Informatics

Human-centric best image selection on photo series

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Michael Pointner, BSc BEd
Matrikelnummer 01427791

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Mitwirkung: Univ.Ass. Dipl.-Ing. Dr.techn. Sebastian Zambanini

Wien, 7. Mai 2020

Michael Pointner

Robert Sablatnig

Informatics

Human-centric best image selection on photo series

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Michael Pointner, BSc BEd
Registration Number 01427791

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Assistance: Univ.Ass. Dipl.-Ing. Dr.techn. Sebastian Zambanini

Vienna, 7th May, 2020

Michael Pointner   Robert Sablatnig

Erklärung zur Verfassung der Arbeit

Michael Pointner, BSc BEd

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 7. Mai 2020

Michael Pointner

Danksagung

Ich möchte mich an dieser Stelle bei meinem Betreuer Sebastian Zambanini, welcher immer kompetent und zeitnah meine Fragen beantwortet hat und bei Problemen stets mit Rat zur Seite stand, bedanken. Weiters möchte ich mich bei Adam Finkelstein (einer der Autoren von Chang et al. [CYW+16]) für das Wiederherstellen ihres Benchmarks bedanken. Mein besonderer Dank gilt Eva Bernroitner, Nathalie Pleidl, Herbert Pointner und Simon Reisinger für das vollständige Bewerten des Gesichtsbilder-Datensatzes. Abschließend möchte ich mich bei meiner Partnerin, Nathalie Pleidl, fürs Korrekturlesen sowie bei meiner Familie und ihr für die Unterstützung während des gesamten Studiums bedanken.

Acknowledgements

I like to thank my advisor Sebastian Zambanini for his competent and fast responses to all my questions and his advice in phases of missing advance in solving a problem. Further, I would also like to thank Adam Finkelstein (one of the authors of Chang et al. [CYW+16]) for restoring their benchmark. My special thanks go to Eva Bernroitner, Nathalie Pleidl, Herbert Pointner, and Simon Reisinger for the complete labeling of the face images dataset. Finally, I would like to thank my girlfriend Nathalie Pleidl for proofreading and my family and her for the support throughout the whole study.

Kurzfassung

Immer wenn mehrere Bilder einer Szene oder eines Moments aufgenommen werden, kann ein notwendiger manueller Nachbearbeitungsschritt darin bestehen, das „beste Bild“ aus jeder dieser Serien nahezu redundanter Bilder auszuwählen, um sie Verwandten/Freunden zu zeigen oder in sozialen Medien zu veröffentlichen. Darüber hinaus hat die Größe der aufgenommenen persönlichen Fotosammlungen aufgrund der Allgegenwart von Digitalkameras zugenommen. Infolgedessen ist das Erstellen einer Zusammenfassung der Fotosammlung durch das Auswählen des „besten“ Bildes jeder einzelnen Serie, aufgrund der großen Anzahl von Fotos, eine zeitaufwändige und eintönige Aufgabe. Es wurden umfangreiche Untersuchungen zur Automatisierung dieser Aufgabe durchgeführt, jedoch aufgrund der Subjektivität, Kontextabhängigkeit und Zielabhängigkeit dieser Aufgabe wurde noch kein allgemein anwendbarer Algorithmus/Neuronales Netzwerk gefunden. Frühere Arbeiten haben ergeben, dass Personen bei der Bewertung von Fotos verschiedener Szenenkategorien sehr unterschiedliche Merkmale betrachten, z. B. bei Fotos, die Personen enthalten, deren Erscheinungsbild die Bewertungsergebnisse stark dominiert. Um dieser Erkenntnis nachzugehen, eruiert diese Arbeit die Verwendbarkeit menschlicher Gesichter zur Vorhersage der menschlichen Präferenz für Paare ähnlicher Bilder, die im selben Moment der Szene aufgenommen wurden, und vergleicht sie mit einer Methode, bei dem die menschliche Präferenz auf Grundlage der ganzen Bilder prognostiziert wird. Durch Extrahieren von Gesichtern und Bewerten der Paarpräferenz zwischen Gesichtern erreicht der in dieser Arbeit verwendete Ansatz nur Anhand der Gesichter eine ähnliche Vorhersagegenauigkeit für Bilder mit Menschen wie die Vergleichsmethode. In Kombination mit der Vergleichsmethode kann die Vorhersagegenauigkeit sogar erhöht werden. Dies bestätigt die hohe Bedeutung menschlicher Gesichter für die Vorhersage von Bildpräferenzen, da Gesichtsmerkmerkmale allein eine Vorhersage-Trefferquote am Stand der Wissenschaft erzielen konnten.

Abstract

Whenever multiple images of a scene or moment are taken, a necessary manual post-processing step might be to select the "best image" out of each of those series of nearly redundant images to show to relatives/friends or post on social media. Furthermore, the sizes of personal photo collections have increased due to the ubiquity of digital cameras. As a consequence, summarizing a collection through selection of the "keepers" of each series is a time-consuming and monotonous task due to the enormous amount of images. There has been heavy research on how to automate this task. However, no generally applicable solution has been found due to the subjectiveness, context-dependency, and objective-dependency of image selection. Previous work has concluded that people look at a very different set of features if evaluating photos of different scene categories. For example, if photos that contain people, their appearance dominates the evaluation results. In order to address this finding, this thesis evaluates the usability of human faces for predicting human preference on pairs of similar images taken of the same moment of scene and compares it to the baseline approach, which evaluates whole images. By extracting faces and just evaluating pair-preference between faces, which are a subset of the features used by the baseline, the approach used in this thesis achieves a similar test set performance on images containing humans as the baseline approach. By combination with the baseline, the performance could even be improved. This confirms the high importance of human faces in predicting image preferences on images containing humans as face features alone achieved state-of-the-art performance.

Contents

CHAPTER 1

Introduction

The ubiquity of digital cameras has led to an increased size of personal photo collections, as the previous work of Chang et al. [CYW$^+$16] has stated. Whenever multiple images of scenes or moments are taken, a necessary manual post-processing step might be to select the best one out of each of those series of nearly redundant images to show to relatives/friends or post on social media (see Figure 1.1).

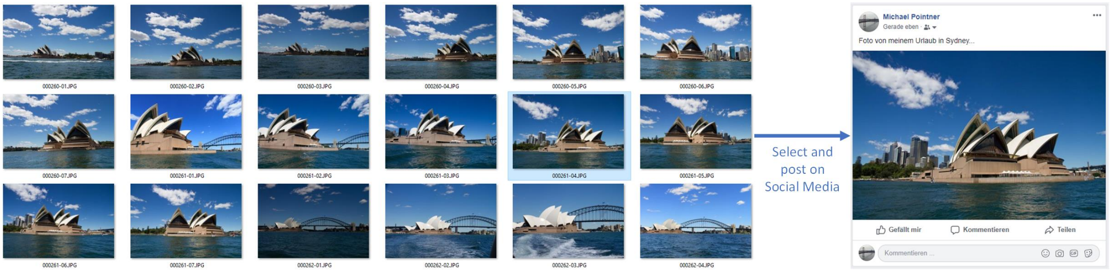


Figure 1.1: Manual selection of the "best" image from a series of nearly redundant images to share on social media.

The process of finding similar images and choosing the "best ones", "favorites", or "keepers" is a cumbersome and time-consuming task [CYW$^+$16]. The grouping of similar images is solved by Chang et al. [CYW$^+$16] through SIFT [Low04] feature matching and checking the similarity of the color histograms. Nonetheless, human preference on selecting images is still an insufficient researched problem. Depending on the selection task and individual preference, humans tend to select different images of series/collection. Therefore, methods in this research differ depending on the selection task and if they predict individual or a group's average preference. Due to the different selection tasks, approaches typically collect their own dataset to prove the effectiveness of their method, especially for series-level selection until now no widely agreed benchmark exists like for image classification [RDS$^+$15] or stereo matching [SS02]. For collection-level selection, the AVA dataset

[MMP12] is a dataset of approx. 255,000 images with labels, but for a benchmark, it is missing a prediction error metric. The recent approach of Chang et al. [CYW$^+$16] tries to establish a benchmark for series-level selection, which is used in this thesis.

1.1 Methodological Background

The focus of this work is series-level best image selection on photos taken of the same scene or moment in individual shots and not in burst mode, which makes a difference in the degree of similarity of the images. To the thesis authors' best knowledge, the approach of Chang et al. [CYW$^+$16] can be considered the best performing for non-burst series-level best image selection, and their dataset and benchmark is best suited for researching this task. For those reasons, their work is used as the baseline. Different to most other approaches, Chang et al. [CYW$^+$16] compute pairwise-preference predictions between two images of a series rather than absolute scores for each image of a series and the winner of a pairwise preference prediction gets compared to the next image in a series until the winner for a series is found. This procedure breaks the task of finding the best image of a series down to a binary preference prediction between two images. The pairwise-preference prediction performance of Chang et al. [CYW$^+$16] is 73% (stated in the paper; a performance of 71.8% could be reproduced, see Section 4.1) and 70.8% on images containing humans, which this thesis aims to improve.

Chang et al. [CYW$^+$16] discovered with their crowd-scored labeling procedure, where they asked participants to name the reason why an image is preferred or rejected, that people look at a very different set of features when evaluating photos of different scene categories. As an example, they mention that in photos containing people, their appearance dominates the evaluation results. This discovery was taken as a basis for the research in this thesis. The idea was to detect the persons in the images and improve the results based on features extracted from the detected persons. One early idea was to extract the pose of a person (e.g., using OpenPose by Cao et al. [CSWS17]), unfortunately, for most people in the dataset of Chang et al., no more than the face and the upper body is visible in the images. While crowd-scored rating their dataset, Chang et al. [CYW$^+$16] also asked participants to name a reason for preferring an image of a pair. The described preference reasons referring persons are mainly attributes of the face ("subjects looking", "subjects faces", "subjects facing", "closer faces", "smiling looking", "subjects eyes"), just a few concerning the position ("closer subjects", "subjects centered"), presence of persons ("shows subjects") and none concerning the pose of persons (Duplicate reasons and reasons concerning just one gender were removed). Therefore, it was decided to limit the features extraction to faces. In summary, the methodical approach of this thesis is to detect and extract faces from the dataset of Chang et al. [CYW$^+$16], pair corresponding faces, label those face pairs for human preference by a group of volunteers, and train a convolutional neural network on those face pairs. The goal of this thesis is to improve the approach of Chang et al. [CYW$^+$16] through combination with a Face Model and get a stand-alone solution with state-of-the-art performance on images containing humans.

1.2 Contribution

There are three main contributions of this thesis:

- Construction of a face preference dataset by extracting the faces from the images of the Auto Triage dataset of Chang et al. [CYW+16] and crowd-sourced labeling of face pairs for human preference.
- Evaluating the suitability of human faces for predicting human preference between two images taken of the same moment/scene.
- Construction of a face preference prediction system consisting of face extraction, face matching, and a face model.
- Evaluating the face preference prediction system as a stand-alone approach and in combination with the baseline approach of Chang et al. [CYW+16].

1.3 Overview

At first, in Chapter 2, the Computer Vision concepts used in this thesis are explained to give the reader the required background knowledge. Next, an overview of the related work for image preference prediction, image summarization, and image selection is given (see Chapter 3). Furthermore, the baseline approach (Chang et al. [CYW+16]) to this thesis is presented (see Section 3.4). The approach of Chang et al. [CYW+16] plays an important role to this thesis since their dataset is used for evaluation, the faces used for training of the proposed method's face model were extracted from their dataset, and a part of the pipeline of the proposed method is based on the approach of Chang et al. [CYW+16]. Afterward, the methodology of the proposed approach (see Chapter 4), consisting of face extraction (see Section 4.2), face matching (see Section 4.3), and a face model (see Section 4.4), is described. The performed experiments are presented in Chapter 5. In this chapter, the suitability of human faces for predicting human preference (see Section 5.2), the performance of the face matching (see Section 5.3), and the results (see Section 5.4 and 5.5) are evaluated. Lastly, a conclusion (see Chapter 6) about the proposed method and its benefits and limitations is given.

CHAPTER 2

Background

Since the approach of this thesis, the baseline, and many other approaches discussed in this thesis use the technique of Convolutional Neural Networks, a short recap will be given to provide the reader with the required preknowledge.

2.1 Structure of a Neural Network

A Convolutional Neural Network (CNN) is a particular form of an Artificial Neural Network (ANN) that is optimized to work with images [ON15]. Artificial Neural Networks, also just called Neural Networks (NN), are computational processing systems that are inspired by how biological nervous systems (e.g., the human brain) operate. ANNs are composed of interconnected layers of computational nodes (called neurons), which work in a distributed fashion to collectively learn from the input in order to optimize its final output. An ANN consists of the input, usually in the form of a multidimensional vector, an output, and in between a number of hidden layers [ON15]. The number of hidden layers chained gives the depth of the model. The name "deep learning" and "deep neural network" arose from this terminology.

2.1.1 Hidden layers

The layers between the input and output are called hidden layers because, in contract to the output layer, the desired output of the neurons in the hidden layers is not known, i.e., not directly specified by the training data [GBC16].

The neurons in the hidden layer will make decisions form the previous layer [ON15]. To make those decisions, most neural networks use affine transformations controlled by learning parameters followed by a fixed nonlinear function called the activation function. Without a nonlinear activation function, the network as a whole would remain a linear function of its input.

2.1.2 Activation function

Activation functions are typically used on top of an affine transformation. The Rectified Linear Unit (ReLU) $g(z) = max\{0, z\}$, the logistic sigmoid $g(z) = \sigma(z) = \frac{1}{1+e^{-z}}$, and the hyperbolic tangent $g(z) = tanh(z)$ are example choices for the activation function. Hyperbolic tangent and logistic sigmoid are closely related: $tanh(z) = 2\sigma(2z) - 1$ [GBC16].

For the output layer, examples of activation function choices are linear $g(z) = z$, sigmoid, and softmax. The softmax function is defined as $softmax(\mathbf{z})_i = \frac{e_i^{\mathbf{z}}}{\sum_j e_j^{\mathbf{z}}}$, where the variables $z_1, z_2, \ldots, z_n$ represent the unnormalized log probabilities. The output is a probability distribution over a discrete variable with n possible values [GBC16].

2.2 Training

In order to measure the difference between the current output of the neural network and its desired output, a loss function (also called cost function or objective function) is used, e.g., mean squared error[GBC16].

There are two learning paradigms of learning an neural network: Supervised and unsupervised learning. While supervised learning is learning through pre-labeled inputs, which act as targets, unsupervised learning differs in that there are no labels involved, and the objective is to reduce or increase an associated cost function [ON15].

2.2.1 Optimizer

The largest difference between linear models and neural networks is that the nonlinearity of a neural network causes most loss functions to become nonconvex, as Goodfellow et al. [GBC16] state. This means that neural networks are usually trained by using iterative, gradient-based optimizers that only drive the cost function to a very low value, rather than the linear equation solvers used to train linear regression models. However, stochastic gradient descent applied to nonconvex loss functions has no convergence guarantee and depends on the values of the initial parameters [GBC16]. A back-propagation algorithm allows the information from the cost to flow backward through the network in order to compute the gradient to adapt the learning parameters [ON15].

2.2.2 Overfitting

Overfitting is when a network is unable to learn effectively due to a number of reasons. A neural network overfitted if it does not generalize well, i.e., it performs well on the training data but poorly on the validation and testing data [ON15].

There are different strategies to reduce the effect of overfitting, such as reducing the complexity of the model or using a regularization strategy such as early stopping, parameter norm penalties, dropouts, or reducing the learning rate on plateaus:

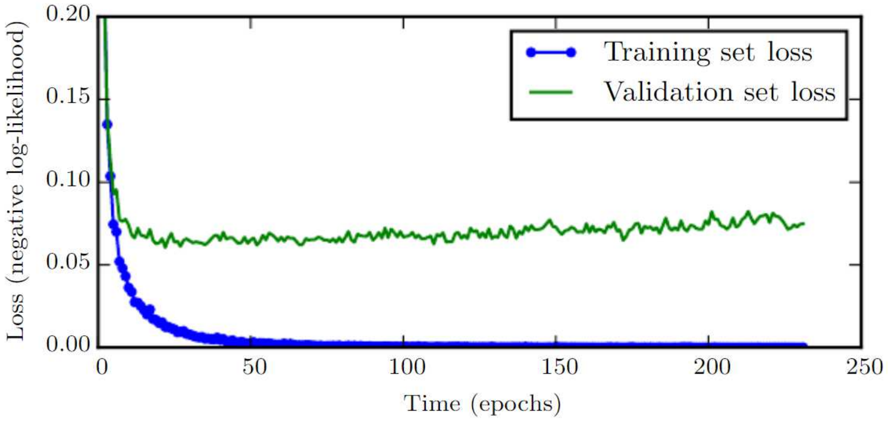


Figure 2.1: When training large models with sufficient representational capacity to overfit its task, it can be observed that the training loss decreases steadily over time, but the validation loss begins to rise again (Image taken from [GBC16]).

- When training large models with sufficient representational capacity to overfit its task, it can be observed that the training loss decreases steadily over time, but the validation loss begins to rise again (see Figure 2.1). A simple and effective strategy to prevent this behavior is stopping the training at the point where the validation loss begins to rise again, a strategy known as early stopping [GBC16].

- Models often benefit from reducing the learning rate once learning stagnates, a strategy called reduce learning rate on plateau[1].

- A different strategy to prevent overfitting is parameter norm penalties that limit the capacity of the model by adding a parameter norm penalty to the loss function. For example, the L^2 parameter norm penalty, commonly known as weight decay, drives the weights w of the model closer to the origin by adding the regularization term $\frac{1}{2}\|\mathbf{w}\|_2^2$ to the loss function.

- Dropout is randomly removing non-output neurons from the network by multiplying its output value by zero. The probability of a neuron to be removed is a hyperparameter fixed before the training begins. Which neurons are removed is decided by a randomly sampled binary mask that changes with each update of the model's learning parameters [GBC16].

2.3 Convolutional Neural Network

As mentioned at the beginning of this chapter, a Convolutional Neural Network (CNN) is a special type of Neural Network, that is optimized to work on image input. As the name

[1] https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/ReduceLROnPlateau

implies, convolutional layers play a vital role in CNNs [ON15]. One of the key differences is that the neurons in a convolutional layer of a CNN are organized into three dimensions, the two spatial dimensionalities of the input (height and width) and the depth, e.g., the color channels of an image. In a convolutional layer, the neurons are also called kernels as they are usually small in spatial dimensionality and only connect to a small region of the layer preceding it [ON15]. CNNs are comprised of three types: Convolutional layers, pooling layers, and fully-connected layers. In convolutions layers, the parameters focus on the use of learnable kernels. These kernels are usually small in spatial dimensionality but spread along the whole depth. A convolutional layer convolves each kernel across the spatial dimensionality of the input to produce a 2D activation map. The objective is that the network will learn kernels that "fire" when they see a specific feature at a given position of the input, commonly known as activations. The second type of layer in a Convolutional Neural Network, pooling layer, aims to gradually reduce the computational complexity of a model by reducing the dimensionality of the activation map. A polling layer operates over each activation map and performs downscaling along the spacial dimensionality, typically using the "MAX" function. The last typically used layer in a CNN and already known from ANNs, the fully-connected layer, contains neurons which are directly connected to the neurons in the two adjacent layers [ON15].

2.4 Perceptron

A perceptron is a particular type of neural network developed by Rosenblatt [Ros58] used for binary classification [PM92]. A multilayer perceptron (MLP) consists of multiple layers of simple, two-state, sigmoid processing elements (nodes) or neurons that interact using weighted connections [PM92].

2.5 Siamese Neural Network

A "Siamese" neural network consists of two identical sub-networks with shared weights joined at their outputs. The two sub-networks extract features from two different images and are trained together [BGL$^+$93]. In the original use case of Bromley et al. [BGL$^+$93], the joining neuron measures the distance between the two features vectors. For the use case in this thesis, the joining neuron is replaced by an element-wise feature subtraction [BGL$^+$93].

2.6 VGG16

The VGG16 is a deep convolutional neural network developed for image recognition developed by Simonyan and Zisserman [SZ15]. The input of the VGG16 is a fixed-size 224 × 244 RGB image. It consists of a stack of convolutional layers, followed by three fully-connected layers and a soft-max layer (see Figure 2.2). All hidden layers use ReLU as activation function. The dimensions of each layer are shown in Figure 2.2. Rather than using relatively large receptive fields in the first convolutional layers (e.g., 11 × 11 in the AlexNet CNN by Krizhevsky et al. [KSH12]), Simonyan and Zisserman [SZ15] use very small 3 × 3 receptive fields throughout the whole net. Simonyan and Zisserman [SZ15] show that be using, for instance, a stack of three 3 × 3 convolutional layers, a 7 × 7 effective receptive field can be obtained. They argue that three non-linear rectification layers instead of a single one make the network more discriminative and reduces the required amount of training parameters [SZ15]. The VGG16 by Simonyan and Zisserman [SZ15] is used in this thesis as sub-network in a Siamese architecture.

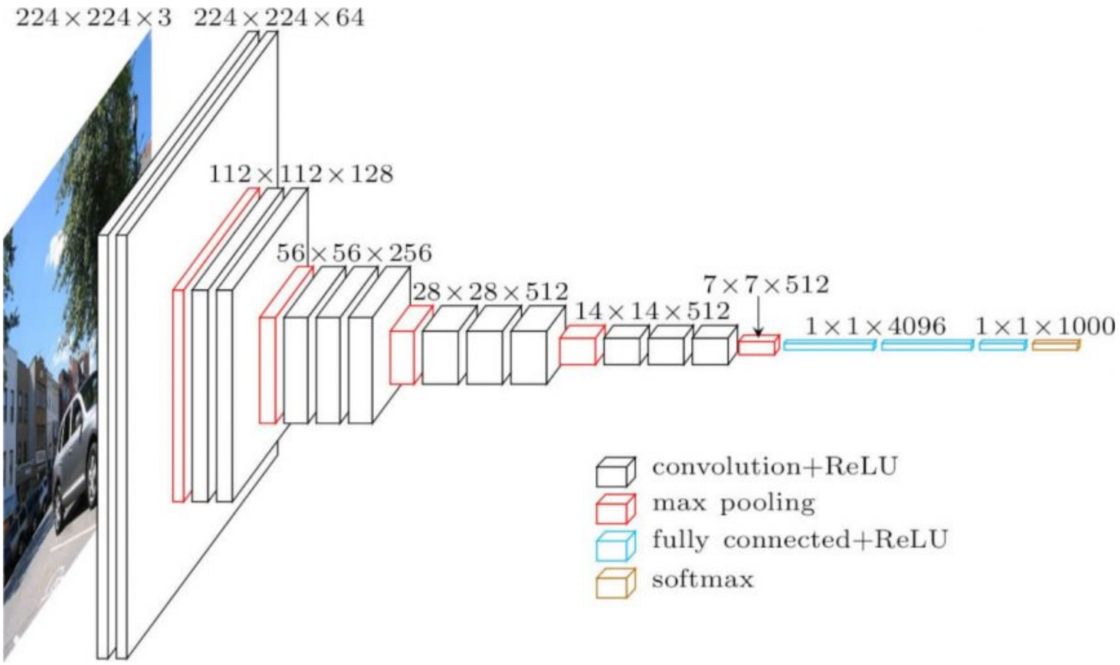


Figure 2.2: The architecture of VGG16 deep convolutional neural network[2]

In this chapter, the pre-knowledge required to understand the methodology of the proposed approach, and its related work was explained. At first, an introduction to neural networks and their training procedure was given. Afterward, the concept of a convolutional neural network (CNN), a particular type of neural network, was presented.

[2] Image source: `https://neurohive.io/en/popular-networks/vgg16` by Muneeb ul Hassan, usage permission granted

Lastly, the concepts of the Perceptron and a Siamese neural network were explained, and the VGG16 deep convolutional neural network was presented.

CHAPTER 3

Related work

Automatic selection, summarization, and quality assessment have become broadly researched topics in the last thirteen years, as will be shown in this chapter. Due to the subjectiveness of this problem, most existing methods only predict the mean opinion score collected on a dataset [TM18]. Previous work differ in their dataset collection procedure and the objective they pursue. Therefore, this literature overview is split based on the objective in approaches that aim to predict an image's preference, summarize an image collection, or select the best image of redundant images. Since those objectives are used differently in literature, a brief description is given, how those objectives are understood in this thesis:

- Under **image preference** is understood how well participants liked a single image without any comparison to other images. Image preference is different called in literature depending on what authors believe causes the preference, e.g. quality, memorability, popularity, interestingness, aesthetics, importance or specificity.
- **Image summarization** is the selection of a representative subset of a collection meant that gives viewers a good summary of the whole collection.
- **Image selection** aims to eliminate near-redundant photos by selecting the best images of a group of similar images.

Those categories can be viewed as different stages or sub-problems of a problem set, since most summarization approaches work on collections where the elimination of near-redundant photos by selecting a favorite was already done by the users that submitted the photos. Alternatively, those near-redundant images are treated as equal, and a diversity term manages that no two near-redundant images get selected for the summary. In contrast, selection approaches apply a preference prediction method in order to select the images with the highest preference from a group of similar images.

In the following sections, the methodology of a few representative papers is summarized to get an overview of what other authors have tried to solve the problem set of image prediction, automatic summarizing, and selecting images. Since there are hardly any similarities and constant jumps may confuse the reader, presenting one after the other, and describing similarities or differences to previously presented approaches was decided. As no common agreed dataset and accuracy metric exists for this problem set, and all have a slightly different goal, every described approach outperforms its baseline approach on its dataset.

3.1 Image Preference

In this section, approaches are presented that only have the image preference prediction as objective or use the prediction for other objectives that summarization or selection, since those will be treated in an own section. In this thesis, image preference is used as an umbrella term for different low- or high-level image attributes that might affect human image preference. As Wang et al. [WVSZ20] has stated, there has recently been an increasing interest in understanding and learning which image attributes reflect the human perception such as image quality, diversity and coverage ([SMJ11]), memorability ([IPTO11], [IXTO11], [GGR$^+$13], [KRTO15], [DPK$^+$15]), popularity ([KDH14]), interestingness ([FHX$^+$14], [IXTO11], [DOB11]), aesthetics ([DOB11], [LLJ$^+$14], [DJLW06]), importance ([BBD$^+$12]) and specificity ([JP15]).

The approaches that are further discussed in this section have in common that they are trained and evaluated on the AVA dataset [MMP12]. The AVA dataset is a large-scale database with a distribution of aesthetic scores for over 250, 000 images. The number of votes per image range from 78 to 549 with an average of 210 votes. Those votes were generated by hundreds of amateur and professional photographers that rated the images with integer scores from 1 (lowest quality) to 10 (highest quality).

As Jin et al. [JSS16] stated, the distribution of aesthetic scores in the AVA dataset is extremely unbalanced, which limits the prediction capability of previous methods [JSS16]. Jin et al. [JSS16] propose to use sample weights during the training of the CNN to overcome this bias. They train both a regression model to predict the aesthetic scores and a histogram prediction model to predict the normalized histogram of user ratings, both by adapting the VGG16 [SZ15] model.

Kong et al. [KSL$^+$16] proposed a method to aesthetically rank photos by training on their own "assembled aesthetics and attributes" (AADB) database as well as Murray's AVA [MMP12] dataset with a rank-based loss function. They trained an AlexNet-based Siamese CNN to learn the difference of the aesthetic scores from a pair of input images, which also indirectly optimizes the rank correlation. Differently to this thesis and its baseline approach of Chang et al. [CYW$^+$16] (see Section 3.4), which both also use an AlexNet-based Siamese CNN, Kong et al. [KSL$^+$16] train their network on absolute aesthetics score differences rather than binary preferences. They achieve 77.33% accuracy on the AVA dataset with their best model.

In contrast to just learning aesthetic scores, Talebi and Milanfar [TM18] introduced a CNN-based image assessment method, which predicts the distribution of human opinions scores, rather than just the mean scores. They argue this leads to a more accurate quality prediction with a higher correlation to the ground truth ratings.

Lu et al. [LLJ+14] use a double-column deep neural network to evaluate the aesthetic quality of a given image. In order to represent images' global cues and local cues, they generate two heterogeneous inputs: A global view of the image and a local view of the image constructed by random crop. Global and local views are trained in different columns of the CNN, and just the final fully-connected layer is jointly trained. Differently to Siamese architectures [BGL+93] (described later in Section 3.4), the CNN-columns do not share weights.

Zhu et al. [LLJ+14] collected a wide range of positive facial expressions that might serve as a portrait for each of their eleven subjects, including both female and male subjects ranging in age from 23 to 50. By showing the subjects a 12-minute collection of short videos that ranged across several emotional categories, they asked their subjects to make their best portrait expression in several posed categories, such as confident, big open-mouthed smile, etc. while filming them during the whole process. Since those recordings contain a highly redundant sampling of common expressions, they implemented a greedy algorithm to select unique expressions from those videos. In order to get labels for the collected unique expressions on the attractiveness and serious axis for each subject, they use Amazon Mechanical Turk to collect pairwise comparisons (e.g., "Is expression A more

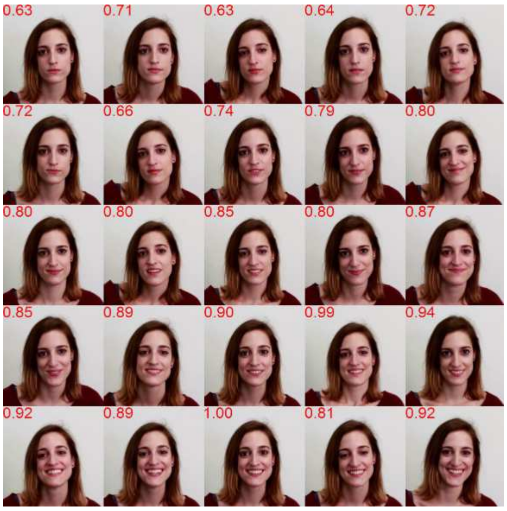


Figure 3.1: Example of the Mirror Mirror dataset by Zhu et al.: Visualization of the most attractive expression of one subject across a range of seriousness levels (seriousness decreases from top-left to bottom-right in reading order; attractiveness score are shown in the corner) [ZAE+14]. This dataset is important to this thesis, as it is used in Section 4.4.1 (Image taken from [ZAE+14]).

attractive than B"). To estimate attractiveness scores $A = \{a_1, \ldots, a_n\}$ and seriousness scores $S = \{s_1, \ldots, s_n\}$, they use the Bradley-Terry model [BT52], which models the probability of choosing expression I_i over I_j as a sigmoid function of the score difference between two expressions, i.e., $P(I_i > I_j) = f(a_i - a_j)$ where $f(u) = \frac{1}{1+exp(-u/\sigma)}$. By solving a maximum a-posteriori (MAP) problem, the scores can be estimated as described in Tsukida and Gupta [TG11]. In Figure 3.1 those estimated scores can be observed for 25 examples of one subject [ZAE$^+$14]. Their dataset is important to this thesis, as it is used in Section 4.4.1.

3.2 Image Summarization

Sinha et al. [SMJ11] propose a summary generation framework for personal photo collections that satisfies three salient properties: quality, diversity, and coverage. To grant diversity of their summary, they model the diversity as the minimum of pairwise distances between the summary photos. Coverage is calculated by the sum of the amount of photos that are represented by each individual photo in the summary. Interestingness is calculated by the presence or absence of appealing and quality attributes in a photo, like if the image is a portrait, group photo, panorama, or the images' color distribution, hue, absence of blur, and coarseness. The three properties quality, diversity, and coverage are combined linearly with equal weights, and the maximum optimization problem is solved in a greedy way. Their dataset was constructed by crawling photo sharing and storage sites of sixteen users. Sinha et al. compare the performance of their approach with k-means clustering and random selection, which it both outperforms [SMJ11].

Simon et al. [SSS07] define an ideal summary if it presents the most interesting and important aspects of the scene with minimal redundancy. Their objective is to automatically create a one page visual summary of a scene or city from photos downloaded from Internet sharing sites that captures the key sites of interest. Their approach examines the distribution of images in a collection to select a set of canonical views to form a summary. They define a canonical view as the image of a site of interest, that shares the most visual features with all other photos of the same site of interest. To achieve this, they use a greedy clustering technique based on SIFT feature co-occurrences to partition the image set into groups of related images and select the canonical view of each group [SSS07]. They define a number of possible criteria for choosing canonical views:

- **Likelihood**: An image should be included in the summary if it is similar to many other images.
- **Coverage**: An image should be included in the summary if it covers a large number of visual features in the scene.
- **Orthogonality**: Two images should not both be included if they are similar to each other.

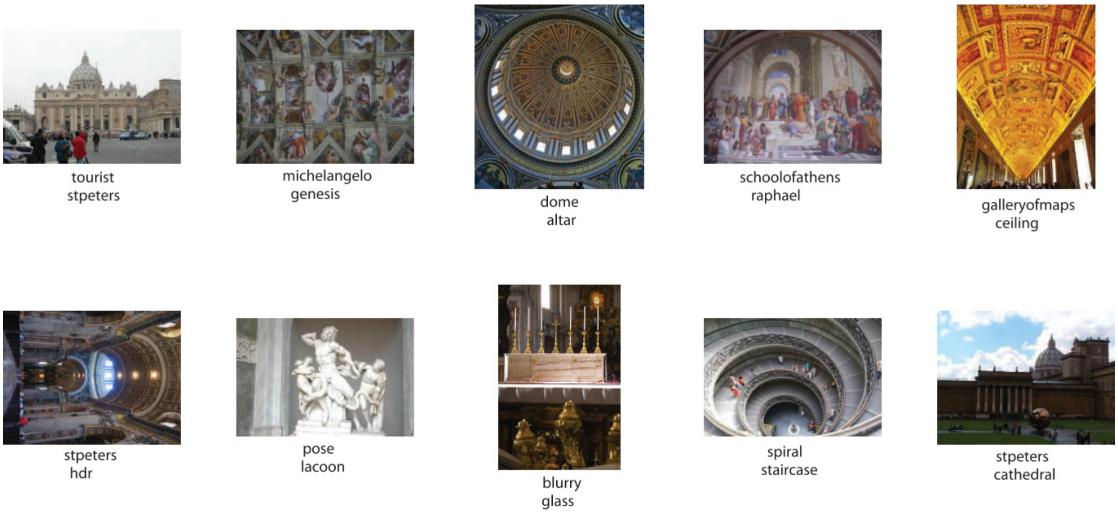


Figure 3.2: A 10-images summary of 2000 images of the Vatican computed by the algorithm of Simon et al. [SSS07] (Image taken from [SSS07]).

Simon et al. [SSS07] mainly focus on the likelihood criteria, enforced by a quality term for each image in the dataset expressing the similarity to its closest canonical view in the summary. However, they also add a cost term that enforces orthogonality to their quality function, and they use a cost term for each canonical view to limit the size of the summary. They use a greedy algorithm that adds those images to the summary that will cause the largest increase in the quality function until the quality term no longer outbids the orthogonality and summary size term. One example of a 10-image summary of 2000 images of the Vatican computed by their algorithm is shown in Figure 3.2 [SSS07].

Samani and Moghaddam [SM18] propose a multi-criteria context-sensitive approach for social image collection summarization. They combine two different sets of features that each look at different criteria for image summarization: social attractiveness features and semantic features. They use social network infrastructure to identify attractiveness features and domain ontology for extracting ontology features. Samani and Moghaddam [SM18] use the following low-level features defined by Geng et al. [GYX+11] as attractiveness features:

- **Perceptual quality**: Brightness, contrast, colorfulness, sharpness, and blur.
- **Aesthetic sensitivity**: Rule of Thirds, simplicity, and visual weights of the subject and the background.
- **Affective tune**: Emotional impact of image visual elements to humans expressed by line-based and color-based features.

For the domain ontology, they apply a hierarchical classification technique, where at each semantic node in the domain ontology, a classifier is applied to discriminate between

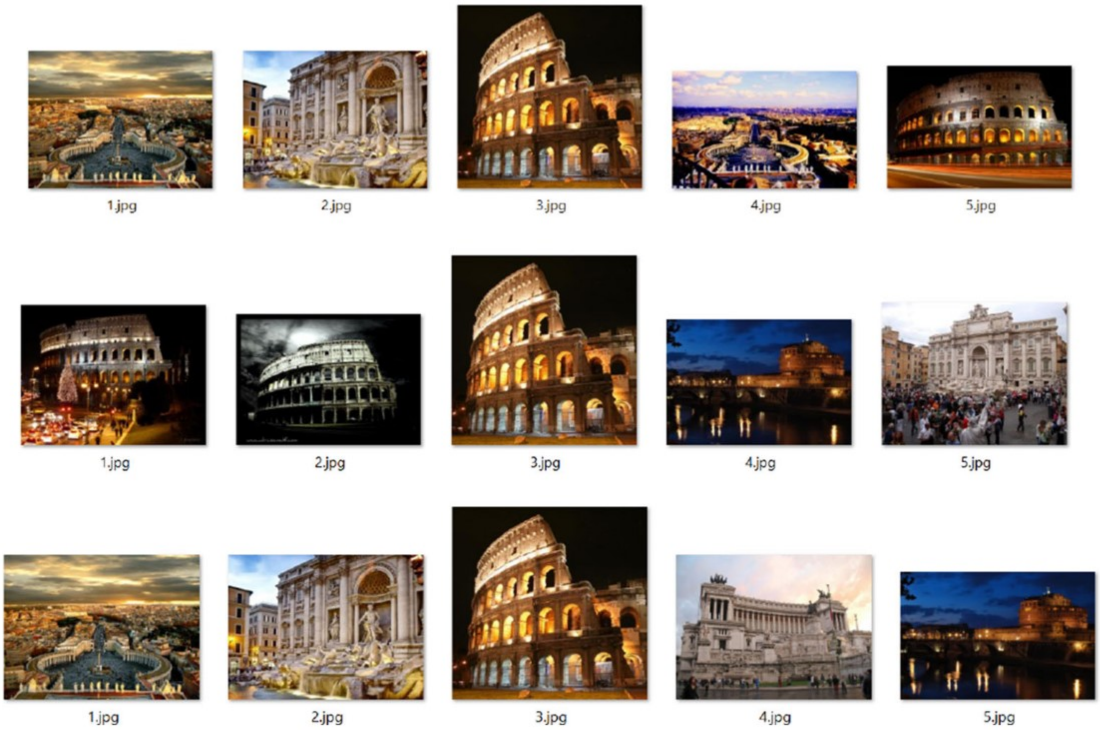


Figure 3.3: The result of summarization approach by Samani and Moghaddam [SM18] of a set of 32 images related to the city of Rome: The top row shows five images with a high attractiveness score, the middle row shows five images with a high semantic score (middle), and the last row shows the combination of both scores (Image taken from [SM18]).

sub-concepts of the current node. For this hierarchical classification process, they use features defined on SIFT [Low04] local features. For both feature sets, they train a support vector classifier. The final summary was produced by aggregating the results on both sets of features [SM18] (see Figure 3.3).

Summarizing photo collections or select the best image of a series of near-duplicate images is a subjective task. Except for Walber et al. [WSS14], who try to consider the individual interests of each user, all presented approaches handle the subjectivity of this task by selecting the images preferred by most of their participants. By recording and analyzing gaze information while the user views a photo collection (see Figure 3.4), information about the user's interests is obtained and used in order to create a personal photo selection. Those gaze paths are recorded in a first experiment, where the participants were just told to get an overview over the collection just with information that they will have to make a selection in later experiments (those without gaze recording). From the recorded gaze paths in the first experiment, measurements for each image (e.g., fixation count, average fixation duration, max visit duration) are calculated and used as features

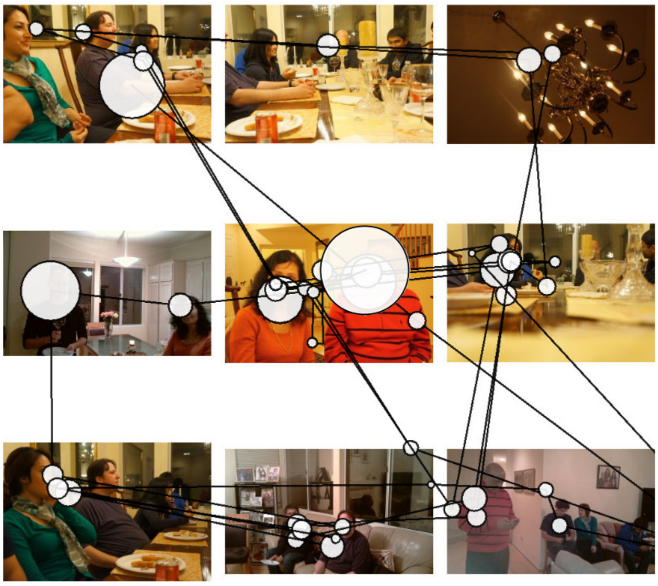


Figure 3.4: Visualization of a gaze path on a photo set by Walber et al. [WSS14] (Image taken from [WSS14])

for machine learning (logistic regression). The selections in the later experiments serve as ground truth labels. The selection talks were "Select photos for your private photo collection" (Task 1), "Select photos for giving your friends or family a detailed summary of the event" (Task 2) and "Select the most beautiful photos for presenting them on the web, e.g., on Flickr" (Task 3). Walber et al. [WSS14] conclude that the selection talk has a weak influence on the selection result. Their overall best selection result has a mean precision of 42.8%, where all measurements are combined (content, context, and gaze). However, it is worth noting that a single eye-tracking measure already had a mean precision of 42.1% without any machine learning [WSS14].

3.3 Image Selection

A strict borderline between image summarization and selection after the definition given at the beginning of this chapter is difficult because some summarization approaches do work on datasets that also contain similar images (e.g. [WSS14]) but treat them as independent images. Therefore, in this section, just approaches are presented that first group similar images and then apply a selection just within this group. The first presented approaches ([CL08], [CSN$^+$15], [LLC10], [YHBO10]) work with hand-crafted features, while the last presented approach ([WVSZ20]) extracts features with a convolutional neural network. The baseline approach Chang et al. [CYW$^+$16] is also a photo selection approach that uses a convolutional neural network. Due to its importance for this thesis, it is presented in a own section (see Section 3.4).

Chu and Lin [CL08] group images based on extracted SIFT-features [Low04] and represent the relationship between near-duplicate photos as a non-directed, non-weighted graph $G =< V, E >$ where any node is at least one time determined as a near-duplicate to another one. Figure 3.5 shows an illustrative example of the relationships between near-duplicate images by Chu and Lin [CL08]. As a measurement to evaluate the centrality value of each node, Chu and Lin [CL08] chose the in-degree of each node. Therefore, in Figure 3.5, image (2) would be selected as the representative photo of the group [CL08].

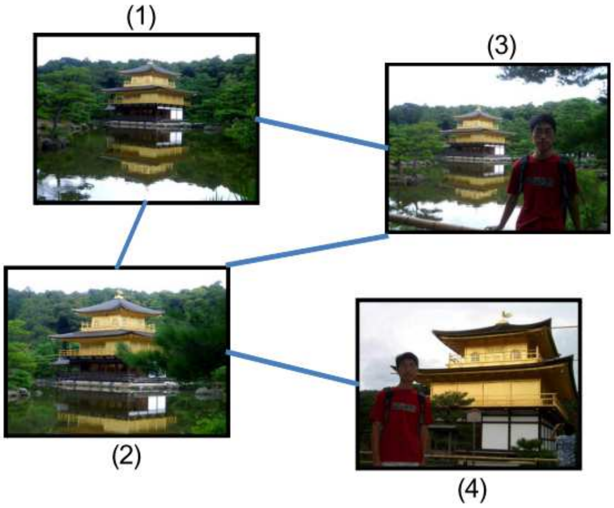


Figure 3.5: Selection of a photo most representative for a group by the number of relationships between near-duplicate photos. The idea behind this procedure is that the photo most similar to others is the most representative one for a group (Image taken from [CL08]).

Ceroni et al. [CSN+15] propose a photo selection method that they call expectation-orientated and that not strictly selects within pictures of the same scene. This is achieved by combining a variety of collection-level and image-level selection criteria in a flexible way. A single model is trained through machine learning to learn the different impacts of both collection- and image-level features, which consist of advanced concept detection, face detection, near-duplicate detection, quality assessment, temporal event clustering. The model predicts the probability of a photo to be selected, i.e., its importance. The predicts get ranked, and a subset of the collection gets selected by taking the top-k of them. In order to construct their dataset, they performed a user study where participants were asked to provide their personal photo collections and select the 20% images most important for revisiting or preservation purposes [CSN+15].

Li et al. [LLC10] propose an automatic aesthetics-based photo quality assessment, cropping-based photo editing, and selection of quintessential algorithm. Their approach differs from previously discussed selection approaches as they take faces into account to select or crop photos. In Figure 3.6 their pipeline is shown with an aesthetic score,

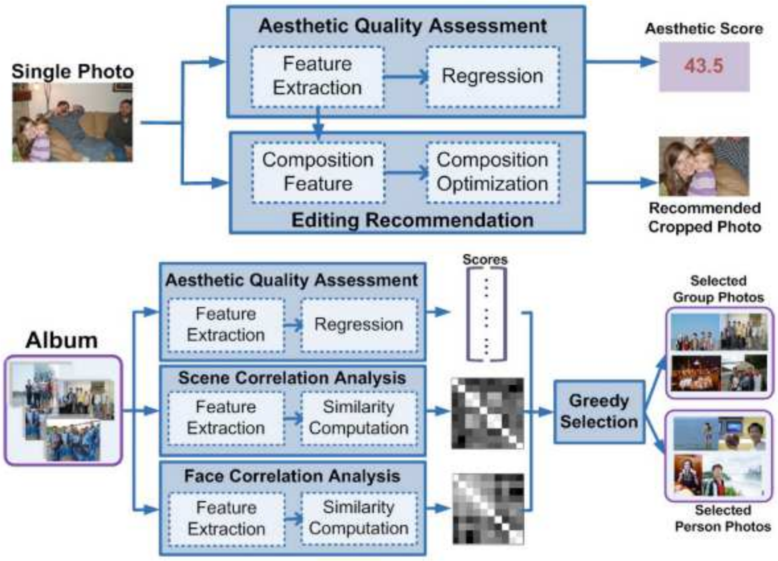
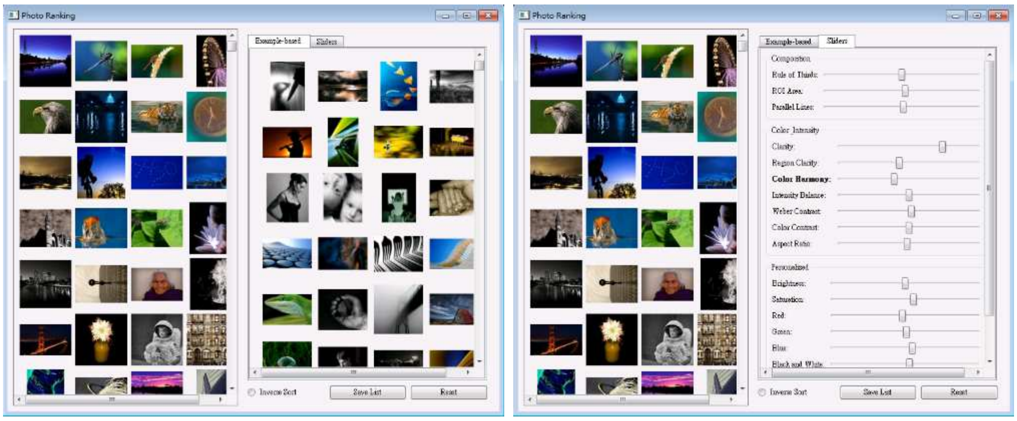


Figure 3.6: Automatic aesthetics-based photo quality assessment, cropping-based photo editing and selection of quintessential algorithm by Li et al. (Image taken from [LLC10]).

recommended cropped photo, and selected group/person photos output. Their aesthetics-based photo quality assessment algorithm considers different aesthetics-related factors, such as color and lighting in face regions, composition (i.e., the special arrangement of visual elements in the photo), and face characteristics. The face features are extracted using a commercial face detector, and the Active Shape Model [CTCG95] is used to detect faces and locate the position of 82 characteristic facial points. A sparsity-embedded linear regression model is used to map those extracted features onto an aesthetic score. As ground truth for the training, an earlier rating survey on 500 photos [LGLC10] is used. Instead of using all features proposed in [LGLC10], the sparse model automatically selects a small subset of important features, such that 35 features remain for group photos and 29 for individual photos. The goal of their photo selection algorithm is to select the most quintessential photos of a collection in two subsets: Group Photos and Single Person Photos. To achieve this, in addition to the aesthetic quality of a photo, they also measure the scene and face identity similarity between the photos in order to cover as many scenes and as many people that have appeared in the collection [LLC10].

Yeh et al. [YHBO10] propose an example- and feature-based reranking approach for selecting the user-preferred best images of a collection (Figure 3.7). They use ListNet [CQL$^+$07] in order to derive the weightings of rules employed to rank the photographs. The ListNet is adopted to train the prediction model in order to find the optimal weightings for each feature. Therefore, the score of each photograph is a linear combination of each feature and its corresponding weighting factor. As features, they use the photograph composition (rule of thirds, simplicity), color, and intensity distribution (texture, clarity, color harmonization, intensity balance, contrast) and personalized features (color



(a) (b)

Figure 3.7: Example- (a) and feature-based (b) personalized ranking of photo collections approach of Yeh et al. (Images taken from [YHBO10]).

preference, black-and-white-ratio, portrait with face detection, aspect ratio) [YHBO10].

In contrast to all other discussed approaches, Wang et al. [WVSZ20] present a photo selection system that runs in real-time on mobile cameras and works on burst images (see Figure 3.8 for example). Their system runs in the viewfinder mode and captures a burst sequence of frames before and after the shutter is pressed. It correctly predicts users' top-1 choice (out of 11 on average) for 64.1% cases and top-3 choice for 86.2% cases on their collected burst dataset with total 14, 769 bursts. They developed a deep neural network ranking model that implicitly learns the subtle visual differences within a sequence of burst images, such as sharpness, eye close or open, the attractiveness of body pose, or overall composition. Multiple network backbones and head design choices were explored by Wang et al. [WVSZ20] in order to develop a light-head network to learn the ranking function. After any backbone network, they append a global averaging pooling and two convolutional layers as head design. Wang et al. [WVSZ20] apply the idea of Generative Adversarial Networks (GANs) to perform feature space augmentation with goal of training the last convolutional layer to be a good ranker of the extracted features. In traditional GANs, the generator is the major learning objective, which is not the case in the approach of Wang et al. [WVSZ20], since the ranker is the learning

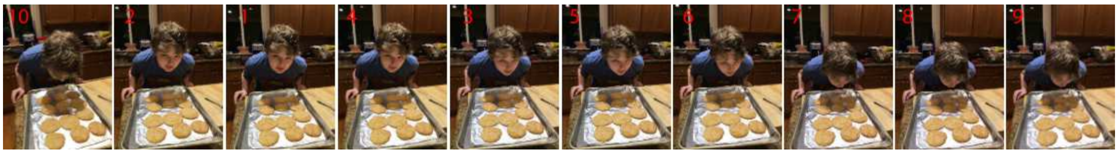


Figure 3.8: Burst ranking predicted by Wang et al. [WVSZ20] (Images taken from [WVSZ20]).

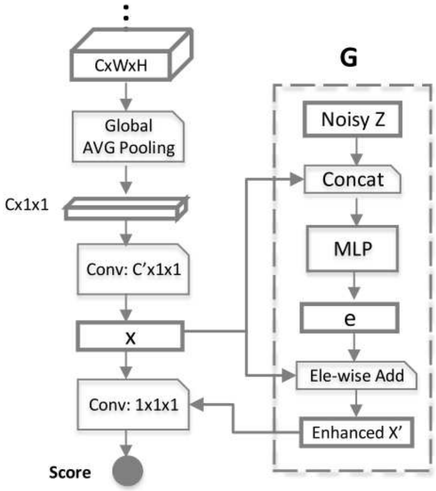


Figure 3.9: Head-design of the CNN architecture of Wang et al. [WVSZ20]. The generator component G only exists during training (Image taken from [WVSZ20]).

objective. In Figure 3.9, the Generator-component ("G") of the GAN is shown. This component only exists during training and takes the intermediate result before the last convolution layer and performs feature space augmentation with a noise function and a multi-layer perceptron. Both ranker and generator are trained iteratively. They showed the improvements in the usage of GAN for three different SqNet as backbone for a varying numbers for the relative attributes C'. The highest improvement that can be observed from their comparison (please refer to Table 1 in [WVSZ20]) is the improvement from 77.1% to 78.9% for $C' = 100$ and Top-2, i.e., for up to 1.8% of the images the correctness of the ranking position was improved. The highest Top-1 performance they could achieve is 65.9% on their burst image dataset.

3.4 Baseline approach Chang et al. [CYW+16]

Because this thesis is partly based on Chang et al. [CYW+16] and their "Auto Triage" dataset is used, their approach and dataset is described more profound than for previous approaches.

3.4.1 Dataset

In order to collect the dataset, Chang et al. [CYW+16] ran a contest in which they asked participants to submit unedited or slightly edited personal photo albums. They collected over 350 album submissions from 96 contributors which contain 15, 545 person photos. After removing extremely redundant shots, e.g., those captured in a burst session, they grouped images, based on SIFT feature matching [Low04] between neighboring images, into series. Within those series, all combinations of two images are paired side-by-side, and participants on Amazon Mechanical Turk (MTurk) were asked if they could one keep one of the two photos, which one they would choose [CYW+16]. From the pairwise comparison results, a global ranking for each series' image is obtained using the Bradley-Terry model. The Bradley-Terry model describes the probability of choosing an image I_i over another image I_j as a sigmoid function of score difference $\Delta_{i,j} = c_i - c_j$ between two photos, i.e.:

$$P(I_i > I_j) = F(\Delta_{i,j}) = \frac{e^{\Delta_{i,j}}}{1 + e^{\Delta_{i,j}}} \tag{3.1}$$

The score parameter c required to compute $\Delta_{i,j}$ can be estimated by solving a maximum a-posteriori (MAP) problem. By assuming a uniform distributed prior, the objective is to maximize

$$\log P_r(S|c) = \sum_{i,j} s_{i,j} F(\Delta_{i,j}) \tag{3.2}$$

which could be solved by using gradient descent. $S = \{s_{i,j}\}$ is the count matrix of all pairwise comparisons of a series, where $s_{i,j}$ is the number of times that photo I_i was preferred over photo I_j by the participants of MTurk [CYW+16].

3.4.2 Methodology

Chang et al. [CYW+16] tried hand-crafted features, CNN (AlexNet [KSH12] and 16-Layer VGGNet [SZ15]), CNN (AlexNet and VGGNet) + hand-crafted features, and CNN Siamese (AlexNet and VGGNet). Since VGGNet Siamese was their best-performing method, this description will focus on this method. The main benefit of using a convolutional neural network is that the network directly learns the image representation and prediction model without needing hand-crafted features. Chang et al. [CYW+16] argue that the success of convolutional neural networks in the ImageNet challenge demonstrates that learned representation can outperform hand-crafted features. For a network to learn a natural image representation, it usually requires millions of parameters. Although Chang et al. [CYW+16] have 12,075 training images in their dataset, learning from scratch can still get stuck in a local minimum. A common solution is to transfer weights from a similar domain, such as the ImageNet challenge, to have a good initialization. Transfer learning involves two steps. At first, the network is extended to fit a new domain by adding a few components. Second, the network is initialized by the model parameters pre-trained on the image classification task and fine-tuned by training with the data for the new task [CYW+16]. Yu and Koltun [YK16] have found that even if the original network is slightly modified, the pre-trained weights can still act as an effective initialization. Chang et al. [CYW+16] tried to design a network based on an existing architecture that fits their problem. In comparison to networks for image classification, the problem of Chang et al. [CYW+16] requires a network to take two images as input and output a pair of scores, which leads them to consider siamese architecture to learn image feature extraction. In a siamese architecture [BGL+93], two input images are processed in two identical subnets with shared weights simultaneously. Differently to the presented approach of Kong et al. [KSL+16], Chang et al. [CYW+16] train their Siamese CNN on binary preference labels instead of absolute aesthetic score differences. To get a binary label out of the extracted feature vectors of the two subnets, Chang et al. [CYW+16] designed a new cost function based on a Siamese architecture.

The input of their model is a pair of images $(I_1, I_2) \in \mathbb{I} \times \mathbb{I}$, where $\mathbb{I}$ is the space of images. For the whole model, they aimed for a function $p : \mathbb{I} \times \mathbb{I} \mapsto \{-1, 1\}$, that is skew-symmetric $(p(I_1, I_2) = -p(I_2, I_1))$ and where 1 means first image is better while -1 means the opposite. To achieve this, Chang et al. [CYW+16] split up the prediction function p into a feature extraction stage $s : \mathbb{I} \times \mathbb{I} \mapsto \mathbb{R}^n$ and a classifier stage $f : \mathbb{R}^n \mapsto \{-1, 1\}$ such that $p(I_1, I_2) = f(s(I_1, I_2))$. A Siamese architecture is used to learn s and a multi-layer perceptron to learn f.

The first stage (s) consists of two identical subnets with shared weights, and the final output is the difference between the outputs of the two identical sub-networks with different input. The difference is passed to the second stage that classifies which image is better (Figure 3.10). For the identical subnets, they tried both AlexNet and 16-layer VGGNet and initialized them with weights trained on the ImageNet dataset. However, they removed the last fully connected and soft-max layers as well as the Dropout layers. As a multi-layer perceptron classifier, they use two hidden layers with each 128-dimension output. Each hidden layer consists of a linear fully connected layer and a *tanh* as non-linear activation layer. As the last step, a two-way softmax indicates which of the two input images is better. The two stages are concatenated and trained together. Each image is resized such that the larger dimension fits the required size of the network (AlexNet: 227 × 227, VGGNet: 224 × 224) and padded with the mean pixel color in the training set. Unfortunately, they did not describe if they padded the image on one side only or on both sides, as discussed more in detail in Section 4.1 in the attempt of reproducing their results.

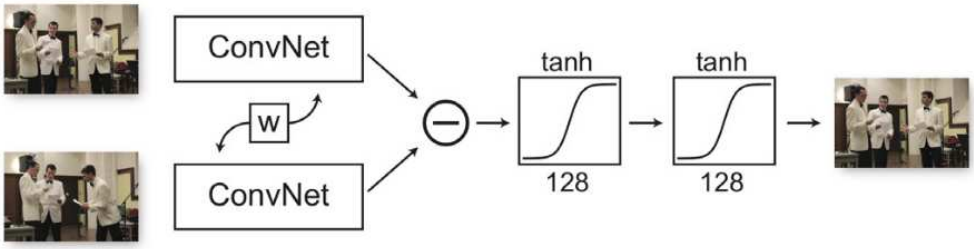


Figure 3.10: Architecture design of Chang et al. [CYW+16]: Features of both images are first extracted in a Siamese architecture by two identical subnets with shared weights. The difference of the features is passed through a two-layer perceptron, and a two-way softmax decides which image is better (Image taken from [CYW+16]).

3.4.3 Results

In order to compare the results of this thesis to Chang et al. [CYW+16], their results will be presented. They compare their different methods at two levels: series-level and pair-level (see Figure 3.11).

On the pairwise level, they choose to just consider pairs with clear human preference as test pairs, i.e., only pairs with majority agreement over 70% calculated by Bradley-Terry model. They argue though their observation that as human opinions become more consistent, performance of their model increases as well on both validation and testing set. As shown in Figure 3.11 (right bar chart), their VGG Siamese model achieves the best overall performance with 73% accuracy. On series level, they compute the logarithm of the joint probability of the decision of each method on all pairs that include the best image of a series. The resulting log-likelihood value is negative and normalized such that random guessing has value −1 (see Figure 3.11 left bar chart). The reason why they use log-likelihood measurement instead of frequency is, to equalize the difficulties of series of different sizes since it is harder to choose the best image from larger series than smaller ones [CYW+16].

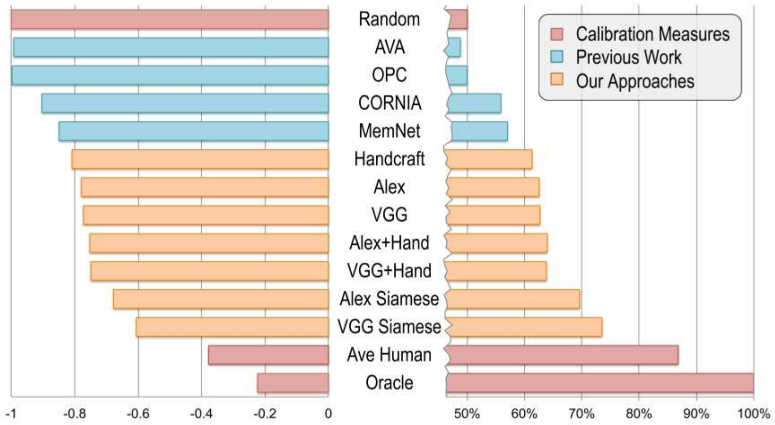


Figure 3.11: Benchmark performance comparison of the methods of Chang et al. [CYW+16] and previous work at series-level (left bar chart - log likelihood, normalized such that −1 is random guess) and pair-level accuracy (right bar chart) in predicting human preference among pairs where humans agree at least 70% of the time (further right indicates better performance in all cases). The "oracle" represents the majority of human votes, while "average human" displays the average performance of a single person (Image taken from [CYW+16]).

In this chapter, related work in the areas of image preference prediction, image summarization, and image selection was presented. The image selection approach of Chang et al. [CYW+16] was described more in detail because it is used as the baseline for this thesis.

CHAPTER 4

Methodology

The methodology of this thesis is partly based on Chang et al. [CYW$^+$16] (see Section 3.4), and their dataset is used for training and evaluation. All models discussed in this chapter are Siamese models, i.e., models that take two images/faces as input (I_1, I_2) and output a pair of scores (S_{I_1}, S_{I_2} with $S_{I_1} = 100\% - S_{I_2}$) that predict the human preference between the pair of image/faces.

As discussed in the introduction, the goal of the thesis is to improve the approach of Chang et al. [CYW$^+$16] through combination with a Face Model and get a stand-alone solution with state-of-the-art performance on images containing humans. The motivation for this thesis bases on a simple idea: A CNN is likely better able to predict facial differences if it just gets extracted face regions as input. The prediction of facial differences on just faces might also be better than the implicit face prediction by Chang et al. [CYW$^+$16]. This is based on the hypothesis that humans decide primarily based on facial features if humans are present in the photo. This hypothesis is strengthened by the MTurkers' comments Chang et al. [CYW$^+$16] received on portrait photos (Reasons for preferred photos: "smiling face", "closer face", "face angle" and "eyes") which indicate the important role facial features play. As it rejects all other features of the image (e.g., context, composition, brightness and blurriness of the background), in favor of capturing more detailed features of the faces, this approach only works on images containing faces.

In Figure 4.1 an overview of the best performing method developed in this thesis is given. First of all, the model of Chang et al. [CYW$^+$16] is reproduced (Section 4.1). Then the faces are extracted (Section 4.2) and matched (Section 4.3). In Section 4.4 two approaches are tried to train a Face Model, one using the Mirror Mirror dataset of Zhu et al. [ZAE$^+$14] and one using a self-constructed face dataset consisting of faces extracted from the dataset of Chang et al. (this approach is shown in Figure 4.1). The second approach required the acquisition of crowd-sourced labels as the face pairs' ground truth.

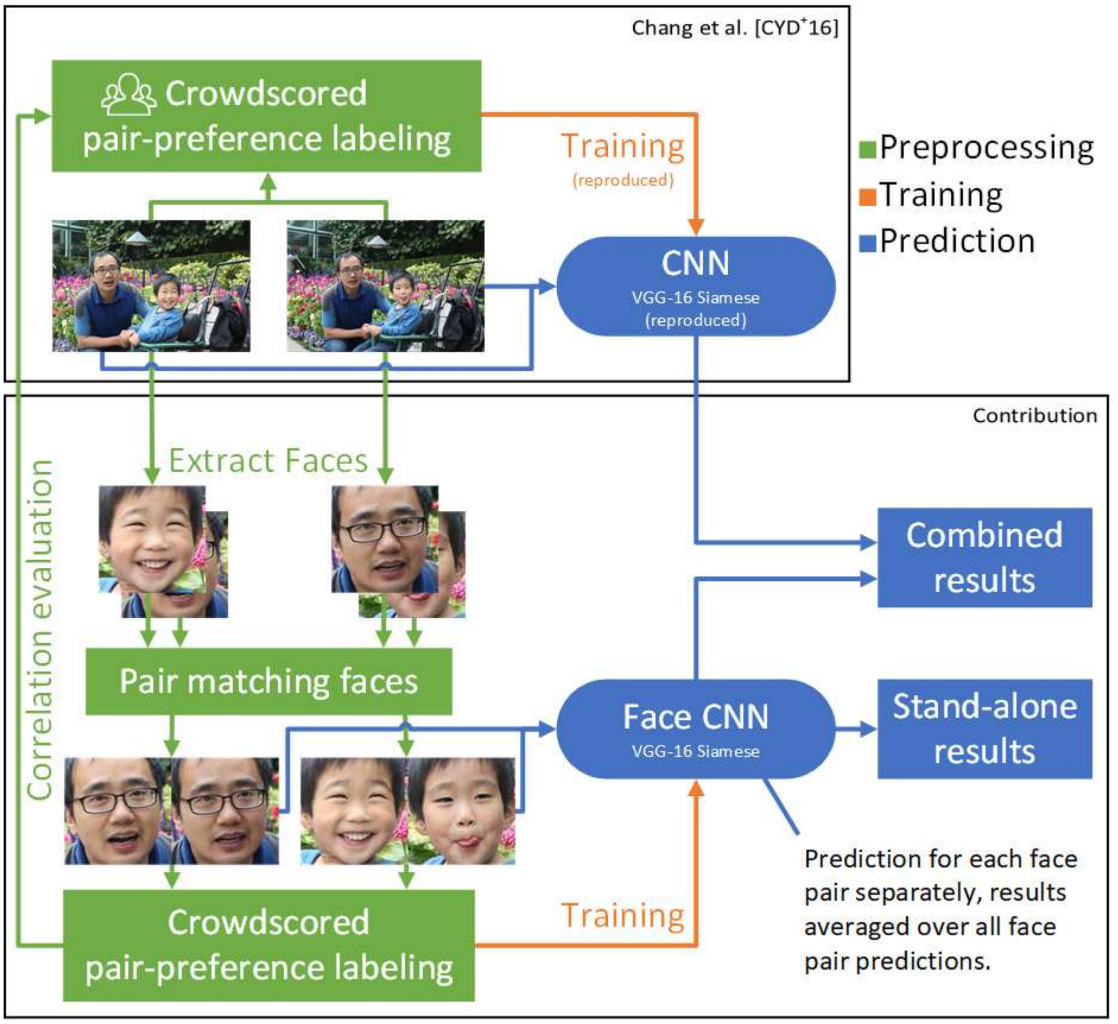


Figure 4.1: The pipeline of the best performing method developed in this thesis. The CNN model and training of the baseline method of Chang et al. [CYW+16] was reproduced. As preprocessing for the own contribution, faces were extracted from each image of an image pair and matching faces were grouped. Each face pair was crowdscored labeled for human preference between the two face images. The correlation of the obtained face labels to the labels on the whole images of Chang et al. [CYW+16] is evaluated and the face labels were used to train a Face CNN. The performance of the Face CNN was evaluated in two ways: As stand-alone approach and in combination with the baseline approach (Images taken from [CYW+16]).

4.1 Reproducing results of Chang et al. [CYW+16]

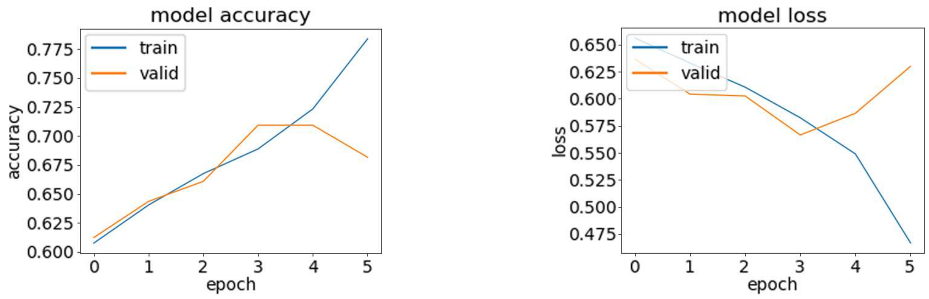
As this work focuses on improving the results of Chang et al. [CYW+16] on just images that contain faces, the approach of Chang et al. [CYW+16] needs to be reproduced in order to also get predictions for Chang et al. [CYW+16] for the combined approach. Further, the reproduction of Chang et al. [CYW+16] is needed to obtain comparison values for just face images.

4.1.1 Retraining model in Tensorflow

The original paper was written in Caffe[1], which is outdated because its successor (Caffe 2) was merged into Pytorch[2]. Furthermore, it was preferred to work in Tensorflow. Therefore, a reimplementation in Keras, the high-level API of Tensorflow, was searched. The repository of Zhijian Liu[3] fit the objective after correcting several errors in the code as described in the Appendix. Training with the described changes to the code and addition of parameters as described in the Appendix, resulted in the following results (Table 4.1):

Set	Train	Valid	Test
Accuracy	73%	71%	70%

Table 4.1: Results of the retrained model of Chang et al. [CYW+16].



(a) Model accuracy

(b) Model loss

Figure 4.2: Model accuracy and loss evolution of the retrained 16-layer VGGNet Siamese approach of Chang et al. [CYW+16]. The training was stopped in the fifth epoch, and the weights of the fourth epoch were restored, since the training and validation accuracies started to diverge.

As visible in Figure 4.2, the validation accuracy decreased while the training accuracy continued to increase, which indicates that the model would start to overfit from here.

[1] `https://caffe.berkeleyvision.org/` accessed on 11.2.2020

[2] `https://medium.com/@Synced/caffe2-merges-with-pytorch-a89c70ad9eb7` accessed on 11.2.2020

[3] `https://github.com/zhijian-liu/auto-triage` accessed on 11.2.2020

Therefore, the training was stopped, and the model weights from the fourth epoch were restored. Since retraining the weights did not reach 73% test performance reported by Chang et al. [CYW+16], two other approaches were tried, which will be described in the next two sections.

4.1.2 Transfering Caffe Weights to Keras Weights

In order to work with the weights trained by Chang et al.[4], multiple repositories [5,6] were tried. Nevertheless, just the weight converter[7] by Pierluigi Ferrari worked. The converter was executed on a Linux distribution (Ubuntu) because installing Caffe (a required dependency of the converter) did not work on Windows. After exporting the weights from the Caffe model into a file, import of the weights into the Tensorflow reimplementation[8] of Zhijian Liu failed because of a layer count mismatch. Since the VGG16 model from Keras is already packed as a model and therefore treated as one layer, the weight importer just detects four layers, including the perceptron. As it turns out, the reason for the contained 33 layer is that the shared weights of two Siamese sub-nets were separately saved for each of the two sub-nets. With the help of the Keras VGG-16 model definition[9], the VGG-16 model was reconstructed as mentioned in the VGG Siamese model definition of Chang et al. with separate VGG16 networks definition but with the same layer names. Importing the weights did only work by name-matching, since the axes did not match. The success of the import was verified by checking if the models' weights are different from null. Unfortunately, the evaluation results with the author's weights were unacceptable (Table 4.2):

Padding color	Padding sides	Train Accuracy	Valid Accuracy
Black	right/bottom	54.0%	52.9%
Mean color train	right/bottom	53.4%	55.4%
Mean color train	left+right/top+bottom	56.1%	55.4%

Table 4.2: Prediction results with the transferred Caffe weights to Keras weights.

Since initializing all weights with random values results in 50% training and 50% validation accuracy, weights were likely not completely/correctly loaded. Of course, it can not be known for sure if the Caffe-to-Keras weight converter by Pierluigi Ferrari worked correctly.

[4] `https://phototriage.cs.princeton.edu/data/vggsms.zip` accessed on 13.2.2020

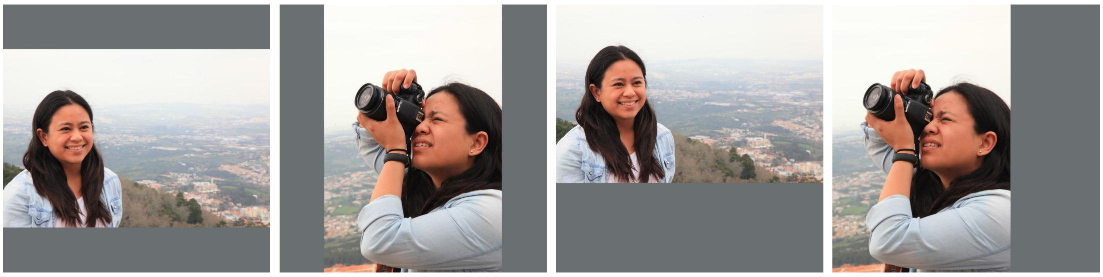
[5] `https://github.com/marvis/pytorch-caffe` accessed on 11.2.2020

[6] `https://github.com/dhaase-de/caffe-tensorflow-python3` accessed on 13.2.2020

[7] `https://github.com/pierluigiferrari/caffe_weight_converter` accessed on 13.2.2020

[8] `https://github.com/zhijian-liu/auto-triage` accessed on 13.2.2020

[9] `https://github.com/keras-team/keras-applications/blob/master/keras_applications/vgg16.py` accessed on 14.2.2020



(a) Top+Bottom (b) Left+Right (c) Bottom (d) Right

Figure 4.3: Different padding possibilities: Both sides (a)(b) or one side (c)(d).

4.1.3 Predict images in Caffe

As neither retraining the model of Chang et al. [CYW+16] using the repository described in Section 4.1.1 nor transferring the weights from the Caffe model to Keras as described in Section 4.1.2 worked, it was left to try loading the Caffe model and its weights in Caffe and exporting the predictions for later use. As discussed before, the authors have not mentioned if they padded the images on both sides (see Figure 4.3a and 4.3b) or just one side (see Figure 4.3c and 4.3d). So both methods were tried and continued with the better performing one. The images are padded with the mean color in the training set (8 bit RGB: Red=106, Green=111, Blue=114), as described in the paper [CYW+16]. Finally, the exported predictions were submitted to the benchmark[10] of Chang et al. [CYW+16]. The results were **70.9%** accuracy in predicting human preference among pairs for right/bottom padding and **71.8%** accuracy for left+right/top+bottom padding.

Unfortunately, 71.8% do not reach the 73% that Chang et al. [CYW+16] described in their paper. However, this was the closest their results could be reproduced. The complete prediction results, including training and validation sets, are shown in Table 4.3.

Set	Train	Valid	Test
Accuracy	79.3%	66.8%	71.8%

Table 4.3: Accuracy of the reproduced prediction results of Chang et al. [CYW+16].

Thus, it can be continued by the contribution of this work, which starts by extracting faces.

4.2 Face Extraction

The faces used in this thesis are extracted from the "Mirror Mirror" dataset by Zhu et al. [CYW+16] or the "Auto Triage" dataset by Chang et al. [CYW+16]. As described

[10] https://phototriage.cs.princeton.edu/benchmark.php accessed on 18.2.2020

later (Section 4.4), model training was evaluated with faces extracted from the "Mirror Mirror" and the "Auto Triage" dataset and continued with the better performing training in the prediction stage. Since the images in both datasets are too widely cropped to serve as face model input for both training and prediction of new images, i.e., showing too much from the background, a face detector is needed. Therefore, the two face detectors haarcascade frontalface detector (Viola-Jones) of OpenCV [JV03] and the standard face detector "Multi-task Cascaded Convolutional Networks" (MTCNN) of FaceNet [ZZLQ16] are tested. These two detectors were selected due to their assumed robustness shown from their application in work other than their original paper and their simple application in the used programming language Python.

In the case of the haarcascade frontalface detector, the faces were padded (left+right / top+bottom) to fit square size. Since the detected faces of MTCNN already have a square bounding box, no padding is needed. The faces are resized to size 224 × 224 to have the necessary size for later training.

For MTCNN, all faces with less than 0.5 confidence were rejected. Both detectors successfully detected all faces in the Mirror Mirror dataset. However, since MTCNN detected more correct faces and less false positives than haarcascade frontalface on the Auto Triage dataset of Chang et al. [CYW+16], it was decided to continue with MTCNN for all use cases, including the Mirror Mirror dataset. Another reason for preferring MTCNN is that MTCNN just wrongly detects the faces of statues and faces in wall posters as false positives, while haarcascade frontalface detector also detects other objects as faces, e.g., parts of buildings. The small amount of false detections were removed manually for the training of the Face Model (Section 4.4). Nonetheless, no manual intervention is performed in the prediction and evaluation.

4.3 Face Matching

To match the faces of the same person, FaceNet [SKP15] is used. FaceNet is a deep convolutional network suitable for face verification, recognition, and clustering. The model is end-to-end trained to produce face embeddings whose squared distance is small for a pair of face images of the same person and large between a pair of face images from different identities.

The embedding of a face can be calculated using FaceNet [SKP15] and the Euclidean distance of its embeddings calculates the similarity distance of a face pair:

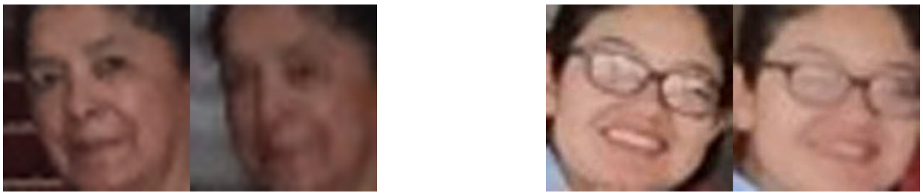
$$d_{NN}(F_1, F_2) = \sqrt{\sum [Embedding(F1) - Embedding(F2)]^2} \tag{4.1}$$

As sharpness differences could be observed in the detected faces (see Figure 4.4 for examples), the embedding of each face was calculated additionally from a Gaussian blurred version of the face with kernel size 9. The distance between two faces (Equation 4.4) was calculated as the minimum of the distance between the embeddings of original faces (Equation 4.1) and the distance between the two faces embeddings, where one was blurred (Equation 4.2 and 4.3):

$$d_{NB}(F_1, F_2) = \sqrt{\sum [Embedding(F1) - Embedding(Gaussian_{9\times 9}(F2))]^2} \tag{4.2}$$

$$d_{BN}(F_1, F_2) = \sqrt{\sum [Embedding(Gaussian_{9\times 9}(F1)) - Embedding(F2)]^2} \tag{4.3}$$

$$d(F_1, F_2) = min(d_{NN}(F_1, F_2), d_{NB}(F_1, F_2), d_{BN}(F_1, F_2)) \tag{4.4}$$



(a) Embedding distance $d = 2.28$ ($d_{NN} = 2.49, d_{NB} = 2.98, d_{BN} = 2.28$)

(b) Embedding distance $d = 2.55$ ($d_{NN} = 2.71, d_{NB} = 3.34, d_{BN} = 2.55$)

Figure 4.4: Two examples of face pairs, where equation 4.4 brought an advantage compared to Equation 4.1. The displayed faces are the unblurred, and the displayed embedding distance is the result of equation 4.4.

$F1$ and $F2$ are faces from different images within one series of the dataset of Chang et al. [CYW$^+$16] resized to 224 × 224. The $Gaussian_{9\times 9}$ blurs an image without changing the image size. Both d_{NB} and d_{BN} are necessary, because it is unknown which face is more blurred of a pair. The observation was that similar blurred faces have a lower embedding distance than faces with different sharpness. This formula change led to a decreases matching distance for face pairs with different blurriness. Figure 4.4 shows two face pairs those embedding distance would have been higher under the usually applied equation 4.1.

Imagine the situation that the face detector has found faces in two images A and B: In image I_A were three faces found, in image I_B were two faces detected. Of course, just two of the three faces from image I_A can match with the two faces from I_B, as it can be assumed that the same person cannot be twice in an image. Therefore, as the first attempt, faces from image I_A and I_B were paired, which had a similarity distance of its FaceNet embeddings lower than a certain threshold.

There are two problems of using a threshold to find matching face pairs: First, it is difficult to find an optimal threshold that works for all image pairs. Second, this does not guarantee the number of matched face pairs is lower or equal to the minimum number of detected faces in both images, and further, it is not granted that the same face does not get matched to two faces in the other image.

For those reasons, a greedy approach was used that fulfills those requirements: All face pairs are organized in a list and ordered by increasing embedding distances. From this list, face pairs are iteratively selected. If none of the faces of a face pair was previously

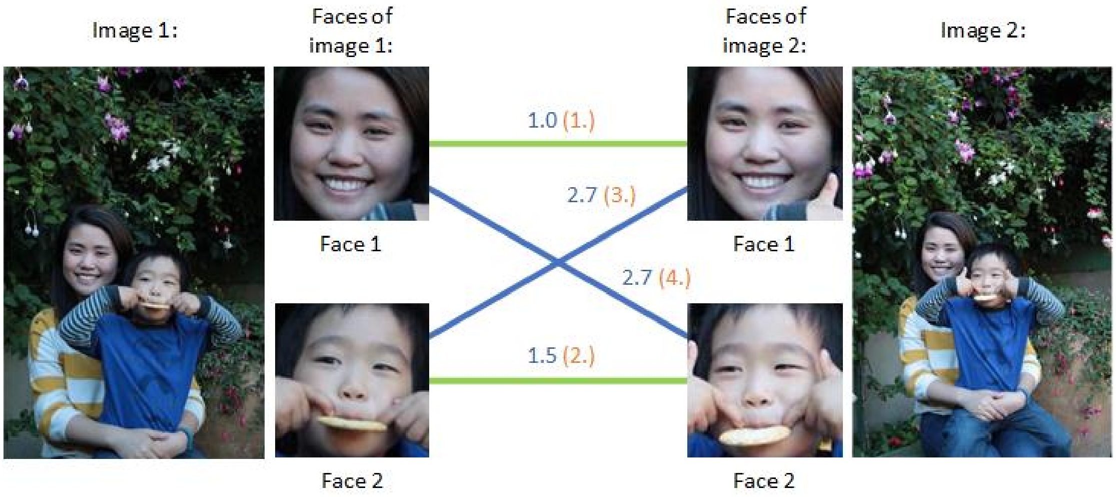


Figure 4.5: Visualization of the greedy face matching selection algorithm. At first, the face matching distances (shown on top of the lines) get sorted (final order in the bracket). Then the faces pairs get checked in increasing order and selected if none of the two faces was selected before: 1) 1.0 → Select and mark both faces as selected 2) 1.5 → Select and mark both faces as selected 3) 2.7 → Face 2 of Image 1 and Face 1 of Image 2 was selected before 4) 2.7 → Face 1 of Image 1 and Face 2 of Image 2 was selected before.

Algorithm 4.1: Greedy face matching

```
Input: List of all face pair combinations $facePairs$ between the $numFaces1$
       faces found in $image1$ and the $numFaces2$ faces found in $image2$
Output: Greedy selection of $min(numFaces1, numFaces2)$ face pairs

$faces1Used \leftarrow$ false$[numFaces1]$
$faces2Used \leftarrow$ false$[numFaces2]$
$facePairsSelected = []$

$facePairsSorted = sortByDistanceASC(facePairs)$

foreach $facePair \in facePairsSorted$ do
    if $faces1Used[facePair.indexFace1] ==$ false $AND$
     $faces2Used[facePair.indexFace2] ==$ false then
        $faces1Used[facePair.indexFace1] =$ true
        $faces2Used[facePair.indexFace2] =$ true
        $facePairsSelected.append(facePair)$
    end
end
return $facePairsSelected$
```

selected by another face pair with a lower matching distance, the face pair gets added to the result list (see Figure 4.5 and Algorithm 4.1). The resulting number of face pairs of this procedure is not higher than the minimum number of faces in each of the two images of an image pair, and it is also granted that no face gets paired twice.

There might be a matching solution with a lower total distance. Therefore, the Hungarian method of Kuhn [Kuh55] was tried, which finds the matching with minimal total distance. However, it led to no noticeable improvement in the results compared to the greedy algorithm.

4.4 Face Model: Dataset construction, Training and Prediction

In order to train a Face Model, two different datasets are validated: "Mirror Mirror" dataset of Zhu et al. [ZAE+14] (Section 4.4.1) and "Auto Triage" of Change et al. [CYW+16] (Section 4.4.2).

4.4.1 Face training with Mirror Mirror dataset

The first attempt was to train a face model based on an existing face dataset with ground truth labels. The dataset of "Mirror Mirror: Crowdsourcing Better Portraits" by Zhu et al. [ZAE+14] was found best suitable because it provides attractiveness labels which are assumed to correlate with human preference over selecting between images based on the facial expressions. In order to validate the assumption, a CNN model based on the attractiveness scores of "Mirror Mirror" dataset was trained and verified by testing the model on the "Auto Triage" dataset by Chang et al. [CYW+16]. In case it would perform well, the following requirements can be considered to be valid:

1. A high correlation between the attractiveness score and human preference on face expressions

2. Transferability of a model trained on the Mirror Mirror dataset to the Auto Triage dataset

3. A high correlation between the human preference of facial expressions and human preference of images containing one or more persons

In order to predict human preference for the Auto Triage dataset with a model trained on the Mirror Mirror dataset, we need all three requirements fulfilled. So in case the model training on the Mirror Mirror dataset works but applying the model on the Auto Triage dataset fails, one of those requirements might not be fulfilled.

4.4.1.1 Data set construction

First of all, the images were split per subject into ten images validation set and ten images testing set. The rest of the 200 - 262 images per subject remained for the training set. One might argue that not having separate images of different subjects not contained in the training set, is problematic since the model has learned images of the same persons as in the test set. However, since the goal is to apply the model onto a different dataset, this does not matter.

The dataset for training, validation, and testing was constructed by first building all possible pairs of images within the partitions mentioned earlier. Due to memory limitations on the training machine, from those pairs, a random subset of four times the number of single images in a set was selected, i.e., each image gets paired with four other images on average. In case the number of available pairs was lower than four times the number of single images, all available pairs were selected. This procedure resulted in 11590 training, 495 validation, and 495 test image pairs among all subjects. In a pair, the image with the higher attractiveness score got the label "1" assigned, the other "0".

4.4.1.2 Model training

The VGG-16 Siamese, pretraining with ImageNet weights, is used as CNN since this network performed best in the work of Chang et al. [CYW+16]. The training results are shown in Table 4.4:

Set	Train	Valid	Test
Accuracy	94.5%	83.2%	89.6%

Table 4.4: VGG-16 Siamese training results with the Mirror Mirror dataset [ZAE+14], faces extracted with the face detector (MTCNN) of FaceNet [ZZLQ16].

Therefore, the training of the Mirror Mirror model has worked well because the accuracy for all three sets is $> 80\%$, and the network did not overfit.

4.4.1.3 Applying the trained Mirror Mirror model to the dataset of Chang et al. [CYW+16]

For applying the trained face model by the Mirror Mirror dataset to the dataset of Chang et al. [CYW+16], it has to be discussed first, how the scores of multiple face pairs are combined. Let (I_A, I_B) be an image pair of the same series with scores $score(I_A)$ and $score(I_B)$ by Chang et al. [CYW+16] and let (F_A, F_B) be any matched face pair $(F_A \in Faces(I_A), F_B \in Faces(I_B))$. Predicting a face image pair (F_A, F_B) with the trained Mirror Mirror model outputs the following two scores: $score(F_A) = [0, 1]$ and $score(F_B) = [0, 1]$ with $score(F_A) = 1 - score(F_B)$. The face scores are combined by averaging:

$$score(Faces(I_A)) = \frac{\sum_{F_A \in Faces(I_A)} score(F_A)}{|F(I_A)|}$$
$$score(Faces(I_B)) = \frac{\sum_{F_B \in Faces(I_B)} score(F_B)}{|F(I_B)|}$$

with $score(Faces(I_A)) = 1 - score(Faces(I_B))$.

In the following Table 4.5 the predictions of the Mirror Mirror Model on the faces get compared to the ground truth of the whole images of Chang et al. [CYW+16] on face pair and image pair level:

(i) Per face pair: The scores of each face pair (e.g., $score(F_A)$ and $score(F_B)$) are compared to the label of the image pair, from which the two faces were extracted.

(ii) Per image pair: The average scores (e.g., $score(Faces(I_A))$ and $score(Faces(I_B))$) of the face pairs get compared to the label of the image pair (e.g., $score(I_A)$ and $score(I_B)$).

Set	Train	Valid	Test
(i) Per face pair	61.1%	53.5%	59.0%
(ii) Per image pair	61.7%	50.8%	57.5%

(a) Without any filters.

Set	Train	Valid	Test
(i) Per face pair	62.2%	55.0%	59.6%
(ii) Per image pair	62.9%	54.1%	57.5%

(b) With different filters: minimum face side length ≥ 50, sharpness (estimated with Variance of Laplacian) ≥ 3 and Face score $\geq 70\%$ since it was assumed that low resolution and blurry images would be very hard to correctly predict. Using those filters did slightly improve the results, but they are still not even close to the results of Chang et al. [CYW+16].

Table 4.5: Mirror-Mirror model applied to the dataset of Chang et al. [CYW+16].

As observable from Table 4.5, the test results with the Mirror-Mirror model did not get beyond 60%, which is far from the 73% performance of Chang et al. [CYW+16]. Possible reasons for the failed transfer to the dataset of Chang et al. [CYW+16] are the different blurriness of the faces, the small amount of subjects in the Mirror-Mirror dataset, or the limited amount of variation in the faces orientation and brightness. Therefore, the approach of using the Mirror-Mirror dataset [ZAE+14] for the training of a face model was discarded. An alternative approach was searched for, as described in the next section.

4.4.2 Face training with Auto Triage dataset and own labels

Since training a face model that transfers to the dataset of Chang et al. [CYW^{+}16] did not work with the Mirror dataset [ZAE^{+}14], the dataset was constructed with face pairs extracted from the dataset of Chang et al. [CYW^{+}16] and volunteers were asked to label those images. The face dataset was extracted from the dataset of Chang et al. [CYW^{+}16] like described in Section 4.2 and 4.3. In Section 4.4.2.1 the labeling process is described, and in Section 5.2, face labels are compared to the labels of Chang et al. in terms of how well faces are suited to predict preference on the whole image. Lastly, the training of the Face Model is described (Section 4.4.2.2).

4.4.2.1 Crowd-scored Labeling

The extracted 2,658 face pairs were uploaded to a web server, and a labeling software was written to allow participants to label the images. Since all participant's native language was German, all texts of the labeling process were displayed in German and later translated with identical meaning to English for this thesis. The content of one page of the labeling process is shown in Figure 4.6.

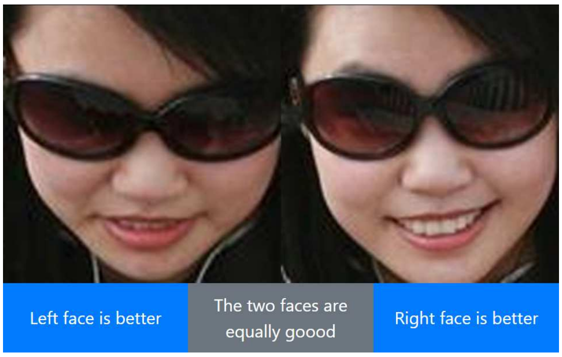


Figure 4.6: Labeling process (translated version; original in German)

In the top 80% of the browser window, one extracted face pair gets displayed in a scaled fashion such that the face pair fits the available space without changing the aspect ratio. Since the faces of the face pairs all got scaled to the required input size of the convolutional network (224 × 224 => 448 × 224 concatenated together), the display size of the participant's device (computer monitor or smartphone) does not matter since it can nowadays be safely assumed to be greater than 448 × 224 pixel. Therefore, no further downsizing of the faces is applied. Note that almost all faces got up-scaled to 224 × 224, and just some outliers get down-scaled (see Figure 4.7).

The participants were asked to choose between three options for each image: "Left face is better", "The two faces are equally good" and "Right face is better". In case of miss-clicking, the participants could correct the last three selections. Five participants completed labeling all images. Therefore, all face pairs have at least five votes by the same fives participants. Some face pairs also have a sixth label because three participants

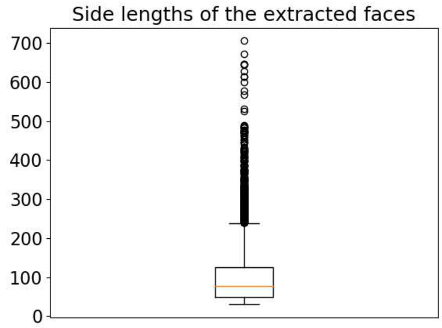


Figure 4.7: As shown in the box-plot, the side lengths of the extracted faces mostly lay below the CNN input side length of 224 pixels.

partly completed the labeling process. The labels are mapped to numbers: "Left face is better" → +1, "The two faces are equally good" → 0 and "Right face is better" → −1. The average of those numbered labels are calculated as the ground truth for each face pair.

4.4.2.2 Model Training based on Crowd-scored Labels

As described in the previous section, the labels are mapped to -1, 0, or $+1$, which results in an average (face pair ground truth) being between -1 and $+1$. An average close to 0 either means that many participants clicked "The two faces are equally good", the options "Left face is better" and "Right face is better" were clicked close to the same amount, or a combination of both. This can be interpreted that on face pairs with an average label close to $+1$ or -1 it was more clear for the participants to tell which face of the pair is better. In contrast, on face pairs with an average label close to 0, the faces are very similar, or the participants had different preferences about what is essential for a face to look good. For training, the face pair images were split into two separate images again and given the labels $[+1, 0]$ if the face pair ground truth was ≥ 0 and $[0, +1]$ if it was < 0. The CNN outputs two scores, s_1 for the first face image of a face pair and s_2 for the second face image of a face pair, while $s_1 = 1 - s_2$. Under the argument that faces with ambiguous human preference could confuse a convolutional neural network, face pairs, whose absolute value of the face pair ground truth lies under a certain threshold, got excluded. In contrast to this expectation, similar faces did not confuse the convolutional neural network. Furthermore, similar faces did prevented overfitting due to the network memorizing the smaller amount of remaining images if the threshold is set higher. Therefore, just face pairs with a ground truth label of exactly 0.0 got excluded, leaving $1,817$ of $2,078$ (87.4%) training, 92 of 104 (88.5%) validation, and 432 of 476 (90.8%) testing face pairs to be used for training and evaluation.

The reimplemented architecture of Chang et al. [CYW$^+$16], as described in Section 4.1.1, was used for training face models with those aggregated face labels. Three different convolutional neural networks (VGG-16, RetNet50, and SqueezeNet) as sub-networks were tried (see Table 4.6) and continued with the best-performing one (VGG-16). The performance of the best performing face model (VGG-16) concerning the whole image ground truth of Chang et al. [CYW+16] is shown in Section 5.4.

Set	Train	Valid	Test
Accuracy	84.8%	79.3%	76.2%

(a) VGG-16 Siamese face training results

Set	Train	Valid	Test
Accuracy	93.2%	75.0%	72.4%

(b) ResNet50 Siamese face training results

Set	Train	Valid	Test
Accuracy	79.3%	76.0%	74.1%

(c) SqueezeNet Siamese face training results

Table 4.6: Face training results of the evaluated convolutional neural networks as sub-networks for the architecture described in Section 4.1.1.

In this chapter, the methodology of the proposed approach was presented. At first, it was shown how the baseline approach of Chang et al. [CYW+16] was reproduced. Then was discussed how faces are detected and matched. Lastly, a face CNN was trained. Model training was tried with faces extracted from two different datasets. The first attempt was to train the face model with faces extracted from the "Mirror Mirror" dataset of Zhu et al. [ZAE+14]. The motivation of using a different dataset for the training of the face model was that for the dataset of Zhu et al. [ZAE+14], facial labels already existed. Unfortunately, the trained model did not perform well on the evaluation dataset of Chang et al. [CYW+16]. However, the second attempt, which uses crowd-scored faces extracted from the "Auto Triage" dataset of Chang et al. [CYW+16] for training, did work, as will be evaluated in the next chapter.

CHAPTER 5

Experiments

In this section, first it is described how the test labels for the evaluation of the experiments are acquired (Section 5.1). Afterward, the suitability of faces for predicting the preference on whole images (Section 5.2) and the performance of the face matching are evaluated (Section 5.3). Lastly, the quantitative results are shown in Section 5.4 and qualitatively analyzed in Section 5.5.

The reproduction of Chang et al. [CYW$^+$16] described in Section 4.1.2 and 4.1.3 was performed under Ubuntu 18.04, the rest of the tasks and the experiments were performed under Windows 10. The test machine's hardware: CPU i5-4690, GPU NVIDIA GeForce RTX 2060. The test machine's software under Windows: Python 3.7.3, Tensorflow 1.14.0, OpenCV 4.1.1, CUDA 10.0.130, CUDNN 7.6.3. For the experiments with the Caffe library under Ubuntu 18.04, Python 3.6.9 and Caffe-cpu 1.0.0-6 amd was used.

5.1 Acquiring test labels

The benchmark[1] provided by Chang et al. [CYW$^+$16] just provides the possibility to evaluate all test images and not just a subset. In order to better evaluate the subset of test images that contain faces, a userscript[2] (see Appendix Algorithm 5.1) was written to fetch the labels from the benchmark site[1]. The algorithm starts with a random 0 and 1 initialization of the labels. It toggles one label and evaluates if the score changed. If the score changed, the information about this improvement/worsening is saved to another array initialized with 0.5 values. In case of a lucky initialization, which happens more often than not, the algorithm terminates in one iteration. Otherwise it repeats until the correct solution is found. In the end, just those labels remain with the initial value of 0.5

[1] `https://phototriage.cs.princeton.edu/benchmark.php` accessed of 17.1.2020

[2] "A userscript is a program, usually written in JavaScript, for modifying web pages to augment browsing" [Wikipedia]

those average human agreement is $\leq 70\%$. By uploading the resulting output as a file to the benchmark[1], the correctness of the determined labels from the Algorithm 5.1 could be verified. Those determined test labels were used for the evaluation in Section 5.4.

Algorithm 5.1: Test labels acquisition

```
Function *evaluate(randomLabels: int[]) : float* is
    $randomLabelsJoinedString \leftarrow randomLabels.join("\backslash n") + "\backslash n"$
    $chars \leftarrow "abcdefghijklmnopqrstuvwxyz0123456789"$
    $length \leftarrow 12$
    $filename \leftarrow randomString(chars, length)$
    $requestMethod \leftarrow "POST"$
    $requestUrl \leftarrow "https://phototriage.cs.princeton.edu/score/score.php"$
    $requestData \leftarrow \{answer : randomLabelsJoinedString, filename : filename\}$
    $answer \leftarrow ajaxRequest(type : requestMethod, url : requestUrl, data : requestData, async : false)$
    return $parseFloat(answer.responseText.substring(59, 64))$
end
do
    for $i \leftarrow 0$ to $2584$ do
        $randomLabels[i] \leftarrow round(random(0, 1))$
        $fixedLabels[i] \leftarrow 0.5$
    end
    $evaluateBefore \leftarrow evaluate(randomLabels)$
    for $i \leftarrow 0$ to $2584$ do
        $randomLabels[i] \leftarrow 1 - randomLabels[i]$
        $evaluateNow \leftarrow evaluate(randomLabels)$
        if $evaluateNow > evaluateBefore$ then
            $fixedLabels[i] \leftarrow randomLabels[i]$
        end
        if $evaluateNow < evaluateBefore$ then
            $fixedLabels[i] \leftarrow 1 - randomLabels[i]$
        end
        $randomLabels[i] \leftarrow 1 - randomLabels[i]$
    end
    $evaluateAfter \leftarrow evaluate(randomLabels)$
while $evaluateAfter \neq 1$;
$print(fixedLabels.join("\backslash n") + "\backslash n")$
```

5.2 Evaluation of the suitability of faces for predicting the preference on the whole image

In order to evaluate the suitability of an image's faces for predicting the preference of the whole image, the aggregated face labels from the last section are compared to the labels of the whole images from Chang et al. [CYW$^+$16]. The evaluation is done by comparing the faces labels to the labels of Chang et al. [CYW$^+$16] on the whole image. Comparing labels instead of predictions is done because labels reflect the best, the predictions can ever get. Therefore, the face labels limit the performance that can be achieved with faces. The comparison is performed in two fashions:

(i) Per face pair: The label of each face pair is compared to the label of the whole image pair.

(ii) Per image pair: The face labels are averaged and then compared to the label of the whole image pair.

This evaluation was done on two subsets: All images containing faces (see Table 5.1a) and just images with human agreement $\geq 70\%$ (on Chang et al.'s label) containing faces (see Table 5.1b). The most significant numbers of those tables are the per image pair comparisons on images with human agreement $\geq 70\%$ that contain faces (Table 5.1b last line). Therefore, for the results, it can be expected in advance that the convolutional neural network would perform around 70% for images with human agreement $\geq 70\%$.

Set	Train	Valid	Test	All together
(i) Per face pair	63.9%	70.2%	61.2%	63.7%
(ii) Per image pair	64.6%	72.1%	62.5%	70.7%

(a) All images

Set	Train	Valid	Test	All together
(i) Per face pair	63.9%	70.2%	61.2%	63.7%
(ii) Per image pair	70.7%	72.1%	72.4%	71.0%

(b) Just images with human agreement $\geq 70\%$ after Chang et al. [CYW$^+$16]

Table 5.1: Agreement of the aggregated face labels from Section 4.4.2.1 with the image labels of Chang et al. [CYW$^+$16]. (i) Per face pair: The label of each face pair is compared to the label of the image it was extracted from. (ii) Per image pair: The average of the labels of face pairs extracted from the same image pair gets compared to the label of the image pair.

5.3 Face matching evaluation

The accuracy of the greedy face matching (see Section 4.3) is evaluated for the test set (see Table 5.2). A face match is correct if both face images of the face match belong to the same person. The face matching succeeded for 79.9% cases of the 354 matched face pairs. The prediction accuracy is similar for both subsets of correct and incorrect face matching, 69.5% on the subset of correctly matched faces, and 70.8% on the subset of incorrectly matched faces. The causes of the 72 cases of incorrectly matched faces are mainly (63.9%) due to failed or not possible face detection, as shown in Table 5.4.

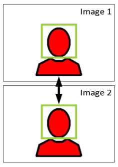
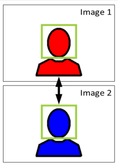
Face matching	Visualization	Number of cases	Percentage	Prediction accuracy
Correct		282	79.7%	69.5%
Incorrect		72	20.3%	70.8%

Table 5.2: Face matching evaluation on the 354 matched face pairs of the test set.

Since the prediction accuracy is similar for both subsets, this raises the question if the greedy face matching algorithm is needed. As shown in Table 5.3, the greedy face matching reduces the number of needed face pair predictions from 1047 to 354 and improves the accuracy from 66.7% to 69.8% on face pair level and from 70.4% to 70.8% on image pair level.

Prediction accuracy	Per face pair accuracy (Number of face pairs)	Per image pair accuracy (Number of image pairs)
All possible combinations of the detected faces	66.7% (1047)	70.4% (226)
With greedy face matching	69.8% (354)	70.8% (226)

Table 5.3: Prediction accuracy difference between using all possible combinations of the detected faces and the usage of the greedy face matching (see Section 4.3). The greedy face matching reduces the number of needed face pair predictions from 1047 to 354 and improves the accuracy from 66.7% to 69.8% on face pair level and from 70.4% to 70.8% on image pair level.

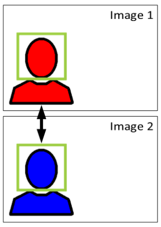
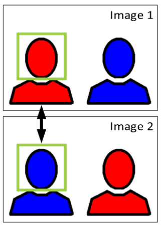
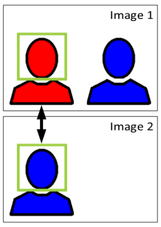
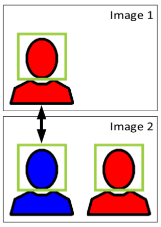
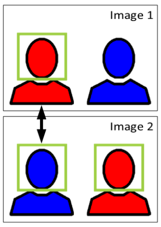
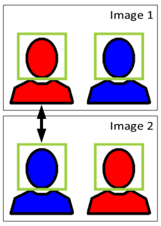
#	Visualization	Number of cases	Percentage	Description
1		22	30.6%	The matching face of each face from the wrongly matched face pair is not present in the other image.
2		23	31.9%	The matching face of each face from the wrongly matched face pair is not detected in the other image.
3		1	1.4%	The matching face of one face from the wrongly matched face pair is not present in the other image, and the other face is not detected in the other image.
4		8	11.1%	The matching face for one face from the wrongly matched face pair was detected, but the face still got matched with a face whose matching face was not present in the other image.
5		7	9.7%	The matching face for one face from the wrongly matched face pair was detected, but the face still got matched with a face whose matching face was not detected in the other image.
6		11	15.3%	The correct matching face of both faces of a matched face pair was detected, but the matching still failed.

Table 5.4: The causes of the 72 incorrect face matchings of the test set are mainly (63.9%) not possible (Case 1) or failed (Case 2 and 3) face detection. Since the order of Image 1 and Image 2 could be exchange and only the symmetric effect is of interest, the red person is either the matched face from Image 1 or Image 2, and the blue person is the matched face of the other image.

5.4 Results

In Table 5.5 the results are compared between the face model, the model of Chang et al. [CYW+16], and the combination of both. The results are presented for each set separately. Meanwhile, the most meaningful is the test set performance because it is not used during training. First of all, the images are filtered by ground truth $\geq 70\%$, as Chang et al. [CYW+16] suggests because it can not be expected from a convolution neural network model to perform well on images with low human agreement. There are between 57% and 60% of images per set that fulfill this criterion. Second, faces are searched in each image of an image pair. In between 15% and 21% (depending on the set) of image pairs contain humans, at least one person's face was detected in each image of the pair. If at least one face gets found in each image of an image pair, those pairwise combinations of faces with the lowest matching distance between all other candidates (see Section 4.3), get predicted from the face model. The results per set are split into two parts:

- Face images: Accuracy comparison of image pairs with faces detected, where predictions exist from both face model and the model of Chang et al. [CYW+16].
- All images: Accuracy comparison of all image pairs, where the accuracy of image pairs without detected faces is just based on the model of Chang et al. [CYW+16], since no faces exist to perform face model prediction.

First, the performance on face images is discussed. The reference is the performance of the model of Chang et al. [CYW+16] on face images. By comparison of Chang et al. [CYW+16] in the "face images" section and "all images" section for validation and test set, one can observe that Chang et al. [CYW+16] perform worse on face images than in general. The performance on the train set does not tell much, since those are the image the model was trained on. Also, the performance on the validation set has to be taken with caution because the amount of images with faces is only 61. The performance of the Face model on each single face pair, i.e., comparison of the face pair prediction with the label on the whole image pair, is between 65.3% (validation) and 69.8% (train) depending on the set (test set: 69.8%). By averaging the predictions of all face pairs of an image pair, the performance increases to 70.2% (train), 68.9 (validation), and 70.8% (test). The per face image pair accuracy is equal to the performance of Chang et al. [CYW+16] for the test set, better for the validation set and worse for the training set. This means that for images containing humans, the prediction on just faces achieves the same performance than the prediction on the whole image (test set). By linearly combining the prediction of Chang et al. [CYW+16] and the face model with equal weights, the performance can be further improved to 81.1% (train), 83.6% (validation), and 71.2% (test). This can be interpreted as the advantages of both models getting combined - Chang et al. [CYW+16] performing better on image where the context is important - and the face model performing better on images where the context hardly differs and the face expression making the difference. Due to the improvements through the combination

Filter:	**Percentage**	**#Images/Faces**
Number of images with ground truth >= 70.0%	56%	6776 of 12075
Number of image pairs with faces detected	17%	1119 of 6776
Just images with faces detected:	**Accuracy**	**#Images/Faces**
Chang et al. accuracy	79.8%	1119
Proposed method per face pair accuracy	69.7%	1679
Proposed method per image pair accuracy	70.2%	1119
Combined accuracy	81.1%	1119
All images:	**Accuracy**	**#Images/Faces**
Chang et al. accuracy	79.3%	6776
Combined accuracy	79.5%	6776

(a) Train set performance

Filter:	**Percentage**	**#Images/Faces**
Number of images with ground truth >= 70.0%	60%	292 of 483
Number of image pairs with faces detected	21%	61 of 292
Just images with faces detected:	**Accuracy**	**#Images/Faces**
Chang et al. accuracy	62.3%	61
Proposed method per face pair accuracy	65.3%	101
Proposed method per image pair accuracy	68.9%	61
Combined accuracy	83.6%	61
All images:	**Accuracy**	**#Images/Faces**
Chang et al. accuracy	66.8%	292
Combined accuracy	71.2%	292

(b) Validation set performance

Filter:	**Percentage**	**#Images/Faces**
Number of images with ground truth >= 70.0%	57%	1469 of 2585
Number of image pairs with faces detected	15%	226 of 1469
Just images with faces detected:	**Accuracy**	**#Images/Faces**
Chang et al. accuracy	70.8%	226
Proposed method per face pair accuracy	69.8%	354
Proposed method per image pair accuracy	70.8%	226
Combined accuracy	71.2%	226
All images:	**Accuracy**	**#Images/Faces**
Chang et al. accuracy	71.8%	1469
Combined accuracy	71.9%	1469

(c) Test set performance

Table 5.5: Results comparison between the face model developed in this thesis and the model of Chang et al. [CYW+16]. Combined results are linear combinations with equal weights of the predictions of both models.

of the predictions of both models on images with humans, the performance on all images also gets increased. However, the improvement is less noticeable, since the majority of images do not contain humans, and therefore, just a prediction through the models of Chang et al. exists.

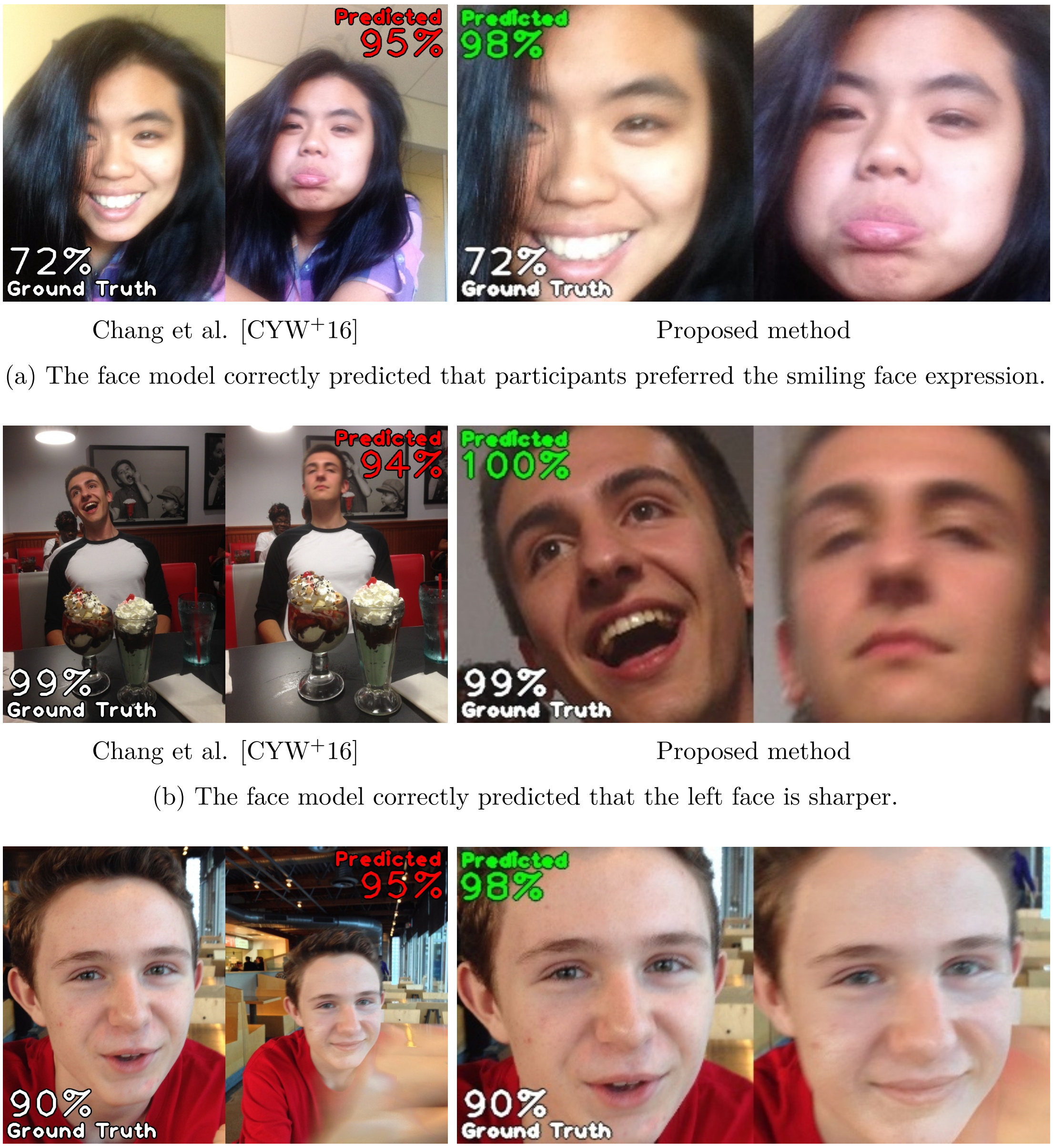
5.5 Evaluation

Since numbers alone give to little insight, the Best-6 (Figure 5.1, 5.2) and Worst-6 cases (Figure 5.3, 5.4) are presented and discussed. Best cases are those where the face model performed significantly better than the model of Chang et al. [CYW$^+$16] and predicted in contrast to Chang et al. [CYW$^+$16] the correct preference. In contrast, the worst cases are those where the face model failed to perform better than Chang et al. [CYW$^+$16] and predicted the incorrect image preference while Chang et al. [CYW$^+$16] predicted the correct preference. Those six examples for best-case and worst-case were selected because they have the highest absolute score difference between the face model and the model of Chang et al. [CYW$^+$16]. In other words, both models predict contrary images with high confidence to be preferred by users.

As can be observed from the best-cases images, the face model performed well, if two requirements are given: There is sufficient difference between the facial expressions and the context is similar in both images, since the face model cannot detect contextual differences. In those cases, the model of Chang et al. [CYW$^+$16] fails to detect that the face is the major difference between both images.

In contrast, the face model failures are mainly due to incorrect face matching caused by failed face detection, e.g. the face of the matching person in the other image could not be detected (see Figure 5.3c, 5.4a, and 5.4c) or is not present at all (see Figure 5.3b). In cases where the background (see Figure 5.3a) or context is important, e.g., singing (see Figure 5.4b), and the faces do not differ much, the face model does also fail.

In this chapter, at first, the acquisition of the test labels for the evaluation was presented. Then the suitability of faces for predicting the preference on the whole image was analyzed. The crowd-scored face pairs match on average in 72.4% cases (test set) with the labels on the whole image pairs by Chang et al. [CYW$^+$16] (for images with $\geq 70\%$ human agreement), which limits the possible performance of face model. Furthermore, the performance of face matching and its influence on the prediction results was evaluated. The main benefit of using face matching is the reduced amount of required pairwise predictions. Lastly, the prediction results were quantitatively and qualitatively evaluated by measuring the accuracy of the predictions on the dataset of Chang et al. [CYW$^+$16] and by showing and interpreting best-case and worst-case prediction examples. Since facial features (proposed approach) performed as good as all features (Chang et al. [CYW$^+$16]), this highlights the importance of faces for human preference prediction.



Chang et al. [CYW+16] Proposed method

(a) The face model correctly predicted that participants preferred the smiling face expression.

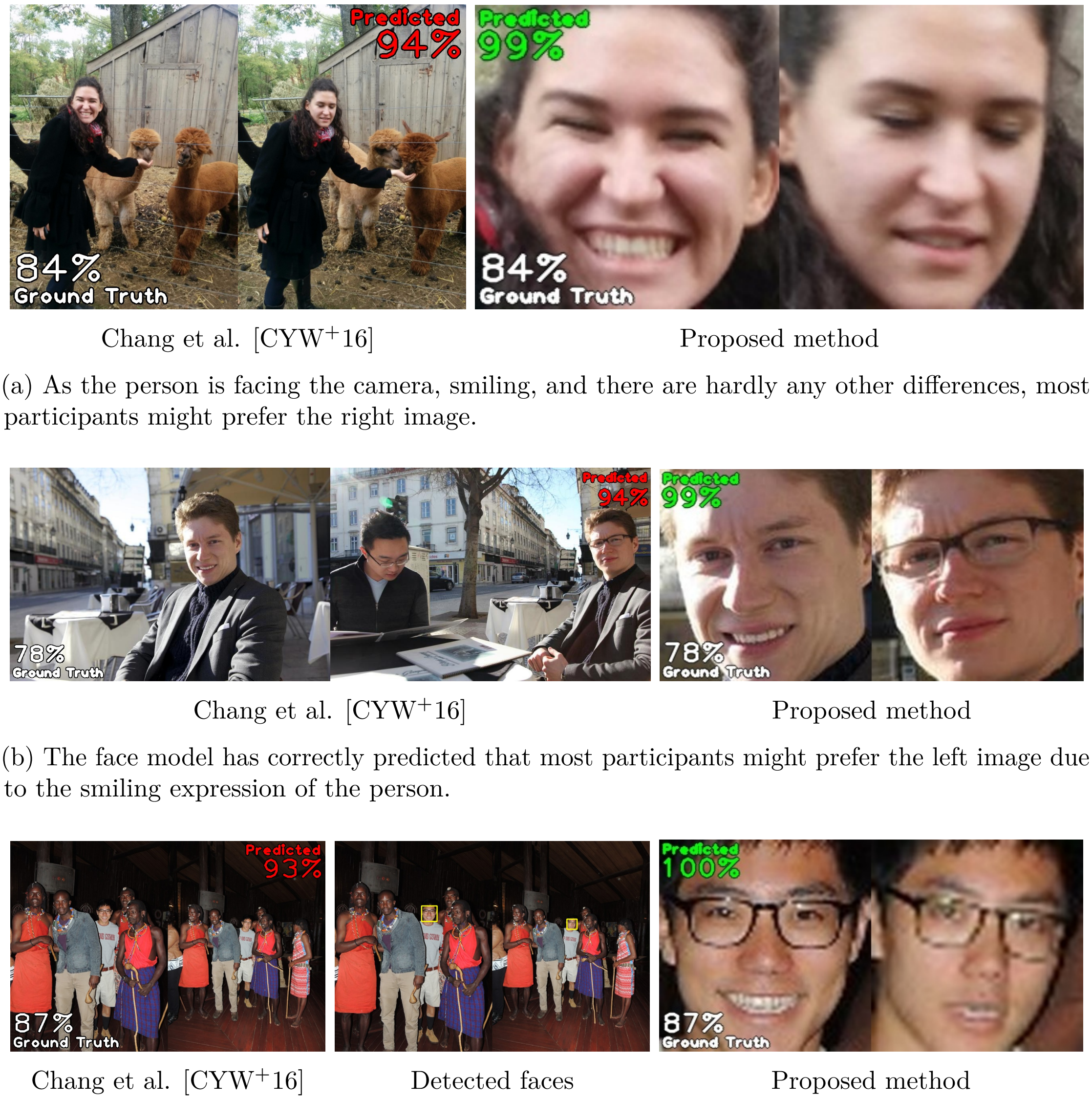
Chang et al. [CYW+16] Proposed method

(b) The face model correctly predicted that the left face is sharper.

Chang et al. [CYW+16] Proposed method

(c) The face model correctly predicted that the left face is sharper, and there is no moving hand in the image.

Figure 5.1: Best cases (1.-3.) of the proposed method: Comparison of the predictions of Chang et al. [CYW+16] and the proposed method on those image pairs where the predictions of both methods are contrary and have the highest prediction confidence. Predictions on the same image (of an image pair) as the ground truth are correct (highlighted in green), predictions on different images are incorrect (highlighted in red). The order of the images is flipped such that the left image is always the ground truth image. The ground truth for both methods is the crowd-scored label of Chang et al. [CYW+16] on the whole image pair. The percentages show the prediction confidence of each method on its predicted image and the participant's agreement on the ground truth.

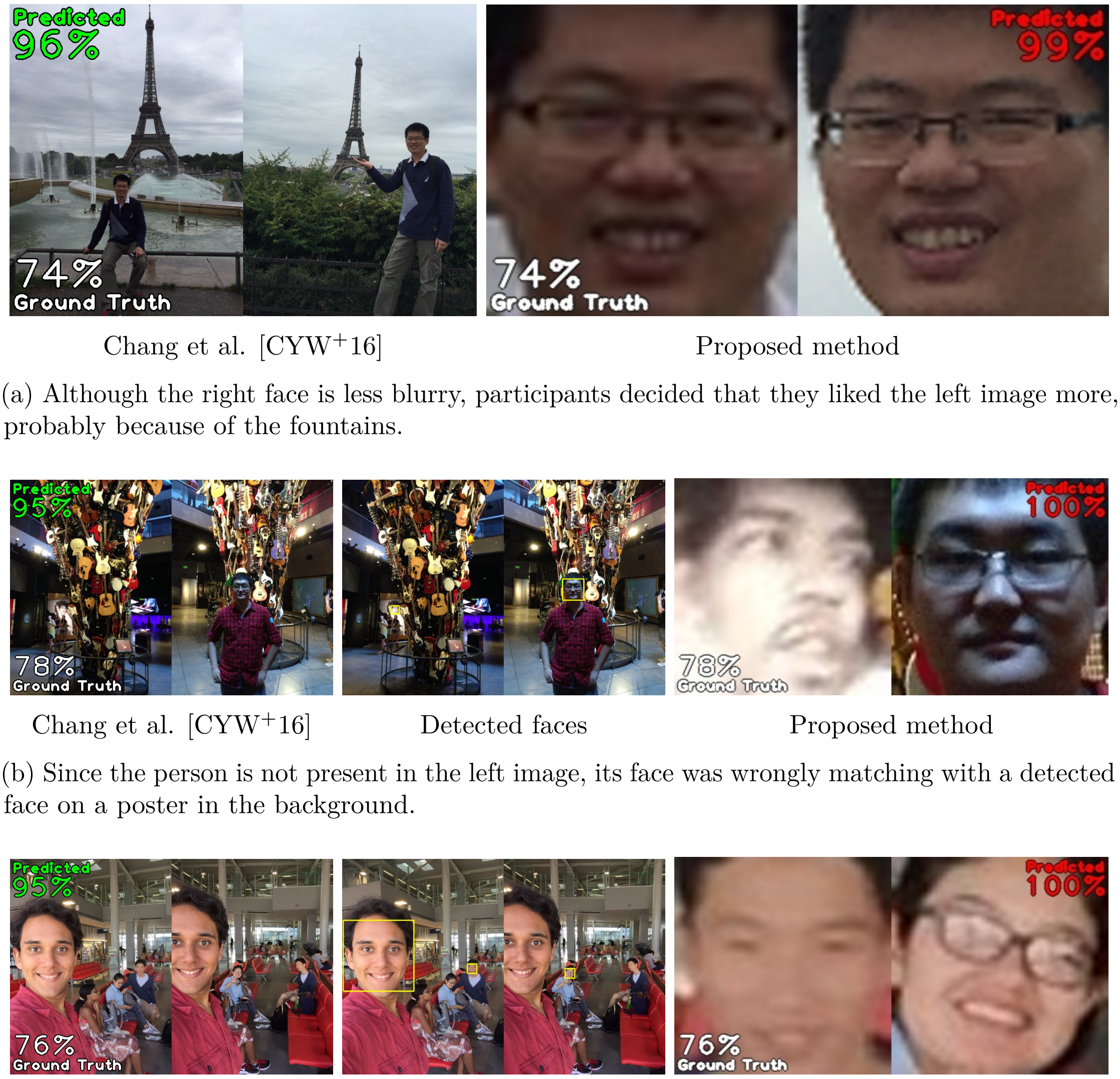


(a) As the person is facing the camera, smiling, and there are hardly any other differences, most participants might prefer the right image.

(b) The face model has correctly predicted that most participants might prefer the left image due to the smiling expression of the person.

(c) The participants might have preferred that the persons are located closer to the camera in the left image, which also results in sharper faces. The face detector has just detected the only white person, which indicates that the MTCNN face detector [ZZLQ16] has a racial bias.

Figure 5.2: Best cases (4.-6.) of the proposed method: Comparison of the predictions of Chang et al. [CYW+16] and the proposed method on those image pairs where the predictions of both methods are contrary and have the highest prediction confidence. Predictions on the same image (of an image pair) as the ground truth are correct (highlighted in green), predictions on different images are incorrect (highlighted in red). The order of the images is flipped such that the left image is always the ground truth image. The ground truth for both methods is the crowd-scored label of Chang et al. [CYW+16] on the whole image pair. The percentages show the prediction confidence of each method on its predicted image and the participant's agreement on the ground truth.



Chang et al. [CYW+16] Proposed method

(a) Although the right face is less blurry, participants decided that they liked the left image more, probably because of the fountains.

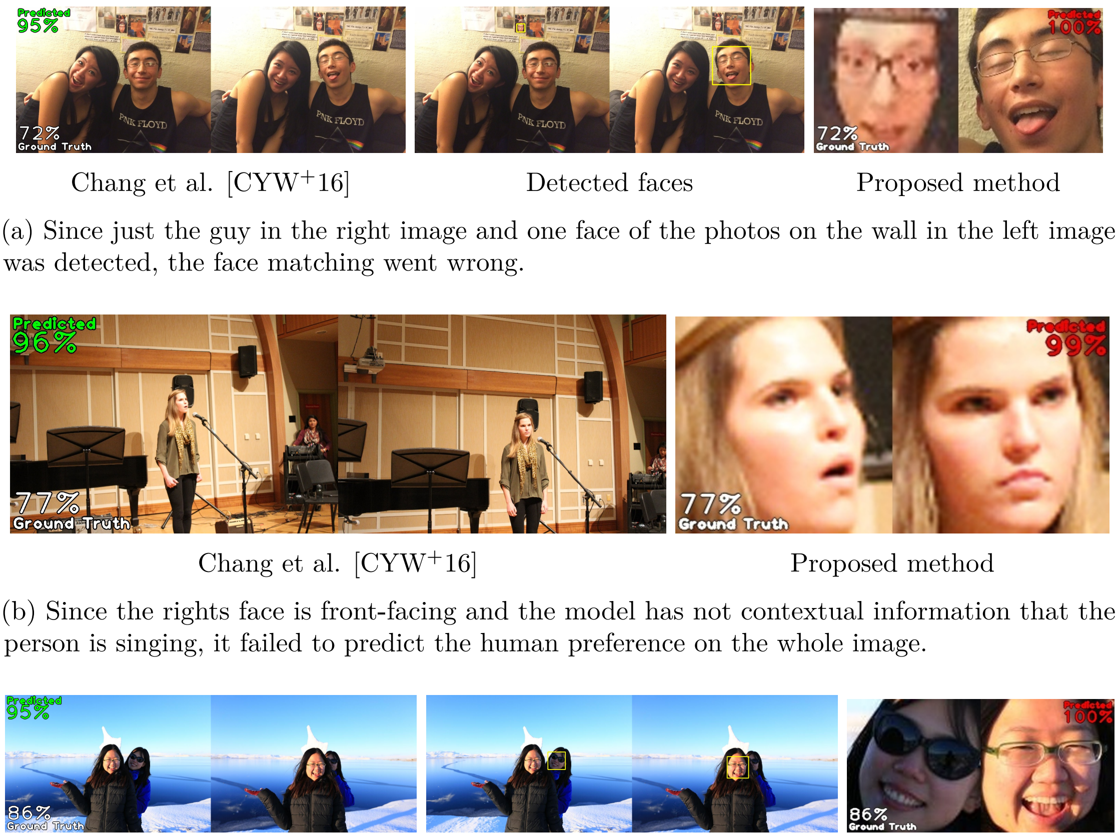
Chang et al. [CYW+16] Detected faces Proposed method

(b) Since the person is not present in the left image, its face was wrongly matching with a detected face on a poster in the background.

Chang et al. [CYW+16] Detected faces Proposed method

(c) Because the face of the person in the front is just partly visible in the right image, it could not be detected, and the face prediction was performed with the wrongly matched faces in the background.

Figure 5.3: Worst cases (1.-3.) of the proposed method: Comparison of the predictions of Chang et al. [CYW+16] and the proposed method on those image pairs where the predictions of both methods are contrary and have the highest prediction confidence. Predictions on the same image (of an image pair) as the ground truth are correct (highlighted in green), predictions on different images are incorrect (highlighted in red). The order of the images is flipped such that the left image is always the ground truth image. The ground truth for both methods is the crowd-scored label of Chang et al. [CYW+16] on the whole image pair. The percentages show the prediction confidence of each method on its predicted image and the participant's agreement on the ground truth.



Chang et al. [CYW+16] Detected faces Proposed method

(a) Since just the guy in the right image and one face of the photos on the wall in the left image was detected, the face matching went wrong.

Chang et al. [CYW+16] Proposed method

(b) Since the rights face is front-facing and the model has not contextual information that the person is singing, it failed to predict the human preference on the whole image.

Chang et al. [CYW+16] Detected faces Proposed method

(c) The face of the person in the front could not be detected, so the prediction was performed on two wrong matched faces.

Figure 5.4: Worst cases (4.-6.) of the proposed method: Comparison of the predictions of Chang et al. [CYW+16] and the proposed method on those image pairs where the predictions of both methods are contrary and have the highest prediction confidence. Predictions on the same image (of an image pair) as the ground truth are correct (highlighted in green), predictions on different images are incorrect (highlighted in red). The order of the images is flipped such that the left image is always the ground truth image. The ground truth for both methods is the crowd-scored label of Chang et al. [CYW+16] on the whole image pair. The percentages show the prediction confidence of each method on its predicted image and the participant's agreement on the ground truth.

CHAPTER 6

Conclusion

The main contributions of this thesis are the construction and crowd-scored labeling of a face preference dataset from the images of the dataset of Chang et al. [CYW$^+$16], the evaluation of the suitability of an image's faces for predicting human preference on the whole image, and the training and evaluation of a face preference model. At first, the architecture and prediction results of the baseline approach of Chang et al. [CYW$^+$16] were reconstructed. Then the faces were extracted from two different choices of datasets using the MTCNN face detector [ZZLQ16] and matched using Facenet [SKP15], such that a face pair consists of a face from each image in an image pair. Prior to using extracted faces from the dataset of Chang et al. [CYW$^+$16] for training, the portrait dataset of [ZAE$^+$14] was tried, which already has labels for attractiveness. Unfortunately, the trained model did perform poorly ($< 60\%$ accuracy) when applied to the dataset of Chang et al. [CYW$^+$16]. However, using the dataset of Chang et al. [CYW$^+$16] also for training as the second approach worked better. The extracted and matched face pairs were crowd-scored for human preference. The obtained face pair scores correlate in 72.4% cases (test set) with the labels on the whole image pairs of Chang et al. [CYW$^+$16] for images with at least 70% human agreement on the label of Chang et al. [CYW$^+$16]. With the collected face labels as ground truth, a Siamese face model was trained and evaluated.

For the reason that the face model is trained with labels on the face pairs but evaluated on the labels of the whole image, the achievable accuracy is limited. In an ideal world, where the face models would perform correctly on all face pairs regarding the face pair labels, the prediction accuracy for the whole image would still be limited to the correlation between the labels on the face pairs and the whole image pairs. This is since facial features are just one of the features that affect human preference on image pairs. However, the importance of facial features likely is dependent on the type of photo, for instance, in a family portrait, the impression of faces might be the only important feature to judge for the best one.

By predicting human preference on image pairs with just the extracted faces and ignoring all other image regions, the approach of this thesis predicts human preference on images containing humans equally well as Chang et al. [CYW$^+$16] (70.8%). As just using facial features performs as good as using all features, this highlights the importance of faces for human preference prediction. Unfortunately, facial preference prediction cannot replace image preference prediction on whole images in all cases, since on images containing no humans, the approach of this thesis cannot be applied. By combining the prediction results of the face model with the predictions of Chang et al. [CYW$^+$16], the results of Chang et al. [CYW$^+$16] could even be further improved to 71.9% owed to the combination of the strengths of both approaches.

As this approach uses face detection and face matching as preprocessing, steps that also have a certain error rate. Therefore, future work needs to be done to get a more robust preprocessing for the face model. As shown in the examples (Figure 5.3b, 5.3c, 5.4a, 5.4b), in some cases the prediction was preformed on wrongly matched faces because the correct matching faces were not detected. This indicates that an enhanced face detection algorithm could improve the proposed method as more detected faces would lower the impact of, for example, detected faces on posters in the background. The main benefit of using face matching is that it reduces the amount of face predictions needed as just faces of the same person are compared. In two group photos with n people, the face matching reduces the amount of needed predictions from n^2 to n. However, on average correct face matching does not significantly (0.4%) improve the prediction accuracy.

A benefit of the proposed method is that it gives more feedback about its decision and is thus more explainable. For example, a possible feedback could be that the faces of two of the three persons in an image pair look better in the first image, and therefore the first image is better in total. In contrast, Chang et al. [CYW$^+$16] consider the whole image, but it remains unclear on what its decision is based upon. In combination with face recognition, the proposed approach could be extended to prefer photos where the person of interest looks better. Due to the proven importance of faces for human preference prediction in this thesis, the work of this thesis might guide future work in areas where face detection is already done for other purposes. For instance, on many mobile devices, face detection is already used in the viewfinder to achieve sharp faces, and near-duplicate image detection is used to help users reduce memory occupation. Therefore, grouping similar images and detect faces is not an additional work on mobile devices that increases the computational cost for preference prediction. However, unfortunately, the VGG-16 convolutional neural network used in this thesis is still too computational and memory intensive for nowadays mobile devices. That smaller convoluation neural networks can adapted to mobile devices was shown by the previous work of Wang et al. [WVSZ20]. The ability of smaller networks to learn face differences with a similar accuracy of $\geq 74\%$ (just 2.1% less than VGG-16) was shown in this thesis exemplary with SqueezeNet. Therefore, a wide application possibility of the findings in this thesis is seen for "Selfies" with the front-facing camera on mobile devices, or more generally, in all cases where the facial appearance of one person or a group is of interest.

Appendix

The corrected errors in the repository of Zhijian Liu[1]:

- In src/data.py line 26

  ```
  image = np.pad(image, ((0, 224 - width), (0, 224 - height),
      (0, 0)), mode = "constant", constant_values = 0)
  ```

  should be "padded with the mean pixel color in the training set" instead of black (zero) padding, according to the paper [CYW+16]. However, it is not mentioned if it should be padded on both sides, such that the image is always located in the middle, or just on the right/bottom side.

- In src/data.py line 30 `if score < 0.5:` should be `if score >= 0.5:` , else the network would learn which image is worse.

- In srt/train.py line 39 `optimizer = SGD(lr = 0.001, momentum = 0.9)` should be also added a `decay=0.0005`, as it is mentioned in the paper. Stochastic gradient descent (SGD) is a method to update the weight using the gradient estimated from a (usually) small subset of training examples. The learning rate (lr) controls that the weights are just updated by a small portion of the negative gradient [Wu17], the momentum accelerates SGD in the relevant direction and dampens oscillations, and decay is the factor that sets the influence of the L2 regulation [2].

- Although L2 regulation is used, the model started to overfit from the fourth epoch. To get rid of overfitting, the patience for the ReduceLROnPlateau callback was reduced from 2 to 0, the factor reduced from 0.5 to 0.1 and an EarlyStopping callback was added with patience=2, min_delta=0 and restore_best_weights=True:

  ```
  ReduceLROnPlateau(monitor="val_accuracy", factor=0.1,
     patience=0),
  ```

[1] https://github.com/zhijian-liu/auto-triage accessed on 11.2.2020

[2] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/SGD accessed of 11.3.2020

```
EarlyStopping(monitor="val_accuracy", min_delta=0, patience
    =2, mode="auto", baseline=None, restore_best_weights=
    True)
```

- As described in the paper, dropouts are removed in the 16-layer VGG-Subnets as well as in the multi-layer perceptron. The reimplementation uses the 16-layer VGGNet from keras.applications.vgg16, which already comes without dropout layers[3]. However, in the multi-layer perceptron one dropout layer is falsely used, which was removed (src/models.py line 76, as it was before the dropout layer was removed):

```
hidden1 = Dropout(0.5)(Dense(128, activation = FLAGS.
    activation)(feature))
```

[3] https://forums.fast.ai/t/difference-b-w-jeremys-vgg16-and-keras-built-in-vgg-16/881 accessed on 17.2.2020

Bibliography

[BBD+12] Alexander C. Berg, Tamara L. Berg, Hal Daume, Jesse Dodge, Amit Goyal, Xufeng Han, Alyssa Mensch, Margaret Mitchell, Aneesh Sood, Karl Stratos, and Kota Yamaguchi. Understanding and predicting importance in images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3562–3569, 2012.

[BGL+93] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Sackinger, and Roopak Shah. Signature Verification using a “Siamese” Time Delay Neural Network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7, 1993.

[BT52] Ralph A. Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: The Method of Paired Comparisons. *Biometrika*, 39(3/4):324, 1952.

[CL08] Wei T. Chu and Chia H. Lin. Automatic Selection of Representative Photo and Smart Thumbnailing Using Near-Duplicate Detection. In *ACM International Conference on Multimedia*, MM ’08, pages 829–832, New York, NY, USA, 2008. ACM.

[CQL+07] Zhe Cao, Tao Qin, Tie Y. Liu, Ming F. Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *ACM International Conference Proceeding Series*, International Conference Proceeding Series, pages 129–136, New York, NY, USA, 2007. ACM.

[CSN+15] Andrea Ceroni, Vassilios Solachidis, Claudia Niederée, Olga Papadopoulou, Nattiya Kanhabua, and Vasileios Mezaris. To keep or not to keep: An expectation-oriented photo selection method for personal photo collections. In *ACM International Conference on Multimedia Retrieval*, ICMR ’15, pages 187–194, New York, NY, USA, 2015. ACM.

[CSWS17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 7291–7299. IEEE, 2017.

[CTCG95] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim R. Graham. Active Shape Models - Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[CYW+16] Huiwen Chang, Fisher Yu, Jue Wang, Douglas Ashley, and Adam Finkelstein. Automatic triage for a photo series. *ACM Transactions on Graphics*, 35(4):1–10, 2016.

[DJLW06] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Studying Aesthetics in Photographic Images Using a Computational Approach. In *European Conference on Computer Vision*, pages 288–301, Berlin, Heidelberg, 2006. Springer.

[DOB11] Sagnik Dhar, Vicente Ordonez, and Tamara L. Berg. High level describable attributes for predicting aesthetics and interestingness. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1657–1664, 2011.

[DPK+15] Rachit Dubey, Joshua Peterson, Aditya Khosla, Ming H. Yang, and Bernard Ghanem. What makes an object memorable? In *IEEE International Conference on Computer Vision*, pages 1089–1097, 2015.

[FHX+14] Yanwei Fu, Timothy M. Hospedales, Tao Xiang, Shaogang Gong, and Yuan Yao. Interestingness prediction by robust learning to rank. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *European Conference on Computer Vision*, pages 488–503, Cham, 2014. Springer.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, volume 29. MIT Press, 2016.

[GGR+13] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, Fabian Nater, and Luc Van Gool. The Interestingness of Images. In *IEEE International Conference on Computer Vision*, pages 1633–1640, 2013.

[GYX+11] Bo Geng, Linjun Yang, Chao Xu, Xian-Sheng Hua, and Shipeng Li. The Role of Attractiveness in Web Image Search. In *ACM International Conference on Multimedia*, MM '11, pages 63–72, New York, NY, USA, 2011. ACM.

[IPTO11] Phillip Isola, Devi Parikh, Antonio Torralba, and Aude Oliva. Understanding the Intrinsic Memorability of Images. In J Shawe-Taylor, R S Zemel, P L Bartlett, F Pereira, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2429–2437. Curran Associates, Inc., 2011.

[IXTO11] Phillip Isola, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. What makes an image memorable? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 145–152, 2011.

[JP15] Mainak Jas and Devi Parikh. Image Specificity. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2727–2736, 2015.

[JSS16] Bin Jin, Maria V. O. Segovia, and Sabine Süsstrunk. Image aesthetic predictors based on weighted CNNs. In *IEEE International Conference on Image Processing*, pages 2291–2295, 2016.

[JV03] Michael Jones and Paul Viola. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 3(14):2, 2003.

[KDH14] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What Makes an Image Popular? In *International Conference on World Wide Web*, WWW '14, pages 867–876, New York, NY, USA, 2014. ACM.

[KRTO15] Aditya Khosla, Akhil S. Raju, Antonio Torralba, and Aude Oliva. Understanding and Predicting Image Memorability at a Large Scale. In *IEEE International Conference on Computer Vision*, pages 2390–2398, 2015.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F Pereira, C J C Burges, L Bottou, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[KSL+16] Shu Kong, Xiaohui Shen, Zhe Lin, Radomir Mech, and Charless Fowlkes. Photo Aesthetics Ranking Network with Attributes and Content Adaptation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision*, pages 662–679, Cham, 2016. Springer International Publishing.

[Kuh55] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[LGLC10] Congcong Li, Andrew Gallagher, Alexander C. Loui, and Tsuhan Chen. Aesthetic quality assessment of consumer photos with faces. In *IEEE International Conference on Image Processing*, pages 3221–3224. IEEE, 2010.

[LLC10] Congcong Li, Alexander C. Loui, and Tsuhan Chen. Towards aesthetics: A photo quality assessment and photo selection system. In *ACM International Conference on Multimedia*, MM '10, pages 827–830, New York, NY, USA, 2010. ACM.

[LLJ+14] Xin Lu, Zhe Lin, Hailin Jin, Jianchao Yang, and James Z. Wang. Rapid: Rating pictorial aesthetics using deep learning. In *ACM International Conference on Multimedia*, pages 457–466, New York, New York, USA, 2014. ACM Press.

[Low04] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[MMP12] Naila Murray, Luca Marchesotti, and Florent Perronnin. AVA: A Large-Scale Database for Aesthetic Visual Analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2408–2415, 2012.

[ON15] Keiron O'Shea and Ryan Nash. An Introduction to Convolutional Neural Networks. *Computing Research Repository*, 2015.

[PM92] Sankar K. Pal and Sushmita Mitra. Multilayer perceptron, fuzzy sets, classifiaction. *IEEE Transactions on Neural Networks*, 3(5):683–697, 1992.

[RDS$^+$15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:815–823, 2015.

[SM18] Zahra Riahi Samani and Mohsen Ebrahimi Moghaddam. A multi-criteria context-sensitive approach for social image collection summarization. *Sādhanā*, 43(9):143, 2018.

[SMJ11] Pinaki Sinha, Sharad Mehrotra, and Ramesh Jain. Summarization of Personal Photologs Using Multidimensional Content and Context. In *ACM International Conference on Multimedia Retrieval*, ICMR '11, New York, NY, USA, 2011. ACM.

[SS02] Daniel Scharstein and Richard Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[SSS07] Ian Simon, Noah Snavely, and Steven M. Seitz. Scene Summarization for Online Image Collections. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. pages 1–14. Computational and Biological Learning Society, 2015.

[TG11] Kristi Tsukida and Maya R. Gupta. How to Analyze Paired Comparison Data. Technical Report. Technical Report 206, Department of Electrical Engineering University of Washington, 2011.

[TM18] Hossein Talebi and Peyman Milanfar. NIMA: Neural Image Assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, 2018.

[WSS14] Tina C. Walber, Ansgar Scherp, and Steffen Staab. Smart Photo Selection: Interpret Gaze as Personal Interest. In *SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2065–2074, New York, NY, USA, 2014. ACM.

[Wu17] Jianxin Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5:23, 2017.

[WVSZ20] Baoyuan Wang, Noranart Vesdapunt, Utkarsh Sinha, and Lei Zhang. Real-Time Burst Photo Selection Using a Light-Head Adversarial Network. *IEEE Transactions on Image Processing*, 29:3065–3077, 2020.

[YHBO10] Che H. Yeh, Yuan C. Ho, Brian A. Barsky, and Ming Ouhyoung. Personalized photograph ranking and selection system. In *ACM International Conference on Multimedia*, MM '10, pages 211–220, New York, NY, USA, 2010. ACM.

[YK16] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *International Conference on Learning Representations*, 2016.

[ZAE+14] Jun Y. Zhu, Aseem Agarwala, Alexei A. Efros, Eli Shechtman, and Jue Wang. Mirror Mirror: Crowdsourcing Better Portraits. *ACM Transactions on Graphics*, 33(6):234:1–234:12, 2014.

[ZZLQ16] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.