TRACING THE COLORS OF CLOTHING IN PAINTINGS WITH IMAGE ANALYSIS

by

Cihan Sarı

B.S., Electrical-Electronics Engineering, Yeditepe University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Systems and Control Engineering

Boğaziçi University

2018

TRACING THE COLORS OF CLOTHING IN PAINTINGS WITH IMAGE ANALYSIS

APPROVED BY:

Assoc. Prof. Albert Ali Salah .....
(Thesis Supervisor)

Assoc. Prof. Alkım Almıla Akdağ.Salah .....
(Thesis Co-supervisor)

Prof. Lale Akarun .....

Prof. Yağmur Denizhan .....

Assist. Prof. Furkan Kıraç .....

DATE OF APPROVAL: 16.03.2018

ACKNOWLEDGEMENTS

Above all, I would like to acknowledge the help and continued support of my principal and assistant advisors Assoc. Prof. Albert Ali Salah and Assoc. Prof. Alkım Almıla Akdağ Salah. I am also extremely grateful to Dr. Ceyhun Burak Akgül for his help all throughout my Master of Science studies and his encouragements and support through my studies.

I would like to give my special thanks to my family, my wife, my parents and my uncle who have been a great help in my academic life.

This thesis would not have been possible without the help, support and love of my friends and family.

ABSTRACT

TRACING THE COLORS OF CLOTHING IN PAINTINGS WITH IMAGE ANALYSIS

In this thesis, we propose an automated methodology to study the trend of color preferences for male and female subjects of paintings over several hundred years. This study is part of a larger project on understanding how sex is defined and described in the Western culture, by tracing the transformation of gender representations in culture, literature, and arts from 17^{th} to 20^{th} century. The data are collected using digital images of paintings from Rijksmuseum Amsterdam. We proposed an approach to extract the dominant color in clothes of the sitters of portrait paintings. The resulting application of this study could provide a useful tool for the digital humanities scholars. Artworks used in this study consists of different artifacts and objects. We ran a face detection algorithm on the Rijksmuseum dataset for the portrait painting collection. Following that, the portraits were classified into their perceived sex by an algorithm, trained on photographic images. Three different face image databases were employed and compared to measure the impact of varied training set conditions on perceived sex classification from paintings. To concentrate on the color information of the sitter, clothing segmentation of the sitter is a necessity. Hence, a simple, yet robust algorithm that uses the location of the face as a landmark to identify a region of interest to represent the clothing in portrait paintings is proposed. This region is used to extract the color distribution, and one dominant color. We contrasted four color extraction methods for this purpose. An interactive interface, where the results of the approach can be viewed and analyzed by an individual is designed. It provides a full overview of the color trends on a temporal axis, thus making it possible to study color preferences in different eras, as well as the changes in color connotation. The interface is designed as a visualization tool for curators or researchers and makes it possible to receive feedback.

ÖZET

TABLOLARDAKİ KİŞİLERİN KIYAFET RENKLERİNİN İMGE ANALİZİ İLE ÇIKARTILMASI

Bu tezde, kadın ve erkek olarak algılanan bireylerin son yüzyıllardaki renk seçimlerini otomatik olarak inceleyen bir sistem önerilmiştir. Bu çalışma, batı sanatındaki cinsiyet tanımını ve algısı, bu tanımın 17. ile 20. yüzyıllar arasında kültür, edebiyat ve sanat dallarında görselleştirilmesi projesinin bir parçası olarak yapılmıştır. Bu çalışmada Rijksmuseum Amsterdam tarafından paylaşılan sayısal imgeler kullanılmış, bu portrelerdeki modellerin kıyafetlerindeki baskın rengin çıkarılması için bir metod önerilmiştir. Bu çalışmanın sonucunda, sayısal sanat bilimiyle ilgili bilim insanlarının kullanabileceği bir uygulama çıkartılmıştır. Çalışmada kullanılan veri kümesi, portrelerin yanısıra, pek çok tarihi eser ve nesneyi de içinde bulunmaktadır. Portre resimlerinin bu kümeden ayrıştırılması için, Rijksmuseum veri kümesine yüz bulma algoritması uygulanmıştır. Arkasından, bu portreler, modelin algılanan cinsiyetine göre ayrıştırılmak üzere, insan fotoğrafları ile eğitilmiş bir sınıflandırıcıdan geçirilmiştir. Bu amaçla, üç farklı yüz veri kümesi kullanılmış ve çeşitli eğitim kümesinin resimlerde cinsiyet algısı üzerindeki etkileri incelenmiştir. Modele ait renk bilgisine yoğunlaşabilmek için, kıyafet bölütlenmesine gerek duyulmuştur. Bu sebepten, yüzdeki simgesel noktaları kullanarak kıyafet için ilgi bölgesi çıkartan basit ama kuvvetli bir algoritma sunulmuştur. Bu bölge, renk dağılımlarını ve hakim rengi çıkarmak için kullanılmıştır. Bu amaçla, dört farklı renk çıkarım yöntemi kıyaslanmıştır. Son olarak, etkileşimli bir arayüz yardımı ile, sonuçların görülebildiği ve incelenebidiği bir platform hazırlanmıştır. Bu platform, kullanıcıları renk eğilimlerini zaman ekseninde görselleyebilmesi ve bu sayede farklı dönemleri renk dağılımlarını ve bu renklerin çağrışımlarını inceleyebilmesini sağlamaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

API	Application programming interface
AJAX	Asynchronous JavaScript And XML
CART	Classification and regression tree
CNN	Convolutional neural network
CSS	Cascading style sheets
DAH	Digital art history
DH	Digital humanities
dpi	Dots per inch
EDM	Europeana Data Model
EM	Expectation-Maximization
FA	Factor analysis
FAMSF	Fine Arts Museums of San Francisco
GPA	Generalized Procrustes Analysis
GMM	Gaussian mixture models
HOG	Histogram of oriented gradients
HSI	Hue Saturation Intensity
HTML	HyperText markup language
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IMDb	Internet movie database
IMoFA	Incremental mixtures of factor analyzers
IT	Information Technology
JPEG	Joint Photographic Experts Group
JSON	JavaScript object notation
LBP	Local binary patterns
LFW	Labeled faces in the wild
MoG	Mixture of Gaussians
PNG	Portable network graphics
RBF	Radial basis function
RDF	Random decision forests

ReLU	Rectification Layer
RGB	Red green and blue
RMSD	Root mean square distance
SC	Scientific computation
SDM	Supervised descent method
SMO	Sequential minimal optimization
SSD	Sum of the squared distances
SVM	Support vector machine
tf-idf	Term frequency–inverse document frequency
VGG	Visual Geometry Group
XML	eXtensible markup language

1. INTRODUCTION

Today, many cultural heritage collections are available online. Digitization and preservation of artworks is a vital occupation of both museums and cultural heritage institutions. Such collections are usually enriched with meticulously tagged metadata about each artwork, including, but not limited to, the origins of the artwork, artist, creation date, among others. Many museums, archives, and libraries have digitized their collections [1].

Application of computer vision algorithms to gain a higher level understanding of digital images and analysis of artworks was an uncommon practice for both art scholars and computer analytic specialists until recently. The assessment and perceptual analysis of paintings were, and still are, mostly performed by human art experts and connoisseurs. Today, the developments in computer vision -especially the re-introduction of deep neural networks- and the increase in computer processing power to process millions of digital images in a reasonable timeframe generated the suitable ground for the maintenance, analysis and information retrieval of digital artwork collections.

Although the skills and perceptions of experts have great value, they are inevitably susceptible to error and subjectivity due to the amount, the subtlety, or the unusual nature of the visual information [2]. There are cases where computers can analyze certain aspects of perspective, lighting, color, the subtleties of the shapes of brush strokes better than even an art expert or connoisseur [3]. Hence, image analysis tools can be used, not as a substitute of the expert, but as a tool to enhance the art historian, curator and conservators' interpretation and analysis. This thesis is a contribution to the analysis of cloth colors in Western paintings.

Color is one of the leading features for carrying different meanings and interpretations in different cultures and ages. For example, the color red is a symbol of luck in China, Denmark, and Argentina, while it has a negative implication in Germany, Nigeria or Chad [4]. Such variations lead to differences in color preferences. Occasionally,

the connotations occur arbitrarily, like in the instance when pink was assigned to baby girls, and blue started to be associated with baby boys at the turn of 19^{th} Century [5]. However, there are times where the color associations have very tangible causes, such as in the case of Marian blue and why over the centuries it was reserved only for the paintings of Virgin Mary. The reason is to be found in the scarcity of the rock lapis lazuli -even more valuable than gold-, from which the blue pigments were extracted [6].

Individual colors have convoluted and contested histories since they have been attached to many symbols at any given time. Our work is inspired by the work of John Gage, an art historian who has devoted 30 years of research on the topic of color. He explains the conundrum of what he terms as "politics of color" in a simple fashion: "The same colors or combinations of colors can, for example, be shown to have held quite antithetical connotations in different periods and cultures, and even at the same time and in the same place." [7].

Although the perception of color is sophisticated and contextual, it is measurable to some extent via image analysis. The human visual system computes color in several stages and achieves independence of spectral variations in illumination, and color constancy [8]. Subsequently, simple pixel-based evaluation of color is a simplification of how colors are perceived in paintings. As a painting is digitized, the sensor properties of the camera, as well as the illumination of the painting will have an influence on the pixel values. Finally, paintings will change colors as they age. When an old painting is removed from its frame, the parts that remain hidden from the damaging light inside the frame stay truer to the original colors, and painting restoration can use this information to correct for colors [9].

Recent developments in the image analysis and computer vision fields have increased their applications in a variety of fields. Body segmentation and gender recognition have been previously applied to photographs. Our study is the first broad investigation of gender recognition and cloth color extraction from paintings.

Portraits served a variety of social and practical functions in Renaissance and Baroque Europe. In Renaissance and European Baroque art history, portraits contained a variety of social and empirical evidence. The conventional aspects of portraiture ensure that each example will bear some resemblance to the next, and yet this general similarity makes the distinctive qualities of each one more noticeable. This general similarity provides possibilities for applying computer-based image analysis to study the peculiar differences within the portraits. These peculiar qualities can emphasize some aspect of the appearance of the sitter, or hint at a personality aspect, suggest the sitter's interest or profession, or his or her social level [10]. The clothing is an important attribute in the portraits Studying the cloth coloring has diverse applications for social scientists and psychologists, historians and curators, and even fashion designers.

The data for this study was compiled from the digital image archive of Rijksmuseum Amsterdam by Mensink and van Gemert, for an image retrieval and annotation challenge [11]. We have used manual annotation to extract the dominant color in clothes of the sitters in portraits for some of the images. However, this is a meticulous task and only performed to assess the accuracy of the proposed automatic approach for the same purpose. One of the aims of this study was to introduce an interactive interface, where the results of the automatic tools can be viewed and analyzed. This interface provides a valuable starting point for developing analysis tools that can be used to advance the understanding of color preference in different time periods, as well as to study the changes of color connotations.

1.1. Contributions

The thesis proposes to employ a combination of algorithmic image processing tools for the study of painting collections. The proposed system finds people in the painting, classifies apparent sex (termed incorrectly as "gender recognition" in the computer science literature), extracts a part of the clothing and models its color distribution for finding the dominant colors. Visualizations are developed to depict the changes in colors over time. Finally, a web interface is developed to help annotations

and analysis[1] .

The earlier results of the thesis were presented in a conference presentation at the Digital Humanities conference [13]. An extended version is submitted for publication in the journal Digital Scholarship in the Humanities, indexed in the Social Science Citation Index and the Arts and Humanities Citation Index [14].

1.2. Organization of the Thesis

The thesis is organized as follows. In Chapter 2 we summarize some of the related work in this area. A number of different algorithms are integrated into the system. Subsequently, most algorithmic related work is relegated to the individual chapters and sections describing these sub-problems. The dataset of digital paintings used in the thesis is described in detail in Chapter 3. While new image collections are made available daily, the Rijksmuseum dataset is valuable in its annotations, and for the number of portrait paintings, it contains.

We have developed a tool we call "Data Supervisor," which is not directly related to the research questions of the thesis, but could be considered as a useful side product of this study. This tool is inspired by the research of Crowley and Zisserman [15] which uses the power of the convolution networks to learn several concepts on the fly and performs image retrieval from the training dataset. The concepts are entered as keywords, and an image search is conducted over the Internet. Image search responses are used to train a classifier to rank the database. Details on each step are explained in Chapter 4.

We describe the face image processing for separating paintings into male and female in Chapter 5. The automatic process includes face detection, as well as classification of the face into one of two classes. We term this problem perceived sex classification (as opposed to gender recognition), as the word "gender" is mostly used to distinguish the social and cultural aspects of differences between men and women,

[1]The interface can be accessed for a limited time at: [12]

whereas “sex” denotes biological differences [16].

In Chapter 6 we introduce a method to measure the clothing color distribution of these paintings. Clothing segmentation algorithms are performed using the location of the face as a landmark, to associate the relevant color information to the sitter. Color extraction methods are used to extract the palette and the dominant color from the clothing. The results are analyzed and displayed in a temporal trend plot using the extracted dominant color.

Finally, we summarize our research in Chapter 7, give links to the developed tools and discuss future work. A visual overview of the flow of the thesis is presented in Figure 1.1.

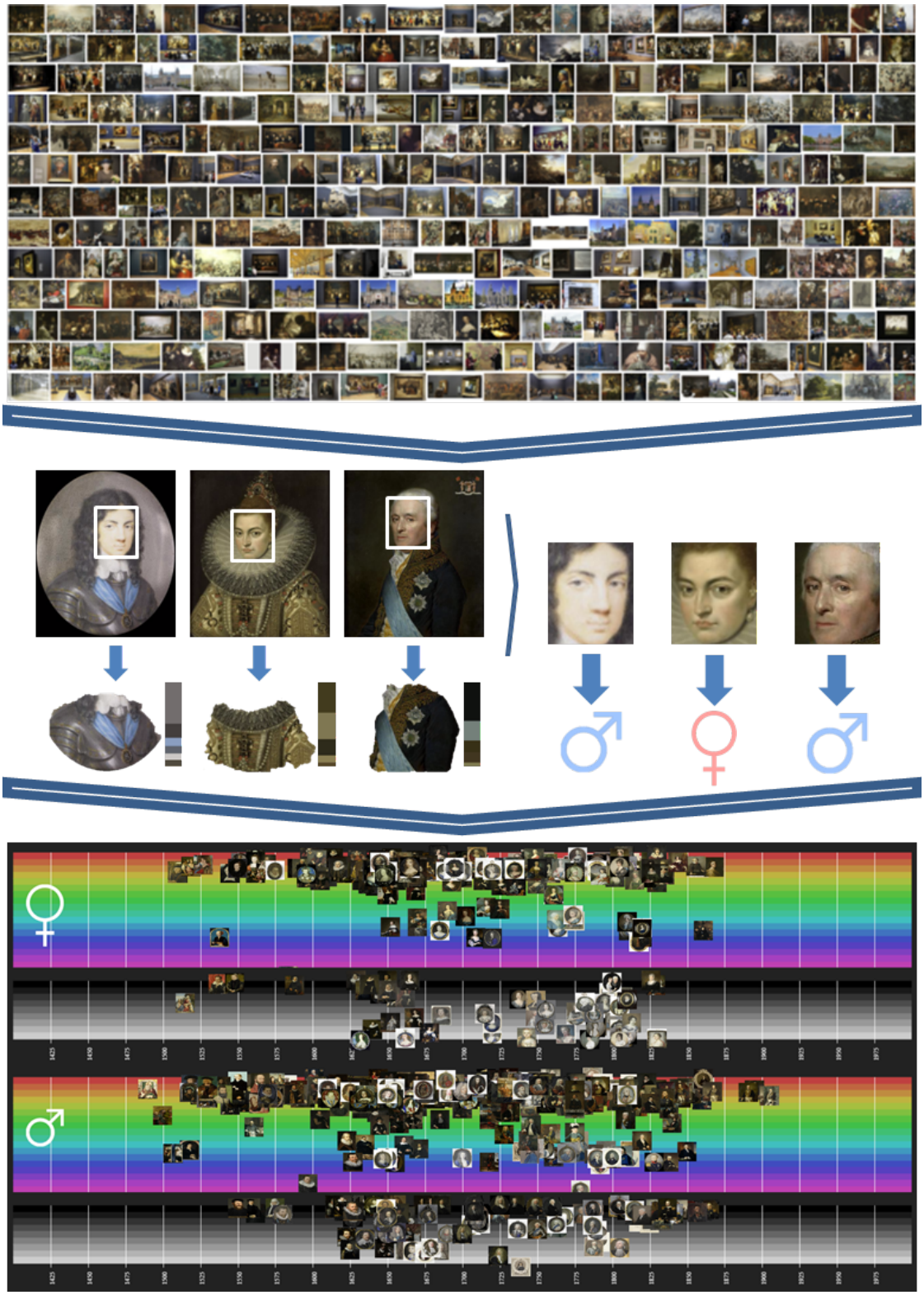


Figure 1.1. A visual illustration of the workflow of the thesis.

2. RELATED WORK

In recent years, there has been an increasing amount of literature on using computer vision methods to address art historical studies. Stork et al. give an overview of some recent computer techniques that have been applied with the purpose of answering art related questions [17]. The focus of their study is on the application of the image analysis algorithms to understanding and processing digital copies of artworks, in particular paintings and drawings. They use digitized media to present the scale and the possibilities in computer sciences to the art scholars.

Barni *et al.* have prepared a similar survey to assess the application of image processing algorithms in the field of arts [3]. They have separated the use of image analysis methods into three categories: Digital acquisition of art paintings, image diagnostics, and the implementation of virtual restoration. In their work, the digital acquisition is defined as reproducing the artwork in the digital form as close as possible to the original. This area involves activities such as archiving, retrieval and dissemination, where the digital format has significant advantages.

A large and growing number of museums, art archives, and libraries are engaged in digitizing their cultural heritage [18]. Primary research questions are related to lighting and camera conditions to minimize the artifacts of digitization and to maximize the quality of the images. The Rijksmuseum in Amsterdam is one of the grand European museums, which is home to many of Dutch masterpieces. Mensink et al. [11] offered a challenge for visual classification and content-based exploration on artifacts digital images based on the digitized portion of the Rijksmuseum collection. In their work, they provide a dataset of 112,039 photographic reproductions of the artworks exhibited in the Rijksmuseum, which is the primary dataset we used for this study.

Virtual restoration is another area that uses image processing techniques to enhance the artworks. The present form of the artwork might have deteriorated or altered due to time or unfortunate circumstances. Restoration of the physical art form is pos-

sible. However, it is probably questionable whether such a treatment could (or even if it should) bring the artwork back to its original appearance. Thanks to the digital representation of artworks, such remedies can be applied digitally without any other permanent changes to the original physical form. Virtual cleaning could help the conservators by showing the expected results of several possible restoration processes. Color restoration, crack removal or lacuna filling are some of the applications, which fall into this category [19].

Image analysis is also very useful for extracting additional information from a painting, without the necessity of physical interaction. For example, spectral data about an artwork can be obtained and used for material detection, authenticity control, and classification tasks [20, 21, 22, 23]. Johnson et al. analyzed the brushstrokes of the paintings of Vincent van Gogh [24]. Their work addressed the question raised by the experts whether some of the paintings belong to the famous artist. Shamir used low-level content descriptors from the digitized artworks to measure the similarities between artistic styles of 11 painters [25]. His results have shown artistic style similarities of the low-level image features between the works of Vincent van Gogh and Jackson Pollock.

Image processing and computer vision can tell a lot about style, but also about the content of paintings. Crowley et al. made use of accessibility of fast learning systems and the Internet and introduced a system where a user can retrieve paintings displaying a variety of visual concepts, such as a bridge or storm, on demand [15]. Their approach relies on pre-computation of a large set of negative samples, and on the fly collection of a small set of positive samples, from which a classifier is quickly created to process the database at hand. We use this concept in the Data Supervisor application, in Chapter 4.

For visual analysis of a large number of artworks, the "cultural analytics" approach proposed by Manovich is very useful [26]. In this approach, each artwork is represented by a thumbnail, on a space spanned by dimensions computed from the artworks through simple and intuitive functions, such as time of creation, mean hue, mean saturation, mean intensity, etc. Manovich displayed a million artworks on a

single visualization to look for patterns in a related study [27]. His work shows that trends can be seen by intelligent grouping and thumbnail displays over a large number of images. This approach has been extended to large collections of user-generated artworks produced in social networks for arts [28].

3. DATA CONSIDERATIONS FOR ANALYZING PORTRAIT PAINTINGS

3.1. Digitized Artwork Datasets

Today, major cultural heritage institution and museums are going through the process of imaging and annotating their objects and also give access to their digital data through access despite the fact that digitizing the artwork collections is tedious, time-consuming and costly [1]. The reason can be found in the fact that such databases, not only help in the preservation of artwork and give the possibility to users to explore the art collections, it can be integrated with similar collections and be used as a valuable source to researchers and scholars for further investigation of artworks using digital methods [29].

The Metropolitan Museum of Art from New York City, USA is one example of such museums. Its collection includes more than two million artifacts from all over the world and five thousand years of world culture to present. Their online collection [30] contains more than 375,000 images of artworks which is open access to the public to use, share, and study. The digitization and curating of this museum is an ongoing work, and the number of available digital images is increasing each year.

The Fine Arts Museums of San Francisco (FAMSF) is another example. Museums have digitized the majority of their archive in the image base "The Thinker" [31]. It contains over 80,000 images of prints, drawings, paintings, textiles and 3-dimensional artwork.

Some of the artworks of the Museum of Modern and Contemporary Art of Trento and Rovereto in Italy are digitized with the MART dataset [32]. The artworks span between 1913 and 2008 and are of Italian, European and American artists. Sartori et al. use the abstract arts available in the MART dataset to analyze the emotions evoked by the artworks [33].

Rijksmuseum Amsterdam is an interesting case, unique in the sense that it was involved in a multimedia retrieval challenge to evoke the interest of computer vision researchers [11]. Subsequently, their data are available in a very research-friendly format. Further details in digitized works of art and associated details are given in Section 3.2.

Some of the smaller datasets from a single art movement or even a single painter also exist. For example, Ginosar et al. use Picasso dataset to detect people in cubist art [34]. The dataset contains 218 of his paintings and the titles.

In addition to the digitized artworks, there is also a large number of digital art datasets available. DeviantArt [35] contains an enormous amount of digital art with hundreds of thousands pieces uploaded every day, albeit not in a very researcher friendly fashion due to lack of polished categories and annotation. Bogazici University's BODAIR (Bogazici-DeviantArt Image Reuse) dataset is a collaboration work with DeviantArt and contains 200 images for each of its six subcategories [36]: animals, food, nature, places, plants and premade backgrounds.

3.2. Rijksmuseum Dataset

Rijksmuseum from Amsterdam is a national museum and one of the finest art museums in the world. It is home to more than a million art objects, including masterpieces of Van Gogh, Vermeer, and Rembrandt.

619,482 digitized works of art are available at the time of this study [37]. Throughout the digitization process, annotations to the records are added when possible from structured vocabularies [38]. For the majority of the paintings, some simple information is available. Associated information that is provided with the dataset is called metadata, and it is highly useful, not only for ordinary filtering purposes but also for extracting or computing higher level information.

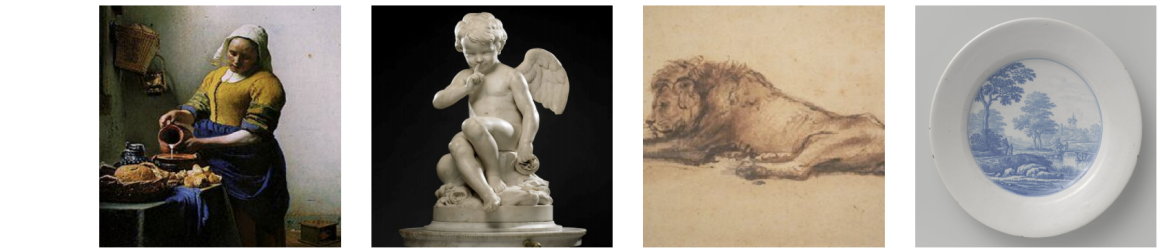
Figure 3.1. Visualization of Rijksmuseum dataset using t-SNE. Figure from [11].

The Rijksmuseum digital data are available through a non-public API access to the collection objects. This server provides data in the format of the Europeana Data Model (EDM) [39]. Some example images and associated metadata can be seen in Figure 3.2.

Mensink et al. [11] introduced a sizable open dataset of art objects from Rijksmuseum to support and evaluate computer-aided art classification and retrieval techniques. The collection is a set of over 110,000 objects consisting of digital images and their metadata descriptions from the Rijksmuseum collection, made public online [37]. The works of art in this date set include paintings from ancient times, medieval ages and the late 19^{th} Century, depicting richness and diversity of Dutch and international cultural heritage. This dataset is used as the primary data in this thesis.

3.3. Dataset Challenges

The Rijksmuseum dataset contains 112,039 digital images. A large portion of the digitized artwork consists of paintings and prints, created by great artists such



Title	The Milkmaid	Seated Cupid	Reclining Lion	Dish with a Landscape
Creator	Johannes Vermeer	Étienne-Maurice Falconet	Rembrandt van Rijn	Frederik van Frytom
Year	1660	1757	1658–1662	1670 – 1685
Type	Painting	Sculpture	Drawing	Saucer
Material	Canvas, Oil paint	White Marble, Copper	Paper, Paint	Faience

Figure 3.2. Example images from the Rijksmuseum dataset and their metadata. Figure from [11].

as Rembrandt, as well as by anonymous painters. Additionally, there are miniatures, nineteenth-century photographs, ceramic and furniture, silverware, among others.

Digital images are visualized in Figure 3.1 using t-SNE. t-SNE is a state of the art unsupervised embedding based on pairwise similarities of data points [40]. It provides a general representation and grouping of the Rijksmuseum material.

The images are saved at 300 dpi quality, with file sizes ranging between 2 to 5 megabytes in JPEG format, with a corresponding XML file that contains the available metadata. Information in the metadata can include title, dates, painter information, if known, in Flemish.

While studying the trends of clothing colors, we faced two challenges during the use of this dataset and its associated metadata. The first challenge was in selecting portrait paintings from a variety of object images. To overcome this obstacle, we used a face detection approach, explained in Section 5.2, where the portraits can be separated from objects for further processing.

Second, as mentioned previously, the annotation of paintings had little information on the elements or the "sitter" of the artworks. Subsequently, we have adopted an automatic approach to determine whether the sitter of a portrait is perceived as female or male. Our approach is explained in Chapter 5.

4. DATA SUPERVISOR

The Data Supervisor is a tool that we have developed to rank a digital collection with respect to concepts of interests. It is a software framework based on MatConvNet [41] and Qt. It includes the VGG deep neural network model, pretrained on the ImageNet [42] database for classifying objects contained in images [43]. Its purpose is to sort any given raw database of images into categories specified by the user. Our purpose in developing this tool with regard to this case study was to identify personal ornaments and embellishments e.g. pieces of jewelry, hair and wig styles, armor, etc. within the portrait paintings of the dataset for further investigation on their relation with perceived sex.

The tool is inspired largely by the work of Crowley and Zisserman [15]. In their research, they provide a tool for users to retrieve a single term, from an image database, and uses a query set to learn this query. In the case of Data Supervisor, we get a number of queries and learn these concept definitions from each other. The algorithmic flow is given below.

(i) Get a series of concepts from the user in form of a text query.
(ii) Generate positive visual examples from the requested concepts.
(iii) Learn visual differences between samples of individual concepts.
(iv) Use this knowledge to rank items in an arbitrary dataset according to the presence of the concepts.

Step 2 is explained in Section 4.1, step 3 in Section 4.2 and finally, classification steps and examples are given in Section 4.3.

4.1. Data Collection

Given a concept (or a set of concepts), the Data Supervisor first uses the Google search engine [44] to retrieve positive image samples. The steps are as follows.

(a) Arm (b) Bag (c) Cup

(d) Hair (e) Jewelry (f) Painting

(g) Pants (h) Plate (i) Portrait

Figure 4.1. The supervisor query results on the Rijksmuseum Dataset.

(i) All given concepts are queried in Google Image search, by invoking an HTTP get request similar to a user's browser. Request contents are generated through observing how similar queries are performed on a browser.

(ii) The response of the query is parsed using regular expressions to filter the non-image contents and hence to extract the image links.

(iii) An HTTP get request is performed on the servers where images are located, using the image uniform resource locators (URLs).

(iv) All pending HTTP get requests are terminated, once a timeout period of ten seconds elapses.

In general, there are around 80 to 100 image URLs returned per query and approximately 50 to 60 images could be received in a timely fashion. In other words, 50 to 60 images for each concept is collected through the Internet in the ten seconds following these steps.

4.2. Learning and Classification

The data collection step collects a moderate amount of images per category. However, this amount is not enough to learn a concept from scratch, especially in a short amount of time. Instead, Data Supervisor uses a pretrained CNN [43] as a feature extraction method.

Chatfield et al. [43] published their VGG model pretrained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [42] database. ImageNet database consists of over 14 million images among 1,000 categories for object localization.

The pretrained network, given an image, generates a rich representation that allows classification of the contained objects [45]. The method involves omitting last few layers of the model that perform classification into ImageNet classes. The output of these specialized layers is directly used to represent the input image. The hypothesis is that this knowledge is sufficiently rich to contain some object representations and visual cues, as well as some translation invariance to detect the presence of objects across the

image. With this logic, every image instance can be projected to this feature space as a point, where if the hypothesis holds, similar concepts should create distinct clusters. Following this idea, a classifier could learn the concept representations in this space with a sufficient number of sample data points.

The Data Supervisor follows this methodology to generate a set of data points per concept, using MatConvNet [41]. A multi-class linear SVM classifier is trained with the representations from each concept (i.e. each concept is used as an individual class). Therefore, the classifier model [46] specializes in distinguishing these concepts' visual representations.

4.3. Data Supervisor Classification

The data classification step uses the outputs from Section 4.2. The data representation space is used to project the instances of the dataset and the concept model is used to measure the likelihood of an instance to belong to any given concept.

The conceptual representation space is not affected by concept representations, and therefore, it is possible to pre-project the painting samples and represent every data instance in this space. Thus, the dataset to be searched for the new concepts is already projected to this space at the time of querying. One could immediately use the visual description models of the query concepts on the projected dataset to rank its points and to retrieve the instances that represent each concept with little additional effort.

Even though we achieved some promising result in case of individual objects, this method did not show to be practically suitable to find the embellishments within the portraits. A group of identified objects within the Rijksmuseum dataset is depicted in 4.1. In this figure, the best five retrieval results are given per query. The top result is shown in the bigger rectangle, and 2^{nd} to 5^{th} results are smaller circles to the right of this rectangle. For each query, a Google image search is performed and results are passed through the network. Network outputs for each category are used as features

for the linear SVM classification to train a model that learns how to distinguish queried terms from each other. Finally, previously extracted Rijksmuseum network outputs are ranked with this classifier to retrieve the most likely images that would represent the concept in question.

It is apparent from this figure that very few objects - almost none - were identified within paintings, in particular in portraits. The observed discrepancy could be attributed to the search result of each object keyword. Such keywords give a specific result of the searched object and it does not provide the object within the context of environment or paintings. This consequently trains the classifier to distinguish the objects where the shape of the object stands alone within the image. Such attributes are obvious when “Bag” or “Cup” are used as the keyword. It is possible to overcome this drawback by careful selection of classes, however, it is against the automated nature of our overall approach. Therefore, we did not pursue this path for acquiring objects within the paintings within the scope of this thesis.

5. PERCEIVED SEX RECOGNITION ON PORTRAIT PAINTINGS

In order to separate the portrait within the Rijksmuseum dataset from other artifacts, we used a face detection algorithm introduced by Viola and Jones in 2001 [47]. This method is a Haar feature-based cascade classifier and uses a series of very fast comparisons to enable rapid and robust face detection.

Once the portraits are determined, and the faces of the sitters are localized, we use a face-based approach to classify the sex of the sitter. However, in order to improve the processing of the faces, we first align them with a general model. It is known that good alignment, also called registration, is a key for robust facial processing [48].

Registration is the determination of a geometrical transformation that aligns points in one picture with the corresponding points in another picture. The anchor points used in facial registration are called "landmarks". These are typically points like mouth and eye corners, the tip of the nose, and points regularly spaced along the boundary of the face. Once such landmarks are located in an image, the image can be transformed, rigidly or non-rigidly, to a target shape, represented as a set of landmark points. Given a set of shapes, it is possible to generate a mean shape that will serve as alignment target automatically, via Generalized Procrustes Analysis (GPA) [49]. Furthermore, if it is used on the training set to extract this mean shape, GPA also produces the alignment of each training sample. Further samples can similarly be aligned to the mean shape. After this step, we can classify portraits into male and female classes.

This process is commonly called "gender recognition" in computer vision and pattern recognition literature [50]. However, what is meant by this term seldom has anything to do with what "gender" means today. In the context of this thesis, we use perceived sex recognition to describe more accurately what is being done, namely, to determine the apparent sex of the person from the facial image.

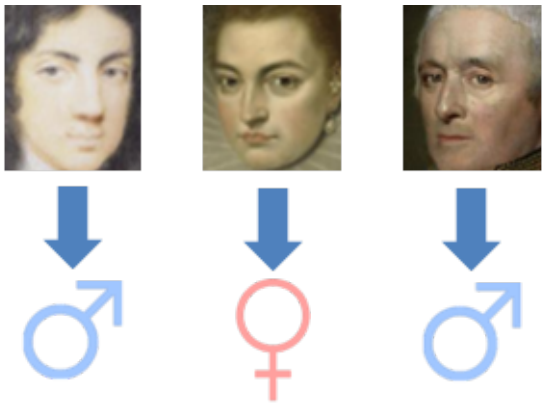


Figure 5.1. Face rectangles cropped from paintings that are used for perceived sex recognition.

Wendy et al. [51] show that human reactions are influenced by perceived sex. This information is acquired prior to the initiation of interaction, seamlessly. Studies show that a vast variety of cues can be used to determine perceived sex; such as gait [52], body shape [53], and smell [54] among many others. Face itself is one such cue that is widely used for perceived sex recognition. The popularity of using face images can be evidenced by the sheer number of studies focused on such images for perceived sex recognition [55, 56, 57, 58, 59, 60].

For perceived sex recognition, we use two types of facial features, namely, local binary patterns (LBP) [61], which are commonly used to describe the face [58, 62, 63, 64, 65], and a deep learning based face descriptor [60]. A portion of the database was annotated manually to measure the accuracy of the proposed perceived sex recognition system. Examples are given in Figure 5.1.

There are several publications from recent years with the aim of automatic perceived sex recognition. The survey by Ng et al. described a variety of approaches to the perceived sex recognition in natural images [66]. This survey shows that the majority of these studies are focusing on either single domain images and publish cross-validation result, or split their dataset into training and test partitions, where these two sets retain links such as similar poses, and lighting conditions among others. Using samples from a single dataset limits the generalization power of such systems. The study of Kayim et al. seeks to eliminate this limitation by using two completely different

training and test datasets [58]. In this thesis, we use three facial image databases to make the approach as robust as possible.

In the following sections, we explain the steps taken for a robust perceived sex prediction methodology. Training and test data collection and annotation are given in Section 5.1. Face detection, facial landmark extraction, and alignment procedures are explained in Section 5.2, Section 5.3 and Section 5.4, respectively. Feature extraction and classification methods are discussed in Section 5.5 and Section 5.6. At the end of this chapter, we discuss style transfer networks and analyze their effects for perceived sex recognition in paintings.

5.1. Datasets

In order to learn the differences between different perceived sex categories, a test dataset of face images from Rijksmuseum paintings and three independent training datasets of face images were used. These are 10k US Adult Faces [67], Labeled Faces in the Wild (LFW) [68] and IMDb datasets, explained in Section 5.1.1, respectively.

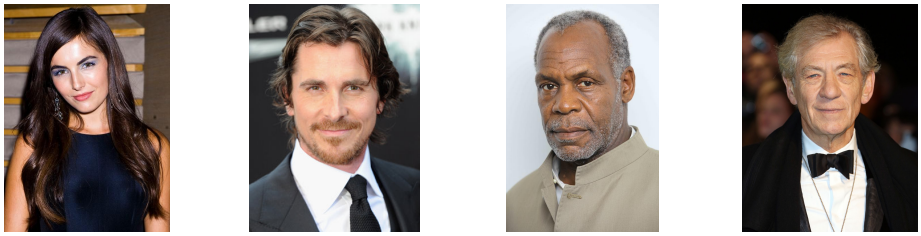
Although the IMDb dataset inherently has male/female annotations, this is not the case for the other datasets. Since wrong annotations can impair system performance significantly, all four datasets are manually cleaned and annotated to ensure a robust training.

5.1.1. IMDb dataset

The IMDb dataset is collected in an approach similar to Jia's work [59]. By using several user-generated IMDb actor and actress lists, such as "100 best actresses" and "50 most talented actors", names of actors and actresses are collected. These lists of actor and actress queries are given in Section A.1 and Section A.2, respectively.

Google image search results for these actors and actresses have a significant bias on posed pictures. As can be seen in Figure 5.2, most of the results are from photo

(a) Adrianne Palicki (b) Adrien Brody (c) Anthony Quinn (d) Bruce Willis



(e) Camilla Belle (f) Christian Bale (g) Danny Glover (h) Ian McKellen

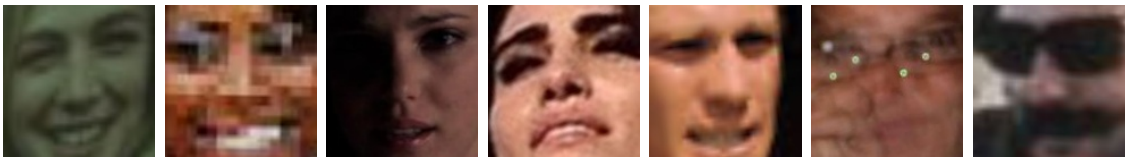
(i) Ivana Baquero (j) Jocelin Donahue (k) Marion Cotillard (l) Natalie Portman

Figure 5.2. Examples from the IMDb dataset.

shoots or intentional poses.

The retrieval of the images was performed with the software explained in Section 4.1. Through this software, a total 5,603 actor and 5,262 actress images are downloaded, and these photos make up the IMDb dataset. This database should not be confused with other similarly named databases in the literature. The actor and actress names provide an automatic annotation for the IMDb set. However, the query images had some noise, and the results had to be verified manually.

IMDb dataset had 5,603 actor images. We have performed the face detection and alignment methods that will be explained in Section 5.2 and Section 5.4, respectively, to these images. This ensured that only the relevant data got annotated. 833 images contained no detectable face. The remaining 4,770 were cleaned and verified by hand. In this set, there were 4,447 faces perceived as male from actor images, and 10 faces



(a) Face images labeled as low-resolution

(b) Images labeled as false face detection

Figure 5.3. Examples of unused images from the IMDb dataset.

regarded as females. There were 217 images where no valid face could be seen - falsely detected by our face detection algorithm, and 96 images with low-resolution faces.

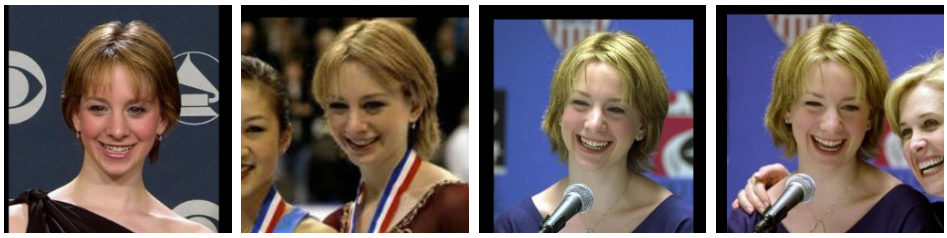
Similar steps are taken for actress images; 4,499 faces made out of 5,262 images that are downloaded from the actress query list. 4,061 of these results belonged to faces perceived as female, 25 of them as male. There were 98 low-resolution faces and 315 images without any face.

Low-resolution faces and false face detection results are not used for further processing and omitted. Some examples of these are given in Figure 5.3. Eventually, face detection and alignment steps filtered out 14.6% of the downloaded images, and hand annotation removed a total of 761 images (no face, low-resolution or wrong face annotations).

5.1.2. Labeled Faces in the Wild

The Labeled Faces in the Wild (LFW) dataset is one of the commonly used datasets for face recognition [68]. It contains 13,233 images from 5,749 unique individuals. True to its name, individual images do not contain deliberately posing people in general, and they appear in the wild as seen in Figure 5.4. The metadata of LFW contains the names of the people in images, which is useful for perceived sex annotation. There are several studies which have used this dataset for this purpose [57, 58].

(a) Aaron Peirsol



(b) Sarah Hughes

Figure 5.4. Example images from LFW dataset. There are occasionally some images with deliberately posing people.

Perceived sex annotation process for ground truth is performed in several iterations over the data. These steps are given below.

(i) A crude annotation is generated using only the names on `genderize.io` [69] application programming interface (API).
(ii) Likely annotated images are grouped, outliers are spotted and moved to their respective category.
(iii) Step 2 is repeated until there are no changes in any category.

There were 13,233 images provided for LFW, from which 572 have been filtered out through face detection and alignment steps. From the remaining 12,661, 3,162 are annotated as feminine names. 2,734 of these faces are perceived as female, 24 with low-resolution and 20 as false hits. The last 384 face images were perceived as males and moved to that category. From the 9,499 images tagged with the masculine name set, 9,295 were perceived as male and 62 as female. 106 of the images were false detection results and, the last 33 had low-resolution. This leaves a set of 9,679 images belonging to the people that are perceived as male and 2,796 that are perceived as female.

5.1.3. 10k US Adult Faces

Bainbridge *et al.* have published the 10k US Adult Faces (will be called 10k from now on) in 2013 [67]. It consists of 10,168 natural face photographs and contains metadata for the sub-sample of the faces [70]. 2,222 of the faces include memorability scores, computer vision and psychological attributes and landmark point annotations. Face photographs are JPEGs with 72 dpi resolution and 256-pixel height.

10k dataset facial landmarks are unfortunately not compatible with our approach, and therefore, face detection and the alignment processes are performed for these images as well. Moreover, re-processing these images ensures that all the inputs of the classifier are processed with the same face detection and alignment steps.

On the remaining 8,875 face images, steps mentioned in Section 5.1.2 are performed. Genderize.io estimation worked very poorly on the file names; there seems to be a very low correlation between the names contained in the individual files, and sex classification performed close to pure chance. In the end, perceived sex for each face image is labeled manually. 5,149 of the images have been assigned to the male class, and 3,726 to female class.

5.1.4. Rijksmuseum

As described in Section 3.2, the Rijksmuseum database consists of 112,039 digital images. We have performed similar steps mentioned earlier on Rijksmuseum to find the individual faces.

The Viola and Jones face detection algorithm returned 162,936 positive hits, with approximately one true positive for fifty false positives. This indicates approximately 2% accuracy. A modern variant of the algorithm uses two parameters to eliminate false positive hits, namely, the scale factor, and minimum neighbors, respectively.

Table 5.1. Face detection results on the Rijksmuseum dataset. Performances are measured from the remaining number of images in the manual cleaned set.

Image scale	**Minimum neighbors**	**Filtered by size**	**Number of positive hits**	**Approximate performance**
1.1	0	no	162,934	0.9%
1.2	1	no	85,417	1.8%
1.2	2	yes	8,611	17.5%

The first parameter controls the scale factor for the image pyramid construction. Lower values result in a taller image pyramid and increase the number of false positive hits. In contrast, high values may result in false negatives, i.e., missing face rectangles.

The algorithm checks every rectangle in the current image of the pyramid against the trained model. The "minimum neighbors" parameter specifies the minimum number of neighboring rectangle candidates that should return a positive hit to retain a certain rectangle. This means that a face is only accepted if the algorithm detects it in several overlapping rectangles. Finally, false positives can be filtered out using the size of the face rectangle. This filter is used as a last step.

Approximate performances with different parameter sets are given in Table 5.1.4. Increasing the number of minimum neighbors required drastically reduces the number of positive hits. With approximately 5 seconds per image to view, to evaluate, to enter results and to move to the next image, manual cleaning with eight hours a day would take for these three different settings, 28 days, 15 days, and 1.5 days, respectively. Therefore, manual cleaning for performance measures is only carried out for the last parameter set.

Face detection and alignment using two minimum neighbors, 1.2 image scale and with filtering out the small face candidates results in 8,611 positive hits. These hits are manually cleaned in the first pass. On this pass, the majority of the text scripts, porcelain sculptures and drawings are removed, and the candidate list size is reduced to

(a) *Kop van een vrouw* from Johannes Petrus van Horstok perceived as male, but actually the poser is a female.

(b) *Portret van Eduard VI (1537-53), koning van Engeland* perceived as female, but in reality the poser is male.

Figure 5.5. Examples of paintings where perceived sex annotations are corrected using the painting titles.

2,462. From these images, each face is assigned into one of the four categorized given as: i- Perceived as female, ii- Perceived as male, iii- False face detection, iv- Low quality or irrelevant. Example paintings of these categories are given in Figure 5.6. Perceived sex of the sitter is a manual annotation, and does not necessarily reflect the correct sex. To improve the annotations, 18 male and 14 female Flemish sex identifying nouns are used to check the titles of the paintings. Words associated with male sitters are as follows: echtgenoot, prins, heer, staatsman, jongeman, keizer, koopman, generaal, vrijheer, kardinaal, koning, hertog, vorst, graaf, officier, kapitein, zoon, vader. The ones associated with female sitters are the following: vrouw, echtgenote, prinses, moeder, grootmoeder, dochter, meisje, hertogin, markiezin, weduwe, actrice, dame, barones, koningin. Perceived sex information of 20 paintings out of the total 1,505 were corrected using the painting titles and mentioned words. King of England Eduard VI (1537-53) was mistakenly perceived as a female, and similarly in Johannes Petrus van Horstok's painting the poser was perceived as male. Thanks to the painting titles these mistakes are amended as seen in Figure 5.5.

The test set is formed using the perceived sex as female and male categories. This set is never used in processing or parameter optimization until the very end, and all

the hyperparameter optimization is carried out on the splits of the training set. The test set is used to measure the end performance of domains, features, and classifiers only once.

5.2. Face detection

Looking for an object with arbitrary size and orientation, everywhere in an image is a very daunting challenge. The Viola and Jones in 2001 published a method based on Haar feature-based cascade classifiers [47]. It addresses this big challenge by using a series of very fast to perform small classifiers to enable a fast face detection evaluation.

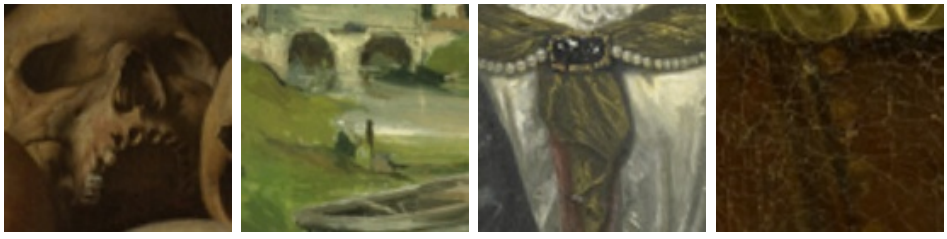
Viola and Jones introduced "integral images". They contain the total intensity values on original image from all the pixels, located in the up-left region of the current pixel. This prepared look-up table enables a constant-time computation for the total pixel value within any rectangle inside the image. Haar-like features given in Figure 5.7 use this incredibly fast calculation speed to decide which candidates can form a face quickly. Different rectangles are iteratively selected by the adaptive boosting algorithm. The first and second features selected in this way is given in Figure 5.7.

Face detection steps check every image for the face candidates. Minimum face size is given to reduce the search space. After each search within the current image, the integral image size is reduced by the scale factor. In this thesis, any face candidate that is smaller than 64×64 pixels is ignored. At each iteration, the integral image size is reduced by a factor of 1.2. This ensures a fast search. However, it is impossible to check every possible scale due to pixel quantization levels. Moreover, rectangles used in the Viola and Jones algorithm are robust, and their responses of adjacent pixels are mostly the same. This ensures each face is detected multiple times and this metric is also used as another method to filter false detection results, by defining a minimum number of triggers in the neighborhood. In the case of multiple detections, only the highest likelihood is considered. In conclusion, the Viola and Jones algorithm works in a robust fashion to find the faces. However, the localization is not its strong suit. Therefore, we need to perform additional steps to align the training and testing images,

(a) Paintings perceived as female

(b) Paintings perceived as male

(c) Low quality paintings



(d) False face detection results, paintings are cropped to detection rectangle

Figure 5.6. Rijksmuseum annotation and cleaning process

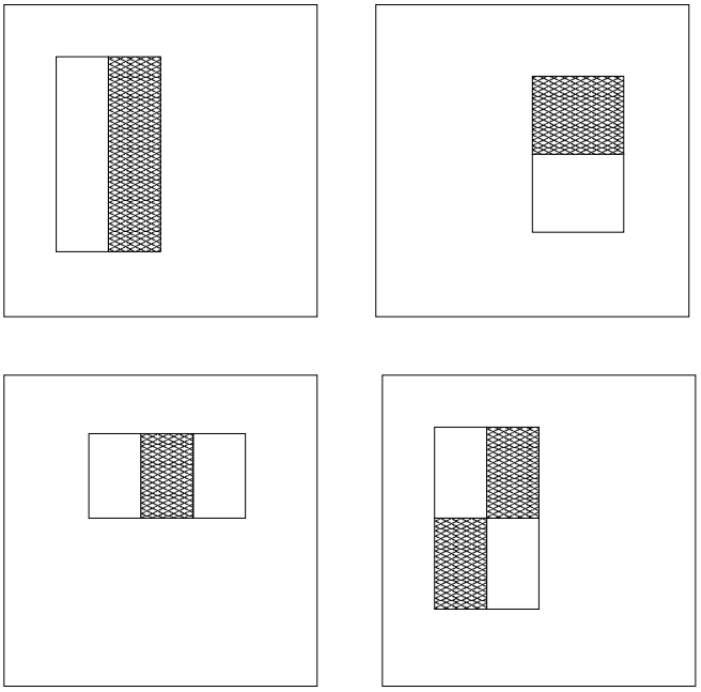


Figure 5.7. Haar-like features that are used for object detection by subtracting the sum of pixels which lie within the white rectangles from the sum of pixels in the gray rectangles. Figure from [47].

to minimize differences caused by the scale and detection location. This alignment is based on the facial landmarks extracted in Section 5.3.

5.3. Face landmarks

The facial landmarks such as nose, eyes, and mouth is a commonly used and more robust approach to align the faces in the wild. The "IntraFace" library, provided by Human Sensing Laboratory from Carnegie Mellon University and Affect Analysis Group of the University of Pittsburgh, based on their joint study [71] excels at addressing this very problem. Xiong et al. propose an approach called Supervised Descent Method (SDM). This approach uses initial landmark points approximated from the face rectangle and then iteratively moves them to their actual locations. The landmark points used with SDM is shown in Figure 5.8. These are the same landmarks that are learned by the algorithm and found on the face rectangles detected by the Viola and Jones algorithm mentioned in Section 5.2.

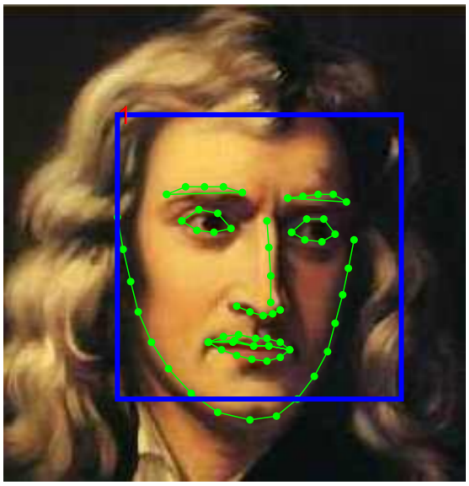


Figure 5.8. The facial landmark locations used by the "IntraFace" library. Isaac Newton's figure from [71].

5.4. Face alignment

Face images, gathered from many domains have various resolutions and orientations. These differences can cause significant challenge to learning the differences between the perceived sex. For that reason, prior to extraction of the facial features, preprocessing steps are used to align and scale the findings. The primary objective of this step is to minimize the differences between faces that belong to the same category while ensuring the unique characteristics of the perceived sex (male or female) is maintained.

5.4.1. Generalized Procrustes Analysis

In order to align the segmented faces from Section 5.2 with respect to each other, a global destination target with optimal facial landmark locations is needed. All face landmark coordinates from training datasets (10k US Adult Faces, IMDb, and LFW) are used to find this general face alignment. Generalized Procrustes Analysis [49] (GPA) is used to compute the optimal facial landmark coordinates to superimpose the golden face.

The Procrustes analysis uses k landmark points in face denoted as X_k such that each face is represented as a combination of these k landmarks. Procrustes calculates translation, uniform scaling, and rotation from these X_k pairs between the samples.

For translation step, all the shapes are moved such that all mean of the coordinates are translated so that the mean coordinates of the landmarks given in Equation (5.1) is a zero vector. This can be achieved by simply moving every point X_k by the $\bar{X}$ as shown in Equation (5.2). This translation step is performed individually on every shape, to align them to the same origin.

$$\bar{X} = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} \frac{1}{k}\sum_{i=1}^{k} x_i \\ \frac{1}{k}\sum_{i=1}^{k} y_i \end{bmatrix} \tag{5.1}$$

$$X_k \rightarrow \begin{bmatrix} x_k - \bar{x} \\ y_k - \bar{y} \end{bmatrix} \tag{5.2}$$

Uniform scaling, like translation step, scales the given shape so that its root mean square distance (RMSD) from the landmarks to the translated origin is 1. This scaling coefficient s is calculated from Equation (5.3) and becomes one when the landmark point coordinates are divided by this scale as shown in Equation (5.4).

$$s = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \tag{5.3}$$

$$X_k \rightarrow \frac{1}{s}\begin{bmatrix} x_k - \bar{x} \\ y_k - \bar{y} \end{bmatrix} \tag{5.4}$$

Computing a rotation alignment is more complex and requires a reference face to perform. Assuming that there are landmark coordinates from two faces, which have scale and translation removed given as X^a and X^b. One can be used to provide a reference orientation, such that an optimum angle of rotation θ exists where the sum of the squared distances (SSD) between the landmark pairs is minimized. This rotation angle θ can be written as Equation (5.5) when X^b is used as a reference. Sum of the squared distances between Equation (5.5) and X^b yields $\sum_{i=1}^{k}(\hat{x}_i^a - x_i^b)^2 + (\hat{y}_i^a - y_i^b)^2$, and taking derivative of SSD with respect to θ and solving when this derivative is zero gives Equation (5.6).

$$\hat{X}_k^a = \begin{bmatrix} \hat{x}_k^a \\ \hat{y}_k^a \end{bmatrix} = \begin{bmatrix} cos\theta\, x_k^a - sin\theta\, y_k^a \\ sin\theta\, x_k^a + cos\theta\, y_k^a \end{bmatrix} \tag{5.5}$$

$$\theta = tan^{-1}\left(\frac{\sum_{i=1}^{k}(x_i^a y_i^b - y_i^a x_i^b)}{\sum_{i=1}^{k}(x_i^a x_i^b + y_i^a y_i^b)}\right) \tag{5.6}$$

Finally, the following steps are performed to extract the golden face landmark coordinates:

(i) Face landmark coordinates from an arbitrary face are chosen as golden face landmark coordinates.

(ii) All faces from the training set are superimposed, – translated, scaled uniformly

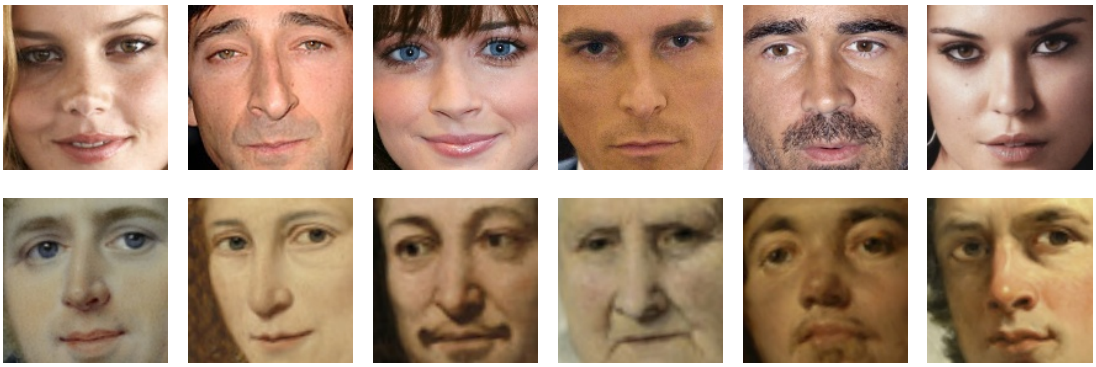


Figure 5.9. Aligned face examples from IMDb and Rijksmuseum datasets. These faces are translated, uniformly scaled and rotated to align with respect to golden face landmark coordinates.

and rotated – with respect to current golden face landmark coordinates.

(iii) Mean shape of the current set of superimposed shapes is computed as next candidate golden face landmark coordinates.

(iv) If the SSD between the current and candidate golden face landmark coordinates is above a threshold, use candidate as the next golden face landmark coordinates and go to Step 2.

Translation, uniform scaling, and rotation operations calculated through alignment process are used to align all the faces to a comparable domain. Alignment results in IMDb dataset and Rijksmuseum paintings can be seen in Figure 5.9.

5.5. Face features

We have used two types of facial features, local binary patterns [61] (LBP) which is commonly used to describe the face [58, 62, 63, 64, 65] and deep learning reinforced Parkhi's visual geometry group face descriptor [60] (VGG).

5.5.1. Local binary patterns

LBP is calculated on face images as follows:

(i) Each face is aligned into 136×136 pixels using golden face landmarks.
(ii) The face area is divided into a grid of 9×9 cells, where each cell is 16×16 pixels.
(iii) A comparison for each pixel in a cell is made circularly, in the counter-clockwise direction. When the center pixel's value is greater than that of its neighbor's value, "0", otherwise "1" is written to form an 8-digit binary number.
(iv) Histogram of the cell is computed and normalized, such that sum of its values would equal to 1. This histogram represents the occurrence of pixels that are smaller or greater than the center.
(v) Barkan et al. shows at most two transition patterns occur more of than the other patterns and reduced the size of the histogram by grouping up all pixels which have more than two transitions [72]. This representation is called uniform binary patterns and reduces the number of bins in Step 4 from 256 to 59.
(vi) Concatenate all 81 cells' uniform binary patterns into a feature vector with 4,779 values, which would have been 20,736 dimensions if regular binary patterns approach were to be pursued.

By iterating above steps for each face image, each training and test face sample is represented as a point in LBP feature space.

5.5.2. VGG-Face

Parkhi's VGG-Face is not a descriptor, but a classifier to be used for face recognition purposes. It is trained to distinguish 2,622 unique individuals, trained on a convolutional neural network (CNN) architecture given in Figure 5.10. The network consists of 8 blocks; each piece contains linear operations such as convolution and non-linear operations, e.g., rectification layer (ReLU) and max pooling. First 5 blocks are called convolutional layers because the linear operator performs linear convolution. The remaining three layers are Fully Connected (FC) as their filter size is the same as

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmx
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Figure 5.10. The network structure of VGG-face. Figure from [60]

.

the input data.

Crowley et al. used a similar network which was trained for ImageNet Large Scale Visual Recognition Challenge with thousand distinct classes and used it for painting content retrieval system [15]. In their work, they have used the results from the penultimate layer for the feature extraction.

We have followed their footsteps on another network published by the same team (visual geometry group), dedicated for faces and used the outputs before last FC layer, which is marked in Figure 5.10 as layer 35. This cut ensures that the network retains a high representation of the facial descriptors, instead of specializing in the 2,622 individuals.

Finally, we have used VGG-Face network and transformed it into a facial descriptor that consists of a CNN network that contains 35 layers. The output of this network is in 4,096 dimensions. In other words, every face sample is projected as a point into this VGG-Face space.

5.6. Classification

For classification, one of the most frequently used classification method, support vector machines (SVM) with radial basis function kernel (RBF) [73, 74] is explained in Section 5.6.1 and Random decision forests (RDF) is given in Section 5.6.2. Classifi-

cation results are discussed in Section 5.6.3.

5.6.1. Support Vector Machines

Support vector machines are a supervised learning model to find the boundary of the samples, which are faces with perceived sex male or female in the context of this thesis. This border is defined from the sample points in the feature space.

Let $\overrightarrow{x}_1, \ldots, \overrightarrow{x}_n$ represent the dataset of n points, spawned through previously mentioned feature extraction steps. Then, $(\overrightarrow{x}_1, y_1, \ldots, \overrightarrow{x}_n, y_n)$, where y_i represents the class of each sample x_i.

SVM defines the classification problem on "Maximum-margin hyperplane" such that Equation (5.7) can be written to define any hyperplane where $\overrightarrow{w}$ is the normal vector to the hyperplane.

$$\overrightarrow{w} \cdot \overrightarrow{x} - b = 0 \tag{5.7}$$

Depending on whether the data is linearly separable in feature space, two hyperplanes shown in Equation (5.8) can be selected to classify it, such that distance between them is maximized. The distance between the planes is $\frac{2}{\|\overrightarrow{w}\|}$, therefore, objective can be written as a minimization for $\|\overrightarrow{w}\|$.

$$\begin{aligned} \overrightarrow{w} \cdot \overrightarrow{x} - b &= 1 \\ \overrightarrow{w} \cdot \overrightarrow{x} - b &= -1 \end{aligned} \tag{5.8}$$

Finally, for all training points x_i where $y_i = 1$, $\overrightarrow{w} \cdot \overrightarrow{x}_i - b \geq 1$ and for $y_i = -1$, $\overrightarrow{w} \cdot \overrightarrow{x}_i - b \leq -1$ ensures that the defined hyperplanes separate the points correctly,

which can be represented by Equation (5.9).

$$y_i(\overrightarrow{w} \cdot \overrightarrow{x}_i - b) \geq 1 \tag{5.9}$$

Combined together as an optimization problem, linear SVM can be defined by Equation (5.10). Namesake of the method comes from the points that lie on this hyperplane, called support vectors, as points that are away from the hyperplane is not used for the solution. The solution $\overrightarrow{w}$ and b is the classifier, such that $sgn(\overrightarrow{w} \cdot \overrightarrow{x} - b) \rightarrow y$.

$$\begin{aligned} &\underset{\overrightarrow{w}, b}{\text{minimize}} \quad \|\overrightarrow{w}\| \\ &\text{subject to} \quad y_i(\overrightarrow{w} \cdot \overrightarrow{x}_i - b) \geq 1, \ i = 1, \ldots, n \end{aligned} \tag{5.10}$$

Boser et al. introduced nonlinear classification using the kernel method. A nonlinear kernel function is defined such that, each point x_i is projected on to a higher dimensional space, where the resulting hyperplane is not necessarily linear in the original input space.

Gaussian radial basis function (RBF) kernel defined on two sample points x and x' is represented as $K(x, x') = e^{-\gamma\|x-x'\|^2}$. Due to the fact that the value of the kernel decreases with distance and ranges between zero and one, it can be used as a similarity measure.

RBF is one of the most popular kernel function used for SVM classification problems [75]. It has two hyperparameters that directly influence learning performance: γ and c. The γ parameter defines the influence of each support vector, and the c parameter trades off the generalization of the classifier. High c values aim to classify more training examples correctly, than lower c values. There are several different approaches for hyperparameter selection. The most common way is to search for optimal c and γ pairs inside a logarithmic grid using cross-validation performances to evaluate each parameter pair.

For perceived sex recognition on paintings, we have used Weka library's implementation for SVM with RBF kernel and grid search algorithm [76]. In this implementation of the grid search, c and γ are mapped from variables α and β. Mapping functions are mostly used as exponential with the customizable base such that $c = \underset{c_{base}}{BASE}^{\alpha}$ and $\gamma = \underset{\gamma_{base}}{BASE}^{\beta}$. These parameters can be represented as a grid with minimum, maximum and step values defined by the user. The algorithm first performs a coarse initial search through this grid using 2-fold cross-validation, where half the training samples are used to train the classifier, to measure the performance on the other half, for both halves. The average performance is used to assess the starting point in the grid. Then, 10-fold cross-validation performance is measured on the current and all adjacent hyperparameter pairs. This process is repeated until either local maximum is reached or parameter pairs hit the edge of the grid. In the former case, algorithm concludes, in the latter case user can opt to extend the grid to resume the search indefinitely, extend the grid a number of times or stop the algorithm.

In this work, we take $c = 10^{\alpha}$ with $\alpha = -1, \ldots, 0, \ldots, 6$ and similarly $\gamma = 10^{\beta}$ where $\beta = -6, \ldots, -2$. Grid is allowed to extend maximum 10 times to pursue the best performance. SVM performances and result comparisons are given in Section 5.6.3.

5.6.2. Random decision forests

Random decision forests (RDF) are introduced by Breiman et al. in 2001 [77]. It consists of an ensemble of decision trees that are constructed from training data, outputting the probability distribution of the class or values defined in the training data for each test sample. These results are then used to estimate the class (classification) or the value (regression).

Each classification and regression tree (CART) [78] divides the feature space into sets of disjoint rectangular regions. These regions are leaves of the tree. Root is the entry point, the whole feature space and through training samples, decisions are made at each region whether to split or end the decision tree in which case the node becomes the leaf.

There are several methods for the split decision. Common uses for classification problems include:

- Gini impurity: $S_G(f) = 1 - \sum_{i=1}^{m} f_i^2$ where f_i is the fraction of items labeled with class i and m is the number of classes
- Information gain: $IG(T, a) = H(T) - H(T|a)$ where $H(T)$ is information entropy of set T and a is the feature's index (x_a). Therefore, it is the change in information entropy from the non-split state to state with the split.
- Misclassification error, gain ratio, deviance, variance, chi-squared distance among others.

At each split, the training set is divided and therefore as the depth increases, the tree starts to overfit. There are several methods to address this issue, such as an acceptable node purity to avoid splits on regions which are not 100% single class (pure), setting a minimum number of training samples that are required for separation, so that decisions are not made by only a fraction of the samples, limiting the maximum depth and pruning the tree nodes after the tree is formed using cross-validation performances.

Breiman et al. introduced bootstrap aggregation (bagging) in 1996 [79]. First, new training sets are generated by random sampling the original one $N' \leq N$ with replacement. This process is called Bootstrap sampling. Bagging is creating a combination of learners, independently trained on distinct bootstrap samples. Therefore, each learner is subject to only a sample of the original training data.

An ensemble of such trees addresses the overfitting nature of the decision trees by training each tree with another bootstrap sample. This ensures that data that the generalization error is reduced and the trees that overfit to their training samples do not necessarily overlap and therefore through training many such trees variance can be minimized.

Furthermore, a second randomization is introduced for the forest on the node level, where at the training of the splits, a random subset of the samples and feature

Table 5.2. Perceived sex recognition performance on Rijksmuseum. All results are comparable and the best result (by a small margin) is acquired with simplest features, and with only IMDb dataset.

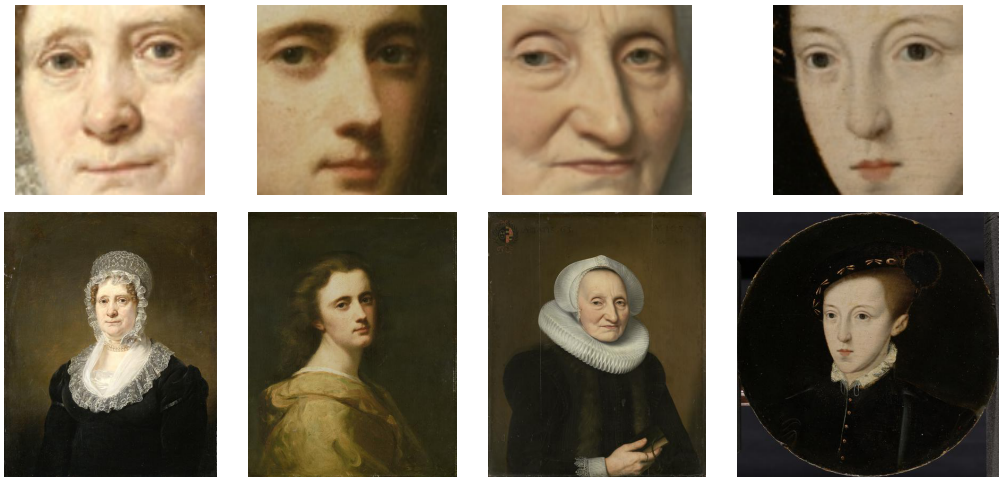
	IMDb	IMDb and 10k	IMDb, 10k and LFW
LBP&RDF	76.35%	76.94%	71.96%
LBP&SVM	**80.40%**	76.41%	76.25%
VGG&SVM	78.14%	76.54%	77.54%

space is considered. This randomness ensures that even the trees that are trained by similar training samples would still split in a different way to increase variety. However, if the feature space is too large with only a small number of informative predictors, significantly lowering the feature space might reduce the accuracy of the forest.

RDF can train fully in parallel - trees need not know one another, and it requires little parameter tuning, unlike SVM where hyperparameter optimization is crucial for the performance. Therefore training a forest takes significantly less effort and computation time than an SVM. However, SVM can outperform RDF [58] with hyperparameter optimization. Perceived sex prediction task used RDF due to long training times and sensitivity to parameter selection on SVM.

5.6.3. Classification results and remarks

The most significant challenge for evaluating perceived sex recognition performance on the paintings was to make sure the ground-truth perceived sex data are actually correct and accountable [80]. From our experience, this demanding task requires a full view of the painting, rather than just the detected face, which can easily be misinterpreted. We have prepared a short list of examples of incorrectly classified paintings to remark how facial rectangle by itself can be tricky in Figure 5.11.



(a) Sitters perceived as female, assigned to the male category by perceived sex recognition method.

(b) Sitters perceived as male, assigned to the female category by perceived sex recognition method.

Figure 5.11. Sitters of portrait paintings assigned to a sex that is different than the annotation by perceived sex recognition method.

Figure 5.12. IMDb female samples tend to have more makeup in general, compared to female sitters in paintings of Western art.

A comparison of the performance of RDF and SVM classifiers over portrait paintings of Rijksmuseum dataset is given in Table 5.2. In order to assess the training dataset, three sets of data are used to train the classifiers; our queried dataset from IMDb, a combination of IMDb and 10K database and finally a combination of IMDb, 10K and LFW. In this table, columns represent the combinations of training datasets, and each row represents a learning methodology. Majority of the performances are very similar. However, when used as a single training dataset, IMDb shows the most promise. It achieves a significant classification performance with 80.40% on SVM classifier, without using any other training dataset. Such result can be interpreted due to the similarity of the posture of actors and actresses pictures in IMDb dataset with the portraits. Hypothesis on this topic will be given at the end of this section.

The detailed results of classification of male and female on the painting where the classifier is trained by IMDb are given in Table 5.3. From total counts of 499 portraits of females, 281 is classified correctly as female. However, in case of the portrait with male sitters, 929 was identified correctly from the total number of 1,006 giving 92.35% performance. Some examples of the training images for the female category are given in Figure 5.12. It is likely that this slightly contradicting result may be due to the cosmetic colors within the training set.

Through our study, facial similarities between poses of 21st-century actors and actresses found in search engine results and poses from paintings in the Western art became notable. This similarity is further shown by the high performance of the gathered IMDb dataset given in Table 5.2. It is normal that the style observed on the Western art portrait paintings is adopted in the modern style of the portrait photography, as

Table 5.3. Perceived sex recognition performance trained from IMDb and tested on Rijksmuseum using LBP features and SVM with RBF kernel.

	Classified as			
Annotated as	Female	Male	Total Count	Performance
Female	281	218	499	56.31%
Male	77	929	1,006	92.35%
Total	358	1,147	1,505	80.40%

well.

5.7. Style Transfer

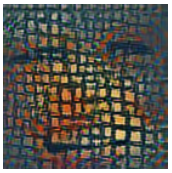
Many researchers showed that it is less costly and preferable to transfer existing knowledge from one domain to another than to try and collect data only available for the domain of the study [15, 81, 82, 83].

Gatys et al. introduced a technique of recomposing an image using the style of other images [84]. This method soon is called "Style Transfer" [82] and became a quick hit on the social media where any image could be stylized using paintings. This process provides 21st century photography to resemble that of the source painting's – or painter's – style. Such a method can be used to transfer more popular and available gender datasets to that from the painting world and increase the available training data. For this reason, we have applied a previously trained neural network to move our training and test database.

Gatys et al. published a network that has 32 predefined styles from various times and artists [82]. Some examples of how a 21st Century photograph is represented in these paintings' style is given in Figure 5.13.

From these 32 styles and possibly endless other paintings we could have used for

(a) *Roy Lichtenstein, Bicentennial Print (1975)*

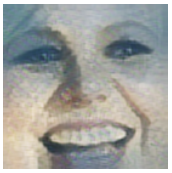


(b) *Paul Klee, Colors from a Distance (1932)*

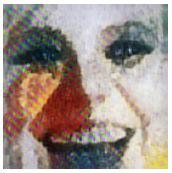
(c) *Jamini Roy, Crucifixion*

(d) *Giorgio de Chirico, Horses on the seashore (1927/1928)*

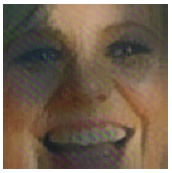
(e) *Vincent van Gogh, Landscape at Saint-Rémy (1889)*



(f) *Juan Gris, Portrait of Pablo Picasso (1912)*

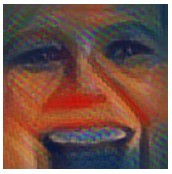


(g) *Severini Gino, Ritmo plastico del 14 luglio (1913)*



(h) *Alice Bailly, Self-Portrait (1917)*

(i) *Grayson Perry, The Annunciation of the Virgin Deal (2012)*

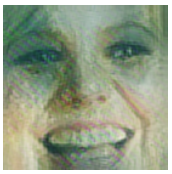


(j) *Edvard Munch, The Scream (1910)*

(k) *Vincent van Gogh, The Starry Night (1889)*

(l) *Wolfgang Lettl, The Trial (1981)*

(m) *Douglas Coupland, Thomson No. 5 (Yellow Sunset) (2011)*



(n) *Berthe Morisot, Under the Orange Tree (1889)*

(o) *Giulio Romano (Giulio Pippi), Victory, Janus, Chronos and Gaea (about 1532-1534)*

(p) *Wassily Kandinsky, White Zig Zags (1922)*

Figure 5.13. Stylized face crop of Adelaide Clemens, showing a subset of the 32 predefined styles distributed by Gatys et al.[82].

Figure 5.14. Aligned face examples from IMDb and Rijksmuseum datasets, style transferred with Giorgio de Chirico, Horses on the seashore (1927/1928).

training our own system, we have picked one that would transfer face rectangles into a domain that would appear the most painting-like at the resolution of the face rectangle. *Giorgio de Chirico, Horses on the seashore (1927/1928)* is empirically chosen as the first such style from the preliminary transfer results given in Figure 5.13. This new style is then applied to all other face crops mentioned in Section 5.2. Style transferred examples can be seen in Figure 5.14. Hence, we have generated another set of training and testing datasets.

In this stylized space, steps in Section 5.5 and Section 5.6 are repeated to measure the difference in the performance. The resulting confusion matrix can be seen in Table 5.4. Stylized framework correctly identified one less female sample Female group, but three more from males. This results in 2 misclassifications less out of previous 295 samples and yields approximately 1% less misclassification. Although the results are only marginally better than previous results, it is evident that style transfer approach can be extended by selecting various paintings. Our hypothesis is that if we can

Table 5.4. Perceived sex recognition performance trained from stylized IMDb and tested on stylized Rijksmuseum using LBP features and SVM with RBF kernel are given next to non-stylized results for easy comparison.

Original IMDb images

	Classified as			
Annotated as	Female	Male	Total Count	Performance
Female	281	218	499	56.31%
Male	77	929	1,006	92.35%
Total	358	1,147	1,505	**80.40%**

Stylized IMDb images

	Classified as			
Annotated as	Female	Male	Total Count	Performance
Female	280	219	499	56.11%
Male	74	932	1,006	92.64%
Total	354	1,151	1,505	**80.53%**

use only a single painting and improve the performance of perceived sex recognition without increasing the amount of training data, then it should be possible to use multiple styles, preferably from the destination schema - e.g., from Rijksmuseum - to enrich and increase the performances further. We talk about style transfer future work further in Chapter 7.

6. CLOTHING COLOR EXTRACTION

In this chapter, we describe how to estimate the dominant clothing color of a sitter in a painting. We need to find some pixels representing the area of interest - clothing in this case - explained in Section 6.1, which will be used for clustering and predicting the dominant color or palette in Section 6.2.

6.1. Segmentation of the clothing

Given a painting including a subject, we would like to determine the main color of the clothing for that subject. Ideally, this requires automatic segmentation of the body area of the person. It is less important to find the exact segmentation boundaries of the person, which is a difficult problem that requires large amounts of annotated images for supervised training [85, 86]. In the PASCAL VOC [85] and MS COCO [86] benchmarks, great amounts of object boundary segmentations were collected from human annotators via Amazon's Mechanical Turk service. This is an expensive process. However, for finding the colors of clothes, a coarse segmentation may suffice.

There are several prior studies with regard to segmentation of human clothing [87, 88]. Kalantidis et al. introduce a system to use a model image for pose estimation and use the pose for clothing segmentation. They use this approach to retrieve products from online shopping catalogs [87]. Gallagher et al. use clothing segmentation to track and to identify the same individual on different photos taken in a reasonable time interval, for example inside a shopping mall [88]. They use the graph cut algorithm and various pictures of the same person to train a similarity metric on the clothing mask.

Similar to Gallagher et al., we have used the graph cut approach; using facial landmark points to initiate the segmentation process on the GrabCut algorithm [89]. Our approach and the results are explained in Section 6.1.1. Moreover, in the direction of Kalantidis et al., a state of the art method to segment the body parts and the pose

from the images is used to improve the accuracy of the segmentation results. This technique is explained in Section 6.1.2.

Although the portrait paintings focus on the sitter's face, they retain a significant amount of background information that disrupts the color extraction of the paintings. Some color palettes before and after clothing segmentation are given in Figure 6.1 and Figure 6.2. We hypothesize the color palette from the clothing of the model will reflect the color scheme that is associated with the perceived sex of the sitter.

Majority of the paintings have non-clothing material, for example, background color, skin color. This can be remedied by a coarse segmentation of the garb mentioned in Figure 6.1 and Figure 6.2. It is seen that in Figure 6.1, the color of the dark background and the sky make the majority of the color palette. Similarly, in Figure 6.2 palette is skewed towards the collar and the background colors. Details on how these palettes are generated and further details on color quantization after the segmentation process are explained in Section 6.2.

6.1.1. GrabCut

The image is represented as an array of $\mathbf{z} = (z_1, \ldots, z_n, \ldots, z_N)$ of points in RGB color space, indexed by n. On this representation, segmentation of the image is $\underline{\alpha} = (\alpha_1 \ldots, \alpha_N)$. In general, α_n is predicted by the algorithm, except when a hard segmentation where $\alpha_n \in \{0, 1\}$ is provided. Low α_n values represent the background and higher values are for the foreground. 0 value is reserved for hard background and 1 for hard foreground information.

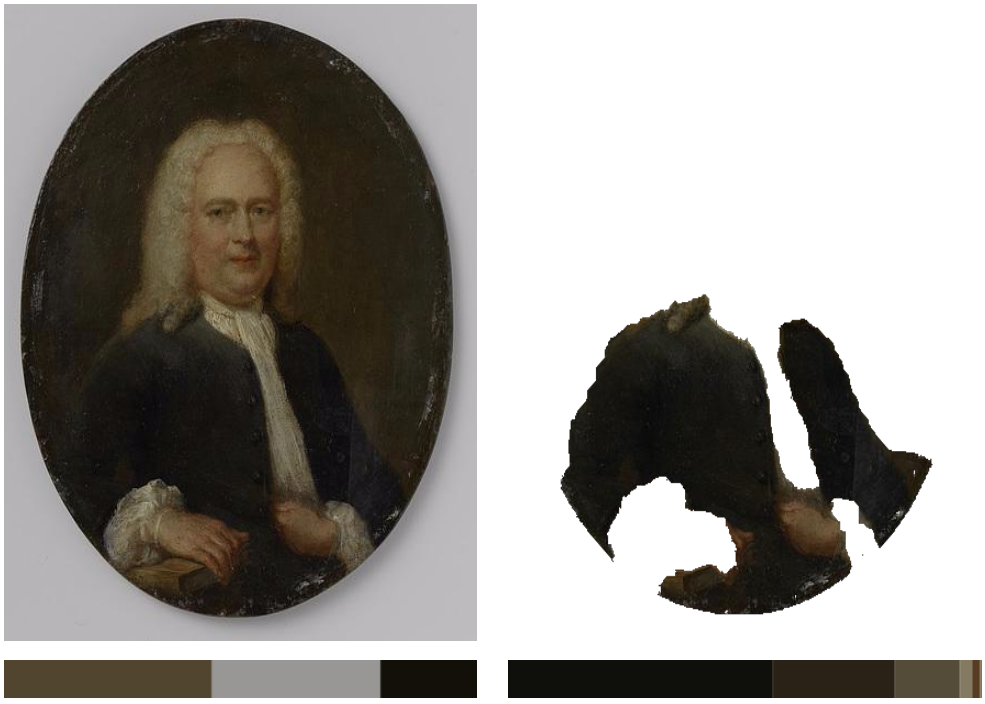
The GrabCut algorithm uses two Gaussian Mixture Models (GMMs) for the background and the foreground. These models have full-covariance Gaussian mixtures with K components, ($K = 5$ for GrabCut [89]). In order to represent GMM components, an additional vector $\mathbf{k} = \{k_1, \ldots, k_n, \ldots, k_N\}$ is used where $k_n \in \{1, \ldots, K\}$ which assigns a unique GMM component from the background or the foreground model to each pixel, depending on $\alpha_n = 0$ in the former case or $\alpha_n = 1$ for the latter. $alpha_n$ takes only

(a) *Portret van een jongetje met een bok*

(b) *Isabella Clara Eugenia van Habsburg (1566-1633).*

Figure 6.1. Extracted palette before and after coarse segmentation (Part 1).

(a) *Portret van Ambrogio Spinola (1569-1630).*



(b) *Portret van een man*

Figure 6.2. Extracted palette before and after coarse segmentation (Part 2).

these two values as the authors avoid soft assignments of probabilities due to significant computational expense that they claim results in little practical benefit [89]. The parameter $\underline{\theta}$ describes parameters for the GMM component given in Equation (6.1), where π are weights, μ means and Σ covariances of the $2K$ Gaussian components (K for the background and another K for the foreground).

$$\underline{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \quad \alpha = 0, 1, k = 1 \dots K\} \tag{6.1}$$

An energy function E is used, where low values represent a good segmentation. The segmentation is guided by the foreground and the background GMM components. This is represented by a "Gibbs" energy of the form shown in Equation (6.2).

$$E(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}) \tag{6.2}$$

U is called the data term and it evaluates the $\underline{\alpha}$ on the data z, with given model $\underline{\theta}$ as shown in Equation (6.3), where $D(\alpha_n, k_n, \underline{\theta}, z_n) = -log\, p(z_n|a_n, k_n, \underline{\theta}) - log\, \pi(\alpha_n, k_n)$. $p(\cdot)$ is the Gaussian probability distribution, and $\pi(\cdot)$ is the mixture weighting coefficients. Therefore, D can be written as Equation (6.4).

$$U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n) \tag{6.3}$$

$$D(\alpha_n, k_n, \underline{\theta}, z_n) = -log\,\pi(\alpha_n, k_n) + \frac{1}{2} log\,det\,\Sigma(\alpha_n, k_n) + \frac{1}{2}[z_n - \mu(\alpha_n, k_n)]^T \Sigma(\alpha_n, k_n)^{-1}[z_n - \mu(\alpha_n, k_n)] \tag{6.4}$$

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n)\in\mathbf{C}} [\alpha_n \neq \alpha_m] e^{-\beta\|z_m - z_n\|^2} \tag{6.5}$$

The smoothness V is written as the Equation (6.5), where $[\phi]$ is the binary function taking values $0, 1$ for the predicate ϕ, C is the set of the neighboring pixels, encouraged by the authors to form this set from horizontal, vertical or diagonal adjacency (8-way connectivity). β and γ represent the smoothness coefficients. Boykov and Jolly [90] shows that β constant can be chosen as:

$$\beta = \left(2\left\langle (z_m - z_n)^2 \right\rangle\right)^{-1}$$

where $\left\langle \cdot \right\rangle$ shows the expectation over the image sample. The contrast term is calculated using Euclidean distance in color space. The constant γ is empirically selected to be 50 by another study from one of the authors [91] and this value is used as constant through the implementations given in OpenCV [92].

This energy minimization scheme works iteratively. Initial trimap T is created by the user by drawing a region of interest as seen in Figure 6.3. Any pixel that is outside this region of interest is put into T_B and inside this region of interest form T_U. The foreground is set to $T_F = \emptyset$. On Step 1, k_n values are enumerated for each pixel n. In Step 2, a set of Gaussian parameter estimations take place. For each GMM component k in a (foreground or background) model, a subset of pixels $F(k) = \{z_n : k_n = k \,\text{and}\, \alpha_n\}$ (value of $\alpha_n = 0$ for the background model and $\alpha_n = 1$ for the foreground the model of the GMM component) is defined to calculate the sample mean $\mu(\alpha, k)$ and covariance $\Sigma(\alpha, k)$ in $F(k)$. Mixture weighting coefficients are $\pi(\alpha, k) = |F(k)|/\Sigma_k|F(k)|$. Finally, a min-cut algorithm is used to segment the

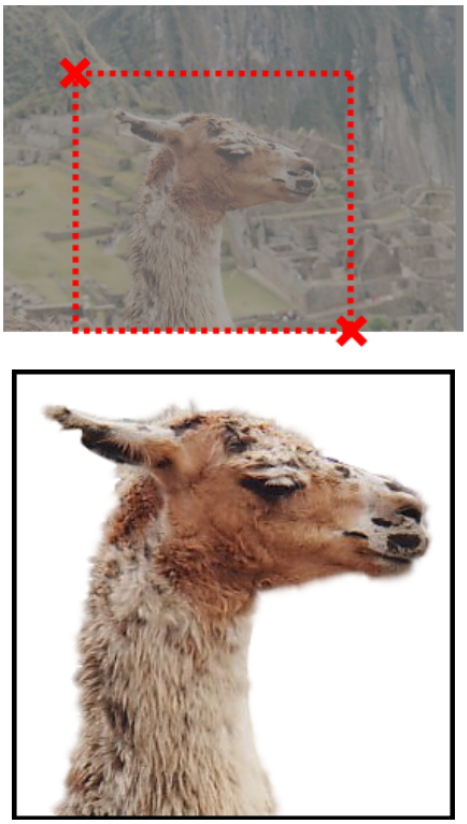


Figure 6.3. GrabCut segmentation example. Figure from [89].

graph to satisfy Equation (6.6).

$$\underline{\hat{\alpha}} = \arg\min_{\underline{\alpha}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}) \tag{6.6}$$

Therefore, each step minimizes the total Energy $\mathbf{E}$ with respect to pixel ownership of the GMM component coefficient sets $\mathbf{k}$, Gaussian parameter estimations θ and segmentation results α in turn. This means that $\mathbf{E}$ reduces at every step, and hence the algorithm is guaranteed to converge.

The initialization mask required by the GrabCut algorithm is generated based on the previously detected facial landmark coordinates. In order to obtain a good position, we use the distances of the facial landmarks as a yardstick. For this purpose,

Figure 6.4. The region of interest for the GrabCut algorithm.

it is possible to define unit distances within the face, by facial registration [93, 94]. In previous studies, faces are aligned by cropping and resizing them to a fixed size (for instance 96×96 pixels), where the center of eyes and mouth lie at a fixed row and column positions.

Let $P_e = \begin{bmatrix} x_e \\ y_e \end{bmatrix}$ denote the center point of the two eyes, and let $P_m = \begin{bmatrix} x_m \\ y_m \end{bmatrix}$ denote the center of the mouth $d_u = \|\overrightarrow{P_e P_m}\|$ is defined to be constant through face registration approach and is equal to the one-third of the face. In other words, the width and the height of the face square in d_u units are $3\,d_u$. The vector that gives the direction for the cloth segmentation area can be set to $\overrightarrow{P_e P_m}$. Initial mask for clothing was determined empirically over a few dozen examples. The location of it is from the center of the face denoted by P_c, in direction of $\overrightarrow{P_e P_m}$ with magnitude d_m and width

w_m and height h_m are given in Equation (6.7).

$$\begin{aligned} d_m &= 6 \quad d_u \\ w_m &= 12 \quad d_u \\ h_m &= 9 \quad d_u \end{aligned} \tag{6.7}$$

Out of total $6d_u$, the first $1.5d_u$ in this direction leads to the bottom of the face, and another $4.5d_u$ ensures the area center is safely inside the clothing. Then, the width and the height of the mask is defined to cover the approximate clothing region. One such region of interest for the GrabCut algorithm is created as seen in Figure 6.4.

With above heuristics, the region may contain additional pixels. Every so often, exposed skin from the cleavage, arms or hands of the sitters will be included in the region. In order to enhance the GrabCut segmentation process, skin-like pixels are identified and excluded from the area of interest. These pixels are located using the colors extracted from the sitter's cheeks. Landmark points extracted for the eyelids are used to estimate the center of both of the eyes. For each cheek, $0.5d_u$ distance is travelled from the center of the eye in $\overrightarrow{P_eP_m}$ direction. An example skin exclusion result is shown in Figure 6.5.

In order to estimate the performance of the segmentation approach, a subset of the paintings are evaluated. From 40 paintings, 13 were segmented completely using only the rectangle mask. Upon tuning the algorithm with the skin color extraction, this amount has been increased to 34.

GrabCut results mostly contain the majority of the colors on the clothing. However, any additional objects with the sitter such as an armor piece or a baby can influence the segmentation results. Color quantization and extraction steps are explained in detail in Section 6.2.

Figure 6.5. Exclusion of skin-like pixels for the GrabCut segmentation. The painting with cheek regions marked yellow is shown on the top. Initial segmentation results without skin exclusion are given to the left and with the skin exclusion algorithm are to the right.

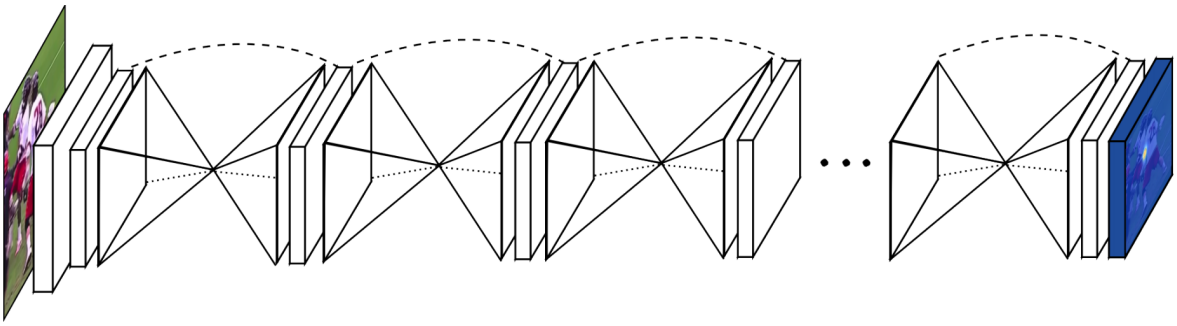


Figure 6.6. Stacked hourglass modules. Figure from [96].

6.1.2. The SURREAL Segmentation Approach

Varol et al. introduced a state of the art segmentation method using synthetic models and have shown very good results on videos [95]. They attempt to address one of the fundamental challenges for the human pose, shape and motion estimation, namely, the need for a huge amount of annotated data. The "Synthetic hUmans foR REAL tasks" (SURREAL) dataset they have introduced contains six million figures with ground truth information that contains synthetically generated, but realistic images of people, rendered from 3D sequences of human motion capture data.

This dataset addresses the segmentation problem by assigning one of 15 predefined categories to each pixel. These are 14 human parts (head, torso, upper legs, lower legs, upper arms, lower arms, hands and feet; separately for right and left when applicable), and the background, respectively.

SURREAL uses stacked hourglass network architecture introduced by the study of Newell et al. "Stacked Hourglass Networks for Human Pose Estimation"[96]. The network structure can be seen in Figure 6.6. SURREAL uses eight such hourglass modules, which are stacked on top of each other. Each stack takes the prior stack's output as its input. SURREAL's network input is a 3-channel RGB image of size 256×256, cropped and scaled to fit a human bounding box using the ground truth. The network output has dimensions $64 \times 64 \times 15$ per stack (i.e., the 14 previously mentioned body parts and the background). It uses cross-entropy loss on all pixels, where the final loss is summed over all 8 stacks. The system is pre-trained with synthetic data

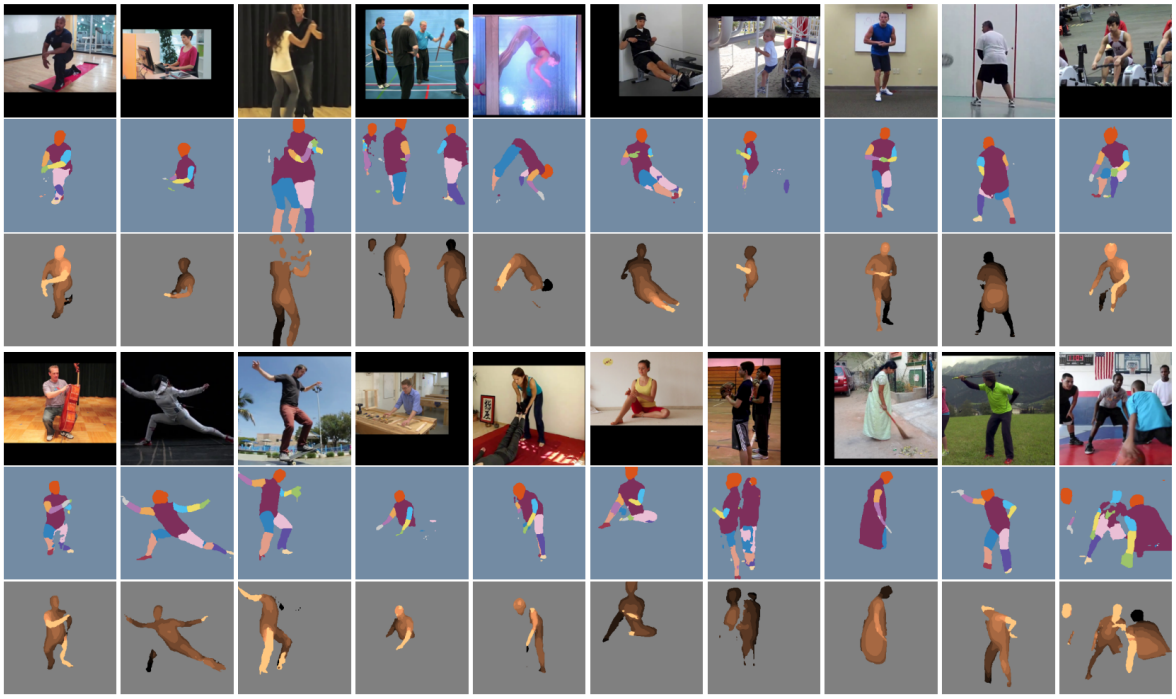


Figure 6.7. SURREAL results. Figure from [95]

for over 50K iterations. They show that CNNs trained on such a dataset allow for accurate human part segmentation in real RGB images, as seen in Figure 6.7.

This system is used to perform clothing segmentation using a model that is trained using lossless renderings (PNG) of the SURREAL training set distributed by the authors.

We have applied The SURREAL segmentation on the Rijksmuseum painting data with its pretrained model. Several examples of the segmentation results are given in Figure 6.8. Unfortunately, segmentation results on paintings were more challenging to generalize from the SURREAL training set, and a system trained on such videos did not perform as well on paintings. We believe there are improvements that can be achieved by using transfer training on the system with paintings. It may be possible to generate "painting" versions of the training images through style transfer and fine-tune the trained network to work better on painting segmentation. This, however, is a significant undertaking and left as a future study.

Figure 6.8. SURREAL results on Rijksmuseum paintings. From left to right, original painting, SURREAL segmentation result and GrabCut segmentation result.

6.2. Color quantization

Once the segmentation is completed, we need to summarize the colors contained by the garment. The result of the segmentation can be treated as a cloud of pixels, for which a representative color or a small set of representative colors should be determined. We do not have any prior assumptions on the shape of this point cloud, nor how many modes it may contain. Subsequently, an unsupervised approach can be adapted to find the dominant forms in the cloud, and to represent them parsimoniously.

For this challenge, we use different clustering algorithms to group shades of the same colors. In this approach, each cluster should represent a unique color inside the region, with weights showing the actual distribution of the said color. Basic statistical approach k-means and the more advanced incremental mixtures of factor analyzers (IMoFA) approaches are used for comparison. While k-means assumes spherical and identical covariance matrices in the clusters, IMoFA allows arbitrary shapes in the color space. The descriptions of these approaches and the experiments are explained in the sections below.

6.2.1. K-means clustering

K-means clustering is one of the fundamental methods for vector quantization [97]. It partitions the given set of instances into k clusters to minimize the within-cluster sum of squares which can be described as Equation (6.8), where S is set of instances, x represents each instance, μ is a cluster center, and k is the number of clusters.

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2 \tag{6.8}$$

Algorithmic steps for the k-means calculation:

(i) Select k cluster centers randomly.
(ii) Calculate the Euclidean distance $||x - \mu_i||^2$ for each data point $x \in S$ and all cluster centers μ_i where $i = 1, 2, \ldots, k$.
(iii) Assign the data point to the cluster center with minimum distance.
(iv) Calculate new cluster centers by $\mu_i = \frac{1}{c_i} \sum_{j=1}^{c_i} x_j$ where c_i is the total number of elements in cluster number i.
(v) Check if new cluster centers are same with the old cluster centers. If there has been no change, terminate. Otherwise, go to step 2.

k-means is fast, robust and very easy to understand. However, it relies on a knowledge of the number of cluster centers, which is not available in a clothing region.

6.2.2. Incremental Mixtures of Factor Analyzers

Salah and Alpaydin introduced the incremental mixtures of factor analyzers (IMoFA) approach, which is a semi-parametric density estimator that shows favorable results on different pattern classification tasks [98]. We have used it to flexibly fit a mixture model to the colors of the clothes and to represent the dominant colors of the paintings. In this section, we summarize this approach briefly.

A mixture model is described by $p(x) = \sum_{j=1}^{J} p(x|G_j)P(G_j)$, where G_j are components, $P(G_j)$ is the prior probability and $p(x|G_j)$ is the probability that the data point belongs to component j.

When we start describing each component as a Gaussian: $p(x|G_j) \sim \mathcal{N}(\mu_j, \Sigma_j)$ we end up with a mixture of Gaussians (MoG). The IMoFA model was initially proposed to deal with issues of high-dimensionality that plagues the training of Gaussian mixtures. It uses a factor analysis (FA) model to keep the covariances of the Gaussian components small and reduces the number of parameters where $\Sigma_j = \Lambda_j \Lambda_j^T + \Psi$ where Σ is the covariance, Λ is the factor loading matrix and Ψ is the uniqueness matrix. The idea

algorithm **IMoFA**(train, validation)
  $[\mathbf{\Lambda}, \boldsymbol{\mu}, \Psi]$ ← train a 1-component, 1-factor model
  oldLikelihood ← -Infinity
  /*Likelihoods are calculated on validation set*/
  newLikelihood ← likelihood($\mathbf{\Lambda}, \boldsymbol{\mu}, \Psi$)
  while newLikelihood > oldLikelihood
  /*Perform a single split*/
    x ← Select a component for splitting
    $[\mathbf{\Lambda}_1, \boldsymbol{\mu}_1, \Psi_1, \pi_1]$ ← EM(split x).
    actionL(1) ← likelihood($\mathbf{\Lambda}_1, \boldsymbol{\mu}_1, \Psi_1, \pi_1$)
  /*Perform a single factor addition*/
    y ← Select a component to add a factor
    $[\mathbf{\Lambda}_2, \boldsymbol{\mu}_2, \Psi_2, \pi_2]$ ← EM(add factor to y).
    actionL(2) ← likelihood($\mathbf{\Lambda}_2, \boldsymbol{\mu}_2, \Psi_2, \pi_2$)
  /*Select the best action*/
    z ← max(action(L1),action(L2))
  /*Update the parameters*/
    $[\mathbf{\Lambda}, \boldsymbol{\mu}, \Psi, \pi]$ ← $[\mathbf{\Lambda}_z, \boldsymbol{\mu}_z, \Psi_z, \pi_z]$
    oldLikelihood ← newLikelihood
    newLikelihood ← likelihood($\mathbf{\Lambda}, \boldsymbol{\mu}, \Psi, \pi$)
  end
  return $[\mathbf{\Lambda}, \boldsymbol{\mu}, \Psi, \pi]$
end

Figure 6.9. The IMoFA Algorithm. Figure from [98].

is to start from a single Gaussian blob fit on the data cloud, and gradually increase complexity by either splitting the Gaussian components or by adding more factors to represent the component covariance in greater detail.

Expectation-Maximization (EM) algorithm [99] is widely used for training Gaussian mixture models (GMM). The EM algorithm is used to calculate maximum likelihood estimates of parameters in each iteration. It is used to update unknown mixture model parameters at each step. It requires that the number of components and the factors in each component be specified in advance to calculate the maximum likelihood estimates. IMoFA addresses this problem by adding components and factors iteratively by checking a validation set for the changes in the likelihood [100]. The design steps and the algorithm are given in Figure 6.9 where z shows the next action and π is the component probability.

The regular IMoFA starts from a single factor, single component mixture, and keeps adding new factors or components. IMoFA exploits fast heuristic metrics to find

one component for splitting and another for factor addition. Split and factor addition are tested on a validation set, isolated from the training set which is used for parameter calculation. An action that leads to the maximum increase in the validation likelihood is chosen, if any. If there are no improvements to be made, then the algorithm ends.

6.3. Chromatic and achromatic colors

The choice of color representation for color clustering is important, as different color spaces define different proximity relations between colors. We have chosen a color space that aligns well with the human perceptual system.

Humans sense color through receptors in their eyes. These receptors provide a source for the understanding of colors as we know it. Colors that simulate all three receptors very closely are perceived as black, gray, or white, i.e., achromatic colors [101]. Chromatic colors, which are the opposite of achromatic colors, simulate receptors in any other way. These colors can be represented in the hue axis of the Hue - Saturation - Intensity (HSI) color space. However, achromatic colors do not have a meaningful color representation.

$$
\begin{aligned}
H &= \begin{cases} \theta & \text{if} \quad B \leq G \\ 360 - \theta & \text{if} \quad B > G \end{cases} \\
\theta &= \cos^{-1}\left(\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]}\right) \\
S &= 1 - \frac{3}{R+G+B}[min(R,G,B)] \\
I &= \frac{R+G+B}{3}
\end{aligned}
\tag{6.9}
$$

HSI color space definitions are given in Equation (6.9) [102]. This color space will be used to represent colors of the paintings calculated in Section 6.3.1. This

representation is either based on hue (chromatic colors) or intensity (achromatic colors). For saturation values close to 0 ($R \approx G \approx B$), hue is meaningless, and undefined in the case of $s = 0$.

The saturation threshold is vital in deciding whether a color is chromatic. However, it is insufficient by itself. This is due to large quantization errors on low-intensity images. Saturation calculation uses the ratio of average and minimum intensity from red green and blue channels. However, when values are very low, for example, the case with Figure 6.10 where although average RGB intensity ($i = 23$) is quite close to blue channel value ($b = 20$) from the quantization point of view (± 1.5 of quanta); however saturation has a rather large value of $\frac{23-20}{23} = 13\%$.

Using saturation and intensity thresholds, one can filter pixels that are colorful or colorless. A threshold value for saturation at $s < 5\%$ is chosen to decide at which point hue represent color, before which saturation values are caused by the fluctuations of the red, green, and blue representation of the painting. Average color intensity is used as another threshold to determine whether a painting has a proper color or if it has very low-intensity values, susceptible to the fluctuations. Similar to the saturation threshold, the intensity threshold is chosen empirically as $i < 10\%$. These values can be adjusted freely on the provided interface, explained in Section 6.5.

6.3.1. Clothing color quantization

For the purpose of this thesis, the aim is to find unique colors that are present in the sitter's garb. However, due to digital representations of the color, different pixels that are perceived the same, do have different values. This is caused by shades of the colors and minimal changes in the pixel representations. Therefore, we need to assess what are these unique colors in a given painting. To address this issue, we group pixels with similar colors together to decide on unique colors and their relation to the entire region of interest (weights). For visualization purposes, we only use the color with the highest regional area to represent each painting. This representation is called the **dominant color** from now on. We have experimented with several approaches for

Figure 6.10. The *Zelfportret by Martin Mytens, 1703.* IMoFA dominant color is computed with HSI $(49, 15\%, 8\%)$ from RGB values $(22, 21, 17)$.

selecting a dominant color from a cloud of pixels.

Color quantization is based on HSI color space defined in Section 6.3. Due to its cyclic nature, Hue degrees are represented by their unit circle coordinates; i.e. $H_1 = \sin(H)$ and $H_2 = \cos(H)$. This representation puts $0 \leq H_1 \leq 1$ and $0 \leq H_2 \leq 1$. Saturation and Intensity values inherently are defined in the same range. Color quantization is performed in this domain, where the pixel value is represented in 4-D space as $[H_1, H_2, S_I]$. Quantization results can be easily reverted back to HSI space by calculating $H = atan2(H_1, H_2)$.

Figure 6.11 shows that a unicolor painting would have little issue with fragmentation where shades of the same color are detected as unique colors by themselves. Therefore, different k values have little influence on the dominant color, except for changing the shade of it. However, this fragmentation of the color has a more significant effect on Figure 6.12. High k values fragment the black into a few clusters, and displays brown as the dominant color, although black seems to have the majority of the clothing area. This fragmentation is very limited when an intelligent tool like IMoFA is in use, due to its power in evaluating whether another cluster is needed or

(a) k-means ($k = 2$)

(b) k-means ($k = 5$)

(c) k-means ($k = 8$)

(d) IMoFA

Figure 6.11. On unicolor clothing seen in the *Portret van een oude man (1639)*, the complexity of the algorithms does no harm and all results are very similar.

(a) k-means ($k = 2$)

(b) k-means ($k = 5$)

(c) k-means ($k = 8$)

(d) IMoFA

Figure 6.12. On multicolor clothing seen in the *Hortensia del Prado (gest 1627). Echtgenote van Jean Fourmenois*, k parameter becomes very important. High values can fragment the dominant color and change the perception, as seen in $k = 8$.

(a) k-means ($k = 2$)

(b) k-means ($k = 5$)

(c) k-means ($k = 8$)

(d) IMoFA

Figure 6.13. On many color clothing seen in the *Portret van Gustav II Adolf (1594-1632), koning van Zweden*, complexity can be deceiving.

k-means captures darker shades of the Swedish blue, whereas IMoFA delivers a lighter tint.

not. Unfortunately, the question - What is the color of the clothing of the sitter? is not an easy one. Figure 6.13 shows that IMoFA found the gold color as the dominant. On this painting, Swedish royal blue is undoubtedly the clothing color. However, this is influenced by the segmentation results seen in Figure 6.15 and the area of the blue is calculated as 22% in comparison to the gold which is 24%. When only the segmented portion is considered, this result makes sense in that the area under the tulle is also gold, and hence the total area of gold is higher than the blue garment or white tulle.

A hundred paintings from both perceived sex are hand-picked for their dominant colors. These samples are picked pseudo-randomly to span through 16^{th} to the 20^{th} century. The performance of the different quantization methods is measured by calculating the distance between the annotation and the computed values. Two distance metrics are used: ΔH and ΔI for chromatic and achromatic colors respectively. ΔH is calculated with distance formula for polar coordinates where $\theta = H$ and $r = 1$ shown in Equation (6.10). Due to ΔH range, $0 \leq \Delta H \leq 2$, $\Delta H/2$ is used for the distance comparison. Finally, ΔI is measured as the absolute value of the difference, i.e., $\Delta I(I_x, I_y) = \|I_x - I_y\|$.

$$\begin{aligned}\Delta H(H_x, H_y) &= \sqrt{(\sin H_x - \sin H_y)^2 + (\cos H_x - \cos H_y)^2} \\ &= \sqrt{\sin^2 H_x + \cos^2 H_x + \sin^2 H_y + \cos^2 H_y - \cos H_x \cos H_y - \sin H_x \sin H_y} \\ &= \sqrt{2 - 2\cos(H_x - H_y)}\end{aligned} \tag{6.10}$$

The performances of the quantization algorithms are illustrated in Table 6.1. k-means performance depends on the k selection as expected. $k = 8$ provides the smallest distance and hence the best results for ΔH. ΔI differences are smaller and more volatile in comparison. As seen in Figure 6.14, although $k = 2$ seems to have the least average distance, it has the highest distance on the first 45 annotated paintings and the low average value is due to low maximum value. $k = 8$ still performs well on the first half of the paintings, where distances are more likely to quantization methods

Table 6.1. Quantization algorithm performances are measured in distances from the hand-annotation. Lower values are better. The second and the third columns represent $\Delta H/2$ and ΔI distances for all the annotated paintings.

Distance	$\Delta H/2$	ΔI
k-means ($k = 2$)	17.8%	10.7
k-means ($k = 5$)	17.0%	9.6
k-means ($k = 8$)	14.1%	8.3
k-means ($k = 11$)	14.6%	8.7
k-means ($k = 14$)	15.6%	9.3
IMoFA	19.0%	10.3

and less likely to segmentation or color selection differences.

Due to its performance over other methods, we have used k-means with $k = 8$ to generate clothing colors and their weights for the rest of the study. Other method results are still presented on the visual interface of the project as a reference, and depending upon feedback could be used for plots and trend analysis.

6.4. Clothing color visualization

The last stage of the processing pipeline is to represent paintings with their dominant colors and produce a visualization of all paintings along a time axis. The plot consists of all the available paintings, placed according to their descriptive feature values with respect to their intensities and years. These plots are either in Hue vs. time, or Intensity vs. time; depending on the color definitions from Section 6.3. In order to analyze the effects of the perceived sex (which is calculated earlier in Chapter 5) and its relation with respect to the clothing color and eras, four plots are prepared. One pair, consisting of both color descriptions with Hue values, and grayscale descriptions in Intensity values, is presented in Figure 6.16. Males and females are shown separately. Details on the access to the plots, construction and customization parameters and

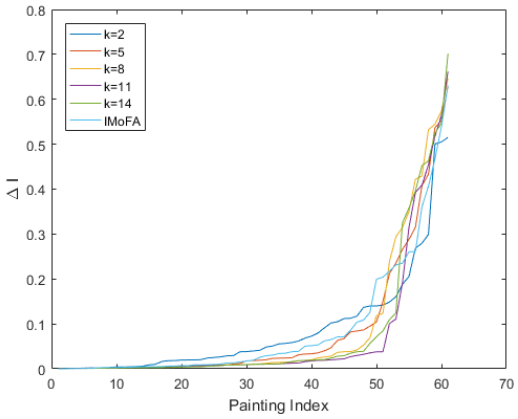


Figure 6.14. ΔI distances of the achromatic paintings. The distance values are ordered to show lower values with lower indices. Painting index reflects the index of paintings in this ascending distances and does not necessarily reflect the same paintings for different methods.

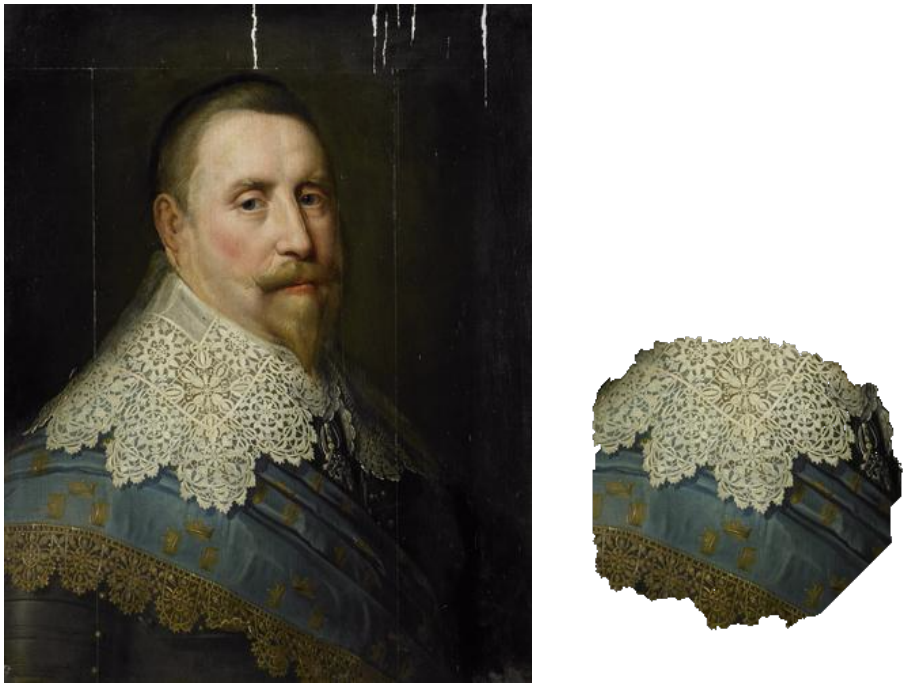


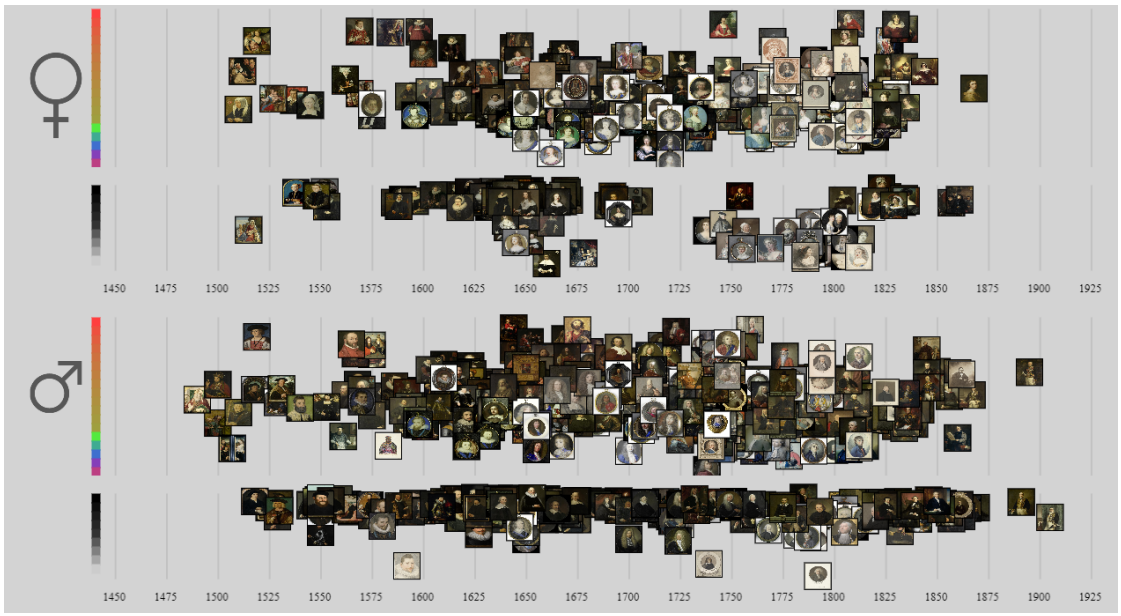
Figure 6.15. The *Portret van Gustav II Adolf (1594-1632), koning van Zweden* segmentation results

others, are explained in Section 6.5.

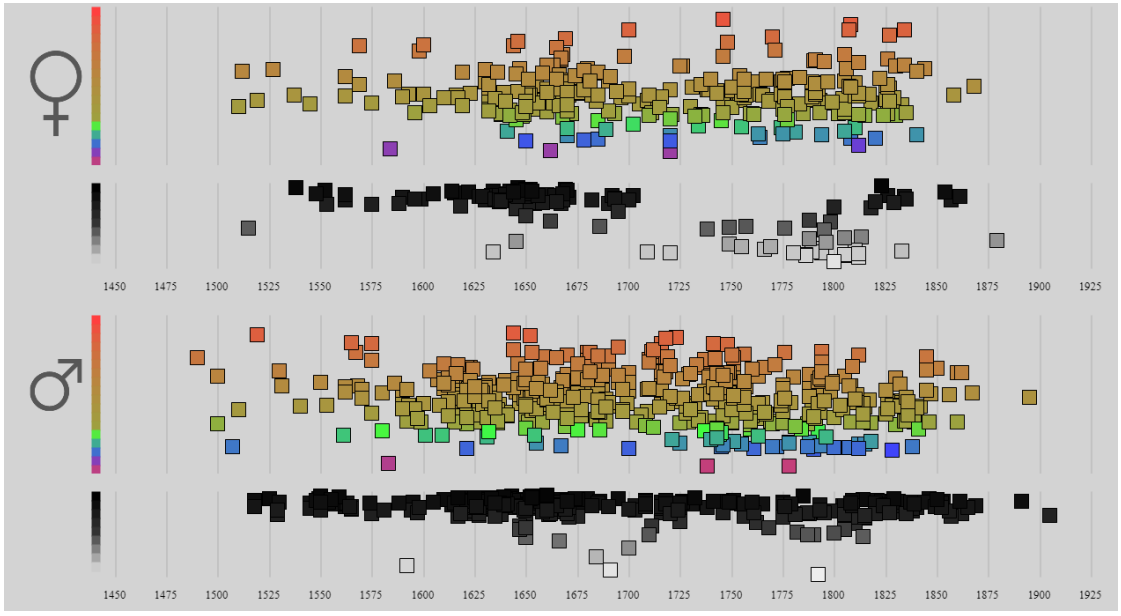
This plotting method is profoundly influenced by Manovich's work [27] and aims to display yet to be uncovered links between the perceived sex of the sitter and that of the era of the paintings. When focused on the data points, one can see more occurrences in lighter colors of the females on the value axis, whereas there are a large number of black (or dark gray) paintings with sitters perceived as male. In addition, orange and its shades are densely populated on the paintings. We believe this is due to several possible reasons. Most obviously, skin pixels that are caught mistakenly by the segmentation algorithm could give this effect. Moreover, the slight orange tilt can be observed in the paintings, which can be either digitization artifact or an artifact from the aging of the pigments in the painting. This tilt can be observed especially on the darker shades of gray, which appears dark-brown and turns the hue towards tints of orange.

Several alterations for the plots are introduced to highlight the possible links of clothing color trends. First one is to display the Hue and Intensity axes in nonlinear fashion. Majority of the chromatic paintings has Hue values in the range of $0^\circ - 60^\circ$ where 0° is the center of the red region, and 60° marks the yellow category, and hence the values in between are the shades of orange. We have altered our Hue axes to use half of the area on this region, which actually spans one-sixth of the total Hue range, and other half on the remaining Hue colors. A similar approach is taken on the two extremes of the Intensity axis. There are seldom examples of paintings on the shades of gray, and the majority of them are either very dark gray or black, or very light gray or white. Hence, the plot area in the Intensity axis is split into three parts: Dark area $[0-20]$ and light area $[80-100]$ individually span 40% of the plot area each, and gray area $[20-80]$, which is rarely used, is at the remaining 20% of the available area. This nonuniform scaling minimizes both the overlapping paintings in highly populated areas and the less used regions in the plot.

Another alternative is to use the term frequency–inverse document frequency (tf-idf) method by treating colors as words and paintings as documents [103]. Originally,



(a) Plot with painting thumbnails



(b) Plot with colors

Figure 6.16. The web interface on painting trends for males and females over time, with chromatic color thresholds $s > 5\%$ and $i > 10\%$.

this method is used to reflect the importance of the words in a document. In the case of this study, it is used to reflect the interest of the dominant colors in a painting. The algorithm computes a weight proportionally to the area of each color inside the painting, which is balanced by the frequency of the color across all paintings.

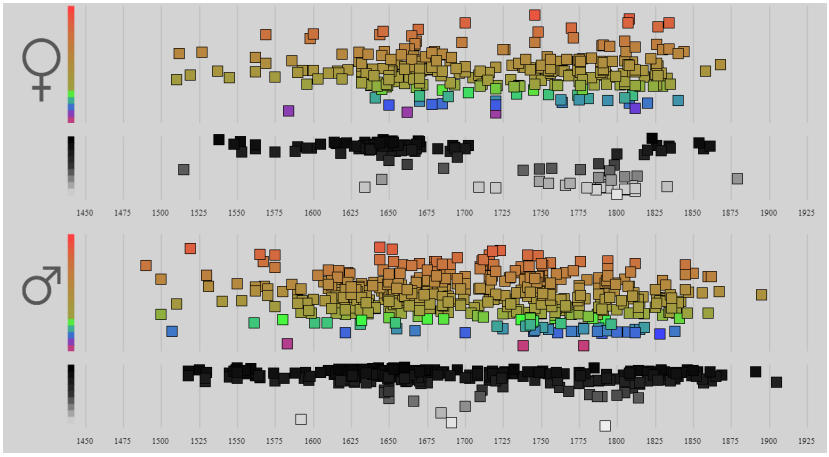
Finally, it is possible to visualize all color areas extracted from the painting simultaneously. This approach would show each painting multiple times in the graph, once for every dominant color. Figure 6.17 shows the trend graph with the methods mentioned previously. The tf-idf algorithm assigns green and blue colors heavy weights and punishes the shades of brown in Figure 6.17b in contrast to Figure 6.17a, and a wider color pool is displayed. Figure 6.17c depicts all the dominant colors extracted from all the paintings, which is a superset of both figures.

6.5. Interactive interface

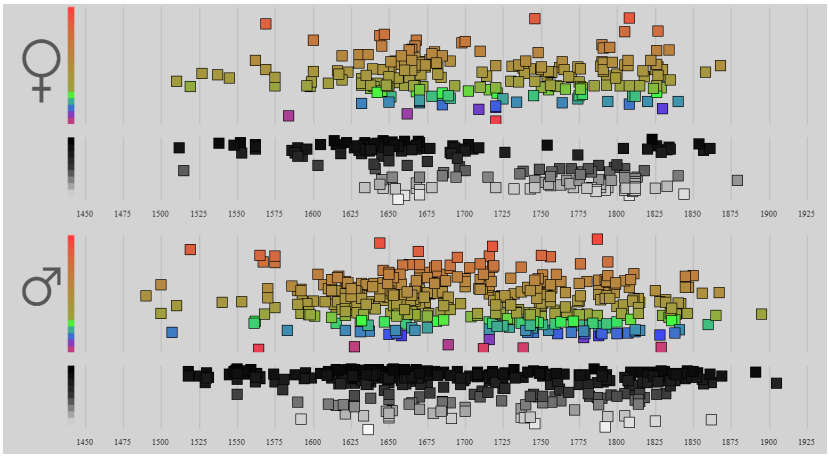
We have designed a web interface for the visualization of the results. In Section 6.5.1 we explain the hardware and the software dependencies and tools that are used to build up the web interface and provide links and repository for references. On the following Section 6.5.2, the interface for painting details and user interactions at this level are explained. Finally, Section 6.5.3 shows the interface to display the clothing color trend and to customize some of the results.

6.5.1. Web technologies

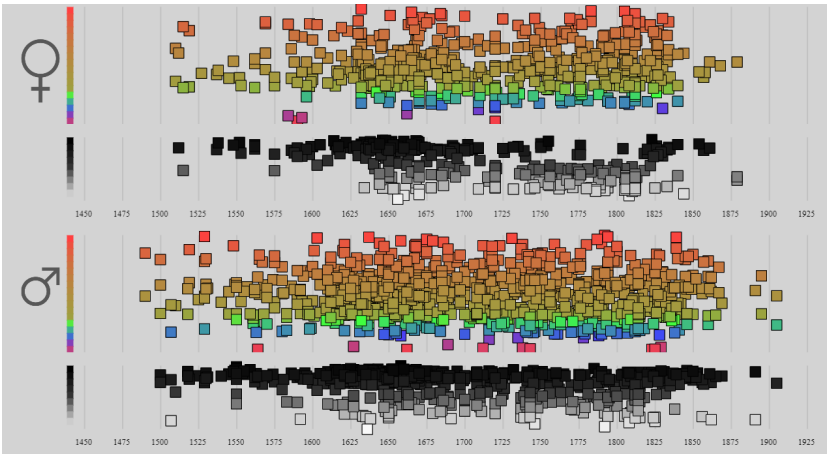
For hosting web interface, Raspberry Pi Model b is chosen as computation device due to its low energy consumption, cost, and stability. Debian operating system and Apache HTTP Server is used to serve Hypertext Markup Language (HTML) interface, which contains HTML, JavaScript and Cascading Style Sheets (CSS) links. All metadata is stored in JavaScript Object Notation (JSON) format and images in JPEG format.



(a) Trends with only the dominant color with the biggest area per painting.



(b) Trends after using tf-idf to weight the areas. The figure shows the biggest weighted area per painting.



(c) Trends using all the dominant colors per painting.

Figure 6.17. Trends for males and females with various dominant color methods.

This interface is produced and used for easy visualization of data, and global development purposes. The principle design decision was to provide visualization tools immediately to curators or researchers alike and give them the ability to provide immediate feedback.

Thanks to the availability of web browsers that are supported in almost any computational device that has user interaction, the web-based design is hugely popular. Therefore, such an interface has a significant advantage that it requires next to no preparation time for the users and can be accessed immediately when a new visualization is integrated. Moreover, there is a vast variety of visualization libraries available in JavaScript [104] that can lead to extensions without much extra effort for further analysis in demand.

Page layouts are prepared by HTML, their color and font styles are defined using CSS, and data acquisition, layout population, displays and actions (e.g., voting) are performed by JavaScript algorithms. JavaScript portion made heavy use of JQuery framework [105] and *FABRIC.JS* is used to create canvases to visualize graphs and plots. All these technologies are prepared as scripts and executed on the user's web browser.

Client-side algorithm for the pages starts with retrieving available painting list from the server. Through this list, the metadata corresponding to the requested painting and all relevant information can be obtained for the artwork, including title, segmentation result, extracted color result, link to painting image file, user votes among many others. Cumulatively collected information is then used to populate the rest of the page on the client-side.

However, to cast actual votes, an interactive technology between the server and the client is required. Apache HTTP Server's built-in Common Gateway Interface enables execution of selected binary files on the server computer (Raspberry Pi Model b in the context of this thesis), using Asynchronous JavaScript And XML (AJAX) calls. These calls let one execute a program on the server with client specified arguments and

retrieve its results.

Votes are kept on the metadata files in JSON format. That means, "casting a vote" is same as modifying JSON in a predictable manner. This action is already well defined and used heavily by HTTP protocol using JSON patch descriptions [106]. A JSON patch is structured as a JSON array of objects with possible operations: add, remove, replace, move, copy, and test. Casting a vote is then replacing an old value with its value incremented by one. Thanks to JSON patch notation, creating a patch that would cast a vote in this description on the client-side is relatively straightforward. However, for the server-side, we need to understand JSON patch and apply it to the correct JSON source file. This program is developed using Niel Lohmann's "JSON For Modern C++" implementation from Niels Lohmann [107] for JSON patch recognition and application, and C++17 standards for basic file read and write operations. It takes JSON source file name and patch contents as arguments, and applies the patch on the server and returns success or failure back to the client. Due to design decisions, this immediately affects the next visualization, as this is the new metadata to be used by the clients on new requests. Moreover, although currently unused, this JSON patch route enables a wild variety of possible changes to the metadata through the same interface where it is possible to display and modify this data immediately.

In conclusion, we have provided a feature complete, fully interactable environment for curators and researchers to visualize the results on a device of their choosing with minimal effort. All the source files can be accessed from the GitHub repository `https://github.com/CihanSari/ClothingWeb`.

6.5.2. Segmentation results

This interface allows the users to vote on the automatic segmentation results, both to help assess the quality of segmentation, and to filter out unwanted or controversial results from the further investigation. For example, Figure 6.18 has a body that has a different proportions than other examples encountered. In this example, due to very broad shoulders, most of the dark garment fell outside our area of interest and marked

Figure 6.18. The *Vermoedelijk posthuum portret van Rudolph van Buynou, drost van Stavoren en grietman van Gaasterland* has a different face to body proportions than the majority of the other examples and hence GrabCut segmentation method worked poorly.

with negative seeds. Therefore GrabCut avoided darker areas and picked the isolated collar for the segmentation. Similarly, Figure 6.19 has an unexpectedly long body in the portrait paintings, and this causes our area of interest definition to mark lower parts of the painting as negative seeds for GrabCut. Moreover, the positive seeds area coincides immediately with the white shirt the sitter is wearing, and due to lack of any positive seeds for the dark cloak on top, that portion is missed. A good segmentation captures the color distribution of the clothing as seen in Figure 6.20. In this section, we describe the software technology and approach used for the design of this interface.

6.5.3. Trends and graphs

Plots given in Figure 6.16 can be customized from these votes by setting minimum acceptable upvotes or a maximum number of downvotes. In addition to votes, trends can also be customized further by parameters given in Figure 6.21. One can hide the painting thumbnails, to have a better representation of the patterns and minimize the overlaps, or show them to have a feeling immediately upon the context of the

Figure 6.19. The *Pieter de Graeff (1638-1707), heer van Zuid-Polsbroek, Purmerland en Ilpendam. Schepen van Amsterdam* has a much larger clothing area than the rest of the paintings and therefore result in a low performance on GrabCut segmentation.

paintings: e.g., background or general palette information. Each thumbnail or square can be selected to preview the painting as seen in Figure 6.22 and its HSI values and contains a direct link to its voting page.

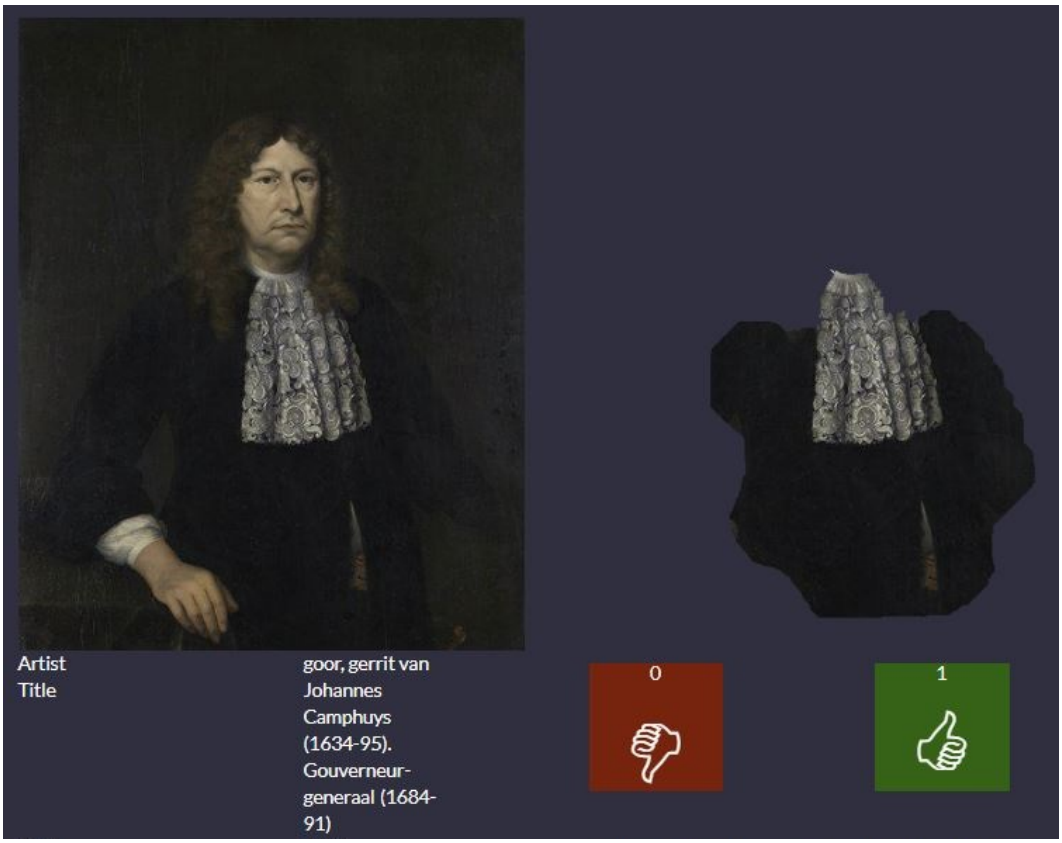


Figure 6.20. Web interface to assess segmentation.

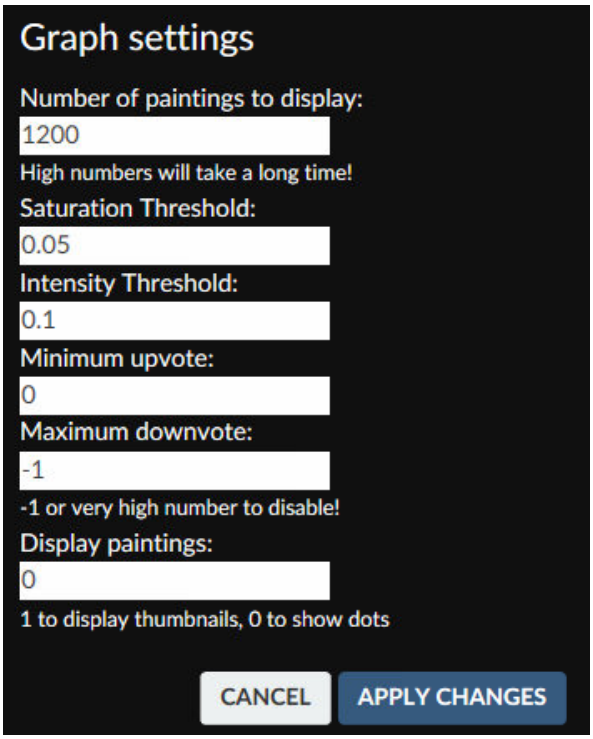


Figure 6.21. The web interface to configure and personalize color distribution.

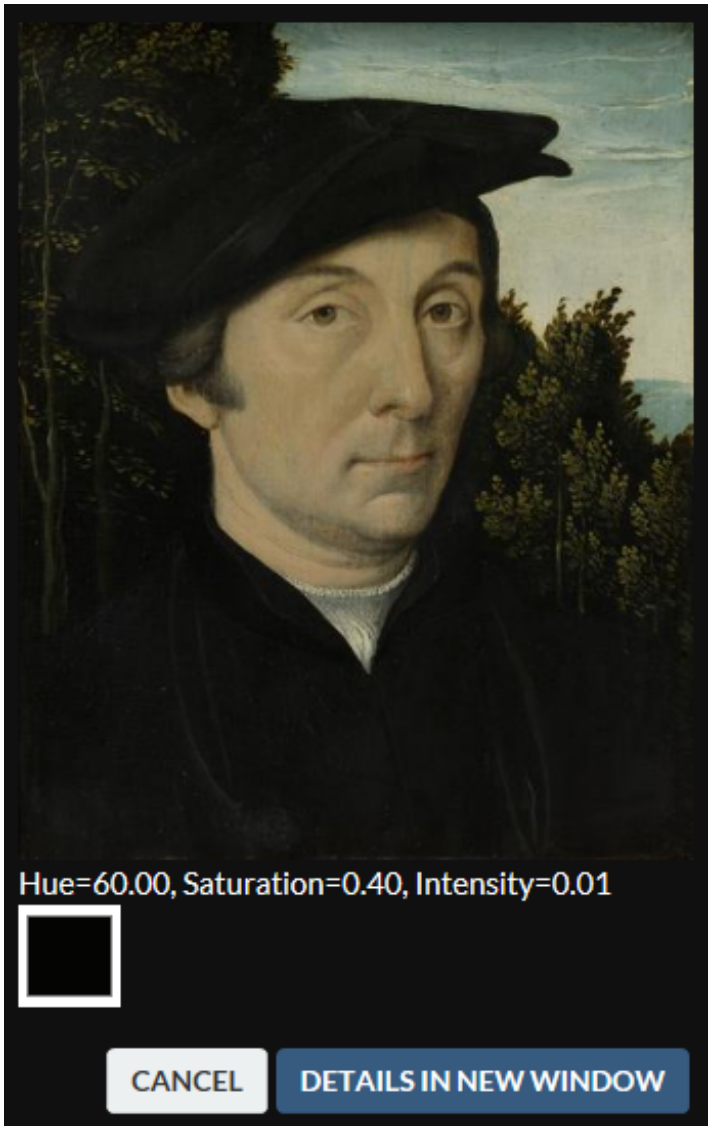


Figure 6.22. Paintings can be enlarged and reviewed in a new window.

7. CONCLUSIONS

In this thesis, we sought to introduce a system and an approach for automatically extracting the main colors of clothing worn by people in Western artworks. This thesis is part of a larger research program to translate Laqueur's (1990) thesis on how sex is understood in Western culture, and the transformation of this understanding in 19^{th} century to a Digital Humanities research question. Laqueur's theory proposes that prior to the 19^{th} century, the bodies of men and women are depicted physically more similar to one another, and described as two distinct types afterwards. We aimed to seek the projections of this theory in the representations of clothing and their colors. A distinction in the representation trends after the 19^{th} century could be a clue to support Laqueur's theory. While we could not observe such a break, our system is far from perfect, and faced numerous challenges, including correct segmentation of clothing, and representation of colors in a perceptually accurate way. Our main focus was to establish the groundwork to put the system in place. We believe that the system we have implemented and the challenges we have discussed will encourage further research in this area.

The digitized artworks of Rijksmuseum Amsterdam are used as the primary drive of the study, as the artworks are the visual sources connecting us to the history of art and humanities. Every period and location has certain dominant color associations and symbolism. To investigate hundreds of thousands of paintings in a single sweep requires algorithmic solutions and automatic analysis tools. Another objective of performing an analysis on the usage of color for male and females along the centuries was to develop and deploy tools for establishing semantic associations of colors for each particular period or study.

In order to separate the portraits painting within the available dataset of artifacts in Rijksmuseum, a face detection algorithm introduced by Viola and Jones in 2001 is applied. This method, based on Haar feature-based cascade classifiers, uses a series of fast and weak classifiers to enable a swift face detection. We realized that on paintings,

this method did not produce perfect results, and there is room for improvement.

Afterwards, facial landmarks – different parts of the face such as nose, mouth, eyes – are detected, and this information is appended to the metadata available from the paintings. The facial landmark points are used to align the faces from in the wild training photos collected from the web and the artworks published by the Rijksmuseum. Moreover, the same landmarks are used to locate clothing masks in the paintings for cloth segmentation. These landmarks are obtained using a Supervised Descent Method (SDM) that initiates landmarks based on the Viola and Jones algorithm results, and then iteratively moves them to their locations.

Prior to alignment of the faces, a global alignment template should be produced. These golden landmark locations are acquired using Generalized Procrustes Analysis on all the training sets, including 10k US Adult Faces, collected IMDb dataset, and LFW. Subsequently, all detected faces in the training set and also in the portraits are aligned to this golden landmark locations using Procrustes analysis.

LBP and VGG feature extractors, SVM and RDF classifiers and samples of training images are used to train perceived sex classifiers. Furthermore, training photographs are recomposed using a paintings style transfer to improve the performance. Through these experiments, 80.5% perceived sex classification performance on artworks is achieved using only natural faces, without any paintings on the training sets. By comparison to between the training sets, we hypothesize a possible indirect link between the poses of the actors and actress of the 21st century and the sitters of the Western art from the performances.

Another essential part of the process is to segment the clothing in the paintings. The portrait paintings that are entirely focused on the sitter's face still have a lot of background noise that disrupts the color representation of the artworks if no clothing mask is used. The grab cut algorithm is used to find pixels representing the area of interest - clothing in this case. Consequently, the segmented clothes are used for clustering and to predict dominant color or palette by applying k-means or IMoFA

approaches.

The secondary aim of this study was to provide a single visualization of our result within an interactive web interface, where the viewer can observe the full scale of the outcome and search for the color patterns through different eras. We take one step further and engage the viewer in voting on the result of the segmentation and dominant color selection, which helps us not only in assessing our contributions, but also provides further annotations for improving the automated results.

This interface is produced and used for easy visualization of data and global development purposes. Principle design decision was to provide visualization tools immediately to curators or researchers alike and give them the ability to provide immediate feedback.

Thanks to the availability of web browsers that already exist in almost any computational device that supports user interaction, e.g., personal computers, laptops, tablets, cellphones web-based design is hugely popular. Therefore, such an interface has a significant benefit that it requires next to no preparation time for the users and can be accessed immediately when a new thought or visualization idea presents itself. Moreover, there is a vast variety of visualization libraries available in JavaScript that can lead to extensions without much extra effort for further analysis in demand.

Page layouts are prepared by HTML, their color and font styles are defined using CSS and data acquisition, layout population, displays and actions (e.g., voting) are performed by JavaScript algorithms. JavaScript portion made heavy use of JQuery framework [105] and *FABRIC.JS* is used to create canvases to visualize graphs and plots. All these technologies are prepared as scripts and executed on the user's web browser.

Client-side algorithm for the pages starts with retrieving available painting list from the server. Through this list, the metadata corresponding to the requested painting and all relevant information can be obtained for the artwork, including title, seg-

mentation result, extracted color result, link to painting image file, user votes among many others. Cumulatively collected information is then used to populate the rest of the page on the client-side. Eventually, we have provided a feature complete, fully interactive environment for curators and researchers to visualize the results on a device of their choosing with minimal effort.

One of the design philosophy of this study was its scalability. Thanks to the increased popularity of open-art, it can be extended significantly by introducing more databases alongside Rijksmuseum, for example, drawing on the Europeana [108], Metropolitan Museum of Art [109], DeviantArt [110], among others.

In the scope of this study, the focus is on the clothing of the sitter; however, same logic and very similar approach can be applied to extract colors of different objects. For example, one could use color distribution analysis on the plates and find the symbolic links between colors over time on tableware. Another possible study could focus on clothing identification and filter paintings by military uniforms, casual or formal wear and analyze the distribution over time with these categories, instead of perceived sex.

We used style transfer for improving the approach to work with paintings. Style transfer can be extended with different styles. Combination of such styles could improve the performance. Moreover, each test sample can be used as a target, and on the calculation time, a new classifier could be trained from a number of training samples with this new style painting. With this method, we could transfer training samples to each test sample's domain and have a unique classifier. This extended method could be performed sparingly, e.g., on samples with high uncertainty.

We hope that we have opened a door with this work, to encourage further research and provide tools for both fellow researchers and curious minds to have access to some tools to work on big data. For the future, we plan to extract background and skin color alongside clothing and use this information to locate outstanding paintings and estimate trends. This work accomplished to develop and fine-tune the algorithm on a relatively small dataset, and another plan is to exploit the availability of other datasets to focus on particular eras for a more refined look.

REFERENCES

1. Berns, R. S., "The science of digitizing paintings for color-accurate image archives: a review" *Journal of imaging science and technology*, Vol. 45, No. 4, pp. 305–325, 2001.

2. Stork, D. G. and J. Coddington, "Computer image analysis in the study of art" *Proc. SPIE*, Vol. 6810, 2008.

3. Barni, M., A. Pelagotti and A. Piva, "Image processing for the analysis and conservation of paintings: opportunities and challenges" *IEEE Signal processing magazine*, Vol. 22, No. 5, pp. 141–144, 2005.

4. Schmidt, R., "Consciousness and foreign language learning: A tutorial on the role of attention and awareness in learning" *Attention and awareness in foreign language learning*, Vol. 9, pp. 1–63, 1995.

5. Paoletti, J. B., "Clothing and gender in America: Children's fashions, 1890-1920" *Signs*, Vol. 13, No. 1, pp. 136–143, 1987.

6. Schmidt, C. M., M. S. Walton and K. Trentelman, "Characterization of lapis lazuli pigments using a multitechnique analytical approach: implications for identification and geological provenancing" *Analytical chemistry*, Vol. 81, No. 20, pp. 8513–8518, 2009.

7. Gage, J., "Color in Western Art: An Issue?" *The Art Bulletin*, Vol. 72, No. 4, pp. 518–541, 1990.

8. Pinker, S., "Visual cognition: An introduction" *Cognition*, Vol. 18, No. 1, pp. 1–63, 1984.

9. Barni, M., F. Bartolini and V. Cappellini, "Image processing for virtual restora-

tion of artworks" *IEEE multimedia*, Vol. 7, No. 2, pp. 34–37, 2000.

10. Sorabella, Jean, 2007, "Portraiture in Renaissance and Baroque Europe", `http://www.metmuseum.org/toah/hd/port/hd_port.htm`, accessed at January 2018.

11. Mensink, T. and J. van Gemert, "The Rijksmuseum challenge: Museum-centered visual recognition" *Proceedings of International Conference on Multimedia Retrieval*, p. 451, ACM, 2014.

12. Sarı, Cihan, 2018, "Web interface of color tracing", `http://www.cihansari.com`, accessed at January 2018.

13. Sarı, C., A. A. Akdağ Salah and A. A. Salah, "Tracing the Colors of Clothing in Paintings with Image Analysis" *Proc. Digital Humanities*, pp. 577–580, McGill University & Université de Montréal, 2017.

14. Sarı, C., A. A. Akdağ Salah and A. A. Salah, "Tracing the Colors of Clothing in Paintings with Image Analysis" *Digital Scholarship in the Humanities*, submitted for publication.

15. Crowley, E. J. and A. Zisserman, "In Search of Art" *Workshop on Computer Vision for Art Analysis, ECCV*, 2014.

16. Haig, D., "The inexorable rise of gender and the decline of sex: Social change in academic titles, 1945–2001" *Archives of sexual behavior*, Vol. 33, No. 2, pp. 87–96, 2004.

17. Stork, D. G., "Computer vision and computer graphics analysis of paintings and drawings: An introduction to the literature" *International Conference on Computer Analysis of Images and Patterns*, pp. 9–24, Springer, 2009.

18. Martinez, K., J. Cupitt, D. Saunders and R. Pillay, "Ten years of art imaging research" *Proceedings of the IEEE*, Vol. 90, No. 1, pp. 28–41, 2002.

19. Cappellini, V., M. Barni, M. Corsini, A. De Rosa and A. Piva, "ArtShop: an art-oriented image-processing tool for cultural heritage applications" *Computer Animation and Virtual Worlds*, Vol. 14, No. 3, pp. 149–158, 2003.

20. Klockenkämper, R., A. Von Bohlen and L. Moens, "Analysis of pigments and inks on oil paintings and historical manuscripts using total reflection x-ray fluorescence spectrometry" *X-Ray Spectrometry*, Vol. 29, No. 1, pp. 119–129, 2000.

21. Balas, C., V. Papadakis, N. Papadakis, A. Papadakis, E. Vazgiouraki and G. Themelis, "A novel hyper-spectral imaging apparatus for the non-destructive analysis of objects of artistic and historic value" *Journal of Cultural Heritage*, Vol. 4, pp. 330–337, 2003.

22. Rosi, F., C. Miliani, R. Braun, R. Harig, D. Sali, B. G. Brunetti and A. Sgamellotti, "Noninvasive analysis of paintings by mid-infrared hyperspectral imaging" *Angewandte Chemie International Edition*, Vol. 52, No. 20, pp. 5258–5261, 2013.

23. Thurrowgood, D., D. Paterson, M. D. De Jonge, R. Kirkham, S. Thurrowgood and D. L. Howard, "A hidden portrait by edgar degas" *Scientific reports*, Vol. 6, p. 29594, 2016.

24. Johnson, C. R., E. Hendriks, I. J. Berezhnoy, E. Brevdo, S. M. Hughes, I. Daubechies, J. Li, E. Postma and J. Z. Wang, "Image processing for artist identification" *IEEE Signal Processing Magazine*, Vol. 25, No. 4, 2008.

25. Shamir, L., "Computer analysis reveals similarities between the artistic styles of Van Gogh and Pollock" *Leonardo*, Vol. 45, No. 2, pp. 149–154, 2012.

26. Manovich, L., "The science of culture? Social computing, digital humanities and cultural analytics" *CA: Journal of Cultural Analytics*, Vol. 1, No. 1, 2016.

27. Manovich, L., "What is visualisation?" *Visual Studies*, Vol. 26, No. 1, pp. 36–49,

2011.

28. Akdağ Salah, A., L. Manovich, A. A. Salah and J. Chow, "Combining cultural analytics and networks analysis: Studying a social network site with user-generated content" *Journal of Broadcasting & Electronic Media*, Vol. 57, No. 3, pp. 409–426, 2013.

29. Miller, S. J., *Metadata for digital collections: a how-to-do-it manual*, Neal-Schuman Publishers New York, NY, 2011.

30. Metropolitan Museum of Art, 2018, "Metropolitan Museum of Art Online collection", `https://metmuseum.org/art/collection`, accessed at January 2018.

31. Fine Arts Museums of San Francisco, 2018, "The Thinker", `https://deyoung.famsf.org/collections`, accessed at January 2018.

32. Yanulevskaya, V., J. Uijlings, E. Bruni, A. Sartori, E. Zamboni, F. Bacci, D. Melcher and N. Sebe, "In the eye of the beholder: employing statistical analysis and eye tracking for analyzing abstract paintings" *Proceedings of the 20th ACM international conference on Multimedia*, pp. 349–358, ACM, 2012.

33. Sartori, A., B. Şenyazar, A. A. Akdağ Salah, A. A. Salah and N. Sebe, "Emotions in Abstract Art: Does Texture Matter?" *International Conference on Image Analysis and Processing*, pp. 671–682, Springer, 2015.

34. Ginosar, S., D. Haas, T. Brown and J. Malik, "Detecting people in cubist art" *Workshop at the European Conference on Computer Vision*, pp. 101–116, Springer, 2014.

35. DeviantArt, 2018, "DeviantArt Community", `http://www.deviantart.com`, accessed at January 2018.

36. Isikdogan, F., İ. Adıyaman, A. A. Akdağ Salah and A. A. Salah, "A New Database and Protocol for Image Reuse Detection" *European Conference on Computer Vi-*

sion, pp. 903–916, Springer, 2016.

37. Rijksmuseum Amsterdam, 2017, "Rijksmuseum Database", `https://www.rijksmuseum.nl/en/rijksstudio`, accessed at December 2017.

38. Dijkshoorn, C., L. Aroyo, G. Schreiber, J. Wielemaker and L. Jongma, "Using Linked Data to Diversify Search Results a Case Study in Cultural Heritage." *EKAW*, pp. 109–120, Springer, 2014.

39. Isaac, A. and B. Haslhofer, "Europeana linked open data–data. europeana. eu" *Semantic Web*, Vol. 4, No. 3, pp. 291–297, 2013.

40. Maaten, L. v. d. and G. Hinton, "Visualizing data using t-SNE" *Journal of Machine Learning Research*, Vol. 9, No. Nov, pp. 2579–2605, 2008.

41. Vedaldi, A. and K. Lenc, "MatConvNet – Convolutional Neural Networks for MATLAB" *MatConvNet – Convolutional Neural Networks for MATLAB*, 2015.

42. Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge" *International Journal of Computer Vision (IJCV)*, Vol. 115, No. 3, pp. 211–252, 2015.

43. Chatfield, K., K. Simonyan, A. Vedaldi and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets" *British Machine Vision Conference*, 2014.

44. Alphabet Inc., 2018, "Google Images", `https://images.google.com/`, accessed at January 2018.

45. Sharif Razavian, A., H. Azizpour, J. Sullivan and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition" *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

46. Chang, C.-C. and C.-J. Lin, "LIBSVM: A library for support vector machines" *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, pp. 27:1–27:27, 2011, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

47. Viola, P. and M. Jones, "Rapid object detection using a boosted cascade of simple features" *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 1, pp. I–I, IEEE, 2001.

48. Salah, A. A., H. Cinar, L. Akarun and B. Sankur, "Robust facial landmarking for registration" *Annales des Télécommunications*, Vol. 62, pp. 83–108, Springer, 2007.

49. Gower, J. C., "Generalized procrustes analysis" *Psychometrika*, Vol. 40, No. 1, pp. 33–51, 1975.

50. Cottrell, G. W. and J. Metcalfe, "EMPATH: Face, emotion, and gender recognition using holons" *Advances in neural information processing systems*, pp. 564–571, 1991.

51. Wood, W. and S. J. Karten, "Sex differences in interaction style as a product of perceived sex differences in competence." *Journal of personality and social psychology*, Vol. 50, No. 2, p. 341, 1986.

52. Barclay, C. D., J. E. Cutting and L. T. Kozlowski, "Temporal and spatial factors in gait perception that influence gender recognition" *Attention, Perception, & Psychophysics*, Vol. 23, No. 2, pp. 145–152, 1978.

53. Cao, L., M. Dikmen, Y. Fu and T. S. Huang, "Gender recognition from body" *Proceedings of the 16th ACM international conference on Multimedia*, pp. 725–728, ACM, 2008.

54. Savic, I., H. Berglund, B. Gulyas and P. Roland, "Smelling of odorous sex hormone-like compounds causes sex-differentiated hypothalamic activations in humans" *Neuron*, Vol. 31, No. 4, pp. 661–668, 2001.

55. Burton, A. M., V. Bruce and N. Dench, "What's the difference between men and women? Evidence from facial measurement" *Perception*, Vol. 22, No. 2, pp. 153–176, 1993.

56. Cellerino, A., D. Borghetti and F. Sartucci, "Sex differences in face gender recognition in humans" *Brain research bulletin*, Vol. 63, No. 6, pp. 443–449, 2004.

57. Shan, C., "Learning local binary patterns for gender classification on real-world face images" *Pattern Recognition Letters*, Vol. 33, No. 4, pp. 431–437, 2012.

58. Kayım, G., C. Sarı and C. B. Akgül, "Facial feature selection for gender recognition based on random decision forests" *21st IEEE Signal Processing and Communications Applications Conference (SIU)*, 2013.

59. Jia, S. and N. Cristianini, "Learning to classify gender from four million images" *Pattern Recognition Letters*, Vol. 58, pp. 35–41, 2015.

60. Parkhi, O. M., A. Vedaldi and A. Zisserman, "Deep Face Recognition" *British Machine Vision Conference*, 2015.

61. Ojala, T., M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns" *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 24, No. 7, pp. 971–987, 2002.

62. Ahonen, T., A. Hadid and M. Pietikainen, "Face description with local binary patterns: Application to face recognition" *IEEE transactions on pattern analysis and machine intelligence*, Vol. 28, No. 12, pp. 2037–2041, 2006.

63. Liao, S., X. Zhu, Z. Lei, L. Zhang and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition" *International Conference on Biometrics*, pp.

828–837, Springer, 2007.

64. Mu, Y., S. Yan, Y. Liu, T. Huang and B. Zhou, "Discriminative local binary patterns for human detection in personal album" *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

65. Shan, C., S. Gong and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study" *Image and Vision Computing*, Vol. 27, No. 6, pp. 803–816, 2009.

66. Ng, C. B., Y. H. Tay and B.-M. Goi, "Recognizing human gender in computer vision: a survey" *PRICAI 2012: Trends in Artificial Intelligence*, pp. 335–346, Springer, 2012.

67. Bainbridge, W. A., P. Isola and A. Oliva, "The intrinsic memorability of face photographs." *Journal of Experimental Psychology: General*, Vol. 142, No. 4, p. 1323, 2013.

68. Huang, G. B., M. Ramesh, T. Berg and E. Learned-Miller, *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*, Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

69. genderize.io, 2018, "Determine the gender of a first name", `https://genderize.io/`, accessed at January 2018.

70. Khosla, A., W. A. Bainbridge, A. Torralba and A. Oliva, "Modifying the memorability of face photographs" *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3200–3207, 2013.

71. Xiong, X. and F. De la Torre, "Supervised Descent Method and its Applications to Face Alignment" *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

72. Barkan, O., J. Weill, L. Wolf and H. Aronowitz, "Fast high dimensional vector

multiplication face recognition" *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1960–1967, 2013.

73. Suykens, J. A. and J. Vandewalle, "Least squares support vector machine classifiers" *Neural processing letters*, Vol. 9, No. 3, pp. 293–300, 1999.

74. Hearst, M. A., S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines" *IEEE Intelligent Systems and their applications*, Vol. 13, No. 4, pp. 18–28, 1998.

75. Chang, Y.-W., C.-J. Hsieh, K.-W. Chang, M. Ringgaard and C.-J. Lin, "Training and testing low-degree polynomial data mappings via linear SVM" *Journal of Machine Learning Research*, Vol. 11, No. Apr, pp. 1471–1490, 2010.

76. Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA data mining software: an update" *ACM SIGKDD explorations newsletter*, Vol. 11, No. 1, pp. 10–18, 2009.

77. Breiman, L., "Random forests" *Machine learning*, Vol. 45, No. 1, pp. 5–32, 2001.

78. Breiman, L., J. Friedman, C. J. Stone and R. A. Olshen, *Classification and regression trees*, CRC press, 1984.

79. Breiman, L., "Bagging predictors" *Machine learning*, Vol. 24, No. 2, pp. 123–140, 1996.

80. Mathias, M., R. Benenson, M. Pedersoli and L. Van Gool, "Face detection without bells and whistles" *Computer Vision–ECCV 2014*, pp. 720–735, Springer, 2014.

81. Zhao, M. and S.-C. Zhu, "Portrait Painting Using Active Templates" *NPAR '11: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*, pp. 117–124, ACM, New York, NY, USA, 2011.

82. Gatys, L. A., A. S. Ecker and M. Bethge, "Image style transfer using convolutional neural networks" *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

83. Dumoulin, V., J. Shlens and M. Kudlur, "A Learned Representation For Artistic Style" *CoRR*, Vol. abs/1610.07629, 2016, `http://arxiv.org/abs/1610.07629`.

84. Gatys, L. A., A. S. Ecker and M. Bethge, "A neural algorithm of artistic style" *arXiv preprint arXiv:1508.06576*, 2015.

85. Everingham, M., L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, "The pascal visual object classes (voc) challenge" *International journal of computer vision*, Vol. 88, No. 2, pp. 303–338, 2010.

86. Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft coco: Common objects in context" *European conference on computer vision*, pp. 740–755, Springer, 2014.

87. Kalantidis, Y., L. Kennedy and L.-J. Li, "Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos" *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pp. 105–112, ACM, 2013.

88. Gallagher, A. C. and T. Chen, "Clothing cosegmentation for recognizing people" *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

89. Rother, C., V. Kolmogorov and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts" *ACM transactions on graphics (TOG)*, Vol. 23, pp. 309–314, ACM, 2004.

90. Boykov, Y. Y. and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images" *Computer Vision, 2001. ICCV*

2001. Proceedings. Eighth IEEE International Conference on, Vol. 1, pp. 105–112, IEEE, 2001.

91. Blake, A., C. Rother, M. Brown, P. Perez and P. Torr, "Interactive image segmentation using an adaptive GMMRF model" *Computer Vision-ECCV 2004*, pp. 428–441, 2004.

92. Bradski, G. and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*, " O'Reilly Media, Inc.", 2008.

93. Cowen, A. S., M. M. Chun and B. A. Kuhl, "Neural portraits of perception: reconstructing face images from evoked brain activity" *Neuroimage*, Vol. 94, pp. 12–22, 2014.

94. Güçlütürk, Y., U. Güçlü, R. van Lier and M. A. van Gerven, "Convolutional sketch inversion" *European Conference on Computer Vision*, pp. 810–824, Springer, 2016.

95. Varol, G., J. Romero, X. Martin, N. Mahmood, M. Black, I. Laptev and C. Schmid, "Learning from Synthetic Humans" *arXiv preprint arXiv:1701.01370*, 2017.

96. Newell, A., K. Yang and J. Deng, "Stacked hourglass networks for human pose estimation" *European Conference on Computer Vision*, pp. 483–499, Springer, 2016.

97. Hartigan, J. A. and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm" *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 28, No. 1, pp. 100–108, 1979.

98. Salah, A. A. and E. Alpaydin, "Incremental mixtures of factor analysers" *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, Vol. 1, pp. 276–279, IEEE, 2004.

99. Dempster, A. P., N. M. Laird and D. B. Rubin, "Maximum likelihood from incom-

plete data via the EM algorithm" *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

100. Ghahramani, Z., G. E. Hinton *et al.*, *The EM algorithm for mixtures of factor analyzers*, Tech. rep., Technical Report CRG-TR-96-1, University of Toronto, 1996.

101. Joblove, G. H. and D. Greenberg, "Color spaces for computer graphics" *ACM siggraph computer graphics*, Vol. 12, pp. 20–25, ACM, 1978.

102. Gonzalez, R. C. and R. E. Woods, *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

103. Salton, G. and C. Buckley, "Term-weighting approaches in automatic text retrieval" *Information processing & management*, Vol. 24, No. 5, pp. 513–523, 1988.

104. Zhu, N. Q., *Data visualization with D3. js cookbook*, Packt Publishing Ltd, 2013.

105. De Volder, K., "JQuery: A generic code browser with a declarative configuration language" *PADL*, Vol. 6, pp. 88–102, Springer, 2006.

106. Bryan, P and Nottingham, Mark, 2013, "JavaScript Object Notation (JSON) Patch".

107. Lohmann, Niels, 2018, "JSON for Modern C++", `https://github.com/nlohmann/json/`, accessed at January 2018.

108. Doerr, M., S. Gradmann, S. Hennicke, A. Isaac, C. Meghini and H. van de Sompel, "The europeana data model (edm)" *World Library and Information Congress: 76th IFLA general conference and assembly*, pp. 10–15, 2010.

109. Weitzmann, K., *Age of spirituality: late antique and early Christian art, third to seventh century: catalogue of the exhibition at the Metropolitan Museum of Art, November 19, 1977, through February 12, 1978*, Metropolitan Museum of Art,

1979.

110. Akdağ Salah, A. A., "The online potential of art creation and dissemination: DeviantArt as the next art venue." *EVA*, 2010.

APPENDIX A: IMDB QUERY NAMES

A.1. IMDb actress names used for queries

Abbie Cornish, Adelaide Clemens, Adelaide Kane, Adrianne Palicki, Alexa Davalos, Alexandra Breckenridge, Alexandra Daddario, Alexandria DeBerry, Alexis Bledel, Alexis Knapp, Alice Braga, Alicja Bachleda, Allison Williams, Aly Michalka, Amanda Peet, Amanda Righetti, Amanda Seyfried, Amber Heard, Amy Adams, Amy Smart, Anessa Ramsey, Angelina Jolie, Anna Kendrick, Anna Skellern, Anna Sophia Robb, Anne Hathaway, April Bowlby, Ariana Grande, Arielle Kebbel, Ashley Greene, Ashley Judd, Ashley Newbrough, Ashley Rickards, Ashley Wood, Astrid Bergès-Frisbey, Beau Dunn, Bella Dayne, Bella Heathcote, Bella Thorne, Bijou Phillips, Briana Evigan, Brooklyn Sudano, Bryce Dallas Howard, Camilla Belle, Carla Gugino, Carlie Casey, Caroline Fauvet, Cassie Scerbo, Charlize Theron, Cherami Leigh, Chloe Bridges, Christina Hendricks, Claire Forlani, Cristina Rodlo, Crystal Reed, Daisy Betts, Dakota Fanning, Daniella Alonso, Danielle Campbell, Daveigh Chase, Delphine Chanéac, Diora Baird, Dominique DuVernay, Elena Satine, Elizabeth Olsen, Elizabeth Rice, Elle Fanning, Eloise Mumford, Elsa Pataky, Emily Browning, Emma Bell, Emma Roberts, Emma Stone, Emmy Clarke, Emmy Rossum, Erika Schaefer, Eva Green, Evangeline Lilly, Gal Gadot, Gemma Arterton, Gemma Ward, Genevieve Padalecki, Giselle Itié, Grace Holley, Grace Phipps, Gwyneth Paltrow, Hailee Steinfeld, Haley Bennett, Haley Ramm, Hannah Marks, Haruka Abe, Hazel D'Jan, Heather Fogarty, Heather Graham, Hilary Duff, Imogen Poots, India Eisley, Isabel Lucas, Ivana Baquero, Ivana Lotito, Jaime King, Jaime Ray Newman, Jaimie Alexander, Jane Levy, Janet Montgomery, Jasmine Waltz, Jennifer Garner, Jennifer Missoni, Jessica Alba, Jessica Biel, Jessica Lowndes, Jessica Lu, Jessica Stam, Jessica Stroup, Jocelin Donahue, Jordana Brewster, Julia Voth, Julianne Moore, Julie Fine, Julie Ordon, Kacey Barnfield, Kalia Prescott, Karina Testa, Kate Beckinsale, Kate Bosworth, Kate Hudson, Kate Mara, Katie Holmes, Katie Parker, Kay Panabaker, Kaya Scodelario, Kelli Barrett, Kelly Brook, Kelly Reilly, Kelly Vitz, Kiele Sanchez, Kirsten Dunst, Kirsten Prout, Kristanna Loken, Kristen Connolly, Kristen Stewart, Kristin Kreuk, Krysten Ritter, Léa Seydoux,

Lake Bell, Laura Marano, Laura Ramsey, Lauren German, Laurie Holden, Leelee Sobieski, Leslie Camila-Rose, Lily Collins, Louise Cliffe, Lucy Hale, Luisa Moraes, Lyndsy Fonseca, Mélanie Laurent, Marie Avgeropoulos, Marion Cotillard, Marlene Mc'Cohen, Mary Elizabeth Winstead, McKenna Knipe, Mia Kirshner, Mila Kunis, Mini Anden, Mira Sorvino, Miranda Cosgrove, Monica Bellucci, Morena Baccarin, MyAnna Buring, Mylène Jampanoï, Nadia Bjorlin, Napakpapha Nakprasitte, Natalia Dyer, Natalie Martinez, Natalie Portman, Natasha Henstridge, Neve Campbell, Nicola Peltz, Nicole Gale Anderson, Nicole Kidman, Nicole Paggi, Nicole Taylor, Nina Dobrev, Nora-Jane Noone, Odette Annable, Olga Kurylenko, Olivia Munn, Olivia Wilde, Ophelia Lovibond, Paula Patton, Peyton List, Phoebe Tonkin, Portia Doubleday, Rachael Leigh Cook, Rachael Taylor, Rachel Blanchard, Rachel Hurd-Wood, Rachel Maxwell, Rachel Nichols, Rachel Skarsten, Rachel Weisz, Rachelle Lefevre, Radha Mitchell, Raquel Alessi, Rhona Mitra, Rose Byrne, Rosie Huntington-Whiteley, Roxane Mesquida, Ryan Newman, Sami Gayle, Sammi Hanratty, Sandra Bullock, Sara Paxton, Sarah Elizabeth Johnston, Sarah Habel, Sarah Hyland, Sarah Lancaster, Sarah Roemer, Sasha Pieterse, Scarlett Johansson, Scottie Thompson, Selena Gomez, Selma Blair, Serinda Swan, Seychelle Gabriel, Shannyn Sossamon, Sharlene Rochard, Shauna Macdonald, Shawnee Smith, Shelley Hennig, Sienna Guillory, Sienna Miller, Skyler Day, Skyler Samuels, Stephanie Honoré, Summer Glau, Tamsin Egerton, Tania Raymonde, Taylor Cole, Tory Taranova, Tyne Stecklein, Vanessa Hudgens, Vanessa Marano, Veronica Taylor, Willa Holland, Zoey Deutch.

A.2. IMDb actor names used for queries

Aaron Eckhart, Adrien Brody, Al Pacino, Alain Delon, Albert Finney, Alec Baldwin, Andy Garcia, Anthony Hopkins, Anthony Quinn, Antonio Banderas, Armin Mueller-Stahl, Arnold Schwarzenegger, Art Carney, Ben Affleck, Ben Johnson, Ben Kingsley, Benicio Del Toro, Bill Murray, Bill Pullman, Billy Bob Thornton, Bourvil, Brad Pitt, Brent Spiner, Brian Cox, Brian Dennehy, Bruce Willis, Charles Bronson, Charlie Sheen, Charlton Heston, Chris Cooper, Christian Bale, Christian Slater, Christoph Waltz, Christopher Eccleston, Christopher Lambert, Christopher Lee, Christopher Walken, Ciarán Hinds, Clint Eastwood, Clive Owen, Colin Farrell, Colin Firth,

Cuba Gooding Jr, Daniel Craig, Daniel Day-Lewis, Danny Glover, David Carradine, David Morse, Dennis Hopper, Dennis Quaid, Denzel Washington, Dexter Fletcher, Don Ameche, Don Cheadle, Don Johnson, Donald Sutherland, Dustin Hoffman, Ed Harris, Eddie Murphy, Edward James Olmos, Edward Norton, Eli Wallach, Elijah Wood, Emile Hirsch, Emilio Estevez, Eric Bana, Eric Stoltz, Ethan Hawke, Ewan McGregor, F. Murray Abraham, Forest Whitaker, Gérard Depardieu, Gabriel Byrne, Gary Oldman, Gary Sinise, Gene Bervoets, Gene Hackman, Geoffrey Rush, George Burns, George C. Scott, George Clooney, Graham Greene, Haing S. Ngor, Harrison Ford, Harry Dean Stanton, Harvey Keitel, Heath Ledger, Henry Fonda, Hugh Grant, Hugh Jackman, Hugo Weaving, Ian McKellen, Jürgen Prochnow, Jack Black, Jack Lemmon, Jack Nicholson, Jake Gyllenhaal, James Coburn, James Cromwell, James Franco, James Purefoy, James Woods, Jamie Foxx, Jan Decleir, Jared Leto, Jason Robards, Jason Statham, Javier Bardem, Jean Reno, Jean-Paul Belmondo, Jeff Bridges, Jeff Goldblum, Jeremy Irons, Jim Carrey, Jim Caviezel, Joaquin Phoenix, Joe Pesci, Joel Grey, John C. Reilly, John Cleese, John Cusack, John Gielgud, John Goodman, John Hurt, John Malkovich, John Mills, John Travolta, John Turturro, John Wayne, Johnny Depp, Jon Voight, Jonathan Rhys Meyers, Joseph Fiennes, Josh Hartnett, Jude Law, Justin Theroux, Keanu Reeves, Keith Carradine, Keith David, Kevin Bacon, Kevin Costner, Kevin Kline, Kevin Spacey, Kiefer Sutherland, Kirk Douglas, Kurt Russell, Kyle MacLachlan, Laurence Fishburne, Lee Van Cleef, Leonardo DiCaprio, Leslie Nielsen, Liam Neeson, Lino Ventura, Louis de Funès, Louis Gossett Jr, Mads Mikkelsen, Marcello Mastroianni, Mark Ruffalo, Marlon Brando, Martin Sheen, Matt Damon, Matt Dillon, Matthew Broderick, Matthew Modine, Mel Gibson, Melvyn Douglas, Michael Caine, Michael Clarke Duncan, Michael Douglas, Michael Fassbender, Michael J. Fox, Michael Keaton, Michael Madsen, Michael York, Mickey Rourke, Morgan Freeman, Nick Nolte, Nicolas Cage, Omar Sharif, Orlando Bloom, Patrick Stewart, Paul Newman, Pete Postlethwaite, Peter Finch, Peter Fonda, Peter O'Toole, Peter Sellers, Peter Ustinov, Philip Baker Hall, Philip Michael Thomas, Philip Seymour Hoffman, Pierce Brosnan, Ralph Fiennes, Ray Liotta, Richard Dreyfuss, Richard Gere, Richard Jenkins, Robert De Niro, Robert Downey Jr, Robert Duvall, Robert Redford, Roberto Benigni, Robin Williams, Roger Moore, Roy Scheider, Russell Crowe, Ryan O'Neal, Sam Neill, Sam Shepard, Samuel L. Jackson, Scott Glenn, Sean Bean, Sean Connery, Sean Penn,

Sebastian Koch, Shia LaBeouf, Sidney Poitier, Stanley Tucci, Steve Buscemi, Steve Martin, Steve McQueen, Sylvester Stallone, Ted Danson, Tim Robbins, Tim Roth, Timothy Dalton, Timothy Hutton, Tobey Maguire, Tom Berenger, Tom Cruise, Tom Hanks, Tom Hulce, Tommy Lee Jones, Tony Chiu Wai Leung, Toshirô Mifune, Val Kilmer, Viggo Mortensen, Vince Vaughn, Vincent Cassel, Vincent Price, Vinnie Jones, Will Smith, Willem Dafoe, William Baldwin, William H. Macy, William Hurt, Woody Allen, Woody Harrelson, Yves Montand.