PH.D. SCHOOL
UNIVERSITY OF MILANO-BICOCCA

Department of Informatics, Systems and Communication
Ph.D. program in Computer Science - XXX Cycle

Learning quality, aesthetics, and facial attributes for image annotation

Ph.D. Dissertation of: Luigi Celona

Supervisor: Prof. Raimondo Schettini
Co-Supervisor: Dr. Paolo Napoletano
Tutor: Prof. Giancarlo Mauri
Ph.D. Coordinator: Prof. Stefania Bandini

Academic Year 2016-2017

Acknowledgements

I am very grateful to Prof. Raimondo Schettini, my supervisor, for believing in me and guiding me throughout this fantastic journey.

I would like to thank Dr. Paolo Napoletano, my co-supervisor, for his friendship with me and for his precious suggestions.

In addition, I would like to thank Dr. Simone Bianco for supporting me and providing useful insights.

I can not forget my friends in the Imaging and Vision Laboratory (IVL), I am happy and honored of all the years spent together. Thanks for all the time spent joking or exchanging knowledge and skills.

Thanks to all my family. The merits of this new goal are first of all because you have made me the person I am today, you have given me an excellent education, continue to be a fundamental guidance, and support me every day.

Thanks to You, my love, for always being next to me.

Abstract

Every day, a large number of digital images are produced by users of social networks, smartphone users, photography professionals, etc. This caused a problem in the management, organization, indexing, and recovery of digital images. In order to ease this problem, several methods have been introduced in the literature to catalog images automatically. These methods are designed to associate images with one or more keywords belonging to a predefined dictionary or to associate images with visual attributes such as, for example, quality, aesthetics, sentiment, memorability, interestingness, and complexity, etc.

This thesis investigates the use of deep convolutional neural network for automatic estimation of image quality and image aesthetics. In the last few years, several methods for automatic image quality assessment have been proposed. Most of them have been designed to deal with synthetically distorted images, which by definition do not truly model distortions afflicting real-world images. In this thesis a method for the automatic quality assessment of authentically distorted images is investigated. It shows better performances than state-of-the-art methods both on synthetically and authentically distorted images datasets.

Differently from the image quality, which characterizes the perceived quality of the image signal, aesthetics depicts perceived beauty. As first step, the problem of aesthetic quality assessment of real-life general content images has been investigated. The proposed solution outperformed state-of-the-art methods on the largest publicly available dataset.

Given that one of the most popular visual contents is the face (e.g. on social networks for photo sharing), aesthetics assessment is, therefore, further investigated on the specific case of portrait images. To this end, in this thesis an algorithm involving the combination of the previously investigated visual attributes (i.e. quality and aesthetics of general content images) and the facial attributes (i.e. smiling, hair style, makeup) description is proposed. Facial attributes description is achieved thanks to two proposed methods. The first algorithm is a robust smile detector (it represents an important visual feature for portrait aesthetics), the second is a multiple-task

model designed in order to simultaneously estimate soft biometrics and attributes such as hair colors and styles, types of beards. While the first algorithm outperforms state-of-the-art methods (also respect to highly distorted images), the multi-task model demonstrates comparable performance. Experimental results for the portrait image aesthetic assessment thanks to the use of the proposed algorithm show promising performance on three standard datasets.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Why automatic tagging?

Over the past decade the amount of images being captured and shared has grown enormously. There are several factors behind this remarkable trend. The first is the diffusion of high-resolution digital cameras often integrated into mobile phones. Digital cameras enable people to capture, edit, store and share images easily in comparison to the old film cameras. Furthermore, photo sharing platforms, such as Instagram and Flickr, and of social networks or instant messaging applications, e.g. Facebook, WhatsApp and Snapchat, are gaining popularity. Figure 1.1 shows the increasing growth of shared images per day on the most popular platforms in the last decade.

Images and video sequences make up about the eighty percent of all enterprise and public unstructured big data. Unstructured data are not easily searchable, so in order to index, organize, and retrieve them it is necessary to assign meta-data able to describe the resource content. Digital photo meta-data is expressed in the form of keywords, also known as *tags*, or captions that can describe the visual and semantic entities within the photos. Several softwares give the possibility to manually add tags to photos, like Photoshop and Picasa. However, due to the growing amount of data, manual annotation of images has become infeasible because it is time-consuming, subjective, non-scalable, and non-uniform in terms of vocabulary. Automatic image annotation, shortly *auto-tagging*, systems are introduced to overcome these limitations. These systems are able to automatically assign a set of tags from a dictionary in order to describe the image content without human intervention, answering to the question

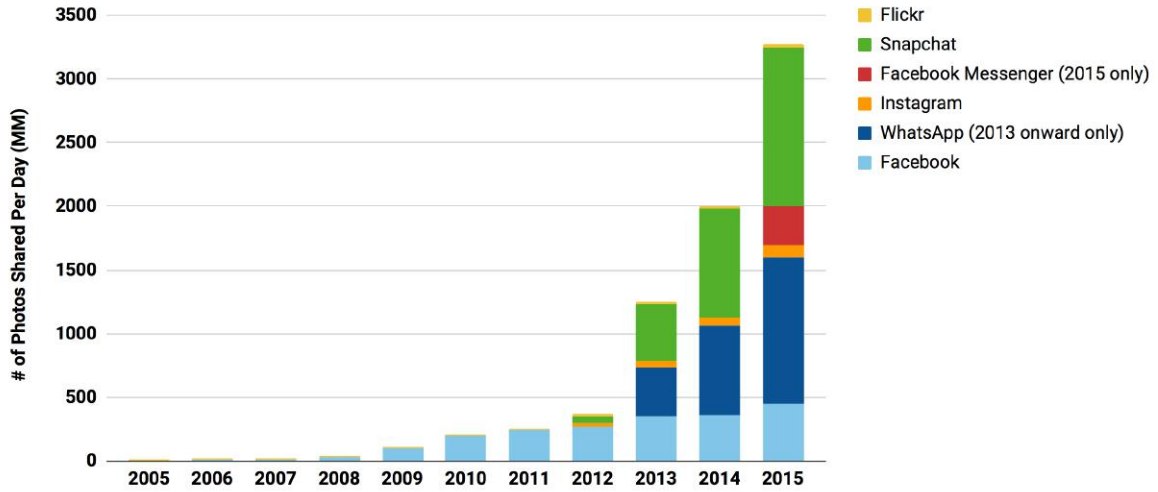


Fig. 1.1 Daily number of shared photos on the most popular platforms between 2005 and 2015. Chart derived from published report by KPCB on Internet Trends.

Fig. 1.2 Examples of images annotated with tag describing semantic content.

What does the image contain? Google Cloud Vision[1], Clarifai[2], Imagga[3], and Irista[4] are examples of automatic large-scale image annotation systems able to tag images pretty carefully.

1.2 The role of visual attributes

Content-based image categorization using accurate and relevant tags provides a good but not exhaustive description. For example, given the two images in Figure 1.2, a state-of-the-art auto-tagging system would certainly assign the "airplane" tag. However, this annotation solely characterizes generic semantic of images and doesn't provide any information about properties of the image content, that can be useful, for example, to measure the appeal of images and decide which one to use for a photo album. In order

[1]https://cloud.google.com/vision
[2]https://www.clarifai.com
[3]https://imagga.com
[4]https://www.irista.com/

(a) (b)

Fig. 1.3 Sample images annotated considering image quality. Yellow boxes contain the result of a common automatic image tagging system: "airplane". Instead green boxes provide a label referring to the quality of the two images, respectively low (a) and high (b).

to enrich image description, complementary interesting offshoots involving feedback, personalization, and emotions might be taken into consideration. These aspects are depicted through the analysis of *visual attributes* that capture appearance and properties of the image content such as quality, aesthetics, sentiment, memorability, interestingness, and complexity. Among them, three visual attributes mainly characterize palatability of images, namely: quality, aesthetics, and sentiment.

Visual quality refers to the quantification of low-level perceptual degradation of a visual stimulus. Specifically, this consists in evaluating whether the image quality is high or low due to the presence or absence of distortions. Thus it tries to answer the following question: *Does the image look qualitatively good?* Figure 1.3 shows the same two airplanes labeled with the corresponding semantic tags as in Figure 1.2, but also two tags describing image quality have been introduced: while the airplane in Figure 1.3a has a "low-quality" tag (lens flare artifact, underexposed), the airplane in Figure 1.3b is labeled as "high-quality" (sharp, proper light).

Visual aesthetics regards the perceived beauty of visual stimulus [65]. Aesthetic evaluation is an extremely difficult problem because is highly subjective: its a combination of genetic predisposition, cultural assimilation, and unique individual experience. Additionally, aesthetics is "naturally" blind: given an image to be evaluated, there doesn't exist a corresponding image with "perfect" aesthetics. Of course, this does not nullify the possibility of comparing the aesthetic appeal of two images, instead, it makes visual aesthetics assessment more challenging and interesting. The goal of algorithmic aesthetic assessment is to predict aesthetic scores considering a series of features such as exposure (luminance distribution), contrast, colorfulness, color saturation, rule of thirds, depth of field, trying to answer the question *Is the image aesthetically good?* In Figure 1.4 two "hamburger" images can be distinguished between "low-level" aesthetics,

(a) (b)

Fig. 1.4 Sample images annotated considering image aesthetics. Yellow boxes contain the result of a common automatic image tagging system: "hamburger". Instead, blue boxes provide a label referring to the perceived aesthetics quality for the two images, respectively low (a) and high (b).

Figure 1.4a, (low contrast, bad composition), and "high-level" aesthetics, Figure 1.4b, (high contrast, color saturation).

Visual sentiment analysis consists in the extraction of the affective content information from visual stimuli. Algorithms for sentiment analysis extract the amount and type of affect that is *expected* to be evoked in the majority of the observers by the perceived content. Specifically, these algorithms try to capture information regarding feelings, emotions, and moods evoked by an image or a video; and in order to do so they exploit several features, like: color statistics, and high-level concepts (e.g. the pair adjective-nouns). Depending on the produced output, sentiment analysis methods can be grouped as follow: algorithms representing sentiment values as a polarity (positive or negative); algorithms able to distinguish among several emotion categories (amusement, anger, awe, contentment, disgust, excitement, fear, and sadness). Figure 1.5 shows two images containing respectively the tags providing a description of the visual content and an answer to the question *Which kind of emotion does the image arouse?* Both the images contain a panda, while the one in Figure 1.5a arouses a negative feeling (it has a prey in its mouth), the one in Figure 1.5b evokes a positive mood (it is resting and eating bamboo).

1.3 Characterize images by evaluating quality and aesthetics

Quality, aesthetics and sentiments are commonly treated as independent problems. However, humans perception of image pleasantness is the result of the relations among the aforementioned aspects. Additionally, these three attributes not only characterize

(a) (b)

Fig. 1.5 Sample images annotated considering image sentiment. Yellow boxes contain the most probable tag of a common auto-tagging system: "panda". Red boxes provide a label referring to the sentiment polarity for the two images, respectively negative (a) and positive (b).

the "global" palatability of images, but also influence each other, for example: perceived image quality impacts on aesthetics [3]; image aesthetics is strongly related to sentiment (i.e. semantic gap) [13]; image quality might impact on visual sentiment. Thus, it is possible to say that these attributes are heavily intertwined and their union can provide a more detailed description of image content properties.

This thesis goes deep into the two problems of automatic image quality assessment and of automatic image aesthetics assessment respectively. Algorithms based on deep learning techniques, specifically on *convolutional neural networks (CNN)*, outperforming the state-of-the-art are proposed.

Differently from many image quality assessment algorithms dealing with automatic image quality assessment of synthetically distorted images [87, 162], which cannot well model the complex mixtures of multiple distortions afflicting real-world images, in this thesis a method for blind image quality assessment on authentically distorted images is proposed [14].

The second visual attribute is image aesthetics. While the aforementioned attribute is related to the perceived quality of the signal, aesthetics depicts perceived beauty. In this thesis the problem of image aesthetic quality assessment is investigated and results show sentiment dependency of aesthetics (i.e. sentiment gap) [13].

Given that one of the most popular visual contents is the face (e.g. on social networks for photo sharing), portrait images aesthetics assessment is investigated. General content image aesthetic assessment systems are not effective for aesthetics assessment of portrait images because they miss information relative to facial attributes, that can encode relevant aspects to guide aesthetic estimation. For this reason, in this thesis an algorithm involving the previous visual attributes (i.e. quality and aesthetics)

and facial attributes description is proposed. Specifically, at first, a robust smile detector algorithm is developed (it represents an important visual feature for portrait aesthetics [117]), then a multiple-task model is designed in order to simultaneously estimate soft biometrics and attributes such as hair colors and styles, types of beards.

1.4 Mimic human subjectivity

It should be evident that semantic content analysis is fairly objective compared to quality, aesthetics and sentiment assessment. Image appeal depends on highly subjective factors not easily describable by low-level features or even image content as a whole. Such aspects could be sociocultural, demographic, or influenced by mood.

For designing and evaluating reliable models that are consistent with subjective human evaluations, collecting a large amount of human perceived scores is necessary.

Crowdsourcing systems like Amazon Mechanical Turk (AMT)[5] are extensively considered as effective human-powered platforms making it feasible to gather a large number of opinions from a diverse distributed populace over the web. Given a collection of images, participants to the task are asked to provide an *opinion* on the perceived visual attribute (quality, aesthetics or sentiment) of the presented images.

A "global opinion" needs to be obtained in order to define the ground-truth to be used for the evaluation of highly subjective problems such as image quality assessment and image aesthetic assessment. Several rules exist to obtain an objective "global opinion" given a collection of subjective labels, such as unanimous agreement and mean opinion score. The unanimous agreement assigns to an image the label all the participants agreed. Obviously this is a very strong constraint and it is mainly used when the number of both participants and labels is very limited (e.g. for sentiment analysis five participants and two classes [166]). On the other hand, the average score for the image I, well known as *Mean Opinion Score (MOS)*, is usually used as measure for the perceived image quality and the image aesthetics; it is given by

$$MOS(I) = \frac{1}{N}\sum_{i=1}^{N} r_i(I), \tag{1.1}$$

where $r_i(I)$ is the i-th individual score given to image I. Therefore, the objective of the image quality and aesthetic assessment systems is to mimic human perception of visual attributes and consequently to obtain an high correlation with the MOS.

[5] http://mturk.com

1.5 Thesis overview

The first part of the thesis introduces deep learning concepts and convolutional neural network (CNN) models that are applied into all of the proposed methods.

Chapter 3 is dedicated to the problem of automatic assessment of signal quality in general content images. The chapter starts with an introduction to the problem of blind image quality assessment, datasets and methods are then described.

Chapter 4 describes in detail visual aesthetic assessment on general content images and the proposed solution to this problem. Finally a brief introduction to the problem of image sentiment analysis is provided.

Chapter 5 addresses the specific problem of visual aesthetic assessment on portrait images. The chapter describes the proposed solution which delves into facial attributes estimation, that can provide useful facial features for a richer representation of the face.

Finally, Chapter 6 ends the thesis summarizing the obtained results, reporting conclusions and giving the directions of future works.

Chapter 2

Convolutional Neural Networks

The last few years have seen an increasing interest of the artificial intelligence community for deep learning techniques [81]. These computational models are representation-learning methods with multiple levels of representation, obtained by composing multiple non-linear processing layers that can learn hierarchical representations with increasing levels of abstraction. The key aspect of deep learning is that, differently from conventional feature, such as local binary patterns (LBP) [110], histogram of oriented gradients (HOG) [28] and scale-invariant feature transform (SIFT) [94], these layers of features are not designed by humans; instead they are learned directly from data.

Convolutional neural networks are deep feed-forward neural network architectures that are easy to train and generalize much better than common neural networks. These architectures have proved to be very effective in many tasks (e.g. image understanding [76], speech recognition [124], robotics [85]) and they are widely adopted by the computer vision community.

2.1 Feed-forward neural networks

Feed-forward neural networks define a mapping $\mathbf{y} = f(\mathbf{x}; \theta)$ and learn the parameters θ that result in the best function approximation. For example, for a classifier, $\mathbf{y} = f(\mathbf{x})$ maps an input $\mathbf{x}$ (e.g. an image) to a category $\mathbf{y}$.

Feed-forward neural networks are called networks because they are typically represented by composing together many different non-linear functions. The model is associated with an acyclic graph, also defined as *architecture*, describing how functions are composed together. For example, given three functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$, the most commonly used structure of neural network is organized into multiple *layers*, i.e. $f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$. Specifically, $f^{(1)}$ is called the first layer of the network, $f^{(2)}$

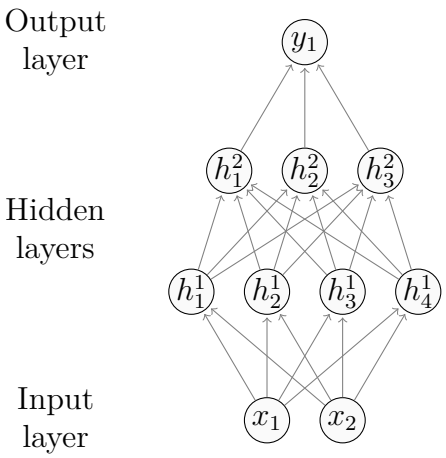


Fig. 2.1 A simple feed-forward neural network architecture.

is the second layer, and $f^{(3)}$ represents the last layer. The overall length of the chain gives the *depth* of the model. Figure 2.1 shows a simple neural network consisting of four layers: the input layer with two units or *neurons*, denoted x_1 and x_2; the first hidden layer composed by four neurons, i.e. h_1^1, h_2^1, h_3^1 and, h_4^1; the second hidden layer composed by three neurons, i.e. h_1^2, h_2^2, and, h_3^2; and finally the output layer having a single neuron, y_1.

Each neuron h_i in the neural network is a computational unit that takes as input the values from the preceding layer (h_i^{j-1}) that feed into h_i^j. As a concrete example, inputs to the neuron labelled h_1^1 in the sample neural network are x_1 and x_2, instead inputs to y_1 are h_1^2, h_2^2, and h_3^2. Given its inputs, a neuron first computes a weighted linear combination of those inputs, parametrized by a weights matrix W and biases b, and then a non-linear activation function is applied. More precisely, let $x = x_1, ..., x_n$ denotes a set of input variables to a neuron h_j, then:

$$h_j = \phi\left(\sum_{i=1}^{n} w_{ji}x_i + b_j\right), \tag{2.1}$$

where the weight w_{ji} describes the interaction between h_j and input neuron x_i, b_j is a bias associated with neuron h_j and ϕ is a non-linear activation function.

When a set of input variables x are fed into a feed-forward neural network, activations of each neuron are computed by following Equation 2.1. Moreover since the activation of each neuron depends only upon the values of preceding layers, activations are computed starting from the first hidden layer (which depend only upon the input values), proceed

layer-wise through the network and finally produce an output, $\hat{y}$. This process where information propagates through the network is called *forward-propagation*.

In order to reduce some *objective* (or *loss*) function between network output $\hat{y}$ and ground-truth y, neural network parameters, weights W and biases b, have to be learned. During training, forward-propagation continues onward until it produces a scalar cost $J(\theta) = \mathcal{L}(f(\mathbf{x}; \theta), y)$. The *backward-propagation* algorithm [158], also called *backprop*, allows the information from the cost function to flow backward through the network in order to compute the gradient. An optimization algorithm (e.g. stochastic gradient descent) is then used to perform learning using the computed gradient.

2.2 Convolutional neural networks

Convolutional neural networks (CNNs) are a specialized kind of feed-forward neural network for processing data that has a known grid-like topology, such as images, videos and time-series. The name "convolutional neural network" indicates that the network employs a specialized kind of linear operation called *convolution*. At the most basic level, a convolutional neural network is a multi-layer, hierarchical neural network. There are only three peculiarities that distinguish CNNs from simple feed-forward neural networks: sparse connectivity, weight sharing, and spatial pooling or sub-sampling layers. A modern deep convolutional neural network consists of several layers, as shown in Fig. 2.2. Several stages of convolution, non-linearity are stacked, followed by more convolutional and fully-connected layers. Intuitively, the low-level convolutional filters, such as those in the first convolutional layer, can provide a low-level encoding of the input data, mid-level filters compose the previous information to a higher level of abstraction: moving to higher layers in the neural network the model encodes more and more complicated structures. In the case of image data, local combinations of edges form motifs, motifs assemble into parts and parts compose objects. In addition to convolutional and fully-connected layers, various optional layers can be considered such as pooling and normalization. The following sections describe in detail components characterizing a classic CNN.

2.2.1 Convolutional layer

In traditional multi-layer networks each neuron is densely or *fully* connected to each of the neurons in the subsequent layer. However, in image processing it is often advantageous to exploit only a small local substructure within the image. For example,

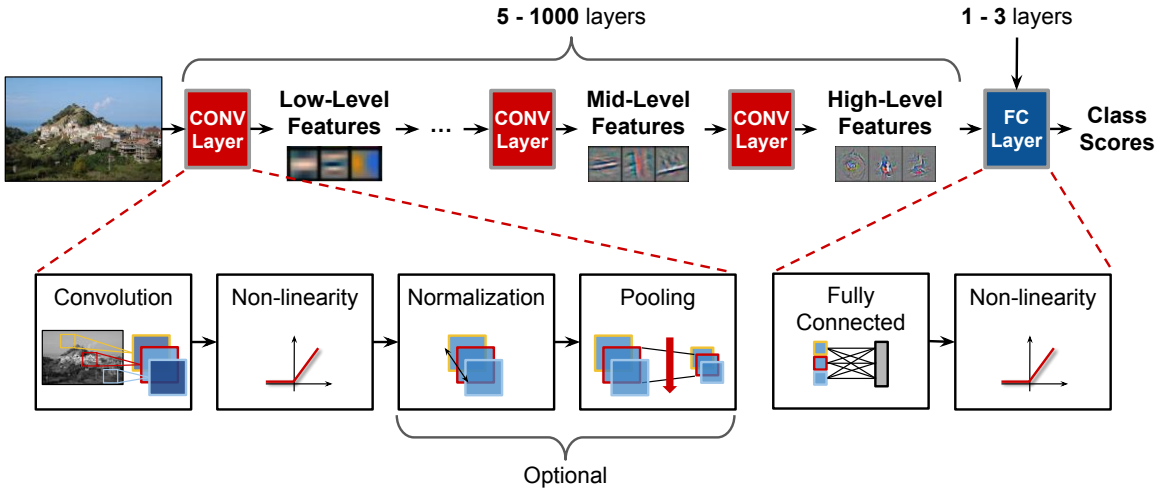


Fig. 2.2 Convolutional neural network.

pixels that are close together in the image (e.g. adjacent pixels) tend to be strongly correlated and can represent meaningful features such as edges, while pixels that are far apart in the image tend to be weakly correlated or uncorrelated. Many standard feature representations used in computer vision problems are based upon local features within the image (e.g. HOG and SIFT). In order to reflect the aforementioned property, a CNN architecture has *sparse* connectivity: the local substructure within a image is captured by constraining each neuron to depend only on a spatially local subset of the variables of the previous layer. The set of neurons in the input layer that affect the activation of a neuron is known as the neuron's *receptive field*. Figure 2.3 contains a graphic comparison between fully and dense connectivity.

The second feature that distinguishes CNNs from simple neural networks is the weight sharing. Apart from limit the number of weights that contribute to an output (sparse connectivity), edge weights in the network are shared across different neurons in the hidden layers. This process can be view as evaluating a kernel or *filter* at every position of the input. Concretely, weight sharing means that rather than learning a separate set of parameters for every location, we learn only one set. Using the same set of filters over an entire input forces the network to learn a general encoding or representation of the underlying data. Constraining the weights to be equal across different neurons has a regularizing effect on the CNN (allowing the network to generalize better), reduces the number of free weights in the CNN (making it easy to

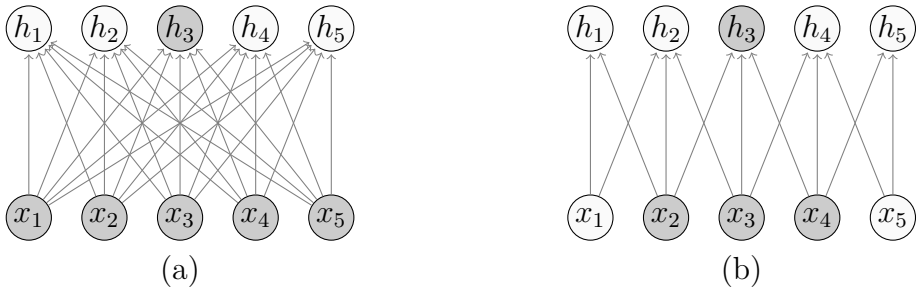


Fig. 2.3 Dense vs. sparse connectivity. Output neuron, h_3, and input neurons in x affecting the neuron h_3 (known as the receptive field of h_3) have been highlighted. (a) Dense connectivity, because all the input neurons affect h_3. (b) Sparse connectivity, as h is formed by convolution with a kernel of width 3, only three input neurons affect h_3.

train), and also reduces the storage required for model weights. Finally, evaluating the filter F over each window in the input I amount to perform a convolution of the input I with the filter F. Thus in the *convolutional step* of the CNN, an input is filtered with F to obtain a response or *feature* map.

The hyperparameters of convolutional layer are its spatial filter size, depth, stride, and padding. Filter size corresponds to the spatial extend (width and height) of the filters that are convolved with the input image at different spatial locations. Generally, the filters are square-shaped and as described in Section 2.8 recent architectures tend to use small filters in order to reduce learnable parameters. The depth of the output controls the number of filters that connect to the same region of the input volume. All of these filters will learn to activate for different feature of the input (e.g. filters of the first convolutional layer may activate in presence of various oriented edges, or blobs of color). The stride controls the filter shift and determines the dimension of the resulting activation map: higher stride reduce receptive fields overlap and reduce spatial dimensions. The padding parameter allows to control the spatial size of activation maps by extending the input activation map. This is commonly done by adding zeros at activation map outer edges.

2.2.2 Non-linear activation layer

A non-linear activation layer is usually applied after each convolutional layer of fully-connected layer. Various non-linear functions are used to introduce non-linearity into a CNN as shown in Figure 2.4. Traditional non-linear activation functions are sigmoid and hyperbolic tangent. These functions tend to saturate respectively at zero and one, and minus one and one, causing the so called *vanishing gradient problem*: if the

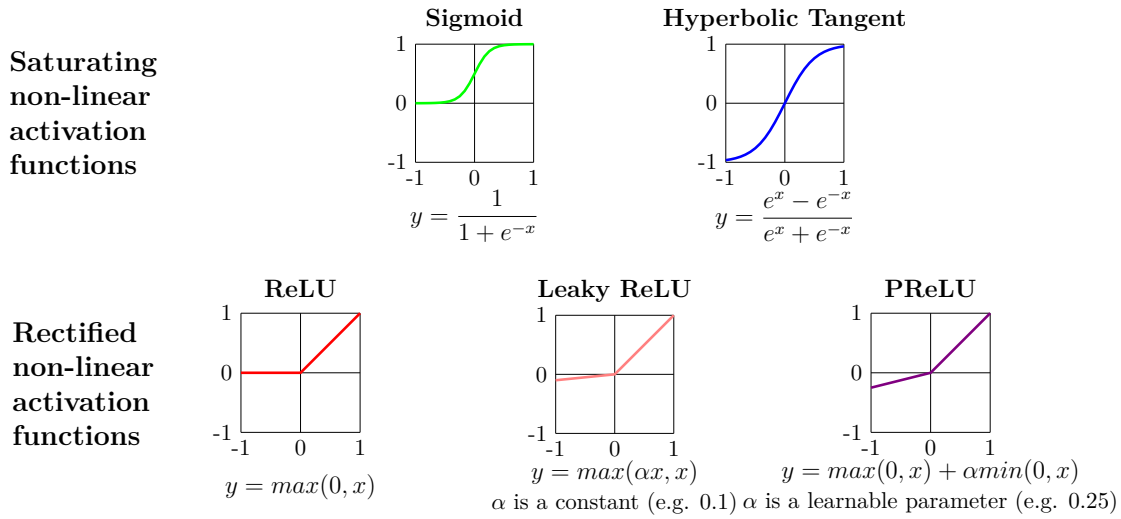


Fig. 2.4 Various examples of non-linear activation functions.

activity in the network during training is close to zero then the gradient for the sigmoid function may go to zero. For this reason non-saturated activation functions such as the Rectified Linear Unit (ReLU) [108] have been introduced. ReLU is a piecewise linear function which prunes the negative part to zero, and retains the positive part. It allows a network to easily obtain sparse representation that is desirable because: more biologically plausible; leads to mathematical advantages, such as information disentangling and linear separability [48]. Due to its simplicity and its stability to enable fast training, ReLU is the most used activation function at the moment. Variations of ReLU, such as the LeakyReLU [97], and the Parametric Rectified Unit (PReLU) [54] have also been explored. In contrast to ReLU, LeakyReLU assigns a non-zero slope controlled by a constant parameter. Instead, in PReLU function the slopes of negative part are learned from data rather than predefined. The authors claimed that PReLU is the key factor of surpassing (4.94% top-5 error) for the first time, human-level performance on ImageNet classification (5.1% top-5 error) [32].

2.2.3 Pooling layer

The presence of sub-sampling or pooling layers is the last aspect that distinguishes CNNs and simple neural networks. The goal of these layers is twofold: reduce the dimensionality of feature maps and confer a small degree of translation invariance to the representation. This last property is useful if it is more important to assess the presence of a feature than its spatial position. A spatial pooling function [18] replaces

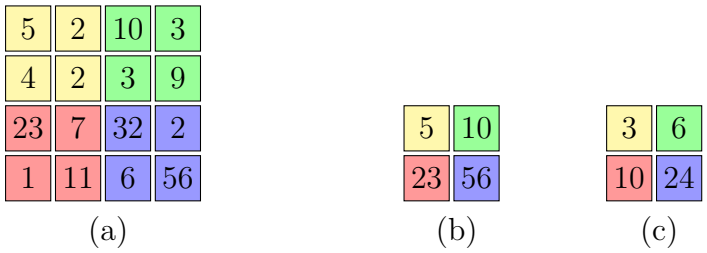


Fig. 2.5 Pooling operation in a convolutional neural network. (a) A 4×4 feature map divided into 2×2 blocks, with stride 2. (b) Max pooling response map contains the maximum value of each block. (c) Average pooling response map consists of the average value of each block values.

output of the net at a certain location with a summary statistic of the nearby outputs. Specifically, the feature map is first divided into a grid of $m \times n$ blocks without overlap and then a pooling method is evaluated over the responses of each block. This process yields a smaller feature map with dimension $m \times n$ (one response for each block). Figure 2.5 shows two different pooling methods (max and average) on a 4×4 feature map. In this case, the feature map is arranged in a 2×2 grid. In the case of max pooling, the response for each block is taken to be the maximum value over the block responses, and in the case of average pooling, the response corresponds to the average value of the block responses.

2.2.4 Normalization layer

Normalization layer enables to control distribution across layers to significantly speed up training and improve performances. Distribution of input layers activations (σ, μ) is normalized such that it has zero-mean and a unit standard deviation. Local Response Normalization (LRN) [76] was extensively used. It was inspired by lateral inhibition in neurobiology where excited neurons (i.e., high value activations) should subdue its neighbors (i.e., cause low value activations).

In Batch Normalization (BN), now considered standard practice in the design of CNNs, the normalized value is further scaled and shifted, as shown in Eq. (2.2), where the parameters (γ, β) are learned during the training phase [61]. Batch Normalization is usually performed between the convolutional or fully-connected and the non-linear function. It alleviates a lot of problems with properly initializing CNNs by explicitly forcing activations through a network to take on a unit normal distribution at the very

beginning of the training.

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}\gamma + \beta \tag{2.2}$$

LRN has been usually performed after the non-linear function, while BN is mostly performed between the convolutional layer and the non-linear function.

2.2.5 Loss functions

Loss layer, also referred to as *cost function*, *objective function* or *criterion*, is one of the essential parts of any neural network. In a supervised learning context, a data loss function measures the compatibility between a prediction (e.g. the class scores in classification), computed for any given training input image, and the corresponding ground-truth label. The loss can be seen as a distance metric that quantifies how far away the predictions are from the ground-truth labels and can be computed as:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i(y_i, \hat{y}_i), \tag{2.3}$$

where $\mathcal{L}$ denotes the overall loss, $\mathcal{L}_i$ the per-example loss, $\hat{\mathbf{y}}$ the network's prediction, $\mathbf{y}$ the ground-truth label, and N the number of training samples. Let's abbreviate $o = f(x_i; \theta)$ to be the activations of the output layer in a CNN, also known as *logits* (given that $\mathcal{L}_i$ is the per-example loss, index i is omitted for simplicity). Several loss functions have been proposed for each kind of task. Hinge loss, also referred as Support Vector Machine (SVM) loss, has been commonly used for classification problems with a single correct label (out of a fixed set) for each example. The multi-class hinge loss can be formalized as

$$\mathcal{L}_i^{Hinge} = -\sum_{j \neq y_i}^{K} \max(0, o_j - o_{y_i} + \delta), \tag{2.4}$$

with K corresponding to the number of classes and δ being the fixed margin. The hinge loss prefers the score of the correct class y_i to be larger than the incorrect class scores by at least a margin δ. If this is not the case, loss is accumulated. In order to strongly penalize wrong predictions, the squared hinge loss can be used. Cross-entropy, shortly XEntropy, is the most popular cost function for single-label image classification in CNNs. It is generalized to multiple classes via the softmax function and the negative

log-likelihood. Mathematically, the cross-entropy loss is defined as

$$\mathcal{L}_i^{XEntropy} = -\log\left(\frac{e^{o_{y_i}}}{\sum_{j=1}^{K} e^{o_j}}\right) = -o_{y_i} + \log\sum_{j=1}^{K} e^{o_j}, \tag{2.5}$$

where K is the number of classes. Both hinge and cross-entropy loss usually result in comparable classification performance. However, unlike the hinge loss, which treats the outputs as uncalibrated scores for each class, the cross-entropy loss results in normalized class probabilities.

To deal with multi-label classification problems, the sigmoid (binary) cross-entropy, shortly BCE, might be used. It is defined as follow:

$$\mathcal{L}_i^{BCE} = \sum_{j=1}^{K} y_{ij}\log\hat{y}_j + (1-y_{ij})\log(1-\hat{y}_j), \tag{2.6}$$

where the labels y_{ij} are assumed to be either 1 or 0, and $\hat{y}_j$ are the probability predictions $\hat{y}_j = \sigma(o_j) \in [0,1]$ using the sigmoid function $\sigma(.)$, already cited in Section 2.4.

For regression tasks in which predicted values are scalars or metrics (i.e. predict quality score of an image), it is common to compute the loss between the predicted score and the ground-truth in terms of L1-norm or L2-norm. The absolute value (AE) loss is the L1-norm of the error for each training sample and is formulated by summing the absolute value along each dimension:

$$\mathcal{L}_i^{AE} = \|o - y_i\|_1 = \sum_j |o_i - (y_i)_j|. \tag{2.7}$$

Minimizing the absolute value loss means predicting the (conditional) median of y. The square error (SE), or euclidean, loss which is the L2-norm of the error is the most used criterion for addressing regression tasks and is defined as follows:

$$\mathcal{L}_i^{SE} = \|o - y_i\|_2^2 = (o_i - (y_i)_j)^2 \tag{2.8}$$

Minimizing the squared error is equivalent to predict the (conditional) mean of y. This means that, if the training set is strongly unbalanced on medium scores, the model might be prone to predict values around the meadium score. L2-norm is much harder to optimize than a more stable cross-entropy loss: while Euclidean loss requires the network to output exactly one correct value for each input, the cross-entropy loss only requires the score magnitudes (not the precise value of each score) to be appropriate.

Additionally, given that L2-norm penalizes large errors more strongly, it is not robust to outliers and causes huge gradients. For these reasons, move from a regression to a classification problem by quantizing the output into bins represents one good option whether possible (as described in Section 3.1).

2.3 Model initialization

Before a neural network training can begin, its parameters needs to be initialized. This operation is very important for a neural network to train effectively. A common initialization method for biases is to set them to zero, however an initialization to small positive values might mitigate the adverse effect of dead neurons when using rectified activation functions. For the weights instead several initialization approaches have been proposed. One of the historical initialization methods for weights is the so called LeCun initialization [82], which proposes to sample weights from a multinomial normal distribution with mean zero and a very small standard deviation. During back-propagation computed gradients are proportional to their weights, i.e. a layer with small initialized weights will produce small gradients and viceversa.

The problem with the aforementioned initialization is that the distribution of the outputs from a randomly initialized neuron has a variance that grows with the number of inputs. Variance of each neuron's output can be normalized to one by using a normalization scheme, called Xavier [47], which scales neuron's weights by the average of the number of its inputs and outputs.The Xavier initialization method can be formalized as

$$\mathbf{w_0} = \mathcal{N}\left(\mu = 0, \sigma^2 = 2/(n_{in} + n_{out})\right) \tag{2.9}$$

where n_{in} and n_{out} denote respectively the number of inputs and outputs of the current layer.

As already said in Section 2.2.4 the batch normalization layer forces activations to take on a unit Gaussian distribution from the early training phases. Thus its introduction makes networks more robust to bad initialization.

2.4 Optimization

Once the gradients of the loss with respect to parameters are computed with back-propagation, they are used to perform gradient descent parameter update. Mathemati-

cally gradient descent updates rule have the following form

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}, \tag{2.10}$$

where θ are network parameters, and η denotes the *learning rate*, a hyperparameter that controls the step size of a single update. Selection of a proper learning rate might be difficult: a learning rate too small leads to slow converge, while a large learning rate can hinder convergence and cause loss fluctuation around the minimum or even diverge.

The commonly used variant of gradient descent is the mini-batch stochastic gradient descent (SGD) [118]. This algorithm performs update for every mini-batch of training example $\mathbf{x}_\mathcal{B}$ and its corresponding labels $\mathbf{y}_\mathcal{B}$. The hyperparameter $\mathcal{B}$ denotes the *batch size*, i.e. how many training examples consider per batch to perform a single parameter update. Mini-batch SGD reduces the variance of the parameter updates leading to more stable convergence. Additionally, it enables optimization and parallelized matrix operations.

Batch size hyperparameter should not affect the convergence behaviour of mini-batch stochastic gradient descent in significant ways: its impact is mostly of computational nature. However, it has to be noted that the batch size directly affects the batch normalization process since the statistics may vary significantly when choosing smaller batch sizes.

2.4.1 Optimization methods

Several improvements to the standard mini-batch SGD update rule exist. Momentum update [122] is the most known optimization approach that results in faster convergence rates of stochastic gradient descent. The momentum loss can be interpreted as the height of a hilly terrain and the optimization process can be seen as a ball rolling downwards on a landscape. Differently from standard SGD update, the gradient navigates along relevant directions and softens the oscillations in irrelevant directions. It's update rule is formalized as

$$\begin{aligned} \mathbf{v} &\leftarrow \mu\mathbf{v} - \eta \cdot \nabla_\theta \mathcal{L} \\ \theta &\leftarrow \theta + \mathbf{v}, \end{aligned} \tag{2.11}$$

where $\mathbf{v}$ is the velocity that accumulates the gradients over time and the hyperparameter μ is referred as *momentum*. The latter parameter increases for dimensions whose

gradients point in the same directions and reduces updates for dimensions whose gradients change directions. Nesterov momentum [136] is a slightly different version of the regular momentum update. It addresses the problem of regular momentum that can miss the minima because of an high value of accumulated velocity. Nesterov momentum allows to slow down before the hill slopes up again. This is given by the following

$$\begin{aligned} \mathbf{v} &\leftarrow \mu\mathbf{v} - \eta \cdot \nabla_{(\theta+\eta\mathbf{v})}\mathcal{L} \\ \theta &\leftarrow \theta + \mathbf{v}, \end{aligned} \tag{2.12}$$

where all variables correspond to the ones of the regular momentum update, except that the gradient is evaluated at $\theta + \eta\mathbf{v}$ instead of the old position θ.

The momentum improvements of SGD manipulate all parameters equally since they employ a global learning rate. Adaptive optimization methods have been also proposed. These approaches allow to adaptively tune the learning rate and perform per-parameter updates. Example of adaptive optimization methods are: Adagrad [35], Adadelta [170] and Adam [74].

2.5 Regularization

In machine learning regularization is any supplementary technique that aims at making the model generalize better, i.e. produce better results on the test set. The huge amount of parameters makes CNNs is prone to overfitting, furthermore non-convexity of loss function implies the presence of many different local minima. In order to mitigate overfitting phenomenon and reach a local minimum that explains the data in the simplest possible way according to the Occam's razor, several regularization methods are introduced.

2.5.1 Network architecture design

Network architecture design is the most incisive form of implicit regularization [15]. The use of shared weights or sparse (local) connections drastically reduce the number of parameters of the network (as considered in Section 5.4.1.2). Deeper architectures tend to act as regularizers over wide ones. This motivates, for example, design choices for the VGG architecture [132] and justify higher accuracy performances than AlexNet architecture [76] on the ILSVRC2012 challenge [32]. VGG architecture is deeper than AlexNet, additionally 5×5 filters are substituted with two layers of 3×3 filters

interleaved by a non-linearity. The result is still a receptive field with size 5×5 pixels but the benefit is twofold: the use of less parameters and an higher capacity.

2.5.2 Early-stopping

Early stopping is the easiest and the most effective form of regularization. The performance measure on the validation set is continuously monitored and the training is stopped once the performance stops improving. In practice, one can save the best-performing model parameters in addition to the current parameters and fall back on the saved one once further improvements seem unlikely.

2.5.3 Dropout

The dropout technique is one of the most effective and simple regularization techniques acting on the network itself [56, 134]. The key idea is to randomly drop neurons from the neural network during training and thus preventing the co-adaptation of features. At each training stage, some neurons are randomly omitted from the network with probability p using samples from a Bernoulli distribution, such that a reduced network is left and individual activations cannot rely on other activations to be present simultaneously. Additionally, another way to view the dropout procedure is as a very efficient way of performing model averaging with neural networks. As previously said, random dropout makes possible to train a huge number of different networks sharing the same weights. Batch normalization (described in Section 2.2.4) fulfills some of the same goals as dropout: removal or reduction in strength of dropout does not imply a loss of generalizability [61].

2.5.4 Weights regularization

Weights regularization is the most common form of regularization. Differently from Dropout it does not rely on modifying the network but the loss function. Specifically it introduces an additional term to the criterion such that the total loss is a combination of data loss (e.g. cross-entropy) and regularization loss as in

$$\mathcal{L}(w) = \underbrace{\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i}_{\text{data loss}} + \underbrace{\lambda R(w)}_{\substack{\text{regularization} \\ \text{loss}}} , \tag{2.13}$$

where $\mathcal{L}_i$ might be any loss function, $R(w)$ is the regularization penalty and λ is a hyperparameter that controls the regularization strength.

The intuition behind weight regularization is therefore to prefer smaller weights, and thus the local minima which have a simpler solution. This technique is also referred as *weight decay*. Given that biases do not interact with the data in a multiplicative fashion, and therefore do not have much influence on the loss, regularization is only applied to the weights of the network.

The regularization penalty, $R(w)$, can be defined in several ways, say: Lasso or L_1 regularization encourages sparsity by computing the L_1-norm of the weights (i.e. sum of the absolute values); the most popular Ridge or L_2 regularization determines small weights and corresponds to the L_2-norm of the weights (i.e. the sum of squares) [76].

2.5.5 Data augmentation

The phrase *The more you see, the more you know* by Aldous Huxley in The Art of Seeing is true for humans as well as neural networks, especially for deep neural networks. Very large CNNs consist of hundreads of billions parameters to be learned and available training data is sometimes not sufficient to train such deep networks. Additionally, the examples collected are often just in a form of "good" data, or a rather small and biased subset of the possible space. Data augmentation techniques enable to simulate diverse label-preserving images for improving the performance of CNNs by making them invariant to some transformations of the images. Practically data augmentation increases the volume of the training dataset by applying several transformations to the original input. Traditional augmentation strategies can be distinguished among: spatial transformations, such as random cropping, rotating, and flipping input images; color jittering by varying brightness, contrast, or saturation of input images. Further strategies distort input images using white noise, motion blur, and compression artifacts (applied in Section 5.4.1.2). This helps to obtain a sort of invariance also to such kind of artifacts. Recently, generative adversarial networks (GAN) [49] have been proved to be very effective in many data generation tasks. This enables to augment available datasets by generating totally new samples [176].

2.6 Data preprocessing

Data preprocessing is an essential part of any automatic learning process. It focuses on adapting the data to simplify and optimize the training of the learning model. The high

abstraction capacity of CNNs allows them to work on the original high dimensional space, which reduces the need for manually preparing the input. However, a suitable preprocessing is still important to improve the quality of results. Preprocessing mainly involves data normalization, which stabilizes training procedure by scaling input data.

In order to keep the range of distributions of feature values under control and avoid the training procedure to oscillate or to move slowly, several forms of data normalization, such as mean subtraction and standardization and have been proposed. Per-pixel mean subtraction is the most used normalization technique whether the statistics for each data dimension follow the same distribution. The idea is to subtract the mean from each data point in order to zero-center them. Average image is obtained by computing the mean across all training set images. Sometimes the average image is replaced with mean pixel (i.e. the overall mean value over each color channel) in order to apply the per-channel mean subtraction. This is useful, for example, when CNN input data size is not prefixed. Switching from mean image to mean pixel and vice-versa as preprocessing for data to be feed into a learned model has no particular effects. The standardization technique normalizes the data dimensions so that they are approximately the same scale. Standardization implies that normalized data have zero-mean and unit standard deviation. This is done for each data point by first subtracting mean and then by dividing standard deviation. Mean and standard deviation could be computed independently for each data point (also known as contrast normalization), or for the entire training set.

2.7 Transfer learning

As explained in the previous section, a CNN consists of million of weights that needs to be learned using a supervised training algorithm on a large-scale annotated databases. Unfortunately, there exists no large-scale dataset for each challenge and a suitable method to face this problem is based on reusing off-the-shelf pre-trained models thanks to *transfer learning* approaches [9, 116, 164]. Specifically, transfer learning techniques first involve the training of a model for a *source* task on a large-scale dataset (e.g. ImageNet), and then the use of the pre-trained model on a *target* task somehow related to the *source* task. Transfer learning can be used in one the following ways:

- Model as feature extractor [34]: The knowledge gained in the *source* model can be used to build features for the data points belonging to the *target* task dataset, and such features (fixed) are then fed to new models. For example, it is possible to feed new images through a pre-trained CNN and use activations of any desired

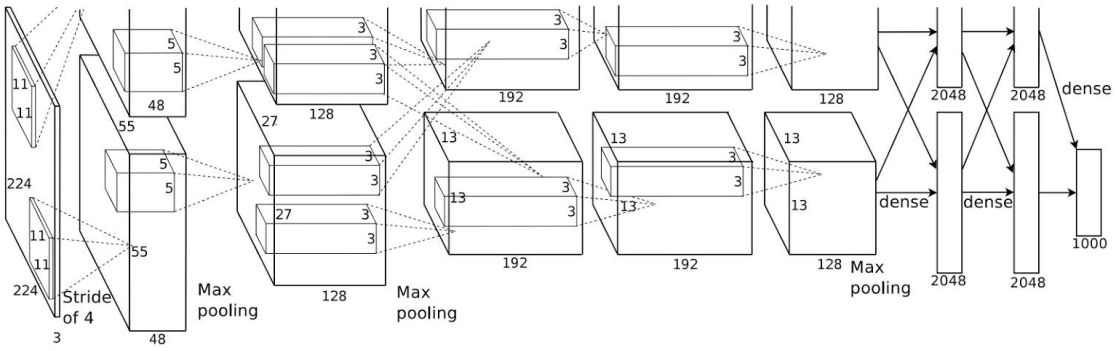


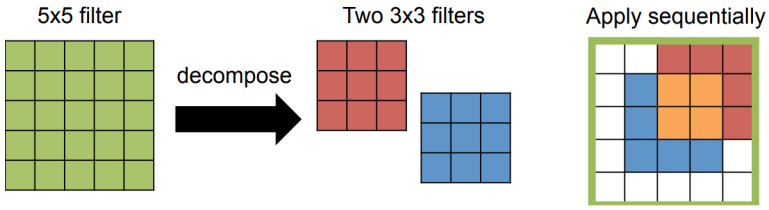
Fig. 2.6 An illustration of the AlexNet architecture. Image from [76].

layer as feature vectors for these images. The features thus built can be used in a classifier for the desired situation.

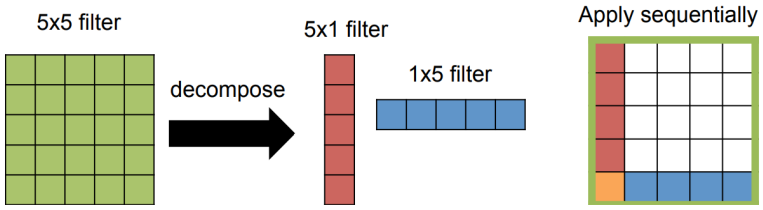
- Fine-tuning the *source* model [46]: In this strategy, a pre-trained CNN can be used as an initialization (rather than random initialization) for a further learning process. The model will be trained on the much smaller user-provided data of the *target* task dataset. The advantage of such a strategy is that weights can reach the global minima without much data and training. According to the amount of available data it is possible to consider whether to fix a portion of the model (usually the beginning layers) and only fine-tune the remaining layers.

2.8 Popular architectures

In computer vision domain, deep CNNs achieved impressive results in several challenges. In 2012, the proposed AlexNet [76] dramatically increased the performance on the 1000-class ImageNet Large-Scale Visual Recognition Competition (ILSVRC2012) [32]. Since then, more complex and deeper CNN architectures, such as VGG-16 [132] and GoogLeNet [139], have been designed in order to gradually improve recognition accuracy on ILSVRC benchmark. These architectures represent the starting point for new models design. *AlexNet* consists of five convolutional (CONV) layers and three fully-connected FC layers (see Figure 2.6 for details). Among the novelties of this network: ReLU non-linearity after each layer CONV or FC; the use of overlapping, instead of adjacent, pooling kernels; introduction of LRN layer; weight decay and data augmentation. In total, AlexNet requires 61M weights. *VGG-16* goes deeper to 16 layers consisting of 13 CONV layers and 3 FC layers. In order to balance out the cost of going deeper, all the CONV layers move from larger filters (e.g. 5×5) to



(a) Costructing a 5×5 support for 3×3 filters. Used in VGG-16.



(b) Costructing a 5×5 support from 1×5 and 5×1 filters. Used in GoogleNet/Inception v3 and v4.

Fig. 2.7 Decomposing larger filters into smaller filters. Image from [137].

multiple smaller filter of size 3×3 still maintaining the same receptive fields (see Fig. 2.7a). In total, VGG-16 requires 138M weights. *GoogLeNet*, also known as Inception net, goes even deeper with 22 layers. It introduces the inception module, shown in Figure 2.8, which is composed of parallel connections having different sized filters and max-pooling, then module output is obtained by concatenating parallel connection outputs. The use of multiple filter sizes enables to obtain a multi-scale processing. The 22 layers consist of three CONV layers, followed by 9 inceptions layers and one FC layer. In total, GoogLeNet requires 7M weights (12 times fewer than AlexNet). Since its introduction in 2014, other versions of GoogLeNet (Inception-v1) have been proposed: Inception-v3 (Inception-v2 is very similar) decomposes the filters as shown in Figure 2.7b to reduce the amount of weights and to go deeper to 42 layers, and introduces batch normalization [140]; Inception-v4 exploits insights from residual blocks [138]. *ResNet* architecture [55], also known as Residual Net, applies residual connections to go even deeper (34 layers or more depending on the version). It addresses the problem of *vanishing gradient* during training (the gradient might shrinks through very deep networks and this affects the ability to update the weights in the earlier layers) by introducing a "shortcut" module (also called residual block) which contains an identity connection such that the weight layers (i.e. CONV layers) can be skipped as shown in Figure 2.9a. Additionally, ResNet uses the "bottleneck" approach of using 1×1 filters to reduce the number of weight parameters (as in Fig. 2.9b). There are various

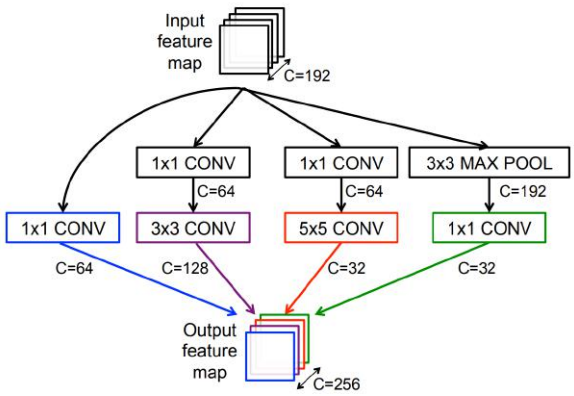
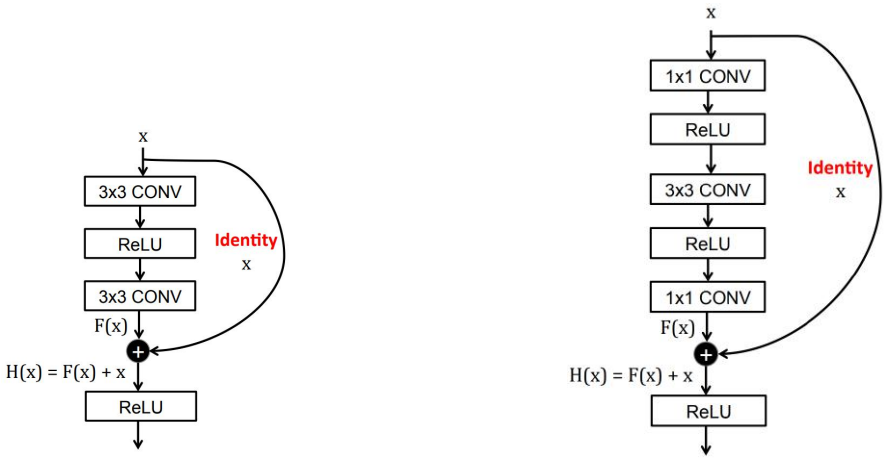


Fig. 2.8 Inception module from GoogleNet architecture with example channel lengths. ReLU nonlinearity after CONV layer omitted for compactness. Image from [137].



(a) Without bottleneck (b) With bottleneck

Fig. 2.9 Residual block from ResNet. Note the ReLU nonlinearity following the last CONV layer in short cut is *after* the addition. Image from [137].

versions of ResNet with multiple depths (e.g. 18, 50, 152); ResNet-50 requires 25.5M weights.

Chapter 3

Blind Image Quality Assessment

In this chapter the image quality assessment problem, methods, datasets and metrics will be described. Furthermore, the proposed solution to this problem will be detailed.

Digital pictures may have a low perceived visual quality. Capture settings, such as lighting, exposure, aperture, sensitivity to noise, and lens limitations, if not properly handled could cause annoying image artifacts that lead to an unsatisfactory perceived visual quality. Being able to automatically predict the quality of digital pictures can help to handle low quality images or to correct their quality during the capture process [19]. An automatic image quality assessment (IQA) algorithm, given an input image, tries to predict its perceptual quality. The perceptual quality of an image is usually defined as the mean of the individual ratings of perceived quality assigned by human subjects (Mean Opinion Score - MOS).

In recent years, many IQA approaches have been proposed [100, 133]. They can be divided into three groups, depending on the additional information needed: full-reference image quality assessment (FR-IQA) algorithms e.g. [36, 160, 112, 155, 154, 16], reduced-reference image quality assessment (RR-IQA) algorithms, and no-reference/blind image quality assessment (NR-IQA) algorithms e.g. [105, 102, 103]. FR-IQA algorithms perform a direct comparison between the image under test and a reference or original in a properly defined image space [25]. Having access to an original is a requirement of the usability of such metrics. RR-IQA algorithms are designed to predict image quality with only partial information about the reference image [25]. In their general form, these methods extract a number of features from both the reference and the image under test, and image quality is assessed only by the similarity of these features. NR-IQA algorithms assume that image quality can be determined without a direct comparison between the original and the image under test [25]. Thus, they can be used whenever the original image is unavailable. NR-IQA algorithms can be

further classified into two main sub-groups: to the first group belong those targeted to estimate the presence of a specific image artifact (i.e. blur, blocking, grain, etc.) [24, 26]; to the second group the ones that estimate the overall image quality and thus are distortion generic [104, 125, 19, 25].

Most of the distortion-generic methods estimate the image quality by measuring deviations from Natural Scene Statistic (NSS) models [19] that capture the statistical "naturalness" of non-distorted images. These models are based on the two following principles: i) good quality real-world photographic images obey certain perceptually relevant statistical laws; ii) common image distortions alter such statistical laws. The Natural Image Quality Evaluator (NIQE) [103] is based on the construction of a quality aware collection of statistical features based on a space domain NSS model. The Distortion Identification-based Image Verity and INtegrity Evaluation (DIIVINE) index [105] is based on a two-stage framework for estimating quality based on NSS models, involving distortion identification and distortion-specific quality assessment. The core of the method uses a Gaussian scale mixture to model neighboring wavelet coefficients. C-DIIVINE [174] is an extension of the DIIVINE algorithm in the complex domain, and blindly assesses image quality based on the complex Gaussian scale mixture model corresponding to the complex version of the steerable pyramid wavelet transform. The BLIINDS-II [123] method, given an input image, computes a set of features and then uses a Bayesian approach to predict quality scores. Such features are obtained by transforming the model parameters of a generalized NSS-based model of local Discrete Cosine Transform coefficients into a vector of features.

The Blind/Referenceless Image Spatial QUality Evaluator (BRISQUE) [102] operates in the spatial domain and is also based on a NSS model. The algorithm quantifies possible losses of naturalness in the image due to the presence of distortions.

The use of a database of images along with their subjective scores is fundamental for both the design and the evaluation of IQA algorithms [128, 45]. Recent approaches to the blind image quality assessment problem use these images coupled with the corresponding human provided quality scores within machine learning frameworks to learn directly from the data a quality measure. The Feature maps based Referenceless Image QUality Evaluation Engine (FRIQUEE) [43, 45] combines a deep belief net and a SVM to predict image quality. Tang et al. [142] define a simple radial basis function on the output of a deep belief network to predict the perceived image quality. They first pre-train the network in an unsupervised manner and then fine-tune it with labeled data. Finally they model the quality of images exploiting a Gaussian Process regression. Hou et al. [58] propose to represent images by NSS features and to train a discriminative

deep model to classify the features into five grades (i.e. excellent, good, fair, poor, and bad). Quality pooling is then applied to convert the qualitative labels into scores. In [96] a model is proposed which uses local normalized multi-scale difference of Gaussian (DoG) response as feature vectors. Then, a three-steps framework based on a deep neural network is designed and employed as pooling strategy. Ye et al. [163] presented a supervised filter learning based algorithm that uses a small set of supervised learned filters and operates directly on raw image patches. Later they extended their work using a shallow convolutional neural network [68]. The same CNN architecture has been then used to simultaneously estimate image quality and identify the distortion type [69] on a single-type distortion dataset [128].

Features extracted from CNN pre-trained for object and scene recognition tasks, have been shown to provide image representations that are rich and highly effective for various computer vision tasks. In this thesis their use for multiple generic distortions NR-IQA and their capability to model the complex dependency between image content and subjective image quality is investigated [4, 26, 144].

The hypothesis motivating this research is that the presence of image distortions such as JPEG compression, noise, blur, etc. might be captured and modeled by these features as well. As shown in Figure 3.1, the resulting activation map produced by convolving filters of the CaffeNet's first convolutional layer learned on the sharp images of ImageNet database is different depending on the quality of image signal. In fact, for a pristine image the result of convolution is a well defined activation map where oblique edges are emphasized, while given a noisy image the result of convolution is very noisy too. Furthermore, the more concepts the CNN has been trained to recognize, the better are the extracted features. The effect of several design choices are evaluated:

i) the use of different features extracted from CNNs that are pre-trained on different image classification tasks for an increasing variety and number of concepts to recognize;

ii) the use of a number of different image sub-regions (opposed to the use of the whole image) to better capture image artifacts that may be local or partially masked by specific image content;

iii) the use of different strategies for feature and score predictions pooling.

In this thesis a novel procedure for the fine-tuning of a CNN for multiple generic distortions NR-IQA is proposed, which consists in discriminatively fine-tuning the CNN to classify image crops into five distortion classes (i.e. bad, poor, fair, good, and

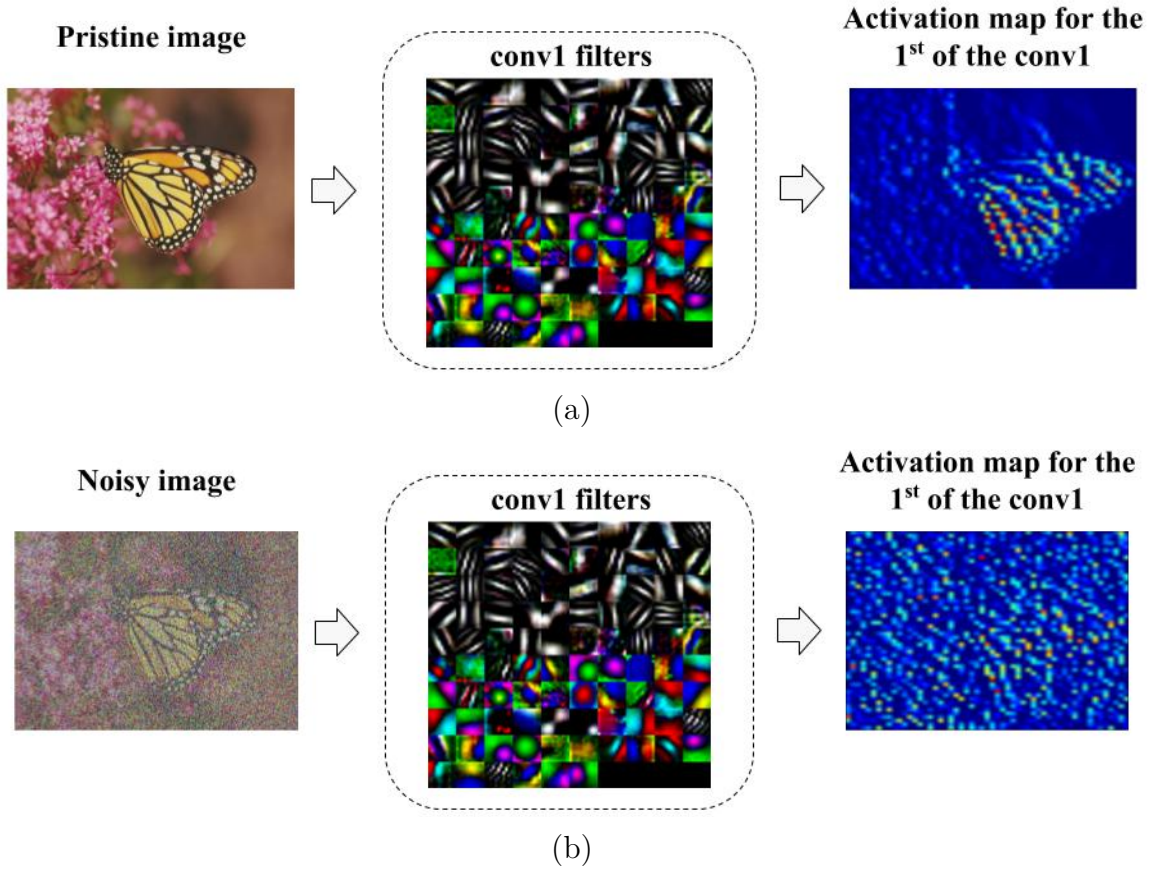


Fig. 3.1 Activation maps of CaffeNet's first filter of the first convolutional layer trained on ImageNet [64]. (a) The convolution between a pristine image of a monarch and the filter produces an activation map where oblique edges are emphasized. (b) Instead, the convolution between a noise version of the same monarch and the filter produces a very noisy activation map.

excellent) and then using it as feature extractor. Whatever is the feature extraction strategy and the related CNN, a Support Vector Regression (SVR) machine [148, 27] is finally exploited to learn the mapping function from the CNN features to the perceived quality scores [87].

Experimental results on the LIVE In the Wild Image Quality Challenge Database show that our method outperforms the state-of-the-art methods compared, including those based on deep learning. Its generalizability is further confirmed on four other benchmark databases of synthetically distorted images: LIVE, CSIQ, TID2008 and TID2013.

The experiments are conducted on the *LIVE In the Wild Image Quality Challenge Database* which contains widely diverse authentic image distortions on a large number of images captured using a representative variety of modern mobile devices [44]. The result of this study is a CNN suitably adapted to the blind quality assessment task that accurately predicts the quality of images with a high agreement with respect to human subjective scores. Furthermore, the applicability of the proposed method on legacy LIVE Image Quality Assessment Database [128], CSIQ [80], TID2008 [114] and TID2013 [113] is investigated.

3.1 Deep Learning for blind image quality assessment

As previously introduced in section 2.7, when small database are available it is a common practice to take a CNN that is pre-trained on a different large dataset (e.g. ImageNet [32]), and then use it either as a feature extractor or as an initialization for a further learning process (i.e. transfer learning, known also as fine-tuning [164, 9]). In this thesis, the Caffe network architecture [64] (inspired by the AlexNet [76]) is used as a feature extractor on top of which a Support Vector Regression (SVR) machine [148, 27] with a linear kernel is exploited to learn a mapping function from the CNN features to the perceived quality scores (i.e. MOS). The detailed architecture of the CNN used is reported in Table 3.1. Given an input image, the CNN performs all the multi-layered operations and the corresponding feature vector is obtained by removing the final softmax nonlinearity and the last fully-connected layer. The length of the feature vector is 4096. A graphical representation of the described approach is reported in Figure 3.2.

The effect of several design choices for feature extraction is investigated. Such solutions are: i) the use of different CNNs that are pre-trained on different image

Table 3.1 Architecture of Caffe network. It consists in 8 weight layers. The ReLU activation layers after each weight layer (except for *fc8*) are not shown for brevity. FC denotes fully connected layer type, while LRN represents the Local Response Normalization layer type.

	conv1	*pool1*	*norm1*	*conv2*	*pool2*	*norm2*	*conv3*	*conv4*	*conv5*	*pool5*	*fc6*	*fc7*	*fc8*
Type	Conv	MaxPool	LRN	Conv	MaxPool	LRN	Conv	Conv	Conv	MaxPool	FC	FC	FC
Kernel size	11×11	3×3		5×5	3×3		3×3	3×3	3×3	3×3			
Depth	96			256			384	384	256		4096	4096	
Stride	4	2		1	2		1	1	1	2			
Padding	0			2			1	1	1				

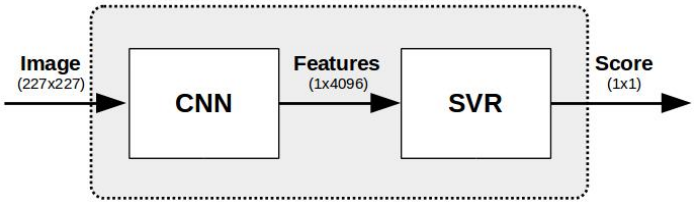


Fig. 3.2 Graphical representation of the main steps of the proposed approach: the input image is fed to a CNN which performs all the multilayered operations and extracts a feature vector. Then, an SVR maps the extracted features to the perceived quality scores (i.e. MOS).

classification tasks; ii) the use of a number of different image sub-regions (opposed to the use of the whole image) as well as the use of different strategies for feature and score prediction pooling; iii) the use of a CNN that is fine-tuned for category-based image quality assessment.

3.1.1 Image description using pre-trained CNNs

Razavian et al. [116] showed that the generic descriptors extracted from convolutional neural networks are very powerful and their use outperforms hand crafted, state-of-the-art systems in many visual classification tasks. Within the approach depicted in Figure 3.2, our baseline consists in the use of off-the-shelf CNNs as feature extractors. Features are computed by feeding the CNN with the whole image, that must be resized to fit its predefined input size (see Figure 3.3.a).

Three different CNNs sharing the same architecture that have been pre-trained on three different image classification tasks are evaluated:

- ImageNet-CNN, which has been trained on 1.2 million images of ImageNet (ILSVRC 2012) for object recognition belonging to 1,000 categories;

- Places-CNN, which has been trained on 2.5 million images of the Places Database for scene recognition belonging to 205 categories;

- ImageNet+Places-CNN [177], which has been trained using 3.5 million images from 1,183 categories, obtained by merging the scene categories from Places Database and the object categories from ImageNet.

3.1.2 Feature and prediction pooling strategies

In the design choice described in Section 3.1.1, the image is resized to match the predefined CNN input size. Since the resizing operation can mask some image artifacts, here a different design choice is considered in which CNN features are computed on multiple sub-regions (i.e. crops) of the input image. Crops dimensions are chosen to be equal to the CNN input size so that no scaling operation is involved (see Figure 3.3.b). Each crop covers almost 21% of the original image (227×227 out of 500×500 pixels), thus the use of multiple crops permits to evaluate the local quality. The final image quality is then computed by pooling the evaluation of each single crop. This permits, for instance, to distinguish between a globally blurred image and a high-quality depth-of-field image.

The use of a different number of randomly selected sub-regions [76], ranging from 5 to 50, is also experimented. The information coming from the multiple crops has to be fused to predict a single quality score for the whole image. Different fusion strategies are experimented:

- *feature pooling*: information fusion is performed element by element on the sub-region feature vectors to generate a single feature vector for each image (see Figure 3.4.a). Minimum, average, and maximum feature pooling are considered. Lets call $f_i^{(j)}$ the $i-$th entry of the feature vector relative to the $j-$th image sub-region. In the case where N_s different sub-regions are considered and each feature vector has size N_d, the three feature poolings considered can be expressed as:

$$F_i^{\min} = \min_{j=1,\ldots,N_s} f_i^{(j)}, i = 1, \ldots, N_d \tag{3.1}$$

$$F_i^{\text{avg}} = \frac{1}{N_s} \sum_{j=1,\ldots,N_s} f_i^{(j)}, i = 1, \ldots, N_d \tag{3.2}$$

$$F_i^{\max} = \max_{j=1,\ldots,N_s} f_i^{(j)}, i = 1, \ldots, N_d \tag{3.3}$$

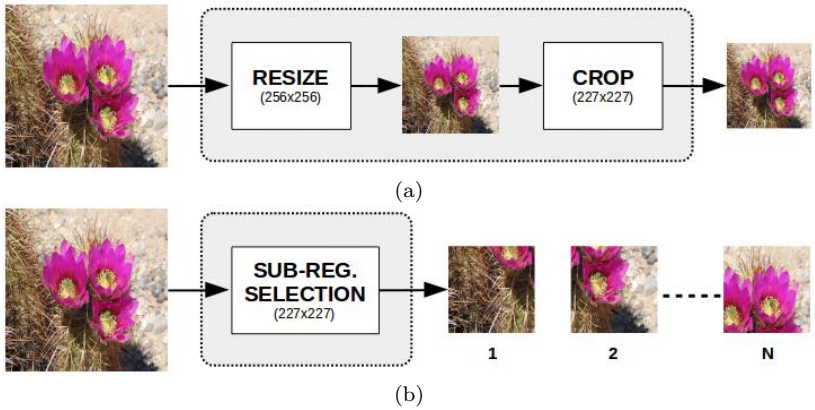


Fig. 3.3 Graphical representation of different design choices: use of the whole image resized to fit the CNN input size (a); and use of multiple image sub-regions taken from the fullsize image (b).

- *feature concatenation*: information fusion is performed by concatenating the sub-region feature vectors in a single longer feature vector (see Figure 3.4.b). Formally, feature concatenation can be expressed as follows:

$$F = \left[f_i^{(1)} \oplus \cdots \oplus f_i^{(j)} \oplus \cdots \oplus f_i^{(N_s)} \right] \tag{3.4}$$

- *prediction pooling*: information fusion is performed on the predicted quality scores. The SVR predicts a quality score for each image crop, and these scores are then fused using a minimum, average, or maximum pooling operators (see Figure 3.4.c). Lets call $q^{(j)}$ the predicted quality score for the $j-$th image crop. The three prediction poolings considered can be formally described as:

$$Q^{\min} = \min_{j=1,\ldots,N_s} q^{(j)} \tag{3.5}$$

$$Q^{\text{avg}} = \frac{1}{N_s} \sum_{j=1,\ldots,N_s} q^{(j)} \tag{3.6}$$

$$Q^{\max} = \max_{j=1,\ldots,N_s} q^{(j)} \tag{3.7}$$

3.1.3 Image description using a fine-tuned CNN

Convolutional neural networks usually require millions of training samples in order to avoid overfitting. Since in the blind image quality assessment domain the amount of data available is not so large, the fine-tuning of a pre-trained CNN exploiting the

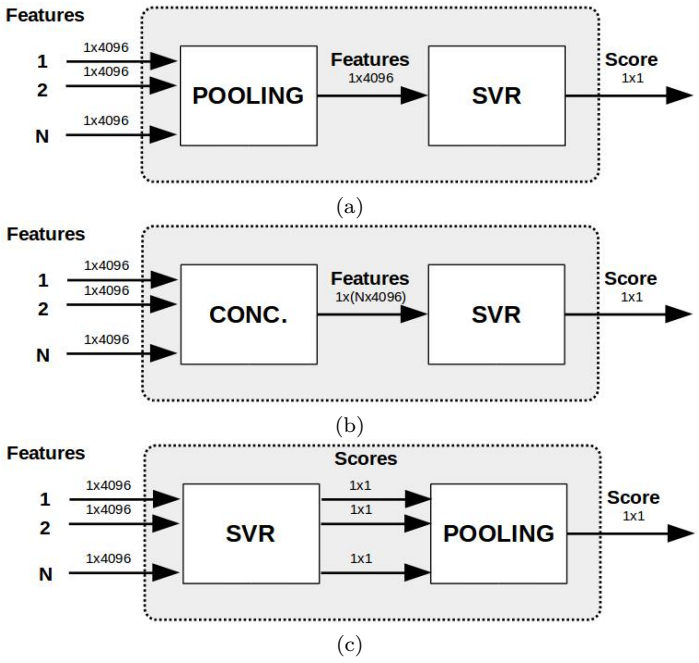


Fig. 3.4 Graphical representation of different design choices to pool information coming from multiple image sub-regions: feature pooling (a), feature concatenation (b), and prediction pooling (c).

available NR-IQA data is investigated. When the amount of data is small, it is likely best to keep some of the earlier layers fixed and only fine-tune some higher-level portion of the network. This procedure, which is also called transfer learning [164, 9], is feasible since the first layers of CNNs learn features similar to Gabor filters and color blobs that appear not to be specific to a particular image domain; while the following layers of CNNs become progressively more specific to the given domain [164, 9].

Firstly a pre-trained CNN is fine-tuned to the image quality assesment task by substituting the last fully connected layer with a new one initialized with random values. The new layer is trained from scratch, and the weights of the other layers are updated using the back-propagation algorithm [83] with the available data for image quality assessment. In this thesis, image quality data are a set of images having human average quality scores (i.e. MOS). The CNN is discriminatively fine-tuned to classify image sub-regions into five classes according to the 5-points MOS scale [135]. The five classes are obtained by a crisp partition of the MOS: bad (score $\in [0, 20]$), poor (score $\in]20, 40]$), fair (score $\in]40, 60]$), good (score $\in]60, 80]$), and excellent (score $\in]80, 100]$). Once the CNN is trained, it is used for feature extraction within the approach depicted in Figure 3.2, just like one of the pre-trained CNNs.

Table 3.2 A comparison of image quality assessment databases.

Database	Number of reference images	Number of distorted images	Number of distortion types	Authenticity of distortions	Subjective score type	Mixture of distortions	Published data
LIVE IQA [128]	29	779	5	Synthetic	DMOS [0,100]	N/A	2003
CSIQ [80]	30	866	6	Synthetic	DMOS [0,1]	N/A	2010
TID2008 [114]	17	1,700	17	Synthetic	MOS [0,9]	N/A	2009
TID2013 [113]	25	3,000	24	Synthetic	MOS [0,9]	N/A	2013
LIVE Challenge [45]	N/A	1,162	Numerous	Authentic	MOS [0,100]	✓	2016

3.2 Image quality databases

Different standard databases are available to test the algorithms' performance with respect to the human subjective judgements. Most of them have been created artificially, while few of them contains images affected by mixtures of authentic distortions, as shown in Figure 3.5. A summary of the attributes of the considered databases is shown in Table 3.2.

3.2.1 Synthetic distortions

Databases presented in this section contain distorted images obtained by synthetically introducing a single type of distortion, such as JPEG compression, simulated sensor noise, or simulated blur to pristine images. They have been widely used for the development of older perceptual image quality assessment systems. **LIVE Image Quality Assessment Database.** The LIVE Image Quality Assessment Database[128], which was the first successful public-domain image quality database and is still the widely used, contains a total of 779 distorted images derived starting from 29 reference images by introducing 7-8 degradation levels of five different single distortions: JPEG and JPEG2000 (JP2K) compression, white noise (WN), gaussian blur (GB), and Rayleigh fast-fading channel distortion. Differential Mean Opinion Scores (DMOS) are provided for each image in the range [0, 100], where higher DMOS indicates lower quality.

Categorical Subjective Image Quality (CSIQ) The Categorical Subjective Image Quality (CSIQ) database [80] includes 866 distorted images derived from 30 original images distorted using six different distortions: JPEG, JP2K, WN, GB, pink Gaussian noise, and global contrast decrements; at four to five levels each. DMOS in the range [0, 1] are provided for each image.

TID2008 The TID2008 [114] contains 1,700 distorted images with 17 different distortions derived from 25 reference images at 4 degradation levels. Each image is associated with a Mean Opinion Score (MOS) in the range [0, 9]. Contrary to DMOS, higher MOS indicates higher quality.

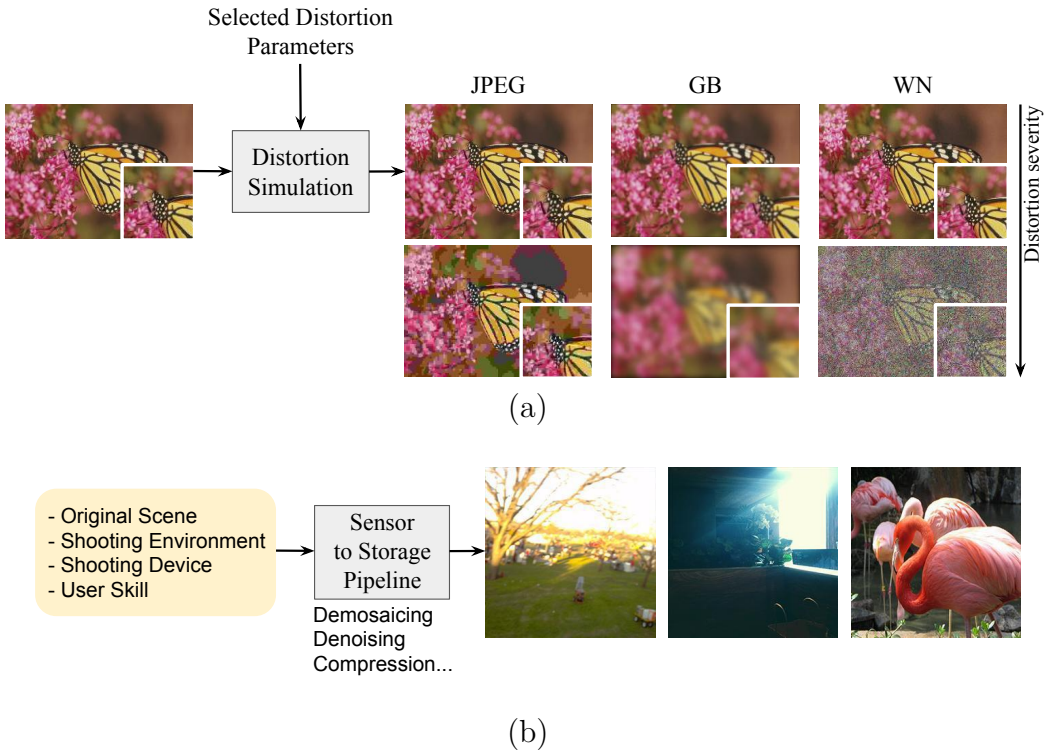


Fig. 3.5 (a) Synthetic and (b) authentic distortions affecting image quality databases.

TID2013 TID2013 [113] includes the largest number of distorted images. It consists of 3,000 distorted images derived from 25 reference images with 24 types of distortions at 5 different levels of distortion. Each image is associated with a MOS.

3.2.2 Authentic distortions

However, as pointed out by Ghadiyaram and Bovik [44]: "images captured using typical real-world mobile camera devices are usually afflicted by complex mixtures of multiple distortions, which are not necessarily well-modeled by the synthetic distortions found in existing databases". The LIVE In the Wild Image Quality Challenge Database [45] contains 1,162 images with resolution equal to 500×500 pixels affected by diverse authentic distortions and genuine artifacts such as low-light noise and blur, motion-induced blur, over and underexposure, compression errors, etc. Database images have been rated by many thousands of subjects via an online crowdsourcing system designed for subjective quality assessment. Over 350,000 opinion scores from over 8,100 unique human observers have been gathered. The mean opinion score (MOS) of each image

is computed by averaging the individual ratings across subjects, and used as ground truth quality score. The MOS values are in the [1, 100] range.

3.3 Evaluation criterions

The different design choices within the proposed approach are compared with a number of leading blind IQA algorithms. Since most of these algorithms are machine learning-based training procedures, following [45] in all the experiments the data are splitted into 80% training and 20% testing sets, using the training data to learn the model and validating its performance on the test data. To mitigate any bias due to the division of data, the random split of the dataset is repeated 10 times. For each repetition the Pearson's Linear Correlation Coefficient (LCC) and the Spearman's Rank Ordered Correlation Coefficient (SROCC) between the predicted and the ground truth quality scores are computed, reporting the median of these correlation coefficients across the 10 splits. LCC is formulated as follow:

$$LCC = \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}, \tag{3.8}$$

where $\bar{\hat{y}} = (1/N)\sum_{i=1}^{N}\hat{y}_i$ is the average of predicted scores and $\bar{y} = (1/N)\sum_{i=1}^{N} y_i$ is the average of the ground-truth scores; while SROCC formula is:

$$SROCC = LCC(\mathrm{rank}(\hat{y}), \mathrm{rank}(x)) = \frac{\mathrm{cov}(\mathrm{rank}(\hat{y}), \mathrm{rank}(y))}{\sigma_{\mathrm{rank}(\hat{y})}\sigma_{\mathrm{rank}(y)}}, \tag{3.9}$$

where $\mathrm{rank}(\hat{y})$ and $\mathrm{rank}(y)$ denote the ranked predicted scores and the ranked ground-truth scores respectively, and $\sigma_{\mathrm{rank}(\hat{y})}$ and $\sigma_{\mathrm{rank}(y)}$ are the standard deviation of the ranked predicted scores and ranked ground-truth scores.

In all the experiments the Caffe open-source framework is used [64] for CNN training and feature extraction, while the LIBLINEAR library [40] is employed for SVR training.

3.4 Experimental results

In this section the performance of each design choice introduced in Section II are evaluated.

Table 3.3 Median LCC and SROCC across 10 train-test random splits of the LIVE In the Wild Image Quality Challenge Database considering only the central crop of the subsampled image as input for the pre-trained CNNs considered.

	LCC	SROCC
Imagenet-CNN	0.6782	0.6381
Places-CNN	0.6267	0.6055
ImageNet+Places-CNN	**0.7215**	**0.7021**

3.4.1 Experiment I: Image description using pre-trained CNNs

The 4096-dimensional features are extracted from the *fc7* layer of the pre-trained ImageNet-CNN, Places-CNN and ImageNet+Places-CNN. Since these CNNs require an input with a dimensionality equal to 227×227 pixels, the original 500×500 images are rescaled to 256×256 keeping aspect ratio, and then the central 227×227 sub-region from the resulting image is cropped out. All the images are pre-processed by subtracting the mean image, that is computed by averaging all the images in the training set on which the CNN was pre-trained. The median LCC and SROCC over the 10 train-test splits are reported in Table 3.3. From the results it is possible to see that ImageNet+Places-CNN outperforms both Imagenet-CNN and Places-CNN, with Places-CNN giving the worst performance confirming our original hypothesis that the more concept the CNN has been trained to recognize, the more effective are its features for modeling generic image content.

3.4.2 Experiment II: feature and prediction pooling strategies

In the previous experiment the resize operation could have reduced the effect of some artifacts, e.g. noise. In order to keep unchanged the distortion level the performances of features extracted from a variable number of randomly cropped 227×227 sub-regions from the original image are evaluated. Given the results of the previous experiment, the only features considered here are those extracted using the ImageNet+Places-CNN.

Three different fusion schemes for combining the information generated by the multiple sub-regions to obtain a single score prediction for the whole image are considered.

The first scheme is feature pooling that can be seen as an early fusion approach, performing element-wise fusion on the feature vectors. The second scheme is feature concatenation, performing information fusion by concatenating the multiple feature

Table 3.4 Median LCC and SROCC across 10 train-test random splits of the LIVE In the Wild Image Quality Challenge Database considering randomly selected crops as input for the ImageNet+Places-CNN and three different fusion approaches: feature pooling, feature concatenation and prediction pooling.

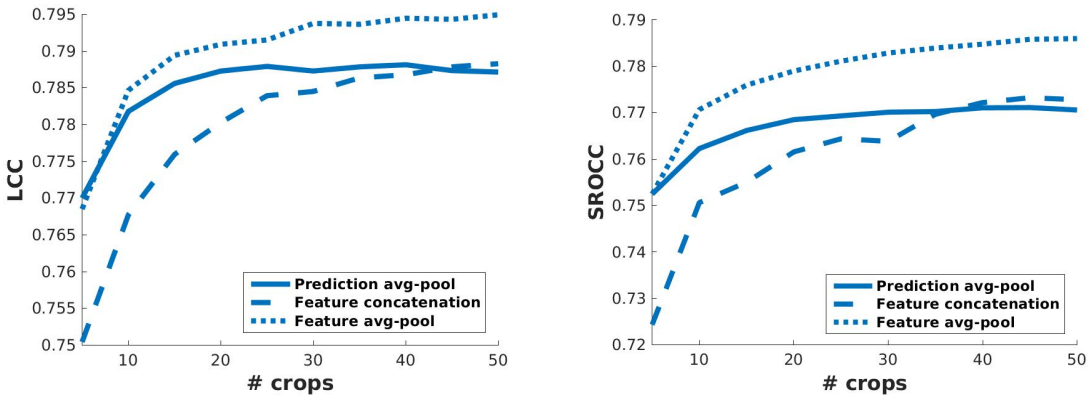
	LCC	SROCC
Feature pooling (avg-pool,@30crops)	**0.7938**	**0.7828**
Feature concatenation (@35crops)	0.7864	0.7724
Prediction pooling (avg-pool,@20crops)	0.7873	0.7685

vectors into a single feature vector. The third scheme is prediction pooling that can be seen as a late fusion approach, where information fusion is performed on the predicted quality scores.

In all the experiments the number of random crops is varied between 5 and 50 in steps of 5. Figure 3.6 shows the plots for LCC and SROCC with respect to the number of crops considered, while the numerical values for the best configurations of each fusion scheme (across pooling operators and number of crops) are reported in Table 3.4. The optimal number of crops has been selected by running the two-sample $t-$test whose results are reported in Figure 3.10. From the plots it is possible to see that feature pooling conveys the best results. Prediction pooling is able to give comparable results with those of feature pooling only when a small number of crops is considered. Finally, feature concatenation gives the worst results, giving comparable results with those of prediction pooling only when a large number of crops is considered. Concerning the best configurations reported in Table 3.4, the output of the two-sample $t-$test shows that the results obtained by feature average-pooling are statistically better than both those obtained by feature concatenation (p-value equal to $3.4{\cdot}10^{-9}$) and prediction average-pooling ($p-$value equal to $8.8{\cdot}10^{-5}$). The difference between feature concatenation and prediction average-pooling is not significative instead ($p-$value equal to 0.23).

3.4.3 Experiment III: Image description using a fine-tuned CNN

In all previous experiments pre-trained CNNs for feature extraction are used. In this experiment instead, the ImageNet+Places-CNN is fine-tuned for the NR-IQA task. The CNN is discriminatively fine-tuned to classify image crops into five distortion classes (i.e. bad, poor, fair, good, and excellent) obtained by crisp partitioning the



(a) (b)

Fig. 3.6 Median LCC (a) and SROCC (b) across 10 train-test random splits of the LIVE In the Wild Image Quality Challenge Database, with respect to the number of image crops given in input to the pre-trained ImageNet+Places-CNN. Three fusion schemes are considered (feature pooling, feature concatenation and prediction pooling), and for each of them only the best configuration over the pooling operators considered is reported.

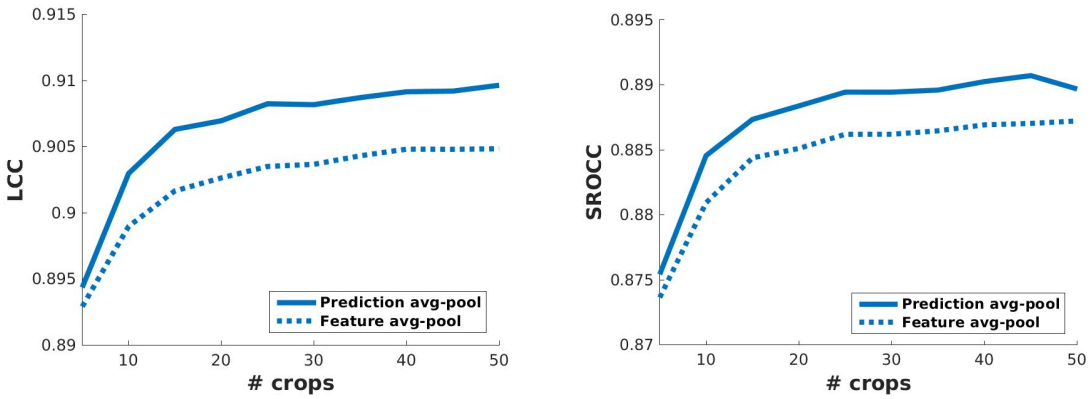
MOS into five disjoint sets. Since the number of images belonging to the five sets is uneven (see Figure 3.9), during training a sample weighting approach [60] giving larger weights to images belonging to less represented distortion classes is used [143, 178]. Weights are computed as the ratio between the frequency of the most represented class and the frequency of the class to which the image belongs. Formally, let us call f_c the frequency of the class C_c, with $c = 1, \ldots, 5$; then the weight w_i for the image I_i can be expressed as:

$$w_i = \frac{\max\limits_{c=1,\ldots,5} f_c}{f_k} \quad \text{where } k = \{c : I_i \in C_c\} \tag{3.10}$$

On the NR-IQA task this weighting scheme gives better results compared to batch-balancing (i.e. assuring that in each batch all the classes are evenly sampled) since it guarantees more heterogeneous batches.

Given the results of the previous experiments, only the performance of the fine-tuned CNN with feature pooling and prediction pooling with the average operator are evaluated. The network is fine-tuned for 5,000 iterations using Caffe framework [64] on a NVIDIA K80 GPU. The total training time was about 2 hours, while predicting the MOS for a single image at test time requires about 20ms.

Figure 3.7 shows the plots for LCC and SROCC with respect to the number of crops considered, while the numerical values for the best configurations are reported



(a) (b)

Fig. 3.7 Median LCC (a) and SROCC (b) across 10 train-test random splits of the LIVE In the Wild Image Quality Challenge Database, with respect to the number of image crops given in input to the fine-tuned CNN. Two fusion schemes are considered (feature average pooling and prediction average pooling).

Table 3.5 Median LCC and SROCC across 10 train-test random splits of the LIVE In the Wild Image Quality Challenge Database considering randomly selected crops as input for the fine-tuned CNN and two different fusion approaches (feature pooling and prediction pooling).

	LCC	SROCC
Feature pooling (avg-pool,@20crops)	0.9026	0.8851
Prediction pooling (avg-pool,@25crops)	**0.9082**	**0.8894**

in Table 3.5. As for the previous experiment, the optimal number of crops has been selected by running the two-sample $t-$test test whose results are reported in Figure 3.11. From the plots it is possible to notice that prediction pooling conveys the best results whatever is the number of crops considered. Concerning the best configurations reported in Table 3.5, the output of the two-sample $t-$test shows that the results obtained by prediction average-pooling are statistically better than those obtained by feature average-pooling (p-value equal to $4.7 \cdot 10^{-4}$).

3.4.4 Comparison with the state-of-the-art BIQ algorithms

In Table 3.6 the results of the different instances of the proposed approach, called DeepBIQ, are compared with those of some NR-IQA algorithms in the state of the art. From the results it is possible to see that the use of a pre-trained CNN on the whole image is able to give slightly better results than the best in the state of the

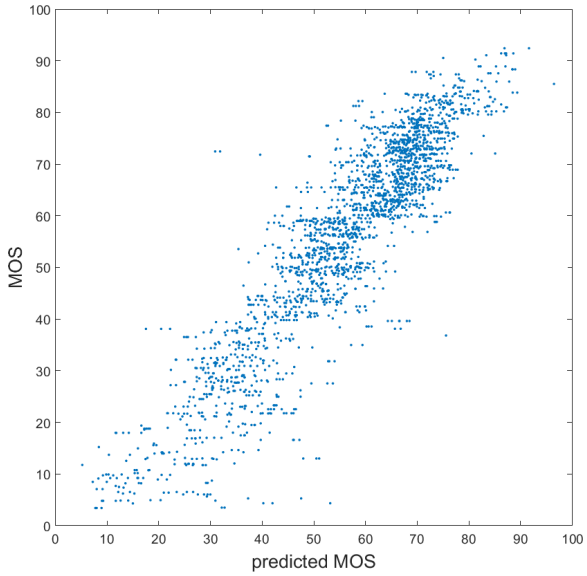


Fig. 3.8 Scatter plot of the MOS predicted by DeepBIQ against the ground truth MOS on the LIVE in the Wild Image Quality Challenge Database.

art. The use of multiple crops with average-pooled features is able to improve LCC and SROCC with respect to the best method in the state of the art by 0.08 and 0.11 respectively. Finally the use of the fine-tuned CNN with multiple image crops and average-pooled predictions is able to improve LCC and SROCC by 0.20 and 0.21 respectively. The scatter plot of the predicted MOS against the ground truth MOS is reported in Figure 3.8. Error statistics may not give an intuitive idea of how well a NR-IQA algorithm performs. On the other hand, individual human scores can be rather noisy. Taking into account that the LIVE In the Wild Image Quality Challenge Database gives for each image the MOS as well as the standard deviation of the human subjective scores, to have an intuitive assessment of DeepBIQ performance the following procedure is employed: the absolute prediction error of each image is divided by the standard deviation of the subjective scores for that particular image. Then a cumulative histogram is built collecting statistics at one, two, and three standard deviations. Results indicate that 97.2% of predictions from the proposed DeepBIQ are below σ, 99.4% below 2σ and 99.8% below 3σ. Assuming a normal error distribution, this means that in most of the cases the image quality predictions made by DeepBIQ are closer to the average observer than those of a generic human observer.

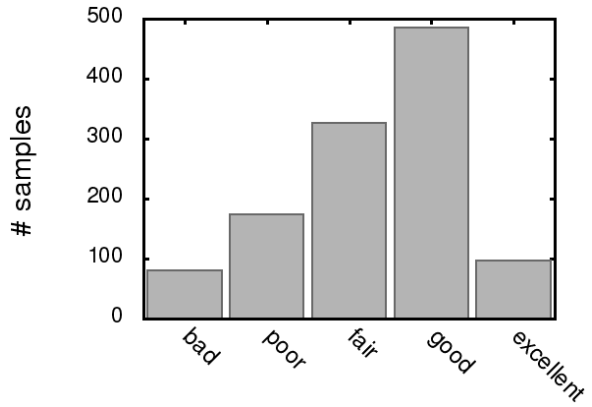


Fig. 3.9 Sample distribution over the five quality grades considered for the LIVE In the Wild Image Quality Challenge Database.

Table 3.6 Median LCC and median SROCC across 10 train-test random splits of the LIVE In the Wild Image Quality Challenge Database.

	LCC	SROCC
DIIVINE [105]	0.56	0.51
BRISQUE [102]	0.61	0.60
BLIINDS-II [123]	0.45	0.40
S3 index [149]	0.32	0.31
NIQE [103]	0.48	0.42
C-DIIVINE [174]	0.66	0.63
FRIQUEE [43, 45]	0.71	0.68
HOSA [162]	-	0.65
DeepBIQ (Exp. I: pre-trained CNN, whole image)	0.72	0.70
DeepBIQ (Exp. II: pre-trained CNN, image sub-regions, feat. avg-pool)	0.79	0.79
DeepBIQ (Exp. III: fine-tuned CNN, image sub-regions, pred. avg-pool)	**0.91**	**0.89**

Table 3.7 Median LCC and median SROCC across 100 train-val-test random splits of the legacy LIVE Image Quality Assessment Database.

Method	LCC	SROCC
DIIVINE [105]	0.93	0.92
BRISQUE [102]	0.94	0.94
BLIINDS-II [123]	0.92	0.91
NIQE [103]	0.92	0.91
C-DIIVINE [174]	0.95	0.94
FRIQUEE [43, 45]	0.95	0.93
ShearletIQM [98]	0.94	0.93
MGMSD [2]	0.97	**0.97**
Low Level Features [75]	0.95	0.94
Rectifier Neural Network [142]	–	0.96
Multi-task CNN [69]	0.95	0.95
Shallow CNN [68]	0.95	0.96
DLIQA [58]	0.93	0.93
HOSA [162]	0.95	0.95
CNN-Prewitt [88]	0.97	0.96
CNN-SVR [87]	0.97	0.96
DeepBIQ	**0.98**	**0.97**

3.4.5 Experiment on benchmark databases of synthetically distorted images

The proposed method is evaluated on these datasets dealing with the different human judgements and distortion ranges by only re-training the SVR, while keeping the CNN unchanged. The experimetal protocol used in [68, 69] is followed. This protocol consists in running 100 iterations, where in each iteration 60% of the reference images and their distorted versions is randomly select as the training set, 20% as the validation set, and the remaining 20% as the test set. The experimental results in terms of average LCC and SROCC values on LIVE are reported in Table 3.7, on CSIQ in Table 3.8, on TID2008 in Table 3.9, and on TID2013 in Table 3.10. From these results it is possible to see that our method, DeepBIQ, is able to obtain the best performance in terms of both LCC and SROCC notwithstanding that differently from the all the other methods reported, the features have been learned on a different dataset containing images with real distortions and not on a portion of the test database itself. Therefore, the results confirm the effectiveness of our approach for no-reference image quality assessment.

Table 3.8 Median LCC and median SROCC across 100 train-val-test random splits of the CSIQ.

Method	LCC	SROCC
DIIVINE [105]	0.90	0.88
BRISQUE [102]	0.93	0.91
BLIINDS-II [123]	0.93	0.91
Low Level Features [75]	0.94	0.94
Multi-task CNN [69]	0.93	0.94
HOSA [162]	0.95	0.93
DeepBIQ	**0.97**	**0.96**

Table 3.9 Median LCC and median SROCC across 100 train-val-test random splits of the TID2008.

Method	LCC	SROCC
DIIVINE [105]	0.90	0.88
BRISQUE [102]	0.93	0.91
BLIINDS-II [123]	0.92	0.90
MGMSD [2]	0.88	0.89
Low Level Features [75]	0.89	0.88
Multi-task CNN [69]	0.90	0.91
Shallow CNN [68]	0.90	0.92
DeepBIQ	**0.95**	**0.95**

Table 3.10 Median LCC and median SROCC across 100 train-val-test random splits of the TID2013.

Method	LCC	SROCC
DIIVINE [105]	0.89	0.88
BRISQUE [102]	0.92	0.89
BLIINDS-II [123]	0.91	0.88
Low Level Features [75]	0.89	0.88
HOSA [162]	**0.96**	0.95
DeepBIQ	**0.96**	**0.96**

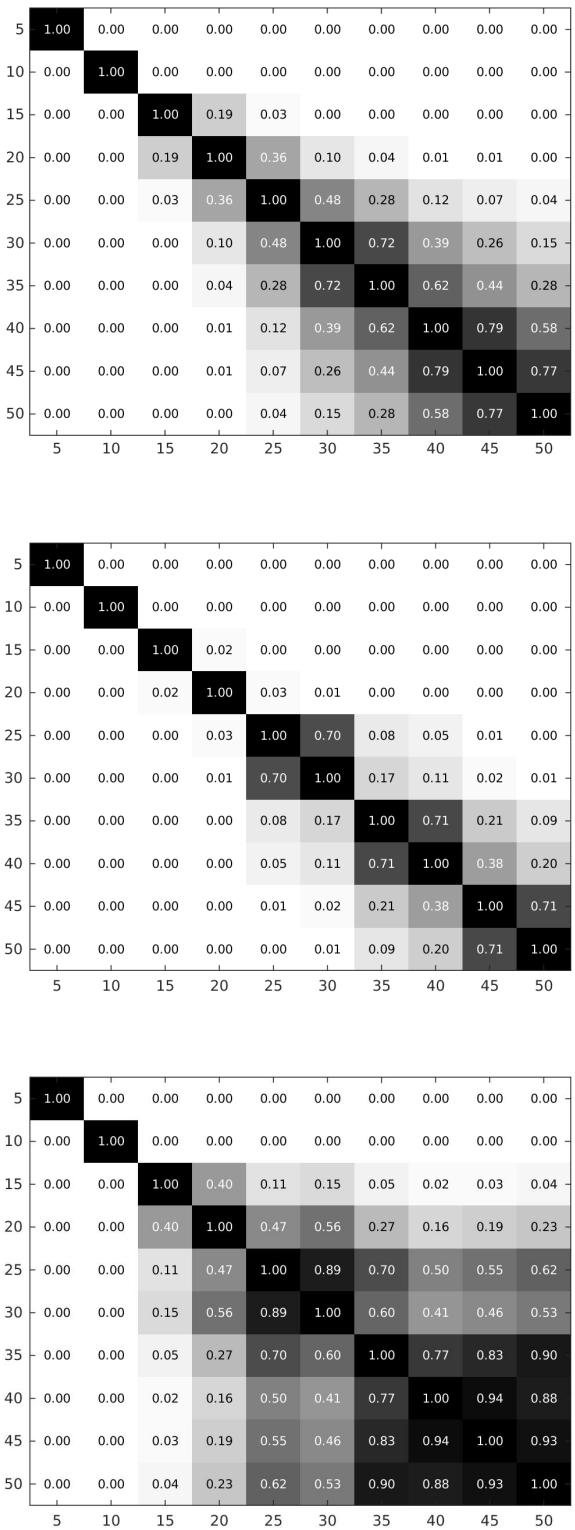


Fig. 3.10 p-values of the two-sample t-test in Experiment II for the different design choices: feature pooling (top), feature concatenation (middle), and prediction pooling (bottom).

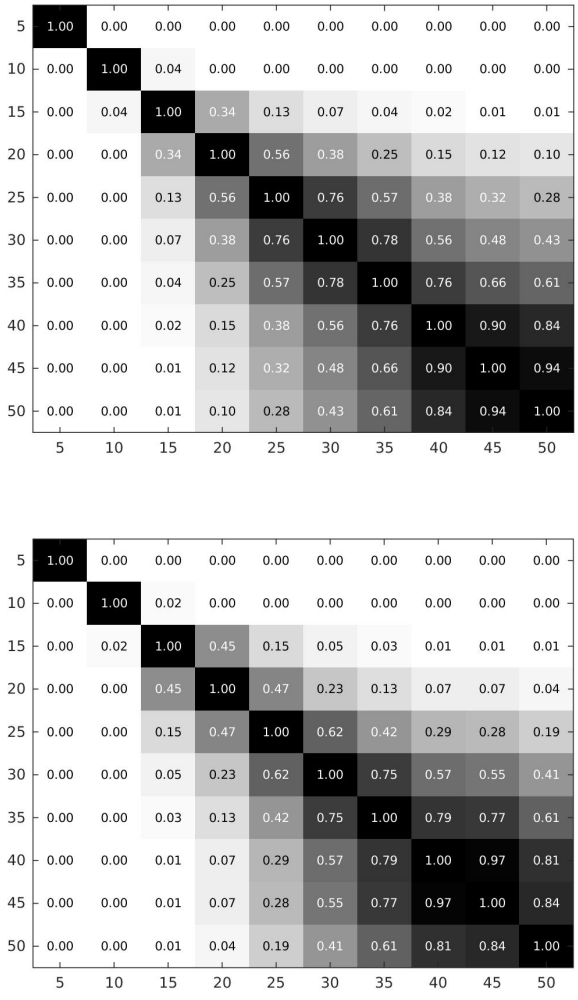


Fig. 3.11 p-values of the two-sample t-test in Experiment III for the different design choices: feature pooling (top), and prediction pooling (bottom).

Chapter 4

General Content Image Aesthetics Assessment and Sentiment Analysis

This chapter focuses on the aesthetics assessment and sentiment analysis problems on general content images. In Section 4.1 image quality aesthetic assessment will be presented and the proposed method will be detailed. Section 4.2 describes the problem of visual sentiment analysis, then it gives an overview of the state-of-the-art methods and finally publicly available databases will be described.

4.1 Image Aesthetics Assessment

The automatic assessment of image aesthetic is a novel challenge for the computer vision community that has wide applications, e.g. image retrieval, photo management, photo enhancement, image cropping, etc. [30, 72]. Because of the subjectivity of humans' aesthetic evaluation, in recent years, many research efforts have been made and various approaches have been proposed [71, 95, 106, 131]. According to the way the problem is formulated, computational approaches can be divided into two groups: aesthetic classification and aesthetic regression. The first group of methods treats aesthetic quality assessment as a binary classification problem, i.e. distinguish between aesthetic and unaesthetic images. Most of these methods have focused on designing features able to replicate the way people perceive the aesthetic quality of images. For example, Datta et al. [29] design special visual features (colorfulness, the rule of thirds, low depth of field indicators, etc.) and use the Support Vector Machine (SVM) and Decision Tree (DT) to discriminate between aesthetic and unaesthetic images. Nishiyama et al. [109] propose an approach based on color harmony and bags of color patterns to characterize color variations in local regions. Marchesotti

et al. [101] demonstrate that generic image descriptors, such as GIST, Bag-of-Visual-words (BOV) encoded from Scale-Invariant Feature Transform (SIFT) information, and Fisher Vector (FV) encoded from SIFT information, are able to capture a wealth of statistics useful for aesthetic evaluation of photographs. Simon et al. [131] show that aesthetic quality depends on context since they obtain more accurate predictions by selecting features for specific image categories. Methods able to learn effective aesthetic features directly from images have been proposed. Lu et al. [95] present the RAting PIctorical aesthetics using Deep learning (RAPID) system, which adopts a Convolutional Neural Network (CNN) approach to automatically learn features for aesthetic quality categorization. Kao et al. [71] train a linear SVM using the features extracted from a CNN pre-trained on ImageNet classification task. The second group of approaches considers aesthetic quality assessment as a regression problem, i.e. they predict an aesthetic rating or score of the images. Datta et al. [29] propose the use of Linear Regression (LR) with polynomial terms of the features to predict the aesthetic score. Bhattacharya et al. [10] propose to use a saliency map and a high-level semantic segmentation technique for extracting aesthetic features subsequently used for training a Support Vector Regression (SVR) machine. Wu et al. [161] design a new algorithm called Support Vector Distribution Regression (SVDR) in order to use a distribution of user ratings instead of a scalar for model learning. More recently, Kao et al. [71] propose a regression model based on CNNs, which achieves the state-of-the-art results on aesthetic quality assessment.

Thesis contribution on this topic is the use of a deep CNN to predict image aesthetic scores. To this end a canonical CNN architecture, originally trained to classify both objects and scenes, is fine-tuned by casting the image aesthetic prediction as a regression problem. Additionally, it is investigated whether image aesthetics is a global or local attribute, and the role played by bottom-up and top-down salient regions [62, 66] to the prediction of the global image aesthetic. For the evaluation the AVA dataset [106] is considered, because it is actually the largest dataset available and the only one providing aesthetic ratings instead of binary classification of aesthetic quality (e.g. “high" or “low”). Experimental results show the robustness of the solution proposed, which outperforms the best solution in the state of the art by almost 17% in terms of Mean Residual Sum of Squares Error (MRSSE).

4.1.1 General content aesthetics database

The Aesthetic Visual Analysis (AVA) dataset [106] is a large-scale collection of images and meta-data obtained from the on-line community of photography amateurs

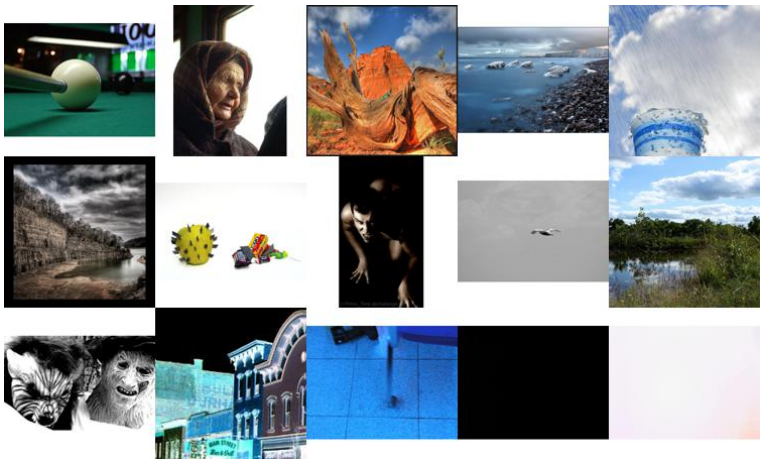


Fig. 4.1 Sample images from the Aesthetic Visual Analysis (AVA) database sorted by their aesthetic score (decreasing from left to right).

and covering a wide variety of subjects on almost 1,000 challenges derived from *www.dbchallenge.com*. Figure 4.1 shows some samples from the AVA dataset. It contains over 255,000 images, both in RGB and grayscale with three types of annotations: aesthetic ratings ranging from 1 to 10; semantic annotations consisting in 66 textual tags describing the semantics of the images; photographic style annotations corresponding to 14 photographic techniques.

4.1.2 Proposed approach for image aesthetic assessment

In this thesis, aesthetic quality assessment is treated as a regression problem because it is closer to the human photo rating process [31]. Given that general content image aesthetics may depend on both the scenes and objects depicted, a pre-trained CNN as generic as possible is chosen for fine-tuning to predict the aesthetic of an unseen image. The network used is the Hybrid-CNN [177], originally trained by merging the scene categories from Places dataset [177] and the object categories from ImageNet [32] for a total of 1,183 different classes; it is a Caffe network architecture [64] (inspired by the AlexNet architecture [76]). The output of the CNN is supposed to be single-real value indicating the predicted aesthetic score, thus before starting the fine-tuning, network architecture is slightly changed: the original last fully connected layer with 1,183 neurons is replaced with a single-neuron layer in order to produce, given an input image, a predicted aesthetic score as a real number ranging between 1 and 10.

The proposed CNN, called *DeepIA*, is obtained by fine-tuning the Hybrid-CNN after replacing the last fully connected with a single-neuron layer and using the Euclidean

loss (as defined in Section 2.2.5, eq. 2.8) instead of the Softmax cross-entropy loss (see Section 2.2.5, eq. 2.5).

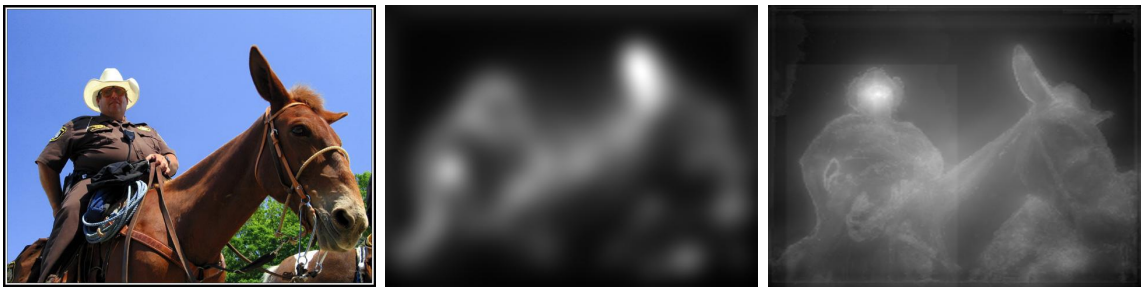
CNN is fine-tuned using Stochastic Gradient Descent (SGD) by chopping and retraining the last fully connected and by slightly updating the weights for the other layers. Batch size of 256 is used, momentum set to 0.9, and a weight decay parameter of 0.0005. Then, the learning rate is initialized to a value of 0.001, and dropped every 20,000 iterations. The model is fine-tuned for a total of 50,000 iterations. In all the experiments the Caffe open-source framework [64] is used for both the CNN training and prediction processes. During the training process, the original images are resized to 256×256 pixels without preserving the aspect ratio and then a random region of 227×227 pixels is extracted from the resized image. This approach increases the training set size in order to avoid overfitting. The mean-pixel value calculated across the training set images is the subtracted from the resized images.

At test time, the original images is first resized to a fixed dimensions and then different design choices are evaluated:

- the images are resized to 256×256 pixels and then the 227×227 pixels central crop is used for image aesthetic prediction;
- the images are resized to 256×256 pixels and then the average the prediction of multiple 227×227 pixels sub-regions (i.e. crops) of the input the image is considered. Ten crops corresponding to the four corners, the center region and their horizontal reflections are taken into account.
- the image pixels are weighted on the basis of their saliency using both a top-down and a bottom-up saliency models. To this end, the saliency map values have been scaled to fit the range $[0, 1]$. The image is then resized to 256×256 and both the central and multiple 227×227 pixels crops are extracted and processed as above. Two different algorithms for estimating salient regions are involved: the Itti et al. [62], which is built upon a biologically plausible computational model of focal bottom-up attention, and the Judd et al. [66], integrating a set of low, mid and high-level image features. In Figure 4.2, the saliency maps predicted by the two considered algorithms are shown.

4.1.3 Evaluation criterions and experimental results

For the experiments the same experimental procedure as [70] is follow. Images whose longest dimension is three times more than the smallest dimension have been discarded,



Original image | Itti saliency map | Judd saliency map

Fig. 4.2 Saliency maps predicted using the Itti et al. [62] and the Judd et al. [66] algorithms on an image of the Aesthetic Visual Analysis (AVA) dataset [106].

resulting in a total of 255,099 images. Among them, 250,129 images are selected for train and 4,970 for test. The average score of user ratings is taken as the images aesthetic quality ground truth. The Mean Residual Sum of Squares Error (MRSSE) is considered for performance evaluation:

$$MRSSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{4.1}$$

where $\hat{y}_i$ is the predicted aesthetic score and y_i is the ground truth of image i. The MRSSE obtained on the AVA dataset by DeepIA approach for the different design choices outlined above, is reported in Table 4.1. The best results are obtained using the average prediction over 10 crops of size 227×227 extracted from the 256×256 image. The second best result is obtained by considering only the central 227×227 crop extracted from the image of size 256 × 256. The use of relatively smaller crops (i.e. 227×227 from 314×314 images) is not able to improve the results, giving a hint that image aesthetic is a global rather than a local attribute. The use of both top-down and bottom-up saliency models to filter out not-salient image content does not help to improve the accuracy of the prediction. In Table 4.2 the best solution proposed is compared with different methods in the state-of-the-art. As a reference, the performance that could be achieved by always predicting an average score of 5 is also reported. From the results it is possible to see that DeepIA outperforms all the methods considered, with a reduction of MRSSE by almost 17% with respect to the best method in the state-of-the-art. In Figure 4.3 the five test images with the smallest MRSSE between ground-truth and predicted aestetic scores are reported. Figure 4.4 reports the ten test images with the largest errors: in the first row we report the top five overestimation errors, while in the second row the top five underestimation errors. The highest errors reported in Figure 4.4 show that sometimes bad predictions reflect a lack

Table 4.1 Performances of aesthetic quality assessment on the AVA dataset.

Method	Image size	#crops	MRSSE
DeepIA+Itti saliency map	256	1	0.5822
DeepIA+Itti saliency map	256	10	0.5766
DeepIA+Judd saliency map	256	1	0.4900
DeepIA+Judd saliency map	256	10	0.4829
DeepIA	314	10	0.4034
DeepIA	256	1	0.3866
DeepIA	256	10	**0.3727**

Table 4.2 Performance comparison of aesthetic quality assessment on the AVA dataset.

Method	MRSSE
Always predicting 5 as aesthetic score	0.5700
BOV-SIFT+rbfSVR ([101] adapted in [71])	0.5513
BOV-SIFT+linSVR ([101] adapted in [71])	0.5401
GIST+rbfSVR ([101] adapted in [71])	0.5307
GIST+linSVR ([101] adapted in [71])	0.5222
Aest-CNN [71]	0.4501
DeepIA	**0.3727**

Fig. 4.3 Top 5 test images with the lowest error between ground-truth (GT) and predicted (PR) aesthetic score.

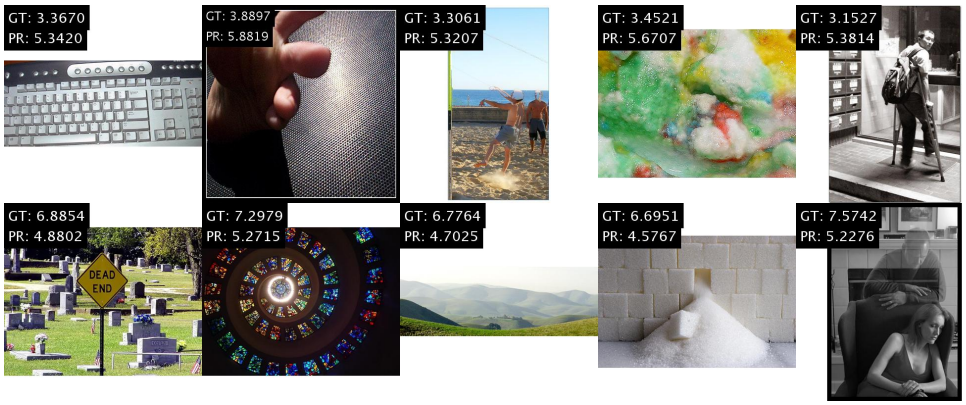


Fig. 4.4 Top 10 test images with the highest error between ground-truth (GT) and predicted (PR) aesthetic score. Test images for which the predicted aesthetic score is overestimated (first row), and images whose predictions are underestimated (second row).

of information consisting in the already defined *aesthetic gap* [31], defined as follows: *The aesthetics gap is the lack of coincidence between the information that one can extract from low-level visual data (i.e., pixels in digital images) and the interpretation of emotions that the visual data may arouse in a particular user in a given situation.*

Finally, since human aesthetic scores are noisy, an analysis of how close is the score predicted by DeepIA with the whole distribution of scores given by the humans to each image is provided. To this end, for each image, the ratio between estimation error and the standard deviation of human scores is measured. The cumulative histogram is reported in Figure 4.5. From the plot it is possible to see that almost 99% of the predictions have an error smaller or equal to a standard deviation value of 1.

4.2 Image Sentiment Analysis

In this section, the problem of sentiment analysis in images is introduced. State-of-the-art methods and available databases will be described.

4.2.1 Introduction

An image is a very effective support for conveying emotions. Through images, people can express their feelings and communicate their opinions. For this reason, recently, understanding the emotion and sentiment from visual content has attracted growing attention. Image sentiment analysis aims to automatically extract the affective content information from visual stimuli. Image sentiment analysis is a very challenging problem

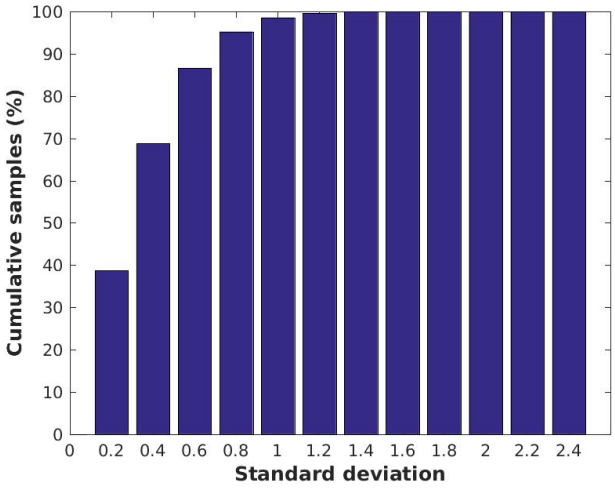


Fig. 4.5 Number of samples (%) with respect to the ratio between absolute estimation error and standard deviations (σ) of human scores.

Fig. 4.6 Common definition of the visual sentiment analysis problem.

due to its high-level of abstraction, to the subjectivity of the human recognition process [65], to the *affective gap*. The affective gap is defined as follow: *The affective gap is the lack of coincidence between measurable signal properties, commonly referred as features, and the expected affective state in which the user brought by perceiving the signal* [52]. To narrow this gap, previous studies have been focused on designing features able to capture high-level semantics related to sentiments in images. In the following section, state-of-the-art methods for image sentiment analysis and image-textual sentiment analysis are introduced.

4.2.2 State-of-the-art methods

Tipically, sentiment analysis can be addressed as a binary classification problem for sentiment polarity prediction (positive/negative) (as shown in Figure 4.6) or as a

multiple classification problem aiming to distinguish among several emotion categories (e.g. amusement, anger, awe, contentment, disgust, excitement, fear, and sadness).

Very few works have been proposed for visual emotion categorization. You [167] proposed a large-scale image dataset for sentiment categorization among 8 classes. They adopted transfer learning strategies: the first consisted in the use of a pre-trained CNN model as feature extractor on top of what they put a support vector machine (SVM) for sentiment categorization; the second exploited the fine-tuning procedure to adapt the same pre-traiend CNN model for sentiment categorization. Experiments have been conducted on the proposed dataset and on other three small dataset in the state-of-the-art.

Many approaches have been proposed for the problem of sentiment polarity estimation. Siersdorfer *et al.* [130] proposed the first algorithm on visual sentiment analysis able to predict the sentiment polarity of images using pixel-level features (i.e. bag-of-visual words and color distribution). Given that sentiment involves high-level abstraction, both [17] and [169] employed visual entities or attributes as mid-level features for image sentiment analysis. In [17], 1200 adjective-noun pairs (ANP), which may correspond to different levels of different emotion categories, were used to collect images from Flickr, then pixel-level features have been used to train 1200 ANP detectors. Finally, the predictions of these 1200 detectors have been used as features for a sentiment classifier. The work in [169] applied a similar mechanism using 102 scene attributes. Deep learning based methods have been proposed. You *et al.* [166] faced the problem of noise training data by proposing a convolutional neural network, proposing a progressive training strategy to fine-tune the deep network. They obtained results on the same set of images in [17] and additionally they collected a new dataset of images from Twitter. Recently, Campos *et al.* [20] fine-tuned a pre-trained CaffeNet for visual sentiment prediction and outperformed the state-of-the-art on the Twitter database. Wang *et al.* [152] proposed the deep coupled adjective and noun neural network (DCAN), consisting in two jointly learned branches (one for adjectives and the other for nouns), that learn middle-level sentiment features then mapped to sentiment polarity. In [165], You *et al.* addressed the problem of visual sentiment analysis by identifying image regions relevant to sentiment prediction. An attention model has been used to learn the correspondence between local image regions and the sentimental visual attributes and, then, a sentiment classifier is built on top of the visual features extracted from the local regions for the final sentiment prediction. Promising results have been shown on the Visual Sentiment Ontology dataset.

There are several methods exploiting both visual and textual information. In [153], Wang *et al.* proposed an unsupervised method for social media images by modeling the interaction between visual and textual information that achieved good results on three large-scale datasets. You *et al.* [168] proposed a cross-modality consistency regression (CCR) scheme for joint textual-visual sentiment analysis: both visual and textual features have been used to learn a regression model. It showed better results both on the state-of-the-art single textual and visual sentiment analysis models and two fusion models. In [146], a large-scale dataset (called Twitter for Sentiment Analysis) of unlabeled tweets (text and images) has been proposed and then has been used for sentiment polarity estimation. A tandem Long Short Term Memory Recurrent Neural Network-Support Vector Machine (LSTM-SVM) architecture has been exploited to classify sentiment polarity of texts. Then images of the tweets, labeled according to the sentiment polarity of the associated text, have been used to fine-tune a pre-trained CNN for sentiment polarity.

4.2.3 Sentiment analysis databases

Several datasets have been proposed for visual emotion analysis. They contain real-world images gathered from image search engines or social networks.

Sentibank. Sentibank [17], also known as visual sentiment ontology (VSO), is the widely-used database for visual sentiment analysis. It contains about one-half million images gathered from Flickr with adjective-noun pairs (ANPs) designed following the Plutchik's Wheel of Emotion (a well known psychological model of human emotions) as queries. The sentiment label of each image is determined by sentiment polarity (positive or negative) of the corresponding ANP.

Twitter. Twitter database [166] is a collection of images extracted from tweets. It contains a total of 1,269 images labeled by employing crowd intelligence to generate sentiment labels. Five Amazon Mechanical Turk (AMT) workers have been recruited for sentiment polarity annotation. Given an image its label corresponds to the unanimous agreed label, i.e. the sentiment label all the five AMT workers gave. Labels for unanimous vote of "at least four agree" and "at least three agree" are also available.

Image Emotion Dataset from the Wild. Image Emotion Dataset from the Wild [167] dataset consists of images collected from image search engines (Flickr and Instagram) using eight emotions (i.e. Amusement, Anger, Awe, Contentment, Disgust, Excitement, Fear, and Sadness) as keywords. The dataset contains a total of about 23,000 images annotated with one of the eight emotion categories used for queries. Ground-truth is human-annotated using AMT.

Twitter for Sentiment Analysis. Twitter for Sentiment Analysis (T4SA) [146] is a recent database composed by a little less than a million tweets, corresponding to 1.5M images. Ground-truth for sentiment polarity is obtained by using machine-based annotations and by selecting only predictions higher than a desired confidence. The resulting dataset without near-duplicates has a total of 974,053 images.

Chapter 5

Portrait images aesthetic assessment

This chapter is about the specific case of image aesthetic assessment, described in the previous chapter, for portrait images. In particular, the problem is first introduced, an analysis of previous methods is reported and a solution involving the combination of visual attributes (i.e. quality and aesthetics of general content images) and of facial attributes (i.e. smiling, hair style, makeup) is developed. Facial attributes estimation is addressed by proposing two methods: a smile detector and a multi-task model.

5.1 Face aesthetics

One of the most common visual content and powerful channel of non-verbal communication is face [8, 159]. A large percentage of images on many photo sharing platforms contains faces, self portraits, or "selfies". An automatic system providing a feedback about facial images is interesting and useful. In fact, an automatic system for portrait aesthetics assessment might sort and edit portrait images, guide into the enhancement of their visual aspects or select a few images from an entire collection.

The prediction of the overall aesthetics of a facial image is the result of the combination of several features encoding relevant information about the global image aesthetics adapted to facial pictures as well as information related to facial expressions and high-level attributes (e.g. smile, age, gender, hair style). It should be clear that face aesthetics is somehow related but is different from facial beauty: the first reflects the attractiveness of a portrait image, instead the second represents the attractiveness of face itself.

The face aesthetics prediction in images is addressed by catching relevant information about quality and aesthetics using methods already described in previous chapters as well as information related to facial attributes that might encode relevant information about face to guide aesthetic assessment.

5.2 Previous works

Aesthetic assessment of portrait images is a challenging task and few approaches exist. Males *et al.* [99] presented the first work on aesthetic quality assessment of head-shots. A support vector machine for binary classification (non-appealing or appealing) have been trained by combining low-level (e.g. contrast and hue count of the whole image) and high-level features (e.g. sharpness and blown-out highlights only of a facial region). Experiments have been carried out on a set of photo collected from Flickr and manually labelled by five people as being aesthetically appealing or not. Unfortunately their database is not publicly available.

Lienhard *et al.* spent a lot of effort on aesthetic assessment of portrait images [90–92]. In [92] proposed a new database, called Human Faces Score (HFS), and developed a method based on image segmentation. More in detail, the input image is segmented in several regions (hair, shoulders, skin, and background) and features (blur, color count, illumination, and saturation) are then computed in each region. Results have been reported both for binary classification (non-appealing or appealing portrait image) and regression (aesthetic score estimation). Recently, in [90, 91] other features, selection strategies both for features and regions, and classifiers have been introduced. The proposed algorithm outperformed state-of-the-art approaches on HFS database, results have been reported also for images containing faces gathered from databases originally developed for general content aesthetic assessment.

5.3 Portrait images datasets

In this section the publicly available databases for portrait image aesthetics are described. Available databases consist of images containing people or groups of people gathered from online photo databases or photo sharing websites (e.g. Flickr, DPChallenge[1]). Given that these photos are collected in real scenarios they present a wide range of subjects, facial appearance, illumination and imaging conditions.

[1] www.dpchallenge.com

Fig. 5.1 6 pictures of 3 people from the Human Faces Scores (HFS) database. On each row images are sorted from low aesthetics to high aesthetics.

Fig. 5.2 Samples from the Face Aesthetics Visual Analysis (FAVA) database sorted by their aesthetic score (increasing from left to right).

Human Faces Scores (HFS). The Human Faces Scores (HFS) [92] database contains 250 images of head-shots, well known as *selfies*. Specifically, 7 images of 20 different people, and 110 additional portrait images have been collected. Examples of images for 3 particular people are given in Figure 5.1. Each image has been rated by 25 human observers on a scale with values ranging between 1 and 6 (6 means the highest quality).

Face Aesthetics Visual Analysis. The Face Aesthetics Visual Analysis (FAVA) database is a subset of the AVA database [106] (already described in Section 4.1.1) containing various images with faces. Each picture is associated with a value between 1 and 10 (10 means highest quality) corresponding to the average of around 210 collected individual scores. Samples are shown in Figure 5.2.

Fig. 5.3 Samples from the Flickr database sorted by their aesthetic score (increasing from left to right).

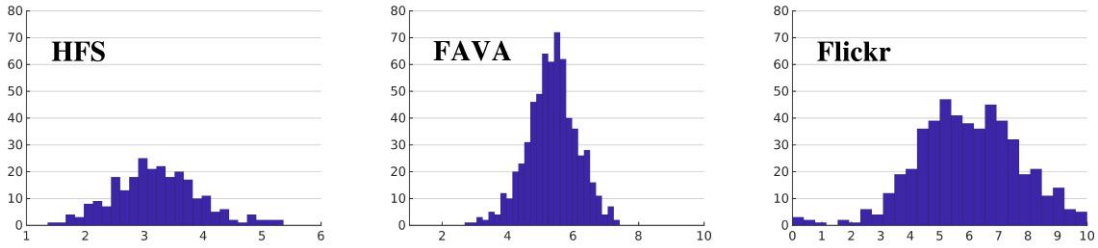


Fig. 5.4 Histograms of ground-truth scores for HFS, FAVA and Flickr databases.

Flickr database. The Flickr database has been gathered on Flickr for general aesthetic assessment [86]. It consists of 500 images associated to a ground-truth score between 0 and 10 (10 means high quality). Photos are either portraits or group of faces. According to [90] only the biggest detected face is considered in each picture. Figure 5.3 shows samples from the database.

The histograms in Figure 5.4 show that HFS presents a higher variance than FAVA, for which there is a lot of images with medium scores. This latter aspect makes the learning step more difficult for the FAVA database, since few samples characterize very low/high aesthetics, prediction performance is likely to be lower for FAVA than for HFS. Since Flickr does not contain only frontal and centered faces but also group portraits, the prediction performance may also be lower than for HFS.

5.4 Facial attributes description

A detailed face description can encode relevant aspects to guide aesthetic estimation. In fact, it includes a wide variety of information related to person's identity, demographic

attributes (gender, age, and ethnicity), mood (facial expressions), and visual attributes (e.g. hair style clothing, face shape). In the last decades, algorithms have been developed for addressing problems such as face recognition and verification [11, 73, 141], facial expressions recognition (e.g. smile detection [15], pain assessment [22]), landmark estimation [175] and facial attributes estimation. These algorithms are used for several applications including video surveillance [23], face retrieval [78, 171] and social media [115]. Depending on the number of aspects simultaneously addressed, existing approaches might be grouped into single attribute and multiple attributes methods. In the following sections a smile detector robust to image distortions is proposed and a multi-task learning approach for multiple facial attributes estimation is presented.

5.4.1 Single face attribute estimation

In this section the smile detection problem is addressed. Smiling is one of the most significant facial attributes [117] and, among facial expressions, is one of the most basic, common and useful in a person's day life [39]. Smiling is an expression denoting happiness, pleasure, satisfaction, or amusement. It is characterized by the upward movements of the lip corners and of the cheeks. In the framework of the Facial Action Coding System (FACS) [79], smile can be seen as the combination of the facial muscles corresponding to the Action Unit six and twelve (AU6 and AU12).

The first works on smile detection used databases taken under constrained laboratory environment. For example, Shinohara et al. [129] used Higher-order Local Auto-Correlation (HLAC) features and Fisher Weight Map (FWM) for facial expression recognition and smile detection and achieved good performance on their own database consitsing of only four people. Bai et al. [7] extracted Pyramid Histogram of Oriented Gradients (PHOG) features from the region of the mouth on the Cohn-Kanade AU-Coded Facial Expression Database.

The first comprehensive work for smile detection in unconstrained scenarios was proposed by Whitehill et al. [157]. At the same time they also made publicly available a new dataset (GENKI) with contents from the web for smile detection in the wild. Using this dataset, Shan [126, 127] proposed a very efficient smile detection approach by simply comparing intensities of a few pixels in a face image [7]. Zhang et al. [173] demonstrated the effectiveness and efficiency of Mouth Features (MF) for smile detection.

More recently, An et al. [5] proposed a fully automated smile detection approach. They showed that adopting three popular feature descriptors (Local Binary Patterns (LBP) [1], Local Phase Quantization (LPQ) [111] and Histogram of Oriented Gradients

(HOG) [33]). They achieved the best results on both the GENKI-4K database and their own collected MIX databases. Gao et al. [42] proposed a semi-automated smile detector, which achieved the best performance on the GENKI-4K database using a combination of features (Raw pixel values, HOG and Self-Similarity of Gradients (GSS)) combining multiple classifiers.

A fully automated approach for smile detection in digital images is presented. It can be used in several applications such as photo selection and shutter control in digital cameras, that operate under a wide range of imaging conditions, such as variations illumination, face pose, occlusion, ethnicity, gender, age, etc. According to the proposal, the input image is processed in order to detect faces using a face detector inspired by Farfade et al. [41]. The faces are then aligned using an eye-based approach using a facial landmarks detector [6] that does not require any manual labeling. Then a Convolutional Neural Network (CNN) is exploited to predict smiling of the detected faces. The CNN architecture has been designed to be trained even when the amount of learning data is limited. The performance of the proposed pipeline is evaluated on the GENKI-4K database [59], the only publicly available dataset in unconstrained scenarios. The proposed pipeline achieves very good results in smile detection accuracy and is more robust to various image distortions and transformations in comparison with the state of the art.

5.4.1.1 Smile detection database

The GENKI-4K database [59] is used for performance evaluation. It is the most challenging and largest available database for the smile detection task in the unconstrained scenario. It contains 4000 facial images of a wide range of subjects with different ethnicity, age, facial appearance, pose, illumination and imaging conditions. All the images are labeled by human coders, 2162 images are labeled as smile and the remaining 1828 images are labeled as non-smile. Although a few images are, probably, incorrectly labeled no change to the groundtruth labels is done. Figure 5.5 shows some typical images of the GENKI-4K database.

5.4.1.2 Smile detection using convolutional neural network

In this section the complete processing pipeline used to classify smiling faces from the whole image using a CNN is outlined. The main steps of the pipeline are shown in Fig. 5.6. Given an image, the faces are first detected and then aligned by fixing the eyes position. Then a CNN is used to understand whether it is a smiling face or not.

Fig. 5.5 Sample typical images from GENKI-4K database. Smiling faces (top) and non-smiling faces (bottom) are shown.

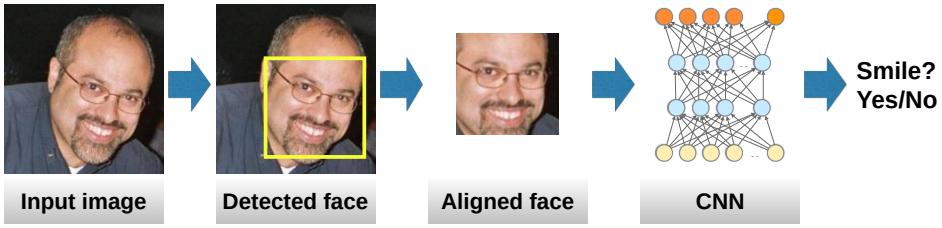


Fig. 5.6 Outline of the proposed method for smile detection. Faces are detected from the original images, then an eye-based face alignment step is performed. Finally facial images are rescaled to a common size and classification is performed using a CNN.

Given an image, the faces are detected using a multi-view face detector inspired by Farfade et al. [41]. The detected faces are aligned fixing the eyes position and then rescaled to a common size. More in detail, the (x, y) coordinates of 49 facial landmarks are obtained using the publicly available implementation of Chehra [6]. Among the detected landmarks, only the two landmarks corresponding to the eyes corners location are considered. These are used in the applied eye-based face alignment method, which consists in fixing the eyes corners distance to 85 pixels using an affine transform matrix, which is composed only of rotation and scaling. Facial images are then obtained by cropping and scaling the transformed images to 36×36 pixels.

Given the cropped and aligned 36×36, a central 32×32 patch is extracted and given as input to a CNN to classify it as smile or non-smile. Different CNN configurations are tested. They are designed to be trained even when the amount of labeled data is limited. The CNN configurations evaluated are summarized in Table 5.1, one per column. In the following the CNNs are referred by their names (A-C). In Table 5.2 the number of parameters for each configuration is reported. The number of weights in configuration C net is larger than the others because of the fully-connected layer. The difference between configuration A and configuration B is that the latter uses two 3×3 convolutional layers instead of a single 5×5 layer. In this way, two non-

Table 5.1 CNN configurations investigated for smile detection (shown in columns). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

CNN Configuration		
A	B	C
4 weight layers	5 weight layers	5 weight layers
input (32 x 32 RGB image)		
conv3-32	conv3-32	conv3-32
maxpool		
LRN		
conv5-32	**conv3-32** **conv3-32**	conv5-32
avgpool		
LRN		
conv5-64	conv5-64	conv3-64
avgpool		
		FC-1024
FC-2		
soft-max		

Table 5.2 Number of parameters of the different CNN configurations considered (see Table 5.1).

Network	A	B	C
# of parameters	79,712	72,544	1,093,472

linearities instead of a single one are incorporate, which make the decision function more discriminative and the CNN deeper. Furthermore, the use of two smaller filters, also decreases the number of parameters from 79,712 (configuration A) to 72,544 (configuration B). In fact, assuming both input and output of the two configurations have C channels, $5^2C^2 = 25C^2$ parameters for a single 5×5 convolutional layer are required; instead, $2(3^2C^2) = 18C^2$ parameters for two-layer 3×3 convolutional stack. Thus, the use of two smaller filters can be considered as a regularization approach (with injected non-linearity). Unlike configuration A, configuration C contains one more fully-connected layer with 1024 neurons before the *FC-2*. In this way, the global properties of previous convolutional layer before the other fully-connected layer are captured.

CNNs are trained from scratch using Stochastic Gradient Descent (SGD) with a batch size of 256, momentum set to 0.9, and a weight decay parameter of 0.002. The learning rate is initialized to a value of 0.001, and drop it by a factor of 10 every 6000

iterations. The model is trained for a total of 30000 iterations. Data augmentation is applied by generating image translations and horizontal reflections: five random 32×32 patches as well as their horizontal reflections from 36×36 facial images. This increases the size of the training set by a factor of 10. The Caffe [64] library is used for CNNs training.

5.4.1.3 Performances evaluation and results

In the experiments, 4-fold cross-validation is performed on the GENKI-4K dataset, meaning that the dataset is randomly partitioned into four subsets. For each round of cross-validation a subset is used for testing and the other three subsets as training. Results are reported in terms of average accuracy over the four rounds of cross-validation.

The prediction made by the CNNs's softmax is computed cropping the central 32×32 patch from the 36×36 facial image. In addition to this single patch prediction, the prediction oversampling the facial image is computed: in this case the prediction is made by considering five 32×32 patches (the four corner patches and the center patch) as well as their horizontal reflections, and averaging the predictions made by the CNNs's soft-max layer on the ten patches.

The performances obtained combining the predictions of the three proposed CNN configurations and the influence of the face alignment step on the overall accuracy are investigated.

The average accuracy of the different instantiations of the proposed pipeline are reported in Table 5.3. From the results it is possible to notice that using a single CNN the best results are obtained with CNN-A and using face alignment. It can be seen that performance can be slightly improved by oversampling the input image and combining the predictions of different CNNs.

At the time the experiments have been performed, the best performance on GENKI-4K database was obtained by Gao et al. [42], who exploiting a semi-automatic procedure (i.e. manual face alignment) report an average accuracy of 94.61%. For sake of comparison their method is therefore reimplemented within the proposed processing pipeline. The comparison with other fully automatic smile detection methods in the state of the art [127, 5, 173, 42] is reported in Table 5.4. It is possible to see that the proposed method is able to outperform the best method in the state of the art, i.e. the reimplementation of Gao et al. [42] in the presented pipeline, by 2.15%. Concerning the proposed method, some examples of misclassified images are reported in Figure 5.7 and 5.8. Figure 5.7 depicts same faces labeled as non-smile in the database that the

Table 5.3 Smile detection accuracy results using the proposed CNN configurations.

CNN config. (see Table 5.1)	Accuracy (%)	
	Without face alignment	With face alignment
A	92.60	93.13
B	92.18	92.80
C	92.70	92.75
A (oversampled)	90.45	93.35
A+B+C (oversampled)	92.53	93.77

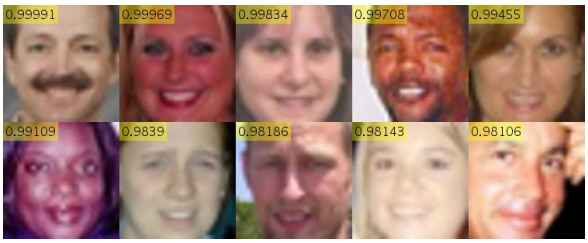


Fig. 5.7 Face labeled as non-smile in the GENKI-4K database that the CNN-A classifies as smile. Images are reported in order of decreasing confidence p(smile).

CNN-A classifies as smile. Instead, Figure 5.8 reports some examples of faces labeled as smile in the database that are classified as non-smile by the proposed approach. From these images it is possible to see that some classification errors are due to incorrect labels in the dataset. Apart from these misclassifications, the greatest source of error is due to very bad facial landmarks localization, as shown in Figure 5.9. In the following section, the robustness of the CNN to bad face alignment and image distortions is therefore investigated.

Imprecise face alignment can be caused both by inaccurate face detection and bad facial landmarks localization. As seen in Table 5.3, the removal of the face alignment step causes a drop in performance for all the CNN configurations investigated. The

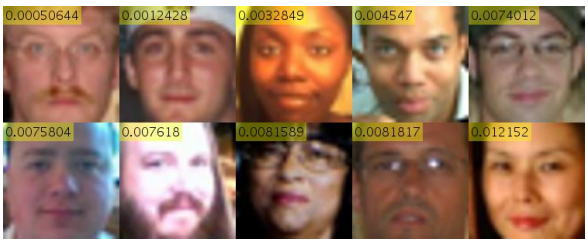


Fig. 5.8 Face labeled as smile in the GENKI-4K database that the CNN-A classifies as non-smile. Images are reported in order of increasing confidence p(smile).

Table 5.4 Comparison with state-of-the-art methods on the GENKI-4K database.

Method	Features	Classifier	Accuracy (%)
An et al. [5]	HOG	ELM	88.50
Zhang [173]	Mouth Features	AdaBoost	89.21
Shan [127]	Pixel difference	AdaBoost	89.70
Gao et al. [42]	Raw pixels+HOG+Self-Similarity of Gradients (GSS)	Linear SVM	91.20
Proposed	CNN	CNN	**93.35**

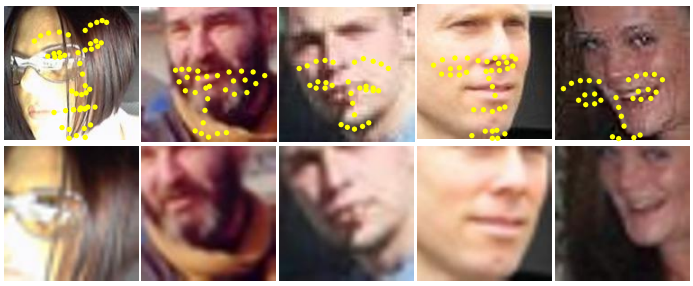


Fig. 5.9 Some misclassified examples caused by bad alignment. The first row shows the faces before alignment with the detected facial landmarks overlaid; the second row shows the faces after alignment.

same is true also for the best algorithm in the state of the art, i.e. Gao et al. [42], whose average accuracy without face alignment drops to 87.78%. To investigate this issue, given the aligned cropped faces of the GENKI-4k database, a new dataset is created by applying some geometric transformations on the 36×36 facial images. These are: rotation of the face around its center with different angles ($-30°$, $-20°$, ..., $30°$); scaling with different scale factors (0.80, 0.90, ..., 1.20); translation with various pixel offsets (-8, -6, ..., 8). For all the transformations, zero-padding is used for pixels falling outside the 36×36 image window. Table 5.5 summarizes the settings for all of the chosen transformations.

A set of three experiments are conducted considering a single geometric transformation at a time. The transformed images are classified using the (transformation-free) trained CNN-A. The results of the performed experiments are reported in Figure 5.10. In the same plots, the results obtained by the proposed implementation of the method by Gao et al. [42] are also reported. From the plots it is possible to notice that CNN-A shows a very high level of robustness against scaling. The performance remains almost unaltered except when the object of interest is small. Regarding translation and rotation the CNN shows a lower level of robustness, with performance significantly decreasing respectively for offsets larger than 5-10 pixels and for rotation angle larger than 10-20 degrees. The comparison between results by the proposed method and

Table 5.5 Types and ranges of the geometric transformations applied to the original images to simulate a bad alignment.

Type	Amount
Rotation	Angle $-30°, -20°, \ldots, 30°$
Scaling	Factor $0.80, 0.90, \ldots, 1.20$
Translation	Offset $-8, -6, \ldots, 8$

Table 5.6 Types and ranges of the distortions applied to original images.

Type	Amount
JPEG compr.	Quality $99\%, \ldots, 0\%$
Gaussian noise	Zero-mean $\sigma^2 = 0.01, 0.02, \ldots, 0.06$
Gaussian blur	Filter size $3 \times 3, 9 \times 9, \ldots, 25 \times 25$ $\sigma^2 =$ Filter size$\times 0.25$
Motion blur	Pixel length $5, 10, \ldots, 30$ Angle $45°$

those by Gao et al. [42] shows a similar trend for the robustness to scale changes, while the presented method results more robust to rotations and translations.

Images available to consumers usually undergo several stages, namely acquisition, compression, transmission and reception, and they may suffer multiple distortions [63]. In this set of experiments the robustness of the proposed CNN with respect to four of the most common image artifacts in real-world digital photos is tested. The considered artifacts are: JPEG compression at different quality indexes ($99\%, \ldots, 0\%$); Gaussian noise with zero-mean and different variances ($\sigma^2 = 0.01, 0.02, \ldots, 0.06$); Gaussian blur varying the filter size ($3 \times 3, 9 \times 9, \ldots, 25 \times 25$) and the variance (corresponding to the filter size multiplied by 0.25); Motion blur with fixed angle ($45°$) and different pixel lengths ($5, 10, \ldots, 30$). The settings for all kinds of image artifacts are summarized in Table 5.6.

Two different experiments are considered: in the first one a single artifact at a time is considered; in the second one images are corrupted by multiple artifacts together. For both the experiments, artifacts are applied on the detected faces after the alignment step. Faces are classified at the increase of the strength of the artifacts using the (distortion-free) trained CNN-A. The results of the single-artifact experiment are reported in Figure 5.11. In the same plots the results obtained by the reimplementation of the method by Gao et al. [42] are also reported. From the plots it is possible to notice that CNN-A shows a very high level of robustness against JPEG compression and Gaussian noise. In both cases the performance remain almost unaltered even for

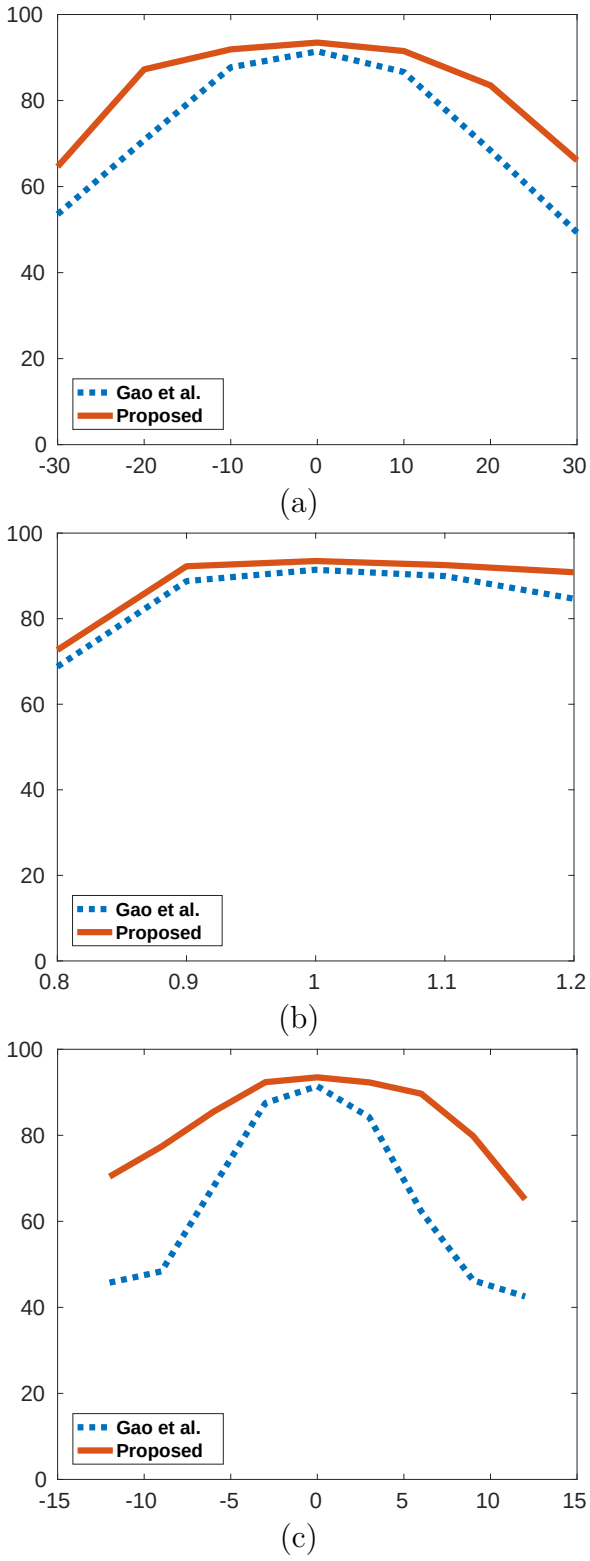


Fig. 5.10 Classification rates varying the rotation angle (a), the scaling factor (b), and the translation offset (c).

large distortion levels. Concerning Gaussian and Motion blur the CNN shows a lower level of robustness, with performance decreasing for filter size and pixel length larger than 10. In comparison with the method by Gao et al. [42] it is possible to see a very similar behavior against JPEG compression and a higher robustness to Gaussian noise and Motion blur. For what concerns Gaussian blur an inversion in performance can be noticed in fact the method by Gao et al. [42] has higher robustness for large filter sizes.

For what concerns the multiple-artifacts experiment, the robustness of the proposed pipeline at six different distortion levels obtained by combining blur, noise, and JPEG compression is tested. An experiment considering a single distortion level at a time for each face is conducted. Specifically, artifacts are applied in the same order they generate in typical imaging pipelines [12]: Motion blur varying pixel lengths (5, 10, 15, 20, 25, 30) and fixed angle 45°, Gaussian noise with zero-mean and different variances (σ^2 = 0.01, 0.02, 0.03, 0.04, 0.05, 0.06), and JPEG compression at different quality indexes (95%, 75%, 60%, 40%, 20% and 0%). In total six different distortion levels are considered. These can be divided into three distortion groups: low distortion (levels 1 and 2), medium distortion (levels 3 and 4), and high distortion (levels 5 and 6). Some samples of face crops after the application of the multiple artifacts at the six distortion levels considered are reported in Figure 5.12. Figure 5.13 shows the results of the performed experiment both on the presented pipeline and the reimplementation of the method by Gao et al. [42]. From the plots it is possible to see that the proposed method has a higher robustness with respect to image artifact for all distortion levels except for the highest one, where the difference between presented method and that by Gao et al.[42] is less than 1%. For intermediate distortion levels the accuracy of the presented method is higher than that achieved by Gao et al. [42], with an improvement higher than 9% for distortion levels from 1 to 5 (with a peak 13.6% improvement for distortion level 3). In this section the effect on the classification accuracy of adding images corrupted by artifacts in the CNN training set is evaluated. Artifact-affected images belonging to one of the three aforementioned distortion groups at a time are added, and classification robustness across all the six distortion levels considered is measured. As for the previous experiments, the CNN-A architecture is used. Two different training setups are considered: in the first one the already trained CNN-A is fine-tuned [116], while in the second one the CNN is trained from scratch. For both the setups, the distortion-free training set is increased by introducing images affected by artifacts belonging to one of the aforementioned distortion groups at time.

The CNN-A is fine-tuned by chopping and retraining from scratch the *FC-2* layer. The same parameters used for training (batch size equal to 256 and momentum 0.9)

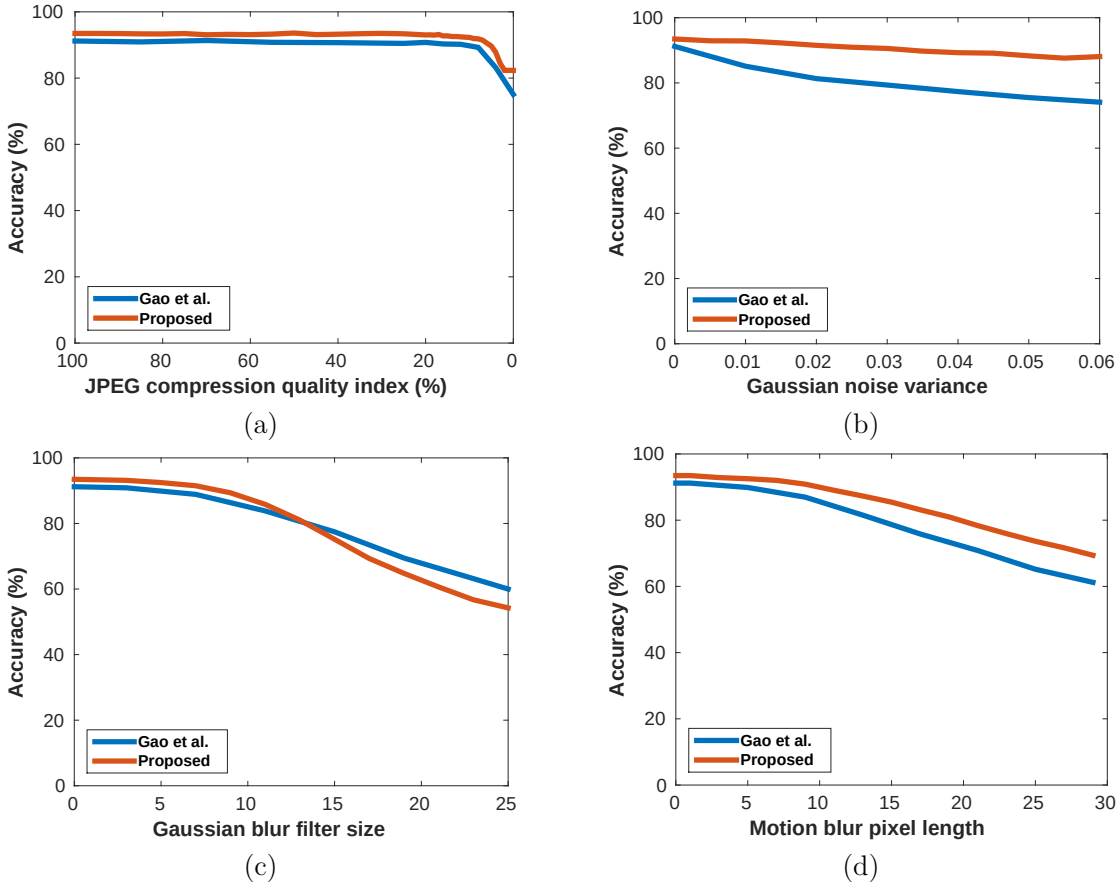


Fig. 5.11 Classification rates varying (a) the JPEG quality index, (b) the variance of zero-mean Gaussian noise, (c) the filter size of Gaussian blur, (d) the pixel length of Motion blur.

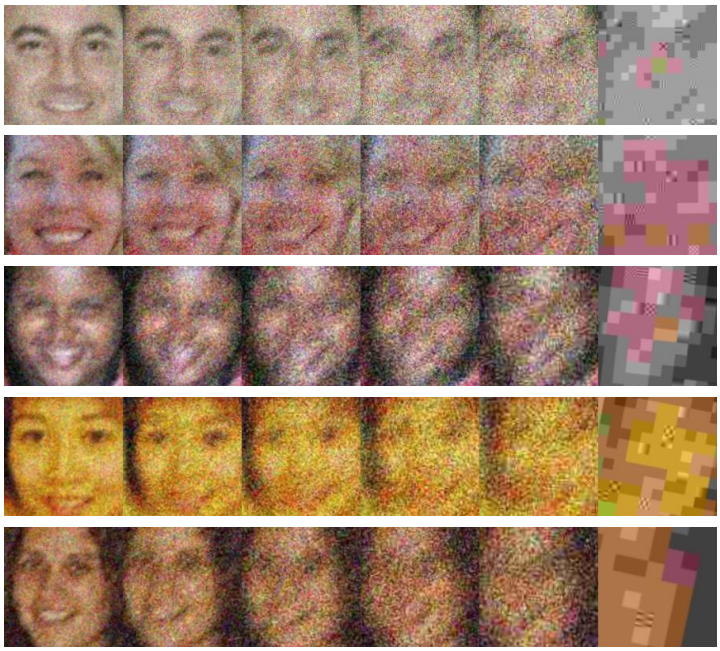


Fig. 5.12 Some samples of face crops after the application of a combination of the three artifacts (Motion blur, Gaussian noise and JPEG compression) at the six distortion levels considered.

except the starting learning rate set to 1e-4, the weight decay parameter set to 2e-4 and the total number of iterations set to 15000 are used for fine-tuning.

Results are reported in Figure 5.14. From the plots is highlighted that:

- adding images with artifacts does not affect the performance on distortion free images and on images with low distortion levels, showing the robustness of the CNN to such training data.

- Adding distorted images in the training set is able to increase robustness with respect to low distortion levels up to 2.7% for both fine-tuned and trained CNNs.

- Robustness increases up to 7.3% and 8.3% for medium level distortion levels for fine-tuned and trained CNN respectively.

5.4.2 Multiple face attributes estimation

With respect to single-task learning based methods, where each task is addressed separately, ignoring any correlations between tasks, MTL based methods enable to learn shared representations [21].

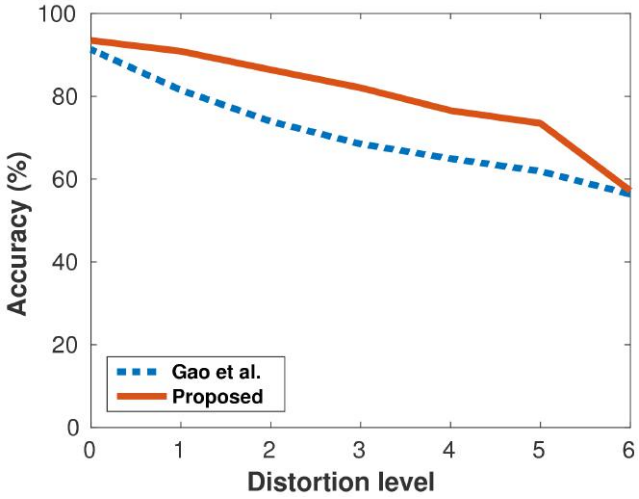


Fig. 5.13 Classification rates applying a combination of three artifacts (Motion blur, Gaussian noise and JPEG compression) with various distortion levels on the original images.

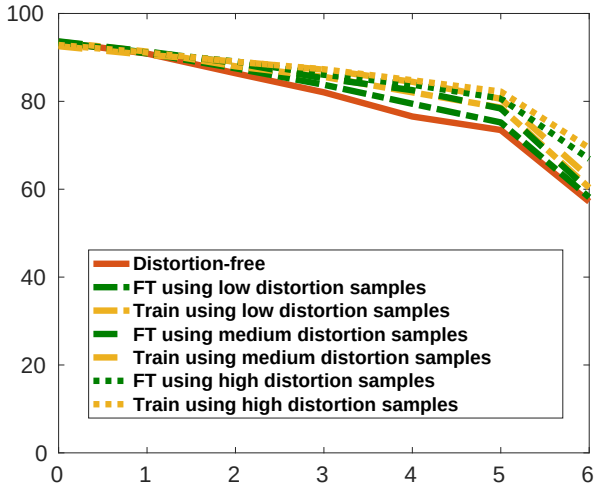


Fig. 5.14 Classification rates of the CNN-A including images with low (green and yellow solid lines), medium (dashed lines), and high distortion levels (dotted lines) in the training set.

A multi-task learning framework based on a convolutional neural network (MTL-CNN) is proposed. It can simultaneously predict soft biometrics (age and gender) and attributes such as hair colors and styles, types of beards, eyes colors etc. The algorithm has shown promising results on several publicly available datasets. As previously mentioned, high-level attributes (smile, presence of make-up, age) convey relevant information about facial image aesthetics.

5.4.2.1 Face attributes databases

Given that there are no public databases containing ground-truth for all the considered facial attributes, for the evaluation of the proposed end-to-end multi-task convolutional neural network three of the widely used publicly available face databases for face attribute estimation are used.

The Adience benchmark. The Adience benchmark [38] is a database designed for age and gender classification. Images were collected from Flickr uploads mainly from smart-phone devices. Faces from the Adience are highly unconstrained, reflecting many of the real-world challenges, such as occlusions, extreme variations in head pose, lighting conditions quality. The database contains about 26K images of 2284 subjects.

CelebA. CelebA is a large-scale face attributes database [93] consisting in more than 200K images of more than 10K celebrities partitioned into training, validation and testing splits with approximatively 162K, 20K and 20K images in the respective splits. Each facial image is annotated with 40 binary attributes (see Table 5.7). Database images present high variations in pose, expression, race, background, imaging conditions.

LFWA. LFWA is an unconstrained database for face attributes estimation [93]. It has 13233 images of 5749 identities annotated with the same 40 attributes as in the CelebA database. The database is partitioned in 6263 training images and 6970 testing images.

5.4.2.2 Deep multi-task learning for attributes estimation

The aim of the proposed approach is to simultaneously estimate a large number of facial attributes using a single model.

Multi-task learning by CNN models demonstrates to be very effective for many face-related tasks [50, 89, 150]. Following this success, a multi-task learning approach based on convolutional neural network (MTL-CNN) to jointly estimate multiple facial attributes from a single face image is proposed. This model takes into account the attribute inter-correlations to obtain informative and robust feature representation.

Table 5.7 List of 40 face attributes provided with the CelebA database.

Attr. Idx.	Attr. Def.	Attr. Idx.	Attr. Def.
1	5oClockShadow	21	Male
2	ArchedEyebrows	22	MouthSlightlyOpen
3	BushyEyebrows	23	Mustache
4	Attractive	24	NarrowEyes
5	BagsUnderEyes	25	NoBeard
6	Bald	26	OvalFace
7	Bangs	27	PaleSkin
8	BlackHair	28	PointyNose
9	BlondHair	29	RecedingHairline
10	BrownHair	30	RosyCheeks
11	GrayHair	31	Sideburns
12	BigLips	32	Smiling
13	BigNose	33	StraightHair
14	Blurry	34	WavyHair
15	Chubby	35	WearingEarrings
16	DoubleChin	36	WearingHat
17	Eyeglasses	37	WearingLipstick
18	Goatee	38	WearingNecklace
19	HeavyMakeup	39	WearingNecktie
20	HighCheekbones	40	Young

Arched Eyebrows, Big Lips, Blond Hair, Heavy Makeup, No Beard, Rosy Cheeks, Wavy Hair, Wearing Lipstick, Wearing Necklace, Young.

Male, Eyeglasses, Wavy Hair, Sideburns, Bushy Eyebrows, Pointy Nose, Mouth Slightly Open, Bags Under Eyes, Wearing Necklace, Young.

Female, 4-6.

(a) (b) (c)

Fig. 5.15 Examples from evaluated databases. (a) Face image from the CelebA dataset and occurring attributes belonging to the set of 40 attributes. (b) Face image from the LFWA with corresponding face attributes coming from the same 40 attributes as in the CelebA. (c) Face image from the Adience benchmark labeled in terms of gender and age group.

This property is desirable for problems, like facial attributes estimation, where classes are correlated with each other. As shown in Figure 5.16, several attributes of the CelebA have strong pair-wise correlations (elements with red color). For example, "Male", "Attractive", and "NoBeard" are highly correlated, this means that gender and presence/absence of beard affect face's attractiveness.

For an input face image, the resulting model is able to jointly predict all the learned facial attributes. Given that many of the public-domain databases provide ground-truth only for a subset of desired facial attributes (e.g. only facial expressions, or soft biometrics), the training set is composed by aggregating data from multiple databases labeled with a single attribute. The MTL-CNN consists of shared parameters for all the attributes, followed by attribute-specific parameters. Shared parameters adapt to the complete set of domains, while attribute-specific parameters deal with the estimation of each attribute. The aforementioned CNN is trained by combining two different losses: the first is used for attributes that are defined as mutually exclusive (e.g. age group and gender), instead the second is used for attributes that are defined as co-occurent. Additionally, a gating mechanism is introduced in order to pass/suppress information. Finally, to better reflect the correlation between facial attributes a label post-processing layer is applied.

Mutually exclusive vs. co-occurent attributes

Many of the considered attributes are mutually exclusive, that is only one class is the correct one. For example, for age-group problem it is not possible that the same subject is simultaneously classified in ranges "15-20" and "25-32". For these attributes, softmax cross-entropy loss (see Section 2.2.5, eq. 2.5), the most popular loss function for single-label image classification in CNNs, is used. On the other hand, co-occurent attributes are present such as "Smiling", and "Mustache". These are attributes that can simultaneously occur and practically this means that their probabilities are independent. The binary cross entropy loss (see Section 2.2.5, eq. 2.6) is used beacuse, unlike the softmax that give a probability distribution around classes, it allows to deal with multi-label problems. Although CelebA and LFWA databases contain facial attributes mutually exclusive such as "BrownHair", "BlackHair" and "BlondeHair" and the use of softmax cross-entropy would enhance the learning algorithm in order to maximize only one attribute among them, given that ground-truth says only presence/absence of the attribute, estimation of these attributes is maintained as multi-label problem (instead of a single-label multi-class problem). The algorithm would adapt in order to learn such dependences.

Center loss

Center loss was first proposed for face recognition task [156] and used for other problems because of its effectiveness for making discriminative embedding features [151]. The purpose of center loss is to minimize intra-class variations while maximizing inter-class variations. The original center loss was designed for the single label classification problem and it is hard to be exploited for multi-label classification. Therefore, the criterion is modified in order to address the multi-label classification problem as follows [107]:

$$\mathcal{L}_i^{Center} = \|e - c_{y_i}\|_2^2, \tag{5.1}$$

where e is the embedding feature vector from penultimate network layer for the i-th sample, c_{y_i} is the class center feature vector for the corresponding ground-truth label.

Label-processing layer

Label-processing layer is designed to reflect relationships among facial attributes. For example, in the CelebA database [93] {"WearingLipstick", "RoseCheeks", "HeavyMakeup"} and {"Male", "Goatee"} are strongly correlated. In order to exploit this information, the co-occurrence matrix $\mathbf{M_c}$ (see Figure 5.16) is computed by counting the number of pair-wise co-occurrences for the 40 facial attributes. The co-occurrent value $\mathbf{M_c}[\mathbf{i}, \mathbf{j}]$ between the i-th and the j-th attribute is calculated higher as i-th and j-th appear together in more images. Label-processing (LP) layer is then formalized as follows:

$$\mathbf{o_{LP}} = ReLU(\mathbf{W}\mathbf{M_c}\mathbf{o_{pred}}) \cdot \mathbf{o_{pred}}. \tag{5.2}$$

Here, *ReLU* is the element-wise ReLU non-linearity (see Section 2.4), $\mathbf{M_c}$ is the co-occurrence matrix, $\mathbf{W}$ is a weights matrix, $\mathbf{o_{pred}}$ is the prediction, and $(\cdot)$ indicates the element-wise multiplication. Whether the predicted probability o_{Male} is around five, while probabilities $o_{HeavyMakeup}$ and $o_{WearingLipstick}$ are high, given that the pair-wise correlation between these attributes is high, the resulting probability for the attribute *Male* due to the Label-processing layer will be penalized.

Gating mechanism

Gate units are employed to select contextual attribute features. The idea is based by the mechanism of gate unit in *Long Short-Term Memory (LSTM)* [57], which is used to learn to remember or forget the history information from long sequence of input

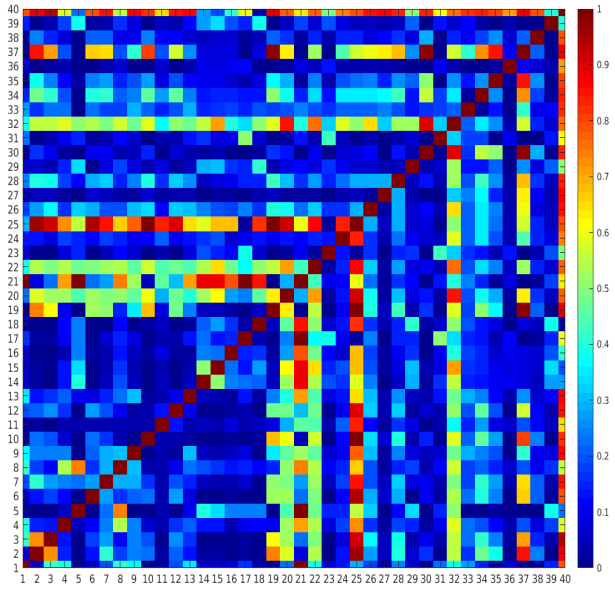


Fig. 5.16 Co-occurrence matrix of the 40 attributes provided with the CelebA database (only training set labels are considered).

data. Differently from LSTM, introduced gate units do not depend on temporal data, but are designed to "remember" or "forget" features across different attributes. The used gate equation is:

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \cdot \mathbf{x}, \tag{5.3}$$

where σ is the element-wise sigmoid non-linearity, $\mathbf{W}$ and $\mathbf{b}$ are learnable parameters, $\mathbf{x}$ is a features vector, and $(\cdot)$ indicates the element-wise multiplication. The gate function is used at feature-level and prediction-level. Feature-level gate is introduced immediately before the classification layer for disentangling the representation of the different attributes. Prediction-level gate controls the co-occurrence between predicted attributes. Differently from the previously described Label-processing layer, which exploits the co-occurrence matrix to find out attributes with strong correlation, prediction-level gate function learns these relationships in a self-supervised manner.

ResNet-50 [55] (see Section 2.8) is used as CNN architecture. It has 50 layers with parameters: the first is a 7×7 convolutional layer followed by four blocks each containing 3, 4, 6, 3 residual units, respectively. The network ends with a global average pooling layer and two fully-connected layers. The first fully connected layer maps the 2048 features into other 2048 features while the second is a 48-way (40 attributes + 8 age groups) fully-connected layer for classification.

The network is trained end-to-end by using random batches of images: half of the samples taken from the Adience benchmark and the remaining half taken from the CelebA database. The face images are pre-processed by firstly aligning them using the ground-truth landmark points to estimate an affine transformation in order to minimize difference between samples from the different databases. Then, contrast-normalization by subtracting the mean and by dividing by the standard deviation for each color channel is applied. Finally, images are sub-sampled in order to fit required network size to 224×224 pixels. MTL-CNN is trained using SGD with Nesterov momentum (see Section 2.4.1), the batch size of 32, the learning rate is kept fixed to a value of 0.0001. Momentum is set to 0.9, and the weight-decay parameter is 5e-4. The model is trained for 80 epochs and the best model is selected using the early stopping strategy (i.e. best performances achieved in all tasks on validation sets). The total loss used to train the model is given by $\mathcal{L}_{total} = \mathcal{L}_{CE} + \mathcal{L}_{BCE} + 0.95\mathcal{L}_{center}$, where $\mathcal{L}_{CE}$ is the softmax cross-entropy loss, $\mathcal{L}_{BCE}$ is the binary cross-entropy, and $\mathcal{L}_{center}$ indicates the center loss.

5.4.2.3 Related works

Eidinger *et al.* [38] conducted extensive tests on the collected Adience benchmark. Levi *et al.* [84] trained two different convolutional neural networks for addressing again the gender and age group on the Adience database. Van de Wolfshaar *et al.* [147] fine-tuned a pre-trained convolutional neural network and then used the deep features for gender classification using a support vector machine.

Other methods allow to predict multiple face attributes at the same time. Kumar *et al.* [77] proposed the first work on the automatic classification of facial attributes. The approach involves the use of independent classifiers for each attribute trained using features (image intensities in RGB and HSV color spaces, edge magnitudes, gradient directions) extracted from hand-picked facial regions. More recent approaches leverage deep learning techniques. Zhang *et al.* [172] presented PANDA, a part-based method involving the use of pose-normalized CNN to infer human attributes from images. More in detail, CNN features extracted from localized regions are used to train SVM classifiers for attribute prediction. Liu *et al.* [93] introduced two large-scale face attribute databases, namely CelebA and LFWA, and utilized a combination of two *localization networks* (LNETs) and an *attribute recognition network* (ANET). LNets are trained in a weakly supervised manner, instead ANet is pre-trained by classifying massive face identities and then fine-tuned by attributes to extract features that are fed into independent linear support vector machines (SVMs) for the final attribute

classification. Rozsa *et al.* [120] and Rudd *et al.* [121] trained CNNs directly over facial attribute data of CelebA obtaining competitive performance. Uricár *et al.* [145] used an ensemble of multi-class SO-SVM predictors on top of deep features learned using a convolutional neural network for apparent age, gender, and smile prediction. The VGG-16 architecture pre-trained on ImageNet has been fine-tuned on IMDB-WIKI and ChaLearn 2015 LAP datasets. Recently, Kalayeh *et al.* [67] showed that semantic segmentation further improves 40 attributes classification accuracy on both the CelebA and LFWA databases.

Deep learning approaches have been proved to be well suited for multi-task learning (MTL). Several MTL approaches have been proposed for attribute estimation. A multi-task restricted Boltzmann machine (MT-RBM) has been used by Ehrlich *et al.* [37] for facial attribute classification. They showed performance improvement over the state-of-the-art on three datasets. Han *et al.* [50] presented a multi-task convolutional neural network able to model both attribute correlation and attribute heterogeneity.

5.4.2.4 Evaluation procedure

Experiments are conducted by simultaneously training the model on Adience benchmark and CelebA database. To supplement the analysis on CelebA dataset and to demonstrate the generality of the proposed method experimental results are reported on LFWA too. Results are reported in terms of accuracy for age and gender classification on the Adience benchmark. For age classification, both the accuracy when the algorithm gives the exact age-group classification and when the algorithm is off by one adjacent age-group (i.e., the subject belongs to the group immediately older or immediately younger than the predicted group) is measured. While performances for CelebA and LFWA databases are expressed in terms of classification error.

Testing for both age and gender classification on Adience benchmark is performed using a standard five-fold, subject-exclusive cross-validation protocol defined in [84], while for both CelebA and LFWA databases results are computed on the testing set. Specifically, the average accuracy over the five-folds of cross-validation is reported for both age and gender classification; instead, average error computed over the classification errors on testing set produced by each one of the five models trained due to cross-validation is reported for facial attributes estimation on CelebA and LFWA databases.

5.4.2.5 Experimental results

In this section, experiments conducted for multiple facial attributes estimation are described. The performances obtained by the different configurations of the proposed method are shown in Table 5.8. In all the experiments, the Center loss is considered with both Softmax cross-entropy and BCE losses. The first experiment consists in the use of the architecture described in Section 5.4.2.2, that is the standard ResNet-50 with the addition of a fully-connected layer immediately before the classification layer that has 48 neurons. For this experiment the classification accuracy for gender is 87.52%, the obtained classification accuracy for exact estimation of age group corresponds to 60.19%. Finally, classification errors for the 40 binary attributes estimation problem are 10.75% and 13.16% for CelebA and LFWA respectively.

The second experiment includes the proposed gate functions. Specifically, both feature-level and prediction-level layers are introduced into the architecture immediately before and after the classification layer respectively. Experimental results show an improvement on all the considered tasks. More in detail, gender accuracy is 92.14%, age group accuracy increased of 3% for exact prediction and 5% respect to previous experiment. The average error on the 40 attributes decreased for both datasets.

The third experiment combines the feature-level gate layer and the Label-processing layer. Specifically, the prediction-layer used for the previous experiment is replaced by the Label-processing as post-processing approach of predicted scores. Gender accuracy is equal to 89.67%, age group classification accuracy is respectively 60.03% for exact prediction and 88.48% for 1-off.

As shown in Table 5.8, the introduction of Label-processing layer improves the gender classification accuracy, but the best result is achieved by using prediction-gate layer instead of Label-processing layer. For age group classification the best performance is still achieved by the solution with gating layers, while in this case the worst performance is achieved by the solution including the Label-processing layer. Finally for binary attributes classification, the lowest error for both the CelebA and the LFWA databases is achieved by the solution including gating functions, while the highest error is the one obtained by the solution without both gating and Label-processing layers.

Table 5.9 reports the comparison in terms of accuracy for the gender classification problem on the Adience benchmark. All the proposed solutions outperform the state-of-the-art methods. More in detail the proposed solution involving the use of gating functions obtained an accuracy 5% higher than the best method proposed in [147].

Table 5.10 reports the comparison in terms of classification accuracy for age-group estimation on the Adience benchmark. The accurcacy both when the algorithm gives

Table 5.8 Performance results for each experiment on all the task considered: Age-group and gender classification (accuracy in %) for the Adience benchmark, and 40 binary attributes estimation (classification error in %) on the CelebA and LFWA databases.

Method	Gender acc. (%)	Age group acc. (%)		Attributes err. (%)	
		Exact	1-off	CelebA	LFWA
ResNet-50	87.52	60.19	87.79	10.75	13.16
Gates	**92.14**	**63.81**	**92.24**	**10.11**	**12.97**
Gate+Labelproc	89.67	60.03	88.48	10.52	13.02

Table 5.9 Gender estimation results on the Adience benchmark in terms of mean accuracy.

Method	Accuracy
Eidinger et al. [38]	77.8
Hassner et al. [53]	79.3
Levi et al. [84]	86.8
van de Wolfshaar et al. [147]	87.2
Proposed (ResNet-50)	87.5
Proposed (Gates)	**92.1**
Proposed (Gate+Labelproc)	89.7

the exact age-group and when the algorithm is off by one adjacent age-group is reported. The best proposed method achieves a performance (63.8%) close to the best in the state-of-the-art (Rothe *et al.* [119]).

Table 5.11 reports the comparison in terms of classification error for the 40 binary attributes estimation on the CelebA and LFWA databases. Experimental results indicate that the proposed method obtains a classification error higher than the best in the state-of-the-art: for the CelebA the error is 3% higher than the best method proposed in [50]; instead, for the LFWA database the error is only 0.1% higher than the best method presented in [67].

In addition, we also evalute the generalization ability of the proposed approach in a cross-database testing scenario. In this testing, the attribute estimation method is trained on one face database, and tested on a different one. Specifically, experiments for each one of the proposed solutions are executed in this scenario. The results reported in Table 5.12. As it is possible to see, the proposed method involving the use of gating functions obtained the best performances on both the databases.

Table 5.10 Age-group estimation results on the Adience benchmark in terms of mean accuracy. The accuracy both when the algorithm gives the exact age-group and when the algorithm is off by one adjacent age-group is reported.

Method	Exact	1-off
Eidinger *et al.* [38]	45.1	79.5
Levi *et al.* [84]	50.7	84.7
Rothe *et al.* [119]	**64.0**	**96.6**
Proposed (ResNet-50)	60.2	87.8
Proposed (Gates)	63.8	92.2
Proposed (Gate+Labelproc)	60.0	88.5

Table 5.11 Attribute estimation performance evaluated by classification error and average precision on the LFWA and CelebA databases.

Method	CelebA	LFWA
FaceTracer [77]	18.9	26.0
PANDA [172]	15.0	19.0
LNets+ANet [93]	12.7	16.1
MCNN-AUX [51]	11.5	13.7
MOON [121]	9.1	-
SSP+SSG [67]	8.2	**12.9**
HDMTL [50]	**7.0**	14.0
Proposed (ResNet-50)	10.7	13.2
Proposed (Gates)	10.1	13.0
Proposed (Labelproc)	10.5	13.0

Table 5.12 Cross-database results for facial attributes estimation.

Method	Database		Avg. error (%)
	Training	Testing	
ResNet-50	CelebA	LFWA	27.3
ResNet-50	LFWA	CelebA	14.0
Gates	CelebA	LFWA	**26.2**
Gates	LFWA	CelebA	**13.7**
LabelProc	CelebA	LFWA	26.8
LabelProc	LFWA	CelebA	13.7

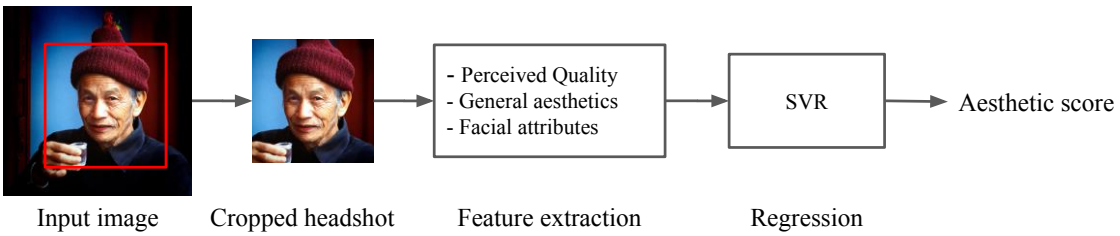


Fig. 5.17 Proposed pipeline for portrait images aesthetic assessment. The face is first detected and cropped, then features for perceived quality (see Section 3), general aesthetics (see Section 4.1), and facial attributes are extracted. Finally, a linear Support Vector Regressor (SVR) is used for aesthetic score prediction.

5.5 Portrait images aesthetics score estimation

Portraits or headshots are defined as frontal or near frontal face images cropped in order to contain the whole face and shoulders. In this section, a method for aesthetic quality estimation of portraits is described. As with the aesthetics assessment of images with generic content, the portrait aesthetics assessment is treated as a regression problem, thus, given a headshot the model estimates a score describing its aesthetic quality.

The proposed approach is depicted in Figure 5.17. Given an image, the faces are first detected using a multi-view face detector inspired by Farfade *et al.* [41]. The detected bounding boxes sizes are increased of 20% in order to include also a portion of the shoulders, then the contained region is cropped and used as the headshot. Multiple features are extracted in order to describe both the visual attributes (quality and aesthetics) and facial attributes of the whole face. CNN features off-the-shelf are used because of the limited amount of available data. Specifically, the DeepBIQ model described in Chapter 3, the DeepIA model presented in Chapter 4 and the FaceA model proposed in Section 5.4.2 are used for extracting features. More in detail, for each model the portrait image is first resized to fit the required size of the input of the considered model, then it is fed into the CNN model in order to extract features from the fully-connected layer just before the classification layer. Two 4,096-dim feature vectors, one describing image aesthetics, the other describing image quality are obtained. A 2,048-dim feature vector is obtained for facial attributes description. A linear Support Vector Regressor (SVR) is used to map the features into an aesthetic score.

Table 5.13 Correlation performances for the three considered databases obtained by using DeepIA method (see Section 4.1) for predicting image aesthetic scores.

Database	LCC	SROCC
HFS	0.27	0.27
FAVA*	0.52	0.51
Flickr	0.39	0.37

*Some images have been used for the DeepIA model training.

5.6 Performance evaluation

For the experiments on the portrait aesthetics estimation, the same evaluation procedure adopted in [90] is followed. More in detail, for each experiment 10-fold cross validation is performed by randomly selecting the training and testing images. This procedure is repeated 10 times to avoid sampling bias. For each repetition the Pearson's Linear Correlation Coefficient (LCC) (see Section 3.3, eq. 3.8) and the Spearman's Rank Ordered Correlation Coefficient (SROCC) (see Section 3.3, eq. 3.9) between the predicted and the ground-truth aesthetic scores are computed, reporting the mean of these correlation coefficients across the 10 rounds. In all the experiments the PyTorch[2] framework is used for feature extraction, and the LIBLINEAR library [40] is employed for SVR training.

5.7 Experimental results

In this section the experimental results are reported. First of all, in order to verify whether the general content images aesthetics can approximate portrait images aesthetics, the proposed DeepIA method (see Section 4.1) is used for predicting image aesthetic scores. Predicted scores are scaled from the original range [1, 10] to the target range for each dataset considered. The obtained results in Table 5.13 show that LCC and SROCC are very low for both HFS and Flickr databases, while correlation values are higher for the FAVA database. This is mainly motivated by the fact that some of the headshots of the FAVA database are cropped from images of the training set previously used for training the DeepAI model. Thus, results confirmed that general content aesthetics is not well suited to address the problem of portrait image aesthetics.

Table 5.14 reports results for all the experiments. The first set of experiments consists in the evaluation of a single feature at time to be fed into the SVR. Aesthetics

[2]www.pytotch.org

Table 5.14 Correlation performances of the proposed solutions for each dataset.

Methods	HFS		FAVA*		Flickr	
	LCC	SROCC	LCC	SROCC	LCC	SROCC
Lienhard *et al.* [90]	**0.73**	-	0.51	-	0.49	-
FaceA-feat	0.50	0.47	0.31	0.32	0.35	0.34
DeepIA-feat	0.66	0.65	0.52	0.49	0.49	0.47
DeepIQA-feat	0.57	0.61	0.37	0.36	0.35	0.34
FaceA-feat+DeepIA-feat	0.71	0.69	0.54	0.54	**0.50**	0.49
FaceA-feat+DeepBIQ-feat	0.67	0.66	0.45	0.47	0.43	0.43
DeepIA-feat+DeepBIQ-feat	0.70	0.70	0.52	0.49	0.47	0.45
DeepIA-feat+DeepBIQ-feat+FaceA-feat	**0.73**	0.71	**0.55**	0.55	**0.50**	0.49

*Some images have been used for the DeepIA model training.

features achieved the best performances on all the datasets respect to quality and facial attributes features, while facial attributes features obtained low correlation performances. Furthermore on FAVA and Flickr databases, this proposed setting obtained comparable results with the state-of-the-art. In the second set of experiments, features are combined by simply concatenating them. Results show that the combination of features improve correlation for all the datasets. The combinations of facial attributes and aesthetic features, and aesthetic and quality features obtained more or less the same results further improving correlation on all the considered datasets. Finally, the best results are obtained thanks to the combination of all the considered features. This last configuration achieves slightly better results than the state-of-the-art methods.

As expected, the general content images aesthetics model used is not effective for portrait images aesthetic estimation. Instead features learned for aesthetics characterization seem to generalize well and are effective for portrait image aesthetics assessment. Additionally, facial attributes features do not seem to consistently provide useful information.

Chapter 6

Conclusions

This thesis shows how deep learning and convolutional neural networks are well suited for describing visual attributes such as quality and aesthetics.

The problem of automatic perceived image quality assessment is investigated. The distortion-generic image quality assessment has been considered. The best proposal, named DeepBIQ, consists of a CNN, originally trained to discriminate 1,183 visual categories, that is fine-tuned for category-based image quality assessment by using multiple random crops from the original images. This CNN is then used to exctract features that are then fed to a SVR to predict the crop-level quality score. Finally, quality scores predicted for each crop are combined using the average pooling fusion scheme in order to obtain the quality score for the whole image. Experimental results both on four benchmark databases of synthetically distorted images and on a database containing images affected by authentic distortions have shown that DeepBIQ is able to outperform all the methods in the state-of-the-art also on all these datasets. Furthermore, in many cases, the quality score predictions of DeepBIQ are closer to the average observer than those of a generic human observer.

The image aesthetic assessment has been addressed at first on general content images and then on the specific case of portrait images. For the general content image aesthetic assessment, the proposed approach consists in fine-tuning a canonical CNN architecture, originally trained to classify both objects and scenes, by casting the image aesthetic prediction as a regression problem. Experimental results have shown the robustness of the solution proposed, which outperforms the best methods in the state-of-the-art. Furthermore, results indicate that image aesthetics is a global attribute, and that the use of a saliency map to filter out not salient regions in the prediction stage does not help to achieve more accurate aesthetic score predictions.

Conclusions

Portrait image aesthetic assessment is investigated to deal with images containing faces. The proposed algorithm involves the use of middle-level features obtained by combining previous visual attributes (i.e. quality and aesthetics) and facial attributes. These features are then fed to a SVR to predict the aesthetic score. Facial attributes estimation algorithms used for feature extraction have been proposed in this thesis. A first proposal consists in a robust smile detector algorithm able to outperform the state-of-the-art methods also for distorted images. Additionally, for dealing with a huge number of attributes a multiple-task model is designed in order to simultaneously estimate soft biometrics and attributes such as hair colors and styles, types of beards. Results collected for the first algorithm have shown better performances respect to the state-of-the-art methods (also respect to highly distorted images). Experimental results obtained by the proposed multi-task model demonstrated comparable performances with state-of-the-art approaches. The proposed method for portrait image aesthetic assessment has achieved comparable results respect to state-of-the-art methods on three databases.

Most of the approaches proposed in this thesis have focused on a single problem at time. The future works are mostly related to consider different visual attributes in a unique framework. In the case of an integrated approach for both quality and aesthetics estimation, it could be possible to simultaneously predict metrics for both the visual attributes and to evaluate the tradeoff between the two aspects. Given the close connection between image aesthetics assessment and image sentiment analysis, the future works relies on the development of algorithms to address this last problem.

References

[1] Ahonen, T., Hadid, A., and Pietikäinen, M. (2004). Face recognition with local binary patterns. In *Computer vision-eccv 2004*, pages 469–481. Springer.

[2] Alaei, A., Raveaux, R., and Conte, D. (2016). Image quality assessment based on regions of interest. *Signal, Image and Video Processing*, pages 1–8.

[3] Algom, D. (1992). *Psychophysical approaches to cognition*, volume 92. Elsevier.

[4] Allen, E., Triantaphillidou, S., and Jacobson, R. (2007). Image quality comparison between jpeg and jpeg2000. i. psychophysical investigation. *Journal of Imaging Science and Technology*, 51(3):248–258.

[5] An, L., Yang, S., and Bhanu, B. (2015). Efficient smile detection by Extreme Learning Machine. *Neurocomputing*, 149:354–363.

[6] Asthana, A., Zafeiriou, S., Cheng, S., and Pantic, M. (2014). Incremental Face Alignment in the Wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1859–1866, Columbus, Ohio, USA. IEEE.

[7] Bai, Y., Guo, L., Jin, L., and Huang, Q. (2009). A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. *International Conference on Image Processing (ICIP)*, (07118074):3305–3308.

[8] Bakhshi, S., Shamma, D. A., and Gilbert, E. (2014). Faces engage us: Photos with faces attract more likes and comments on instagram. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 965–974, New York, NY, USA. ACM.

[9] Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. *Unsupervised and Transfer Learning Challenges in Mach. Learn.*, 7:19.

[10] Bhattacharya, S., Sukthankar, R., and Shah, M. (2010). A framework for photo-quality assessment and enhancement based on visual aesthetics. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 271–280. ACM.

[11] Bianco, S. (2017). Large age-gap face verification by feature injection in deep networks. *Pattern Recognition Letters*, 90:36–42.

[12] Bianco, S., Bruna, A. R., Naccari, F., and Schettini, R. (2013). Color correction pipeline optimization for digital cameras. *Journal of Electronic Imaging*, 22(2):023014–023014.

[13] Bianco, S., Celona, L., Napoletano, P., and Schettini, R. (2016a). Predicting image aesthetics with deep learning. In *Advanced Concepts for Intelligent Vision Systems: 17th International Conference (ACIVS 2016)*, volume 10016, pages 117–125. Springer International Publishing.

[14] Bianco, S., Celona, L., Napoletano, P., and Schettini, R. (2018). On the use of deep learning for blind image quality assessment. *Signal, Image and Video Processing*, 12(2):355–362.

[15] Bianco, S., Celona, L., and Schettini, R. (2016b). Robust smile detection using convolutional neural networks. *Journal of Electronic Imaging*, 25(6):063002–063002.

[16] Bianco, S., Ciocca, G., Marini, F., and Schettini, R. (2009). Image quality assessment by preprocessing and full reference model combination. In *IS&T/SPIE Electronic Imaging*, pages 72420O–72420O.

[17] Borth, D., Chen, T., Ji, R., and Chang, S.-F. (2013). Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 459–460. ACM.

[18] Boureau, Y.-L., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2559–2566. IEEE.

[19] Bovik, A. C. (2013). Automatic prediction of perceptual image and video quality. *Proceedings of the IEEE*, 101(9):2008–2024.

[20] Campos, V., Jou, B., and Giro-i Nieto, X. (2017). From pixels to sentiment: Fine-tuning cnns for visual sentiment prediction. *Image and Vision Computing*.

[21] Caruana, R. (1998). Multitask learning. In *Learning to learn*, pages 95–133. Springer.

[22] Celona, L. and Manoni, L. (2017). Neonatal facial pain assessment combining hand-crafted and deep features. In *New Trends in Image Analysis and Processing – ICIAP 2017 Workshops*. Springer.

[23] Chen, Q., Yang, L., Zhang, D., Shen, Y., and Huang, S. (2017). Face deduplication in video surveillance. *International Journal of Pattern Recognition and Artificial Intelligence*, page 1856001.

[24] Ciancio, A., Da Costa, A. L. N. T., da Silva, E. A., Said, A., Samadani, R., and Obrador, P. (2011). No-reference blur assessment of digital pictures based on multifeature classifiers. *IEEE Trans. on Image Proc.*, 20(1):64–75.

[25] Ciocca, G., Corchs, S., Gasparini, F., and Schettini, R. (2014). How to assess image quality within a workflow chain: an overview. *Int. J. on Dig. Lib.*, 15(1):1–25.

[26] Corchs, S., Gasparini, F., and Schettini, R. (2014). No reference image quality classification for jpeg-distorted images. *Digital Signal Processing*, 30:86–100.

[27] Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

[28] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 1, pages 886–893. IEEE.

[29] Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2006). Studying aesthetics in photographic images using a computational approach. In *Computer Vision–ECCV 2006*, pages 288–301. Springer.

[30] Datta, R., Li, J., and Wang, J. Z. (2007). Learning the consensus on visual quality for next-generation image management. In *Proceedings of the 15th international conference on Multimedia*, pages 533–536. ACM.

[31] Datta, R., Li, J., and Wang, J. Z. (2008). Algorithmic inferencing of aesthetics and emotion in natural images: An exposition. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 105–108. IEEE.

[32] Deng, J., Berg, A., Satheesh, S., Su, H., Khosla, A., and Fei-Fei, L. (2012). Imagenet large scale visual recognition competition 2012 (ilsvrc2012).

[33] Déniz, O., Bueno, G., Salido, J., and De la Torre, F. (2011). Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603.

[34] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655.

[35] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

[36] Eckert, M. P. and Bradley, A. P. (1998). Perceptual quality metrics applied to still image compression. *Signal processing*, 70(3):177–200.

[37] Ehrlich, M., Shields, T. J., Almaev, T., and Amer, M. R. (2016). Facial attributes classification using multi-task representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 47–55.

[38] Eidinger, E., Enbar, R., and Hassner, T. (2014). Age and gender estimation of unfiltered faces. *Information Forensics and Security, IEEE Transactions on*, 9(12):2170–2179.

[39] Ekman, P., Friesen, W. V., and O'Sullivan, M. (1988). Smiles when lying. *Journal of personality and social psychology*, 54(3):414–420.

[40] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *The J. of Mach. Learn. Res.*, 9:1871–1874.

[41] Farfade, S. S., Saberian, M., and Li, L.-J. (2015). Multi-view face detection using deep convolutional neural networks. *Cornell University Library*.

[42] Gao, Y., Liu, H., Wu, P., and Wang, C. (2015). A new descriptor of gradients self-similarity for smile detection in unconstrained scenarios. *Neurocomputing*.

[43] Ghadiyaram, D. and Bovik, A. C. (2014a). Blind image quality assessment on real distorted images using deep belief nets. In *Global Conf. on Signal and Information Processing (GlobalSIP)*, pages 946–950. IEEE.

[44] Ghadiyaram, D. and Bovik, A. C. (2014b). Crowdsourced study of subjective image quality. In *Asilomar Conf. Signals, Syst. Comput.*

[45] Ghadiyaram, D. and Bovik, A. C. (2016). Massive Online Crowdsourced Study of Subjective and Objective Picture Quality. *IEEE Trans. Image Process.*, 25(1):372–387.

[46] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

[47] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.

[48] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323.

[49] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[50] Han, H., K. Jain, A., Shan, S., and Chen, X. (2017). Heterogeneous Face Attribute Estimation: A Deep Multi-Task Learning Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

[51] Hand, E. M. and Chellappa, R. (2017). Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *AAAI*, pages 4068–4074.

[52] Hanjalic, A. (2006). Extracting moods from pictures and sounds: Towards truly personalized tv. *IEEE Signal Processing Magazine*, 23(2):90–100.

[53] Hassner, T., Harel, S., Paz, E., and Enbar, R. (2015). Effective face frontalization in unconstrained images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304.

[54] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[55] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[56] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[57] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[58] Hou, W., Gao, X., Tao, D., and Li, X. (2015). Blind image quality assessment via deep learning. *Neural Networks and Learning Systems, IEEE Trans. on*, 26(6):1275–1286.

[59] Http://mplab.ucsd.edu (2009). The MPLab GENKI Database, GENKI-4K Subset.

[60] Huang, Y.-M. and Du, S.-X. (2005). Weighted support vector machine for classification with uneven training class sizes. In *2005 Int. Conf. on Mach. Learn. and Cybernetics*, volume 7, pages 4365–4369. IEEE.

[61] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.

[62] Itti, L. and Koch, C. (2001). Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203.

[63] Jayaraman, D., Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012). Objective quality assessment of multiply distorted images. In *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, pages 1693–1697. IEEE.

[64] Jia, Y. e. a. (2014). Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, pages 675–678. ACM.

[65] Joshi, D., Datta, R., Fedorovskaya, E., Luong, Q.-T., Wang, J. Z., Li, J., and Luo, J. (2011). Aesthetics and emotions in images. *IEEE Signal Processing Magazine*, 28(5):94–115.

[66] Judd, T., Ehinger, K., Durand, F., and Torralba, A. (2009). Learning to predict where humans look. In *IEEE International Conference on Computer Vision (ICCV)*.

[67] Kalayeh, M. M., Gong, B., and Shah, M. (2017). Improving facial attribute prediction using semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[68] Kang, L., Ye, P., Li, Y., and Doermann, D. (2014). Convolutional neural networks for no-reference image quality assessment. In *CVPR*, pages 1733–1740.

[69] Kang, L., Ye, P., Li, Y., and Doermann, D. (2015). Simultaneous estimation of image quality and distortion via multi-task convolutional neural networks. In *ICIP*, pages 2791–2795. IEEE.

[70] Kao, Y., Wang, C., and Huang, K. (2015a). Visual aesthetic quality assessment with a regression model. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 1583–1587. IEEE.

[71] Kao, Y., Wang, C., and Kaiqi, H. (2015b). Visual aesthetic quality assessment with a regression model. pages 1583–1587.

[72] Ke, Y., Tang, X., and Jing, F. (2006). The design of high-level features for photo quality assessment. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 419–426. IEEE.

[73] Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., and Brossard, E. (2016). The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882.

[74] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[75] Kottayil, N. K., Cheng, I., Dufaux, F., and Basu, A. (2016). A color intensity invariant low-level feature optimization framework for image quality assessment. *Signal, Image and Video Processing*, pages 1–8.

[76] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105.

[77] Kumar, N., Belhumeur, P., and Nayar, S. (2008). Facetracer: A search engine for large collections of images with faces. In *European conference on computer vision*, pages 340–353. Springer.

[78] Kumar, N., Berg, A. C., Belhumeur, P. N., and Nayar, S. K. (2009). Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE.

[79] Lapointe-Garant, M.-P., Huang, J.-G., Gea-Izquierdo, G., Raulier, F., Bernier, P., and Berninger, F. (2010). Use of tree rings to study the effect of climate change on trembling aspen in Québec.

[80] Larson, E. C. and Chandler, D. M. (2010). Most apparent distortion: full-reference image quality assessment and the role of strategy. *JEI*, 19(1):011006–011006.

[81] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[82] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[83] LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. (2012). Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer.

[84] Levi, G. and Hassner, T. (2015). Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In *Proc. ACM International Conference on Multimodal Interaction (ICMI)*.

[85] Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40.

[86] Li, C., Loui, A. C., and Chen, T. (2010). Towards aesthetics: A photo quality assessment and photo selection system. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 827–830. ACM.

[87] Li, J., Yan, J., Deng, D., Shi, W., and Deng, S. (2016a). No-reference image quality assessment based on hybrid model. *Signal, Image and Video Processing*, pages 1–8.

[88] Li, J., Zou, L., Yan, J., Deng, D., Qu, T., and Xie, G. (2016b). No-reference image quality assessment using prewitt magnitude based on convolutional neural networks. *Signal, Image and Video Processing*, 10(4):609–616.

[89] Li, Y., Wang, Q., Nie, L., and Cheng, H. (2017). Face attributes recognition via deep multi-task cascade. In *Proceedings of the 2017 International Conference on Data Mining, Communications and Information Technology*, page 29. ACM.

[90] Lienhard, A., Ladret, P., and Caplier, A. (2015a). How to predict the global instantaneous feeling induced by a facial picture? *Signal Processing: Image Communication*, 39:473–486.

[91] Lienhard, A., Ladret, P., and Caplier, A. (2015b). Low level features for quality assessment of facial images. In *VISAPP 2015-10th International Conference on Computer Vision Theory and Applications*, pages 545–552.

[92] Lienhard, A., Reinhard, M., Caplier, A., and Ladret, P. (2014). Photo rating of facial pictures based on image segmentation. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 2, pages 329–336. IEEE.

[93] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738.

[94] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[95] Lu, X., Lin, Z., Jin, H., Yang, J., and Wang, J. Z. (2014). Rapid: Rating pictorial aesthetics using deep learning. In *Proceedings of the ACM International Conference on Multimedia*, pages 457–466. ACM.

[96] Lv, Y., Jiang, G., Yu, M., Xu, H., Shao, F., and Liu, S. (2015). Difference of gaussian statistical features based blind image quality assessment: A deep learning approach. In *ICIP*, pages 2344–2348. IEEE.

[97] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30.

[98] Mahmoudpour, S. and Kim, M. (2016). No-reference image quality assessment in complex-shearlet domain. *Signal, Image and Video Processing*, 10(8):1465–1472.

[99] Males, M., Hedi, A., and Grgic, M. (2013). Aesthetic quality assessment of headshots. In *ELMAR, 2013 55th International Symposium*, pages 89–92. IEEE.

[100] Manap, R. A. and Shao, L. (2015). Non-distortion-specific no-reference image quality assessment: A survey. *Information Sciences*, 301:141–160.

[101] Marchesotti, L., Perronnin, F., Larlus, D., and Csurka, G. (2011). Assessing the aesthetic quality of photographs using generic image descriptors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1784–1791. IEEE.

[102] Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Trans. on Image Proc.*, 21(12):4695–4708.

[103] Mittal, A., Soundararajan, R., and Bovik, A. C. (2013). Making a "completely blind" image quality analyzer. *SPL*, 20(3):209–212.

[104] Mittal, A. e. a. (2015). No-reference approaches to image and video quality assessment. *Multimedia Quality of Experience (QoE): Current Status and Future Requirements*, page 99.

[105] Moorthy, A. K. and Bovik, A. C. (2011). Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Trans. on Image Proc.*, 20(12):3350–3364.

[106] Murray, N., Marchesotti, L., and Perronnin, F. (2012). Ava: A large-scale database for aesthetic visual analysis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2408–2415. IEEE.

[107] Na, S., Yu, Y., Lee, S., Kim, J., and Kim, G. (2017). Encoding video and label priors for multi-label video classification on youtube-8m dataset. *arXiv preprint arXiv:1706.07960*.

[108] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, number 3, pages 807–814.

[109] Nishiyama, M., Okabe, T., Sato, I., and Sato, Y. (2011). Aesthetic quality classification of photographs based on color harmony. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 33–40. IEEE.

[110] Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987.

[111] Ojansivu, V. and Heikkilä, J. (2008). Blur insensitive texture classification using local phase quantization. In *Image and signal processing*, pages 236–243. Springer.

[112] Pappas, T. N., Safranek, R. J., and Chen, J. (2000). Perceptual criteria for image quality evaluation. *Handbook of image and video processing*, pages 669–684.

[113] Ponomarenko, N., Ieremeiev, O., Lukin, V., Egiazarian, K., Jin, L., Astola, J., Vozel, B., Chehdi, K., Carli, M., Battisti, F., et al. (2013). Color image database tid2013: Peculiarities and preliminary results. In *Visual Information Processing (EUVIP), 2013 4th European Workshop on*, pages 106–111. IEEE.

[114] Ponomarenko, N. e. a. (2009). Tid2008-a database for evaluation of full-reference visual quality assessment metrics. *Adv. of Mod. Rad.*, 10(4):30–45.

[115] Qi, G.-J., Aggarwal, C., Tian, Q., Ji, H., and Huang, T. (2012). Exploring context and content links in social media: A latent space method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):850–862.

[116] Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, pages 806–813.

[117] Redi, M., Rasiwasia, N., Aggarwal, G., and Jaimes, A. (2015). The beauty of capturing faces: Rating the quality of digital portraits. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–8. IEEE.

[118] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

[119] Rothe, R., Timofte, R., and Van Gool, L. (2016). Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, pages 1–14.

[120] Rozsa, A., Günther, M., Rudd, E. M., and Boult, T. E. (2016). Are facial attributes adversarially robust? *arXiv preprint arXiv:1605.05411*.

[121] Rudd, E. M., Günther, M., and Boult, T. E. (2016). Moon: A mixed objective optimization network for the recognition of facial attributes. In *European Conference on Computer Vision*, pages 19–35. Springer.

[122] Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

[123] Saad, M. A., Bovik, A. C., and Charrier, C. (2012). Blind image quality assessment: A natural scene statistics approach in the dct domain. *Trans. Image Process.*, 21(8):3339–3352.

[124] Sainath, T. N., Mohamed, A.-r., Kingsbury, B., and Ramabhadran, B. (2013). Deep convolutional neural networks for lvcsr. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8614–8618. IEEE.

[125] Seshadrinathan, K. and Bovik, A. C. (2011). Automatic prediction of perceptual quality of multimedia signals—a survey. *Multimedia Tools and Applications*, 51(1):163–186.

[126] Shan, C. (2011). An efficient approach to smile detection. In *Face and Gesture 2011*, pages 759–764. IEEE.

[127] Shan, C. (2012). Smile detection by boosting pixel differences. *IEEE Transactions on Image Processing*, 21(1):431–436.

[128] Sheikh, H. R., Wang, Z., Cormack, L., and Bovik, A. C. (2005). Live image quality assessment database release 2.

[129] Shinohara, Y. and Otsu, N. (2004). Facial expression recognition using fisher weight maps. In *International Conference on Automatic Face and Gesture Recognition*, pages 499–504. IEEE.

[130] Siersdorfer, S., Minack, E., Deng, F., and Hare, J. (2010). Analyzing and predicting sentiment of images on the social web. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 715–718. ACM.

[131] Simond, F., Arvanitopoulos Darginis, N., and Süsstrunk, S. (2015). Image Aesthetics Depends on Context. *International Conference on Image Processing*, (1).

[132] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

[133] Soundararajan, R. and Bovik, A. C. (2013). Survey of information theory in visual quality assessment. *Signal, Image and Video Processing*, 7(3):391–401.

[134] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

[135] Streijl, R. C., Winkler, S., and Hands, D. S. (2016). Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227.

[136] Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147.

[137] Sze, V., Chen, Y.-H., Yang, T.-J., and Emer, J. (2017). Efficient processing of deep neural networks: A tutorial and survey. *arXiv preprint arXiv:1703.09039*.

[138] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning.

[139] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[140] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[141] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708.

[142] Tang, H., Joshi, N., and Kapoor, A. (2014). Blind image quality assessment using semi-supervised rectifier networks. In *CVPR*, pages 2877–2884.

[143] Ting, K. M. (2002). An instance-weighting method to induce cost-sensitive trees. *Knowl. and Data Eng., IEEE Trans. on*, 14(3):659–665.

[144] Triantaphillidou, S., Allen, E., and Jacobson, R. (2007). Image quality comparison between jpeg and jpeg2000. ii. scene dependency, scene analysis, and classification. *Journal of Imaging Science and Technology*, 51(3):259–270.

[145] Uricár, M., Timofte, R., Rothe, R., Matas, J., and Van Gool, L. (2016). Structured output svm prediction of apparent age, gender and smile from deep features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–33.

[146] Vadicamo, L., Carrara, F., Cimino, A., Cresci, S., Dell'Orletta, F., Falchi, F., and Tesconi, M. (2017). Cross-media learning for image sentiment analysis in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 308–317.

[147] van de Wolfshaar, J., Karaaba, M. F., and Wiering, M. A. (2015). Deep convolutional neural networks and support vector machines for gender recognition. In *Computational Intelligence, 2015 IEEE Symposium Series on*, pages 188–195. IEEE.

[148] Vapnik, V. N. and Vapnik, V. (1998). *Statistical learning theory*, volume 1. Wiley New York.

[149] Vu, C. T., Phan, T. D., and Chandler, D. M. (2012). S3: A spectral and spatial measure of local perceived sharpness in natural images. *Trans. Image Process.*, 21(3):934–945.

[150] Wang, F., Han, H., Shan, S., and Chen, X. (2017a). Deep multi-task learning for joint prediction of heterogeneous face attributes. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 173–179. IEEE.

[151] Wang, H., Li, Z., Ji, X., and Wang, Y. (2017b). Face r-cnn. *arXiv preprint arXiv:1706.01061*.

[152] Wang, J., Fu, J., Xu, Y., and Mei, T. (2016). Beyond object recognition: visual sentiment analysis with deep coupled adjective and noun neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 3484–3490. AAAI Press.

[153] Wang, Y., Wang, S., Tang, J., Liu, H., and Li, B. (2015). Unsupervised sentiment analysis for social media images. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[154] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Proc.*, 13(4):600–612.

[155] Wang, Z., Sheikh, H. R., and Bovik, A. C. (2003). Objective video quality assessment. *The handbook of video databases: design and applications*, pages 1041–1078.

[156] Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer.

[157] Whitehill, J., Littlewort, G., Fasel, I., Bartlett, M., and Movellan, J. (2009). Toward practical smile detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2106–2111.

[158] Williams, D. and Hinton, G. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–538.

[159] Willis, J. and Todorov, A. (2006). First impressions: Making up your mind after a 100-ms exposure to a face. *Psychological science*, 17(7):592–598.

[160] Winkler, S. (1999). Issues in vision modeling for perceptual video quality assessment. *Signal Processing*, 78(2):231–252.

[161] Wu, O., Hu, W., and Gao, J. (2011). Learning to predict the perceived visual quality of photos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 225–232. IEEE.

[162] Xu, J., Ye, P., Li, Q., Du, H., Liu, Y., and Doermann, D. (2016). Blind image quality assessment based on high order statistics aggregation. *IEEE Trans. Image Process.*, 25(9):4444–4457.

[163] Ye, P., Kumar, J., Kang, L., and Doermann, D. (2013). Real-time no-reference image quality assessment based on filter learning. In *CVPR*, pages 987–994.

[164] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *NIPS*, pages 3320–3328.

[165] You, Q., Jin, H., and Luo, J. (2017). Visual sentiment analysis by attending on local image regions. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[166] You, Q., Luo, J., Jin, H., and Yang, J. (2015). Robust image sentiment analysis using progressively trained and domain transferred deep networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 381–388. AAAI Press.

[167] You, Q., Luo, J., Jin, H., and Yang, J. (2016a). Building a large scale dataset for image emotion recognition: the fine print and the benchmark. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 308–314. AAAI Press.

[168] You, Q., Luo, J., Jin, H., and Yang, J. (2016b). Cross-modality consistent regression for joint visual-textual sentiment analysis of social multimedia. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 13–22. ACM.

[169] Yuan, J., Mcdonough, S., You, Q., and Luo, J. (2013). Sentribute: image sentiment analysis from a mid-level perspective. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, page 10. ACM.

[170] Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

[171] Zhang, L., Wang, X., Kalashnikov, D. V., Mehrotra, S., and Ramanan, D. (2016). Query-driven approach to face clustering and tagging. *IEEE Transactions on Image Processing*, 25(10):4504–4513.

[172] Zhang, N., Paluri, M., Ranzato, M., Darrell, T., and Bourdev, L. (2014a). Panda: Pose aligned networks for deep attribute modeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1644.

[173] Zhang, Y. (2012). A novel approach to detect smile expression. In *International Conference on Machine Learning and Applications*, pages 482–487. IEEE.

[174] Zhang, Y., Moorthy, A. K., Chandler, D. M., and Bovik, A. C. (2014b). C-diivine: No-reference image quality assessment based on local magnitude and phase statistics of natural scenes. *Sign. Proc.: Image Comm.*, 29(7):725–747.

[175] Zhang, Z., Luo, P., Loy, C. C., and Tang, X. (2014c). Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108. Springer.

[176] Zheng, Z., Zheng, L., and Yang, Y. (2017). Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *The IEEE International Conference on Computer Vision (ICCV)*.

[177] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495.

[178] Zhou, Z.-H. and Liu, X.-Y. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *Trans. on Knowl. and Data Eng.*, 18(1):63–77.