AI-powered Classification and Query of Color Patterns

With Applications to Movie Pictures

Master Thesis

August 2020

Linda Samsinger

E-mail: linda.samsinger@uzh.ch
Student ID: 08-338-386

Supervisors:
Prof. Dr. Renato Pajarola
Prof. Dr. Barbara Flückiger
Dr. Giorgio Trumpy
Gaudenz Halter, Dr. Alexandra Diehl

Visualization and MultiMedia Lab
Department of Informatics
University of Zürich

University of Zurich UZH

VISUALIZATIONANDMULTIMEDIALAB

To Ernst Samsinger, my *grandfather*

Abstract

An exhaustive evaluation of 29 multi-class and multi-label classification algorithms for mapping self-specified color name categories to all color space values in the CIE-L*ab color solid enables an effective color-aware search system. Based on these classified colors, higher-chromatic patterns from color theory and its rules, such as color contrasts, can be detected in a repository of movie pictures by an exploratory attempt to concretize their scientific definitions from the realm of art. Color histograms are drawn indirectly from color palettes instead of images for pairwise histogram similarity computation. Hence, a retrieval system involving three components is built: (a) a query of colors in images or their color palettes, (b) their top-n similarity and (c) their automated color contrast annotation. The implementation of the proposed method is conducted on the ERC FilmColors project's sample movie *Jigokumon* which consists of 569 subsequently shot video frames. A best macro F1-score of 92.7% was achieved using an Extra Trees classifier on Gaussian multi-label color classification which outperforms other task-adapted classifiers in this line of research. The resulting system is adaptable to digital movie databases (DMDb) with implications for 21$^{\text{st}}$-century cinematography.

Acknowledgement

I would like to thank my project supervisors Dr. Prof. Renato Pajarola and Dr. Prof. Barbara Flückiger for providing me with a thesis topic that sparked a long-lasting interest in colors. Their guidance and mentorship throughout the project duration have been essential to its successful completion. I would also like to extend my gratitude to my supervisors Dr. Alexandra Diehl and Dr. Giorgio Trumpy, who shared their insights about color visualization with me. Special thanks go to Gaudenz Halter for his support and the candid discussions on technical issues, who reached out to help according to the need of the hour to bring this piece of work to its full fruition.

Contents

1. Introduction

This thesis summarizes a semestral research project on color analysis that lies at the cross-section of the natural sciences and the arts. While color has been the subject of many books from ancient philosophers to present-day publications, with the advent of artificial intelligence in the 21^{st} century, the golden rules of harmony and balance from color theory are again open for new reinterpretations.

These developments have taken place, for example, in the emerging field of the digital humanities. A digital archive of images taken from the artistic works of the grandmasters of cinema can be annotated with color-based labels, analyzed with sophisticated statistics and queried by using indexing methodologies from computer science. The contributions have been to design efficient colorimetric mechanisms by leveraging machine learning for color classification to query a database of digital movie images and color palettes. Furthermore, the project involved a good chunk of data analysis conducted on color dictionaries and images, prototyping, testing and deployment of a color-aware search system.

In the age of big data, many databases have recorded a steady increase of multimedia records that harbor many challenges for matching, search and retrieval systems for automated labeling and search query formulation of images. Especially in film analysis and visual search, color management is critical to assessing the visual styles, moods and emotions of the predominant color scheme of color palettes used as a hallmark indexing method for analyzing movies. Color may serve as the common denominator shared among cinematographers to convey a certain underlying message that can be identified using color-based search functions.

Nowadays, it has become commonplace to query large collections of digital image archives based on a queried color of interest. Since color is the single most important low-level cue attracting a viewer's attention when watching a movie, other elements of art such as lines, forms, textures and shapes can easily be neglected. Colors pop out first and capture the attention of the viewer, hence their centrality in feature selection. However, the role of mapping intersubjective color names to an all-numerical color space remains a challenge. A considerable step forward may be to use machine learning methods to accomplish color classification. As such, colors are constitutive of color palettes from which higher-level chromatic patterns such as color contrasts can be derived, automating the color classification process to the highest accuracy possible could gain high significance in the arts. From color palettes extracted from an image, not only their color categories can be predicted, but also their similarity. The color distribution of an image is summarized using color histograms on the color palettes.

However, conventional and unconventional solutions that support the classification and search of color have a few shortcomings. Some conventional solutions found online require that one or more colors be associated with textual keywords that are manually annotated. Manual descriptors of colors can vary from person to person, which makes searching for color palettes comprising the queried color prone to error. For example, *blue* might be used by different people to describe different colors or *pink* and *rose* might be used by different people to describe the same color. Color-based image search still presents a challenge because of the following:

- Most color-based computations rely on the HSV color space instead of LAB, although the LAB color space is more extensive, perceptually uniform and more precise than others. In consequence, conversion into LAB space does not introduce rounding errors and values do not lose precision.
- Automated color classification is based on a combination of self-specified color name dictionaries, color name categories and a more exhaustive set of machine learning estimators that may reshape how color classification can be performed.

- Mappings between artistic and scientific models need to be defined heuristically to determine decision boundaries that have been latently known among experts in art and color theory but will be made concrete through driving the evaluations to and fro.

Therefore, this thesis undertakes to overcome these shortfalls by choosing the LAB space for all color-based computations, utilizing AI technology for color classification and delimiting the color theory rules for image processing applications. Thus, it provides an efficient way to classify colors from movie pictures or their derivative representations, such as color palettes, and automating the color-based annotation procedure. This automation could have an impact on how film-making is done in the future. The movie industry could have a more efficient instrument at their hands for using colors more effectively to tell a story. Selecting the right colors to set the tone of a movie could make or break the audience's reviews of that particular movie. A movie director needs to access knowledge about how to avoid such outcomes.

1.1. Goal of the Thesis

Film scholars often need to find temporal segments or single image frames within a large corpus of video material based on a query image, color or color palette. This includes queries to find all color palettes containing specific colors by name and querying them by similarity. Furthermore, it is often not only the actual colors present which are of interest but also the color contrasts formed by the presence of two or more colors. Hence, this thesis' primary goal is to allow the user to query a database of digital video image frames and their corresponding color palettes by color name, input image frame similarity and color contrast. From this follows a secondary goal, color and contrast classification, which is needed to reach the primary goal.

There are essentially three tasks and one optional task that are achieved as part of these two goals during the course of this project:

1. **Color name**: Given a hierarchically structured color palette, classify the contained colors such that a palette is searchable using categorical names. The best estimator needs to be determined that classifies a per-frame color palette's dictionary or basic colors. These are made searchable in a subsequent step.
2. **Similarity ranking**: Implement a method to compute the distance between hierarchically structured color palettes. From the color distribution in the input video sequence screenshots, similar screenshots are determined that display said color distributions.
3. **Color contrasts**: Classify the patterns within the color palettes in different types of color contrasts defined by the ERC FilmColors project[1] - it is shown how with one model video's screenshots, their corresponding color contrast types can be determined using the color distributions from the color palettes.
4. **Web application**: Implement a web-based user interface that allows the user to select or create a palette and generate a list of similar color palettes. This goal is optional. Since the three tasks mentioned above took more space than expected, the web application's proof of concept will be outlined in Appendix C.4.

Note: The described methods to accomlish the tasks stated above may be applied to other input video clips. This will make it possible for film scholars to determine motion pictures of similar color distribution or color contrast across different movies. Finally, the analysis could uncover the hidden truth about why a particular movie genre employs the same color patterns or color contrasts. In turn, this finding could lead to a theory of color usage in movies that movie directors could use as a tool for creating an enriched cinematographic experience.

[1]The ERC FilmColors project sets out to "investigate the relationship between technological processes and the aesthetics of film colors through a new interdisciplinary approach", `https://blog.filmcolors.org/2015/06/15/erc/`. This project is devised by the ERC FilmColors' research team headed by Prof. Dr. Flückiger. Hence, this project subscribes to the same research objectives.

1.2. Project Description

The project can be described as bridging the artistic concepts such as color theory and color harmony and the scientific color space models to classify colors and their patterns. The application of these concepts is backed by the terms established by the ERC FilmColors project. The results of the project are planned to be integrated into a video annotation tool named VIAN. VIAN was developed by the ERC FilmColors project in collaboration with the Visualization and MultiMedia Lab of the Department of Informatics. It extracts color palettes to describe the visual content of video segments and frames. This project's scope builds on the framework of VIAN: the extracted color palettes are used to index color and contrasts based on an automated classification procedure that enables the querying of the ERC FilmColors database by keyword and visual content. Hence, the implementation should be in line with the work that was carried out in VIAN.

The project consists of the following main parts, as seen in Figure 1.1: users may be asked to choose an input video clip which will then be subsampled into screenshots, also known as movie pictures or video frames. The color composition of the image frame will be represented by way of hierarchical color palettes that are extracted from video frames. The hierarchical color palettes' color will be categorized into color names that have two different levels of granularity: a lower-level set of dictionary color name and a higher-level set of basic colors. The user can query the color palettes using color names on either level and search for similar image frames that display similar color distributions using color histograms. Based on the per-frame color classifications, the hierarchical color palettes are categorized into different kinds of color contrasts for annotation and retrieval.

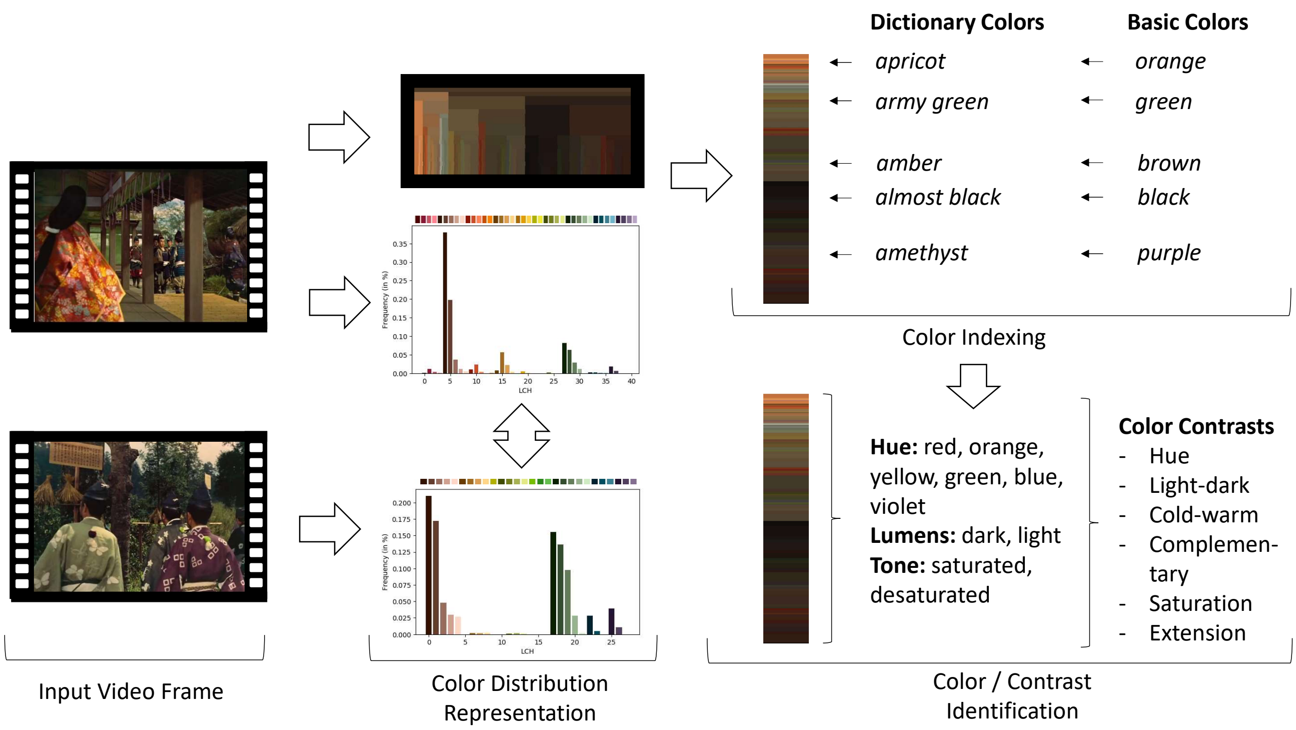


Figure 1.1.: Project overview: video frames are converted into hierarchical color palettes and color histograms. From hierarchical color palettes a flat color palette representative of the video frame is extracted. The flat color palette's colors are classified into dictionary and basic colors. A classified six basic colors are summarized into tables of hue, lumens and tone. The tables form the basis for color contrast categorization. Another video frame's color histogram is extracted and the similarity distance is compared with the original video frame's color histogram.

1.3. Problem Statement

After the year 2000, content-based image retrieval, such as colors in images, has become an increasingly important research field [DJLW08]. Studies on feature selection have revealed the importance of colors as a low-level cue for visual content. Research on the reasons for and the consequences of this shift has predominantly focused on developing variants of color histograms, a few color classification systems and content-based image retrieval. However, there has been little work exploring multi-class and multi-label classifiers to categorize color names. The mapping of color names to all colors of a particular color space model or vice versa using a broad range of possible data-driven AI models and color palettes as higher-level chromatic representation have laid idle compared to other studies on color histograms. Research into the domain of contrast classification, too, may yet have a significant academic impact.

This project is concerned with color classification using the LAB space (luminance, green-red, blue-yellow) instead of the more common HSV space (hue, saturation, value) and comparing an complete series of 29 different machine learning model classes for indexing and querying color images as compared to other studies where only three to five estimator classes are usually developed. Color classification will match the color palettes' quantized colors to its text-based form, which automatically becomes annotated. The annotation consists of color descriptors describing the color distribution of the original video frame. This surrounding text allows the user to use a standard text-based image search for querying a movie picture database. Contrary to most studies, the attempt here is to compute color histograms on color palettes instead of directly transforming the images into color histograms to compare their performances eventually. As with what is commonly performed in the literature, the similarity between two color histograms in this thesis is measured based on the Euclidean Minkowski metric [CCL+01], because the magnitude of the color vector inside a color solid matters.[2] More details about the origins of these problems can be found in Section 2.3. Thus, the following assumptions and requirements are posited to address and confine the problem statement.

Assumptions

The following assumptions are made to restrain the scope of the subject matter to a few trains of thought from which to draw more precise and concise conclusions. In reality, the assumptions need to be relaxed to apprehend the complex. However, for the sake of producing tangible domain-specific results, a simplification of the world to a few more or less axiomatic statements shall be permissible for now.

1. It is assumed that an exemplary demonstration of processing a color name dictionary, a proposed set of basic colors and a movie clip from the ERC FilmColors database can be generalizable to other color name dictionaries, sets of basic colors and movie clips; viable options are mentioned in due course.
2. It is assumed that the static representation of video frames in quality and number will persist over time, i.e. any posterior treatment of the video sequence such as scaling, sampling, color grading, bleeding, flickering or dithering are not taken into consideration. Such alterations may introduce unwarranted or perhaps deliberate noise into the process of annotating the images based on their originally conceived color distribution.
3. It is assumed that the horseshoe LAB color space is the most suited of all for all color-based computations because it encompasses the domain of visible light and more possible designations of color values than any other color space – this is due to the acceptance of floating-point values with potential infinite precision and besides being perceptually uniform, it exhibits the useful property of being device-independent, i.e. LAB color values will always display the same across all devices. Hence, for all color-related computations, the LAB color space is used. However, the RGB color space (red, green, blue) is used to visualize the colors on-screen such that LAB color values need to be squashed to fit the requirements of the monitor display.

[2]The magnitude of a word's vector does not matter by comparison.

4. It is assumed that for each color center of a color class, the distribution of color values follows a normal Gaussian distribution with a mean and a standard deviation. As the Gaussian distribution often occurs in nature, such an assumption may be judicious. Under the condition that a test for normality fails in most cases, such an assumption shall be rejected, however.

5. It is assumed that for all extended and simulated color values created during upsampling, the non-automated classification into class labels is correct. If the ground truth consists of a limited evaluation of samples from an expert where such samples cannot be sampled from an objective color space, the classified class labels are assumed to be correct. Other hand-made best-educated guesses for classification are also considered to be correct.

Requirements

The following requirements are specified as a means for limiting the project's scope. The research group together with the author of this thesis have discussed and, in consequence, have set forth plausible and rigorous requirements.

1. Color names and their values can vary across epochs, languages, gender and cultures because of socio-cultural differences.[3] These are controlled for in this project by rooting all taxonomies, nomenclatures and special terms related to colors in the English language in use since the year 2000.

2. Any distance metric will be based on the default Euclidean distance as part of the Minkowski [4] measure for distances. Taking the Euclidean is in line with the research group's dispositions of this project.

3. If the best hyperparameter value is found at either range limit of hyperparameter values passed to the search method, the upper or lower bounds will be extended and the training of the model will be reiterated accordingly to find a superior hyperparameter value score.

1.4. Chapter Overview

Hereinafter, this thesis is organized as follows.

1. **Motivation**: In Chapter 2, the incentives, reasons and interests are set out for conducting research in this project area. The ERC FilmColors project, the VIAN tool and related studies are the main reasons for setting up this research project.

2. **Background**: In Chapter 3, background information is given about the salience and evolution of retrieval systems for color palettes. Central concepts are framed and the project's architecture is explained.

3. **Methodology**: In Chapter 4, the methods and scientific techniques for tackling the three tasks are employed. In each section dedicated to a task, the technical solution and implementation are described in detail.

4. **Implementation**: The technical procedures for implementing and reproducing the project are demonstrated in Chapter 5. The practical implementation is comprised of the core functionalities, file components and challenges encountered during the project.

5. **Results**: The outcome of the implementation of the project is described and showcased in Chapter 6. Exemplary search queries for all three tasks and the corresponding search results are shown.

6. **Discussion**: The results are discussed in light of the theory and background in Chapter 7. By starting with an overall evaluation, each of the three tasks' advantages and disadvantages is analyzed more closely.

[3] Evidence for such variance is described in `http://www.empiricalzeal.com/2012/06/05/the-crayola-fication-of-the-world-how-we-gave-colors-names-and-it-messed-with-our-brains-part-i/`

[4] The Minkowski distance is a distance or similarity measurement between two points in the normed vector space (N-dimensional real space). It is a generalization of the Euclidean distance and the Manhattan distance, `https://iq.opengenus.org/minkowski-distance/`

2. Motivation

In film and visual studies, it is not uncommon to use electronic records such as videos for subsampling and deriving images of interest, such as digitized color screenshots of various deposited cinematographic works, or other suitable content to perform color analysis via annotation. Note that the audio-channel is eclipsed when converting from video to image format. Digital images are extracted for every pre-specified interval of the video. A user searching an image by color may think of an autumn leaf color as *red*, *yellow* or *green*, but movie pictures may include thousands of shades of *red*, *yellow* or *green* depending on the lighting and retouches of the film sequence. Hence, a too simplistic approach towards a color-aware search of such images may produce search results that are unwanted, inadequate or confounding to the user.

2.1. ERC FilmColors Project

The European Research Council (ERC) has awarded an advanced grant of EUR 2.9 million to promote a research project called *"Film Colors. Bridging the Gap Between Technology and Aesthetics"*[1] that explores the impact of film color processes and their limitations to on-screen aesthetics [FHW19]. Their task is to use technology to restore the cultural heritage of cinematic development. Historical films are restored through digitization, notwithstanding the parallel development of tools that allow for scientific analysis of the movie clips from a physical, chemical art-historical and color-theoretical perspective. The results of the project are of interest to different groups of experts such as web developers, artists, illustrators, marketers and advertising experts, fashion, interior and graphic designers, researchers, film historians, archivists, curators and students in the field of color science, color harmony, image processing, computer vision and film and visual studies or the general public.

2.2. VIAN tool

Since existing solutions for film analysis in the digital humanities address perceptual and spatial color information only tangentially, a *Visual Video Annotator* (VIAN) was built to address the semantic aspects of film color analysis. *"The tool enables expert-assessed labeling, curation, visualization and classification of color features based on their perceived context and aesthetic quality. It is the first of its kind that incorporates foreground-background information made possible by modern deep learning segmentation methods. The proposed tool seamlessly integrates a multimedia data management system, so that films can undergo a full color-oriented analysis pipeline"* [HBRFP19]. The aim is the development of an *"interactive crowdsourcing platform that allows users to apply (semi)automatic and manual tools for their film color analyses and then compares their results to the analyses of the research team and other users"* [FH18]. VIAN is an application for *"temporal segmentation, visual annotation and computational analysis of films"* [FH18].

The VIAN web app project taps into a comprehensive database of more than 400 commercial video clips from 1895 to 1995 for annotation, visualization and analysis using VIAN. The video clips were divided into video segments. Each segment consists of a varying number of screenshots. The video segments were manually categorized into different kinds of colors, color contrasts, and color schemes. However, such textual annotation of a video sequence through a set of external keywords describing the video frames' visual content raises several problems [CBP99]:

[1] `https://www.film.uzh.ch/de/research/projects/verbund/ercfilmcolors.html`

- Manual annotation may encounter failures in human judgment due to a lack of expertise or knowledge of decision boundaries or biased cognitive perception or impaired awareness.
- Manual annotation is not easily scalable or requires expensive and repetitive human labor such that annotations only exist for video segments instead of screenshots.
- Manual annotation is based on vague rules about classification that may differ from annotator to annotator.
- Textual features do not support retrieval by similarity.

Hence, an automatic image annotation classifier that learns to name the colors of a movie frame's color palette and classifies the color patterns into different kinds of color contrasts would supplant and extend the manual annotation. VIAN uses superpixel segmentation and a bottom-up clustering approach to extract hierarchical color palettes from video frames. A color palette can be a list of BGR (blue, green, red) colors and the color's probability values or a color palette image. For alternative methods on how to extract colors from images, see Appendix A.9. These color palettes can be further processed to devise a retrieval system for digital image archives based on color names and color contrasts.

One of the functionalities of the VIAN tool is to extract color palettes from video images. The method used is one of agglomerative clustering. The color distribution of an image is represented at different levels of granularity. Since the VIAN tool's automated color palette extraction algorithm based on the SEEDS method already exists, it was adapted and integrated into this project. In the context of VIAN, a color palette is the product of a SEEDS superpixel segmentation [VdBBR$^+$12] followed by a bottom-up clustering. This clustering procedure results in a hierarchically structured color palette, where every parent node is assigned the average LAB color of its children, allowing the user to define the merge depth of interest to obtain a comprehensive representation of the original video frame.

2.3. Related Work

An extensive body of work already exists that pertains to color classification and image matching and retrieval. The theories proposed in the computer graphics communities are to understand in the context of color theory. The revised literature is organized based on the three project tasks: color-based machine classification, content-based image retrieval and color contrast.

.

2.3.1. Task 1: Color-based machine classification

A great deal of academic work on finding a mapping from space color values to basic colors has been carried out. Traditionally, academic researchers have used the traditional chip-based method for learning color names [vdBSK08, Moj05]. Another method to learn color names is by using tagged color name images [LLYZ12]. Results have shown that color names learned from such images significantly outperform color names learned from labeled color chips on retrieval and classification [vdWSV07]. This project follows in yet another line of researchers that use fuzzy sets to assign each pixel in a color space a basic color that follows a probabilistic distribution [BVB08].

The classification can be done using histogram analysis for color cluster detection [Tom92]. In studies where machine learning for image retrieval was used, only a limited set of two to five classifiers are usually compared [SKK12, LLYZ12]. Traditionally, task-adapted models from text analysis such as a generative model of probabilistic Semantic Analysis [vdWSV07] from topic modeling or supervised Latent Dirichlet Allocation [SS12] or probabilistic Latent Semantic Analysis [SF10] have been prevalent in learning color names. So far, not much work has been done using multi-class and multi-label models for learning.

Many researchers use the HSV or a variant of the HSV color space [SF10] or RGB [LLYZ12, CSMS17] for computations instead of LAB. Some compare different color spaces against each other when training a color classifier [vdWSV07]. Results have shown that the LAB color space achieves a better performance of the learnt model for image retrieval when compared to RGB and HSV [SS12]. Also, most studies on color classifications are related to a domain of expertise such as traffic light [BW16], lip [ZLY+10] and hair [Sar16] color classification. Experiments to develop a global method for color classification are relatively few and far between [Tom92].

2.3.2. Task 2: Content-based Image Retrieval (CBIR)

The task of content-based image retrieval (CBIR), defined as technology that helps to organize digital picture archives by their visual content [DJLW08], has grown extensively after the year 2000, bringing together fields such as computer vision, but also machine learning, human-computer interaction, database systems, data mining, information theory, statistics and psychology [DJLW08]. Systems for image retrieval and forerunners to VIAN are QBIC, BlobWorld, VIPER/GIFT, SIMBA and SIMPLIcity [DKN04]. In the beginning, image databases were managed using textual image descriptors for indexing. However, this approach was time-consuming and failed to do justice to the full visual richness of the content images [CCL+01]. More recently, these *bag-of-words* approaches were gradually replaced by a new generation of *feature-based* approaches. Feature-based approaches use low-level visual features such as color, texture, shape and layout [FSA+95] to mine not only conceptual but also perceptual visual content that allows similarity retrieval in addition to exact retrieval [GJ96].

As for colors, they are more easily quantized than words [CCL+01]. Also, they are more directly relatable and capture an image more comprehensively. The visual information in an image is typically characterized by the arrangement of color pixels such as in a color histogram [SB91] and by the subsequent distance metric calculations [CB99]. A simple color histogram's effectiveness was made evident in an experimental study by Cinque et al. [SB91]. By building on top of color histograms, progress has been made with indexing methodologies that rely on weighted, color-spatial information (Spatial-Chromatic Histogram, color correlograms, color coherence vector) for retrieving quantized images that call into question the sufficiency of standard color histograms [CCL+01, CTO97].

HSV was extensively employed to generate a color-based histogram representation of the original image [SC95]. The cylindrical HSV color space was the preferred method because HSV is the only color space where decision boundaries for different colors will be linear [BSD+13], as seen in Figure 2.1. Furthermore, HSV seems to compensate for artifacts and color distortions. Also, manipulating one of the three color channels will not lead to false colors [SC95]. Studies that compare one color space to another most often incorporate RGB and HSV [JFS95]. However, in this project, a novel approach using the LAB color space for all computational implementations of color is used. The LAB space is chosen due to its perceptual uniformity and LAB's superiority in terms of the unlimited number of colors it can describe. In addition, through non-linear functions, machine learning classifiers can learn to establish non-linear decision boundaries in such high-resolution color spaces.

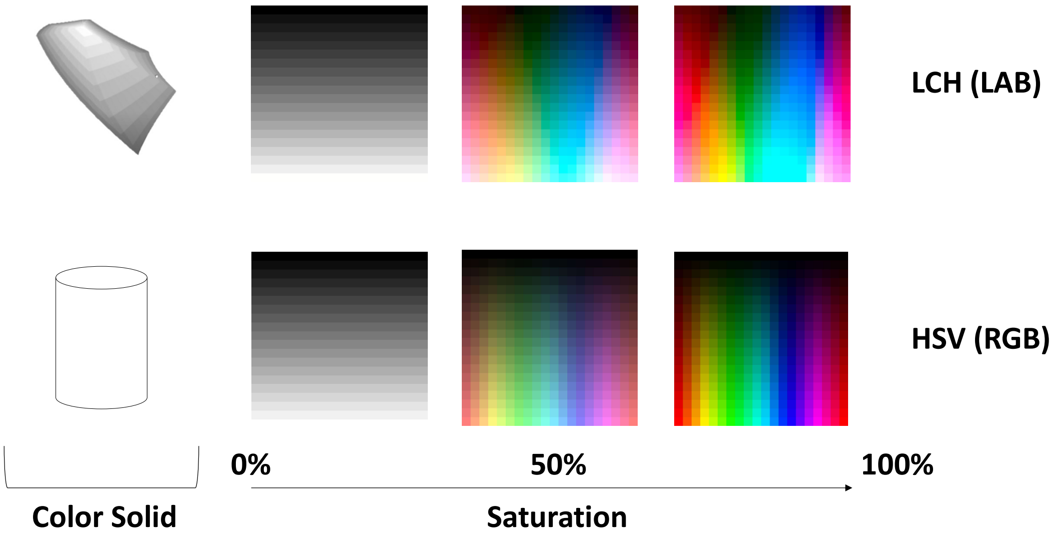


Figure 2.1.: HSV versus LCH color spaces; Comparing the decision regions, HSV has linear decision boundaries, but LCH is perceptually uniform.

2.3.3. Task 3: Color contrast

While much work already exists about color harmonization, i.e. finding colors that match each other, for example, with *Adobe CC's Kuler*, these efforts are mostly linked to generating new colors that enhance the color harmony of a given image [COSG+06] instead of identifying color harmonies in an image. Also, rather than finding color harmonies, the focus here is to capture color disharmonies, or color contrasts, in a series of images from an input video clip. In the same vein, it seems that no prior academic work has been done so far that attempts to define the rules governing the different kinds of color contrasts from color theory through a lens of clear-cut programming logic. Therefore, it is desirable to use modern technologies to make art theory applicable to querying images more precisely and efficiently.

3. Background

To reach the project's objective, a preliminary definition of the critical concepts employed and main architecture components is essential. A brief overview of the most fundamental terms and how they relate to each other, as well as the structural pillars for navigating through the project's tasks, is presented.

3.1. Concepts

Conceptual configurations are outlined in this thesis to provide a thorough understanding of the embodiments. Complex and repetitive features may be omitted or simplified to avoid losing the ones skilled in the arts in minor details. The color-aware search of electronic records associated with one or more images may be enabled by associating the collection of images with representative palette colors in a suite of color palettes drawn directly from the images. This process necessitates a choice of a select few color spaces, color categories, data sets, color representations and color contrasts.

Color spaces: LAB, RGB, HSV, HEX color spaces or color order systems are continuously scaled spaces that are based either on additive color mixtures (RGB), subtractive mixture (CMYK) of colorants or principles of color perception or color appearance (CIE-L*ab) [Fai97]. The most important color spaces for this project are the RGB (+L), HSV, CIE-L*ab (CIE-L*ch) hereafter LAB (LCH) and HEX color space. The LCH color space is derived from the LAB color space by plotting the Cartesian color value coordinates into cylindrical form. For any color space given, any other color space can be derived automatically by color value conversion, see Appendix C.1 for online color conversion tools. Each pixel in a colored image can be represented using different color spaces that capture the color distribution in the image. Each color space has its advantages and disadvantages. The reasons for choosing one against another are described below.

RGB is the most frequently used color system for digital images due to its compatibility with computer monitors [SC95]. The *red*, *green*, and *blue* primary colors correspond to the intensities of the electron beams that excite *red*, *green*, and *blue* phosphors on a color Cathode Ray Tube [BBK82], hence the colors will vary from monitor to monitor. HSV is more easily understandable because it allows the user to specify color along the channels of hue, saturation and value that are more relatable for describing colors than the *red*, *green*, and *blue* primaries of the RGB color system [BBK82].

The LAB (LCH) color space is used for any computations with colors because it is a perceptually uniform (or linear) color space; a uniform change in LAB space will produce the same change in visual significance as compared to HSV in Figure 3.1. Hence, it is possible to interpolate between color values in the system unambiguously [Fai97]. Because the LAB space allows for unlimited precision, it is larger than all other color spaces; users can search more specifically about a color because more information is contained in each color value. In Figure 3.2, the chromaticity diagram shows the LAB color gamut, which is larger than the RGB and CMYK color space. On the other hand, beyond that which can be perceived by the human visual system, covering the full gamut of LAB nuances of colors will be less likely to have any practical applicability. For example, these color values cannot be visualized to the viewer. Since the LAB space can only be used under limited viewing conditions, e.g. daylight illuminants, high luminance levels and standardized illuminating geometries [ML07], it serves as a specification system for colorimetric computations only.

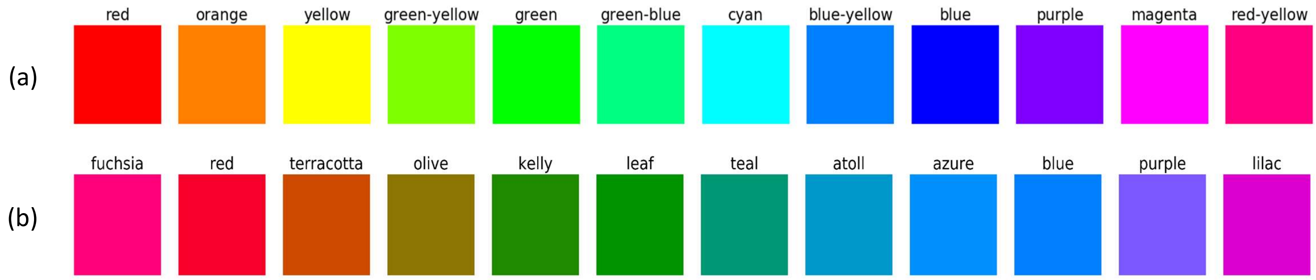


Figure 3.1.: Color swatches in 30° intervals around the 360° color wheel: (a) in HSV space, the green-yellow, green and green-blue are hardly distinguishable; (b) in LCH space, the colors are perceptually situated at equal intervals

For visualization purposes, the colors' RGB values are displayed because a computer cannot render all LAB colors. The LAB values are projected into RGB space to fit into what is portrayable on a computer. LAB color values may also exist outside of visible light. However, the displayed colors will vary across devices in other color spaces but LAB.

HEX is the color space of choice for web design and web applications. HEX is equivalent to RGB using hexadecimal notation. Hence, the conversion HEX-RGB is less trivial than RGB-LAB. Since this project's optional task is to develop such a web application, the HEX color system was included. For an extended list of color spaces please consult Appendix A.6 [NC15].

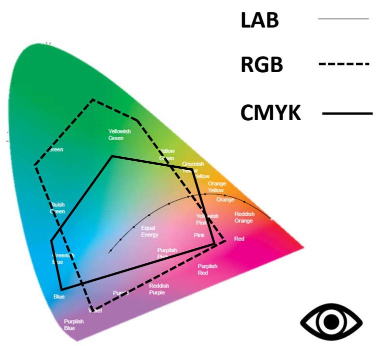


Figure 3.2.: LAB, RGB and CMYK color spaces. LAB encapsulates the two other color spaces RGB and CMYK

Color categories: Most color spaces are based on three **primary colors**. For example, RGB is based on *red*, *green*, and *blue* and CMYK is based on *cyan*, *magenta*, and *yellow*. **Basic colors** extend this limited set to a common set of half a dozen to a dozen colors that the average English speaker uses in daily speech to designate colored objects (see Appendix A.2). Different color theorists propose different basic color systems based on different theories of what basic colors should embody. For an overview of the most prominent color theorists and their basic color system see Appendix A.8. **Dictionary colors** are an extension of basic colors; this is the entire vocabulary of all words for colors the English speaker uses at a more advanced language level. The number of dictionary colors can consist of more than a dozen to circa 1'000 different color names, depending on the creative imagination of the color name collector. Beyond color names, colors can be specified more precisely by using numbers with ranges that depend on the chosen color space. These colors are called **space colors**.

Borderline colors are defined as colors at the border of two or more color categories. For example, it is unclear whether *chartreuse* is *yellow* or *green*, because it is situated at the decision boundary of the two basic color categories. While some would classify *chartreuse* as *yellow*, others would classify it as *green*. Even others would classify *chartreuse* as both *yellow* and *green* simultaneously. Hence, *chartreuse* is a borderline color.

Color data sets: For categorizing space colors or dictionary colors into basic colors (or primary colors/dictionary colors), a data set must be acquired that maps features to labels. A **color name dictionary** (CND) assigns each unique dictionary color to a space color (see Appendix A.5.5). In contrast, a color naming system list dictionary colors only without providing the corresponding space color values. An example of such a color naming system is the Sundberg Color Thesaurus (see Appendix A.5.2). Another way of building a data set is to subsample a color space of choice for a given interval and to manually classify these space colors into color categories to build a **color space grid**.

Color representations: A single color can be represented as a text, numeric or as an image called a **color patch** (see Appendix A.1). A distribution of colors is present in **color images** as well as their derived form: **color histograms** and **color palettes**. Color histograms, which are the more basic representation of the color distribution of an image as compared to color palettes, may supplement the analysis. In contrast to color palettes, color histograms illustrate the frequencies of each pixel in the image in a two-dimensional plane, whereas color palettes are obtained by averaging the color value of superpixels in the image at different degrees to convey information about the color composition of an image. Color palettes can be flat or hierarchical depending on the extraction process (see Appendix A.7).

Color contrasts: Color contrasts are observed effects of disharmony present by the combination of colors in an image as opposed to color harmonies.[1] The contrast exists because of a high-level chromatic pattern observed in a color image's color distribution. These high-level chromatic patterns are rules that describe the graphic content on an emotionally expressive level as opposed to an impressive or symbolic level (see Appendix C.5) [CBP99]. For example, the two colors' property can be distinguished as a positive and negative pole or more than two contrasting poles. The following classes of color contrasts are short-listed by the ERC FilmColors project: **contrast of hue**, **light-dark contrast**, **cold-warm contrast**, **complementary contrast**, **contrast of saturation** and **contrast of extension** – in total six contrasts of color. Color contrasts can be determined by using only six basic colors: the primary and secondary colors (the secondary colors are mixed of the primary colors) to determine the color contrasts.

3.2. Architecture

The four tasks which form part of the project contain the following functional research goals:

- Task 1: Search color names
- Task 2: Find similar images
- Task 3: Identify color contrasts
- Task 4: Build a web application

[1] Color harmonies, as described by Johannes Itten, is the effect on the viewer of two or more combinations of colors. However, it is not just any effect, but a harmonious balance, a symmetry of forces between colors. Color harmonies are primaries, split primaries, complementary, split complementary, analogous, triads, tetrads and squares of colors – in total eight harmonies of color. They require the use of twelve colors: the three primaries, three secondaries and six tertiaries.

Task 1 is the most work-intense task of all and, accordingly, more has been written on that particular subject matter. Task 2 is the shortest because it does not require the heavy-loaded training of machine learning estimators as in Task 1 and Task 3. Task 3 is similar to Task 1 but differs from Task 1 because it imposes rules in addition to the color classification analysis. Task 4 is an optional development of a web tool that consolidates all functionalities Task 1-4 via a graphical user interface.

In Figure 3.3, the project's task components are shown for schematic analysis. A summary of a possible workflow is described in the following. Note that the retrieval system components can be queried asynchronously. However, the color palette and color histogram representations and related classification predictions need to be computed before raising a search query.

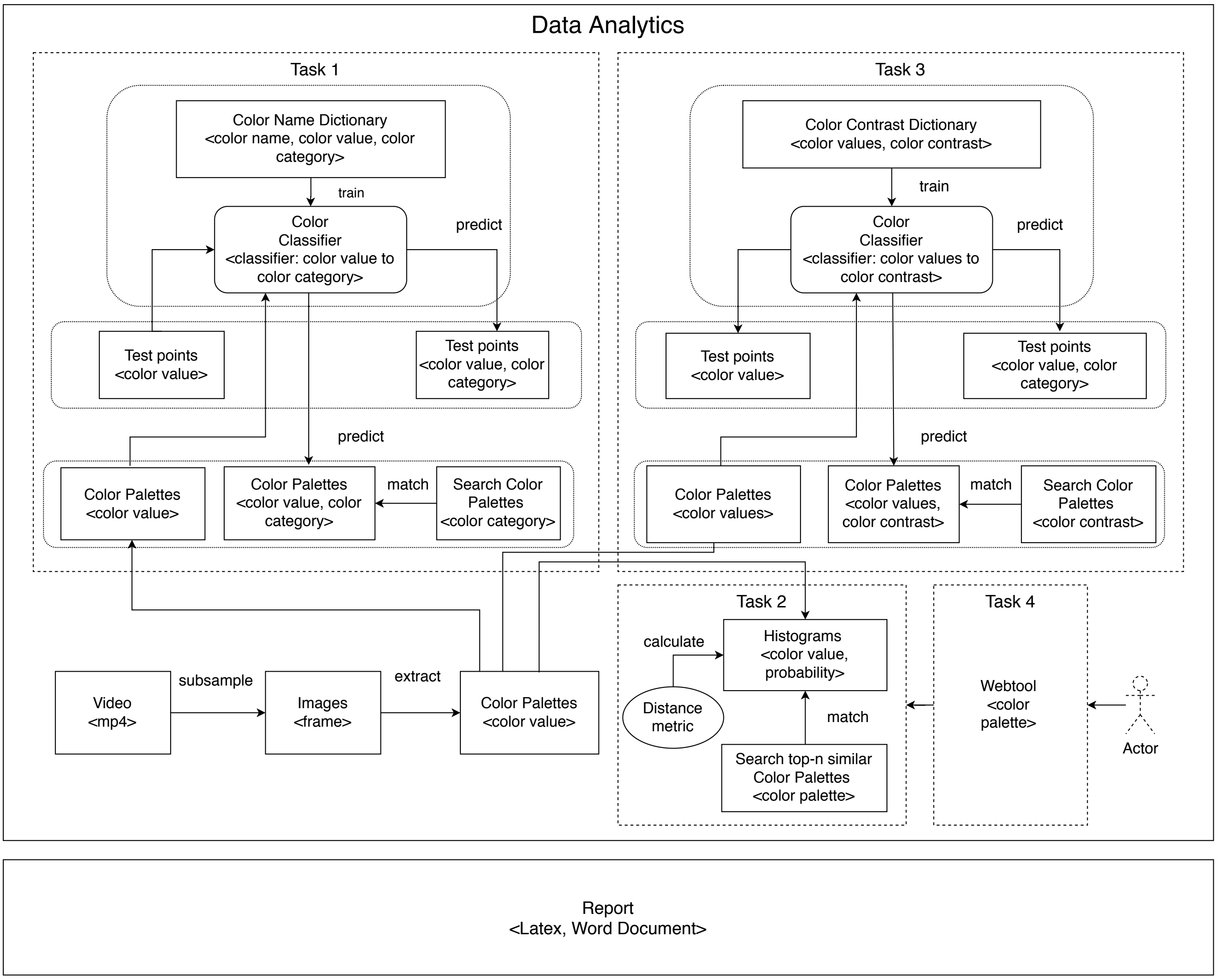


Figure 3.3.: Overview of tasks

At the start, images are produced by subsampling a film sequence into sequential screenshots by the interval. A (hierarchical) color palette is extracted from the image. The color palette can contain colors that are not part of the colors in the CND. The images or their representative color palettes consist of palette colors represented as text, number or image. The machine learning classifier is fed with these new colors from the color palette for predicting the color category of each color patch within the color palette. Such a predictor needs to be built for both Task 1 and Task 3.

Task 1 is accomplished in this step: the trained classifier is used on the color palette of an image to predict a color category for a given palette color. The trained classifier is the best model across all model classes and all models in a model class. The color categories can either be a dictionary, basic or contrast color represented as text, number or image.

Then, machine learning classifiers are trained for Task 1 and Task 3. The classifiers attempt to learn the CND's mapping of colors to a small number of color categories. A CND consists of colors represented as text, number or image. A good classifier can generalize well of what it has learned to new colors by predicting their correct color categories. Training a classifier consists of matching a dictionary color name to a color category where the color category can be itself a dictionary color, a basic color or a contrast color represented as text, number or image. The color category is then used for indexing the images and palettes. The indexing is required for retrieving the results of the search query.

Next, all color palettes are transformed into histograms and the pairwise distances are calculated between them such that Task 2 is accomplished. For combinations of two-paired color histograms of images or color palettes, the distances are calculated for retrieving top-n similarity.

The palette colors are transformed into contrast colors, from where color contrasts can be deduced. Contrasting color patterns of a color palette is derived by applying the rules of color contrast to identify a color palette's color contrast for Task 3. Color contrasts are represented as text or color bar image.

Optionally, the functionalities of Task 1 to Task 3 are made available on a web tool. Mockups of the web tool are in the Appendix C.11 Similar color palette search tools exist online, see Appendix C.2 for a benchmark of ten online color palette search tools.

4. Methodology

4.1. Task 1: Color Name Prediction

The FilmColors project's Visual Annotation webtool (VIAN) predefines categorical color names such as *orange*, *copper* and *mustard* that can be used for the search query. A function in VIAN is able to extract a color palette from an image in hierarchical order, i.e. get n-most relevant color patches at the lowest palette level and hierarchically merge them in a tree until the highest level is reached with only one color that is the average color of the image. The task is to make the colors in these color palettes searchable based on the VIAN color name options. The search query can be operated on two different levels of granularity: basic colors and dictionary colors.

Task 1: Given a hierarchically structured color palette, classify the contained colors such that a palette is searchable using categorical names.

Users that specify colors using natural-language-based systems instead of a more scientific real number based system were found to be more accurate despite the coarser granularity of that system: users are more apt to choose a color when provided a narrower range of choice and using more humanistic factors as color descriptors [BBK82]. There are essentially two ways to query a color by natural language color categories.

First, a basic color can be queried by matching the corresponding color name found in one of the colors of a color palette. For example, the user could query *red*, a basic VIAN color. Then, the user gets all color palettes containing this category's corresponding color names such as *red*, *ruby*, and *crimson* etc. which are all mapped to *red*. Second, a dictionary color can be queried by matching the corresponding color name found in one of the colors of a color palette. For example, the user could query dictionary color *avocado* whose basic color is *green*. Then, the user gets all color palettes containing color name *avocado* only. The user could query *red* specified as dictionary color and not as basic color as well. Then, the user gets all color palettes containing color name *red* only.

Colors from a random color space instead of their color names are too specific for querying a color palette. For instance, if a user decides to query the RGB(215,36,77) triplet and expects to receive all color palettes that contain that color, the user will most likely have no search results, because the likelihood of the queried color to occur in one of the 101 color patches of all color palettes is comparatively low. The color palette's colors need to be clustered together into a dictionary color or a basic color at an even higher level to be effectively searchable.

In Figure 4.1, an overview of the different design components is shown. For building a machine learning classifier for color classification, any CND can serve as data set. Such color name dictionaries are formalized as <EFFCND source-system>, where <source> is a placeholder for the data source's name and <system> is any basic color category system. In this project, the model will learn to predict color categories of color names or values found in the color palettes of movie frames from the ERC FilmColors project Movie Frame Database (MFD)[1]. Thanks to the hand-coded categorization of correct classes, the model can be applied, evaluated and improved. After this stage, the model is ready for deployment: the finished model can be applied to other movies.

[1]The Movie Frame Database (MFD) consists of movies such as *2001: A Space Odyssey*, *L'Inhumaine*, *Jigokumon*, *Fantasia*, *Nerven*, *Alien*, *Lawrence of Arabia*, *Top Gun*, *Amer* and *Drive*

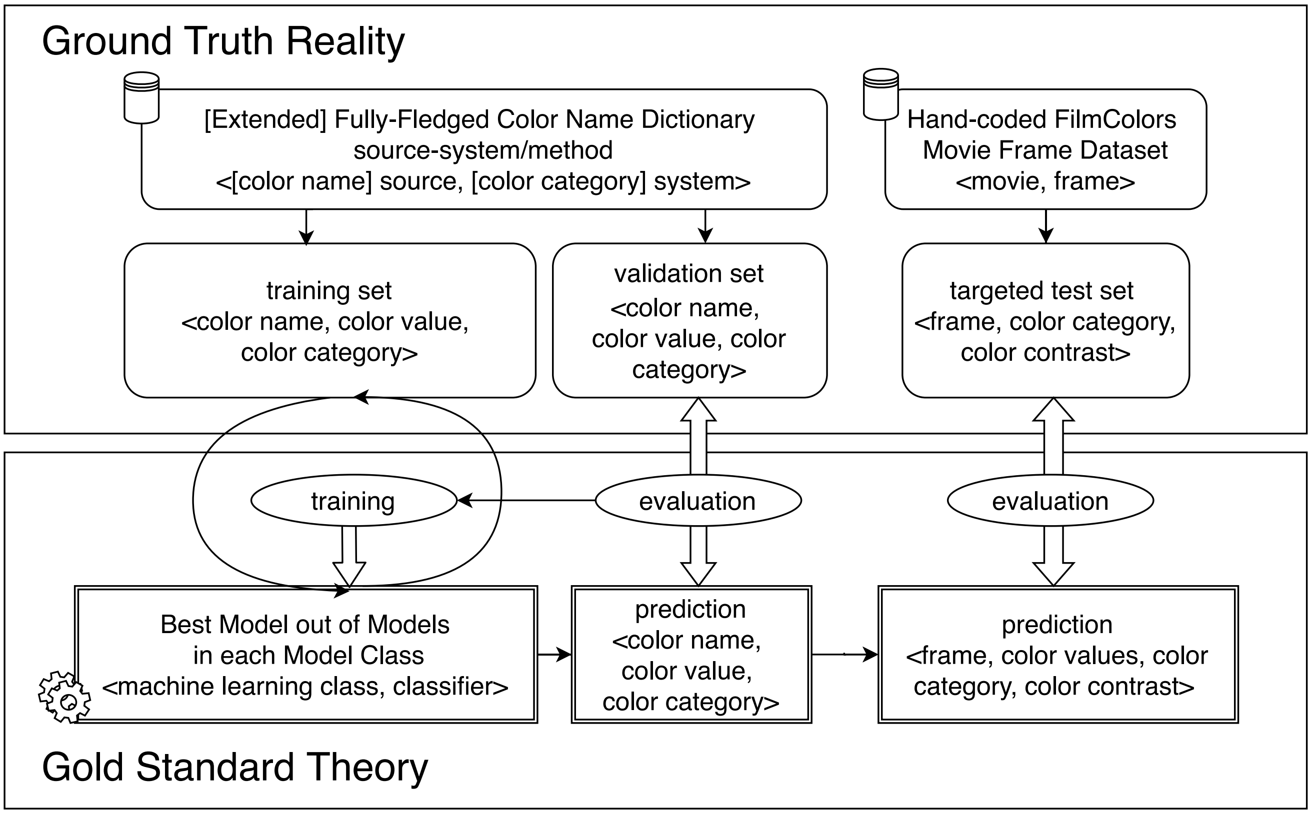


Figure 4.1.: Building a machine learning classifier from a Color Name Dictionary (CND) to the FilmColors Digital Image Movie Frame Database (MFD)

4.1.1. Data Augmentation

From one of these data sources, a raw color name dictionary is downloaded and saved to the local machine. Only English color names are retained. For example, in the EPFL Color Thesaurus data set [LLBS12] (see Appendix A.5.3), the raw color name dictionary has 6'167 rows of colors in nine different languages; with attributes such as their English names, sRGB and LAB color values. For example, for *baby blue* a discrete color value in sRGB or LAB is returned. Thus, we process this CSV-data set in Excel by removing all non-English color names. There remain 720 colors. This raw list of pairs consisting of a color name and a color value is what is called the original color name dictionary.

The original Color Name Dictionary (CND) needs at least a list of different unique color names and a corresponding discrete color value of any color space (RGB, HEX, CMYK, LAB, HSV). Typically, the builder of a CND provides a HEX color code for each color name.

Starting from the original CND, a Fully-Fledged Color Name Dictionary (FFCND) can be developed. An FFCND contains an index of all unique color names, the color name's language, the color names and color values (RGB, HSV, LAB, HEX) in it. For further details, please see Appendix A.4 for how to build an FFCND. If one of these attributes is missing, it can be derived by inference or conversion.

It is always better to have more data in the data set or colors for each basic color category because it will help assess the boundaries of a basic color more accurately. Especially, if the number of colors per basic color is imbalanced across all basic colors, it is recommendable to get more cluster points for basic colors lacking in representative colors. In general, for all unique color names with their corresponding unique color values, one-to-many color names can be mapped to one (multi-class) or many (multi-label) basic color category/ies.

In case the classification labels consist of draws from a set of basic colors, the data set needs to be extended if a basic color does not have at least three corresponding dictionary color values in the CND. Only if there are a minimum of three feature values per label can a simple machine learning classifier be trained. For instance, this is part of the functional requirements for training a k-nearest neighbors (KNN) algorithm. For example, basic color label *ultramarine* has no representative feature value. First, for color name *ultramarine*, a color value needs to be found. Color name *ultramarine* can also be swapped with an exact color name synonym. Then, other color names mappable to *ultramarine* need to be found, such as *azure blue* and *cobalt blue*. These new color names should not be already existing color names, because all dictionary colors must be unique. These two additional color names are classified into *ultramarine* as well. The values of these two color names must be close enough to *ultramarine* or, at best, be at the border of *ultramarine* that slowly fade into another basic color. Since the color boundaries are unknown, the latter is subject to experiments and the former is actively practiced.

In case the classification labels consist of dictionary colors, the data set needs to be extended with at least two more representative dictionary colors as features for the dictionary color labels. For example, color name *ochre* is not a basic color. However, all unique color names are transformed into basic color labels, where one class label is mapped to many different and unique color names (1:N). While dictionary color names are unique, their respective color values do not have to be. Indeed, nine color names have the same color value in RGB as another color name. There are even more for LAB color values: 38 color names share the same LAB color value as another color name. Starting from there, at least two more color names representing *ochre* need to be found, for example, *mustard* and *amber*. These near-synonym color names should not already be existing color names. Some original color names such as *cream* and *peach* are very close to each other in color space, which makes it difficult to find similar color names for each of these basic color categories. Following in this vein, we create a data set where there exists only one category class *cat* and at least two more color extend the color values for each unique color name. For example, *ochre* (RGB) is classified into *ochre*, we derive *ochre2* (RGB) that is classified into *ochre* and *ochre3* (RGB) classified into *ochre* as well.

Now that the case objectives are clear, four different methods can extend the dictionary, which are elaborated as an extension by interval, extension by search, extension by upsampling (resampling) and extension by imputation. The methods can be combined at will to cancel out any disadvantages found in the other methods and emphasize the advantages of the owned method.

Method 1: Extension by Interval For colors with one color value only, such as VIAN's *apricot*, two more values need to be obtained before upsampling. As long as each element of the LAB color triplet is not at the domain's marginal limit, the two additional values are computed by adding .5 and subtracting .5 all elements in the LAB color triplet such that the average continues to be the original color value and the standard deviation is as small as possible. Otherwise, an extension by search is proposed. Note that a scale of 1 or more could not have been implemented in a 0-255 integer-based RGB color space.

Method 2: Extension by Search For colors with no color value or one color value only, such as VIAN's *ultramarine* or *apricot*, new color values can be obtained by searching an online database of digital images for the target color. Since the color value represents the color authentically, a maximum of two color values, two in the case of *ultramarine* and one in the case of *apricot* need to be searched before upsampling. After downloading Google images for the color using Chromedriver, different average values can be computed for the target color by using different set lengths of Google images.

A caveat that marks this procedure is the calculating of color values for the first few images only. For example, averaging a few images could lead to a greater color bias than the average for a greater set of images. In addition, semantic errors occur for example for *adobe*. Searching for `adobe` and `color` on *Google Image Search* will produce image results that display the *Adobe CC* software functionalities applied on colors instead of the orangey color patch for the color name *adobe*. Another source of error are duplicate image downloads of different image sizes. The images can have different non-derivable URLs among them and different file sizes. When taking the average, the color values will be biased towards the images that exist doubly: once as a thumbnail and once as a background image. As a solution, it is possible to use Flickr instead of Google to source images, because Flickr's image stock is predicated on photography. Anomalies form the exception from the vast majority – by manually adjusting the keywords in the Google search bar for the anomalies only, the correct images can be quickly obtained. For the same images, it is possible to refine the search criteria for additional color values or resolution. Another possibility is to remove images with histograms that are the same to another image's histogram.

Method 3: Extension by Upsampling Upsampling based on randomly generating resampled Gaussian-distributed data values is the best approach for extending the dictionary. Random samples are drawn for a given mean color value and standard deviation of a color category. The assumption of the Gaussian underscores the validity of this approach and predicts good results.

Method 4: Extension by Imputation Missing color values can be imputed using a `Simple Imputation` method from Scikit-learn, which will impute all missing values with the median. Since the distribution will not be Gaussian, other imputation methods should be preferred. Another method for completing missing values is by using `k-Nearest Neighbors Imputation`. This method will impute all missing values using the Euclidean distance to the specified number of neighbors. The distribution will not be Gaussian either, that is why other extension methods should be preferred. However, a majority vote of nearest neighbors represents the color category well after cross-checking.

Apart from these vertical methods, there are two techniques of extending the dictionary horizontally, i.e. from color name to color value or from color value to color name, to find a color name's value and name.

Technique 1: Finding a Color Name's Value

For finding a color name's value, different kinds of methods exist. For a given color name such as *emerald green*, the corresponding color value of any color space may be searched in three different ways: by using a color survey, a data-driven search engine look-up or an online color name-value look-up.

Color survey The XKCD Color Survey[2] had participants survey name colors. The result of this experiment was a CND. A user of such a CND could easily look up a color name's color value. Such a dictionary look-up works for any new color name as long as many participants to the survey determine the color name's color value by assigning a color name to a color patch.

Data-driven search engine look-up The CND from the Color Thesaurus project was built using the first hundred *Google Image Search* results for a given color name in a data-driven search effort. Using the same approach, a home-bred script to login to *Google Image Search* to fetch the first one hundred images for color name *ultramarine* in English. The search query as described by Albrecht Lindner, Nicolas Bonnier and Sabine Süsstrunk[3] consists of the <color name> and the <color> in English [LLBS12]. For *ultramarine* this would mean to insert `ultramarine` and `color` in the search bar. In Figure 4.2, sample images for *ultramarine* are shown. The search query can be adapted for any other color. For example, the set of VIAN colors can be adapted to include new color names.

[2]`https://blog.xkcd.com/2010/05/03/color-survey-results/`
[3]`https://www.epfl.ch/labs/ivrl/research/image-mining/multi-lingual-color-thesaurus/`

All *white*, *black* and desaturated (*gray*) values are subtracted from the image by extracting their color values into a mask and removing the mask from the image layer.[4] Otherwise, when calculating the average color across all images, the *white* would tint the *ultramarine* into a diluted color value – a bias that is mitigated by building a mask on top of the image that isolates the color. As a result, we get a very unbiased, saturated *ultramarine* color with RGB color value (63,61,143), as seen in Figure 4.2.

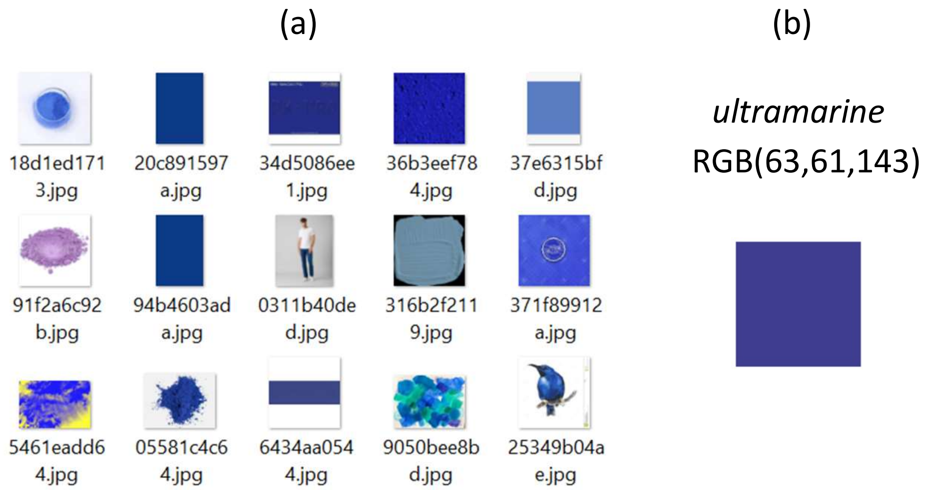


Figure 4.2.: (a) First 10 search results on *Google Image Search* for `ultramarine` and `color`; (b) averaged *ultramarine* from 100 first *Google Image Search* results for Ultramarine

Instead of *Google Image Search*, images can be downloaded from other search engines such as *Bing Image Search*, *Yahoo Image Search* or *Getty Images*. The *Google Image Search* engine was considered because its services are currently the most widely-used and complete. Besides, image posts can be leveraged from social media platforms. For instance, images can be sourced from Pinterest, Flickr or Instagram as well. Since these platforms' main focus is on tagged photos, they may be richer and more robust data sources for finding a color name's color value.

Online color name-value look-up There are web tools that take a color name as input and return a collection of color images and their HEX codes. An example of such a web tool is *Picular*[5]. *Picular* matches a given color name to a set of corresponding HEX codes. The user can decide which HEX code color for *hot pink* to take or the average of all HEX code colors could be computed. For colors that are not typically associated with *hot pink*, averaging the HEX colors would result in a biased *hot pink* color. Other online services such as *Wikipedia*'s infobox or websites such as *Resene*[6] can help with finding a color value for a given color name.

Technique 2: Finding a Color Name

Finding color names is more complicated than finding color values. Color names are not derivable by the use of mathematics, such as color values. Instead, color names are vaguer; they leave room for associations about colors, often making use of poetic imagery, real-life items or scenes from nature. An example of such topical groupings for the EPFL Color Thesaurus is described in Appendix A.5.4. This is because the accuracy of the color description is less important in everyday language – to acquire a vocabulary of all possible color value's color names is too expensive and time-consuming. Depending on the color space more than 15 million color values would have to be named.

[4]At an earlier stage, cropping the image to the centerpiece was a less generalizable technique to extract the wanted color.
[5]`https://picular.co/`
[6]`www.resene.co.nz/swatches`

Finding new color names can be done using web tools as a second approach. For example, the search is made easy using web tools such as *chir.ag*[7]. It takes a HEX code value of a color and returns the corresponding color name. Hence, the HEX codes for the most salient *ultramarine* color images can be entered for finding related color names. Another tool is *Color-Blindness' Color Name & Hue*[8]. It comprises 1'640 different color names. Furthermore, it is possible to enter RGB, HSB and HEX codes for finding related color names.

4.1.2. Classification Methods

Classification of colors (as opposed to regression) from a high-resolution color space down to a limited set of basic colors requires the specification of color tolerances around a basic color centroid. The classification can be done by utilizing simple statistical tools or complex data-driven algorithms. In this project, the former was conducted in combination with the latter in sequential order.

Once the set of color categories is determined, the labels need to be assigned from the color names as features. There are four techniques by which the color names as features were classified in a preliminary step into color categories as labels:

1. **Provided classification**
2. **Last-word classification**
3. **Manual classification**
4. **Mapped classification**

These classification methods are daisy-chained together in a quasi-sequential process, such that the color names in the Fully-Fledged Color Name Dictionary (FFCND) are categorized into one-to-many basic colors. At the start, the raw CND might have already provided groupings of colors into basic color categories. In such a case, a **provided classification** exists. For example, Werner's Nomenclature of Colors (see Appendix A.5.1) clusters color names together into ten different basic colors. Otherwise, **last-word classification** can speed up the categorization process: for two-named colors such as *purple blue*, the color is categorized using the second word *bue* while *blue purple* would figure in *purple*. For example, for *apple green* the corresponding basic color is *green*. All colors that are not derivable by a second-word match or identical match by last-word classification were classified using **manual classification**. **Mapped classification** can only be done if the data set is already labeled into categories that have a wider set than the color categories it should have eventually. For example, if the data set is labeled a mapping between the remaining color categories and the smaller number of color categories can be established for color category classification.

When classifying borderline colors into basic colors, the second basic color category can be derived easily from the original CND for example in color names such as *bluey green*, *orange red* or *yellow-green*. These are all two-gram-word color names for halfway hues that can be directly extracted as **provided classification**. Otherwise, based on **manual classification**, the colors are scanned by hand. Finding border colors and appending a second basic color to the first basic color category is another viable option. Note that quarter-way hues of the form *yellowish green* were not considered as borderline colors.[9]

[7] `http://chir.ag/projects/name-that-color/`
[8] `https://www.color-blindness.com/color-name-hue/`
[9] The EPFL Thesaurus has no color names that could designate quarter-way hues.

There are three different methods by which color names can be classified independently from each other by a machine learning classifier, if more than two class labels are present:[10]

- **Multi-class** (or **multinomial**) **classification**
- **Multi-label classification**
- **Multi-output-multiclass classification**

The multi-output-multiclass classification has each sample's labels attributed to non-binary properties. It requires that the number of properties (*movie* and *color*) and the number of classes per property (*Jigokumon* and *2001: A Space Odyssey*; *green* and *red*) are greater than two. This type of classification is a generalization of multi-class classification where only one property is considered and multi-label classification where only binary attributes are considered. Since multi-output-multi-class classification can be transformed into a multi-label problem, only multi-class and multi-label processing are checked.

Multi-class Classification In multi-class classification, every color name has exactly one-color category. For multi-class classification, the first column of a basic color category suffices, while for multi-label classification, both columns containing basic color categories need to be considered. From a multi-class perspective, the second column is a basic color category that is the second most preferred color category, such that color at the border of two color categories is assigned the two color categories with unequal weight. In multi-label classification, it is assumed that both color categories have equal weight, however.

The FFCND is extended to an Extended Fully-Fledged Color Name Dictionary (EFFCND) with two more columns for basic colors: *cat1* and *cat2* (see Appendix A.5.7). Just like the color names have color space values computed in the EFFCND, the *cat1* and *cat2* color names' values can be found by averaging all related color names' color values (or codes) for a given color space. An alternative method is to search a specific color name in the name column and visualize the color with the corresponding color values, for example, by filling a patch with sRGB values. These values are not included in the EFFCND to save space and memory but instead are evaluated on-the-fly in LAB space.

Multi-label Classification In multi-label classification, some color names can have two color categories. The colors at the border between two basic colors are challenging to categorize. Technically, a third or fourth, etc. basic color categorization could be created, meaning that the featured color would be situated at the border of three or four basic colors. For simplicity, we assume that all colors can be categorized into a maximum of two basic colors only.

Any two basic colors' number of borders for borderline colors is not equal to the number of all possible combinations of basic colors when given the set of basic colors. For instance, there exists two basic colors *orange* and *blue* that do not share a border due to their complementarity. *Orange* is at the opposite extreme of *blue*. Wittgenstein remarked on colors, *"They cannot blend into each other, just like you cannot simultaneously take the directions right and left at an intersection"* [G17]. Calling something *orange-blue* or *blue-orange* is equivalent to saying *gray*. The greater the number of color categories, the more borders exist.

Moreover, color borders in theory might not apply to color borders in practice. Taking Johannes Itten's six basic colors, it can be imagined that there are five borders to each color: the color's analogous colors on the color wheel plus *white*, *gray* and *black*. The color *yellow*'s border color are *orange*, *green*, *black*, *white* and *gray*. However, in practice, most people would classify the colors at the border between *yellow* and *black* as *green*, a muddy *moss green*, thus concluding that *yellow* has no border with *black* as seen in Figure 4.3. In contrast, *blue* shares a clear border with *black*, *gray* and *white* in theory as well as in practice.

[10] https://scikit-learn.org/stable/modules/multiclass.html

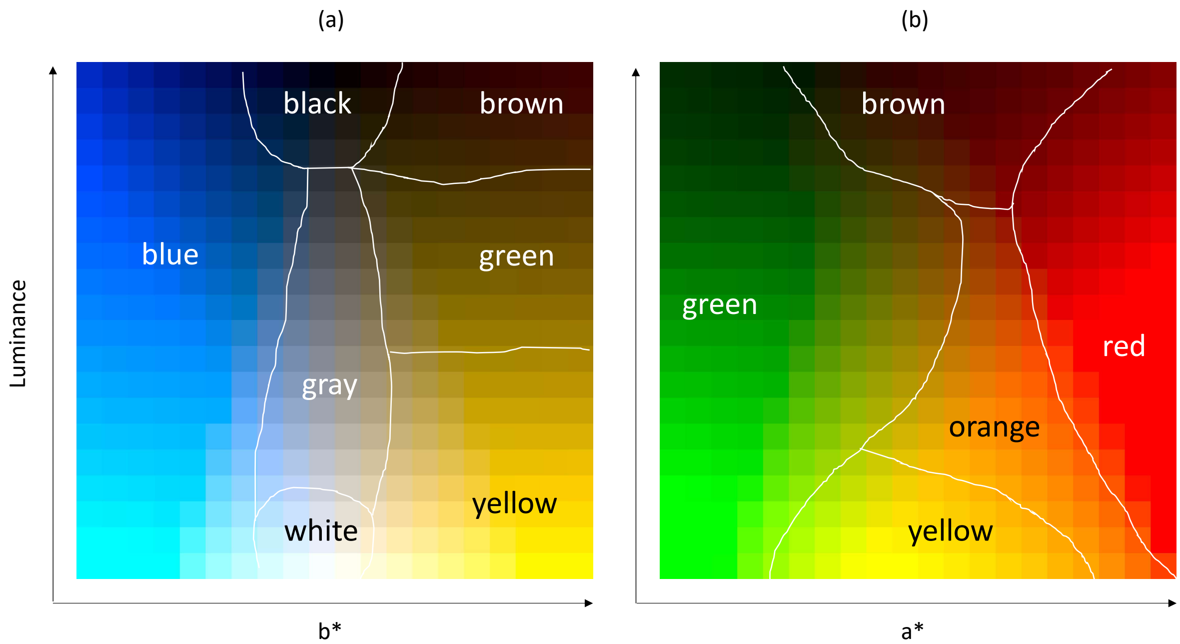


Figure 4.3.: Color borders for *yellow* in LAB space: the decision boundaries are non-linear; *yellow* is only visible on two matrices out of six where the first matrix (a) has a=0 and the second matrix (b) has b=128 for full ranges in the other dimensions.

The machine learning classifier partitions the LAB space into regions to determine the decision boundaries where each basic color is at the center of all its related color names. For a given new color value, it should be possible to determine which basic color it belongs to by looking at the new color value's basic color cluster. The decision boundaries dividing each basic color from another helps to delineate these regions. The decision boundaries can delineate these regions either by a hard (multi-class) or a soft (multi-label) border.

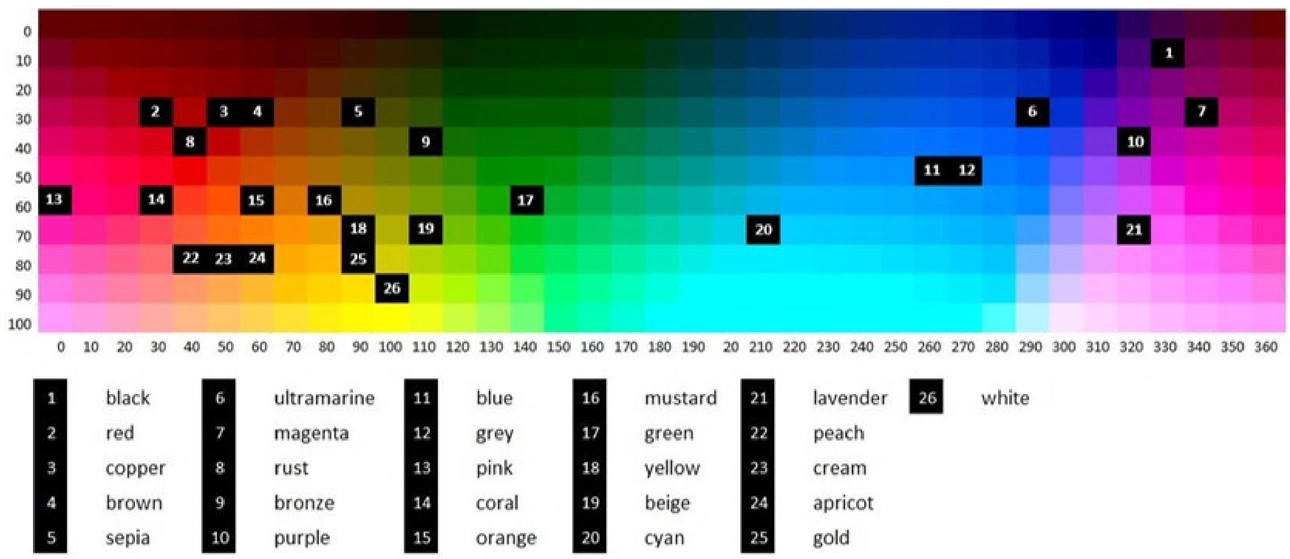


Figure 4.4.: EPFL Color Thesaurus class center averages VIAN colors in LCH 2-D space

The following phenomenon explains the hard border or soft border conceptualization of decision boundaries. At the border between *yellow* and *green*, it is typically difficult to decide to which the *yellow-green* color belongs. The user is either forced to decide whether the border is situated between the most *greenish yellow* color and the most *yellowish green* color. This hard decision boundary needs to be fixed. However, in the process of fixing, the border itself is a color calculated of these two colors and the user is left to decide about the color class again in an infinitely recurring loop. Another approach would be to allow for single colors at the border to be classified into both *yellow* and *green*. This multi-label approach could be extended to a bandwidth of colors at the border, classified to both *yellow* and *green* at the same time. An intrinsic property of such borderline colors is their almost unclassifiable character. This is due to the fuzziness of the boundaries between color categories. While the core center of a color category could be determined by surveying for a typical color value that depicts the color, the color category's boundaries may be a non-linear, non-concentric deformity around the color nucleus. In addition, the boundary may be not as clear-cut, but a gradual transition to the realm of another color category. The degree of fuzziness makes the difference between a hard and a soft border between color classes.

A common approach towards determining decision boundaries would be to determine the center of all basic colors from training points representing the basic color, calculating the distance between the centers and taking the perpendicular in the middle of that distance. Based on 3-D Voronoi tesselation of the LAB space, such a method seems promising for determining the decision boundaries between colors. However, such an approach will fail because in LAB space, the color regions are not equally sized. For example, taking the centers of *blue* and *violet* which are both basic colors, the distance and the perpendicular decision boundary at the center is computed. As seen from Figure 4.5, mis-classification of the *blue* points between the decision boundary and the center point of *violet* will lead to false positives for *violet* because *violet* is a minority class.

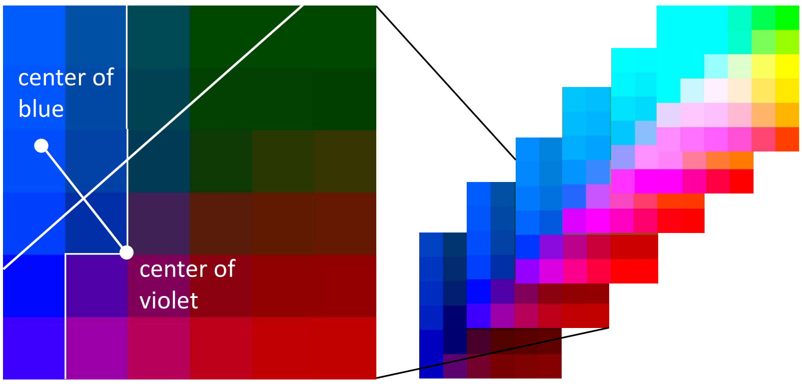


Figure 4.5.: *Blue* and *violet* color regions, their centers and potential decision boundaries in LAB space: a LAB color solid is divided by five to obtain six LAB planes with 0-100 luminance values at intervals of 20; one such plane is depicted with a decision boundary perpendicular to the line that intersects both blue and violet class centers which is less optimal than a non-linear decision boundary

Further, one could argue that the more training points there are, the better the trained decision boundary between basic color classes. However, in an imbalanced data set where more values of one class are present than another, the true center will not change the more data points are available. Hence, the decision boundary will remain the same as with less training points under the assumption that the true center was correctly calculated. Typically, more training points only serve to get closer to the true center. However, if the color class region is bigger or smaller across the LAB space in either case, the decision boundary will attribute more predicted colors to the smaller regions than they deserve.

For instance, in Figure 4.6 *blue* takes up a fairly large color region in LAB space as compared to *violet*. Training more or fewer color points will not resolve this fundamental problem. This method will assign *blue* a bigger region and *violet* a smaller region, as they are physically present and compare in reality.

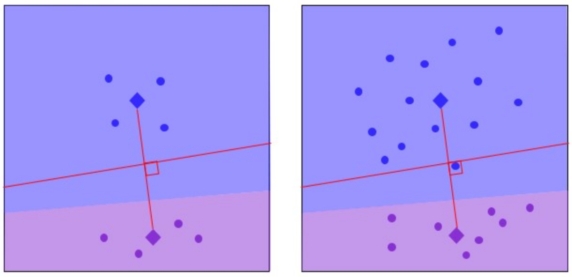


Figure 4.6.: Simulated decision boundary bias with a simple dissection - a few and many training points will not cause the decision boundary to realign

4.1.3. Machine Learning

For building a machine learning classifier, a data set with features and labels is needed. This data set will be an EFFCND where the features are the three-dimensional LAB color values with color names. The labels are basic color categories, which are categorical. The following steps need to be taken to preprocess the data set, train the classifier on the data set and test the classifier on the data set: for preparing the data, imputation and preprocessing stages need to be passed through. For training, the scoring methods, classifiers and metrics need to be selected before the training phase. Testing the classifier for model selection happens during the test phase. An overview of building and deploying machine learning model is shown in Figure 4.7.

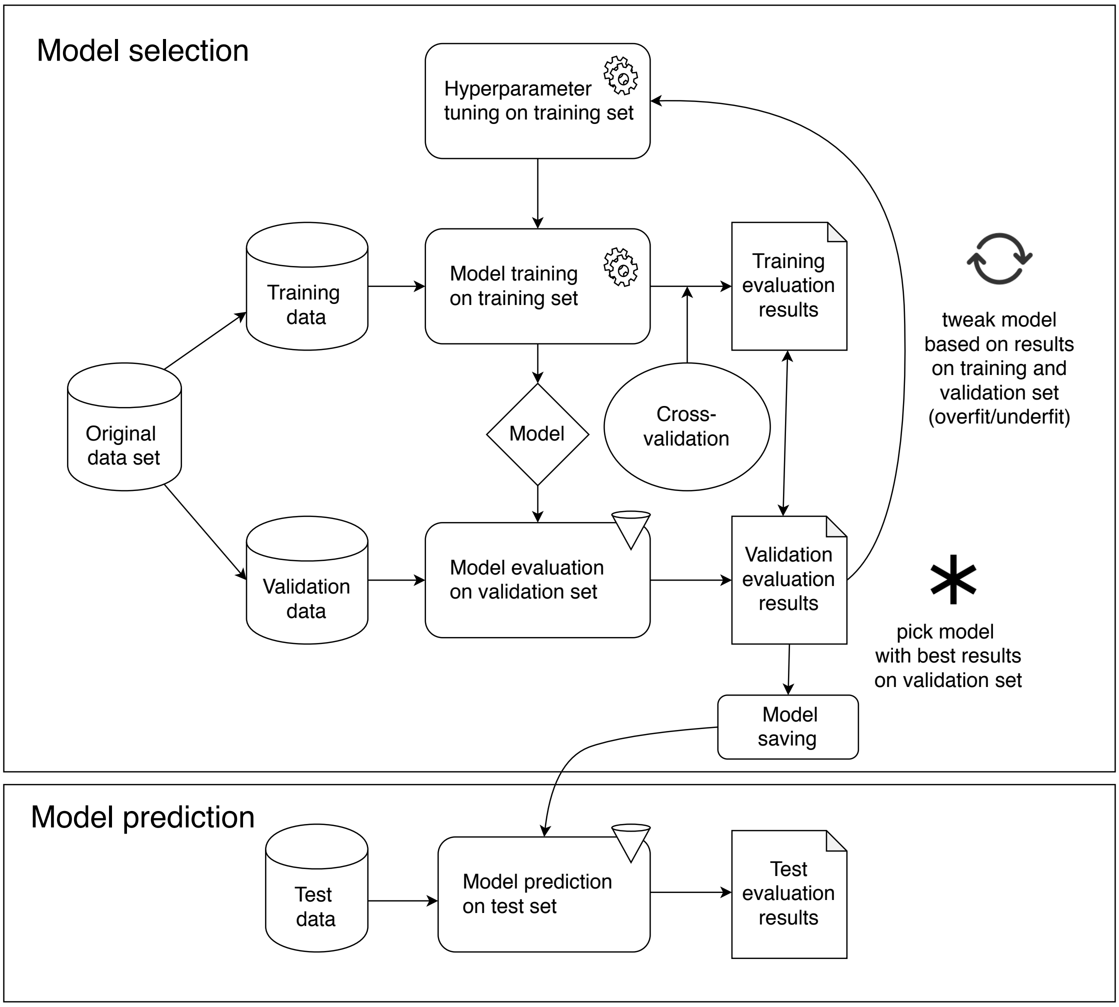


Figure 4.7.: Building and deploying a machine learning model: model selection is the process of figuring out which model performs the best. For this, the original data is split into a training and test set. Different classes of models are fit on the data and the models are evaluated and improved until the best model is found. The model predicts new data entries on the test data before the results are evaluated.

Imputation: For color categories that do not have enough instances for the estimator to learn, simulated color values need to be generated. A balance of class value counts is the objective because the accuracy metric is biased in the presence of imbalanced classes. Otherwise, the classifier could default to predicting the majority negatives 100% of the time and still get very high accuracy of 99% in the presence of a minority 1% positives. A classifier needs to correctly predict all minority class instances to be considered a good classifier. For imbalanced class problems, the more minority class instances are correctly predicted, the better the solution [HM15]. In addition, certain estimators require a minimal number of instances per class (KNN: 3 instances, CNN: 10’000 instances) or the performance will deteriorate. Hence, all minority classes were upsampled to obtain the same class frequency as the modal class by drawing random samples from each color category’s assumed normal distribution.

Alternatively, a threshold may also be determined to which all minority color categories with fewer color values can be upsampled. In general, for each color category, there need to be at least three basic colors for the classifier to learn the distribution. The more, the better. At the minimum, at least two more color values need to be obtained for color categories with only one color value. The fewer the samples per color category, the lower the standard deviation of its assumed Gaussian distribution. In this case, the CND can be extended by interval for two more values or extended by search to obtain at least one more color value before resampling. Based on a test for normality, the data suggest on majority vote a normal distribution of the color categories. Hence, the

Gaussian-based random generation of synthetic data coincides with the assumption that a color category follows a normal distribution, as proposed in Section 1.3.

Preprocessing: The LAB color values were standardized using the `Standard Scaler` that rescales the LAB value range to a range of $[-1, 1]^3$. The color categories were encoded using the `Label Encoder` for multi-class and the `Multi-label Binarizer` for multi-label classification. (Due to the nominal form of the labels, this task is originally a multi-class multi-output classification problem. However, it can be transformed into a multi-label classification problem using the `Multi-label Binarizer` in which case a superset of classifiers can be evaluated.)

Scoring: The scoring methods used were `test-train split`, `cross-validation`, and `Grid Search CV`. The data is shuffled because the samples are not independently and identically distributed if sorted by the color category. In the first case, the data is split into 90% training and 10% test set such that the test set consists of all possible color categories. In the second case, the data set was split into five samples, where four of the five samples constitute the training set. The remaining 1/5$^{\text{th}}$ was the test set such that each partition turns into the test set once in each round. The five results from each of the five possible splits of the entire data set were averaged. Classifiers that did not have a method for predicting probabilities could not be cross-validated. While `cross-validation` allows for tweaking each model class individually, `Grid Search CV` is less accessible to readjustments of hyperparameter values because the training for all estimators is done in one step. However, both `cross-validation` and `Grid Search CV` with five folds execute fundamentally the same action. For reproducibility, a random state was set when possible.

Classifiers: 23 multi-class machine learning classifier classes[11] and six multi-label machine learning estimator classes[12] were trained using (semi-)supervised learning since the labels in the original data set are given. Some of the classifiers employ a One-vs-Rest strategy, others use a One-vs-One strategy, even other classifiers use both methods, see Appendix B.1.

Metrics: All fourteen multi-class metrics[13] and ten multi-label metrics[14] were applied for scoring the training and test set. They all follow the convention that higher return values are better than lower return values. Different metrics evaluate different characteristics of the classifier [HM15]; however, the standard metric chosen for maximization was any kind of accuracy score, because accuracy is the most often used performance metric in machine learning experiments. The classifier does not predict some labels in the data set. Hence, the metric fails to determine the performance score for such color categories and is removed for this reason.

Training phase: During training, the classifiers learn to draw decision boundaries between basic color categories. They learn what color names are attributed to which basic color categories for future inference. The classifier could then take a new color name and predict its most likely basic color category based on what it has learned.

[11] LogisticRegression, LogisticRegressionCV, RidgeClassifier, RidgeClassifierCV, GaussianNB, KNeighborsClassifier, NearestCentroid, RadiusNeighborsClassifier, LinearSVC, DecisionTreeClassifier, ExtraTreeClassifier, ExtraTreesClassifier, RandomForestClassifier, LinearDiscriminantAnalysis, QuadraticDiscriminantAnalysis, LabelPropagation, LabelSpreading, GaussianProcessClassifier, SVC, SGDClassifier, Perceptron, PassiveAggressiveClassifier, GradientBoostingClassifier.

[12] KNeighborsClassifier, RadiusNeighborsClassifier, DecisionTreeClassifier, ExtraTreeClassifier, ExtraTreesClassifier, RandomForestClassifier

[13] Balanced accuracy score, Matthews correlation coefficient, fscore (micro, macro, weighted), fbeta score (micro, macro, weighted), precision (micro, macro, weighted), recall (micro, macro, weighted). Precision and recall, when applied individually, only take into account the proportion of True positives and neglect the True negatives. These two metrics are generally used when there is a class imbalance problem. True negatives are equally important because both true positives and true negatives need to increase to obtain correct predictions with a classifier. They need to be viewed in conjunction with each other to be robust.

[14] Accuracy score, fbeta score (macro, micro, samples, weighted), hamming loss, jaccard (macro, micro, samples, weighted).

Test phase: The prototype model with the best score from the training data set is applied to the test set/s and averaged when using cross-validation. The test set's score is computed by measuring the difference between the model's predicted labels and the true labels from the test set. The best model is located at the minimum of the error score or maximum performance score on the test set. In case of an overfit and if such classifier parameters permit the regularization of an overfit situation, the hyperparameter values were specified such that the overfit pitfall was mitigated.

Two examples of machine learning cases are described: one for multi-class, the other for multi-label classification. For multi-class classification, all 28 VIAN color categories from the EEFFCND Thesaurus-VIAN were fit. Since the EEFFCND Thesaurus-ITTEN has only six color categories with at most two labels, the multi-label classification task was conducted on this data set.

4.1.4. An Example: Multi-class Thesaurus-VIAN

All 23 multi-class machine learning classifiers were trained on the EFFCND Color Thesaurus-VIAN data set, taking the LAB color values as a feature and the 28 VIAN basic colors as labels or all 720 dictionary colors as labels. The VIAN color categories are very imbalanced – *green* has more than 120 discrete LAB values, while *apricot* has only one value. Hence, the data set was upsampled to include 3'419 rows in total. For each classifier, all possible hyperparameter values were screened one-by-one: some were left as the default, some were set to a constant, others were optimized using a range of possible values. For further optimization, these ranges were revisited.

For multi-class classification, the data split's ratio is 90:10 or for 5-fold cross-validation 80:20. The evaluation metric is either accuracy or the F1-score or a variant of these metric classes, depending on which metric scores higher. All classifiers without a method for predicting probabilities were removed in 5-fold cross-validation because they could not classify under such conditions. In Table 4.1, all 23 multi-class classifiers' balanced accuracy performance is presented. In Table 4.2, the results from 5-fold cross-validation are shown.

Rank	Classifier	Training	Test	Hyperparameters
1	GaussianNB	0.742	0.751	default
2	RandomForestClassifier	0.798	0.747	n_estimators: 69, max_depth: 8
3	Extra Trees Classifier	0.871	0.745	n_estimators: 20, max_depth: 1
4	GradientBoostingClassifier	0.957	0.738	learning_rate: 0.1, n_estimators: 180, max_depth: 2, init: zero
5	QuadraticDiscriminantAnalysis	0.751	0.738	default
6	LabelSpreading	0.735	0.725	n_neighbors: 10, alphas: 0.2
7	SVC	0.695	0.724	C: 1, gamma: scale, shrinking: True, class_weight: None, decision_function_shape: ovo
8	KNeighborsClassifier	0.998	0.723	n_neighbors: 9, leaf_sizes: 6
9	GaussianProcessClassifier	0.694	0.719	default
10	LabelPropagation	0.735	0.718	n_neighbors: 10
11	LogisticRegressionCV	0.675	0.707	solver: saga, multi-class: auto
12	LogisticRegression	0.665	0.705	solver: saga, multi-class: auto
13	LinearDiscriminantAnalysis	0.645	0.685	solver: lsqr, shrinkage: auto
14	DecisionTreeClassifier	0.682	0.681	max_depth: 7
15	ExtraTreeClassifier	0.931	0.673	max_depth: 15
16	RadiusNeighborsClassifier	0.998	0.656	radius: 300, leaf_size: 100
17	LinearSVC	0.638	0.644	multi_class: crammer_singer
18	NearestCentroid	0.629	0.631	shrink_threshold: 1
19	PassiveAggressiveClassifier	0.509	0.561	C: 1, class_weight: balanced
20	SGDClassifier	0.380	0.416	penalty: l2, learning_rate: adaptive, class_weight: None, eta0: 0.8
21	RidgeClassifierCV	0.355	0.352	class_weights: none
22	RidgeClassifier	0.301	0.310	alpha: 50000
23	Perceptron	0.250	0.246	penalty: l2, class_weight: balanced

Table 4.1.: Evaluation metrics for multi-class classifiers on data split into test and training set by a 90:10-ratio. The performance metric is balanced accuracy.

The top-performing classifier for a test-train split during the test phase was the Gaussian Naïve Bayes classifier as seen in Table 4.1. The best model for cross-validation is Quadratic Discriminant Analysis as seen in Table 4.2. Both classifiers have no substantial hyperparameter values to optimize or to specify as constants. Furthermore, both classifiers draw decision boundaries from fitting Gaussian conditional densities to each class of the data using (Naïve) Bayes' rule such that both assume that the likelihood of the features is bell-shaped.

The likelihood of the features is assumed to follow a normal Gaussian distribution, see given Equation 4.1:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \tag{4.1}$$

The mean μ and the standard deviation σ (or the corresponding variance σ^2) of each color category is estimated using maximum likelihood, i.e. frequency counting before determining the mode.[15]

[15] https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes

Rank	Classifier	Training	Test	Hyperparameters
1	QuadraticDiscriminantAnalysis	0.750	0.736	default
2	GaussianNB	0.746	0.735	default
3	GradientBoostingClassifier	0.998	0.719	learning_rate: 0.1, n_estimators: 180, max_depth: 2, init: zero
4	RandomForestClassifier	0.998	0.712	n_estimators: 66, max_depth: 10
5	ExtraTreesClassifier	0.998	0.711	n_estimators: 15, max_depth: 10
6	SVC	0.750	0.706	C: 20, gamma: auto, kernel: rbf, shrinking: False, class_weight: None, decision_function_shape: ovo
7	LabelPropagation	0.998	0.690	n_neighbors: 17
8	LabelSpreading	0.998	0.686	n_neighbors: 6, alphas: 0.2
9	GaussianProcessClassifier	0.697	0.678	default
10	LogisticRegressionCV	0.678	0.665	solver: saga, multi-class: auto
11	LogisticRegression	0.666	0.658	solver: saga, multi-class: auto
12	RadiusNeighborsClassifier	0.998	0.650	radius: 300, leaf_size: 100
13	LinearDiscriminantAnalysis	0.652	0.642	solver: lsqr, shrinkage: auto
14	KNeighborsClassifier	0.998	0.630	n_neighbors: 24, leaf_sizes: 21
15	ExtraTreeClassifier	0.923	0.621	max_depth: 15
16	DecisionTreeClassifier	0.433	0.408	max_depth: 10
17	PassiveAggressiveClassifier	0.165	0.175	C: 0.01, class_weight: balanced
18	Perceptron	0.166	0.170	penalty: l2, class_weight: None
19	SGDClassifier	0.163	0.166	penalty: elasticnet, learning_rate: adaptive, class_weight: balanced, eta0: 1.1

Table 4.2.: Evaluation metrics for multi-class classifiers on 5-fold cross-validation. The performance metric is balanced accuracy.

Performance: Gaussian Naïve Bayes' balanced accuracy score yielded a very high-performance score of 75.14% for the test set compared to an almost equally high 74.25% on the training set with a test size of 10%. For 5-fold cross-validation, Quadratic Discriminant Analysis's balanced accuracy score was 73.58% on the test set and 74.99% on the training set. Quid, both models generalize very well to unseen data.

The best model's decision boundaries are visualized in Figure 4.8 by projecting the 3-D LAB color values from the EFFCND Thesaurus-VIAN data set into 2-D space based on t-SNE's[16] component transformation function. In a One-vs-Rest way, color category *amber* serves as an example against all non-*amber* color points. The Gaussian Naïve Bayes' decision boundaries circumscribe the class center precisely, whereas the multivariate Gaussian distribution of a Quadratic Discriminant Analysis classifiers is more fuzzy and permeable.

[16] The Scikit-learn method `t-distributed Stochastic Neighbor Embedding` is used for dimensionality reduction, `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html`

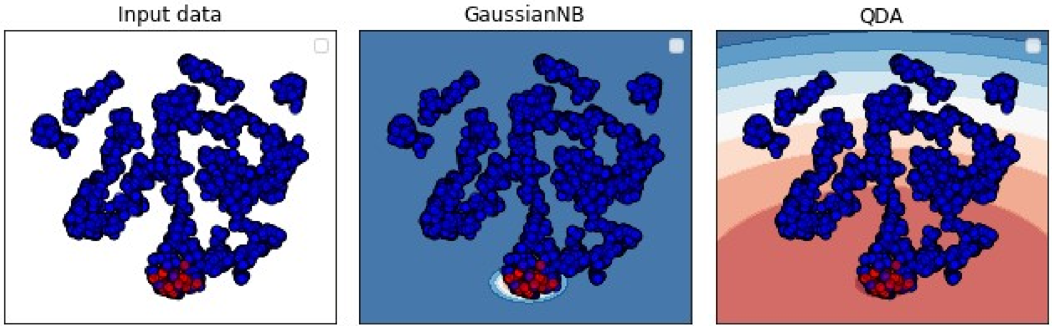


Figure 4.8.: Classifier comparison: input data are the 3-D color triplets projected into a 2-D plane. Gaussian Naïve Bayes and Quadratic Discriminant Analysis as best models are capable of drawing decision boundaries along all *amber* color categories where the former is more precise than the latter.

Efficiency: Due to its lack of hyperparameters, the model is efficient in training and can scale up well. It took 0.10 seconds to train the Gaussian Naïve Bayes classifier for the train-test split. The Quadratic Discriminant Analysis classifier needed 0.94 seconds. Thus, both classifiers rank equally on the performance and efficiency scale. With this speed, it can be concluded that both would integrate into a well-rounded user-experience.

4.1.5. An Example: Multi-label Thesaurus-ITTEN

Johannes Itten[17] defines six basic colors that may be extended to twelve basic colors [IB70, Itt74]. The six or twelve basic colors are called ITTEN colors, because Johannes Itten uses these to explain his color contrasts [Itt74]. For classification into all six kinds of color contrasts defined within the scope of this thesis, the training of a set of six basic colors is sufficient. The six colors are derived by Johannes Itten by combining together the three primaries *red*, *yellow* and *blue* and the three secondaries *orange*, *green* and *violet*. The secondaries are obtained by calculating the sum of each possible pair of primaries (*yellow* + *red* = *orange*; *green* + *blue* = *green*; *red* + *blue*= *violet*). When including the six tertiaries, *red-orange*, *yellow-orange*, *yellow-green*, *blue-green*, *blue-violet*, *red-violet*, twelve colors remain which Johannes Itten situates by location on a 12-hue color circle [Itt74]. The tertiaries are obtained by mixing a primary with a secondary color (*yellow+orange* = *yellow-orange*, *red+orange=red-orange*; *red+violet=red-violet*; *blue+violet=blue-violet*; *blue+green=blue-green*; *yellow+green=yellow-green*).

The EFFCND used in this example of multi-label classification is the Thesaurus-ITTEN data set with LAB color values and two columns with six ITTEN basic color categories as multi-labels. The ITTEN color categories are imbalanced. Hence, the data set was upsampled to include 1'035 rows by 26 columns in total.

A CND such as the EPFL Color Thesaurus can be labeled with basic colors. The VIAN color categories with 28 VIAN colors from the EFFCND Thesaurus-VIAN were mapped to the ITTEN colors to create the EFFCND Thesaurus-ITTEN. For further details, please consult Appendix A.3. In addition, *purple* is renamed to *violet* because of Johannes Itten's slightly different color terminology. Furthermore, all non-categorizable colors such as *black*, *white* or *silver* are left blank. A preliminary exploratory data analysis of the processed CND yields that in contrast to the orangey bias in the VIAN colors, most of the EFFCND Thesaurus-ITTEN dictionary colors are *blue*, the complementary color to *orange* (see Appendix A.3). *Yellow* is the least represented ITTEN basic color category.

[17] Johannes Itten, a Swiss-born teacher of Visual Art, was most notably practicing his color theory at the Bauhaus School of Art and Design. He became an eminent figure in color theory after the publications of his books: *The Elements of Color* and *The Art of Color*.

In multi-label classification, the data split's ratio is 90:10 or for 5-fold cross-validation 80:20. Then, the data were fit by multi-label classifiers. Only the Radius Neighbor Classifier had no method for predicting probabilities and was removed for 5-fold cross-validation. Table 4.3 shows the fbeta macros scores for the six multi-label estimators, which scored higher than the accuracy metric for a train-test split. In addition, Table 4.4 presents the fbeta weighted scores for 5-fold cross-validation. Fbeta's beta value equals to 1, for which case the score is equal to an F1-score. The *weighted* attribute in the fbeta weighted score accounts for class imbalance, while *macro* for the fbeta macro score means that for each label the unweighted mean was calculated.[18]

Rank	Classifier	Training	Test	Hyperparameters
1	ExtraTreesClassifier	0.319	0.927	n_estimators: 10, max_depth: 25, min_samples_split: 2, min_samples_leaf: 1
2	KNeighborsClassifier	0.996	0.907	n_neighbors: 8, leaf_sizes: 1
3	RandomForestClassifier	0.892	0.877	n_estimators: 300, max_depth: 6, min_samples_split: 10, min_samples_leaf: 1
4	ExtraTreeClassifier	0.935	0.870	max_depth: 14
5	DecisionTreeClassifier	0.991	0.843	max_depth: 13
6	RadiusNeighborsClassifier	0.996	0.128	radius: 300, leaf_size: 1

Table 4.3.: Evaluation metrics for multi-label classifiers on test-train-split. The performance metric is the fbeta macro score.

Rank	Classifier	Training	Test	Hyperparameters
1	RandomForestClassifier	0.911	0.783	n_estimators: 300, max_depth: 6, min_samples_split: 5, min_samples_leaf: 1
2	ExtraTreeClassifier	0.966	0.730	max_depth: 13
3	KNeighborsClassifier	0.996	0.723	n_neighbors: 6, leaf_size: 6
4	ExtraTreesClassifier	0.993	0.723	n_estimators: 15, max_depth: 15
5	DecisionTreeClassifier	0.988	0.714	max_depth: 12

Table 4.4.: Evaluation metrics for multi-label classifiers on 5-fold cross-validation. The performance metric is the fbeta weighted score.

The top-performing classifier for a test-train split of ratio 90:10 was the Extra Trees Classifier, also known as Extremely Randomized Trees, as seen in Table 4.3. In 5-fold cross-validation with 5 rounds of splits of 80:20, the Random Forest Classifier scored the highest, as shown in Table 4.4. Both classifiers are ensemble methods, thus they are both meta estimators that fit several decision tree classifiers on various sub-samples of the data set and uses averaging to improve the predictive accuracy and control for over-fitting.

Performance: Extra Trees Classifier's macro F-beta measure[19] yielded the highest performance score of 92.71% for the test set as compared to a meager 31.86% on the training set with a test size of 10%. For 5-fold cross-validation, the much more reliable result of the Random Forest Classifier's F1-weighted performance score was resolved to 78.31% on the test set and 91.06% on the training set.

[18] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.fbeta_score.html`
[19] Fbeta's beta was set to 1, which is equal to the F-score.

Multi-labels are positioned in a very high-dimensional and sparse class space as compared to multi-class labels. Ensemble methods are a powerful way of dealing with these high dimensions. Thus, they have gained popularity in the multi-label literature despite their slow computation speed [RBHP11]. Ensemble methods use multiple classification algorithms on the data. Hence, their superiority in high-dimensional space can be attributed to the increased flexibility in learning the real classification pattern.

Efficiency: Since the splits of the Extra Trees Classifier are randomized, it typically requires less time to compute an Extra Trees Classifier than a Random Forest Classifier. The Extra Trees Classifier needed 0.47 seconds to train, while the Random Forest Classifier was much slower with 35.07 seconds.

4.2. Task 2: Image Similarity Calculation

The ERC FilmColors project's Visual Annotation webtool (VIAN) features a function that is able to extract a color palette from an image in hierarchical order. At the lowest level are the most colors extracted for the color palette. At the highest level, all colors were merged to form a single color average for the image. The task is to get a color palette and compare it to other color palettes based on their similarity to the first color palette. For the first color palette, the distance is calculated to all other color palettes to determine the top-k nearest color palettes.

Task 2: Implement a method to compute the distance between hierarchically structured color palettes.

The different steps for executing Task 2 are outlined in Figure 4.9. From the set of video frames, color palettes are extracted. The most representative lowest level of the hierarchical color palette is retained, consisting of a flat color palette. The color palettes are transformed into histograms for which the pairwise distances are calculated using a distance metric. The distance metric represents how close the color distribution of two video frames is to each other, which can also be thought of as a measure of similarity. The complete matrix holding all similarity values for all pairs can be queried for an input video frame.

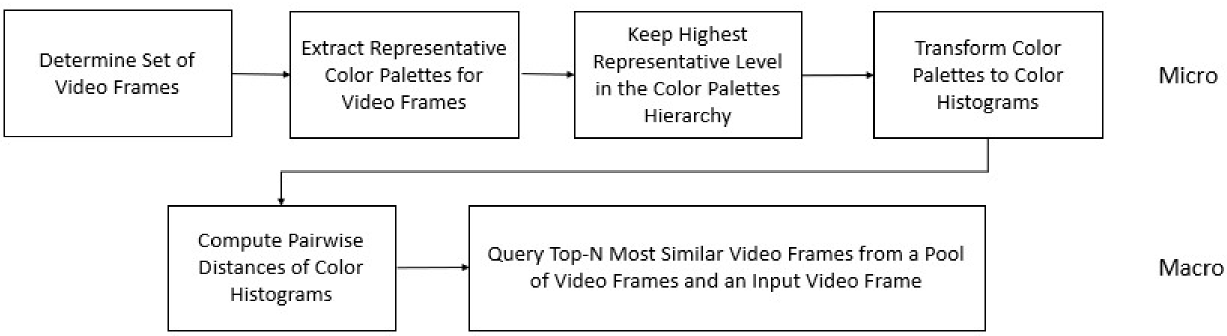


Figure 4.9.: Algorithm for processing video frames to color palettes and to color histograms

The procedure is based on the assumption that for similar images, colors in a histogram close to each other are also closer together in another histogram. This idea is leveraged to find the most similar video frame or color palettes for a given video frame or color palette. The emphasis is on video frames because it makes more sense for the user to find similar images of a video clip instead of finding the most similar color palettes for a given color palette.

4.2.1. Histogram Conversion

Each video frame is an image from which a color palette is extracted. The color palette extracted is the lowest row in the hierarchical color palette, because all higher rows are aggregates of the lowest row. The lowest level represents the original image the best. This color palette is loaded as an image, then converted into a histogram that shows the color channels as a distribution in three dimensions of the image (BGR, RGB, LAB). Each color palette's histogram is dense as opposed to sparse because the employed function works well with dense histograms. For comparing the performance of similarity, color histograms are directly extracted from the original video frame image. At each step, two histograms are compared to each other. The total number of pairwise combinations of histograms is $(n * n - n)/2$, which will be extended by symmetry and a diagonal of 1 to a $n * n$ matrix. In the case of 100 images, $100 * 100$ values need to be computed to fill up the matrix. This high number of values can easily become a considerable challenge in terms of performance.

4.2.2. An Example: *Jigokumon*'s Movie Picture Similarity

The color motion picture film *Jigokumon* ("Gate of Jigoku", 地獄門) with project id 7 was chosen from the ERC FilmColors MFD on the grounds of director Teinosuke Kinugasa's masterful use of color during the film-making of *Jigokumon* in 1953. Martin Scorsese dubbed the dazzling color cinematography as "one of the ten greatest color achievements in world cinema" [20].

The movie was converted into a sequence of 569 video frames and their pairwise similarity performed on their color palette-histogram representations were computed. Depending on the combinatorial method, combination, permutation or cartesian product, the amount of time needed to process all $569 * 569 = 323'761$ combinations of frame-pairs could become a strain on performance. In terms of speed, all three methods were experimented with one-by-one to identify the fastest approach on ten sample frames. The results of these experiments are shown in Table 5.1.

Method	Time (in seconds)
Combinations	1.48
Permutations	2.82
Cartesian Product	3.14

Table 4.5.: Performance of combinatorial methods

As expected, using combinations to iterate through all possible combinations of pairs is the fastest approach, see Figure 5.1. (In terms of memory, nothing can be done - either the user has enough memory to process the frames or an out-of-memory error will be thrown at any given point in time.) These combinations of histograms are then evaluated pair by pair. Either the pair is further away or smaller apart from each other. Hence, an appropriate measure needs to be found to calculate the distance between each pair of histograms. There are four possible metrics of distance or similarity between histograms which are correlation, chi-square, intersection and Bhattacharyya distance.[21]

The correlation metric is defined in Equation 4.2:

$$d(H_1, H_2) = \frac{d((H_1(I) - H_{-1})(H_2(I) - H_{-2})}{\sqrt{\sum_I (H_1(I) - H_{-1})^2 \sum_I (H_2(I) - H_{-2})^2}}, \tag{4.2}$$

where $H_{-k} = \frac{1}{N} \sum_J H_k(J)$ and N is the total number of histogram bins.

[20] https://eurekavideo.co.uk/movie/gate-of-hell/

[21] In fact, a Hellinger coefficient is computed which is related to Bhattacharyya distance.

The chi-square metric is defined in Equation 4.3:

$$d(H_1, H_2) = \frac{(H_1(I) - H_2(I))^2}{H_1(I)} \tag{4.3}$$

The intersection metric is defined in Equation 4.4:

$$d(H_1, H_2) = \sum_I min(H_1(I), H_2(I)) \tag{4.4}$$

The Bhattacharyya metric is defined in Equation 4.5:

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{H_{-1}H_{-2}N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}} \tag{4.5}$$

As input, three images from the movie *Jigokumon* are given: frames 45442, 45479 and 45472. The corresponding color palette was extracted from the images, as shown in Figure 4.10.

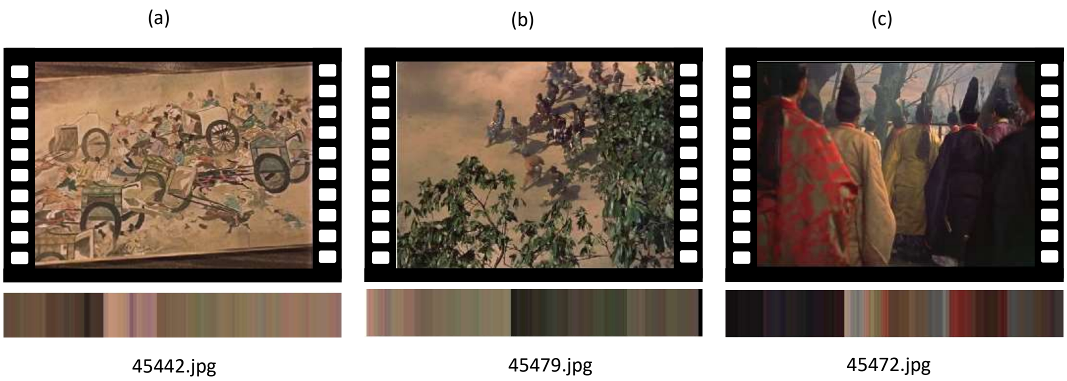


Figure 4.10.: Video frames from the movie *Jigokumon* - frames (a) and (b) are similar; frames (a) and (c) are perceived as dissimilar. This is mostly due to the redness in frame (c)'s color palette.

The first image (a) serves as a base image to which the other two images are compared. The base image is very similar to the second image (b) and very dissimilar to the third image (c) in terms of color distribution. For correlation, 1 means a perfect positive match, -1 means a perfect mismatch. The range 0-1 represents the strength of the match. For chi-square and Bhattacharyya, the opposite holds, the lower the metric, the more similar the match.

Method	Image 1- Image 1	Image 1-Image 2	Image 1-Image 3
Correlation	1	1	-0.001
Chi-square	0	513’200	398’200
Intersection	400’000	2’000	2’000
Bhattacharyya	0	0.995	0.997

Table 4.6.: Performance of distance metrics

The best metric should reflect the similarities between the images in equal proportion; (a) Image 1 should differ only slightly in similarity as compared to (b) Image 2 and Image1 should differ starkly from (c) Image 3. When comparing the different methods in Figure 4.6, only correlation succeeds to meet this criterion. While intersection and Bhattacharyya distance both express the higher similarity between the first two images and the lower similarity between the first and last images in relative terms, they fail to make it clear that the similarity between the first two images is almost equally high in absolute terms. Chi-square fails to manifest any of these relations. Hence, we settle for correlation as the most reliable similarity metric.

4.3. Task 3: Color Contrast Estimation

An image is made of a combination of colors that can form contrasts of warm and cold, dark and light, strong and weak. Such color contrasts can be analyzed using an image's color palette. For a given color palette, the color palette is classified into all possible color contrast categories. For each contrast, a color palette can or cannot manifest a palette contrast.

Task 3: Classify the patterns within the color palettes in different types of color contrasts defined by the ERC FilmColors project

The touchstone of this task is embodied by the challenge of correlating visual descriptions from the realm of arts with the instrumental charting of colors from the natural sciences. While the former is understood readily by many people in a general way, the latter is more preoccupied with describing a color with maximum accuracy for communicating colors unambiguously.

4.3.1. Color Contrasts

Since simultaneous contrast, successive contrast and aerial perspective do not lend themselves well for categorizing images due to their conceptual fuzziness, only six out of nine color contrasts remain for further learning. The reasons why the other kinds of color contrasts were not taken into account are described below.

For simultaneous contrast, if the eye generates the complementary color spontaneously such as in experiments where *gray* looks *reddish gray* if it is surrounded by *green* or the tendency of pure colors to make other colors look like their own complement [Itt74], this cannot be objectively calculated because the phenomenon occurs in the human visual system outside of what is computable by a machine. What color is established inside the eye is not accessible by a computer.

The same goes for successive contrast. If this is the effect of previously-viewed color fields on the appearance of currently-viewed test fields, this can only occur as a subjective phenomenon external to what is objectively determinable in an image. In Johannes Itten's words, "If we gaze for some time at a *green* square and then close our eyes, we see, as an afterimage, a *red* square. [...] The Eye posits the complementary color; it seeks to restore the equilibrium of itself" [Itt74].

As for aerial perspective, there is another effect generated when the viewer increases the distance to an image with fore- and background. In such a case, the effect is that the contrast between the fore- and background decreases the further the viewer creates a distance between the image and the viewer. On VIAN, the sample pictures do not contain successive pictures to show this effect. Instead, independent pictures are shown with a camera perspective from the top. Since this is a perspective and not a contrast, it does not fall into the task description's proposal to handle contrasts only.

Hence, the color contrasts left are six in kind: a contrast of hue, light-dark contrast, cold-warm contrast, complementary contrast, a contrast of saturation and contrast of extension. For further details about color contrasts, please refer to Appendix B.2. Figure 4.11 illustrates all applicable color contrasts by using three exemplary color schemes per color contrast. Sample images classified into each of these contrasts may be viewed on the ERC FilmColor's Timeline of Historical Film Colors webpage [22].

[22] https://filmcolors.org/

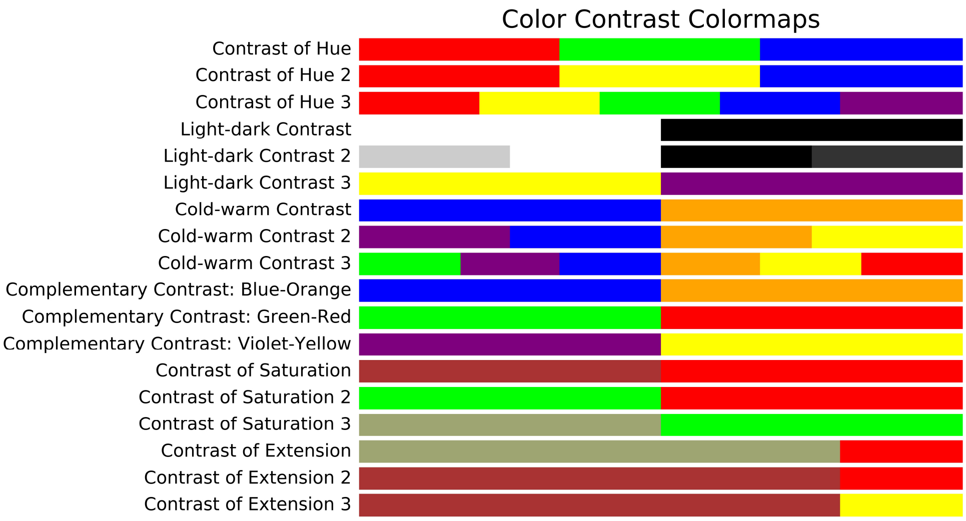


Figure 4.11.: Applicable color contrasts illustrated with three examples per contrast as color maps

4.3.2. Classification Rules

Based on the artistic definitions and considerations about color contrasts and their applicability to data-driven image processing, the following principles were employed to color palettes to scientifically determine whether contrast is present in a color palette or not. The color contrasts were derived from color models, which suggest that colors follow perfect symmetric order. For example, Johannes Itten's color star and the color wheel position colors are situated at equal angles around the color wheel. However, the LAB color space is a comparatively deformed and curved horseshoe where colors may be quantized to much greater precision than in the artistic color models. They are adapted to a semantic space to fit a proposed mathematical definition.

The rules that determine the color contrast concepts can all be explained using the LCH color space. The LCH color space consists of three color channels: luminance, chroma and hue.[23] From this systemization, an automatic classification of images into all color contrasts is enabled.

- **Luminance (l)**: light-dark contrast
- **Chroma (c)**: contrast of saturation, contrast of extension (specialization of contrast saturation)
- **Hue (h)**: contrast of hue, cold-warm contrast (generalization of contrast of hue), complementary contrast (specialization of contrast hue)

When viewing images, rarely do the desaturated, light or only dark colors juxtaposed next to each other have a contrasting effect. If ever, their effect is derived from positioning the saturated next to the desaturated, as seen in the contrast of saturation and its special case, the contrast of extension; dark next to light colors, as seen in light-dark contrast. However, for other contrast categories such as the contrast of hue, warm-cold contrast, and complementary contrast, the contrast only has a direct bearing when saturated, and medium luminance values are chosen together with a play on the portion size of the color pixels occupied within an image.

For a cold-warm contrast, complementary contrast, a contrast of saturation and extension, a saturation level of above 30 and non-dark values defined with a luminance above 25 were considered. For forming a contrast of hue, a cold-warm contrast and complementary contrast, the color must have a minimum size of 5% to be seen as

[23] Instead of taking the l-channel, one could imagine taking the v-channel in HSV colors. But since we compute color values in LAB, the HSV color system was not a viable option.

a color of importance next to other colors for gaining contrasting momentum, unless the color is saturated with a saturation of above 75. Independent of size, if the color is too dark, no contrast of hue, cold-warm, complementary, saturation and extension can ever evolve from such colors. Hence, they were removed from the equation.

Due to the extraction method of color palettes that represent the images, they cannot account for small color pixels however important they may seem for viewing different contrasts in an image. Such small, local information may also be lost during the annotation process out of negligence.

The problem with ITTEN colors is the absence of *brown* as a color category and the existence of *brown* in many images. Since the classifier interprets *brown* as *orange*, the portion of *orange* in the image is a multiple of what a human annotator may see as *orange*. The human annotator cannot see *brown* as a legitimate color for the constitution of a contrast of hue, a cold-warm contrast or a complementary contrast, hence a ceiling is defined to clip off *oranges* of size bigger than 20%. Since large amounts of saturated *orange* usually do not abound in video images, fixing a ceiling can be a quick fix. A possibility is to include *brown* as a color category for training the estimator only to remove this color category again during the decision-making process.

1. **Contrast of hue** For a contrast of hue, if there are at least three different ITTEN colors present in a color palette, the color palette is said to display a contrast of hue. Added to this are only color pixels that are not too dark or too desaturate or *browns* classified as *orange*. For small-sized colors of less than 5%, they need to be saturated enough to have a contrasting effect.

2. **Light-dark contrast** For a light-dark contrast, simultaneous luminance values of more than 75 and less than 25 on a 0-100 range in the color palette seemed to be a good starting point for specifying a light-dark contrast. However, there were too few light values for typical images that display a light-dark contrast. Hence, this light value was decreased to 55. In practice, it is not enough to just 'have' dark values; they must cover at least 40% of the image to have an effect, as seen in a light-dark contrast. Most typically, the stark contrast between much darkness and small sparks of light leads to a light-dark contrast, while the reverse is not true. In the middle, it seems that equal portions of dark and light are acceptable, too. However, certain images that contain a light-dark contrast go undetected. For small and light color pixels, the color palette extraction procedure diminishes their lightness by averaging them off into a more somber color.

3. **Cold-warm contrast** For a cold-warm contrast, if there are simultaneously at least one of three cold ITTEN colors coupled with at least one of three warm ITTEN colors in the color palette, the color palette is said to display a cold-warm contrast. One of the challenges encountered when classifying images into cold-warm contrasts by hand is simultaneous contrast. An image that showed *red* and *browns* were classified by the machine as such, but the eye tricks the labeler to believe that instead of *brown green* is seen due to *red*. The eye needs to see the *red*'s complementary color *green* to fulfill a need for balance.

4. **Complementary contrast** For a complementary contrast, if there is simultaneously at least one ITTEN color and its respective opposite ITTEN color to form one of three possible complementary ITTEN color pairs in the color palette, the color palette is said to display a complementary contrast.

5. **Contrast of saturation** For a contrast of saturation, if there are simultaneously saturation values of more than 50 and less than 25 on a 0-100 range in the color palette, the color palette is said to display a contrast of saturation.

6. **Contrast of extension** For a contrast of extension, if there are simultaneously saturation values of more than 75 and less than 25 on a 0-100 range in the color palette (first condition for a contrast of saturation) and an additional second condition for area discrepancy is fulfilled, the color palette is said to display a contrast of extension. The second condition suggests the saturated ratio to the total be at most 2% and the desaturated ratio of at least 98%.

4.3.3. An Example: *Jigokumon*'s Color Contrasts

Based on the upsampled EEFFCND Thesaurus-ITTEN data set with 1'035 instances, a KNN classifier was fit for classifying any new LAB color into one of the six ITTEN colors. After fitting a series of machine classifiers to the data, the best model was determined. A KNN model was selected with 23 nearest neighbors and leaf size of 1 as the accuracy score of 87% on the test set with slight underfitting was convincing. The predicted and true distribution of ITTEN basic colors can be gathered from Table 4.7.

		Predicted label					
		blue	*green*	*orange*	*red*	*violet*	*yellow*
True label	*blue*	10	1	0	0	1	0
	green	1	16	0	0	0	2
	orange	1	0	13	1	1	0
	red	0	0	1	16	0	0
	violet	0	0	0	5	13	0
	yellow	0	0	1	0	0	10

Table 4.7.: Confusion matrix for six ITTEN basic colors trained on KNN-23. All errors except for true *orange* predicted as *blue* are analogous colors to the true color.

In Figure 4.12, the LAB space is sectioned into six basic colors *blue*, *green*, *orange*, *red*, *violet* and *yellow* based on the best KNN model. A t-SNE dimensionality reduction technique was used to project the 3-dimensional LAB color space values into 2-dimensional values. This reduction allows to plot the decision boundaries to 2-D. The color classification is a first step towards naming colors in color palettes. In a next step, the rules of color contrast need to be applied to the palette colors' color distribution. Eventually, each color palette can be classified into different kinds of color contrasts.

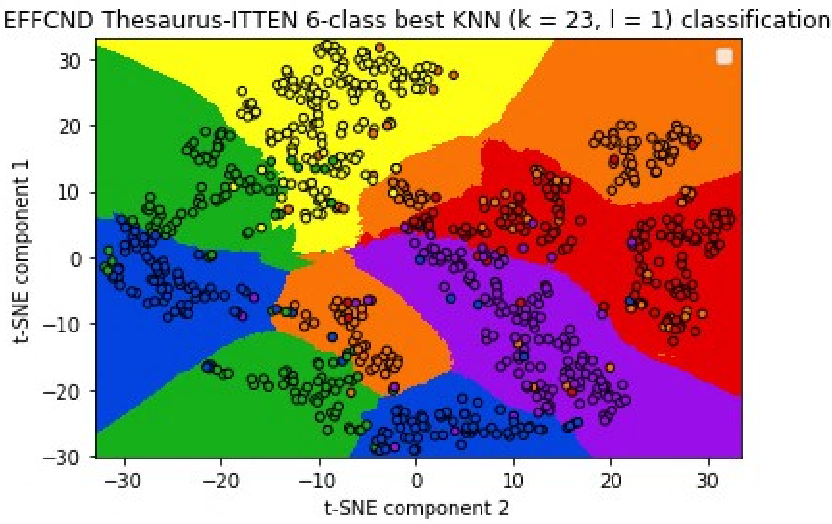


Figure 4.12.: Decision boundaries in LAB space for multi-class KNN model trained on EEFFCND Thesaurus-ITTEN with six basic colors. Most of the data points inside a cluster have the same color as the cluster's color.

For evaluation, a comparison is made between the machine learning predictions of each color palette (or each video frame from the movie *Jigokumon* that was classified into all types of color contrasts with the equivalent hand-labeled classification which is the ground truth. The hand-labeling of all 569 movie frames into six color contrasts and the three corresponding tables hue, lumens and tone was carried out in this project and in collaboration with the *Farb-Licht-Zentrum* of the *Zürcher Hochschule der Künste* (ZHdK)[24] and Prof. Flückiger from the ERC FilmColors project. The color contrast classification algorithm was adjusted after assessing the predictions with the ground truth in order to obtain more true positives and true negatives than false positives and false negatives. The confusion matrices in Figure 4.13 show the results of these adjustments. The MFD was not taken as a control set because the problem with comparing is that contrasts are defined per segment and not per snapshot. A movie snapshot is the same as a frame of a movie, i.e. an image. While the classifier has different contrast classifications per image, the same contrast classification holds for multiple images in the ERC FilmColors MFD.[25]

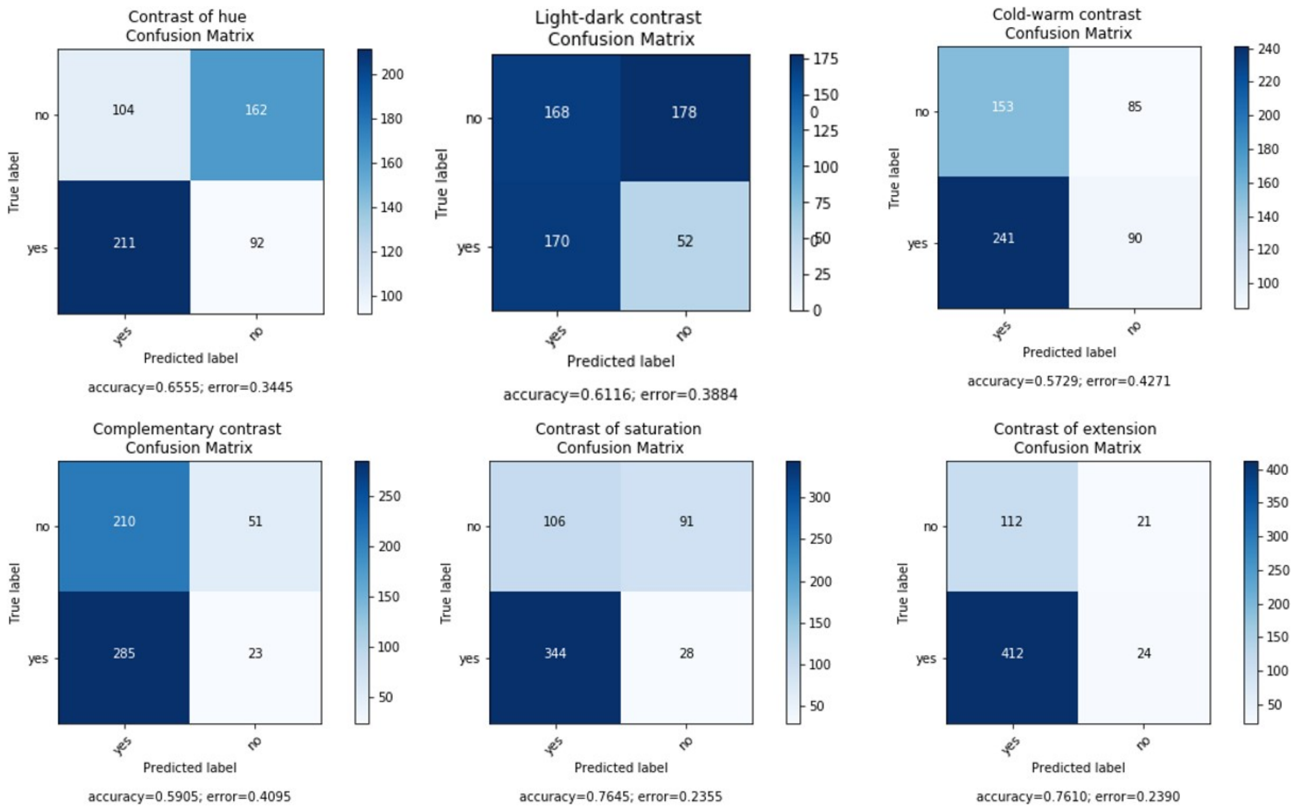


Figure 4.13.: Confusion matrices for all six color contrasts after tweaking the classification rules to target more true positives and true negatives as compared to false positives and false negatives.

In sum, the movie *Jigokumon* the color cinematography consists of many *green-red* color-complementary contrasts and light-dark contrasts which account for many true positives. The strong, almost artificially saturated color accents provide a good breeding ground for a contrast of saturation if not a contrast of extension. Many colors pop into the eye while watching the movie. The user can choose to leverage smaller color pixel values for color contrast classification or a different movie. The results from the confusion matrices need to be taken with a grain of salt because the hand-labeling was not done by experts and a lot of vagueness and elusiveness in classifying video images into color contrasts exists.

[24] https://www.zhdk.ch/forschung/farblichtzentrum

[25] In the Movie Frame Data set (MFD), a segment never has more than three different color contrast classifications; the maximum number of classifiable contrasts for a segment (or snapshot) is three.

5. Implementation

The programming language used for implementing the functions, methods and classes of this project is Python, because of its versatility and ease of use. In addition, VIAN's framework is built in Python, which is why all code is written in Python. The core Python-libraries that were employed are OpenCV, Scikit-learn, Pandas and Numpy. OpenCV is used for color conversion, image manipulation and computer vision. The OpenCV module is the centerpiece of all image processing, color-based calculations and visualizations performed on VIAN [FH18]. Scikit-learn's core functionality is to offer a free toolkit for machine learning in Python built on top of SciPy.

The entire project's file structure is accessible as a Github repository for implementation: `github.com/palettpen`. All documents were made available under an open-source license. The file structure's most important functionalities, components and challenges are described in the section below.

5.1. Functionalities

The organization of the source code and the dependencies are shown in Figure 5.1. It can be divided into six different module classes from 0 to 5 in loose sequential order. Each module's filename contains the class' digit and a second digit that allows for sequential execution of the files. The module classes can be grouped into five functional categories: utilities, dictionary, category, models and retrieval.

- **Utilities**: There are two main module components in this class: color conversion and visualization. The first contains functions for converting colors between color spaces and the second consists of a visualization toolkit for 1-D, 2-D or 3-D display of different object items. The utilities are instrumental in the processing and conceptualization of functions from other module classes.

- **Dictionary**: The raw color name dictionary is enriched with color values for other color spaces using color space conversion as a subcomponent. Color categories are added using a basic color system or the set of dictionary colors already present in the raw color name dictionary. The color name dictionary is extended by utilizing an automated web scraper for downloading images from online digital image datastores. It can also be extended by interval or randomly generating synthetic data from the same Gaussian distribution per label.

- **Category**: The center of each color category, such as the set of basic color labels, can be determined computationally through various means. Considerations about where to locate the class center were required for the conceptualization of the distribution of class centers in the LAB color space.

- **Models**: All possible multi-class and multi-label machine learning classifiers applicable to color classification were implemented based on the enriched CND. It is possible to run `GridSearch` over all combinations of hyperparameter values for all values or to leave out `GridSearch` in favor of a `train-test-split` data set and a 5-fold-`cross-validation` split scoring method of the data set.

- **Retrieval**: The retrieval system for classified video images is based on corresponding color palettes. Hence, the input video clip may be subsampled to a series of video image frames at a specified interval. Color palettes are extracted from these video images by using a given agglomerative clustering algorithm. Color histograms are computed on these color palettes for similarity evaluation. The retrieval system's core functionality is the search query and the search results obtained from the search query. A search query can be raised for color categories (additional specifications are the level in the hierarchy of the color palette, the threshold as a percentage of the color pixel's size in the image and the number of palettes colors to display), color contrast category (additional specifications on the level in the hierarchy of the color palette and the number of palettes colors to display) and top-n most similar images for an input image.

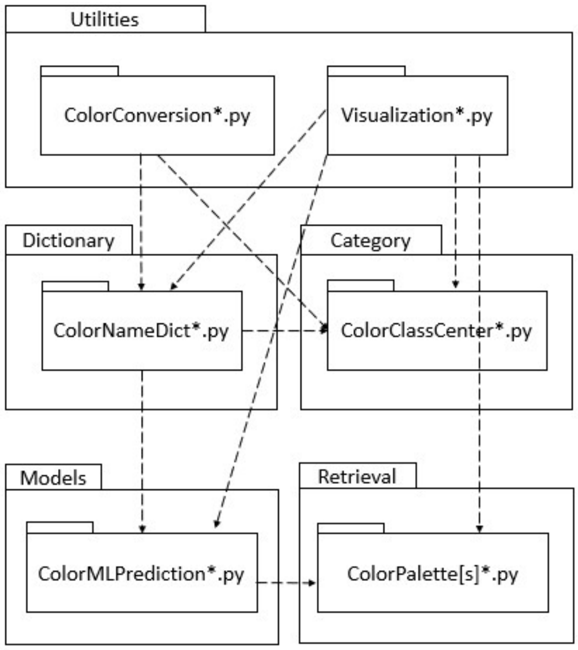


Figure 5.1.: Source files and dependencies

5.2. Components

The project's main components are critical for implementing the tasks. Most of the components are replaceable at a higher likelihood than the processing steps. The file structure's critical components can be divided into three categories: color name dictionary, machine learning and the FilmColors project.

Color name dictionary The CND source used was the EPFL Color Thesaurus. The dictionary was extended with a basic color system unless dictionary colors served as labels to the color features. Since a balance of class frequencies was advantageous for training the classifiers, the dictionary was upsampled using either method for upsampling to generate more color values. The dictionaries that were used in this project are listed in Table 5.1

Processing Level	Source Dictionary	Basic Color System	Color Category	Resampling Method	Class Balance
EFFCND	Thesaurus	VIAN	Basic color	-	Imbalance
EFFCND	Thesaurus	ITTEN	Basic color	-	Imbalance
EEFCND	Thesaurus	-	Dictionary color	Interval	Balance
EEFCND	Thesaurus	-	Dictionary color	Search	Balance
EEFFCND	Thesaurus	VIAN	Basic color	Interval	Balance
EEFFCND	Thesaurus	ITTEN	Basic color	Interval	Balance

Table 5.1.: Processed Color Name Dictionaries

Machine learning Machine learning classifiers were trained using multi-class and multi-label classification with or without a grid. The best models were saved to the disk. Two lab reports, one for multi-class and the other for multi-label classification are available as Excel files. They cover the whole machine learning process from imputation, preprocessing and normalization, scoring and short-listing suitable classifiers to metrics specification. The results for train-test split (90:10) and 5-fold cross-validation are provided for each classifier as well as other Scikit-learn classes for embedding and plotting functions. The classes, functions, methods, parameters and hyperparameter value ranges, constants and default values were the object of meticulous study. The decisions for or against are explained in detail in the two lab reports.

FilmColors project The video frames formed the basis for extracting hierarchical color palettes. These color palettes are saved as CSV files where information about each row in the hierarchy, corresponding bins, bin widths, cumulative bin widths, bin width ratio and the respective BGR color values for each bin are stored. The color palettes are available as images for the lowest row and all rows. The color classification is based on the lowest row. Also, from the lowest row of the color palette the similarity performance scores were derived from comparing color histograms. Furthermore, the color palettes for the entire video clips are stored in a single HDF5 file. PDF files were generated for color contrast annotation of video frames from the video clip *Jigokumon* (project id 7). There exist multiple PDF versions based on multiple images per page or a single image per page, see Appendix B.3. The annotation was done using the hand-labeled color contrast classification from the ERC FilmColors project task force and the other classification was done using a machine learning classifier. In addition, all data from the classifier's classification ranging from color contrasts to the hue colors' image size ratio, lumens and saturation values were exported into a similar data set as the one given on hand-coded MFD classification.

5.3. Challenges

The main challenges encountered during this project were educational. Chiefly, approaching a new endeavor with a structured mindset where breaking up the problem step-by-step into individual components and tackling the components in small bite-sized pieces can become a winning strategy.

Two 1.61GHz CPU processors and up to 8 GB RAM were used for most computations, but at times, training the 29 machine learning classifiers over 24 performance scores required more power, time and memory. This challenge was mounted using hyperparameter value combinations of less than a hundred and outsourcing portions of the execution to *Google Colaboratory*, where the processing is powered by a Tesla K80 GPU and a total of 12 GB of RAM with up to 12 hours of wall-clock runtime.

A real challenge was the sheer amount of possible hyperparameter values from which to choose from and tune the classifiers. Values for which the default value or a constant did not make sense needed to be optimized. Some parameters from the documentation did not work anymore because the classifier's configurations were updated. Other classifiers lacked a method required to do cross-validation. Certain multi-label and multi-class metrics did not apply to multi-label and multi-class classifiers. A considerable drawback were nicely presented code snippets posted online that did not compile. This challenge was taken on by detailed documentation in the lab reports of all possible options that formed the basis of informed decision-making.

Keeping account of all project script files across different file versions, applications and devices may be quite demanding. However, it is essential to avoid overwriting a critical piece of the puzzle. However securely backed-up the project is, issues of habitually forgetting to persist the produced work or running into program crashes will regularly surface. Thus, the data may be held on a portable USB stick and routinely committed to an online Github repository for greater security.

6. Results

The results of the search queries were measured against standard performance metrics and, if not applicable, two randomly sampled picks are drawn as unit tests for manual assessment. The search queries enable the user to retrieve and analyze video images based on novel and versatile techniques. Each one of the three tasks' search query results is described as follows.

As with previous examples, we use *Jigokumon* as a sample video clip to show the search results from querying the video frames. The interest of the movie lies in its unusual use of color. We subsample *Jigokumon* into consecutive images from which a color palette per image and a color histogram per color palette are extracted for all video frames. The extraction is hierarchical – the lowest row contains the highest amount of different colors for a video image. The highest row contains one color aggregate for the entire video image.

6.1. Task 1: Querying Color Names

Based on color classification, a user can query the retrieval system that contains a stack of subsampled video images and their corresponding color palettes. To this end, the following parameters need to be specified:

- Name of the color category (basic color or dictionary color), LAB tuple, e.g. *mustard*
- Palette depth, string or integer, e.g. "row 20"
- Threshold ratio in percent, integer, e.g. 5
- Number of color patches in the displayed color palette, integer, e.g. 10

The user enters *sepia* as input color and finds an exact match of the color category in the palette colors extracted from video images. A prior indexing of palette colors into color categories based on the machine learning estimator speeds up the matching process. Defining a color space range for *sepia* for which the values of the input color *sepia* needs to be numerically fitted instead of annotated by a nominal descriptor will slow down the retrieval process because of the additional search space that needs to be explored.

The user can set which palette row in the hierarchical color palette to fix. It was fixed to the lowest row 'row 20' because the highest level makes only one single color available for the entire image. All palette colors are then classified into one out of 28 VIAN colors using the top-rated machine learning classifier. The user's VIAN color search key is matched to the predicted VIAN colors of each color palette. The number and names of all successfully matched color palettes are shown as a result of the query. The color palettes first n-to-all color patches can be displayed in the search result.

The threshold ratio indicates the percentage amount of area in the image the input color must fill out minimally to appear in the search results. Sometimes, it is not useful to the user to get all images where the input color makes up only a small portion of the image. Setting a threshold will help the user to sift out such images from the result. At the lowest level of an image's color palette, all 101 colors have a certain ratio width, typically, fluctuating around 1 percent from the total width.

If a user wants to search for images by color, what is usually required is to search for a most dominant color in all images of the search result. This method can be extended to include the second most dominant, third-most dominant, ... These are colors of an image that form the first 101 colors of an image's color palette. Consequently, the search results were sorted by highest palette color area match in descending order.

For example, the user can search for a basic color such as the VIAN basic color *mustard* with palette depth 20, threshold 1 and 10 color patches displayed in the color palettes. The Gaussian Naïve Bayes classifier is fit on the EFFCND Thesaurus-VIAN such that the user can search *mustard* in one of 569 color palettes of the movie *Jigokumon*. The number of palettes that were found containing the color *mustard* is 52. Fixing the threshold to 1%, we get only 52 color palettes back for *mustard*. With a threshold floor of 0, however, the number of color palettes found increases to 177. The first three search results are displayed in Figure 6.1.

Figure 6.1.: Top-3 search results out of 52 for search query `mustard`

In another example, the user searches for a dictionary color instead of a basic color such as the more fine-grained color name *avocado* with palette depth 20, threshold 1 and 10 color patches displayed in the color palettes. If one of these color palettes contains the color *avocado*, the color palette will be shown to the user. This time the Gaussian Naïve Bayes classifier is fit on the EFFCND Thesaurus-Interval such that the user can search *avocado* in one of 569 color palettes of the movie *Jigokumon*. The number of palettes that were found containing the color *avocado* is 58. The first three search results are displayed in Figure 6.2. The highest-ranked video image contains the most *avocado* colors, followed by a second and third video image with less *avocado* colors.

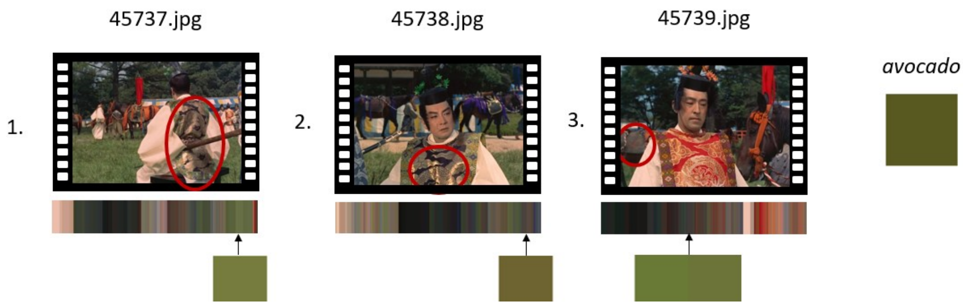


Figure 6.2.: Top-3 search results out of 58 for search query `avocado`

6.2. Task 2: Querying Similar Images

The results are shown from querying similar images. Based on pairwise histogram distance calculations, a user can query the retrieval system based on a stack of subsampled video images, corresponding color palettes and color histograms. To this end, the following parameters need to be specified in the search query:

- Collection of color palettes (all rows or specific rows), JPG, e.g. 45442_D0_100_lab_palette, 45443_D0_100_lab_palette, ...
- Color space (LAB or RGB), string, e.g. 'lab'
- Color palette or image id, string, e.g. '45542'
- Top-n results, integer, e.g. 10

With an increase in volume in the collection of color palettes, an exponential increase in the number of pairwise combinations of histogram distances needs to be computed. This procedure slows down the process and can lead to impasses. However, three different methods of performing pairwise combinations have been evaluated and thus, this processing step has been optimized.

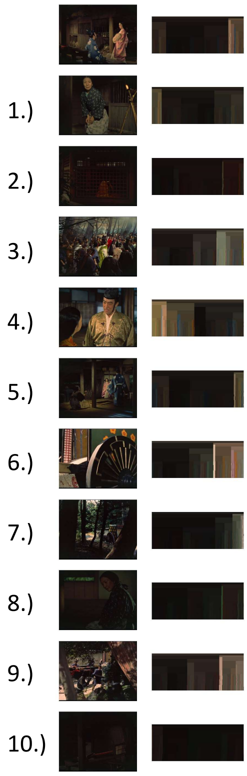
While the input object can be the video frame or the color palette, the returned results may also show a limited set of video images or color palettes from the entire collection. It is more probable that amateurs search for similar video images and experts may search for both images and color palettes. Hence, more attentional weight is assigned to the query and return of video images than color palettes.

Since there is no standard performance metric for evaluating similarity, a random two video images, with ids '45843' and '45505', are picked for color histogram conversion. A color histogram is extracted from the corresponding LAB-based color palettes as well as directly from the video images. The top-10 most similar images from the entire collection of 569 video images from *Jigokumon* is evaluated as illustrated in Figure 6.3.

Search query:

45843.jpg

Rank	Palette	Correlation
0	45843	1.00
1	45840	0.94
2	45826	0.93
3	45466	0.93
4	45724	0.91
5	45839	0.88
6	45480	0.88
7	45498	0.88
8	45831	0.87
9	45487	0.87
10	45832	0.87



Search query:

45505.jpg

Rank	Palette	Correlation
0	45505	1.00
1	45507	0.87
2	45506	0.84
3	45511	0.78
4	45510	0.76
5	45691	0.75
6	45692	0.74
7	45514	0.73
8	45668	0.72
9	45897	0.71
10	45532	0.71

Figure 6.3.: Top-10 similar search results for two random picks. A pairwise distance calculation of histograms based on color palettes was conducted.

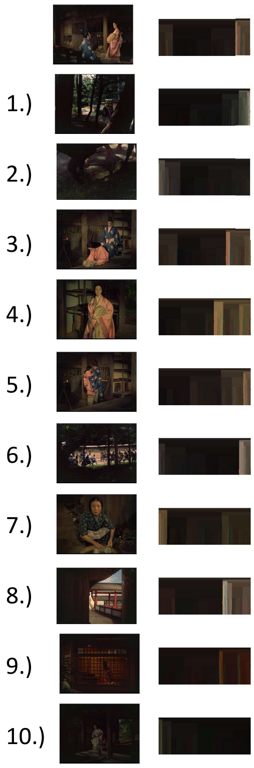
The correlation metrics for the palettes-based histogram calculation indicate that the search results' returned images are all less than 50% similar except in one case. In the first pick, the search results show images that are astonishingly similar to the input image. In contrast, the second pick's target images do not show nearly the same amount of similarity.

An image-based histogram calculation yields much higher correlation scores: over 85% or over 70% for each pick. These higher scores mean that the classifier was able to find images that are more similar to the input image. An inspection of Figure 6.4 provides evidence for this conclusion.

Search query:

45843_D0_100_lab_palette.jpg

Rank	Palette	Correlation
0	45843	1.00
1	45498	0.53
2	45481	0.40
3	45850	0.40
4	45838	0.38
5	45849	0.34
6	45486	0.33
7	45821	0.32
8	45593	0.32
9	45827	0.31
10	45835	0.29



Search query:

45505_D0_100_lab_palette.jpg

Rank	Palette	Correlation
0	45505	1.00
1	45535	0.18
2	45463	0.13
3	45688	0.11
4	45585	0.10
5	45551	0.09
6	45525	0.09
7	45861	0.09
8	45699	0.09
9	45807	0.08
10	45486	0.08

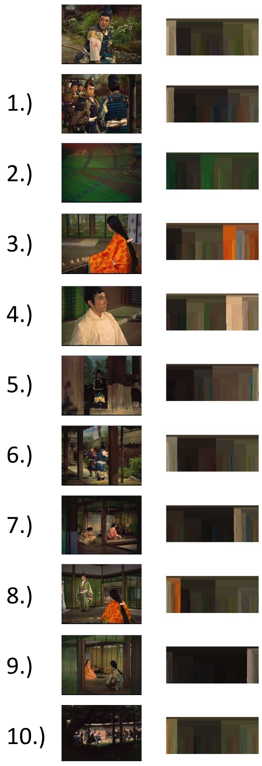


Figure 6.4.: Top-10 similar search results for the same two random picks as above. However, the histograms are based on color images.

The findings show that computing a color histogram on color images rather than LAB-based color palettes can deliver a higher similarity score among the search results. However, for the first pick, despite the higher correlations score for images, the returned color images are more similar to the input color image when choosing a palette-based computation method. The reverse is true for the second pick: both correlations and returned results are better using an image-based histogram procedure. The assessment could extend to all video frames; however, due to the limited resources, only a random pick was assessed. Since these picks are somewhat contradictory, more hand-picked investigations and evaluations should be done in the future. For the time being, a hand-picked ranking of most similar images for both input images, which could serve as ground truth, could look like what is presented in Figure 6.5. These top-3 most similar items for the input query images and palettes were curated by hand.

Figure 6.5.: Top-3 most similar hand-ranked video frames for frames 45843 and 45505 from *Jigokumon*

6.3. Task 3: Querying Color Contrasts

The results show the search returns for a search query on color contrasts. Based on color classification and an additional classification procedure for applying the rules of contrast on top of a color palette's predicted color names, a user can query the retrieval system based on a stack of subsampled video images and the corresponding color palettes. Each video frame and the attributed color palette is classified into different color contrasts before the user starts the query. Two possible queries can be raised depending on the user requirement. The first query is more important than the second because a color contrast is specified. The second query specifies the image or palette to look up such that all the annotated color contrast types are shown in the search result. In the end, it makes more sense for the user to search video frames by color contrast instead of color palettes.

For the first query, the following parameters need to be specified:

- Kind of color contrast, string, e.g. 'cs' (=contrast of saturation)
- Palette depth, string or integer, e.g. "row 20"
- Number of color patches in the displayed color palette, integer, e.g. 10

For the second query, the first specification from the above query needs to be replaced by:

- Color palette or image id, string, e.g. '45542'

In the first query, the user can search for a particular color contrast among all video frames or their respective color palettes. From a collection of images or color palettes, an exact match of color contrast is returned. For example, from all 569 colors palettes of movie *Jigokumon*, 152 colors palettes were found for color contrast 'complementary contrast: violet-yellow'. These 152 color palettes can be displayed with the first ten most dominant color patches. The three tables hue, lumens and tone, can be presented as supplementary information. The tables form the basis of contrast classification.

In the second query, the user can search for a single video frame or color palette and get information back about which of the six color contrasts, a contrast of hue, a light-dark contrast, complementary contrast, cold-warm contrast, a contrast of saturation or contrast of extension are present. In addition, three tables, hue, lumens and tone, can be queried for a particular color palette upon specification.

Figure 6.6 shows sample search results for the first search query, which is the only search query defined as a task goal for Task 3. All six color contrasts were searched in the *Jigokumon* movie frames. The results were extracted from the PDF-file where all movie frames are annotated according to their type of color contrast and the three tables hue, lumens and tone. The examples were picked by hand; they consist of the most conspicuous of all search results.

Figure 6.6.: Search query results from *Jigokumon* for each of six applicable color contrasts; (a) the contrast of hue is made evident in the saturated *red*, *violet*, *orange* and *green* colors set side-by-side; (b) the light-dark contrast stems from small areas of very light colors surrounded by darkness; (c) the warm *yellow* color provides a cold-warm contrast to the cold *green* color; (d) *red* is a contrast to *green* because both colors are complementary to each other; (e) looking at the saturated *red* and *orange* colors, the surrounding colors seem very toned-down by contrast; (f) the small, saturated *red* pops out against a more monotonous-colored background

A system that allows for searching color contrasts from digital images by way of classifying the colors of their assigned color palette has not been developed so far to the best of one's knowledge. However, this novel approach to contrast classification will depend crucially on an accurate color classifier and an accurate set of rules for color contrast classification that should overlap as much as possible with human conceptions about whether an image shows a certain color contrast or not.

7. Discussion

The results presented for each task offer several advantages and limitations regarding previously achieved research results in studies on color and contrast classification, similarity measurement and image retrieval. In addition, the results are also evaluated with respect to the research goals that were presented at the beginning of this project.

The techniques proposed in this thesis has several general advantages. First, it uses the all-encompassing LAB space for all computations with color pixels. Second, the color name dictionaries and the set of color categories used for training the classifiers are replaceable. Third, a much wider variety of different classification methods can be used to eventually search for colors and color contrasts than in other studies.

The approach taken in the course of this project exhibits some limitations. First, many of the definable colors in LAB space do not make practical sense because they are not within the human visual system's scope of perception. Since a user cannot become aware of that reality, it may make little sense to utilize these spatial extensions. Second, it would be useful to have *brown* as a color category for improved color contrast classification. However, Johannes Itten, who coined the terms used in the kinds of color contrasts, omitted *brown* as a stand-alone color on his color wheel. Third, the performance evaluations are mostly subject to non-objective assessments, leading to a less rigorous and scientific method of experimentation.

7.1. Task 1: Color Classification

All Scikit-learn multi-class and multi-label estimators and the entire list of valid performance metrics were trained in an thorough training and testing procedure. The machine learning results show that color classification for the best built multi-label machine learning estimator can be as accurate at predicting more than 97.4% of the color names correctly, on average, and for any given color value. However, the literature favors a Probabilistic Latent Semantic Analysis (PLSA) model, but the overall equal error rate is only 93% at best [vdWSV07]. Also, for Convolutional Neural Networks (CNN) applied to color classification from deep learning, a score of 85% is maximally achieved [WLW^{+}15]. Hence, our classifier is capable of outperforming other color classification estimators.

The expectation was confirmed that the color categories in LAB space are distributed by the Gaussian since the best performing machine learning classifiers for multi-class classification use the Gaussian to search the hypothesis space. As for multi-label classification, the literature has already discovered that ensemble methods achieve superior performance in training high-dimensional multi-labels [RBHP11].

The results show that colors based on color categories such as basic colors and dictionary colors are searchable using a machine learning classifier to categorize colors. The user can search color palettes by color name. However, the problem with color names is that they evoke different colors for different people of the English-speaking world. Not only do they have different colors in mind, but the colors are also not defined in a precise and clear way. Instead, color names serve to obscure the meaning of the alluded color. As early as in 1905, Albert Munsell already remarked: "the terms used for a single hue, such as pea green, sea green, olive green, grass green, sage green, evergreen, invisible green, are not to be trusted in ordering a piece of cloth. They invite mistakes and disappointment. Not only are they inaccurate: they are inappropriate" [Mun05]. Also, it may be that for a set of very different images, the palette colors are very similar, in which case additional palette colors would help in the search.

7.2. Task 2: Color-based Image Similarity

A novel method for performing similarity computations on the color distribution of digital images was proposed. The method extracts the color histograms from the images' color palettes instead of the images themselves. In addition, a color contrast classification that maps the rules from color theory to a scientific color space for annotating video images based on color contrast has been pioneered in this project. Since no standard performance measure has [DKN04] and shall probably ever be established in image similarity, the search results need to be evaluated by hand. Other studies have used color histograms directly from images; hence a comparative evaluation with these procedures was the objective. However, due to the randomly selected and limited hand-picked test samples, a better method could not be determined. Weights could be assigned based on the distance between the input and the target image to improve on both image and color palette-based similarity computations. Since most similar images are situated nearby, such an additional distance-based weight could help improve performance.

7.3. Task 3: Color Contrast Classification

In contrast classification, the classifier is only as good as the extent to which it makes sense to annotate a color contrast. For instance, for video images where *green* and *red* are achromatized and relatively dark, a human annotator is likely to fail to classify the image into a complementary contrast of green and red. At the same time, the machine-trained classifier will automatically conclude, despite the shades, tones and size, that a *red-green* complementary contrast is present. Another example is a *red-green* complementary contrast in very small local portions of the image that the viewer fails to see. A threshold can be set in both cases.

The annotator could also be biased because of how color contrast in images affects the perception of color compositions in images [LLYZ12]. It helps to have peer discussions during the annotation process to document the different trains of thought. Such a judgment can be crystallized further by polling a group of experts.

Technically, it would be feasible to classify images into color contrasts using a deep learning model. However, a deep learning model requires a data set containing a massive volume of hand-labeled images. These images need to be classified based on the (non-)existence of color contrasts in them. Since there exists only a segment-level manual classification of contrasts in the data set and 7'218 images contained in the data set are not enough to train a sufficiently significant deep learning model, this approach was dismissed.

While much research has been done to propose methods that enhance a digital image's color harmony, the same does not hold for color disharmonies. Studies that search to increase color disharmonies by proposing such colors may have no application areas because disharmonies trigger uncomfortable states of being.

When querying images for color contrasts, the search results can be sorted in order of the appearance. However, a better method would be to have the most salient color palettes, epitomizing the color contrast, rank first in the search results. Color palettes where the searched color contrast appears to be unclear should rank low. Such a procedure would necessitate a ranking of color palettes based on how good they portray the color contrast. A color palette where the contrast of hue is not preponderant because the hues do not take up a large part of the image or are comparatively too tinted, toned or shaded to make a difference, should rank at the end of the returned search results.

Based on the annotated color contrasts, the symbolic character of the employed color may become more apparent. Such interpretations can be of great help for film producers and analysts who want to reproduce a symbolic atmosphere. They can get inspiration from analyzing different masterworks of cinema with contrast classification and use the context in which specific contrasts appear as a tool for their film-making.

8. Conclusion

In this project, a color-based visual retrieval system was presented for movie pictures. The contributions are three-fold: (1) a prototype that supports the search for color images or the corresponding color palettes by basic color or dictionary color search query; (2) a new method for approaching digital image retrieval based on similarity was proposed, i.e. a color histogram converted from an image's color palette instead of the image directly; (3) an automated system for color contrast annotation and the query of six types of color contrasts.

First, the assumption that color class centers follow a Gaussian distribution was confirmed by experiments. Hence, future algorithms may consider a limited color distribution sampled from the Gaussian family of distributions. *Second*, color classification models are more accurate in a multi-class than in a multi-label approach: for the Thesaurus-ITTEN data set the multi-class KNN23 had an accuracy of 0.87, whereas the multi-label KNN21 had an accuracy of 0.85. This goes somewhat against the upper claims that colors are distributed according to the Gaussian. However, due to the small difference in accuracy, the scores might change again for another color name dictionary. A performance review still needs to be made on a collection of color name dictionaries and their multi-label and multi-class accuracy scores. This project was limited to compare only two color name dictionaries: the Thesaurus-VIAN and the Thesaurus-ITTEN color name dictionary. *Third*, ensemble methods are good at multi-label classification: in multi-label classification, the trained Extra Trees classifier is best suited among all 29 classifiers for classifying dictionary color names to basic color categories (macro f1-score = 0.93) for the Thesaurus-VIAN data set, unless underfit is an issue. In that case, the KNN classifier (macro f1-score = 0.91) would be the best-trained classifier. This proposed color classification estimator outperforms standard or task-adapted classifiers from related studies. *Forth*, basic colors, as well as dictionary colors, can be classified using the trained classifier. Instead of using task-adapted text classification algorithms, colors can be directly computed based on their color space LAB value. *Fifth*, digital images or their color palettes can be searched by basic colors or dictionary colors. Since users have a hard time specifying the colors they have in mind using space color values, having a large set of dictionary colors for querying colors can be helpful. Finding the basic color category for any color can help with labeling colors to the most common denominator of color words. *Sixth*, a threshold for color dominance in the image can be set. A user might not want to query a color and retrieve images where said color has the size of a single color pixel on the image. Images where the color is easily visible need to be listed in the search result. A threshold leaves it to the user to decide which images to neglect and which movie pictures to retain. Seventh, randomly picking two test images for evaluating the color histograms on color palettes and the color histograms on images is not conclusive. More samples need to be drawn and compared to arrive at a reasonable conclusion. *Seventh*, although possible, Johannes Itten's definition of color contrasts are captured in more detail when defined mathematically than artistically. For example, in art, a light-dark contrast is defined as the positioning of light and dark colors next to each other. However, from a programmer's point of view, a light-dark contrast requires at least 50% of dark color values and less than 50% of light color values in the image. Preferably, a light-dark contrast is seen in an image with very bright light sources of color against a dark background where dark is less than 25% of luminance in LCH space and light is more than 75% of luminance. Such evaluations need to be determined iteratively by thinking about the concepts and what they mean to the viewer in a more specific way. *Eighth*, the best multi-class KNN classifier for the Thesaurus-ITTEN data set can classify images based on six different types of color contrasts: a contrast of hue, light-dark contrast, cold-warm contrast, complementary contrast, a contrast of saturation and contrast of extension (maximum accuracy = 0.76). This classification is possible due to the ITTEN basic colors which consist of six colors *red*, *orange*, *yellow*, *green*, *blue* and *violet*.

The techniques were implemented on a collection of 569 video frames from the movie *Jigokumon* and their color palettes. For contrast classification, all video frames were annotated by hand. Each image was carefully considered based on each of the six contrast types and compared against a color contrast estimator. Mostly, the judgements were led by intuition and common sense and the estimator was adjusted to learn the underpinnings of these considerations. The construction of the algorithms was motivated by the framework of a visual video annotation tool that was developed as part of the ERC FilmColors project which enables the user to analyze and restore colors in movies. For example, samples of screenshots taken from different movies were hand-labeled into color contrasts. This manual classification is extended by the estimator's automated contrast categorization procedure.

Future work may focus on extending the methods proposed in this project to other domains in art and computer vision such as architecture, interior design, the fashion industry, medicine, marketing and the food industry. Considerations from the artistic color theory domain include extending the tasks to color harmonies such as split complementary, split primaries and tetrads.

In addition to color contrast, the emotional expressions encoded manually in the FilmColors database could be used and evaluated against a machine learning classifier that categorizes video frames by the corresponding emotional encoding [MH10]. Another idea for future work could be to evaluate whether negative emotions are associated with certain kinds of color palettes or color contrasts. In contrast, positive emotions correspond to a set of different color palettes and color contrasts. In addition, besides texture, a feature that has already been analyzed within the ERC FilmColors project, other low-level features such as shape and layout could be used in conjunction with colors.

A multifunctional web application as a sequel to this project can be a built-in extension to VIAN that allows a user to use a graphical user interface to employ the functionalities described above within a few mouse clicks. This sequel could become the Swiss Army knife among the tools of visual analysis and annotation of VIAN. Relevance feedback from users labeling the accuracy of similar images can be used to improve image rank [CWT08]. Thus, this project's achievements could provide the impetus for new and innovative developments of the future.

A. Appendix I

A.1. Color Designation

A vocabulary of words such as *orange*, *copper* or *mustard* can be used to depict colors and to search color palettes. Such colors are also present in color palettes, which makes them searchable. Natural language color names are categorical, meaning they can be mapped to discrete color values in so-called Color Name Dictionaries (CND). Such specifications are color notations or graphics systems that can realize up to 4096 different colors. The colors contained in them will have at least one of the RGB, HSV or LAB color value conversions pre-defined in a Color Name Dictionary (also called Color Naming System). Depending on the color space (RGB, HSV or LAB) chosen by the builder of the dictionary the color displayed is as close as possible to the color that is commonly visualized when evoking the color name with the dictionary builder. The color spaces utilized here are the most prominent color spaces found in the computer graphics literature. If one of the color spaces is missing, the others can be easily derived by computation. For example, the hexadecimal code (HEX code) or the CMYK values are also important color values for web-visualization and printing. The color names can be very diverse ranging from colors such as *crimson* or *chartreuse* to *azure* or *mint*. Visualizing the color's image gives a visual sense of colors. In HSV, the h-channel can represent a color's hue or as a categorical name (basic color) for the color that is represented on a color wheel. Table A.1 illustrates an example of different color designations for the color *sunflower yellow*.

Text	Color name	Color category			
	sunflower yellow	*yellow*			
Number	RGB	HSV	LAB	HEX	CMYK
	(255, 218, 3)	(51.0, 98.8, 100.0)	(88.22, 1.48, 86.80)	#ffda03	(0, 15, 99, 0)
Image	Color image				
					

Table A.1.: Example of different color designations for *sunflower yellow*; a color can be represented as text, number or image

Beyond the natural language color names in English and their color values commonly represented numerically as triplets of real numbers, the color name dictionaries can be extended to include classifications of all color names to primary colors such as *red*, *green* or *blue*. Some color names, such as *black* can be mapped to a smaller set of space colors than the color name *almost black*. In turn, *almost black* can be mapped to a smaller set of space colors than the color name *dark color*. These dictionary colors have their own particular set of space color values and discrete color value in the original color name dictionary. In Table A.2, this process corresponds to moving down the ladder from Level 3 to Level 2, if color names are mapped to basic colors and from Level 4-6 to Level 3, if color values are mapped to color names. At Level 6, the specification of colors is infinite, because, with every added order of precision, a color's manifold becomes unlimited. The RGB space, while canonically divided into three 0-255 integer ranges, can be rescaled to real numbers of the interval [0,1], thus introducing floating-point values of arbitrary precision. In Table A.2, an overview of all levels of granularity is shown for color designations.

	Color Name Designations			Numeral and/or Letter Color Designations		
Level of Granularity	Level 1 (coarse)	Level 2	Level 3	Level 4	Level 5	Level 6 (fine)
Number of Segments of Color Solid	3	12	500	256x256x256 = 16’777’216	>3 trillion	infinite
Type of Color Designation	Primary colors: color space channels	Basic colors: hue names	Color names: dictionary colors, palette colors	Grid-sampling of color system	Interpolation of grid-point values, HSV space	CIE L*ab space, XYZ space
Example	blue	green	vermillion	RGB (22,122, 108)	HSV (27.1, 0.8, 0.7)	LAB (99.314, -24.1, 22)
Color Systems	RGB: red, green, blue CMYK: cyan, magenta, yellow Primary: red, yellow, blue	Boynton Itten	EPFL Color Thesaurus XKCD Color Survey	RGB cube	HSV cylinder	CIE L*ab horseshoe

Table A.2.: Color designations by granularity

A classification of all color names to their respective basic color category can mean that all color names are mapped to one basic color category. For example, color name *blue-green* is classified to *green* because the emphasis on the second word *green* makes the color more *green* than *blue*. Alternatively, more leeway is given to colors at the border between two basic colors. Whether it is a border or rather, a gray zone between two basic colors is left to the user to decide. Color name *blue-green*, for example, can be classified to *blue* and *green*. In that case, a hard border makes way for a soft border. In machine learning words, a multi-class classification approach turns into a multi-label approach. The user can decide which approach to take.

A.2. Basic Color Sets

The color names in a color name dictionary need to be classified into a color category. This color category forms a set of basic colors to which all color names can be categorized. The purpose of having basic color categories is to group color names. Hence, each color name can be related closer or further apart from another color name, depending on its basic color category. Basic colors, or original or component or standard or primary colors, are categories that group other color names together. Berlin and Kay studied how native speakers delimited the boundaries and foci of such basic colors most fundamental and universal to all languages [BK91]. They conclude that basic color categories have evolved over time: the simplest color lexicon includes only *black* and *white*, *red* is added at the next stage, then either *green* or *yellow* or both, then *blue*, *brown* and finally combinations of *pink*, *purple*, *orange* and *gray*. For instance, *olive*, *seafoam* and *lime* are color names or terms that can all be classified into the color category *green* which makes *green* a basic color. Typically, a basic color lexicon consists of not more than a dozen basic colors. More nuanced versions of the color may be mapped to a basic color. The number of basic colors native English speakers use to name all shades of colors has evolved over the centuries. How many and which of these basic colors should make it up to the list depends on different schools of thought in color theory. Basic colors are not primary colors. Primary colors are colors from which all other colors can be derived such as *red*, *yellow* and *blue* – mixed together they form either *orange*, *green* or *violet*, the secondary colors. Tertiary colors can be mixed from taking a primary and a secondary. The primary colors in (computer) vision are defined by the three chromaticities *red*, *green* and *blue*. From these three color channels all other colors are derived. The primary colors in print are *cyan*, *yellow* and *magenta* . Mixing together different proportions of inks can produce all derivative color forms. Basic color implies more than just primary colors. It extends to all colors, mostly from six to twelve, that people can easily or commonly distinguish one from another. For example, in his book, artistic color theorist Johannes Itten derives six basic colors *red*, *orange*, *yellow*, *green*, *blue* and *violet* – the colors of the rainbow based on Isaac Newton's color experiments with a prism infused with light.

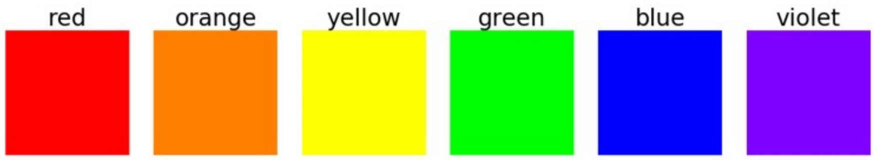


Figure A.1.: Johannes Itten's system of six basic colors categories

A more scientific way to define basic colors is by looking at different color spaces and their corners. For example, in RGB space, we find the following six colors in each corner of the cube: *red*, *green*, *yellow*, *blue*, *violet* and *cyan*. In the same vein, LAB space corners contain nine colors *blue*, *cyan*, *yellow*, *brown*, *pink*, *orange*, *red*, *green* and *magenta* . On the other hand, Boynton proposes eleven colors "that are almost never confused" . They are *orange*, *brown*, *gray*, *pink*, *magenta*, *yellow*, *green*, *red*, *blue*, *white* and *black*. Since *black*, *white* and *gray* have no chroma, they are all positioned at the same location on the color wheel. The ISCC-NBS system denotes *red*, *orange*, *brown*, *yellow*, *green*, *blue*, and *purple* as the seven basic colors. Together with *black*, *white* and *gray* the basic color terms of English are enumerated. Speakers of English perceive these basic colors as different and typically use these distinctions in everyday language to make use of color as discursive descriptors. While in numerical notation systems of color, *brown* is simply a dark shade of orange, in natural language *brown* appears as a separate color category because it has culturally emerged to be perceptually different than orange. In the ERC FilmColors project, however, 28 basic color categories are searchable on VIAN. Most of the color names are warm colors and some are too similar to form color categories of their own. For example, *peach* and *cream* are basic colors, even if their color image is very alike.

Origin	Basic Colors	Frequency
Johannes Itten	red, orange, yellow, green, blue, violet	6
RGB space	red, green, yellow, blue, violet, cyan	6
LAB space	blue, cyan, yellow, brown, pink, orange, red, green, magenta	9
Werner's Nomenclature of Colors	White, gray, black, blue, purple and violet, green, yellow, orange, red, brown	10
Robert Boynton	orange, brown, gray, pink, magenta, yellow, green, red, blue, white, black	11
VIAN (ERC FilmColors)	amber, apricot, beige, black, blue, bronze, brown, copper, coral, cream, cyan, gold, green, grey, lavender, magenta, mustard, orange, peach, pink, purple, red, rust, sepia, silver, ultramarine, white, and yellow	28

Table A.3.: Set of basic color systems; colors are sorted alphabetically or listed in order of appearance

Based on these differing systems of defining basic colors, a set of basic color categories is proposed which takes into account the preceding considerations about basic colors to make color categorization as workable as possible. These are *pink*, *magenta*, *red*, *orange*, *yellow*, *brown*, *green*, *cyan*, *blue* and *violet* – ten basic colors. It is assumed that *white*, *gray* and *black* are not colors because they occupy the same location on a color wheel. When deciding which colors to incorporate in the search request, another interesting idea would be to include the Pantone® Color Institute's Color of the Year. The Color of the Year 2020, for example, is *classic blue*. The user could obtain all color palettes (and by extension all images) for which *classic blue* constitutes a distinct part.

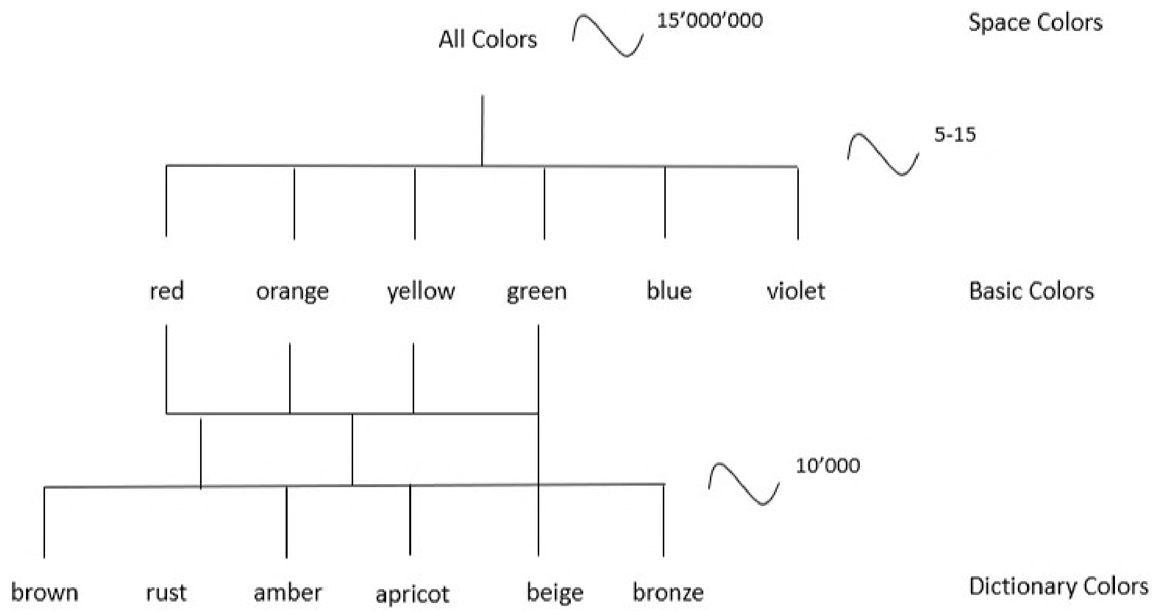


Figure A.2.: Space colors, basic colors and dictionary colors

A.3. Visualizing Basic Color Categories

The colors in a color name dictionary can be visualized together with their basic colors. The color names of EFFCND can be plotted one-by-one or in a color cloud depending on the color space chosen. The same can be done with the color names' color categories, the basic colors. The class center average of each basic color category is computed using the average color values of all color names with the same basic color category.
In this example, the EFFCND EPFL Thesaurus – VIAN is used. The color names are based on the EPFL Color Thesaurus and the basic colors are taken from the VIAN web tool. Note that this EFFCND is interchangeable with other color name dictionaries. Below, the three first color names are visualized together with their basic color category. The top-3 colors and their basic colors in the EFFCND EPFL Thesaurus – VIAN are displayed in Figure A.3.

When plotting all colors in an EFFCND into different color spaces, the colors' basic colors are used for coloring in their data point dots. The EFFCND uses as source the EPFL Color Thesaurus and as system VIAN's 28 basic colors. Plotting all colors and their basic color class center averages into LAB space yields the 3-D plots depicted in Figure A.4.

Rank	Color name	Color image	Basic color (VIAN)	Color image
1	'adobe'		'orange'* *basic color's values found as color name	
2	'algae'		'green'* *average of all color names' LAB values with that basic color	
3	'amber'		'orange'	

Figure A.3.: Top-3 colors and their basic colors in the EFFCND Thesaurus–VIAN

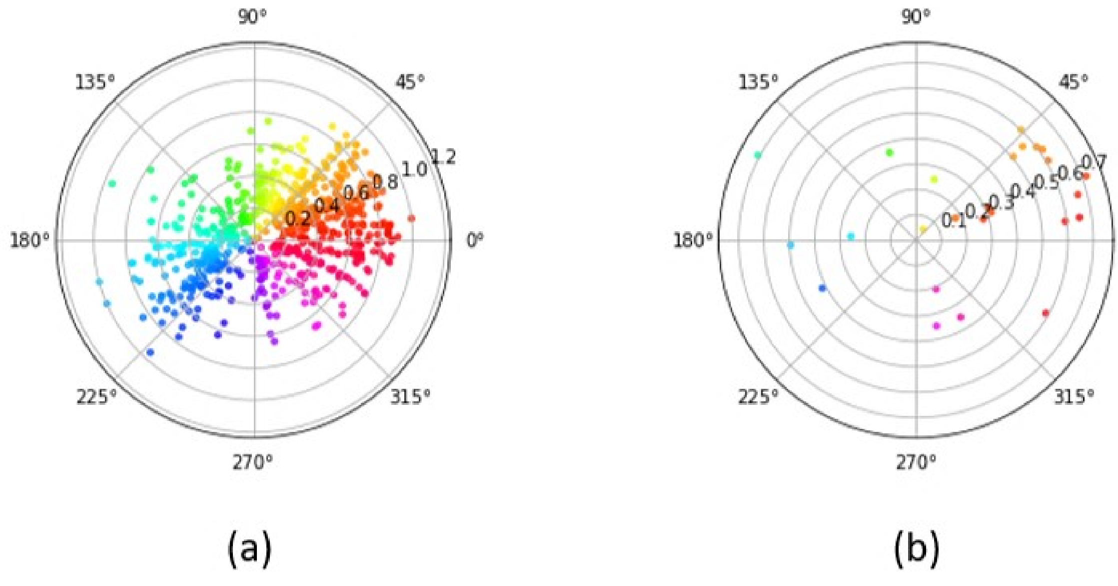


Figure A.4.: (a) All colors of the Thesaurus on the color hue wheel; (b) VIAN's basic color class center averages on the color hue wheel; Note that the colors' chroma level along the second chroma-axis has not been visualized in the colored data points

In LCH space, the colors are situated on a hue circle because the LCH cone has a circular form at the top of the cylindrical color solid. Plotting them in 2-D, as seen in Figure A.4, can give a better overview of the color hue distribution in an EFFCND. To the left, all colors plotted while to the right, the basic colors are shown by taking the average of corresponding colors in the EFFCND. (A less exhaustive method would be to plot the basic color found inside the color names, but some basic colors in VIAN such as *green* are not included in the EPFL Color Thesaurus.)

The VIAN colors are biased towards *orange* and *red* color tones on the color wheel. These are predominantly warm colors made available for the user to search. If we take the averages of all CIE-Lab VIAN color values for a VIAN color category and plot the resulting color class centers, we find mostly earthy warm color hues in the color scheme.

In Figure A.5, the LAB average of all color thesaurus values for a VIAN color category is plotted. We get a unique value for all 28 VIAN colors. We sort them by their luminance value from dark to light.

Figure A.5.: Thesaurus-VIAN's class center averages as color palette

Some colors like *sepia*, *bronze* or *copper* and *brown* or *yellow* and *gold* or *peach*, *apricot* and *cream* are very close to each other – almost indistinguishable to the human eye. Thus, it is advised to extend the VIAN colors to another set of colors so that the user has a larger variety of colors to choose from for the search query.

Suppose we project these VIAN color points to a constant chroma of 100. In that case, that is the highest saturation for a color, convert the LAB colors to LCH, and round the LCH values off to the nearest 10, colors *orange*, *amber* and *silver* become identical in value. Since *orange* seems to be a good basic color, *orange* was left in the data. The LCH color space is divided into ten luminance levels ranging from 0 (dark) to 100 (light) and 36 hue angles of 10°-steps around the color wheel.

The VIAN colors in LCH space are heavily clustered into warm, earthy color tones, i.e. *browns*, *reds*, *oranges*, and *yellows*. This is best seen in *peach*, *cream* and *apricot* – VIAN colors that follow each other closely. Only nine out of 26 different VIAN colors (34.6%) include cold, clear color tones, i.e. *greens*, *blues*, *purples*, and *magentas*. Thus, it is strongly advised to balance off the predetermined set of VIAN colors to incorporate a more equal distribution across the LCH color space.

A.4. Transforming Basic Color Categories

Basic color sets can be recoded to adhere to a more limited set of basic color categories. For example, VIAN consists of 28 different color categories. These color categories were added to an EFFCND to form EFFCND Thesaurus-VIAN. The ITTEN colors are comprised of only six color categories. In Table A.4, a recoding of VIAN's basic colors to ITTEN's basic colors is shown.

VIAN color category	ITTEN color category
Amber	Orange
Apricot	Orange
Beige	Green
Black	To delete
Bronze	Green
Brown	Orange
Copper	Orange
Coral	Red
Cream	Orange
Cyan	Blue
Gold	Yellow
Grey	Blue
Lavender	Violet
Magenta	Violet
Mustard	Yellow
Peach	Orange
Pink	Red
Rust	Orange
Sepia	Green
Silver	To delete
Ultramarine	Blue
White	To delete

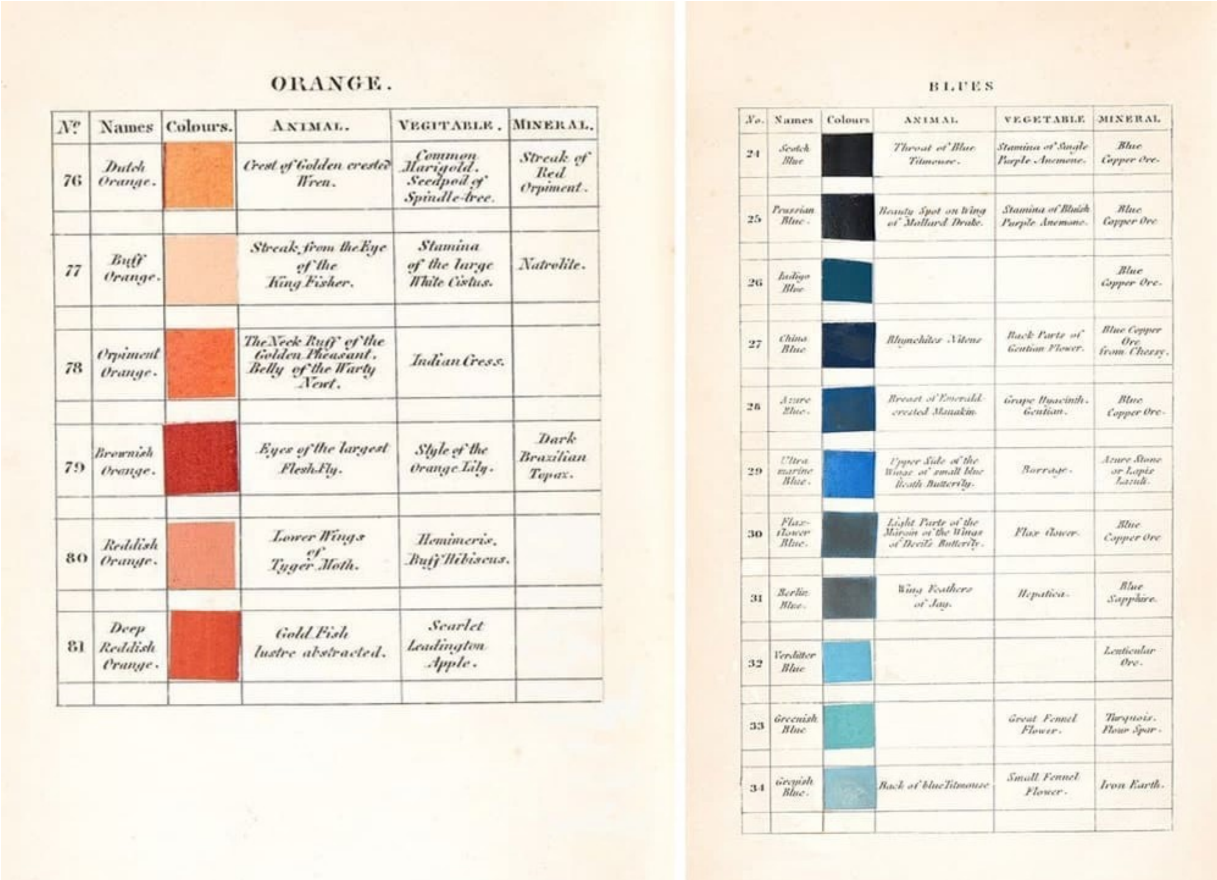
Table A.4.: Basic color category transformation VIAN-ITTEN

A.5. Color Naming Systems

Color naming systems are lists that include a certain number of unsorted color names only. It is the most ancient form of color order systems. While practical in application, there are a few shortcomings that color naming systems exhibit: they are not arranged in a perceptually ordered manner and controlled communication of colorimetric accuracy is hardly achieved by using a color naming system. Examples of such a system are Werner's Nomenclature of Colors (110 colors), the Sundberg Color Thesaurus (240 colors), the EPFL Color Thesaurus (843 colors) and the Pantone Color Formula Guide (942 colors).
As an example of such a color naming system, Abraham Werner and Patrick Syme (1821) developed the world's first encyclopedia of colors. While Werner described colors and gave examples from nature on where to find them, Syme added color patches to Werner's color names and subsequently published the nomenclature of colors. This color name dictionary contains 110 color names.

A.5.1. Werner's Nomenclature of Colors

Werner's nomenclature of colors is shown in Figure A.6.



ORANGE.

N.º	Names	Colours.	Animal.	Vegetable.	Mineral.
76	Dutch Orange.		Crest of Golden crested Wren.	Common Marigold. Seedpod of Spindle-tree.	Streak of Red Orpiment.
77	Buff Orange.		Streak from the Eye of the King Fisher.	Stamina of the large White Cistus.	Natrolite.
78	Orpiment Orange.		The Neck Ruff of the Golden Pheasant. Belly of the Warty Newt.	Indian Cress.	
79	Brownish Orange.		Eyes of the largest Flesh Fly.	Style of the Orange Lily.	Dark Brazilian Topaz.
80	Reddish Orange.		Lower Wings of Tyger Moth.	Hemimeris. Buff Hibiscus.	
81	Deep Reddish Orange.		Gold Fish lustre abstracted.	Scarlet Leadington Apple.	

BLUES

No.	Names	Colours	Animal.	Vegetable.	Mineral.
24	Scotch Blue		Throat of Blue Titmouse.	Stamina of Single Purple Anemone.	Blue Copper Ore.
25	Prussian Blue.		Beauty Spot on Wing of Mallard Drake.	Stamina of Bluish Purple Anemone.	Blue Copper Ore
26	Indigo Blue				Blue Copper Ore.
27	China Blue		Rhynchites Nitens	Back Parts of Gentian Flower.	Blue Copper Ore from Chessy.
28	Azure Blue.		Breast of Emerald-crested Manakin	Grape Hyacinth. Gentian.	Blue Copper Ore.
29	Ultra marine Blue.		Upper Side of the Wings of small blue Heath Butterfly.	Borrage.	Azure Stone or Lapis Lazuli.
30	Flax-flower Blue.		Light Parts of the Margin of the Wings of Devils Butterfly.	Flax flower.	Blue Copper Ore
31	Berlin Blue.		Wing Feathers of Jay.	Hepatica.	Blue Sapphire.
32	Verditter Blue				Lenticular Ore.
33	Greenish Blue			Great Fennel Flower.	Turquois. Flour Spar.
34	Greyish Blue.		Back of blue Titmouse	Small Fennel Flower.	Iron Earth.

Figure A.6.: Werner's nomenclature of 110 colors

A.5.2. Sundberg Color Thesaurus

Ingrid Sundberg, a collector of words, created the Sundberg Color Thesaurus, which conveys the color image for a limited number of color names in the form of a color palette. An extract of the Sundberg Color Thesaurus' is shown in Figure A.7.

A.5.3. EPFL Color Thesaurus

The EPFL Color Thesaurus is the main color name dictionary used during this project. The color names together with the corresponding color swatches are exhibited in Figure A.8

A.5.4. Linguistic Color Name Clusters

Natural language categories were found from categorizing the color names included in the color name dictionary EPFL Thesaurus into semantic clusters. In Figure A.9, they are sorted by most frequent to least frequent. The color names within each category were grouped into semantic clusters and visualized as a pie chart in Figure A.9. The set of familiar color naming terms are based on real-life categories used by speakers of English to name colors such as the name of food items, fruits and beverages; weather and water formations; stones, trees, flowers and plants, animals and bodily features, textures, industries, nationalities. Some color names bear colors that are not directly derivable from their linguistic meaning. For example, eggshells can be *brown* or *grayish-white*. However, conceptually the color *eggshell* can never be *brown*, but must denote a *grayish-white* color. Hence, color names and their semantic content must be learned as an ontology of its own.

Figure A.7.: Sundberg Color Thesaurus color palette

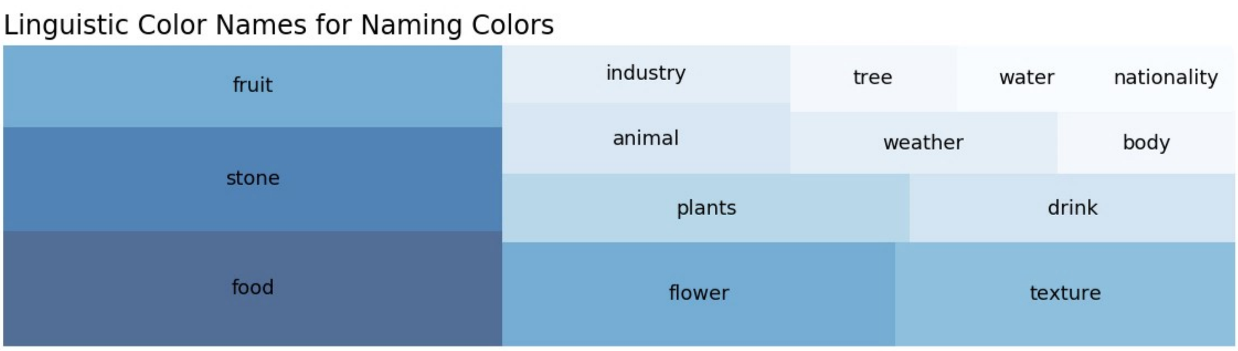


Figure A.9.: Linguistic categorical themes used in EPFL Thesaurus for naming colors

food	stone	fruit	flower	texture	plant	drink
bubblgum	ruby	cherry	rose	leather	straw	milk
cream	emerald	peach	geranium	silk	lichen	latte
chocolate	amethyst	melon	buttercup	lace	moss	coffee
vanilla	sapphire	lime	lavender	tiffany	mint	mocha
truffle	jade	lemon	peony	powder	fern	tea
cocoa	turquoise	berry	mauve	eggshell	algae	butterscotch
caramel	cobalt	grape	azalea	celadon	leaf	Burgundy
cinnemon	coral	tangerine	cornflower	bottle	sap	bordeaux
butter	pearl	banana	goldenrod	window	lawn	claret
egg	seashell	grape	marigold	denim	swamp	merlot
olive	clay	orange	carnation	velvet	grass	wine
soup	terracotta	grapefruit	sage	rust	evergreen	champagne
mushroom	ochre	apple	indigo	gold	shamrock	
pumpkin	sand	plum	lilac	copper	puce	
aubergine	sandstone	apricot	heather	bronze	wheat	
eggplant	mud	blueberry	violet	silver		
aspargus	slate	cerise	wisteria	brick		
celery	earth	raspberry	sunflower	steel		
avocado	brick	berry	dandelion	gunmetal		
pea	mud	cranberry	orchid			
maize	charcoal	watermelon	mulberry			
squash	petrol	strawberry	fuchsia			
tomato	amber					
potato	ocher					
chestnut	stone					
walnut	cement					
peanut	parchment					
hazel	desert					
pistachio						
mustard						
seaweed						

animal	industry	weather	body	tree	water	nationality
peacock	royal	sun	skin	willow	aqua	Prussian
oyster	navy	sunshine	blood	forest	seafoam	Indian
canary	hospital	storm	bruise	oak	ocean	British
duck	hunter	snow	blush	jungle	sea	Irish
teal	military	rain	tan	pine	spring	French
fawn	army	cloud	poo	mahogany		
camel	camo	sky				
pig	marine	frost				
salmon	battleship	ice				
sepia						
turtle						

Table A.5.: Linguistic color name categories and their color names

A.5.5. Color Name Dictionary

A Dictionary of Color Names is a list of unique color names such as *cream*, *seafoam* and *midnight blue* with a corresponding color value. A dictionary presupposes a mapping of the verbal color form to a numerical value. Such lists of colors encode color standards. Their purpose is to have a mathematical definition of a color name common to all users who choose to abide by the definitions set forth by the color standard. While the color value given online is typically a HEX code, the preferred color space for sampling and visualizing the colors is RGB (or BGR when taking the inverse of RGB). Since each RGB channel falls into the range of 0-255 , a Color Name Dictionary (CND) can consist of a maximum of $256x256x256 = 16'777'216$ color names. These are all possible colors of an RGB color cube. Other colors outside this range are possible in other color spaces but cannot be made perceivable to the human visual system. Most color name dictionaries are much shorter due to the relatively imprecise way of language in dealing with visible colors.

The purpose of a color name dictionary is to establish synonyms between color names and numeral or letter (HEX code) designations such that they can be used interchangeably. For example, *navy* is a color name that evokes a specific color and is understood by the public. In principle, *navy* can be translated to a set of color values that form a close-proximity cluster in a high-resolution color space.

Color name dictionaries can be acquired from online and offline sources. The advantage of offline sources of data is their availability in libraries. However, online sources may come in more formats, but the quality could be less reliable than from offline sources. Homepages that use color name dictionaries to drive online traffic usually do not disclose their color name dictionaries. The quality of the data depends on the number of surveyed users and matching methodology.

Color name dictionaries can be found in books or on websites dealing with color conversion. A good source of color name dictionaries are web tools that use them to generate colors. The following are some prominent examples of sources for retrieving a color name dictionary. Other examples include the Resene Colors, the Crayola colors, X11-colors, NCS, NBS, RAL colors and Colors USA, UK and Australia. The following three color name dictionaries are sorted in ascending importance of standardization and commercial impact. The EPFL Color Thesaurus was used in this project as an example for building a classifier from a color name dictionary. The others serve as illustrations of a color name dictionary.

1. The EPFL Color Thesaurus Albrecht Lindner et al. (2012) at the EPFL have created a Color Thesaurus `https://colorthesaurus.epfl.ch/`, which contains a color value for color names in a particular language. The dictionary was built based on the first 100 image results on Google Image Search for `color` and `color name` to derive a color name's color value. HSV channels as well as sRGB, HEX and CIE-Lab color values, are included. The dictionary can be used in conjunction with a color wheel picker.

2. The XKCD Color Survey Randall Munrow has polled the XKCD Color Survey (2010) with 200'000 participants that resulted in a data set with 954 most common RGB monitor colors, as defined by several hundred thousand participants in the color name survey. The participants – including color blind people - were made to name colors. This color name dictionary contains the color names and their corresponding HEX color codes.

3. The Inter-Society Color Council (ISCC) and the National Bureau of Standards (NBS) The Inter-Society Color Council (ISCC) and the National Bureau of Standards (NBS) (1976) have developed a lexicon called the ISCC-NBS Method of Designating Colors and a Dictionary of Color Names, usually shortened to the term Color Names Dictionary (CND). It is a system of color names based on five levels of lightness with default medium (very light, light, medium, dark, very dark) and four levels of saturation with default vivid (greyish, moderate, strong, vivid). The hues are specified on three different levels (13 basic colors, 29 nuanced colors, 267 adjective colors). Based on this system, a total of 627 color names are possible.

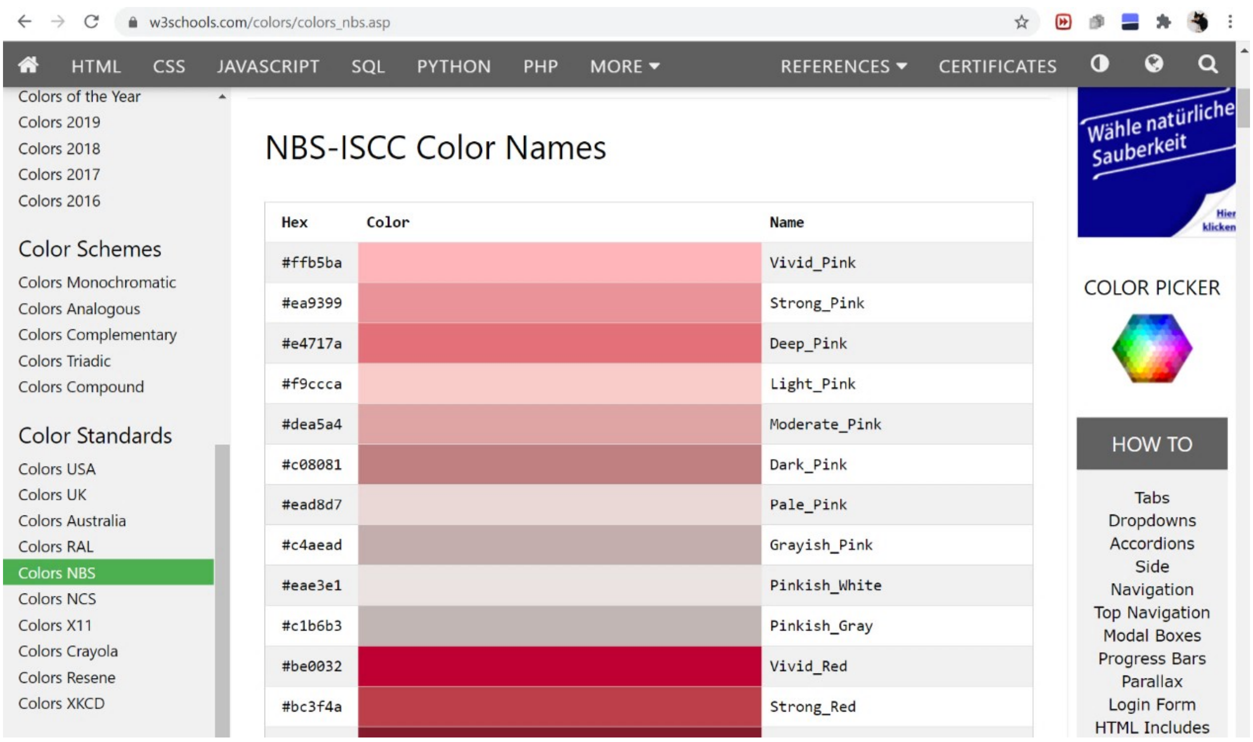


Figure A.10.: The NBS Color Name Dictionary on W3schools.com: each color name is matched to a color value, in this case it is the HEX code, the color image of the color name is also displayed

A.5.6. Fully-fledged Color Name Dictionary

The following attributes need to be present in a Fully-Fledged Color Name Dictionary (FFCND): id, lang, name, srgb, srgb_r, srgb_g, srgb_b, hsv, hsv_h, hsv_s, hsv_v, lab, lab_l, lab_a, lab_b, hex. Other color spaces can be added, such as HSL and LCH, but this is not necessary. The color's image can be plotted by filling up an array with the same RGB value per pixel. An example of a Fully-Fledged Color Name Dictionary with header, entries and type would look like Table A.6.

Id	lang	name	srgb	srgb_r	...	hsv	...	lab	...	Lab_b	hex
1	eng	Adobe	(249, 168, 27)	249				(70, 20, 75)		75	
2	eng	Algae	(81, 90, 50)	81				(37, -11, 22)		22	
3	eng	Amber	(184, 97, 25)	184				(51, 31, 52)		52	
...											
Int	Str	Str	Tuple	Int	...	Tuple	...	Tuple	...	Int	Str

Table A.6.: Fully-fledged Color Name Dictionary: header, entries and type

A.5.7. Extended Fully-fledged Color Name Dictionary

In total, an EFFCND needs to have the following attributes for a given system of basic colors: id, lang, name, srgb, srgb_r, srgb_g, srgb_b, hsv, hsv_h, hsv_s, hsv_v, lab, lab_l, lab_a, lab_b, hex, cat1, cat2. (HSL and LCH color spaces can be added optionally.) The color's image can be plotted by filling up an array with the same RGB value per pixel. In Table A.7, an extract of an extended fully-fledged color name dictionary for header, entries and type is shown.

Id	lang	name	srgb	...	hex	cat1	cat2	lab	...	Lab_b	hex
1	Eng	Adobe	(249, 168, 27)			*orange*		(70, 20, 75)		75	
2	Eng	Algae	(81, 90, 50)			*green*		(37, -11, 22)		22	
3	Eng	Amber	(184, 97, 25)			*orange*		(51, 31, 52)		52	
...											
Int	Str	Str	Tuple	...	Str	Str	str	Tuple	...	Int	Str

Table A.7.: Extended fully-fledged Color Name Dictionary: header, entries and type

A.6. Color Spaces

sRGB 0-255, sRGB 0-1.0, RGB Adobe 98, Adobe Wide Gamut RGB, ProPhoto RGB, scRGB, DCI-P3, Rec. 709, Rec. 2020, ACES, HSL 0-1.0, HSV 0-1.0, HSI 0-1.0, CMY 0-1.0, CMYK 0-1.0, CMYK 0-100, CIE-31 XYZ, CIE-64 UVW, Yxy, CIE-L*ab, CIE-L*uv, CIE-L*Ch(ab), CIE-L*Ch(uv), HunterLab, HEX, YIQ, YCbCr

A.7. Color Palettes

A color palette is a collection of relatively few colors established from images by reducing the full gamut of colors to a set of representative color pixels. The palette colors of the color palette are extracted to provide a coarse grain coverage of the color distribution associated with the collection of images. At the same time, palette colors may be determined from a high-resolution color space as a number triplet. Finding other forms of representation, such as in text or palette color patch are possible also. If a color's image is given and more than one such color image is displayed next to each other in spatial location, these color patches form a collection of colors, the so-called color palette.

A collection of color palettes is called a color palette data set matrix. A palette data set X contains an arbitrary number of color palettes N such that $X = P_1, \ldots, P_N$. A color palette P is defined as a collection of an arbitrary number of color patches K such that color palette $P_i = [p_1^1, \ldots, p_N^K]$ that are juxtaposed one after the other, sorted or unsorted.

For conventional flat color palettes, most color palettes found online have a color patch count of five. As for hierarchical color palettes, a plurality of palette colors can be arranged in a semantically-purposeful hierarchical structure having levels of different degrees of granularity. Both types of color palettes are shown in Figure A.11.

More formally, we define a color palette data set X, where X comprises a certain number N of color palettes P such that $X = P_1, \ldots, P_N$. Each color palette P is made of a set of k different colors p_n^k such that that palette $P_n = [p_n^1, \ldots, p_n^k]$.

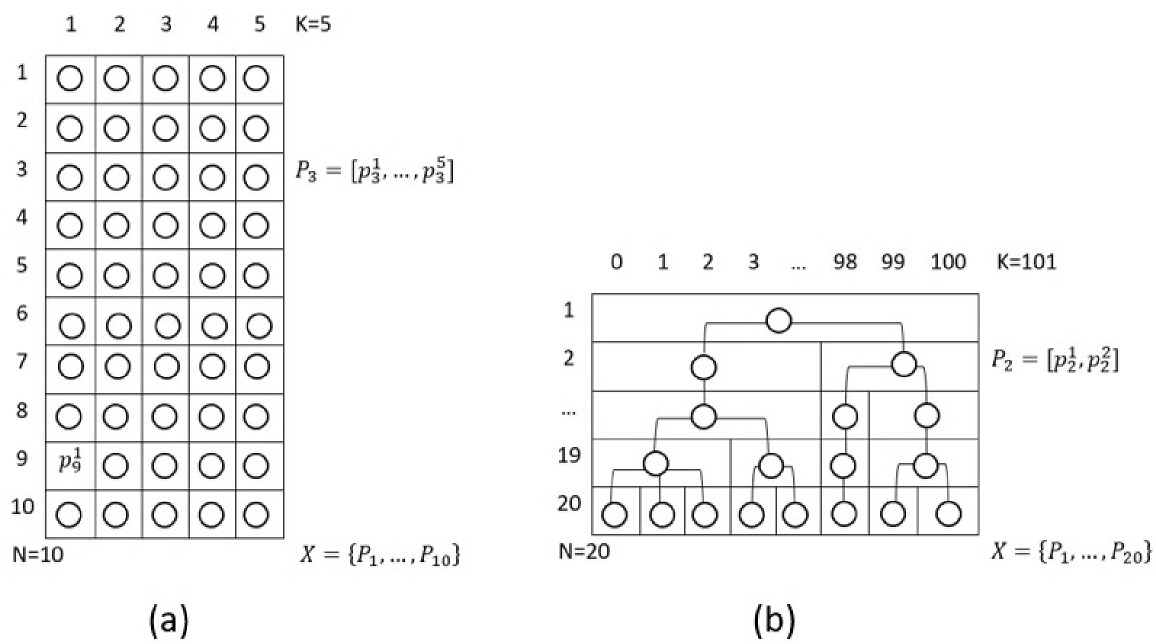


Figure A.11.: (a) Anatomy of a flat color palette; (b) anatomy of a hierarchical color palette

While flat color palettes typically consist of mutually independent colors and palettes, hierarchical color palettes comprise colors aggregated based on a bottom-up agglomerative clustering approach. Hence, they are interdependent across palettes. A **bin** is defined as the space occupied by a single color within a color palette. In a hierarchical color palette, such as in Figure A.12, the topmost palette P_1 has only one bin, whereas the lowest palette P_20 has the maximum number of bins a palette can have. In this case, there are 101 bins at the lowest level. When summing up all bins for each palette, the total number of bins equals 1024.

Figure A.12.: An example of video frame 45468's hierarchical color palette, drawn from the movie *Jigokumon*.

The color palette implementations are as diverse as their use-cases: the palette colors of a color palette may be hand-picked manually or generated automatically. For example, by agglomerative clustering the color distribution at different levels of granularity. The former method was chosen as part of the VIAN tool, an automated color palette extraction algorithm based on the SEEDS method. The SEEDS' color space is not LAB32. LAB32 is usually preferred, but it led to crashes in the C-code, which is why BGR was converted to LAB-U8. In VIAN, a color palette is the product of a SEEDS superpixel image segmentation followed by a bottom-up clustering. This clustering results in a hierarchically structured color palette, where every parent node is assigned the average

LAB color of its children, allowing the user to define the merge depth of interest with a comprehensive result.

Classification of color palettes and by extension images can only happen if there are categories that are fixed – palette colors and the number of possible color palettes, if they are extracted from images, are almost infinite in number. For color distributions that cannot be assigned to a task of classification, only tasks of color transformation can happen, such as (re-)colorization. However, color contrasts, harmonies and color schemes are fixed categories because they are one level higher up the hierarchy. In contrast, palette colors are more granular and hence, one level lower down.

A.8. Color Theorists

Eminent theorists of color have created color spaces (spheres, solids or trees) to locate and quantify color. A short overview of the most influential scholars on color theory is presented.

In color science, **Isaac Newton** (1642-1726) fundamentally shaped our modern understanding of light and color. He used a prism to refract white light to different degrees to revolve it into seven primary colors, *red*, *orange*, *yellow*, *green*, *blue*, *indigo* and *violet-purple* (ROY G BIV) [New52], that explains the colors of the rainbow as seen in Figure A.13. He observes that these original colors can be compounded to a variety of intermediate shades. He discovers the visible spectrum of light (400-700 millimicron wave lengths) – also called *gamut* - in a series of scientific experiments [New].

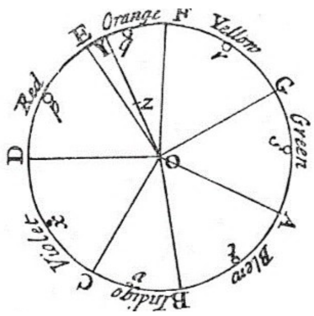
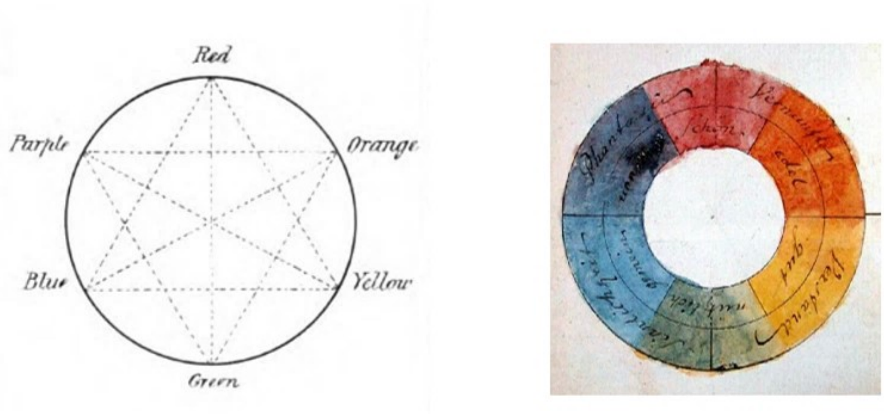


Figure A.13.: Newton's seven primary colors

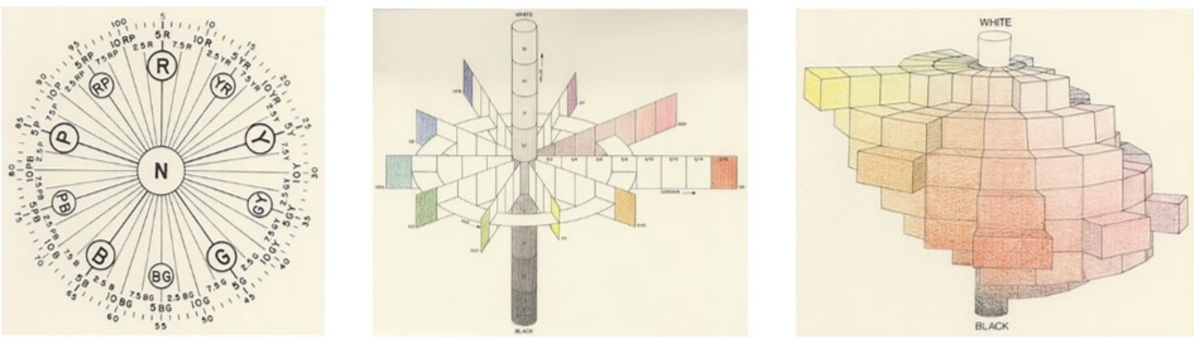
Johannes Wolfgang von Goethe (1749-1832) builds on top of Isaac Newton's understanding of color as a physical phenomenon to include the perceptual, psychological and subjective way of viewing colors. He makes contributions to color theory in book *Farbenlehre*[vG10] by deriving color contrasts, such as the effect of light and dark on the eye (*white* to *black*, the laws of color harmony, such as the three complementary colors (*yellow-red* to *blue*, *green* to *red* and *blue-red* to *yellow*), the achromatism and hyperchromatism of object-glasses, and the augmentation of color. He uses six primary colors to form based on using a prism to elaborate a color star as shown in Figure A.14 – *red*, *blue* and *yellow* are the original colors while *orange*, *violet* and *green* are their complements. He assigns the *yellow-blue* contrast in color to the plus-minus polarity. He creates a scale of 44 colors attained by the coloring of metals in a physical experiment.



(a) (b)

Figure A.14.: (a) Goethe's color star; (b) Goethe's color star

Albert Munsell (1858-1918) develops a color system based on a three-dimensional color space along the axes of hue, value and chroma (similar to saturation) [Mun05]. First, he introduces five basic colors: *red*, *yellow*, *green*, *blue* and *purple* which are visually equidistant in hue (like peeling an orange into sections) and adds another five colors between them as seen in Figure A.15. Then, he divides the hue wheel into 100 equidistant steps, sections the value axis into ten units and suggests a irregular chromatic scale, as shown in Figure A.15. He devised this color system to communicate colors more easily along a color tree. He has combined the art and science of color into a single color theory for matching colors more effectively.



(a) (b) (c)

Figure A.15.: (a) Munsell's ten hues; (b) Munsell's color tree - hue and value; (c) Munsell's color tree - open-scale chroma

Johannes Itten (1888-1967) uses subjective feelings and objective color principles to describe his theory of color [IB70]. His famous *color star* of twelve color hues which is shown in Figure A.17 is used to develop different methods of color contrast [Itt74]. He has a remarkable impact on present-day color theory through his development of color contrasts [Itt74]. He derived many contrasts such as contrast by hue, contrast by value (light-and-dark contrast), contrast by temperature (cold-warm contrast), contrast by saturation, contrast by extension (based on Goethe), contrast by complements (neutralization) and simultaneous contrast. His insights are fundamental to the development of a contrast classification algorithm, as seen in this project.

(a) (b)

Figure A.16.: (a) Johannes Itten's color star "Farbstern"; (b) Johannes Itten's color circle "Farbkreis"

Josef Albers (1888-1976) systematically demonstrated color's elasticity by making one color look like two, two like one and three like two [AW06]. He conducted studies on simultaneous contrast in color deception [AW06]. His thought on color relativity has changed the face of color theory: he distinguishes factual colors from actual colors (physical fact and psychic effect) [AW06]. **Factual colors** are defined as how we name, measure or locate color (theory, rule-based, fixed, absolute, isolated). **Actual color** refers to how color appears when it exists in a context (practice, trial-and-error, fluid, contingent, integrated).

A.9. Color Extraction

The web tool *Adobe Color* is used for color extraction `https://color.adobe.com/de/create/image`. On the web tool, click `create` then `theme extraction`. The user can drag-and-drop an image for which a color scheme of five color patches will be automatically extracted from data points on the image. There are six different ways of placing the data points on the image: colorful, light, muted, strong, dark, none. The color scheme's HEX color codes are displayed below the color patch. The HEX color codes can be copied to the clipboard using a click functionality. The web tool's save&use functionality is only open to users with an *Adobe Creative Cloud* account. In general, the number of color patches in a color scheme can grow from one to five (Adobe Color) to a hundred-one (VIAN) to an infinite amount of colors. That is for a gradient range between two to infinite many data points on an image (see Adobe Color web tool). Such colors are directly taken from the image, i.e. they are neither an aggregate average of an area in the image nor an aggregate of multiple disparate color points.

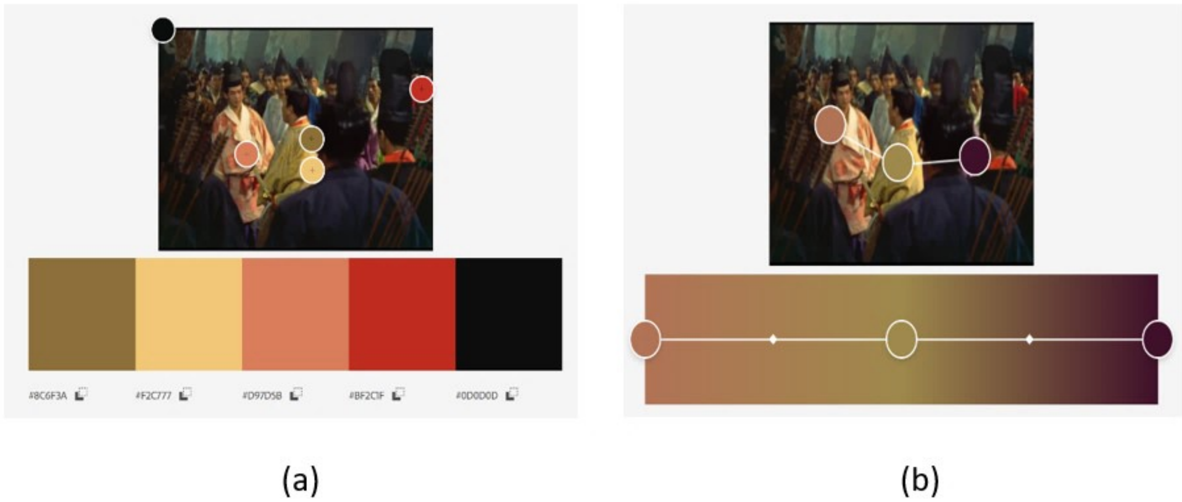


Figure A.17.: (a) Color palette extracted from an image; (b) color range extracted from an image

In contrast, color schemes of an image can hold colors that are not direct image color take-outs but derived by aggregation. For example, the aggregation could occur by selecting an area in the image and averaging all color pixels in the area to form a superpixel. Another method is to lay a grid on the image to extract direct color pixel take-outs and perform a hierarchical agglomeration on these colors from the lowest to the highest level. The highest level in the hierarchy consists of a single color that is the average of all colors. This average of all colors extracted from an image is the average color of an image. It is not equal to the dominant color of an image, which is the color pixel that occupies the most space in an image.


EFFCND Thesaurus-VIAN
light grey
pastel red
bright red
blood red
dusky rose
faded red
dried blood
blood
red
burnt red
brick red
fire engine red
brownish pink
dirty pink
dusky pink
scarlet
reddish
bruise
indian red
mahogany
rust red
orangey red
dull red
vermillion
leather
salmon pink
tomato
reddish orange
tomato red
almost black
brownish red
dark peach
orangered
orangish red
dark coral
red orange
black
salmon
brownish grey
rusty red
coral
clay
orange pink
dark salmon
terracota
orange red
pinkish orange
tangerine
blood orange
pinkish brown
dark khaki
light mauve
reddish brown
blush pink
pale rose
grey brown
peachy pink
blush
burnt yellow
adobe
yellow ochre
dark mustard
yellow tan
ugly brown
mustard
squash
golden rod
golden yellow
mustard yellow
marigold
orangey yellow
dark yellow
tan green
dark gold
tea
sandy yellow
brown yellow
yellow brown
yellow orange
yellowish orange
sandstone
brownish yellow
golden
sand yellow
straw
gold
muddy green
dark sage
faded yellow
dull yellow
greeny grey
off yellow
dark grey
greenish beige
green brown
greenish brown
goldenrod
browny green
lichen
maize
brown green
moss
pastel yellow
brownish green
butter
banana
canary
butter yellow
canary yellow
banana yellow
poo
sunflower yellow
yellow
sunflower
pale olive
dark olive
light seafoam green
light seafoam
greenish cyan
seafoam
forrest green
hunter green
mint
bluish grey
dark mint
ocean green
hospital green
racing green
pale aqua
dark seafoam green
duck egg blue
slate green
light forest green
dark seafoam
greenish turquoise
aqua green
british racing green
light blue green
pale teal
teal green
viridian
light sea green
greenish blue
greenish teal
bluey green
bluish green
light greenish blue
deep aqua
robin's egg
pale cyan
light teal
pale turquoise
eggshell blue
slate
robin egg blue
robin's egg blue
greyish green
pale
light aqua
light turquoise
tiffany blue
dark aqua
teal
bright turquoise
dark cyan
bluegreen
blue green
aqua
aqua marine
cyan
turquoise
ocean
turquoise blue
seafoam blue
navy
perrywinkle
sapphire
dark navy blue
dark periwinkle
periwinkle
indigo
dark navy
lavender blue
bluey purple
pale grey
light lilac
midnight purple
heather
dark lavender
very light purple
royal purple
dark violet
light purple
dark lilac
deep lavender
purple
light lavendar
deep violet
lavender
grey purple
grape purple
light lavender
lilac
greyish purple
wisteria
medium purple
pale purple
deep lilac
light violet
pale lilac
purplish grey
soft purple
warm purple
pastel purple
purpleish
darkish purple
purple grey
dusky purple
pale lavender
baby purple
bluey grey
liliac
dirty purple
faded purple
grape
purplish
dark purple
deep purple
pale violet
dusty purple
very dark purple
amethyst

A. Appendix I


dark sand
cocoa
red brown
dull pink
russet
cherry
reddy brown
chocolate
light peach
deep brown
burnt umber
light salmon
rust
pinkish tan
copper
faded orange
pale salmon
hazel
burnt siena
melon
rust orange
creme
earth
burnt orange
medium brown
burnt sienna
dull orange
chocolate brown
baby poop
brick
chestnut
poo brown
toupe
poop
dark tan
terracotta
milk chocolate
rust brown
brown
tan
greeny brown
dark cream
bright orange
deep orange
greyish brown
dark orange
sienna
terra cotta
pale orange
brick orange
peach
light orange
orange brown
grapefruit
rusty orange
very dark brown
browny orange
olive brown
dusty orange
brownish orange
greenish tan
pale yellow
mud green
sun yellow
sunshine yellow
medium grey
off white
light yellow
lemon
lemon yellow
khaki green
military green
cool grey
olive yellow
sage
dandelion
green yellow
yellowish green
yellow green
white
eggshell
drab
pale olive green
pea soup
avocado
light olive
bright olive
neon yellow
pea green
forest
asparagus
pea soup green
avocado green
light olive green
army green
camouflage green
yellowgreen
moss green
celery
light grey green
dark lime
dark olive green
fern
camo green
algae
pea
tea green
olive
olive green
light yellow green
light lime
soft green
light lime green
apple green
very pale green
cool green
pistachio
dull green
light moss green
pale lime
dark forest green
deep teal
aqua blue
aquamarine
teal blue
greyish teal
grey teal
bright aqua
dark turquoise
deep turquoise
tealish
steel
dusty teal
dull teal
greeny blue
bright cyan
dark teal
bright light blue
azure
light aquamarine
dark aquamarine
muted blue
warm grey
light sky blue
pale sky blue
sea blue
bright sky blue
ice blue
pastel blue
slate grey
baby blue
peacock blue
gunmetal
very light blue
really light blue
sea
powder blue
sky blue
ocean blue
cool blue
pale blue
very pale blue
soft blue
light light blue
water blue
light blue
cadet blue
light cyan
lightblue
greyish blue
grey blue
charcoal
petrol
dirty blue
charcoal grey
greyish
steel blue
blue grey
dusty blue
greyblue
eggplant
rich purple
eggplant purple
plum purple
light rose
lavender pink
twilight
dusk
red purple
orchid
aubergine
plum
pale magenta
red violet
purplish pink
pinky purple
purply pink
barney
purpley pink
magenta
pinkish
dusty lavender
pink
bright pink
hot pink
violet red
dark magenta
reddish purple
dull purple
mulberry
deep magenta
greyish pink
dark plum
light plum
dark hot pink
rose
muted purple
neon pink
dark fuchsia
dark pink
purple red
light pink
dust
purple brown
shocking pink
baby pink
light eggplant
rose pink
bubblegum pink
deep pink
old rose
brownish purple
purplish brown
rosa
bubble gum pink
pinkish grey
purpley grey
bubblegum
carnation pink
mauve

Figure A.8.: EPFL Color Thesaurus

B. Appendix II

B.1. Multi-class Classification Strategies

There are two multi-class classification strategies: One-vs-All and On-vs-Rest. While One-vs-All can handle multi-class and multi-label classification, One-vs-Rest can only handle multi-class problems.

1. One-vs-all (OvA) classifier: also known as One-vs-Rest, this strategy comprises the fitting of one classifier per class. One-vs-all is the default strategy for multi-class classification. A class is fitted against all the other classes for each classifier. It is computationally efficient and easily interpretable. In addition, it is possible to gain knowledge about the class by inspecting its corresponding classifier. This strategy can also be used for multi-label learning by fitting on a 2-D matrix in which cell $[i, j]$ is 1 if sample i has label j and 0 otherwise. It requires to fit as many classifiers as there are classes. This strategy can be used with basic color names; with all color names as basic colors, the training period will take a considerable amount of time since one classifier needs to be trained per class.

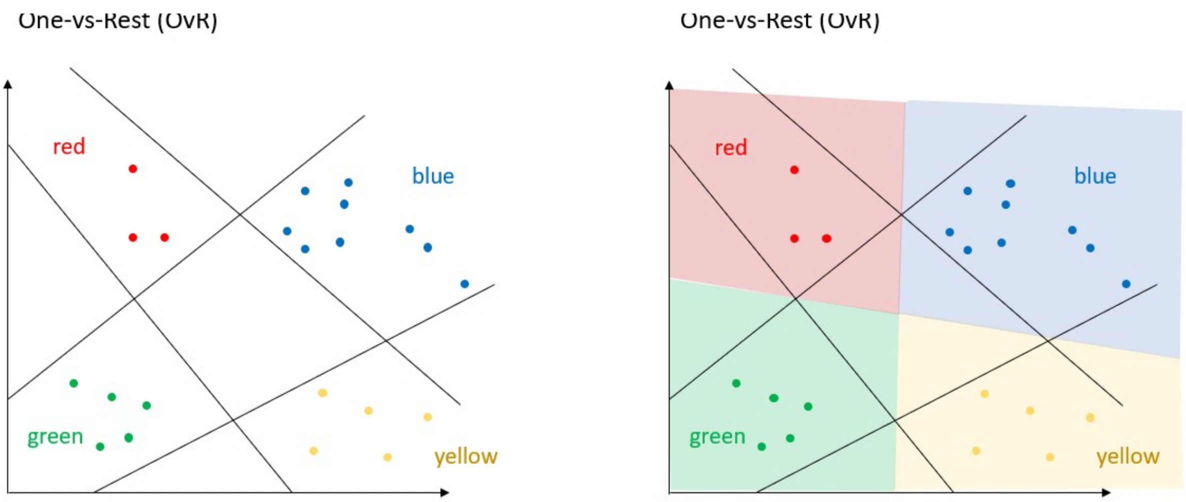


Figure B.1.: One-vs-Rest (OvR): computing decision boundaries

2. One-vs-one (OvO) classifier: the One-vs-One strategy comprises the fitting of one classifier for all combinations of class pairs. It requires the training of $n\_classes * (n\_classes - 1)/2$ classifiers. This method is usually slower than the One-vs-All classifier due to its $O(n\_classes^2)$ complexity. The advantage of this method may come to light with kernel algorithms that do not scale well with $n\_samples$, because each learning problem only involves a small subset of the data. In One-vs-Rest, the complete data set is used $n\_classes$ times. For example, in a multi-class classification problem with four classes: *red*, *blue*, *green* and *yellow* a One-vs-All strategy consists of four binary classification data sets as follows:

1. Problem 1: *red* vs *blue-green-yellow*
2. Problem 2: *blue* vs *red-green-yellow*

3. Problem 3: *green* vs *red-blue-yellow*
4. Problem 4: *yellow* vs *red-blue-green*

However, in a One-vs-One strategy, the binary classification data sets extend one data set for each class versus every other class as shown in the following:

1. Problem 1: *red* vs. *blue*
2. Problem 2: *red* vs. *green*
3. Problem 3: *red* vs. *yellow*
4. Problem 4: *blue* vs. *green*
5. Problem 5: *blue* vs. *yellow*
6. Problem 6: *green* vs. *yellow*

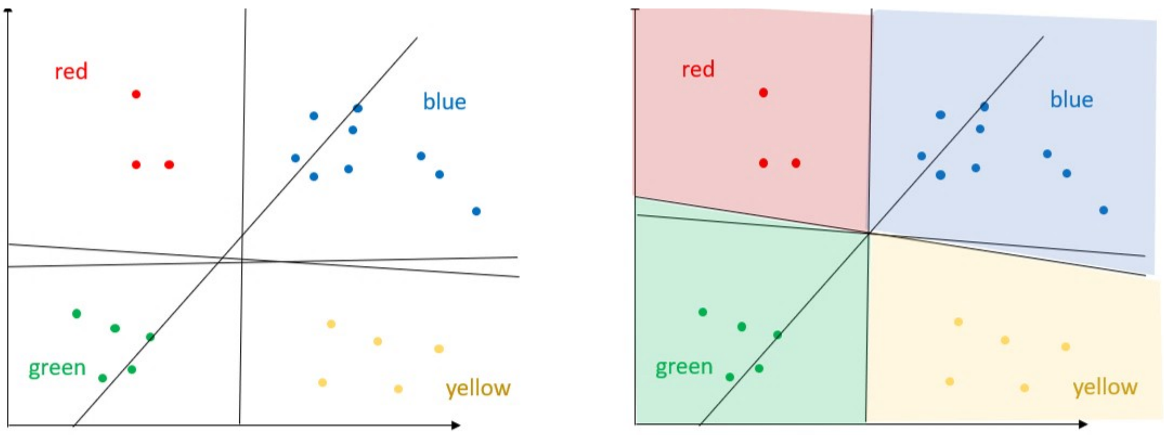


Figure B.2.: One-vs-One (OvO): computing decision boundaries

These are two more data sets than in the OvA strategy. Nevertheless, the decision boundary's function shape is the same for either OvR or OvO. For reasons of performance, the OvA strategy should be preferred over the OvO strategy. For OvA, either a *micro average* or a *macro average* can be chosen. In *micro average*, all components of performance are summed for each class. In *macro average*, the performances of each class are averaged.

B.2. Color Contrasts

In order of appearance, the rules of operation need to be derived for the following six classifiable color contrasts adapted from Johannes Itten's color theory for categorizing images into color contrasts. The definitions of the color contrasts are given below.

1. Contrast of hue At least three distinct hues are required to achieve this effect. On VIAN, the definition extends to rather pure colors than muted colors. While complementary colors have the highest contrast of hue, analogous colors have the lowest contrast of hue. The hues considered are *red*, *orange*, *yellow*, *green*, *blue* and *violet* because only these six colors are referred to be Johannes Itten in his book *The Art of Color*. These hues are made up of the three primary and three secondary colors. A combination of three primaries is the most extreme

example of a contrast of hue. The more hues are employed further away from these three primaries, the less high the contrast intensity of the contrast of hue. The three secondaries is a weaker color combination than the three primaries when forming contrasts of hue.
Example combinations: *yellow*, *red*, *blue*; *red*, *blue*, *green*; *blue*, *yellow*, *violet*; *yellow*, *green*, *violet*, *red*; *violet*, *green*, *blue*, *orange*, *black*.

2. Light-dark contrast This contrast is achieved when an image simultaneously depicts dark and light colors such that the contrast is striking. The most definite instance of light-dark contrast is a black-and-white image. This property is also defined on VIAN. VIAN further remarks that there are all kinds of *gray* shades in-between and chromatic color values. Johannes Itten states that the further away removed from a white-black image into a monotonous *gray* or grayed chromatic image, the lower the light-dark contrast.
In *Art of Color*, Johannes Itten illustrates chromatic light-dark contrast using twelve equidistant steps of lightness across twelve basic hues of the color circle to form a $12x12$ matrix. In the matrix, *yellow* is the lightest and *violet* the darkest of all saturated hues (*yellow*: 4 steps, *green* and *orange*: 6 steps, *red* and *blue*: 8 steps, *violet*: 10 steps). Therefore, the strongest light-dark contrast among hues are complementary colors *violet-yellow* (stated on VIAN, too).

3. Cold-warm contrast The cold-warm contrast makes use of temperature to divide hues into cold and warm. For the six basic ITTEN colors, the warm colors are *red*, *orange*, *yellow* and the cold colors are *green*, *blue* and *violet*. As explained on VIAN: *"colors like yellow, orange and red are commonly considered as warm and green, blue and violet are cold colors – however, this classification is relative."* In general, the colors in each class can be reached by juxtaposing *blue* to *green* and *violet* to *blue*, hence these colors are analogous to each other on a color wheel. An image containing both cool and warm colors can demonstrate a cold-warm contrast. At the center of each is *orange* and *blue*. These two colors are complementary to each other. Thus, the strongest cold-warm contrast is orange-blue.
While color can be classified in absolute terms into either warm or cold, warm or cold colors can be classified differently when viewing colors in relative terms. A color usually classified as warm can become cold and a cold color can be warm when juxtaposed to a colder or warmer color. For example, *red* is a warm color. However, if it is situated next to *orange*, it is the colder color of the two colors. *violet* is a cold color. However, if it is situated next to *blue*, it is the warmer color of the two colors. Therefore, cold-warm contrasts can occur within a set of only warm or only cold colors as well.
Suppose the relativist approach is taken instead of an absolutist approach. In that case, all non-monochromatic images will have a cold-warm contrast because it suffices to have two different colors to be able to say that one color is colder or warmer than the other. In the vast majority of cases, that would mean all images have a cold-warm contrast. However, the task is to have an image with and without a cold-warm contrast. Hence, an absolutist approach is favored in this project because it shrinks the number of possible images when filtering for a cold-warm contrast. The cold-warm contrast can be illustrated by the following metaphorical opposites listed in Table B.1.

Cold	Warm
Blue	Orange
Wet	Dry
Water	Soil
Business	Home
Man	Woman
Airy	Dense
Future	Past
Calm	Vivid
New	Old
Silver	Gold

Table B.1.: Cold versus warm's other metaphorical and semantic opposites

4. Complementary contrast The contrast of these complementary color pairs stems from the opposing effect of two colors when situated adjacent to each other. On VIAN, this is described as *"complementary colors are color pairs – when mixed, they produce neutral gray"*. The eye demands this effect. Two colors in an image are complementary if mixed together, they produce *gray* or *white* depending if color is a material or light. In Johannes Itten's words, the physical mixture of complementary colors yields *white*; the pigmentary mixture, however, yields *gray-black*. Thus, the opposition is evened out by the mixing of the complementary colors [Itt74]. For a given color, there exists only one color complementary that is at the diametrical opposite of the color. While all colors situated at 180° from each other on the color wheel are complementary, the following color pairs are the commonly illustrated six complementary colors:

a. *Red-green* (r-g)
b. *Blue-orange* (b-o)
c. *Violet-yellow* (v-y)

5. Contrast of saturation This contrast plays out when pure, intense colors are placed next to toned-down, diluted colors in an image. The presence of both luminous and greyish colors produces this contrast. On VIAN, this is explained by the opposition of saturated, clear colors with desaturated toned-down colors. It is further remarked that the contrast between both is relative.
While other color contrasts, for example, a light-dark contrast has the complementary color pair *violet-yellow* as a prime example of light-dark contrast; a cold-warm contrast is best exemplified in complementary color pair *blue-orange*, it is predicted that complementary colors *green-red* could serve as a prime example of a contrast of saturation. In that case, *red* would be the most saturated and *green* the most desaturated color among all six basic colors. A logical basis for such a statement still needs to be found.

6. Contrast of extension A contrast of extension (or contrast of proportion) is present whenever the colors employed occupy such disharmonic areas relative to each other to make the minority color look distressful and more provocative in resistance to the majority color. This contrast vanishes when harmonic and occurs when colors are disharmonic. In disharmony, the minority color dominates the scene and the effect is expressive.
On VIAN, this is defined as a contrast of quantity concerning the proportion of size among at least two daubs of color on an image. Hence, it is a contrast of many and less, big and small. However, in practice, there is only one color present surrounded by achromatic *gray*. For this reason, a practical definition of the contrast of extension could be a small saturated color area against a big, more desaturated area. Such a contrast is typically seen in a small *red* area in an image that pops out against a more neutral background area.
A contrast of extension can intensify other contrasts such as the light-dark contrast. For instance, if the dominant dark sky contrasts with a dot of a bright star, there is a contrast of extension because the sky enhances the significance of the star by occupying large, disharmonic proportions in the area surrounding the star.

B.3. Color Contrast Classification

PDF files were generated for machine-classified and hand-labeled color contrasts. An excerpt from these files entitled *FilmColors Project: Film Screenshots and Information about Color Contrasts* is shown in Figure B.3. There exist versions with single and multiple images per page, classi fied and hand-labeled.

FilmColors Project:
Film Screenshots and Information about Color Contrasts

movie_name : Jigokumon
project_id : 7
genres : Melodrama,History,Romance
year : 1953
countries : Japan

Number of segments: 54 [312, 365]
Number of screenshots: 569 [45442, 46010]

Color Contrast Classification:

1. Contrast of hue: At least 3 different colors exist.

2. Light-dark contrast: Luminance values (LCH color space) of more than 75 and less than 25 are simultaneously present.

3. Cold-warm contrast: At least one cold and one warm color is present at the same time.

4. Complementary contrast: At least green-red or blue-orange or violet-yellow are simultaneously present.

5. Contrast of saturation: Saturation values (LCH color space) of more than 50 and less than 25 are simultaneously present.

6. Contrast of extension: Sub-category of contrast of saturation where a saturated area of less than 2%, but bigger than .5% pops out against a desaturated area of more than 98%.

DISCLAIMER: Medium ranges for lumens and tone are not indicated (dark<25, light>75) as they are not contributing factors for contrast classification.

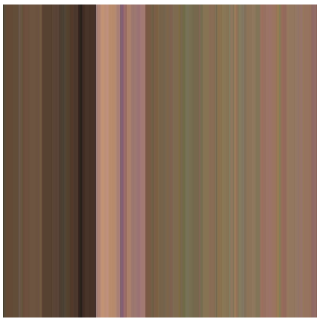
Figure B.3.: First page of *FilmColors Project: Film Screenshots and Information about Color Contrast* of movie *Jigokumon*.

Screenshot: 1

Start of Segment

segment_id : 364

screenshot_id : 45442



Color Contrasts:

- Contrast of hue (4 colors)
- Cold-warm contrast (cold colors: 13.32, warm colors: 86.69)
- Complementary contrast: green-red (12.14: 3.99)

Colors:

- Red: 3.99
- Orange: 82.7
- Green: 12.14
- Violet: 1.18

Lumens:

- Dark: 5.58

Tone:

- Desaturated: 95.65

Figure B.4.: Second page out of 570 pages of machine-classified *FilmColors Project: Film Screenshots and Information about Color Contrast* of movie *Jigokumon*.

Segment: 1 (segment_id : 364)

Number of screenshots: 10, screenshots = [45442, 45451]

Color Contrasts:

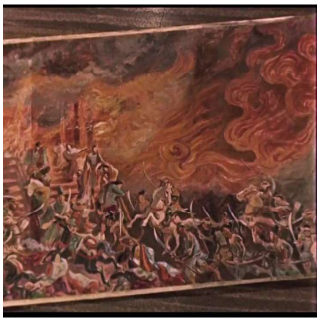
- light_dark_contrast

45442.jpg

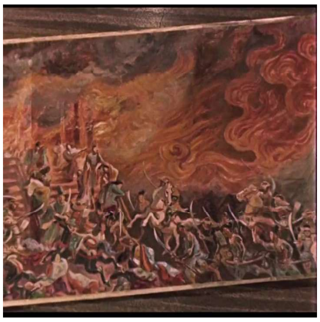
45443.jpg

45444.jpg

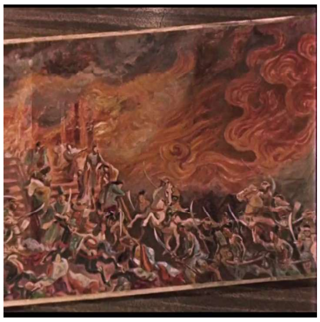
45445.jpg



45446.jpg



45447.jpg



45448.jpg

45449.jpg

45450.jpg

Figure B.5.: Second page out of 88 pages of hand-labeled *FilmColors Project: Film Screenshots and Information about Color Contrast* of movie *Jigokumon*.

C. Appendix III

C.1. Color Conversion Tools

The preferred websites for converting color values to different color spaces are as follows.

1. Nixsensor.org
2. EasyRGB.com
3. colorizer.org
4. Rapidtables.com
5. Colorthesaurus.com

Some of the websites display the color, some only deal with color values.

C.2. Color Palette Search Tools

1. Coolors
Description: Adobe's *Coolors* has two main features: generating color palettes and exploring trending color palettes. In searching trending color palettes, a search bar pops up where basic colors can be searched. These basic colors are *red*, *orange*, *brown*, *yellow*, *green*, *turquoise*, *blue*, *violet*, *pink* and *gray* – a total of 10 basic colors. The searched color name will be added as a tag to the filters. The filters can be extended to more color names in the search query. The searched basic colors are mapped to a set of HEX colors for fetching corresponding color schemes. It suffices to have one color in this set of HEX colors included in the color scheme for the color scheme to display in the search results. The color scheme has the HEX color shown when hovering over a color patch in the color scheme. The resulting color schemes contain five color patches. These color schemes are made by *Coolors*' users. The color schemes can be viewed, saved or exported. Also, there exists the option to open the chosen palette in the *Coolors* color palette generator or copying the URL of the color scheme.
Evaluation: It is also possible to search color schemes by HEX code, but the possibility to search color palettes by HEX code is not obvious in the search bar. There are no names for the color schemes being displayed.

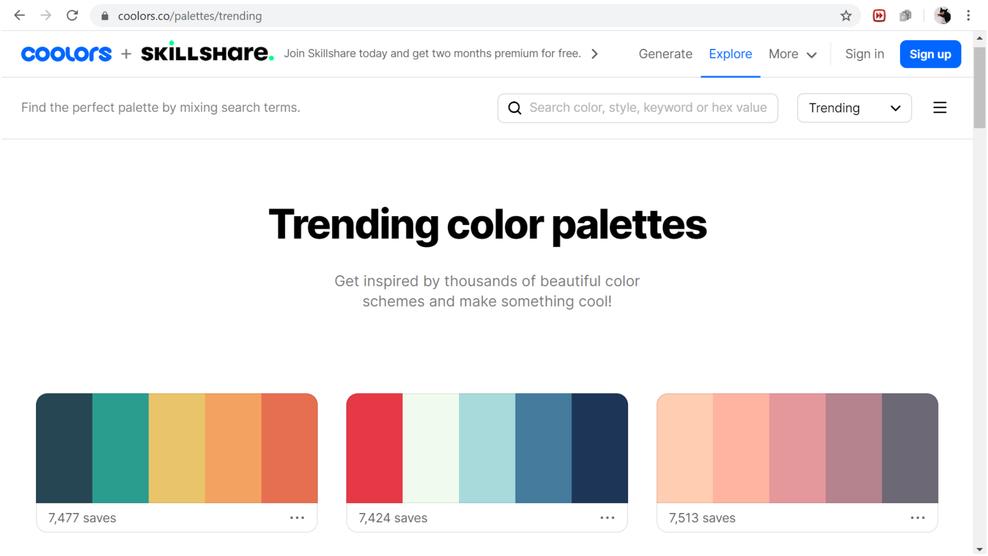


Figure C.1.: Coolors' web page

2. Muzli – Designers' Secret Source

Description: Searching for color names, such as *aliceblue*, *dark* or *antiquewhite*, or code (HEX code) in the search bar. Color name is converted to a HEX code. Color name is mapped to one HEX code only. For example, *brown* is "#A52A2A". Color scheme of five color patches are given in the result. Color schemes are analogic, mono, triade, complementary, tetrade and random.

Evaluation: *Azure* as "#F0FFFF" is not how the general public sees the color *azure*. "Dark" is mapped to "#222222" only, but conceptually "dark" could be any color. In the search bar, some color names are recommended in a list after typing the first few letters; however, the colors are mostly neutrals or very light or dark colors only.

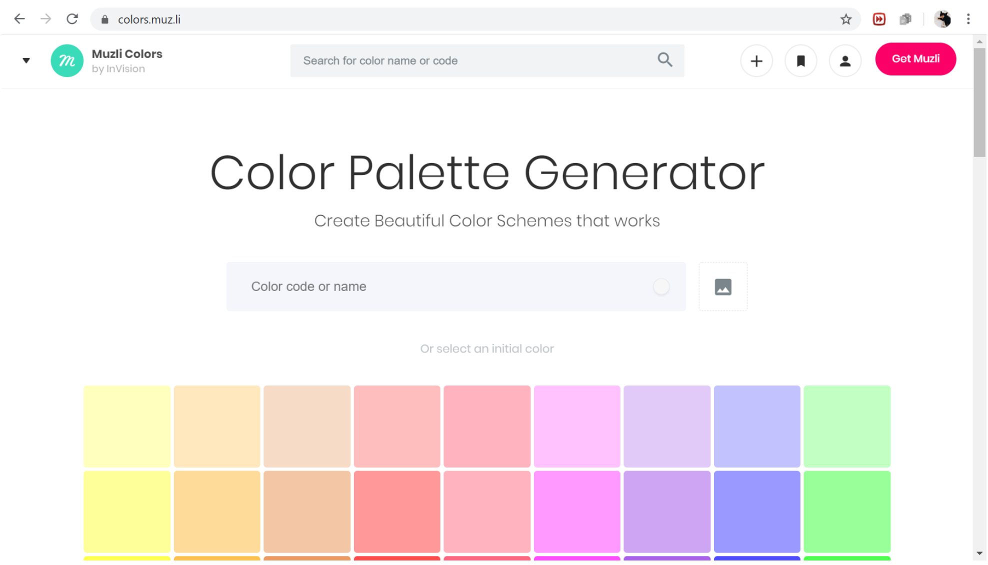


Figure C.2.: Muzli' web page

3. ColourLovers

Description: The site has a color palette maker and a search tool for exploring user-created color palettes. The search bar for color palettes is more sophisticated: not only is it possible to search color palettes by key (color names such as *wine*, *chocolate* and *teal*) and HEX code, it is also possible to search colors by RGB and HSV values. The HSV axes are displayed as color bars next to the color picker window square. The search results are sortable by loves, name, date added, views and comments in descending or ascending order. Also, they can be sorted by hues. The search results can be filtered by publication date. The palettes in the search result are named, the user naming the palette is displayed too. In addition, each color palette comes with information about the number of comments, favorites, views and loves it had attained so far. There are five color swatches in a color scheme.

Evaluation: Although this site's search functionality is the most sophisticated compared to other similar service providers, the mappings of *teal*" to the color palettes containing the color is more fallacious than not. However, for *goldenrod*, it is surprisingly accurate. In between searches such as *crimson* are some false results mixed with some correct search results. Hence, it is not clear how the color names are mapped to the colors in the color palette. It is not clear why the color swatches vary in proportion in some color schemes and in others, they are equally spaced.

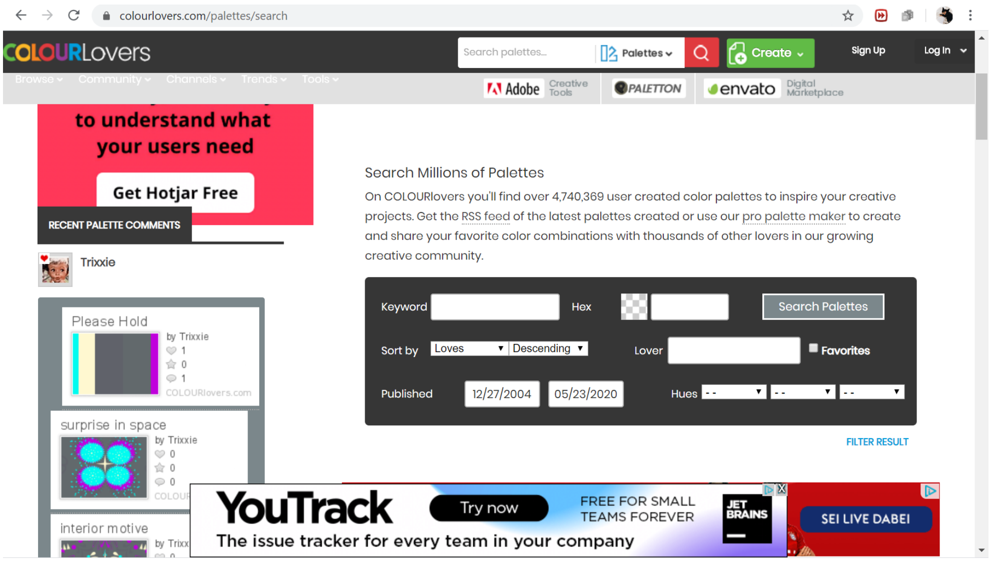


Figure C.3.: ColourLovers' web page

4. Color Hunts

Description: The search bar for color palettes suggest the color hues that are searchable which are *red*, *orange*, *brown*, *yellow*, *green*, *turquoise*, *blue*, *purple*, *pink*, *gray*, *black* and *white* – twelve hues in total. The color palettes are also searchable by contrast, such as warm, cold, bright and dark. The color palettes can be searched by texture such as gold, neon, pastel, skin or by themes such as vintage, retro, wedding, Christmas and Halloween. Searching color palettes by climates such as sunset, summer, autumn, winter and spring is possible. The search results contain four color patches and information about the number of likes and the elapsed period since the color scheme has been published online. When hovering over the color swatches the HEX code of the color displays.

Evaluation: The color palettes are not named. While all color swatches' proportional size remains equal across the color schemes, it is not clear why the colors occupy varying proportionate spaces within the color scheme.

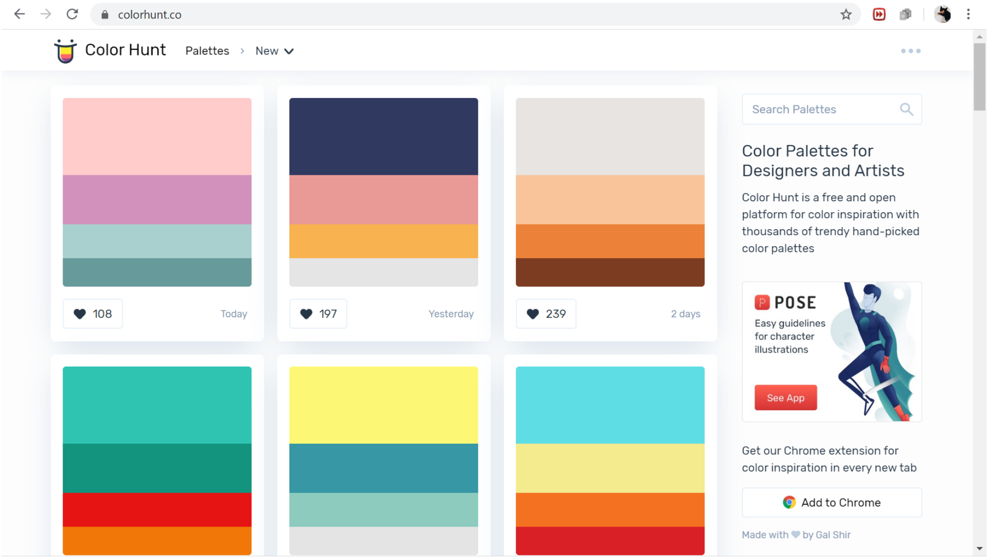


Figure C.4.: Color Hunt' web page

5. Design Seeds for all who love color

Description: The search functionality for color palettes is triggered by clicking on a color swatch in a 4x11 matrix of color swatches. The color swatches correspond to color names such as *blush*, *ice* or *metal* which are linked to a set of HEX codes. If a color palette has one of these HEX code colors, it figures among the search results. The search results contain all the color palettes by name, date of publication, color patches and username. There are always six color patches within a color scheme. The resulting color of a color palette is displayed with their HEX code.

Evaluation: The color names of the searchable colors are not displayed on the searchable color chart. The image from which the color scheme is derived is viewable, but one color scheme might take up too much space because of the clean and informative exhibition of the color schemes.

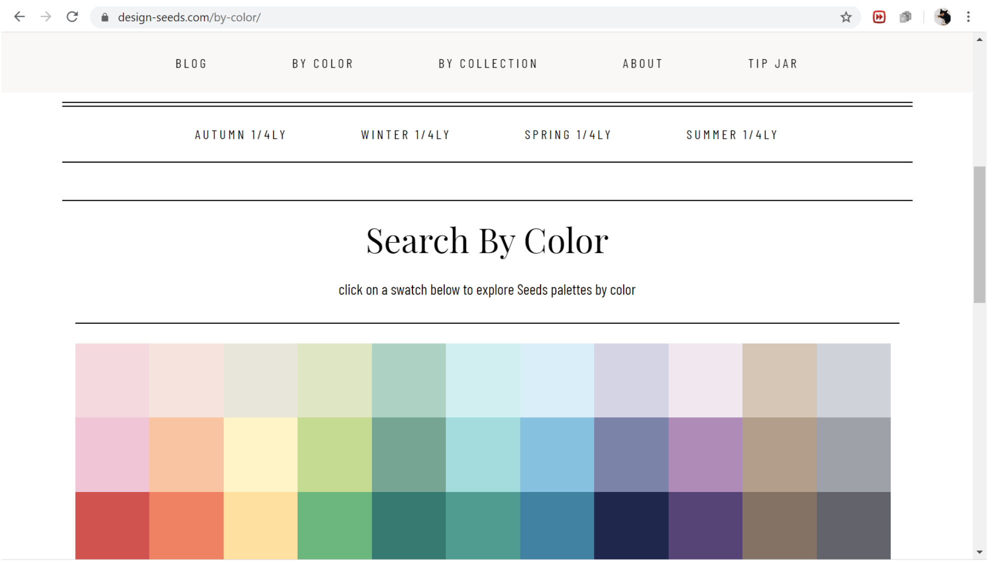


Figure C.5.: Design Seeds' web page

6. Color-hex

Description: The search bar takes color names such as *chartreuse*, *baby blue* or *sunflower*, but also proper names such as "Arabica", "Paris" or "Alps" and adjectives "frenzy", "playful" or "bitter". The search returns color

schemes (called color palettes) containing five color patches and the name of the color scheme matched to the searched keyword. The name of the color scheme usually follows the pattern "left-most-color-name to right-most-color-name", but descriptive names are also possible. The search results are clickable. On click, the color scheme is shown more prominently, with its title name, the username of the person who created the color scheme, and the color patches of the color scheme together with a list of its HEX code (on hovering as well) and RGB values. The color scheme is downloadable as a PNG image. The HEX codes are clickable links that redirect to a page where the HEX code color and RGB values are showcased. Supplementary information of colors values in many color spaces (HSL, HSV, CMYK called "process color", XYZ, Yxy, Hunter Lab, CIE-Lab, web-safe HEX color code) is displayed. Also the HEX color's dominant color content (*red*, *green* or *blue*) is described. More information about the HEX color's base numbers (binary, octal, decimal and hex) is provided in a table. The HEX color is recast to color schemes of shades and tints with eleven color patches each. Two bar charts show the relative percentage of the RGB and CMYK channels for the HEX color. The HEX color is recast into color schemes of triadic (3), analogous (3), monochromatic (7) and complementary color (2) with the number of colors for the color scheme given in brackets. The HEX color can be previewed against a *black* and *white* background. There is information about the CSS embeddings of the HEX color and the HEX codes of related colors.
Evaluation: The information given in the redirected pages is exhaustive, but at the same time, the search results could have more information on the first view, for example, about the HEX codes of each color in the color scheme and the download functionality. A feature for easier handling of the color space values such as the direct copying onto the click board could help the user when working with color schemes. The color schemes in the search result are not sortable.

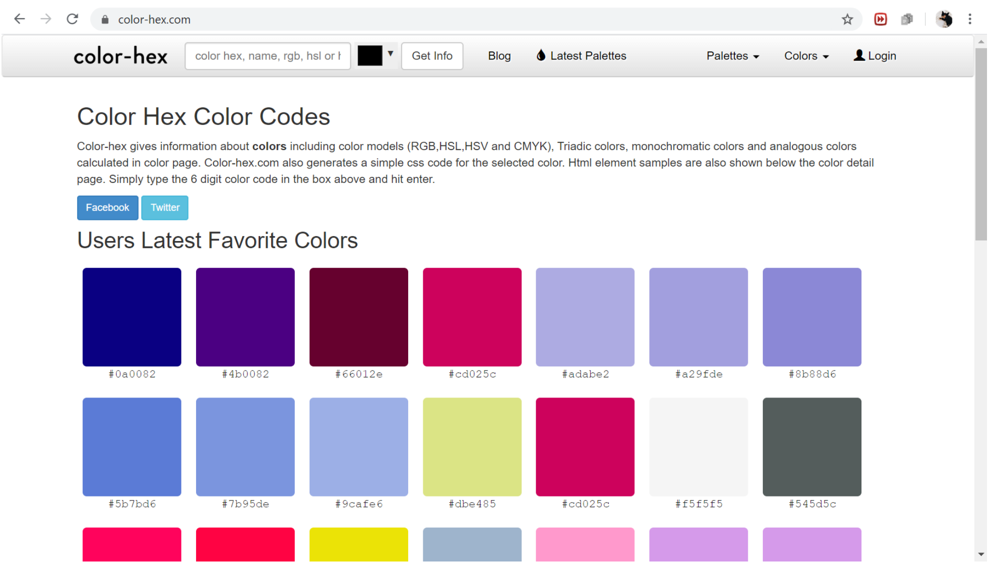


Figure C.6.: Color Hex' web page

7. Adobe Color CC Explore

Description: The search bar prompts the user to enter a keyword-based on colors, atmospheres or real names such as the ocean, wine, moonlight, happy or water. If a search for `sunflower` is carried out, most of the images have sunflowers in them or a sunflower text line. Thus, the keywords do not correspond to the search result's color schemes, but rather to the objects in the images. First, the images are searched in the Adobe Stock of images, then the images' color schemes are rendered in the following step. The search is not performed on the color but the object. The camera icon in the search bar makes it possible to upload an image on click and search similar images with most of the same color patterns in common. The images in the search result have their color scheme of five color patches displayed at the bottom of the image. Sometimes the color scheme is returned without an image. When clicking on a color scheme, the HEX codes are shown for each color in the color scheme. Functionalities such as downloading or including the color schemes to a library require an *Adobe Creative Suite* account.

Evaluation: It confuses the user to see these different keyword prompts in the search bar because this series of prompted words do not follow a logical sequence. The user cannot infer from the prompts which keywords are searchable and which are not. However, usually, such a distinction is the purpose of making keyword prompts in the search bar. In addition, returning both images with color schemes and color schemes in the search result seems to be too eclectic. On hovering over the image, the color scheme overrides the whole image. Hovering, too, adds too much dynamism to the webpage.

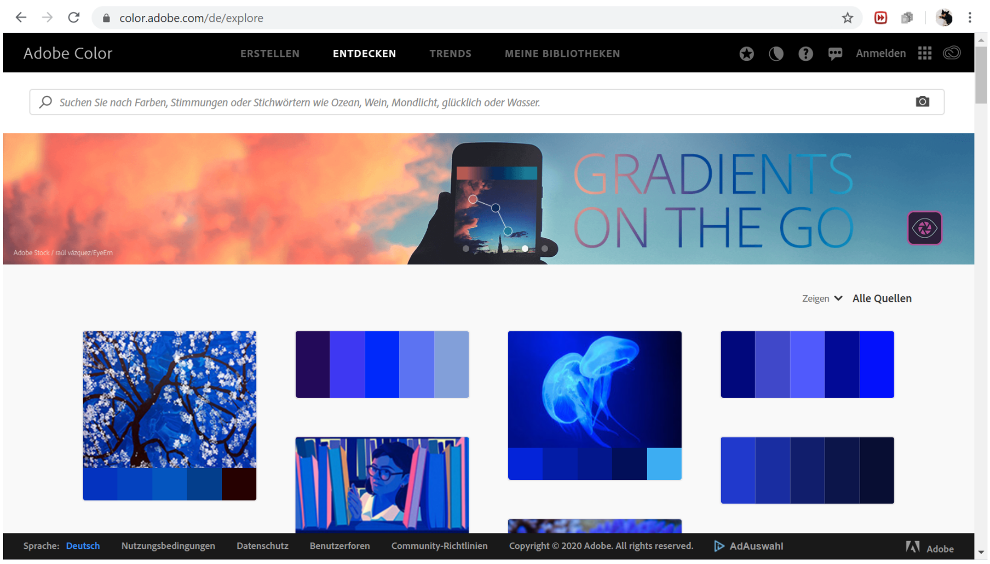


Figure C.7.: Adobe Color' web page

8. Colr

Description: The search bar takes words as keywords and matches them to the tags attached to each color scheme. The user can create their color scheme. The number of color patches in a color scheme is user-defined, as well as the tags assigned to the color scheme. If there is only one color patch, it is called color and it is returned in the search result on the left. If there are more than one color patches, it is called a color scheme and these are returned in the search result on the right. The color scheme can be edited, expanded; similar schemes can be searched as well as matching images. The color scheme is downloadable as an ACO file and there is a tag history for each color scheme. The colors of a color scheme are chosen using either a HEX code or a color picker. Matching paints and their color names are given in a separate window.

Evaluation: Tagging color schemes involves a lot of time and effort. Since tags are user-defined, any user can assign tags to colors and color schemes. If the matching is not done over a predefined list of color names and their corresponding colors, color name definitions will fluctuate over time and can be arbitrary at worst.

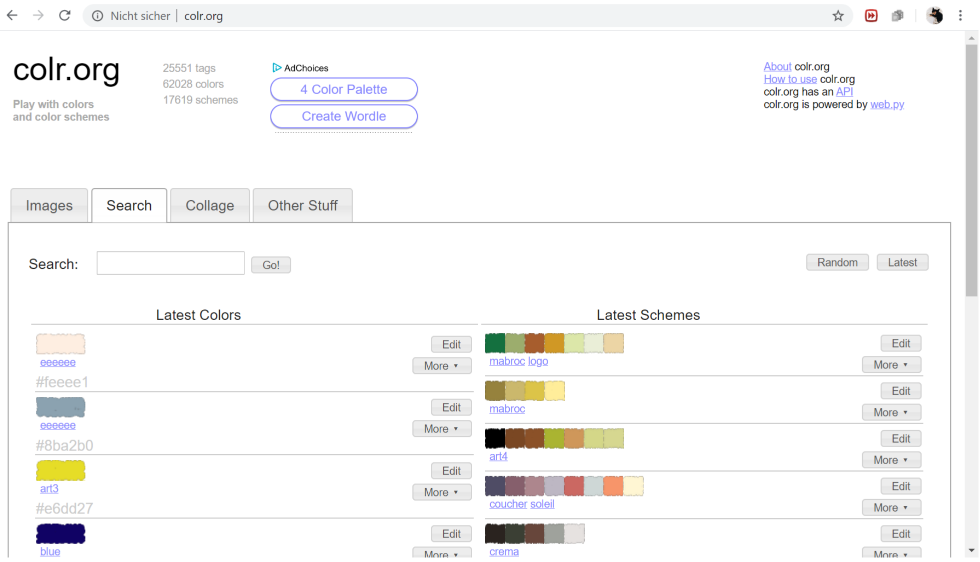


Figure C.8.: Colr' web page

9. Color Combos

Description: This search bar prompts the user to enter a color name or HEX code to find related color schemes of varying color patches called "color combo". There is a list of all color names possible in the "popular color tags" section. The search result includes all color schemes, their names <"ColorCombo"> and <digit>, four color patches, the number of views, the number of comments and the number of likes. The color name is matched to different HEX code colors. Each unique HEX code color is assigned multiple tags. For example, color code "#097054" is assigned the tags 097054, *blue green* and *watercourse*. The word tags can be assigned to different color codes as well. The administrator of the website creates the tag descriptions. When clicking on a color scheme, the HEX, RGB, CMYK and HSV color codes are displayed with the color image and the tags. There are functionalities for testing (editing) and copying the color scheme.

Evaluation: The tooltip could be improved to display only the HEX code color of the selected color in the color scheme instead of listing the HEX codes all at once. The color schemes rely on HEX codes mostly. It is not possible to choose from all possible color names from the menu all in one go.

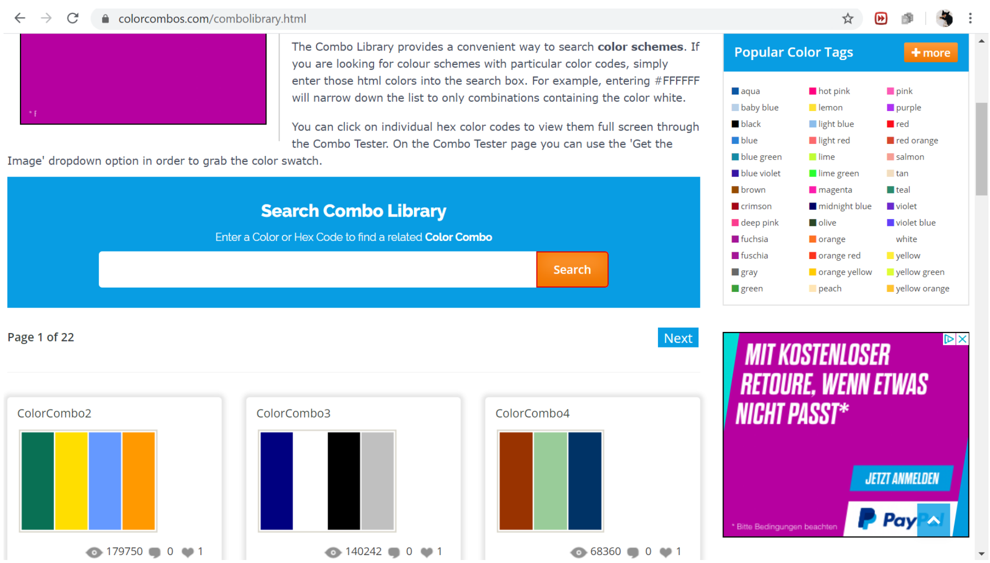


Figure C.9.: Color Combo's web page

10. Colorion

Description: Related Palettes can be searched for a given color. The colors can be chosen from a drop-down menu in the search bar. The color names are listed under color categories such as *pink*,*purple*, *red*, *orange*, *yellow*, *green*, *cyan*, *blue*, *brown*, *white* and *gray*. For example, for *pink*, six discrete color names, together with the color image, are displayed. The search result contains color schemes of varying lengths. Below the color schemes, different buttons exist for sharing, saving, liking, obtaining more information and downloading the color scheme. When hovering over a color in a color scheme, its HEX code displays and by clicking, the HEX code is copied to the clipboard. In the information section, the HEX code, the related color, the related HEX code and the related color name is given for each color. The relation binds all similar HEX code colors to a single color name category. There is a color representative for all similar HEX codes colors with the same color name. In addition, the RGB and HSL values are listed with the type (dark/light), and the colors obtained from darkening or lightening the original color. Finally, the original color's complementary color is shown. While the HEX code has an exact match, the color names are matched to multiple HEX codes.
Evaluation: The color schemes are not named. The color patches look like location icons. The RGB and HSL values are formatted in a way that is not easily transferable. There is a listed CND mapping color names to HEX codes and color images.

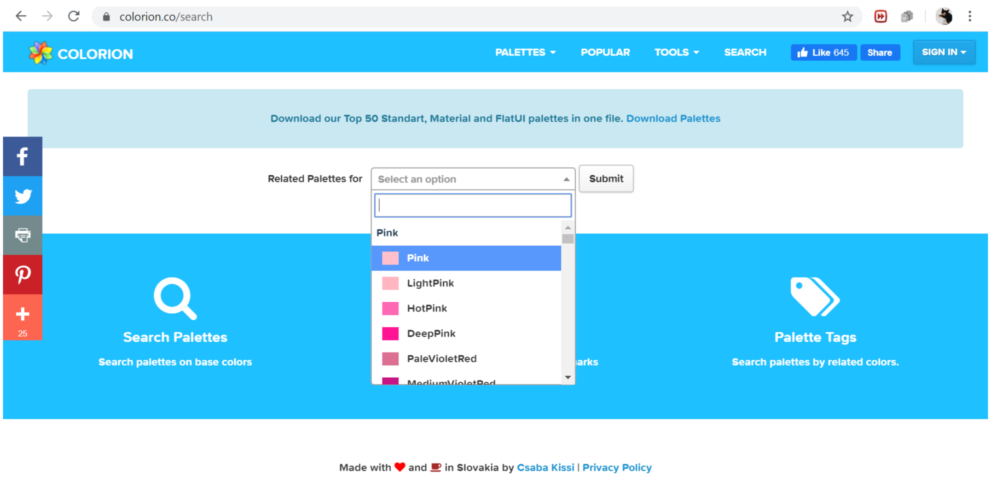


Figure C.10.: Colorion's web page

A Google search with the keywords `search color palettes` was performed. The following ten websites were paged and analyzed. A major finding was the reliance on HEX code as a color space value indicator for colors. An advantage of HEX code is that unlike RGB, HSV and LAB values, HEX codes can be written in one go without defining three different channel values. Also, as most homepages are designed for web developers and designers, the web-safe HEX code was used. Although free of charge, these professional websites still set themselves apart from academic projects such as the Color Survey or Color Thesaurus. The workings behind-the-scenes are not as readily documented or explained. Another insight is the number of color patches that the homepages use customarily: they typically use five color patches to compose a color palette. By collecting all the different features found on these homepages, evaluating the advantages and disadvantages becomes easier than before. Based on criteria of usability, user-friendliness and intuition, UI/UX design finally needs to meet the needs of the target group for a lasting impact. From ten different websites, *Colr* seems to come closest to what is being envisioned in this task.

The color palette search functionalities on the homepages can be split into a search query phase, a matching functionality phase and a search results phase. The search bar receives the search query as input, processes it by

meeting the condition and returns the desired results from the query.
When entering a keyword in the search bar, the search bar can try to auto-complete the query. It does so by listing prompts that have the same type as the keyword and for which a basic requirement should be that the prompts logically follow in the line of the keyword that is entered. Also, if not all prompts, i.e. the entire list of fine-grained color names can be reached with the scrollbar, the top-n most similar keywords concerning the first keywords should be displayed. The entire list of possible color names can be made accessible separately on a redirected webpage.

Each search query can be made with a single entry or with multiple `color` keyword entries. A color can be represented as a word, number or image.

Textual		Visual
Words	Numbers	
		Color image
	HEX code	
Color name		Color picker
Tag	RGB, HSV, HSL, CMYK, XYZ, Yxy, Hunter Lab, CIE-Lab	Predefined color palette
		User-defined color palette

Table C.1.: Web tool's search query

The color names can be extended to include words that are semantically strung further away from the names usually attributed to colors. They are either higher-level classifications of color names such as color attributes (dark, light, saturated, desaturated, gaudy), color schemes (analogic, mono, triad, complementary, and tetrad) and color contrasts (warm-cold, bright-dark), or a tangent to the field colors. These are, for example, textures (gold, neon, pastel, skin), themes (vintage, retro, wedding, Christmas, Halloween), climate (sunset, summer, autumn, winter, spring). Nouns on travel (Arabic, Paris, Alps) or adjectives that are reminiscent of feelings (frenzy, playful, bitter) can evoke a particular color or class of colors that are made searchable as a result.

Matching procedure The matching procedure can be simply HEX code to HEX code, but beyond a low-level matching, such as matching dictionary color to HEX code or basic color to HEX code, the matching needs to be specified more clearly. All dictionary colors are converted to a HEX code in a dictionary. The HEX triplet is predicated on the 24-bit RGB color scheme, where one byte represents a number in the range of 00 to FF and the three bytes forming the HEX code represent the *red*, *green* and *blue* color channels of the RGB color space. As in the RGB color space, the number of colors represented by HEX code is exactly $2 * *24 = 16'777'216$. A Color Name Dictionary has one dictionary color name assigned to one HEX code. There exist also match-ups where one color name can be matched to multiple HEX codes. The color palettes are returned that contain the searched HEX code color patch in them.

For the homepages, it is not known exactly how a color contained in the color palette but not in the dictionary is matched to a HEX code (or color name). It is also complicated to verify whether all possible HEX codes are grouped by forming as many groups as there are color names in the dictionary. This process would imply assigning a set of HEX codes to one color name such that all HEX codes are covered. It is also unknown whether some HEX codes are left out because of the large number of possible HEX codes. The machine learning classifier that was trained as part of this project will find the closest match, however, and assign a color name or color category to any color in RGB space. Otherwise, the discrepancy between the number of colors of the Color Name Dictionary and the possible HEX patch colors are stellar: for an average 720 possible dictionary colors, there exist $255 * *3 = 16'581'375$ possible patch colors. The uncharted territory would make up more than 99% of all colors from a random color palette. These are too many patch colors left unclassified and unsearchable, which could contribute to the rapid demise of the respective website's reason for initiation.

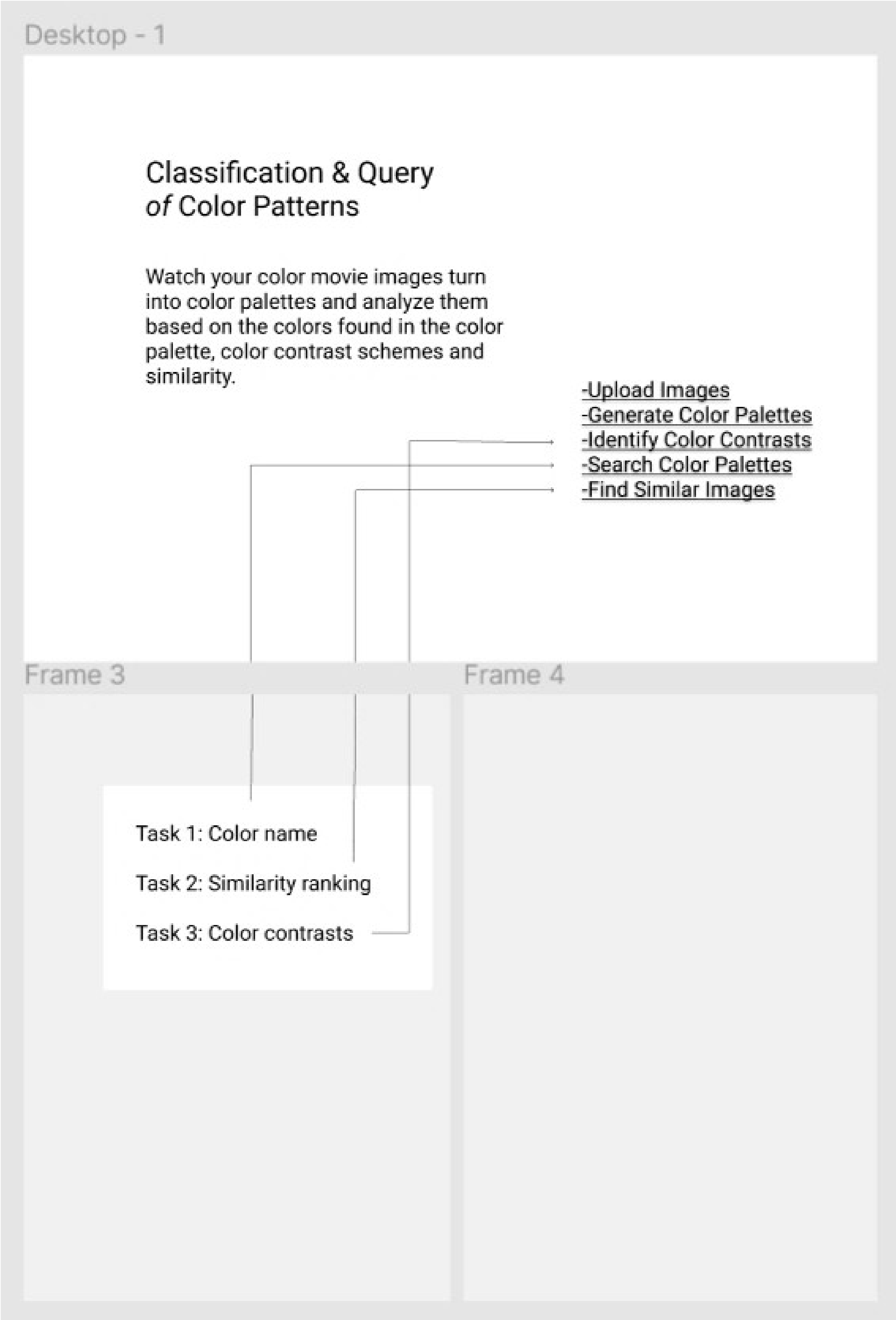
An alternative to color categories in text format is the use of tags. The user or system administrator tags all color palettes. The search is based on one or multiple tags because the color palettes are assigned one or multiple tags. Such a method is not recommended because of the squandering of resources, time and semantic integrity in assigning such tags to every color palette.

Display	Event
	copy-paste: patch color HEX code and all other color value spaces, palette
Searched keyword	*download*: palette PNG file, ACO file
Original image of color palette with color palette shown appended to the bottom of the image	*sorting of results*: by loves, name, date added/of publication, views, hues, comments, color patches, username; descending + ascending
Meta-information above the palette: Name of the displayed color palettes (<color name of most left to color name of most right>, descriptive names, <colorcombo> + $digit$), username	*filtering of results*: publication date *hovering*: patch displays color image, tags, HEX code, RGB, HSV, HSL, CMYK called "process color", XYZ, Yxy, Hunter Lab, CIE-Lab,
Color palette of x patches where x = 5 (default); equally-spaced vs. unequally-spaced color patches Info box in the patch: HEX code; Meta-information at the bottom of the palette: [number of likes, saves, shares, exports, comments, favorites, views, loves, elapsed period, tags, tag history]	*clicking*: HEX color's dominant color content (R, G or B), HEX base numbers (binary, octa, decimal and hex), shades and tints of 11 patches to the searched patch, relative percentage of RGB and CMYK channels for HEX code, patch color schemes: triadic, analogous, monochromatic and complementary color, preview on black and white, CSS embeddings, related HEX colors, their color names and patches, basic HEX code color category

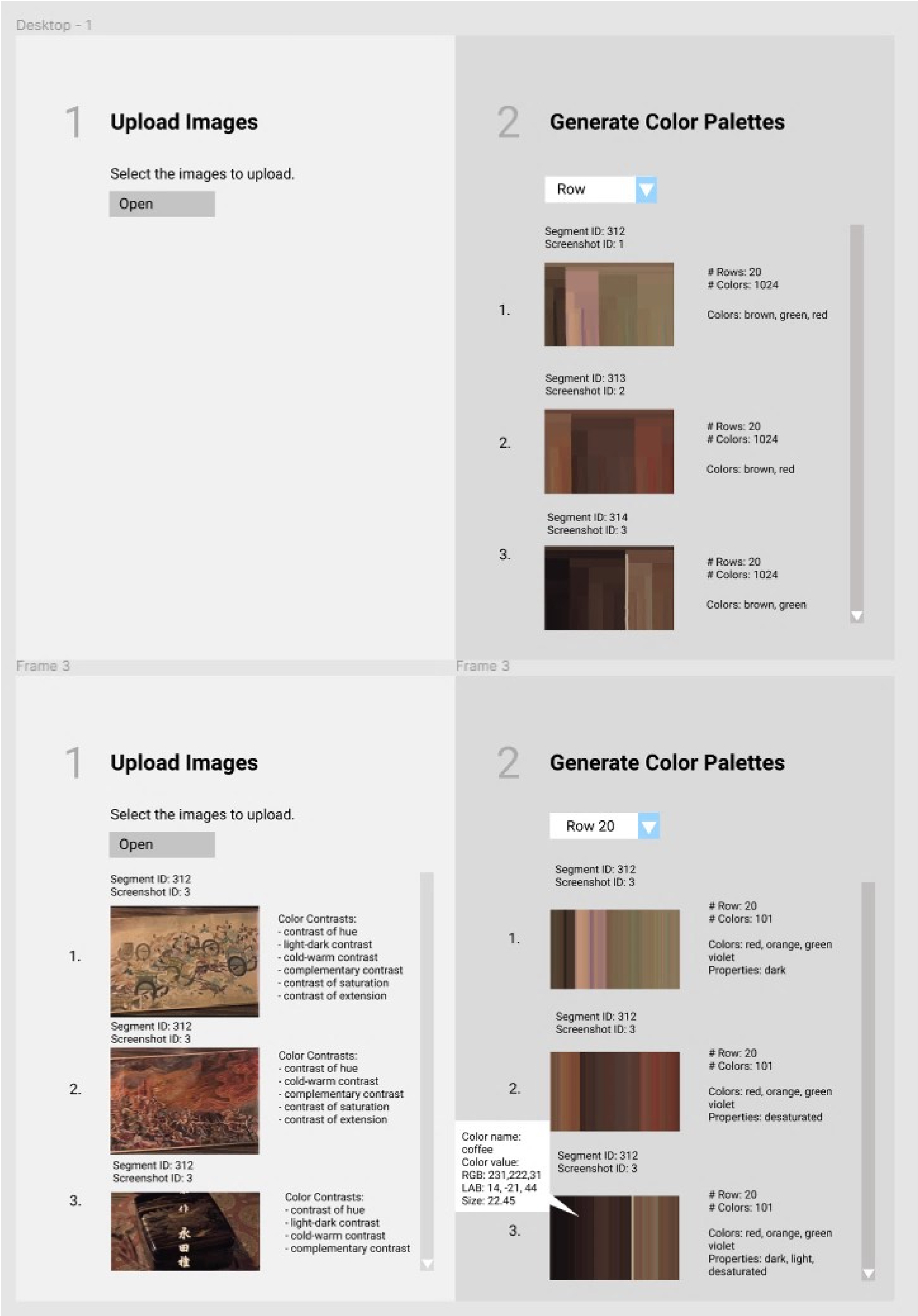
Table C.2.: Web tool's search results: what can be displayed in the search results are listed at the left; at the right the possible events are described independently of what can be displayed.

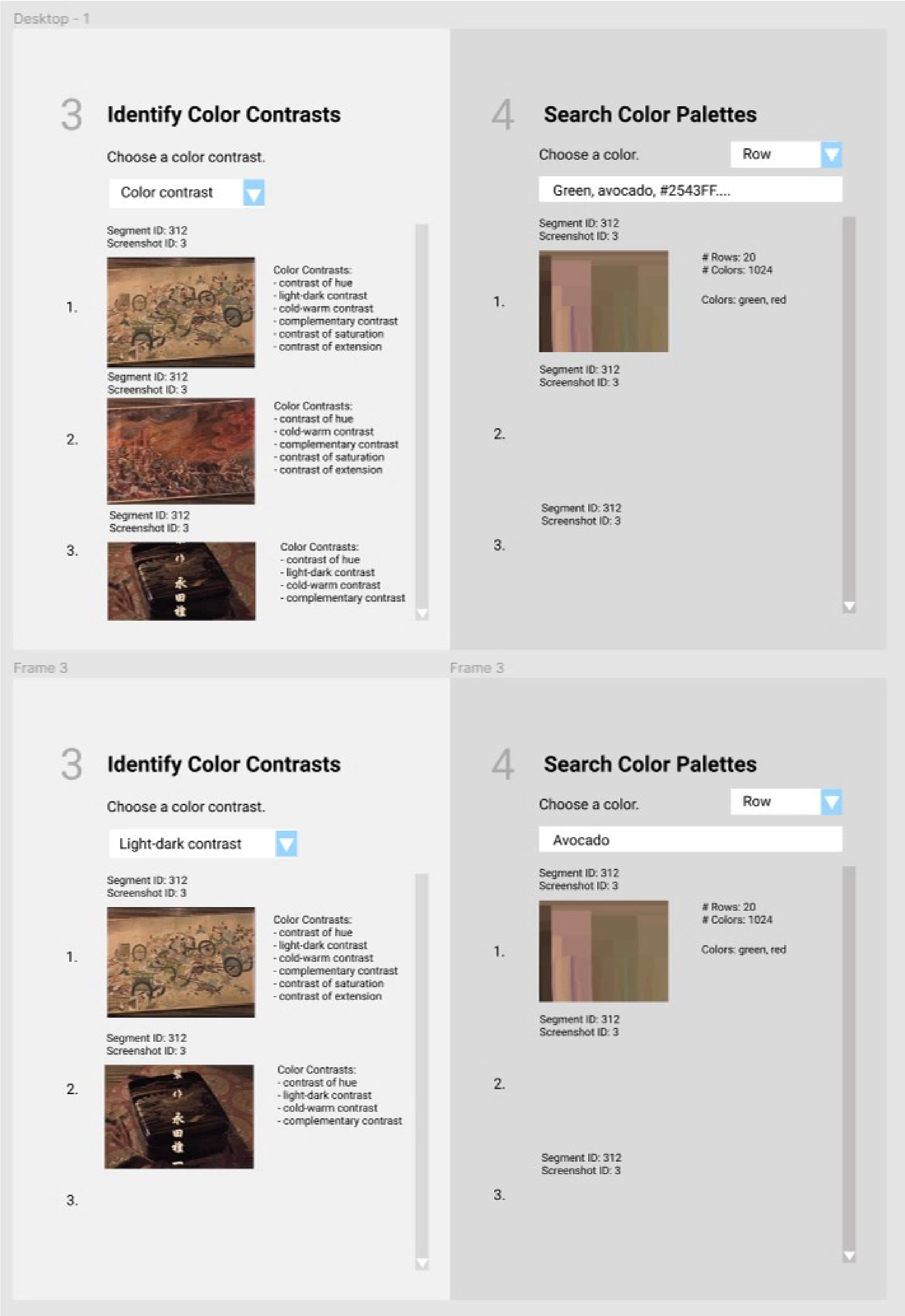
C.3. Web Tool Mockups

On Figma, designing the user interface for a web tool implementation of Task 1-3 resulted in four desktop displays. The mockups are extended with frames that display what happens after the user triggers an event on the website or describes the functionalities intrinsic to the desktop frame.


Desktop - 1
Classification & Query
of Color Patterns
Watch your color movie images turn into color palettes and analyze them based on the colors found in the color palette, color contrast schemes and similarity.
-Upload Images
-Generate Color Palettes
-Identify Color Contrasts
-Search Color Palettes
-Find Similar Images
Frame 3
Task 1: Color name
Task 2: Similarity ranking
Task 3: Color contrasts
Frame 4

C. Appendix III




Desktop - 1
3 Identify Color Contrasts
Choose a color contrast.
Color contrast
Segment ID: 312
Screenshot ID: 3
1.
Color Contrasts:
- contrast of hue
- light-dark contrast
- cold-warm contrast
- complementary contrast
- contrast of saturation
- contrast of extension
Segment ID: 312
Screenshot ID: 3
2.
Color Contrasts:
- contrast of hue
- cold-warm contrast
- complementary contrast
- contrast of saturation
- contrast of extension
Segment ID: 312
Screenshot ID: 3
3.
Color Contrasts:
- contrast of hue
- light-dark contrast
- cold-warm contrast
- complementary contrast
4 Search Color Palettes
Choose a color.
Row
Green, avocado, #2543FF....
Segment ID: 312
Screenshot ID: 3
1.
Rows: 20
Colors: 1024
Colors: green, red
Segment ID: 312
Screenshot ID: 3
2.
Segment ID: 312
Screenshot ID: 3
3.
Frame 3
3 Identify Color Contrasts
Choose a color contrast.
Light-dark contrast
Segment ID: 312
Screenshot ID: 3
1.
Color Contrasts:
- contrast of hue
- light-dark contrast
- cold-warm contrast
- complementary contrast
- contrast of saturation
- contrast of extension
Segment ID: 312
Screenshot ID: 3
2.
Color Contrasts:
- contrast of hue
- light-dark contrast
- cold-warm contrast
- complementary contrast
3.
Frame 3
4 Search Color Palettes
Choose a color.
Row
Avocado
Segment ID: 312
Screenshot ID: 3
1.
Rows: 20
Colors: 1024
Colors: green, red
Segment ID: 312
Screenshot ID: 3
2.
Segment ID: 312
Screenshot ID: 3
3.

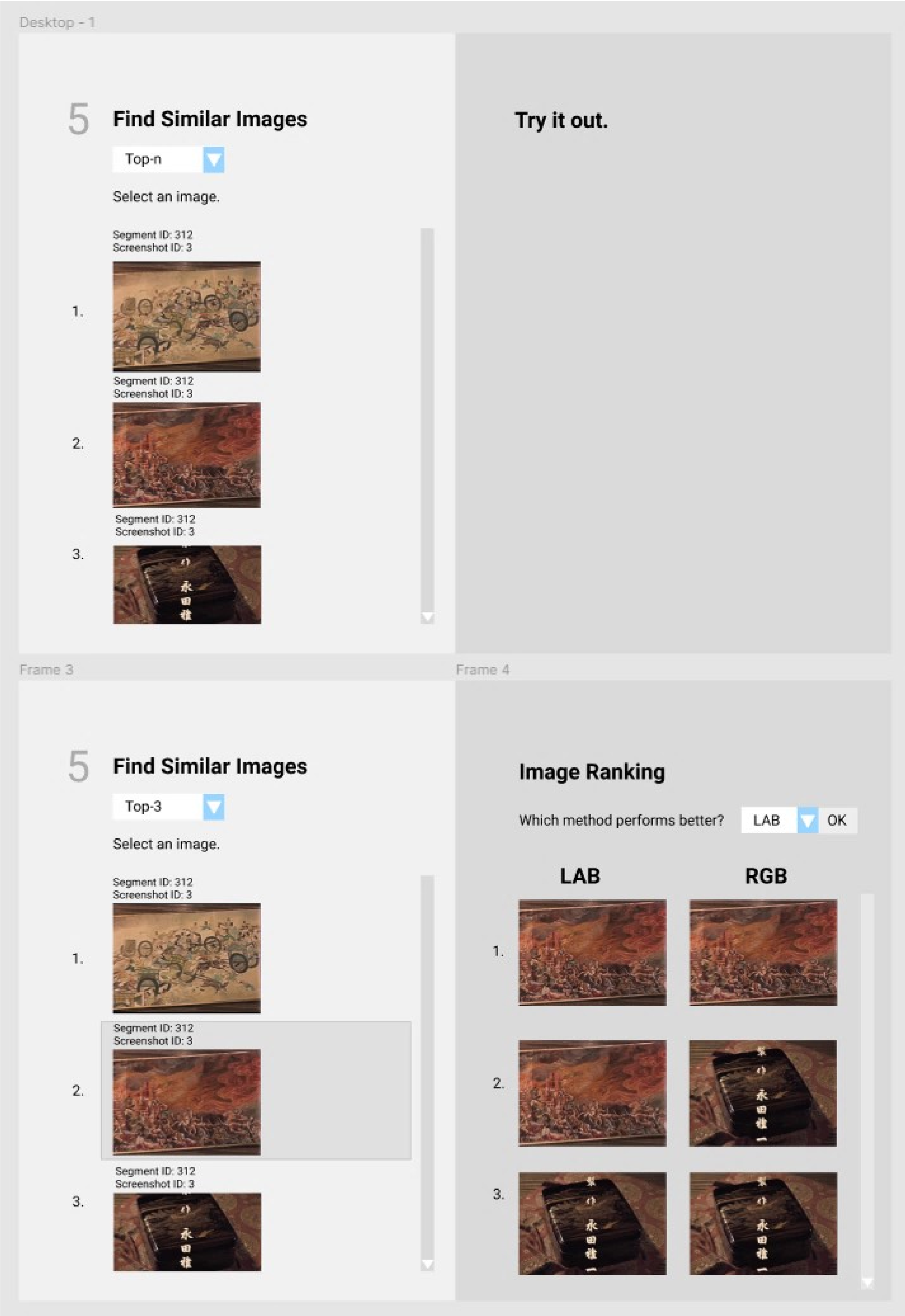


Figure C.11.: Mockups for the optional Task 4

C.4. Web-app Framework

For this task, a virtual environment is set up in the project's directory. If the packages are not installed globally on the local system but only installed in a virtual environment in the project's directory, it is easier for other people to work on the same project. This is because the requirements and packages used for deploying the project can be downloaded from the project's virtual environment called `env`.
There are two main open-source Python frameworks used for web development: **Flask** and **Django**. Both frameworks are considered stable in the Python web development community. Functions written in Python code can be mapped to visualization in the web app. While the vanilla micro-framework Flask is used for easy grasp and minimalistic web apps, Django is more tolerant of more Javascript code to help build a web application. Flask is suited for beginners that have no web experience. Databank provisioning can be available through a plugin. Django provides much more boilerplate code: this makes it more complicated to understand, but Django offers more such as user administration panels, user permission and authentication functionalities. Database connectivity can be established. Django is for users with Python and web development experience. Considering both frameworks' pros and cons, Flask was chosen to be the main framework of the web app because of its sleek and user-friendly property.

The web app is a proof of concept. A proof of concept need not be deployed on the web using Google App Engine, AWS or Heroku. Usually, deploying a web app on the internet needs more consideration of cryptographic libraries specific to web security such as authentication and authorization services when setting up a webpage, because web attacks are frequent. The web app is in no way subject to static or dynamic program analysis tools such as *Google Analytics* for tracking traffic on the web page.

Instead of color contrasts, future development could occur in defining color harmonies or color schemes to search in color palette and by extension in an image. For example, color scheme *gaudy* is closely linked to the color contrast of hue or split complementary colors could be identified as an extension to the complementary contrast developed as part of this project.
When defining a set of colors for the search query, color blindness could be a matter of concern in viewing the colors. On *Color-hexa*, there is a color blindness simulator that displays how the color is seen by different kinds of color-blind users.

C.5. Color Intepretations

The color usage in movies can be interpreted by their symbolic, expressional or impressionistic character - three concepts about color which are proposed by Johannes Itten. Developments to further annotate colors in video frames based on these interpretations are made easier by the results of this thesis. The VIAN tool can further incorporate an automated analysis of colors along with these categories. Do the colors used in the video frame have symbolic meaning? Are the colors used to express an emotion? Are the colors used to produce an aesthetic effect on the viewer? There are color dictionaries that map colors to symbolic representations and emotions. However, aesthetic considerations require an innate capacity to perceive beauty where theories about color harmony and color science cannot explain.
For example, *blue* can symbolize water, sky or the business realm. The emotions *blue* evoke are calmness, depth and sadness. *Blue* is used to give the impression of shadows or cold materials when used in fine-art painting.

Color	Symbolic	Expressional	Impressionistic
	Association	Emotion	Aesthetics
blue	water, sky, business	calm, deep, sad	Dark shadow, contrast to complementary orange, juxtaposed to analoguous purple (color harmony: color contrasts and harmonies)
Example	Blue is used in cleaning products to represent water as the prime resource used for cleansing.	Blue is used in a movie to evoke feelings of sadness.	Blue is the color of choice next to orange to bring about a pleasing effect to the viewer.

Table C.3.: Symbolic, expressional and impressionistic interpretations of color

List of Figures

List of Tables

Bibliography

[AW06] Josef Albers and Nicolas F. Weber. *Interaction of Color*. Yale University Press, 2006.

[BBK82] Toby Berk, Lee Brownston, and Arie E. Kaufman. A human factors study of color notation systems for computer graphics. *Commun. ACM*, 25(8):547–550, 1982.

[BK91] Brent Berlin and Paul Kay. *Basic Color Terms: Their Universality and Evolution*. University of California Press, 1991.

[BSD+13] Anurag Bhardwaj, Atish Das Sarma, Wei Di, Raffay Hamid, Robinson Piramuthu, and Neel Sundaresan. Palette power: enabling visual search through colors. In Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy, editors, *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 1321–1329. ACM, 2013.

[BVB08] Robert Benavente, Maria Vanrell, and Ramon Baldrich. Parametric fuzzy sets for automatic color naming. *Journal of the Optical Society of America. A, Optics, image science, and vision*, 25:2582–93, 11 2008.

[BW16] Jeffrey L. Binangkit and Dwi H. Widyantoro. Increasing accuracy of traffic light color detection and recognition using machine learning. In *2016 10th International Conference on Telecommunication Systems Services and Applications (TSSA)*, pages 1–5, 2016.

[CB99] Carlo Colombo and Alberto Del Bimbo. Color-induced image representation and retrieval. *Pattern Recognit.*, 32(10):1685–1695, 1999.

[CBP99] Carlo Colombo, Alberto Del Bimbo, and Pietro Pala. Semantics in visual information retrieval. *IEEE Multim.*, 6(3):38–53, 1999.

[CCL+01] Luigi Cinque, Gianluigi Ciocca, Stefano Levialdi, A. Pellicanò, and Raimondo Schettini. Color-based image retrieval using spatial-chromatic histograms. *Image Vis. Comput.*, 19(13):979–986, 2001.

[COSG+06] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. Color harmonization. *ACM Trans. Graph.*, 25(3):624–630, July 2006.

[CSMS17] Jesus Chamorro-Martínez, José Manuel Soto-Hidalgo, Pedro Manuel Martínez-Jiménez, and Daniel Sánchez. Fuzzy color spaces: A conceptual approach to color vision. *IEEE Transactions on Fuzzy Systems*, 25(5):1264–1280, 2017.

[CTO97] Tat-Seng Chua, Kian-Lee Tan, and Beng Chin Ooi. Fast signature-based color-spatial image retrieval. In *Proceedings of the International Conference on Multimedia Computing and Systems, ICMCS 1997, Ottawa, Ontario, Canada, June 3-6, 1997*, pages 362–369. IEEE Computer Society, 1997.

[CWT08] Jingyu Cui, Fang Wen, and Xiaoou Tang. Intentsearch: interactive on-line image search re-ranking. In Abdulmotaleb El-Saddik, Son Vuong, Carsten Griwodz, Alberto Del Bimbo, K. Selçuk Candan, and Alejandro Jaimes, editors, *Proceedings of the 16th International Conference on Multimedia 2008, Vancouver, British Columbia, Canada, October 26-31, 2008*, pages 997–998. ACM, 2008.

[DJLW08] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Ze Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60, 2008.

[DKN04] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval: A quantitative comparison. In Carl Edward Rasmussen, Heinrich H. Bülthoff, Bernhard Schölkopf, and Martin A. Giese, editors, *Pattern Recognition, 26th DAGM Symposium, August 30 - September 1, 2004, Tübingen, Germany, Proceedings*, volume 3175 of *Lecture Notes in Computer Science*, pages 228–236. Springer, 2004.

[Fai97] Mark D. Fairchild. *Color appearance models*. Addison-Wesley-Longman, 1997.

[FH18] Barbara Flueckiger and Gaudenz Halter. Building a crowdsourcing platform for the analysis of film colors. *Moving Image*, 18:80+, 2020/8/19/ 2018. 1.

[FHW19] Barbara Flückiger, Eva Hielscher, and Nadine Wietlisbach. *Color Mania – Materialität Farbe in Fotografie und Film*. Lars Publishers, 2019.

[FSA+95] Myron Flickner, Harpreet S. Sawhney, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by image and video content: The QBIC system. *Computer*, 28(9):23–32, 1995.

[GJ96] Georgy L. Gimel'farb and Anil K. Jain. On retrieving textured images from an image database. *Pattern Recognit.*, 29(9):1461–1483, 1996.

[G17] Frederik A. Gierlinger and Štefan Riegelnik. *Wittgenstein on Colour*. De Gruyter, Berlin, Boston, 2017.

[HBRFP19] Gaudenz Halter, Rafael Ballester-Ripoll, Barbara Flueckiger, and Renato Pajarola. VIAN: A Visual Annotation Tool for Film Analysis. *Computer Graphics Forum*, 2019.

[HM15] Mohammad Hossin and Sulaiman M.N. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process*, 5:01–11, 03 2015.

[IB70] Johannes Itten and Faber Birren. *The Elements of Color*. A VNR book. Wiley, 1970.

[Itt74] Johannes Itten. *The Art of Color: The Subjective Experience and Objective Rationale of Color*. A VNR book. Wiley, 1974.

[JFS95] Charles E. Jacobs, Adam Finkelstein, and David Salesin. Fast multiresolution image querying. In Susan G. Mair and Robert Cook, editors, *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1995, Los Angeles, CA, USA, August 6-11, 1995*, pages 277–286. ACM, 1995.

[LLBS12] Albrecht J. Lindner, Bryan Zhi Li, Nicolas Bonnier, and Sabine Süsstrunk. A large-scale multilingual color thesaurus. In *20th Color and Imaging Conference, CIC 2012, Los Angeles, California, USA, November 12-16, 2012*, pages 30–35. IS&T - The Society for Imaging Science and Technology, 2012.

[LLYZ12] Yuanliu Liu, Yudong Liang, Zejian Yuan, and Nanning Zheng. Learning to describe color composition of visual objects. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012*, pages 3337–3340. IEEE Computer Society, 2012.

[MH10] Jana Machajdik and Allan Hanbury. Affective image classification using features inspired by psychology and art theory. In Alberto Del Bimbo, Shih-Fu Chang, and Arnold W. M. Smeulders, editors, *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*, pages 83–92. ACM, 2010.

[ML07] Luo Ming and Changjun Li. *CIE Color Appearance Models and Associated Color Spaces*, pages 261 – 294. 03 2007.

[Moj05] Aleksandra Mojsilovic. A computational model for color naming and describing color composition of images. *IEEE Trans. Image Process.*, 14(5):690–699, 2005.

[Mun05] Albert H. Munsell. *A Color Notation*. G. H. Ellis Company, 1905.

[NC15] Antal Nemcsics and Jose Caivano. *Color Order Systems*, pages 1–16. 01 2015.

[New] Isaac Newton. *New Theory about Light and Colour*. Library of Alexandria.

[New52] Isaac Newton. *Opticks, Or, A Treatise of the Reflections, Refractions, Inflections & Colours of Light*. Dover Books on Physics. Dover Publications, 1952.

[RBHP11] Jesse Read, Albert Bifet, Geoff Holmes, and Bernhard Pfahringer. Streaming multi-label classification. In Tom Diethe, José L. Balcázar, John Shawe-Taylor, and Cristina Tîrnauca, editors, *Proceedings of the Second Workshop on Applications of Pattern Analysis, WAPA 2011, Castro Urdiales, Spain, October 19-21, 2011*, volume 17 of *JMLR Proceedings*, pages 19–25. JMLR.org, 2011.

[Sar16] Saman Sarraf. Hair color classification in face recognition using machine learning algorithms. *American Scientific Research Journal for Engineering, Technology, and Sciences*, 26:317–334, 2016.

[SB91] Michael J. Swain and Dana H. Ballard. Color indexing. *Int. J. Comput. Vis.*, 7(1):11–32, 1991.

[SC95] John R. Smith and Shih-Fu Chang. Single color extraction and image query. In *Proceedings 1995 International Conference on Image Processing, Washington, DC, USA, October 23-26, 1995*, pages 528–531. IEEE Computer Society, 1995.

[SF10] Boris Schauerte and Gernot A. Fink. Web-based learning of naturalized color models for human-machine interaction. In Jian Zhang, Chunhua Shen, and Glenn Geers, editors, *International Conference on Digital Image Computing: Techniques and Applications, DICTA 2010, Sydney, Australia, 1-3 December, 2010*, pages 498–503. IEEE Computer Society, 2010.

[SKK12] Hyojoo Son, Changmin Kim, and Changwan Kim. Automated color model–based concrete detection in construction-site images by using machine learning algorithms. *Journal of Computing in Civil Engineering*, 26(3):421–433, 2012.

[SS12] Boris Schauerte and Rainer Stiefelhagen. Learning robust color name models from web images. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012*, pages 3598–3601. IEEE Computer Society, 2012.

[Tom92] Shoji Tominaga. Color classification of natural color images. *Color Research & Application*, 17(4):230–239, 1992.

[VdBBR+12] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. volume 111, 10 2012.

[vdBSK08] Egon L. van den Broek, Theo E. Schouten, and Peter M. F. Kisters. Modeling human color categorization. *Pattern Recognit. Lett.*, 29(8):1136–1144, 2008.

[vdWSV07] Joost van de Weijer, Cordelia Schmid, and Jakob J. Verbeek. Learning color names from real-world images. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE Computer Society, 2007.

[vG10] Johannes Wolfgang von Goethe. *Farbenlehre*. Tubingen : J.G. Cotta'schen Buchhandlung, 1810.

[WLW+15] Yuhang Wang, Jing Liu, Jinqiao Wang, Yong Li, and Hanqing Lu. Color names learning using convolutional neural networks. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 217–221, 2015.

[ZLY+10] Lili Zheng, Xiaoqiang Li, Xiping Yan, Fufeng Li, Xiaoyan Zheng, and Wei Li. Lip color classification based on support vector machine and histogram. In *2010 3rd International Congress on Image and Signal Processing*, volume 4, pages 1883–1886, 2010.