{-1}\|\|$ or $TH < \|\|color^0 - \vec{color}^{-2}\|\|$, the segmentation results are discarded. C – same as for top_N_colors_C	5, 10, 15, 20, 25, 30, 35, 40, 45, 50

TABLE 4.3: Parameters for optimization

4.2 Results

4.2.1 Color extraction

Performing the parameter optimization, has highlighted that the system performs the best with the settings as shown in Table 4.4

No	System Part	Group	Parameter	Best Values
1	Face Segmentation	Group1	Unary weight	9
2			Pair-wise weight	31
3			SP_amount	350
4		Group2	hair_sample_size	0.15
5			hair_TH	0.35
6	Color Extraction		hist_bin	16
7			hist_bins_2	128
8		Group3	top_N_colors_{C}	4
9	Temporal Analysis		TH_{C}	50

TABLE 4.4: Best parameters for each variable of the system

The validation cases when run with optimal parameters for each group separately, have generated the results given in Table 4.5:

	Group1	Group2	Group3
mErr	45.55	30.07	40.74
Stddev	32.76	28.97	33.96
% error	13.21	9.36	12.09

TABLE 4.5: Results per group

The performance on the validation videos with the best settings are is given in Table 4.6:

	Overall
mErr	31.92
Stddev	26.58
% error	11.03

TABLE 4.6: Overall performance

4.2.2 Color Correction

As mentioned in Section 3.3, the actual color correction procedure and its performance is out of scope of this project. Instead, Figure 4.2 demonstrates the color correction algorithm in action by showing images before and after the correction. The images in the figure provide an understanding of what to expect from this functionality and how colors get adjusted.

For all three image pairs in Figure 4.2 the error (Euclidean distance) between the sampled colors from CCC and ground truth values as per equation 4.1. The results in Table 4.7 are provided in the same order as in Figure 4.2 for comparison.

$$error = \sum_{n=1}^{24} \|sampled_n - GT_n\|_2 \tag{4.1}$$

Error	Pic1	Pic2	Pic3
Before	460.95	481.23	443.52
After	139.05	144.25	141.13

TABLE 4.7: Errors before and after correction for images as per Figure 4.2

FIGURE 4.2: Color Correction in action. Top row shows original image with color correction chart (CCC) for reference. Bottom row shows corrected images. Right image - all CCC colors Sampled, GT and Corrected. Gamma = 2.2, Polynomial degree = 1

4.3 Discussion

The results achieved from validation run have highlighted, that the system on average produces an error of 32, which represents Euclidean distance in LAB space. This is a large deviation from acceptable standards in color modeling, since the 'just noticeable' difference for LAB color space is considered 3.2 [73]. Moreover, the standard deviation of 26.6 suggests that the system exhibits a lot of variance, thus the error can fluctuate significantly depending on the case. Since no previous approaches are known, which aim specifically for color extraction through segmentation, the proposed system is difficult to be compared to current state of the art methods.

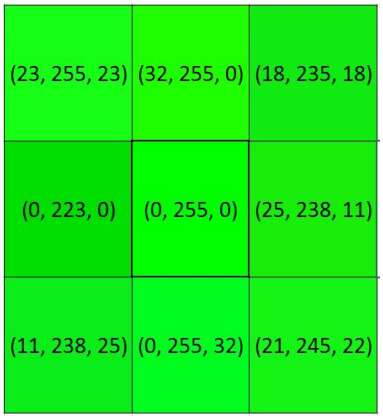

FIGURE 4.3: Color error visualization. The middle square is reference color, while others differ from it by around 32 in Euclidean space

There are a variety of areas, which can potentially generate this error, from which the majority attributes to poor segmentation of a face (See Figure 4.4). Firstly, the system lacks the ability to filter out face occlusions, meaning the colors can be calculated incorrectly in the presence of very ubiquitous hair/face accessories. Even though the system is capable of discarding small accessories, the larger ones, e.g. a cap, will definitely produce color extraction errors. Secondly, having a busy and dynamic background could result in poor background-foreground segmentation. This will propagate to later stages in the system, forcing pieces of background to be segmented as either skin or hair. Also, a possible

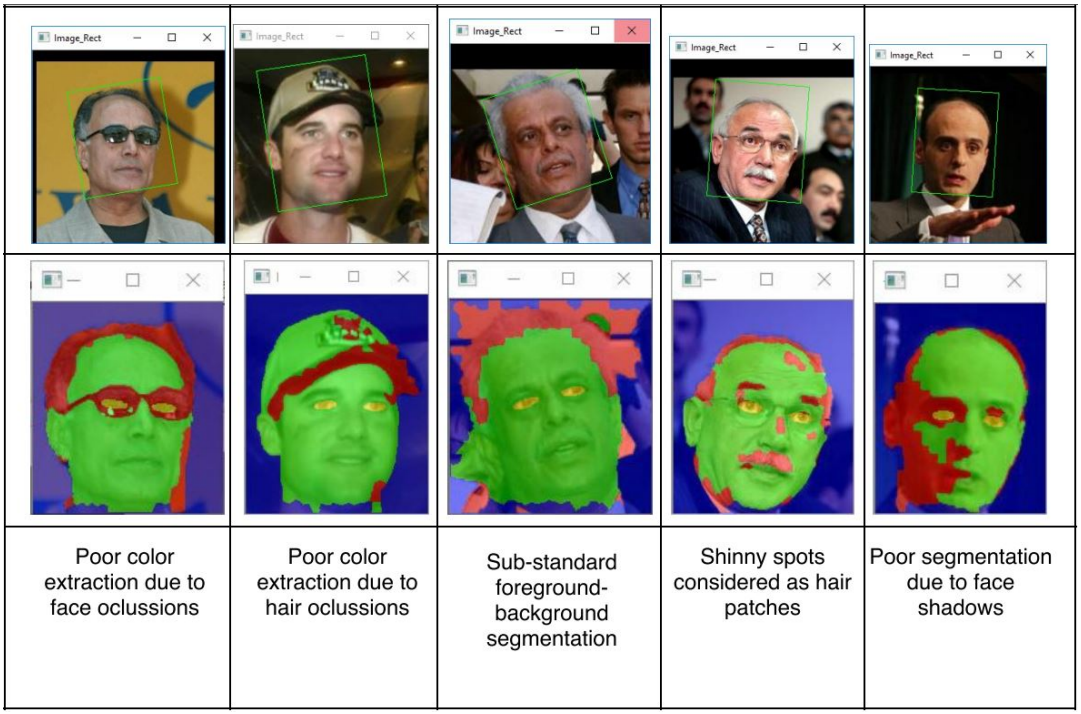

FIGURE 4.4: Examples of poor color extraction due to poor segmentation

source of error could be lighting conditions. In an environment with very strong light sources, skin might have shinny spots or hair become lighter. Such regions can potentially experience poor segmentation, due to color models incorrectly identifying those regions.

Nevertheless, the obtained error is compensated by the fact the system is capable of running around 5fps on a standard CPU, which satisfies customer requirements for this system. Moreover, since the extracted colors will be used for fashion recommendations, error = 32 (11%) in color extraction is still sufficiently low. The system operating at this error rate will be capable of distinguishing at 16bit (4096 colors) color resolution. Such resolution will provide sufficient accuracy to recommend items based on colors of facial attributes as well as classify subjects into seasonal pallets [74]. Figure 4.6 shows examples of good segmentation quality, which resulted in low color extraction error. Moreover, while $error = 32$ seems notable, the Figure 4.3 demonstrates that perceptually such error is not that significant. The figure shows the difference in color tones with error of 32: the middle square has the reference color (RGB), while others differ by around 32 in the Euclidean RGB. Such error is noticeable when shown in comparison, however in reality for the designed application, it will not make any significant difference.

FIGURE 4.5: imCube in La Maquinista Shopping Center in Barcelona, Spain

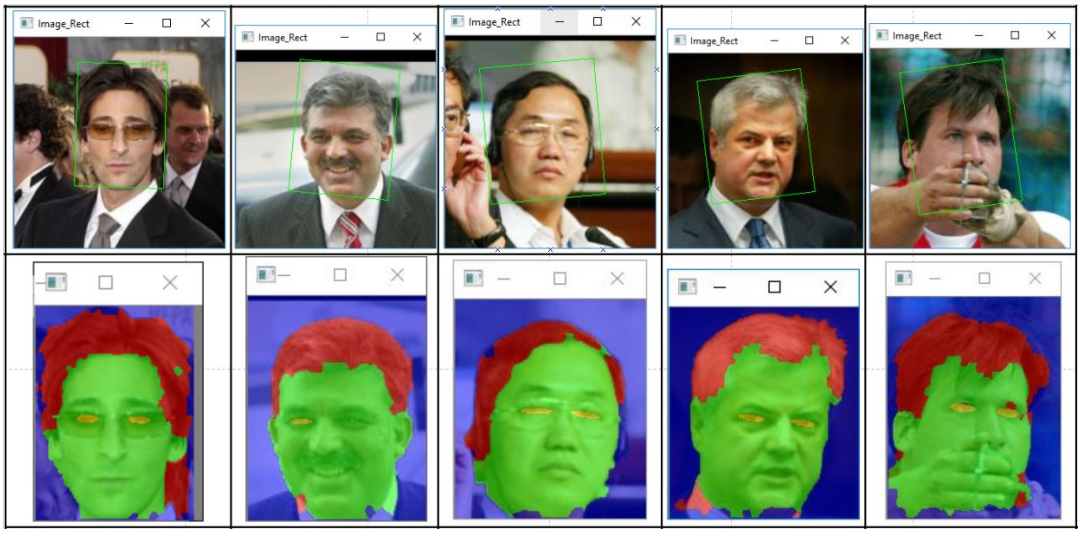

FIGURE 4.6: Examples of well segmented faces, which result in very good color extraction performance.

4.4 Real Scenario Usage

The designed code was implement into Imersivo's flagship product imCube. (See Figure 4.5. imCube is currently being developed as a shopping platform designed to enable brands to reach customers in otherwise inaccessible/inconvenient places, e.g. airports, stations or even abroad. It was installed in one of biggest shopping centers in Barcelona for proof of concept. Since this was only a demo run, only a handful of samples were collected for validation of color extraction functionality. Figure 4.7 below demonstrates example segmentation quality achieved by the system, which is the direct indicator of color extraction quality.

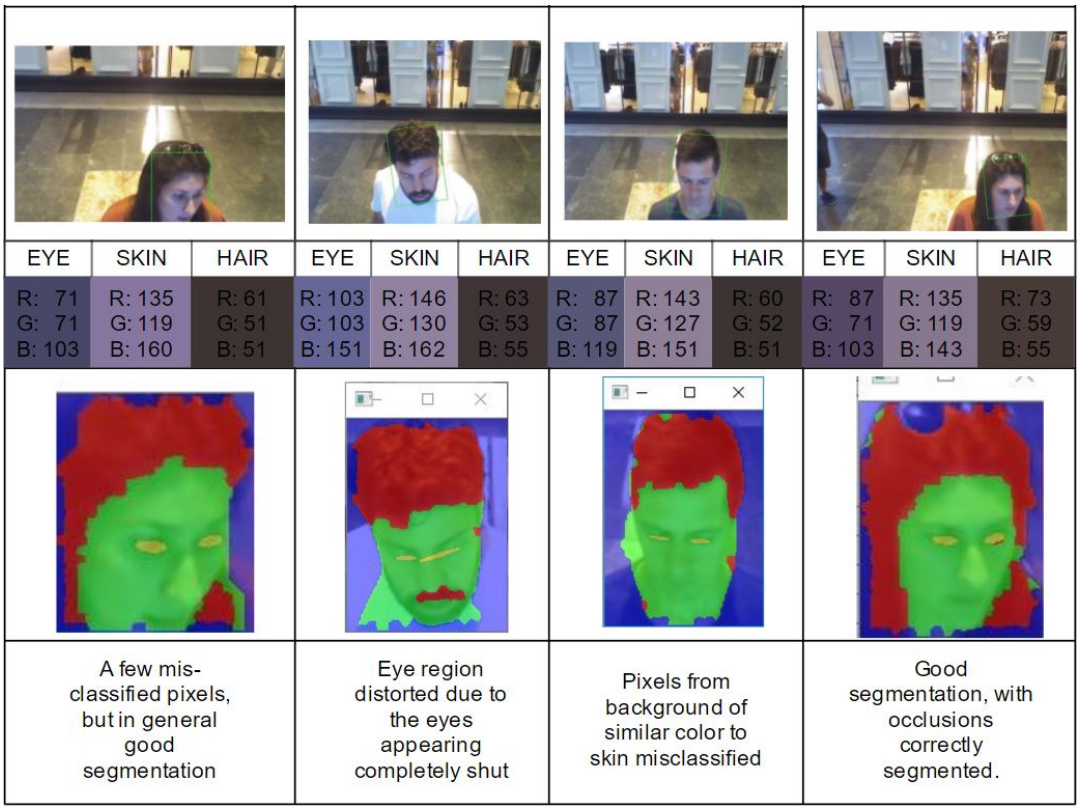

FIGURE 4.7: Samples of segmentation from deployment trial.

Chapter 5

Conclusion

5.1 Problem addressed

This project has been done in collaboration with Imersivo SL, who requested to develop a system capable of extracting color of semantic face components: skin, hair and eyes. The developed algorithm is to be merged into an existing system, which performs fashion recommendations, based on the colors of facial attributes of a subject. Because of the pre-existing system, some hardware limitations applied. In particular, the customer has requested that the system would be capable of running on a CPU with at least 1fps within reasonable accuracy. While the latter requirement was very vague, it was the burden of the author to prove the performance of the system is sufficient for the purpose.

5.2 Proposed solution

The presented paper describes the method used for building a color extraction system for semantic face components: skin hair and eyes. This is achieved by the system pipeline as follows:

1. HOG detector for face detection - firstly all faces in a frame need to be located
2. Facial landmark fitting - this operation serves multiple purposes in the system. It allows to accurately estimate face pose and face size, thus helping to extract more accurate color models in later stages. Also, it removes the need to segment eye-area using optimization techniques, which reduces system complexity and speeds up the process.
3. 1st stage bi-partite graph-cut - background is removed leaving only the face of interest
4. SLIC over-segmentation - by subdividing a face ROI into super-pixels (SP), allows for much more rapid segmentation as $no\_of\_SP \ << \ no\_of\_pixels$. Moreover, oversegmentation captures a lot of important boundaries in the ROI, making the segmentation and subsequent color extraction much more accurate.
5. Gaussian color modeling - the segmentation process is governed by modeling colors belonging to a specific category. Colors from pre-defined regions of each face are extracted, which are converted into color models for each semantic category.
6. 2nd stage bipartite graph-cut - uses graph-cut with α-expansion for dividing the face into hair and skin based on the color models. Although α-expansion

for this application is superfluous, this gives the ability to scale and adapt the approach for other applications.

7. Color extraction - once the pixel-wise assignments for each semantic category are known, accurate representative colors are extracted using histogram analysis.

8. Temporal analysis - if the color extraction is performed from a video stream, temporal analysis is performed as a way of self-checking. This mitigates chances for incorrect final results.

9. (Optional) Color Correction - when in deployment, an optional color correction method could be used to mitigate effects of lighting and camera brightness/contrast.

The system has fulfilled the customer requirement for speed and is capable of running at 5fps on a standard CPU. Moreover, for all three facial components the system has provided 11% mean color error over the validation set. This was shown to be perceptually negligible, thus acceptable for the application in question.

5.3 Problems encountered

Although the proposed system has demonstrated sufficient accuracy for the particular application, the standard deviation (StdDev) of the error is quite alarming. The system performed at mErr = 32 (11%) with StdDev = 26.5. This source of error and high StdDev can be attributed to the following shortcomings of the system:

1. Inability to handle large occlusions, like caps, scarfs, helmets etc. If some accessories cover large part of a particular face component, it is likely it will be classified as being part of it, e.g. helmets being treated as hair. If an occlusion is small, its effect is negligible, since final color extraction is performed through histogram analysis. On the contrary, large occlusions become the dominant color in the histogram, thus providing incorrect results.

2. In the presence of glasses, especially sunglasses, the method fails for eye color extraction. Large amount of population wear glasses and, depending on the deployment environment of the system, sunglasses could also potentially be a very common accessory. This can provide errors due to reflections and color distortions from glasses or due to being completely occluded.

3. Bald-spot on the forehead often misleads the system into thinking the person is completely bald.

4. Strong lighting intensities, can generate light reflections on the skin and change the color tone of hair, making segmentation (and subsequently color extraction) incorrect.

5.4 Adaptability

Although this system was designed to extract colors of semantic face components, it is possible for it to be easily adaptable to other applications. The only component which needs to be changed in the system is the object detector. Once the object in question is detected, the further pipeline is still applicable and even scalable to larger

number of semantic categories, since the second stage of graph cut method is performed using multi-class alpha-expansion. For example, a viable application for this pipeline would be color extraction of currently worn clothes through segmentation of human body.

5.5 Future work

The obvious next step for improving the proposed system is addressing the issues discussed in 5.3 Problems encountered. Since occlusions in general are very ubiquitous in the wild, the system must be smart enough to recognize them. Moreover, improvements are necessary to enable more accurate segmentations of bald-spots and bald people. Since the accuracy of the proposed system relies largely on the quality of segmentation, this poses another issue – all current SOTA methods for segmentation are based on CNNs. Introduction of a CNN would significantly increase the inference time on a CPU, possibly making the system unusable for the application it was designed with current hardware. A possible and already applied approach is to employ facial priors, which enables to build much more compact neural networks, without sacrificing the accuracy as presented by Liu et al [1]. If the size of segmentation CNN is made sufficiently small, even a CPU can handle a feed-forward cycle of 1fps. This would significantly improve the segmentation quality, thus potentially making this color extraction system a SOTA.

Bibliography

[1] Sifei Liu et al. "Multi-objective convolutional learning for face labeling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3451–3459.

[2] Stavros Tsogkas et al. "Deep learning for semantic part segmentation with high-level guidance". In: *arXiv preprint arXiv:1505.02438* (2015).

[3] Karthik Sridharan et al. "A probabilistic approach to semantic face retrieval system". In: *Audio-and video-based biometric person authentication*. Springer. 2005, pp. 85–100.

[4] Khalil Khan, Massimo Mauro, and Riccardo Leonardi. "Multi-class semantic segmentation of faces". In: *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 827–831.

[5] Andrew Kae et al. "Augmenting CRFs with Boltzmann machine shape priors for image labeling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2019–2026.

[6] Xiao Liu et al. "Weakly supervised multiclass video segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 57–64.

[7] Anestis Papazoglou and Vittorio Ferrari. "Fast object segmentation in unconstrained video". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 1777–1784.

[8] Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla. "Label propagation in complex video sequences using semi-supervised learning." In: *BMVC*. Vol. 2257. 2010, pp. 2258–2259.

[9] Gabriel Brostow et al. "Segmentation and recognition using structure from motion point clouds". In: *Computer Vision–ECCV 2008* (2008), pp. 44–57.

[10] Davis E. King. "Dlib-ml: A Machine Learning Toolkit". In: *Journal of Machine Learning Research* 10 (2009), pp. 1755–1758.

[11] 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*. 2014, pp. 1867–1874.

[12] Proyecto Fin De Carrera and Ion Marques. "Face recognition algorithms". In: *Master's thesis in Computer Science, Universidad Euskal Herriko* (2010).

[13] Inseong Kim, Joon Hyung Shim, and Jinkyu Yang. "Face detection". In: *Face Detection Project, EE368, Stanford University* 28 (2003).

[14] 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*. Vol. 1. IEEE. 2001, pp. I–I.

[15] Gary Bradski et al. "The opencv library". In: *Doctor Dobbs Journal* 25.11 (2000), pp. 120–126.

[16] Cha Zhang and Zhengyou Zhang. *A survey of recent advances in face detection*. 2010.

[17] Pedro F Felzenszwalb et al. "Object detection with discriminatively trained part-based models". In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2010), pp. 1627–1645.

[18] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. "Active appearance models". In: *European conference on computer vision*. Springer. 1998, pp. 484–498.

[19] Anil K. Jain, Yu Zhong, and Sridhar Lakshmanan. "Object matching using deformable templates". In: *IEEE Transactions on pattern analysis and machine intelligence* 18.3 (1996), pp. 267–278.

[20] Alan L Yuille. "Deformable templates for face recognition". In: *Journal of Cognitive Neuroscience* 3.1 (1991), pp. 59–70.

[21] James Coughlan et al. "Efficient deformable template detection and localization without user initialization". In: *Computer Vision and Image Understanding* 78.3 (2000), pp. 303–319.

[22] Yali Amit and Alain Trouvé. "Pop: Patchwork of parts models for object recognition". In: *International Journal of Computer Vision* 75.2 (2007), p. 267.

[23] Martin A Fischler and Robert A Elschlager. "The representation and matching of pictorial structures". In: *IEEE Transactions on computers* 100.1 (1973), pp. 67–92.

[24] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. "Robust object detection with interleaved categorization and segmentation". In: *International journal of computer vision* 77.1-3 (2008), pp. 259–289.

[25] 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*. Vol. 1. IEEE. 2005, pp. 886–893.

[26] Santosh Divvala, Alexei Efros, and Martial Hebert. "How important are "deformable parts" in the deformable parts model?" In: *Computer Vision–ECCV 2012. Workshops and Demonstrations*. Springer. 2012, pp. 31–40.

[27] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. "Neural network-based face detection". In: *IEEE Transactions on pattern analysis and machine intelligence* 20.1 (1998), pp. 23–38.

[28] Christophe Garcia and Manolis Delakis. "A neural architecture for fast and robust face detection". In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Vol. 2. IEEE. 2002, pp. 44–47.

[29] Fernando De la Torre and Minh Hoai Nguyen. "Parameterized kernel principal component analysis: Theory and applications to supervised and unsupervised image alignment". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[30] Michael J Jones and Tomaso Poggio. "Multidimensional morphable models". In: *Computer Vision, 1998. Sixth International Conference on*. IEEE. 1998, pp. 683–688.

[31] Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. "Robust and efficient parametric face alignment". In: *Computer Vision (ICCV), 2011 IEEE International Conference On*. IEEE. 2011, pp. 1847–1854.

[32] Xuehan Xiong and Fernando De la Torre. "Supervised descent method and its applications to face alignment". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 532–539.

[33] Zhanpeng Zhang et al. "Facial landmark detection by deep multi-task learning". In: *European Conference on Computer Vision*. Springer. 2014, pp. 94–108.

[34] Amin Jourabloo and Xiaoming Liu. "Large-pose face alignment via CNN-based dense 3D model fitting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4188–4196.

[35] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[36] Reza Oji and Farshad Tajeripour. "Full object boundary detection by applying scale invariant features in a region merging segmentation algorithm". In: *arXiv preprint arXiv:1210.7038* (2012).

[37] Yainuvis Socarrás Salas et al. "Improving hog with image segmentation: Application to human detection". In: *Advanced Concepts for Intelligent Vision Systems*. Springer. 2012, pp. 178–189.

[38] Akira Suga et al. "Object recognition and segmentation using SIFT and Graph Cuts". In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE. 2008, pp. 1–4.

[39] Paolo Piccinini, Andrea Prati, and Rita Cucchiara. "SIFT-based segmentation of multiple instances of low-textured objects". In: *International Journal of Computer Theory and Engineering* 5.1 (2013), p. 41.

[40] Eduard Trulls et al. "Dense segmentation-aware descriptors". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2890–2897.

[41] Patrick Ott and Mark Everingham. "Implicit color segmentation features for pedestrian and object detection". In: *Computer vision, 2009 IEEE 12th international conference on*. IEEE. 2009, pp. 723–730.

[42] Bram Van Ginneken et al. "Active shape model segmentation with optimal features". In: *IEEE transactions on medical imaging* 21.8 (2002), pp. 924–933.

[43] Meijuan Yang et al. "Medical Image Segmentation Using Descriptive Image Features." In: *BMVC*. 2011, pp. 1–11.

[44] Chenxi Zhang, Liang Wang, and Ruigang Yang. "Semantic segmentation of urban scenes using dense depth maps". In: *European Conference on Computer Vision*. Springer. 2010, pp. 708–721.

[45] Julia Diebold et al. "Interactive multi-label segmentation of RGB-D images". In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2015, pp. 294–306.

[46] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. "Perceptual organization and recognition of indoor scenes from RGB-D images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 564–571.

[47] Camille Couprie et al. "Toward real-time indoor semantic segmentation using depth information". In: *Journal of Machine Learning Research* (2014).

[48] Saurabh Gupta et al. "Learning rich features from RGB-D images for object detection and segmentation". In: *European Conference on Computer Vision*. Springer. 2014, pp. 345–360.

[49] Bharath Hariharan et al. "Simultaneous detection and segmentation". In: *European Conference on Computer Vision*. Springer. 2014, pp. 297–312.

[50] Dan Ciresan et al. "Deep neural networks segment neuronal membranes in electron microscopy images". In: *Advances in neural information processing systems*. 2012, pp. 2843–2851.

[51] Yaroslav Ganin and Victor Lempitsky. "Nˆ 4-fields: Neural network nearest neighbor fields for image transforms". In: *Asian Conference on Computer Vision*. Springer. 2014, pp. 536–551.

[52] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440.

[53] Pedro HO Pinheiro and Ronan Collobert. "Recurrent Convolutional Neural Networks for Scene Labeling." In: *ICML*. 2014, pp. 82–90.

[54] Jonathan Warrell and Simon JD Prince. "Labelfaces: Parsing facial features by multiclass labeling with an epitome prior". In: *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE. 2009, pp. 2481–2484.

[55] Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Hierarchical face parsing via deep learning". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 2480–2487.

[56] Brandon M Smith et al. "Exemplar-based face parsing". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 3484–3491.

[57] Carl Scheffler and Jean-Marc Odobez. "Joint adaptive colour modelling and skin, hair and clothing segmentation using coherent probabilistic index maps". In: *British Machine Vision Association-British Machine Vision Conference*. EPFL-CONF-192633. 2011.

[58] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. "Key-segments for video object segmentation". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1995–2002.

[59] Peter Ochs and Thomas Brox. "Higher order motion models and spectral clustering". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 614–621.

[60] Xue Bai et al. "Video snapcut: robust video object cutout using localized classifiers". In: *ACM Transactions on Graphics (ToG)*. Vol. 28. 3. ACM. 2009, p. 70.

[61] Brian L Price, Bryan S Morse, and Scott Cohen. "Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues". In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 779–786.

[62] D Tsai, M Flagg, and J Rehg. "Motion coherent tracking with multi-label mrf optimization, algorithms". In: (2010).

[63] Tianyang Ma and Longin Jan Latecki. "Maximum weight cliques with mutex constraints for video object segmentation". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 670–677.

[64] Dong Zhang, Omar Javed, and Mubarak Shah. "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 628–635.

[65] Chris Stauffer and W Eric L Grimson. "Adaptive background mixture models for real-time tracking". In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* Vol. 2. IEEE. 1999, pp. 246–252.

[66] Mechthild Stoer and Frank Wagner. "A simple min-cut algorithm". In: *Journal of the ACM (JACM)* 44.4 (1997), pp. 585–591.

[67] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38.

[68] Radhakrishna Achanta et al. "SLIC superpixels compared to state-of-the-art superpixel methods". In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2274–2282.

[69] Yuri Boykov, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.11 (2001), pp. 1222–1239.

[70] *Energy Minimization with Graph Cuts*. `http://profs.etsmtl.ca/hlombaert/energy/`. (Accessed on 05/15/2017).

[71] VP Lopez et al. "ChaLearn LAP 2016: first round challenge on first impressions-dataset and results". In: *ChaLearn Looking at People Workshop on Apparent Personality Analysis, ECCV Workshop Proceedings*. 2016.

[72] Lewis R Goldberg. "The structure of phenotypic personality traits." In: *American psychologist* 48.1 (1993), p. 26.

[73] WS Mokrzycki and M Tatol. "Colour difference ΔE–A survey." In: *Machine Graphics & Vision* 20.4 (2011).

[74] Carole Jackson. *Color Me Beautiful: Discover Your Natural Beauty Through the Colors That Make You Look Great and Feel Fabulous*. Ballantine Books, 2011.