Local No-Reference Image Quality Assessment Using Convolutional Neural Networks

Bachelorarbeit

vorgelegt von

Oliver Marcel Wiedemann

an der

Universität Konstanz

Mathematisch-Naturwissenschaftliche Sektion

Fachbereich Informatik und Informationswissenschaft

1. Gutachter: Professor Dr. D. Saupe

2. Gutachter: PD Dr.-Ing. C. Borgelt

Konstanz, 2018

Abstract

The abstract concept of perceptual image quality has been studied almost exclusively as a global image property. In this thesis, the notion of subjective opinion scores is extended to spatially small regions of 64×64 pixels. A novel dataset of 32.000 individually annotated patches is created and used to train a convolutional neural network for local image quality assessment. We evaluate its predictiveness on an auxiliary dataset consisting of locally quality-annotated images. Additional experiments show that averaging quality maps created by our model already outperforms many traditional global image quality assessment methods. A meta aggregation model on top of a combination of spatially small feature maps, one of them being a quality map created by our model, performs close to the state of the art. As an application, we use our model to guide the bit allocation scheme in an extended JPEG image compression algorithm.

Zusammenfassung

Perzeptuelle Bildqualität wurde im Kontext menschlicher Wahrnehmung bisher fast ausschließlich als globale Eigenschaft von Bildern betrachtet. Diese Thesis erweitert den Begriff subjektiver Qualitätsbewertungen auf kleine Bildbereiche von 64×64 Pixeln. Ein neuartiger Datensatz von 32.000 individuell bewerteten Bildausschnitten wird erstellt, auf welchem ein Convolutional Neural Network für lokale Qualitätsvorhersagen trainiert wird. Die Aussagekraft dieses Modells wird auf einem zweiten Datensatz getestet, welcher aus Bildern mit zusätzlicher lokaler Qualitätsinformation besteht. Weitere Experimente zeigen, dass bereits der Mittelwert aller lokalen Qualitätsvorhersagen unseres Modells auf einem gegebenen Bild die globale menschliche Wahrnehmung besser approximiert als viele klassische Methoden. Ein Metamodell, welches auf kombinierten, räumlich kleinen Feature Maps arbeitet, erzielt eine Leistung nahe des Standes der aktuellen Forschung. Abschließend werden Qualitätsvorhersagen des lokalen Modells als Gewichtungsschema für die Kompressionsstärke in einem erweiterten JPEG Algorithmus evaluiert.

Contents

1 Introduction

1.1 Motivation

Digital images are ubiquitous in our modern world as they enable capturing, storage, transmission and manipulation of visual information. Passing through an intricate processing pipeline until being presented to an observer, images are prone to a variety of impairments and distortions. Physical limitations of camera equipment, printers and screens as well as more deliberate tradeoffs during encoding and compression influence the final result.

The discipline of Image Quality Assessment (IQA) is concerned with quantifying visual quality as perceived by humans. Although this concept lacks a generally accepted definition, studies have shown to yield reproducible mean opinion scores when querying a sufficiently large group of people on their opinions on a set of images. With millions of uploads to social media platforms per day, the amount of image data has reached an overwhelming volume. Automated prediction of image quality has numerous applications. From news outlets seeking suitable material for publications through media providers measuring the performance of their streaming services to researchers developing image compression methods: subjective studies are often too time- and cost-intensive to be viable. In this thesis, a machine-learning based approach to *local* image quality assessment is presented and evaluated.

1.2 Structure of this Thesis

Following this introduction, Section 2 establishes the state of the art and identifies room for possible improvements. In Section 3, notions of and assumptions on the concpet of subjective image quality are discussed and the KonPatch database is introduced. Section 4 provides technical background on the machine learning models that are trained and tested in Section 5. A practical application of one of our models in the domain of image compression is presented in Section 6. The results of the experiments are conclusively discussed in Section 7.

Parts of the work presented in this thesis have been published in [1] and [2].

2 Related Work

Image quality assessment (IQA) is a comparatively young field that emerged with the prevalence of digital cameras. The relevant literature is nevertheless extensive: IQA methods exist on a range from traditional signal processing and statistics to machine learning, there are papers on the human visual system on a neuropsychologic level and reports on subjective studies and the design of image quality databases. This section is compartmentalized to achieve a comprehensive depiction of the state of the art.

2.1 Databases

Researchers have published various databases to study perceptual image quality as a consensus-based property of images. From a coarse perspective, one usually distinguishes three classes of IQA related databases:

2.1.1 Artificially Distorted Databases

Artificially distorted databases are created by applying distortions, e.g. Gaussian blur, to a set of source images. The main advantages are that such databases contain multiple versions of the same source image and that apriori information about the severity of the distortion is available.

The baseline for modern IQA was set with the "LIVE Public-Domain Subjective Image Quality Database" [3, 4]. It consists of 29 pristine images which were distorted using JPEG and JPEG2000 compression, white noise, Gaussian blur and fast Rayleigh fading. The latter is statistically modeling the error introduced by physical signal propagation, for example along a wire or through a radio transmission. The LIVE database was subjectively rated in a controlled lab environment with more than 20 ratings per image. A related study conducted by Sheikh *et al.* [5] on this dataset compares full-reference IQA methods, of which some are introduced in Section 2.2.2.

A less noticed example published by Le Callet and Florent is the "IRCCyN/IVC Subjective Quality Assessment Database" [6], an early release containing 10 original images impaired with 4 types of distortions. Subjective scores were also collected in a lab study with 15 observers per image.

Ponomarenko *et al.* released the "Tampere Image Database (TID) 2008" [7]. Compared to LIVE it lacks in terms of content diversity with only 25 source images. By applying 17 types of distortions with 4 levels each, the set contains a total of 1700 distorted images. The authors also published a reference comparison of algorithms on their database [8]. Three years later, an extended version with both increased content diversity and more artificial distortions was released under the name TID2013 [9].

The "CSIQ Image Quality Database" by Larson and Chandler [10, 11] is notable for grouping source images into five content categories: animals, landscape, people, plants and urban. In a wider sense, this approach of relating content to visual quality is taken up again in publications where transfer-learning is used to gear models originally intended for object recognition towards quality assessment, for example by Varga *et al.* [12].

The previously discussed databases make the assumption that image quality can be sufficiently modeled by applying a single distortion to a pristine image. The "LIVE Multiply Distorted Image Quality Database" [13] was the first to addressed this issue by artificially corrupting source images with multiple distortions at the same time. However, this database was by far not as widely adopted as its single-distortion predecessor.

Artificially distorted databases have shortcomings: images in-the-wild are seldom impaired by a single distortion, but rather by an unknown mixture of distortions. Authentic mimicking may be non-trivial, e.g. for a motion blurred subject in front of an unimpaired background. Such distortions vary spatially and are content and context-dependent. Another issue is content diversity. Since artificially distorted databases aim to include many types of distortions at multiple levels, they quickly grow in size relative to the number of source images. This limitation raises the question whether they are suitable test sets for IQA algorithms at all.

2.1.2 Authentically Distorted Databases

The recently favored approach to refrain from artificially distorting a small set of pristine source images and rather collect large sets of images of diverse content and quality. The "Camera Image Database 2013" [14] is probably the first 'authentically' distorted dataset that was released for the purpose of image quality assessment. It consists of 480 images taken with 79 different devices, which are arranged in 8 clusters according to scene descriptions. Ghaiyaram and Bovik published the "LIVE in the Wild Image Quality Challenge Database" [15]. With 1162 images and over 350.000 opinions, it quickly became as popular as its artificially distorted predecessor.

The largest dataset according to our knowledge is the "Konstanz Image Quality Database 10k" [16], consisting of 10,073 source images that were assessed in a crowdsourcing study with roughly 1.2 million ratings. The purpose of this dataset is to serve as a training set for machine learning algorithms.

2.1.3 Miscellaneous Databases

There are noteworthy datasets besides those that were annotated especially for IQA. The "Waterloo Exploration Database" [17], which is taking a different approach to quality assessment, consists of 4744 original images and 94.880 artificially distorted derivatives. Impairments are introduced by JPEG, JPEG2000, white noise and Gaussian blur. Instead of having all images subjectively rated, the authors ran 20 published IQA metrics on their dataset and performed a systematic comparison, revealing discrepancies between known artificial distortions and predicted image quality for more than 6 million image pairs.

The "ImageNet" [18, 19] dataset was released 2010 and provides an "ontology of images built upon the backbone of the WordNet structure" [18]. It is probably best-known for the ImageNet large-scale visual recognition challenge (ILSVRC). Though not directly quality-related, this dataset was used to train deep convolutional neural networks from

scratch that were later fine-tuning to predict quality, see Section 2.2.3.
Closer related to image quality is the last example in this list, Vu *et al.*'s "CSIQ Local Image Sharpness Database" [20], which consists of only six images that were lab-annotated for locally perceived sharpness by eleven judges.

2.2 Objective Image Quality Assessment

Image quality assessment algorithms have been proposed merely in the last two decades. Generic fidelity metrics that originated in the signal processing domain have been known for a longer period and serve as a baseline for this particular task.

One usually distinguishes three classes of IQA algorithms in the literature, depending on the amount of information they require besides the image under assessment. Full-reference methods rely on the availability of a pristine, undistorted reference image and perform a comparison to the distorted version. No-reference algorithms, on the other hand, do not require further information at all, which is why they are also referred to as "blind" methods and deemed to be the most complicated. Reduced-reference algorithms require only some additional knowledge, e.g. the type of the predominant distortion in the image.

2.2.1 Representation of Visual Information

The way we interact with digital visual information is based on the Young-Helmholtz theory of trichromatic color vision [21], which states that the human eye is capable of interpreting mixtures of red, green and blue colored light. Devices at the endpoints as well as the internal representations are commonly designed in accordance with this principle.

Definition 1. A *digital image* is a matrix $\mathcal{I} \in \mathcal{C}^{m \times n}$, where $\mathcal{C}$ is a discrete, finite set of color values. Throughout this text, $\widetilde{\mathcal{I}}$ shall refer to a pristine, undistorted reference version of $\mathcal{I}$.

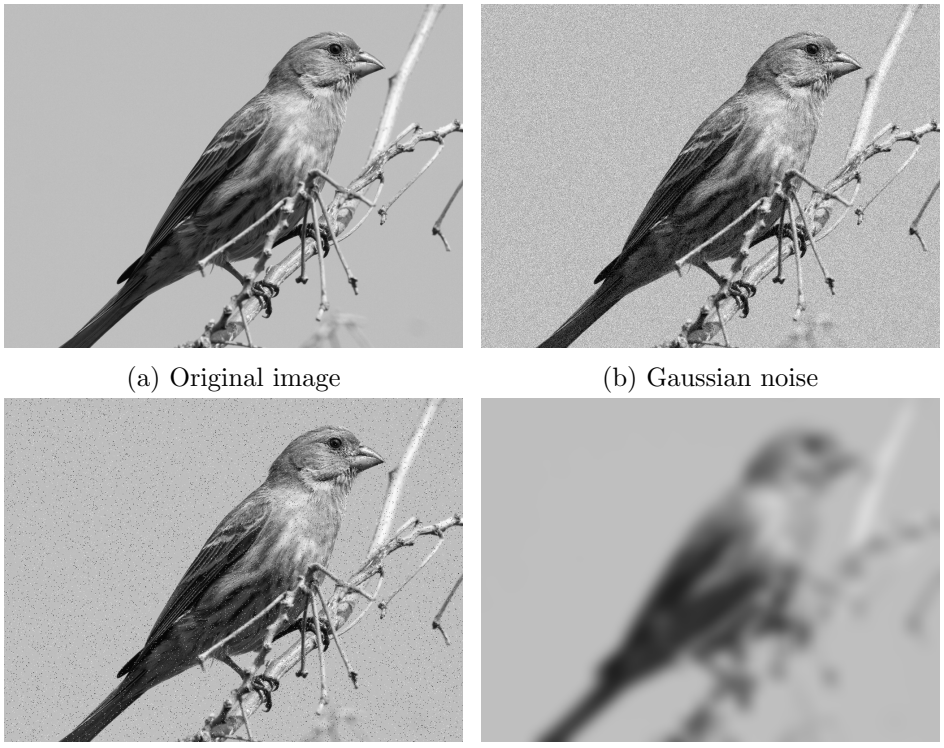
2.2.2 Full-Reference IQA

In signal processing, the *Mean Square Error* and the *Peak Signal to Noise Ratio* [22] are prominent signal fidelity metrics. For grayscale images where e.g. $\mathcal{C} = \{0, \ldots, 255\}$, they are defined as follows:

$$MSE(\mathcal{I}, \widetilde{\mathcal{I}}) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\mathcal{I}_{i,j} - \widetilde{\mathcal{I}}_{i,j}\right)^2 \tag{1}$$

$$PSNR(\mathcal{I}, \widetilde{\mathcal{I}}) = 10 \log_{10} \left(\frac{MAX_{\mathcal{I}}}{MSE(\mathcal{I}, \widetilde{\mathcal{I}})} \right) \tag{2}$$

Where $MAX_{\mathcal{I}}$ is the maximal possible value of an element of $\mathcal{I}$. For color images, Salomon [22] defines the *PSNR* on luminance values corresponding to the RGB values, which are calculated pixel-wise according to [23] as: $L = 0.299R + 0.587G + 0.114B$.



(a) Original image (b) Gaussian noise

(c) Salt-and-pepper noise (d) Gaussian blur

Figure 1: All distorted images have the same *MSE* of 480 relative to the original.

Despite their prevalence, the *MSE* and consequently the *PSNR* are improper metrics for perceived visual quality, as illustrated in Figure 1. All distorted images have the same *MSE* relative to the original but vary in quality. The simplicity of the *MSE* is also its flaw: its pixelwise, sign-independent formulation does not consider the rich structure that natural images exhibit. A study conducted by Huyn *et al.* [24] on the correlation of MOS values with *PSNR* for different video codecs confirms what is showcased in our example. Allegedly, the metric is not only unsuitable cross-content but also cross-distortion. A more thorough discussion of the *MSE* is given by Wang *et al.* [25].

A metric geared specifically towards perceived visual error is the "Structural Similarity Index" [26]. The approach uses three functions comparing luminance, contrast and structure. Contrast is measured by the standard deviation of an image and structure by the correlation between the distorted image and the pristine reference.

Further extending this paradigm shift from pixel-level to higher-level features, the "Feature Similarity" as proposed by Zhang *et al.* [27] constructs maps based on phase congruency and image gradient magnitudes, which are then combined to a global quality score. This approach is braced by a neuroscientific study by Henriksson *et al.* [28] that investigated visual cortex activity in relation to phase congruency phenomena using magnetic resonance imaging of the human brain.

2.2.3 No-Reference IQA

Not resorting to additional information besides the image under assessment renders no-reference or "blind" IQA algorithms the most challenging class in this comparison. Statistical image analysis influenced early efforts in this domain. As Srivastava *et al.* stated:

> "Even though images are expressed as elements of a large vector space (e.g. the space of rectangular arrays of positive numbers) [...], the subset of interesting images is rather small and restricted". [29]

Natural Scene Statistics is an umbrella term for algorithms which assume that perceptual distortions can be measured as deviations of certain statistical properties from those observed on unimpaired images. Moorthy *et al.* [30] presented an approach working in two modular stages that aims to identify the prevailing distortion first and assesses its severity afterward utilizing a tailored metric. BIQI, their reference implementation, employs a support vector machine that classifies wavelet coefficients, using the LIVE database [4] for training and evaluation. The "Distortion Identification-based Image Verity and INtegrity Evaluation Index" (DIIVINE), also published by Moorthy *et al.* [31] superseded BIQI's performance by far. The fundamental approach of correlating transform-domain coefficients with MOS values is also used, with modifications, in [32–34] and [35].

Machine Learning The increase in computing power during the last years enabled the application of machine learning models to problems that were previously too complex to solve. Neural networks as a prominent example achieve outstanding results, e.g. close to human performance on object recognition tasks [19, 36].
Kang *et al* [37] first experimented with convolutional neural networks for the task of image quality assessment. Operating on 32×32 pixel patches as an input, their shallow architecture consists of one convolutional layer with 50 kernels of 7×7 pixels, followed by simultaneous min- and maxpooling and a three-layered fully-connected predictor. Their method set a new state of the art for blind IQA and performed comparably to the best performing full-reference methods on the LIVE dataset.

In subsequent years, this approach has been further refined: Bosse *et al.* [38] proposed a deeper structure of ten convolutional layers, employing maxpooling after each second layer. A two-layered, fully connected head was used to estimate quality scores. Training was conducted on randomly sampled 32×32 pixel patches taken from the LIVE dataset.

Quality annotated datasets are rather small with at most a few thousand images in total, thus insufficient to train very deep neural networks from scratch. Bianco *et al.* [39] proposed transfer-learning [40] CNNs that were originally intended for classification tasks and have been pre-trained on the ImageNet dataset.

The practice of replacing or randomly re-initializing the fully-connected layers at the head of the network is reasonable:

> "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" [40]

The recently proposed DeepRN [12] by Varga *et al.* refines and extends this principle: a fine-tuned ResNet-101 [41] generates 2048 feature maps to which spatial-pyramid pooling [42] is applied in order to obtain a fixed-size output vector independent of the input image's resolution. This vector is normalized and fed into a fully connected, five-layered network that is trained not just to predict a single quality score per image, but the distribution of subjective votes [16] on a 5 point ACR [43] scale.

2.2.4 Reduced-Reference IQA

An IQA method belongs to this class if it does not rely on the availability of a reference image, but requires additional information besides the image under assessment. This includes methods that adapt their prediction based on knowledge about the predominant distortion type or that are designed to work exclusively on images affected by one specific distortion. Wang *et al.* [44] proposed an algorithm able to identify blurring and blocking effects introduced by JPEG compression that was intended as a precursor for the development of no-reference methods.

Furthermore, one can argue that a whole set of NR-IQA algorithms actually belongs to this category, despite being advertised differently by the respective authors. Methods that were developed with a specific artificially distorted database in mind, e.g. [30, 31], which work by identifying the predominant distortion within a known set of possible distortions. This constitutes a high degree of prior knowledge and one has to argue why such an algorithm would generalize to out-of-dataset inputs.

2.3 On the State of the Art and Possible Improvements

We can identify aspects in the field of image quality assessment that have not been touched upon yet and that would benefit from further research:

Databases: Existing machine learning methods that work on image patches are trained assuming that global MOS scores are valid throughout entire images from which training data is sampled [38]. We create a novel dataset of 32.000 individually quality-annotated image patches for training and testing of machine learning based IQA methods.

Locality: Most quality assessment algorithms proposed in the literature calculate a global score per image and are incapable of predicting local quality scores. Approaches that are theoretically able to do so, e.g. [38], are only evaluated on a global scale. We develop a local quality predictor based on a convolutional neural network. For validation,

we furthermore propose a second dataset of 125 images that were sampled from KonIQ-10k [16] and locally annotated for perceived visual quality in a crowd study.

Variable Compression: As an application, we use the quality maps created by our local model to substitute saliency maps generated by eye-tracking studies in the lab. They serve as a bit allocation scheme in an extended JPEG [45] compressor, enabling the fully automated application of a variable image codec that was previously developed by our group [2].

3 Concepts and Assumptions on Local Image Quality

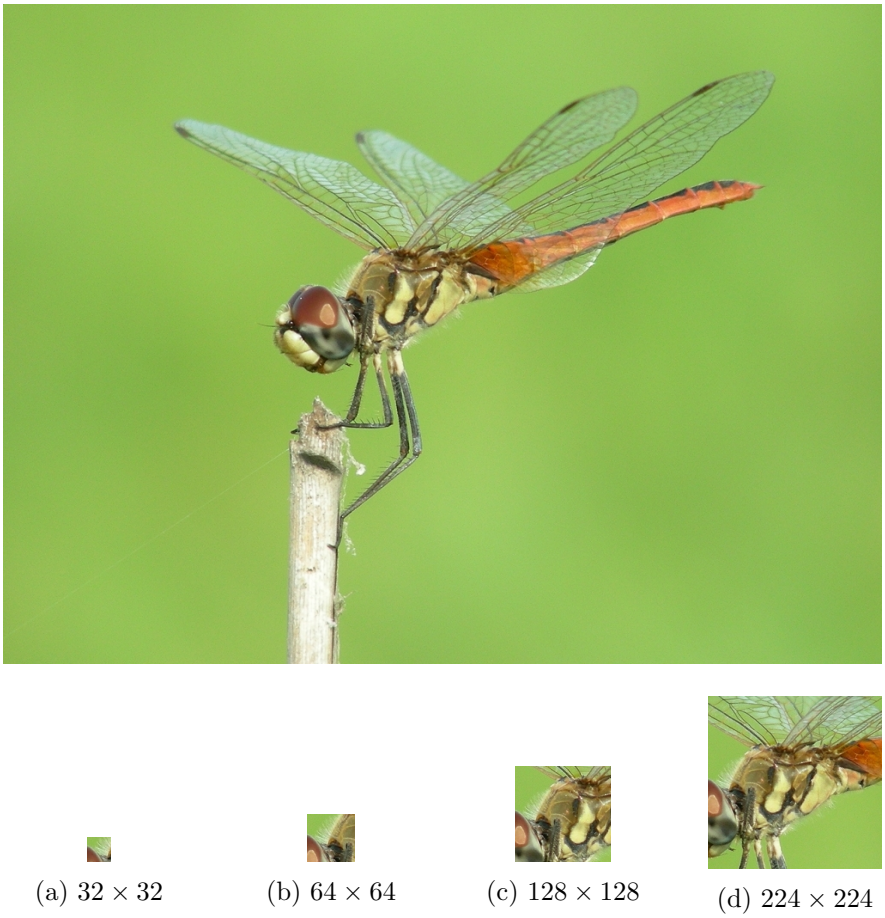
This section aims to introduce the philosophy behind our understanding of image quality and explains what we are trying to quantify in the subsequently presented experiments. Particular domains such as e.g. medical imaging may have obvious measures for the value of an image, such as the resolution of an MRI scan. The same does not necessarily hold for natural images in the sense of photography. While research on the human visual system and the psychophysiological processes involved in visual perception [28, 46] is relevant for this topic, we chose to approach perceptual image quality from a data scientific perspective.

Humans mean opinion scores on image quality correlate highly between subjective studies, as shown by Hosu *et al.* [47]. Quality has been studied extensively as an agreement based property on a global, per-image scale. We make the following assumptions to study perceptual quality on smaller image regions:

1) A single, isolated pixel can not hold quality information.
2) Quality is a reasonable concept only on sufficiently large image patches.
3) Image content can influence perceived quality.

Determining an appropriate patch size involves a tradeoff: an IQA method that is capable of assessing smaller patches allows higher spatial precision for image segmentation and is likely of lower computational complexity due to the lower input dimensionality. However, small patches are more difficult to assess for humans: Figure 2 depicts a source image with exemplary choices for patch sizes. The smallest size of 32×32 pixels was used in an IQA method by Bosse *et al.* [38]. The largest size of 224×224 pixels is the standard input size for ImageNet classification models. We hypothesize that judgments of quality are affected by expectations on how content should be depicted in images. Large patches may introduce related biases due to e.g. preferences on the motif. In this sense, we tried to choose a patch size that is as small as possible but as large as required.

We found 64×64 pixel patches sampled from 1024×768 pixel images to be a resolution that is assessible for humans but that does not reveal unnecessary content cues. This ad-hoc choice is an aspect of our work that is likely to be not defendable against all objections, but at this point a decision had to be made.



(a) 32×32 (b) 64×64 (c) 128×128 (d) 224×224

Figure 2: Content perceptibility relative to patch size.

3.1 KonPatch

Based on our assumptions on local image quality we created KonPatch, which is a novel dataset of 32.000 individually annotated 64×64 pixel patches that is intended as a training set for local quality prediction models. The dataset was generated as follows: we randomly selected 500 images from KonIQ-10k [16]. Those were excluded from the remaining dataset to guarantee that subsequent experiments are never carried out on images which our local predictor was (partially) trained on. From each of the selected 500 images, we sampled 64 patches at random locations. This set of $32,000$ patches was annotated with binary labels in an attempt to flag patches that look *not* indicative of being sampled from an high-quality image. Initially, a lab experiment with only one vote per patch was conducted. Each patch p has an associated source image $\mathcal{I}$ in KonIQ-10k with a global MOS score $mos(\mathcal{I})$. This allows building a continuously rated dataset on top of a binary classification study, by defining the score $\mathcal{S}$ of a patch p as follows:

$$\mathcal{S}(p) = \begin{cases} mos(\mathcal{I}) & \text{if } p \text{ was marked as high quality} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Additionally, a crowd experiment was conducted on this dataset, but the results were not used in this work. Histograms of the score distributions for both versions of the dataset are given in Section 9.1, however a thorough comparison is left as future work.

4 Neural Networks for Image Quality Assessment

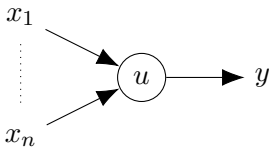
This section introduces the concepts used to build a local, no-reference image quality predictor. The assumption is that image quality can be modeled as a function

$$\psi : \mathcal{C}^{64\times64\times3} \to \mathbb{R},\ p \mapsto \psi(p)$$

where $\mathcal{C} = \{0, \ldots, 255\}$ is the set of possible values for each channel of a 24 bit RGB image. Instead of defining ψ directly, we will use a machine learning approach to construct an approximation with respect to the KonPatch dataset.

4.1 Neural Networks

Neural networks are biologically inspired computational models build of simple numeric processing units that mimic the functionality of neurons as found in the human brain. A neuron u takes a vector $x \in \mathbb{R}^n$ as an input and returns a value $y \in \mathbb{R}$, more specifically:



$$u(x) = \varphi\left(\sum_{i=1}^{n} \hat{w}_i x_i + b\right) \tag{4}$$

Figure 3: Schematic and specification of a neuron.

where $\hat{w} \in \mathbb{R}^n$ is a weight vector, $b \in \mathbb{R}$ is a bias and $\varphi : \mathbb{R} \to \mathbb{R}$ is a non-linear *activation* function. This representation can be simplified by eliminating the distinction between weights and bias by introducing an 'always-on' input and concatenating the bias with the weight vector:

$$\hat{w}^\top x + b = \begin{bmatrix} b & w^\top \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}$$

In the context of a neuron η, the terms 'weights' and 'network input' subsequently refer to the augmented vectors $w_\eta := \begin{bmatrix} b & \hat{w}^\top \end{bmatrix}$ and $in_\eta := \begin{bmatrix} 1 & x \end{bmatrix}^\top$. Considering the affine transformation as the *network input function* of a neuron is a convenient perspective for cases where w is not treated as a given model parameter, e.g. in model optimization.

$$f_{net} : \mathbb{R}^{(n+1)} \times \mathbb{R}^{(n+1)} \to \mathbb{R},\ (in, w) \mapsto w^\top in \tag{5}$$

Definition 2. A neural network is a directed graph $(\mathcal{N}, C)$, where $\mathcal{N}$ is a set of neurons and $C \subseteq \mathcal{N} \times \mathcal{N}$ is a set of connections between neurons.

This structure allows deducing the concepts of successor and predecessor sets for a neuron u, which are defined as:

$$succ(u) = \{v | (u, v) \in C\}$$
$$pred(u) = \{v | (v, u) \in C\}$$

One requires a partition[1] of $\mathcal{N}$ into three subsets such that

$$\mathcal{N} = \mathcal{N}_{\text{in}} \cup \mathcal{N}_{\text{hidden}} \cup \mathcal{N}_{\text{out}}$$

A neuron is called *input*, *hidden* or *output* neuron according to the partition it is associated with. All neurons in a neural network have to respect the underlying graph structure such that the dimensions of the network input functions are compatible.

One identifies the activation level of input neurons componentwise with the input vector $x \in \mathbb{R}^n$ that is presented to the network by setting $u_i = x_i$ for all $u_i \in \mathcal{N}_{\text{in}}$. Analogously, the network's output $y \in \mathbb{R}^m$ is defined as the activations of the output neurons by setting $y_j = \eta_j$ for $\eta_j \in \mathcal{N}_{\text{out}}$.

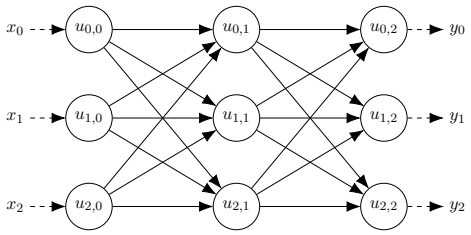


Figure 4: A fully-connected neural network with three layers.

Under mild assumptions on the activation functions, a multi-layer network can approximate a large class of functions up to arbitrary precision given that it is equipped with sufficiently many neurons, as shown by Hornik [48]. Before discussing how to construct suitable weights, we will introduce an extension to plain neural networks that is better suited for image processing tasks.

4.2 Convolutional Neural Networks

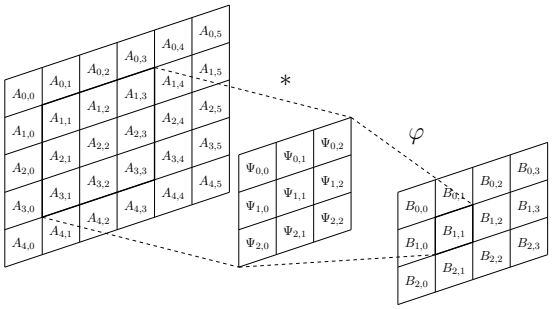
A drawback of fully-connected neural networks is that they do not exploit the structure of images. Local relationships between neurons are not spatially independent and relevant patterns have to be represented by weights and biases at *all* possible locations. *Convolutional Neural Networks* were proposed by Lecun in 1995 [49].

[1] Therefore $\mathcal{N}_{\text{in}}, \mathcal{N}_{\text{hidden}}$ and $\mathcal{N}_{\text{out}}$ are non-empty and mutually exclusive and $\mathcal{N}_{\text{hidden}} \cap (\mathcal{N}_{\text{in}} \cup \mathcal{N}_{\text{out}}) = \emptyset$

They entail a concept of spatial proximity that can be understood from two perspectives:

a) Convolution of the input feature map with a kernel in the spatial domain.

b) Weight-sharing between neurons at different positions within a layer.



$$B_{i,j} = \varphi(A * \Psi) \tag{6}$$
$$= \varphi\Big(\sum_{u=0}^{k}\sum_{v=0}^{k} A_{i+u,j+v} \cdot \Psi_{u,v}\Big)$$

Figure 5: Schematic drawing of a convolutional layer with a single 3×3 kernel.

Convolutional layers accept an input tensor $A \in \mathbb{R}^{m\times n\times d}$ and convolve it spatially with a 'kernel' $\Psi \in \mathbb{R}^{k\times k\times d}$. Despite the name, this operation is usually implemented as cross-correlation in most libraries. A non-linear activation function φ is applied to the result, yielding an output 'feature map' B. Figure 5 illustrates this process, for $A \in \mathbb{R}^{4\times 5}$ and $\Psi \in \mathbb{R}^{3\times 3}$. Formula (6) describes the performed computation. It is common to perform this operation with multiple kernels per layer, whereafter the resulting two-dimensional feature maps are concatenated along a third dimension (stacked). Zero-padding can be used to prevent decreasing the spatial dimension in each layer as required by the cross-correlation, but is omitted in Figure 5 for simplicity.

Pooling A pooling layer performs a spatial dimensionality reduction of the feature maps it receives as an input. This is conventionally done by partitioning the input feature maps into disjoint submaps of quadratic shape, however non-quadratic and overlapping submaps are possible. A function $\varphi : \mathbb{R}^{m\times n} \to \mathbb{R}$ is applied to each submap of shape $m \times n$ independently. Pooling does neither constitute a recombination of values between feature maps nor change the number of feature maps.
As Goodfellow states, "pooling helps to make the representation become approximately invariant to small translations" [50]. Many tasks, such as image quality assessment, do not require pixel-level accuracy. Pooling can therefore reduce the computational cost by deliberately ignoring irrelevant information. A popular implementation is maxpooling, which replaces feature map activations in a certain environment with the maximal present activation. Other popular choices include for example minpooling and average-pooling.

In applications with fixed input- and output dimensions it is common to combine convolutional and fully-connected layers. This is implemented by vectorizing the output feature maps of the last convolutional layer and passing them as an input to subsequent convolutional layers.

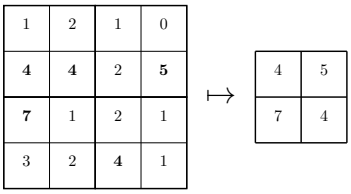


Figure 6: Maxpooling with a 2×2 window and a stride of 2.

4.3 Parameter Estimation from Training Examples

So far, our framework allows applying neural networks only in a 'forward' fashion assuming given weights for each neuron. This section introduces the concepts that enable to approximate a function by choosing a neural network architecture and 'learning' from examples.

4.3.1 Measuring Model Error in Supervised Learning

Supervised learning [50] is an approach to create a machine learning model $M : X \to Y$ that relies on the availability of training examples

$$L = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

For each *data point* x_i there exists a *label* y_i that represents the desired output value. Creating sufficiently large datasets is a burden that usually renders supervised learning costly, but it has shown to achieve outstanding results in domains such as object recognition [51]. In KonPatch, data points and labels are given as 64×64 pixel RGB patches and quality scores in $[0, 1]$.

In supervised learning, a given model instance M is repeatedly evaluated for its 'fitness' with respect to L in terms of a loss function[2]

$$J_L : F(X, Y) \to \mathbb{R}_0^+,\ M \mapsto J_L(M)$$

which is required to be representable in terms of pointwise losses as

$$J_L(M) = \sum_{(x,y)\in L} e_Y(M(x), y) \tag{7}$$

[2] $F(X, Y)$ is the space functions $X \to Y$

where $e_Y : Y \times Y \to \mathbb{R}_0^+$ is differentiable, and finally, if M is a neural network, in terms of component-wise losses as

$$J_L(M) = \sum_{(x,y)\in L} \sum_{1\leq i\leq m} e(u_i, y_i) \tag{8}$$

again, $e : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_0^+$ is a differentiable and $u_1, \ldots, u_m \in \mathcal{N}_{\text{out}}$. Supervised learning aims to iteratively minimize J_L by adapting the model with respect to the training data.

4.3.2 Gradient Descent

Gradient descent [50] is an approach to minimize a function

$$f : \mathbb{R}^n \to \mathbb{R}$$

starting from a point $x_0 \in \mathbb{R}^n$ by performing iterative updates according to f's gradient:

$$x_{n+1} = x_n - \lambda \cdot \nabla f|_{x_n}$$

Where $\lambda \in \mathbb{R}^+$ is a parameter governing the step size. Given a neural network M which is parameterized by a family of weights W and evaluated with respect to a loss function J on a training set L, it is possible to employ this procedure to approximate

$$\underset{W}{arg\,min}\ J_L(M_W)$$

As this problem is non-convex in general, gradient descent is not guaranteed to converge to a global optimum. Proposed by Augustin-Louis Cauchy [52] in 1847, the algorithm has since been extended in multiple ways.

Stochastic Gradient Descent is a variant of gradient descent that is popular in machine learning, as it helps to circumvent memory limitations. Computing gradients for all training examples is expensive due to the number of parameters in modern neural network architectures, the size of the required datasets and the resulting number of intermediate results that have to be stored. Stochastic gradient descent [50,53] approximates the true gradient at point W on a subset $L' \subseteq L$ of the dataset:

$$\nabla J_L(M_W)\big|_W \approx \nabla \sum_{(x,y)\in L'} e_Y(M_W(x), y)\Big|_W$$

The optimization step is performed with the approximate gradient instead of the true gradient. The cardinality of L' is called batch size. For a single training example per step this procedure is an instance of *online learning* [54].

Momentum is an extension that introduces a dependency between optimization steps:

$$\Delta x_{n+1} = \lambda \cdot \nabla f|_{x_n} + \alpha \Delta x_{n-1}$$
$$x_{n+1} = x_n - \Delta x_n$$

The initial momentum is defined as zero by setting $\Delta x_0 = 0$ and $\alpha \in \mathbb{R}^+$ is a parameter controlling the previous step's influence on the current step. A discussion of the effect of this extension on the training of deep neural networks is given by Sutskever *et al.* [55].

While there exist further independent extensions to gradient descent, e.g. *averaging* as proposed by Polyak *et al.* [56], it is common practice to use an algorithm that combines multiple improvements, such as Adam [57].

4.3.3 Backpropagation

Explicit gradient computations in neural networks rely on an algorithm known as backpropagation [58]. In a neural network $M = (\mathcal{N}, C)$, we consider a neuron $u \in \mathcal{N}$ parameterized by a weight vector w_u. By exploiting formula (7) and the linearity of differentiation it suffices to examine the loss $e := e_Y(M(x), y)$ for a single pattern $(x, y) \in L$ in the training set.

The output of u depends on w_u only through $net_u = f_{net}(in_u, w_u)$, thus it is

$$\begin{aligned} \nabla e|_{w_u} &= \frac{\partial e}{\partial net_u} \frac{\partial net_u}{\partial w_u} \\ &= \frac{\partial e}{\partial net_u} in_u \end{aligned} \tag{9}$$

by the chain rule and the fact that $f_{net}(in_u, w_u) = w_u^\top in_u$. The first factor depends on the choice of e_Y. For regression tasks such as the one imposed by KonPatch, the (mean) squared error is a common choice.

$$e_Y(M(x), y) = \sum_{v \in \mathcal{N}_{out}} (y_v - out_v)^2$$

where $out_v \in \mathbb{R}$ is the activation of output neuron v with corresponding training label $y_v \in \mathbb{R}$. Thus, the first factor in Formula 9 can be rewritten as

$$\frac{\partial e}{\partial net_u} = \sum_{v \in \mathcal{N}_{out}} \frac{\partial (y_v - out_v)^2}{\partial net_u} = -2 \sum_{v \in \mathcal{N}_{out}} \big(y_v - out_v\big) \frac{\partial out_v}{\partial net_u} \tag{10}$$

In case u is an output neuron:

$$\nabla e|_{w_u} = -2(y_u - out_u) \frac{\partial out_u}{\partial net_u} in_u$$

In case u is not an output neuron, the measurable error at the output neurons is only indirectly dependent on w_u and net_u through the successor neurons of u. First, we define an auxiliary variable based on Formula 10:

$$\text{Let } \delta_u = \sum_{v \in \mathcal{N}_{out}} \left(y_v - out_v\right) \frac{\partial out_v}{\partial net_u}$$

by application of the chain rule and exchanging the sums:

$$\begin{aligned}\delta_u &= \sum_{v \in \mathcal{N}_{out}} \sum_{s \in succ(u)} \left(y_v - out_v\right) \frac{\partial out_v}{\partial net_s} \frac{\partial net_s}{\partial net_u} \\ &= \sum_{s \in succ(u)} \delta_s \frac{\partial net_s}{\partial net_u}\end{aligned}$$

due to the graph structure, out_u is one component in in_s, which is a parameter to the affine transformation resulting in net_s. Therefore it is

$$\frac{\partial net_s}{\partial net_u} = w_{su} \frac{\partial out_u}{\partial net_u}$$

where $w_{su} \in \mathbb{R}$ is the component in $w_s \in \mathbb{R}^{|pred(s)|+1}$ associated with out_u. Consequently, we arrive at the following formula

$$\delta_u = \Big(\sum_{s \in succ(u)} \delta_s w_{su} \Big) \frac{\partial out_u}{\partial net_u}$$

Combined with Formula 10, we get a recursive definition of the gradient:

$$\nabla e|_{w_u} = -2 \Big(\sum_{s \in succ(u)} \delta_s w_{su} \Big) \frac{\partial out_u}{\partial net_u} in_u$$

The last factor containing partial derivatives depends on the choice of the activation functions. In our models we use the ReLu and the Sigmoid function for this purpose:

$$relu(t) = max(0, t) \qquad \frac{\partial relu}{\partial t} = \begin{cases} 1 \text{ if } t > 0 \\ 0 \text{ if } t < 0 \\ \text{undefined if } t = 0 \end{cases}$$

$$\sigma(t) = \frac{1}{1 + e^{-t}} \qquad \frac{\partial \sigma}{\partial t} = \sigma(t)(1 - \sigma(t))$$

We follow the common strategy of defining the ReLu derivative as 0 for $t = 0$.

5 Image Quality Models and Experiments

5.1 Patchnet

Patchnet is a convolutional neural network with 14 layers in total. An overview of the architecture is given in Fig. 7. In this depiction, the leftmost plane represents a 64×64 pixel input patch. From the left to the right, the data passes 7 *convolutional layers* whose output tensors are drawn as cuboids and 3 *maxpooling* layers, represented by dashed funnels. The output feature maps of the last convolutional layer are vectorized and passed to a four-layered, fully connected predictor with 1024, 16, 8 and finally 1 neuron. Rectified Linear Units [59] are used as activation functions except for the last predictor neuron, which uses a sigmoid function to bound the output to $[0, 1]$.

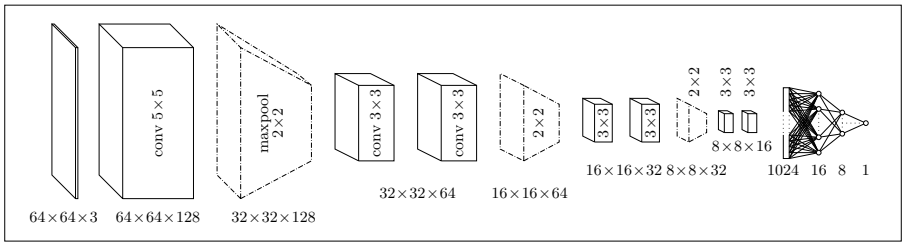


Figure 7: Architecture of Patchnet.

5.1.1 Training

We randomly selected 20% from KonPatch for testing, completely excluding those data points from the training procedure. The remaining set of 25.600 labeled patches was split in five parts of equal size for cross-validation. For each part, we ran a separate training instance starting from randomly initialized weights. In the i-th training instance, the i-th part of the training set was used for validation, while the remaining 4 parts were used in the actual optimization.

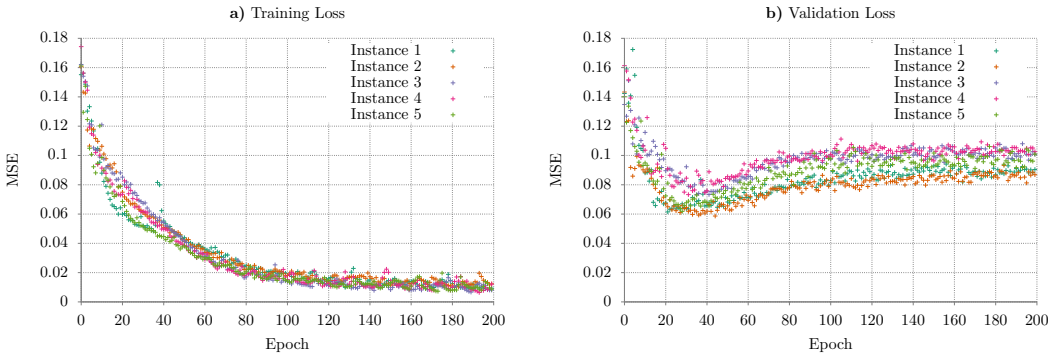


Figure 8: Training and validation losses per epoch.

The model starts to overfit after roughly 30 epochs, as the validation loss in Figure 8b indicates. Since Patchnet is a small CNN compared to recently published architectures we chose not to use Dropout [60], but early stopping [58] as a mitigation to overfitting. For this purpose, we stored the model with the best performance on the respective validation set for each of the five training runs. The resulting model instances show consistent performance on the test set, as shown in Table 1.

Instance	MSE	MAE	SROCC	PLCC
1	0.0611	0.1472	0.677	0.752
2	0.0616	0.1437	0.670	0.748
3	0.0680	0.1493	0.654	0.718
4	0.0711	0.1784	0.649	0.710
5	0.0592	0.1506	0.678	0.757

Table 1: Performance metrics for each instance on the test set.

Patchnet is implemented in Keras [61]. We use an Adam [57] optimizer with a batch size of 512. Training was performed on Nvidia K40 GPUs with TensorFlow [62] as a backend.

5.2 Indicator Map Generation

As Patchnet allows predicting perceptual quality on local image regions, it can be applied to whole images in a sliding window fashion as shown in Figure 9.

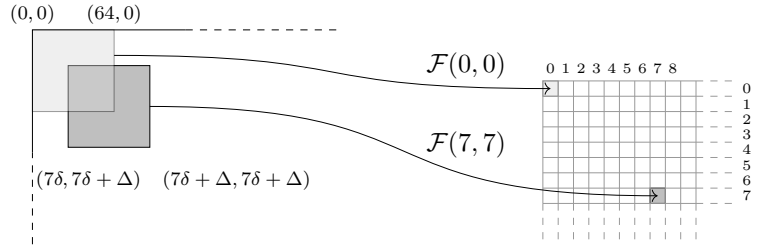


Figure 9: Application of Patchnet to whole images.

The parameter δ adjusts the step size between adjacent applications of the model, the window size Δ is fixed to 64 pixels due to the model's specification. For a given input image of $m \times n$ pixels, this procedure results in an output feature map of dimension

$$\left\lfloor \frac{m-\Delta}{\delta} \right\rfloor \times \left\lfloor \frac{n-\Delta}{\delta} \right\rfloor$$

A spatial subsampling of the input image can be performed by choosing $\delta > 1$.

5.3 Local Quality Assessment on Entire Images

There exists no published local image quality database to the best of our knowledge. In order to gain some insight into the performance of Patchnet, we created an auxiliary dataset. We sampled 125 images from KonIQ-10k [16] and ran a crowdsourcing experiment to generate local image annotations.

For each image, we asked a total of 30 participants to do either of the following:

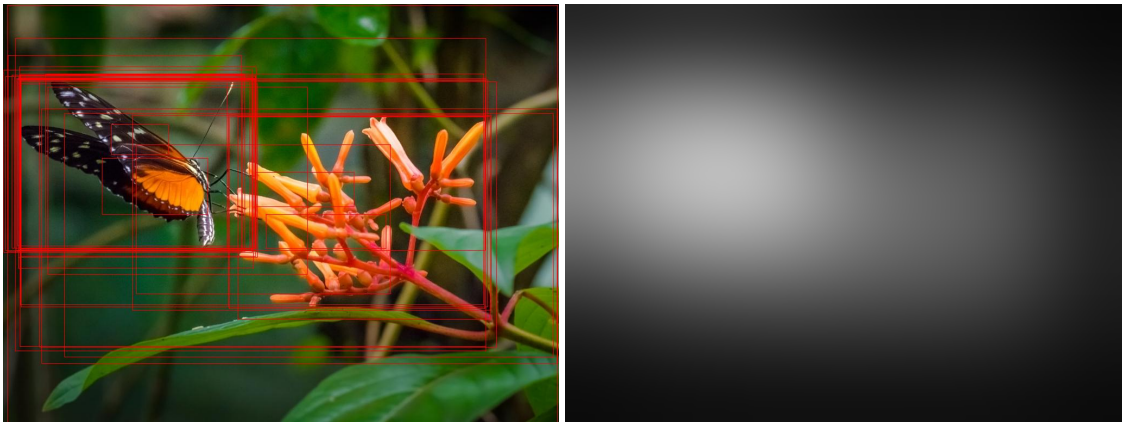
a) Place bounding boxes tightly around areas they deemed as being of high quality.

b) Select a checkbox to indicate that there are no high-quality areas in the image.

The quality score for a pixel $\mathcal{I}_{i,j}$ is calculated as the normalized vote count:

$$q_{i,j} = \frac{\#\text{bounding boxes that included } \mathcal{I}_{i,j}}{\#\text{participants}} \tag{11}$$

Areas within an image that were marked multiple times by the same user were counted only once to prevent unintended peaks from overlapping bounding boxes. Due to their rectangular shape, bounding boxes can mismatch the area that study participants intended to select. This may yield an imperfect segmentation but it is still chosen over manual, pixel-wise segmentation for practical reasons. As a mitigation for this problem, we post-processed the raw feature maps using a Gaussian filter with a standard deviation of 10% of the image width.



(a) Rectangles indicate selected areas. (b) Post-processed qualitymap.

Figure 10: Creating a local quality dataset.

It is noteworthy that the resulting quality maps indicate "absolute" quality, not just relative differences within an image, due to the normalization with respect to the number of study participants. The resulting quality maps are used as ground-truth data to benchmark Patchnet's predictions.

5.3.1 Experimental Results: Local Model Performance

In accordance with our dual understanding of image quality we investigate our model's performance not only on isolated patches as done in Section 5.1.1, but also in terms of predicting quality maps for whole images to examine whether the local predictions match the perception of humans.

As Patchnet was trained on a different dataset than the one created for this evaluation, quantifying the model's performance is not as straightforward as calculating the MSE in this case. We chose a measure based on the Receiver Operating Characteristic (ROC) [63] that works as follows: Let $G \in [0,1]^{n \times m}$ be a ground truth quality map, $P \in [0,1]^{n \times m}$ a prediction created by applying Patchnet in a sliding window fashion with a stride of 1. The result is centered and zero-padded to match G's resolution. We apply blurring using a Gaussian filter with a standard deviation equal to 10% of the image width to make the predictions more robust against small spatial shifts.
The ground truth map is binarized using a threshold t_G by setting it to 1 for all values greater than t_G and 0 otherwise. For a sufficient number of prediction thresholds $t_P \in [0,1]$, one can plot the true positive rate versus the false positive rate. The connecting line of these points is the ROC Curve. The area under this curve is a measure of the binary prediction performance of the model for a thresholded feature map.

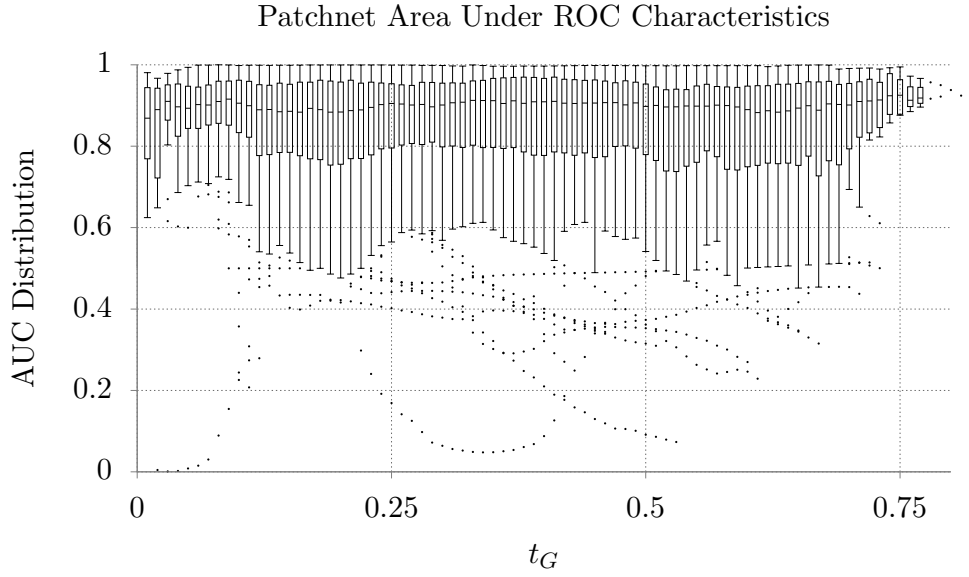


Figure 11: Boxplot of the AUC versus ground truth thresholds.

Figure 11 presents statistics about the AUC distributions depending on the t_G threshold on the horizontal axis. Each box covers the distribution from the 0.25 to the 0.75 quartile with a marker for the median. The whiskers extend to the most distant point whose value lies within 1.5 times the interquartile range. Outliers are represented with dots. We omitted plotting statistics for $t_G > 0.8$ the AUCs are undefined in this range.

5.4 Global MOS Prediction by Feature Aggregation

In addition to local quality experiments, we conducted global, per image MOS predictions based on spatially small feature maps. The purpose of this effort is to gain insight into the relationship between local and global image quality and to compare the performance of our model to IQA methods found in the literature.
We applied the sliding window approach depicted in Figure 9 with a stride of 4 to the remaining 9500 images from KonIQ-10k that were not used for patch selection earlier. Thus, the obtained feature maps are only 5.5% of the input image's resolution. We want to investigate how local quality patterns in images correlate with global scores.

	KonIQ-10k		LIVE in the Wild	
	SROCC	PLCC	SROCC	PLCC
Patchnet	0.667	0.573	0.512	0.527
FISH	0.560	0.513	0.500	0.503

Table 2: Correlation coefficients for mean feature map values with global MOS scores.

The correlation coefficients of mean feature map values relative to global MOS scores are presented in Table 2 as a sanity check. Our model outperforms FISH [64], a wavelet transform based measure proposed by Vu *et al.* that is pitched as a local sharpness metric in the original publication. It was applied to the input images using the same sliding window approach. The correlations not particularly strong, which is expected when simply taking the mean of the feature maps.

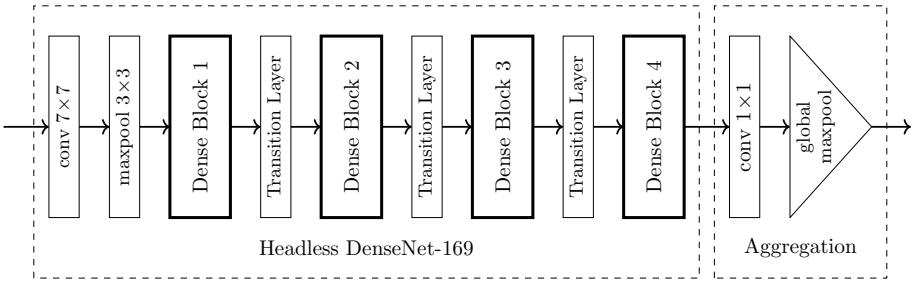
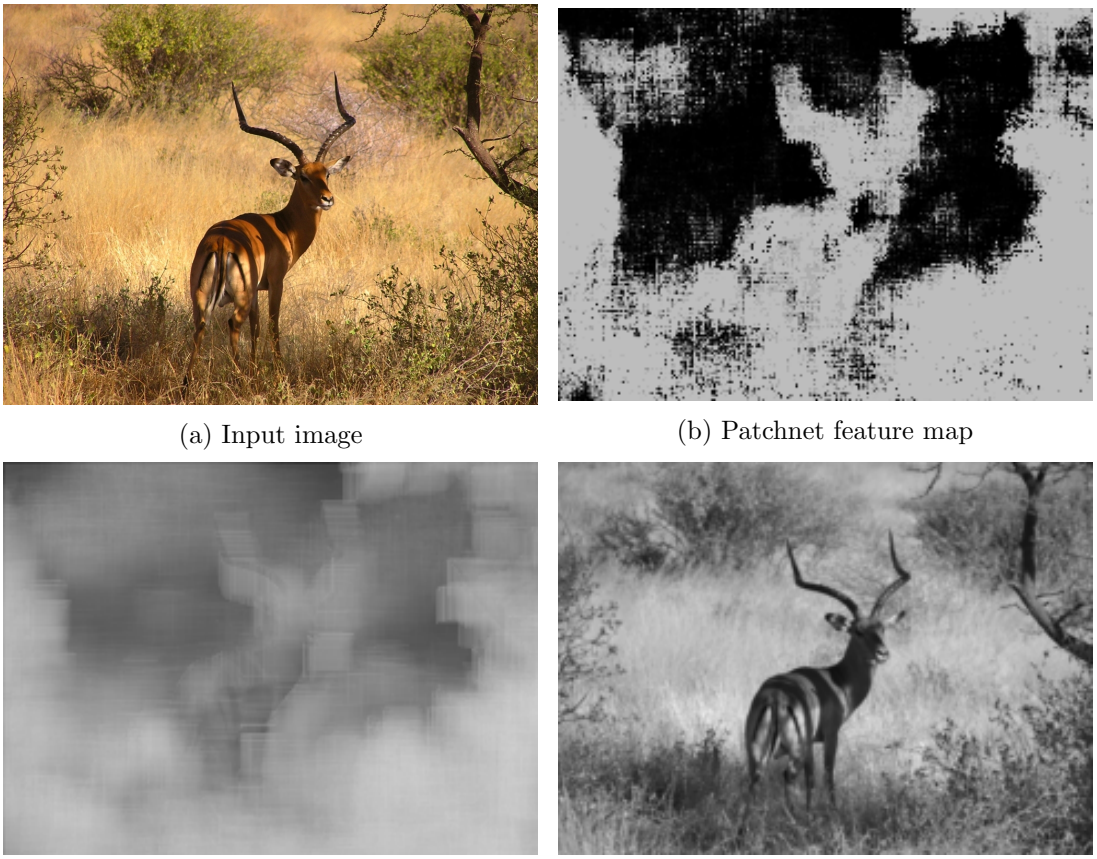


Figure 12: DenseNet-169 based meta aggregation model for global MOS prediction.

To predict global quality scores as precise as possible, we trained a neural network on top of data resulting from a feature combination. Additionally to Patchnet maps, we utilized FISH sharpness maps and a downscaled grayscale version of the original input image. Examples are given in Figure 13. We split the dataset of 9,500 images according to the commonly used 60/20/20 scheme into training, validation and test sets. Training data was artificially augmented: each of the 5700 images was taken once unmodified and in three versions that were randomly rotated between $+10°$ and $-10°$. We cropped the images to the rectangle of maximal valid size, cutting off any undefined regions that

were introduced by the rotation. Additionally, all training images were flipped both horizontally and vertically, which resulted in a total 64,400 feature maps of 82 different resolutions.



(a) Input image

(b) Patchnet feature map

(c) FISH feature map

(d) Grayscale / luminance map

Figure 13: Example image and resulting feature maps.

Very deep convolutional neural networks with shortcut connections have been proposed recently in the image classification community as a mitigation for the common problems of gradient decay and overfitting [41, 65]. We chose a DenseNet-169 architecture [66] that was geared towards the task of global MOS regression by exchanging the network head. Instead of class probabilities, a simple 1×1 convolutional layer followed by global maxpooling is used. This head intuitively performs a reweighing of the three-dimensional DenseNet output feature maps into a two-dimensional global quality map and returns the highest prediction as a score, which has the advantage that the model is not constrained to an input of a specific resolution. The structure of this meta aggregation model is given in Figure 12.

5.4.1 Experimental Results: Meta-Aggregation Performance

We report the correlation coefficients for the test set predictions of our meta-model in comparison to traditional, well-known no-reference IQA methods in Table 3. The correlation coefficients for KonIQ-10k are taken from [16] for BIQI, BLIINDS-II, BRISQUE; DIIVINE and SSEQ and from [12] for the remaining methods. The highest value per database and correlation measure is printed in bold.

	KonIQ-10k		LIVE in the Wild	
	SROCC	PLCC	SROCC	PLCC
BIQI [30]	0.54	0.61	0.29	0.38
BLIINDS-II [32]	0.57	0.58	0.44	0.48
BRISQUE [33]	0.70	0.70	0.59	**0.63**
DIIVINE [31]	0.58	0.62	0.43	0.46
SSEQ [34]	0.59	0.61	0.45	0.50
Our Model	**0.79**	**0.81**	**0.60**	0.62

Table 3: Correlation coefficients in comparison to traditional NR-IQA methods.

A comparison to recent, machine learning based methods is given in Table 4.

	KonIQ-10k		LIVE in the Wild	
	SROCC	PLCC	SROCC	PLCC
KangCNN [37]	0.63	0.67	0.71	0.73
BosICIP [38]	0.65	0.67	0.70	0.70
DeepBIQ [39]	0.90	0.92	0.89	0.91
DeepRN [12]	**0.92**	**0.95**	**0.91**	**0.93**
Our Model	0.79	0.81	0.60	0.62

Table 4: Correlation coefficients for recent IQA methods.

The horizontal line indicates a paradigm shift. BosICIP and KangCNN work, similar to the precursor we employed as a sanity check, by averaging a number of predictions made on spatially small image patches. The lower two methods DeepBIQ and DeepRN on the other side have access to larger areas, respectively even the whole input image, and were transfer-learned from ImageNet classification models.

6 Application: Variable Compression

This section presents an application of Patchnet in the domain of image compression.

6.1 Standard JPEG Compression

As an introduction, a summary of the JPEG [45,67,68] compression algorithm is given. It takes an *RGB* image of 24 bits per pixel as an input and initially performs a coordinate transformation into the *YCbCr* color space according to [67]. Each unsigned integer representing a channel component in the image is shifted from range $\{0, \ldots, 2^8 - 1\}$ to a signed integer in $\{-2^{8-1}, \ldots, 2^{8-1} - 1\}$. The resulting representation is spatially divided into non-overlapping 8×8 pixel blocks which are transformed into the DCT domain according to formula (12). Here, $f_c(x, y)$ is the value of channel c at position (x, y):

$$F_c(u, v) = \frac{1}{4} C_u C_v \left(\sum_{x=0}^{7} \sum_{y=0}^{7} f_c(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right) \tag{12}$$

$$\text{where } C_u, C_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases}$$

As "sample values typically vary slowly from point to point" [45], the DCT will accumulate most of the signal's energy in its low-frequency components. The ISO standard [68] defines separate quantization tables Q_l, Q_c for the luminance and chrominance channels. In accordance with RFC2435 [69], a user-provided value $q \in \{1, \ldots, 99\}$ is used to scale these tables coefficientwise according to the following scheme:

$$s = \begin{cases} 5000/q & \text{for } 1 \leq q \leq 50 \\ 200 - 2q & \text{for } 51 <= q <= 99 \end{cases}$$

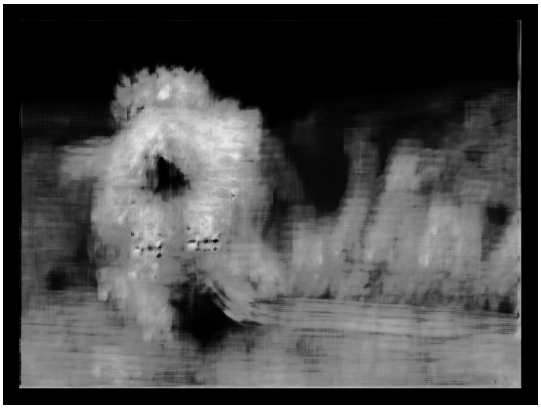
$$Q_s(u, v) = (Q(u, v) \cdot s + 50)/100$$

With subsequent clipping to the range of 8 bit unsigned integers. In the next step, the 8×8 matrices of DCT coefficients are quantized by division and subsequent rounding:

$$\bar{F}_c(u, v) = \left[\frac{F_c(u, v)}{Q_s(u, v)} \right]$$

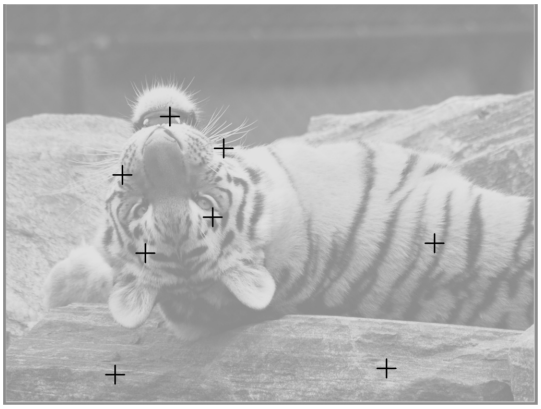
Finally, the quantized matrices are reordered in a "zig-zag" sequence and entropy coded to further reduce file size without additional loss [68].

(a) Original image at 6.637 bpp.

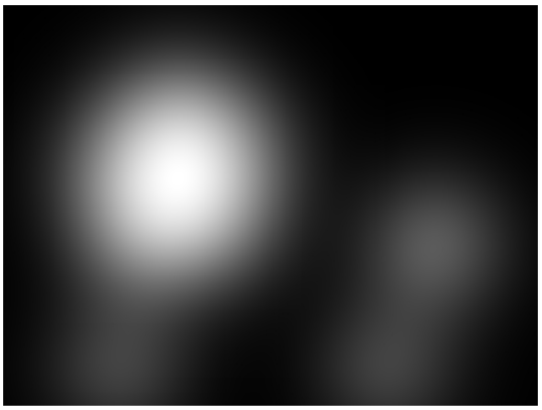


(b) Zero padded Patchnet qualitymap.

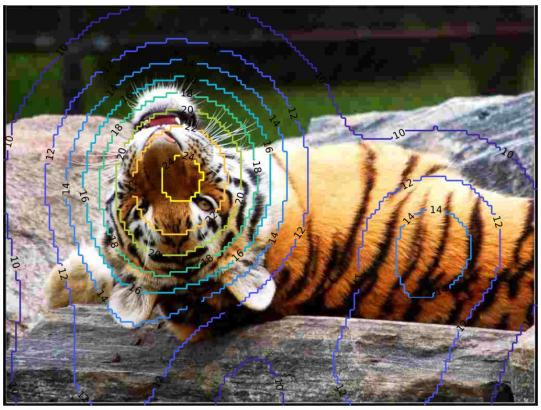
(c) Pruned, binarized qualitymap.



(d) Cluster centroid positions.



(e) Blurred centroids.



(f) Quality level contours at 0.304 bpp.

Figure 14: Intermediate results in variable JPEG compression.

6.2 Variable JPEG Compression

A restriction of JPEG is the global quality parameter q, which is used to scale the quantization tables for each 8×8 block in an entire image. Enforcing uniform quality is not always reasonable. Certain areas within an image may be of higher interest to an observer than others. Our group already proposed an extension to JPEG that allows adjusting this parameter more flexible [2]. The original approach utilizes saliency information that is gathered in user studies, either as self-reported data in crowd experiments or by collecting ground-truth with an eye tracker in the lab. Each image to be compressed is viewed by a number of participants, resulting in saliency maps with per pixel information. The quantization matrices for a given block are scaled according to its mean saliency value. As the study presented in [2] has shown, this bit allocation scheme can indeed be used to either reduce file size at an equal perceived quality level or improve perceived quality at an equal file size. The major disadvantage of this method is the manual generation of saliency maps.

As an application, we experiment with Patchnet feature maps as a replacement for user-generated saliency maps. The assumption is that high-quality regions in images correlate with salient regions. This would be the case e.g. in images where the salient motif, the region of interest, is properly focussed, while the less relevant background is blurred.

6.2.1 Proposed VarJPEG Algorithm

For a given input image, the proposed algorithm called VarJPEG, works as follows:

i) **Feature Map Generation**: We apply Patchnet in a sliding window fashion as depicted in Figure 9 with a stride of 1. The resulting quality map is zero-padded to match the input image's resolution.

ii) **Feature Map Pruning**: Fixation maps as used in [2] are binary matrices that indicate which pixels in an image were focussed by a participant in an eye-tracking experiment. While these are sparse due to the limited viewing time per image, our quality maps can have non-zero activations in large portions of the image. We prune the feature map by setting all activations in the 90 percentile to 0. The remaining coefficients with the highest ratings are set to 1. The result is a binary quality map with at most 10% of the coefficients indicating high quality.

iii) **Clustering**: As in the original approach, we perform a k-means clustering with $k = 8$. The reason to *not* use the Patchnet feature map directly is the memory requirement imposed by storing blockwise quality information. This significantly reduces the bit budget for actual image data. Clustering allows storing the relative coordinates of the centroids. In combination with the next step, this procedure serves as a low-overhead approximation of the original Patchnet quality maps.

iv) **Blurring**: We blur the clustered maps with a Gaussian kernel with a standard deviation σ equal to 10% of the image width. The values in the blurred feature

map are normalized to $[0, 1]$. Note that the resulting quality map M still has the same spatial resolution as the input image.

v) **Quantization**: For a given 8×8 pixel block b in the input image we calculate the block quality score q_b as

$$q_b = round(\widetilde{q} + \Delta\, mean(M|_b)) \tag{13}$$

where $\widetilde{q}$ is a global quality parameter, $mean(M|_b) \in [0, 1]$ is the average quality for block b as given by the corresponding values in M and $\Delta = 15$ is the difference between the highest and the lowest possible quality value.

An example of this procedure with intermediate feature maps is shown in Figure 14.

The DCT coefficients in each block are independently quantized according to q_b as in standard JPEG. While we experimented with different values for σ and Δ in [2], it remained an open question on how to choose these parameters optimally. The number of clusters in the k-means and in this approach also the feature map pruning procedure can arguably be enhanced, but these objectives are left for further research.

6.3 Compression Experiments

To investigate possible bitrate savings with our proposed compression algorithm, we rely on yet another measure, the Just Noticeable Difference (JND) [70]. We carried out an experiment on the same 125 images that were sampled from KonIQ-10k in Section 5.3.

For each source image $\mathcal{I}$ with bitrate $br_{\mathcal{I}}$, we used a binary search on the global quality parameter $\widetilde{q}$ to generate 100 compressed versions of $\mathcal{I}$ with equally distributed bitrates in $[br_{\mathcal{I}},\ 0.02 \cdot br_{\mathcal{I}}]$. Sufficient approximation of the desired bitrates is possible due to $\widetilde{q}$ being a floating point variable that is subsequently rounded after adding the scaled local quality score, as described in Formula 13.

We asked the participants of a crowd study to report their JND threshold. For this purpose, the source image was displayed as a reference side by side with a test image. In the initial state, the test image had approximately the same bitrate as the source image without any visible difference. Participants were asked to slowly degrade the test image using a slider until they start noticing distortions. Functionality implemented in JavaScript allowed us to replace the test image according to the slider movements. Using this setup, we collected 15 votes for each source image.

For comparison, we reran the experiment with test images compressed by standard JPEG. We compressed each source image with every possible quality level $q \in \{1, \ldots, 99\}$ and assessed the JND using the same crowdsourcing experiment as for our proposed variable compression algorithm.

6.3.1 Experimental Results: Savings at the JND Compression Level

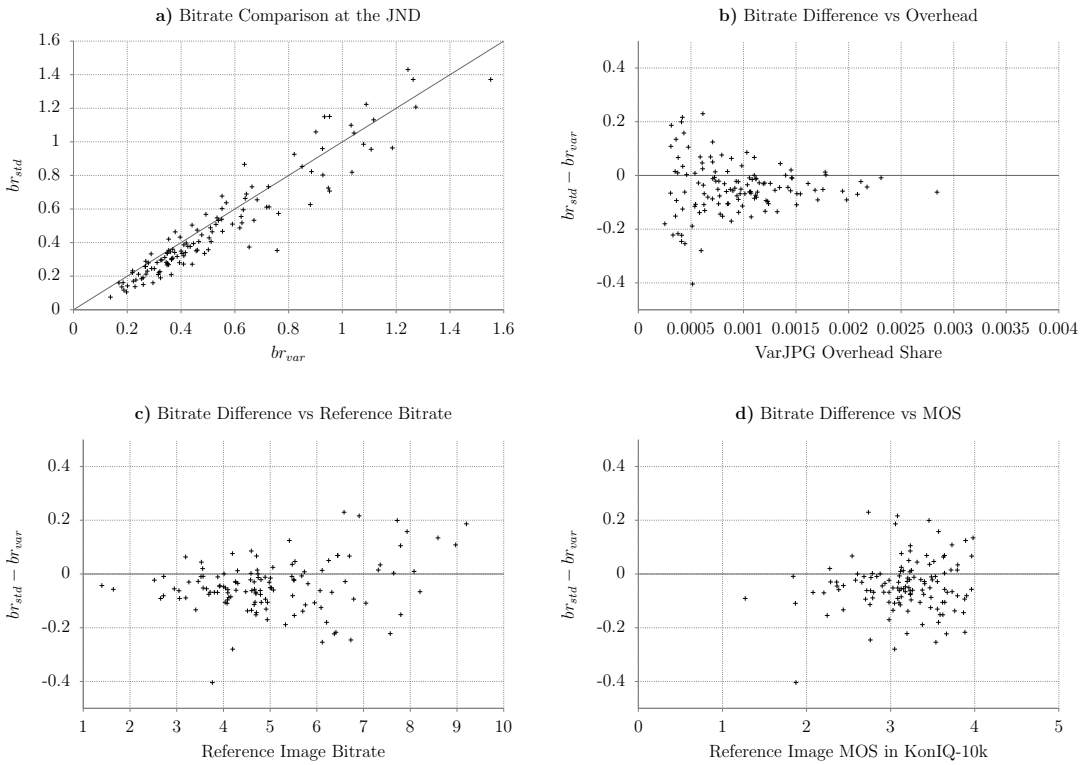


Figure 15: Performance comparison at the 0.25 percentile JND.

In Figure 15, we print the results of this study following the definition of Wang *et al.* [71], who defined the JND for a given image as the compression rate were more than 25% of the study participants noticed a difference to the reference image. The distributions in the diagrams may give some indication for which types of images VarJPEG may be superior to JPEG:

a) This diagram compares the bitrates of the VarJPEG compressed images with the JPEG compressed images who are just noticeably different from the pristine reference. Markers plotted over the diagonal line denote images that benefit from being compressed using our approach at JND compression level in terms of bitrate.

b) VarJPEG introduces a constant overhead to store the relative coordinates of the cluster centroids, depicted e.g. in Figure 14d. This diagram shows the bitrate difference between standard JPEG and VarJPEG at JND compression level depending on the proportion of the overhead relative to the VarJPEG file size.

c) This diagram shows the difference between standard JPEG bitrates and VarJPEG bitrates relative to the reference image's bitrate.

d) This diagram shows the difference between standard JPEG bitrates and VarJPEG bitrates relative to the reference image's mean opinion score in KonIQ-10k.

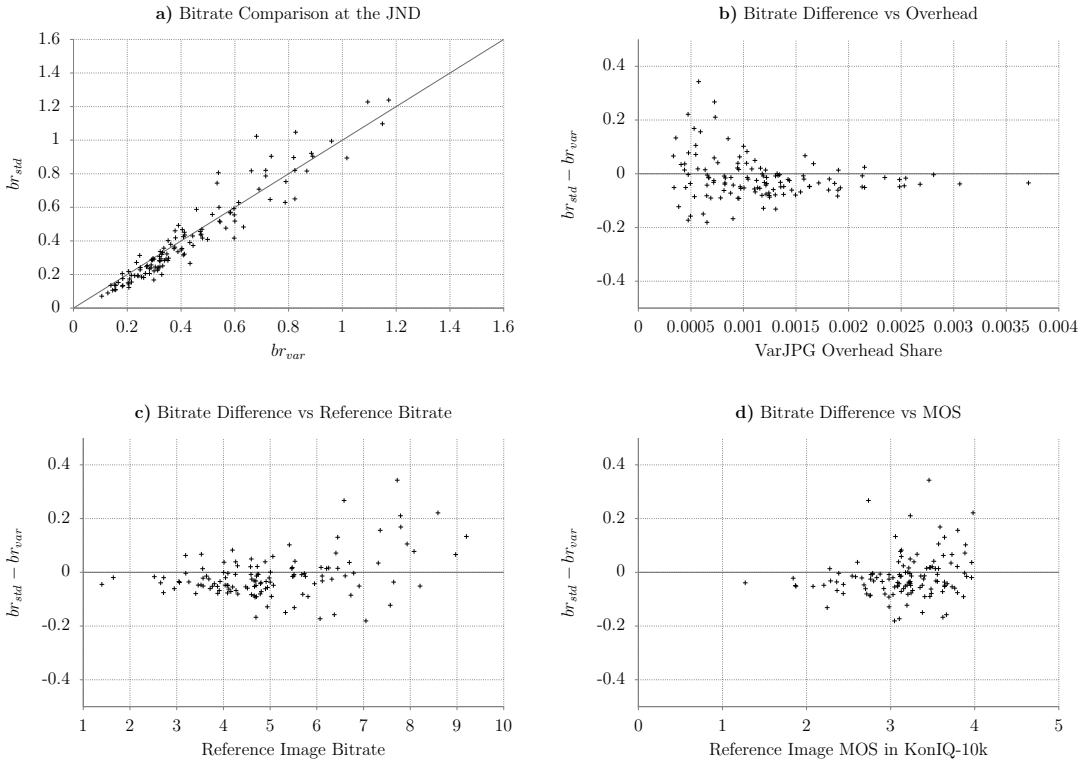


Figure 16: Performance comparison at the 0.5 percentile JND.

The diagrams in Figure 16 show the results when taking a different definition of the JND. Here, we compared the images that already 50% of the study participants reported as being different from the reference.

7 Evaluation

7.1 Patchnet and KonPatch

A performance evaluation of Patchnet from Figure 8 and Table 1 alone may lead to pessimism, as the MSE and MAE metrics are comparatively high given that the value range of the labels is $[0, 1]$. However, one has to consider the intrinsic difficulties of the dataset on which these numbers are reported. KonPatch was created making a compromise: fast subjective assessment of patches using a binary voting procedure with a single vote per image patch was traded for the introduction of a hard decision boundary. This is a drawback in our data model, as we assume that humans can distinguish more than two levels of image quality.

To re-introduce finer distinctions in the labels, we applied the scoring procedure described in Formula 3. A histogram of the resulting dataset is depicted in Figure 17a. It shows two clearly separated clusters with few training examples in between. Therefore, though the error metrics would be comparatively high for a classical regression task, this does not directly impose a problem in our case given that the distance between the two clusters is large. The resulting Patchnet feature maps, e.g. Figure 13b or Figure 14 may not be perfectly smooth, but they carry more nuances than a mere binary classification. From Figure 8a we can argue that Patchnet has sufficient *capacity* for the given training task, as all training instances converge towards zero over time. Figure 8b reveals the problem of overfitting after $30-40$ epochs. This behavior is also observed reliably across training instances. The early-stopped models which performed best on the validation set transferred to the unseen test set without a decrease in performance as shown in Table 1. We therefore have reason to believe that Patchnet will generalize well to unseen image data.

7.2 Local Quality Assessment on Entire Image

As quality prediction on isolated patches is not the primary use case for our model, the evaluation presented in Section 5.3 is a better measure of its utility. For this task, we gather statistics on the area under ROC curves for the 125 locally annotated quality maps from our auxiliary dataset. Each datapoint that is utilized in one of the box plots in Figure 11 presents the AUC of an image for variable predictor thresholds t_P at a fixed ground truth threshold t_G, which is plotted on the horizontal axis. On the one hand, Figure 11 shows cases where the predictions of our model are off: dots below 0.5 indicate an image where the false positive rate of the predictions is higher than the true positive rate at a specific threshold t_G. On the other hand, the model performs well besides these outliers: With mean AUC values around 0.9 and the 0.25 percentile around 0.8 throughout the whole t_G range, we can argue that the predictions of our are significantly better than random guesses for the majority of the images in our auxiliary dataset.

7.3 Global MOS Prediction

To further compare our model against existing image quality assessment algorithms from the literature, we resort to global MOS predictions due to the lack of an adequate local benchmark dataset. When comparing the correlation coefficients given in Table 2 and Table 3, one can see that an aggregation method as simple as taking the mean value of Patchnet feature maps already outperforms many traditional IQA methods. This promising result is further improved by the feature-aggregation approach presented in Section 5.4. Table 4 shows an interesting pattern: Our meta-model slightly outperforms KangCNN and BosICIP on KonIQ-10k, but is itself outperformed by DeepBIQ and DeepRN.

While it is possible that this constellation is purely by chance, the performance gap between these types of models may have a more systematic background: the lower performing examples, including our approach, work by aggregating a global score from local

predictions on spatially small image regions. The latter two models accept much larger image regions as an input: DeepBIQ was transfer learned from an ImageNet classification model and consequently accepts 224×224 pixel patches. DeepRN goes even further and is able to assess images independently of their resolution, above a certain minimum, by using spatial pyramid pooling [42] to reshape the outputs of the last convolutional layer to a vector of a fixed size. One possible explanation for the performance gain from using larger image areas for global MOS predictions is that human perception is indeed influenced by high level features. Our observations suggest that perceptual image quality is a mixture of local, technical quality e.g. in terms of sharpness and global, content dependent aspects related to aesthetical preferences and image composition.
Besides hypothesizing about explanations for this performance gap, it is a fact that our meta model performs close to the state of the art in global image quality prediction.

7.4 Variable JPEG Compression

The expectation on this application of our model was to achieve at least a slight improvement compared to standard JPEG compression. In terms of the utilized JND bitrate measure, this would mean that at least a subset of the 125 images in our dataset would have a lower bitrate when being variably compressed up to the point of just noticeable difference relative to the reference image.

As shown in Figure 15a, this especially not the case for images with very low bitrates at the JND compression rate. Since our proposed algorithm introduces a constant overhead to store the cluster centroids, this systematic inferiority may be due to JPEG having a slightly larger bit budget. Conversely, one could expect our algorithm to work better on images with higher bitrates at the JND, as the relative impact of the overhead would be lower. However, this also not the case, as no clear trend regarding the spread of data points is visible at higher values for br_{std} and br_{var}.

Figure 15b shows the bitrate difference between standard JPEG and our algorithm at the JND compression level on the vertical axis. The horizontal axis depicts the share of the overhead introduced by our algorithm with regard to the total file size the compressed image possesses at the JND. We would expect to see a clearer trend in favor of VarJPEG for images where the relative overhead is less impactful. One can at best speak of a very slight trend in this regard, but again a clear conclusion is not possible.

Figure 15c plots the bitrate difference versus the reference image's bitrate. As we already expected, VarJPEG requires slightly higher bitrates for references images with lower bitrates. In this plot, the range of very high reference image bitrates actually shows a slightly favoring picture for VarJPEG, but samples in this area are sparsely distributed. The last plot in Figure 15d compares bitrate differences with the reference image's mean opinion score. The circularly shaped distribution in this image is shifted slightly downwards in favor of standard JPEG, but doesn't indicate a clear relationship with MOS values. Wile the plots in Figure 16 seem to be slightly more in favour of our method, arguing based on this interpretation leads to being content with the judgments of less skeptical users, as the definition of the JND is shifted to the point where 50% of the participants reported a difference.

The observed underperformance of our implementation in comparison to standard JPEG was unexpected with regard to the previous success in [2]. One possible explanation is the JND compression level as a point of comparison or more specifically our approach to identify this point in our user studies: The Gaussian distributions in our local quality map deliberately enforce certain regions to be of higher quality than others. Ideally, we would like the users to be satisfied especially with regions of high interest.

VarJPEG enforces lower quality in image regions where the Patchnet feature map contains lower coefficients, leading to more artifacts in the background at the same average bitrate compared to standard JPEG. Even if participants reported JND levels honestly and accurately, the focus is set wrongly to the background. The interface used in the crowd study and a relatively high choice for the maximum quality difference likely contributed to this issue, as fast deferring of the slider visualizes early distortions in low-quality regions with a flickering effect. Overall, the plots show that there exist cases in which VarJPEG achieves better performance than standard JPEG, even in the chosen metric. Further optimization of the algorithm to achieve more systematic benefits however remains an open research topic.

8 Conclusion

This thesis covers the whole pipeline from task specific data generation throughout the construction of a current machine learning model to an elaborate performance analysis with regard to different aspects and usecases. The contributions include a novel, local image quality assessment database and a reference implementation to solve this task. We managed to create a model that is able to predict human judgements on the vaguely defined concept of local image quality.

With emerging technologies like virtual reality and ultra high resolution video the question of how to provide the best possible experience for a given bandwidth will become even more urgent. Though we are still far from understanding how exactly the perception of quality in multimedia works, approaches like the one presented in this thesis are promising models for the meanwhile and will likely receive more attention from the research community in the future.

Acknowledgment

I thank the German Research Foundation (DFG) for financial support within the project A05 of SFB/Transregio 161.

I thank my supervisors Dietmar Saupe and Christian Borgelt for always having a sympathetic ear and their advice on the issues I encountered while working on this thesis.

I thank my colleagues at the multimedia signal processing group for their support and the many fruitful discussions we had with a cup of tea.

I thank my parents for their unconditional support for all I do. I would not be where I am today without their help.

Last but not least, I thank my girlfriend Désirée for her patience, which I often stretched during the last few weeks.

9 Supplementary Material

9.1 KonPatch: Quality Score Histograms

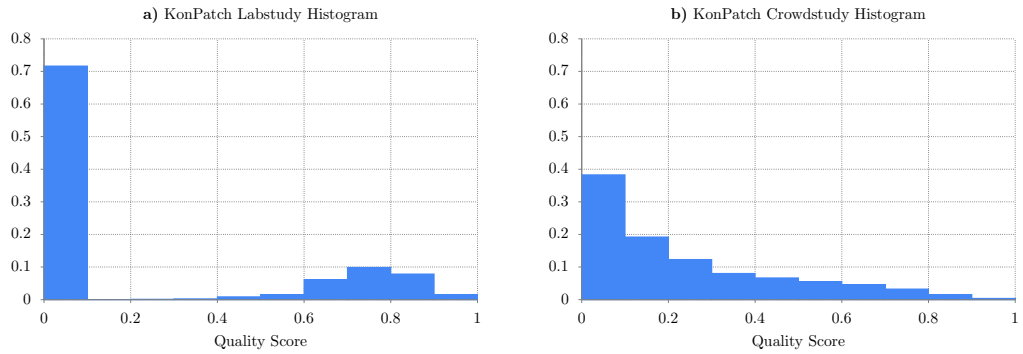


Figure 17: KonPatch quality score histograms.

The crowd study we conducted on KonPatch was designed as follows: We asked 10 participants per patch for a binary classification whether the presented image patch looks indicative of being sampled from a high-quality image. The histogram shown in Figure 17b depicts the distribution of quality scores that are generated by taking the fraction of positive answers relative to the total number of answers per patch.
This procedure creates a totally different distribution of scores compared to the lab results, where negative answers are mapped to a score of zero and positive answers to the mean opinion score of the source image that the patch was sampled from, as described in Formula 3.

9.2 KonPatch: Examples

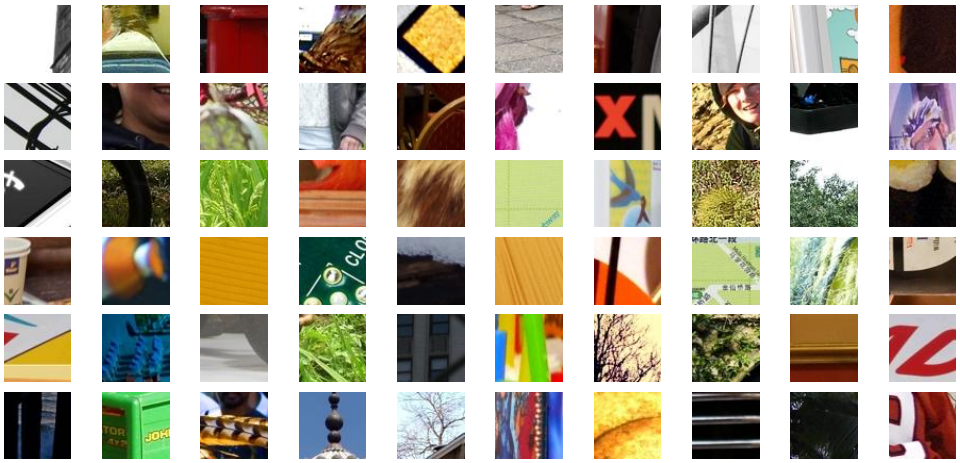


Figure 18: Patches marked as indicative of being sampled from a high quality image.

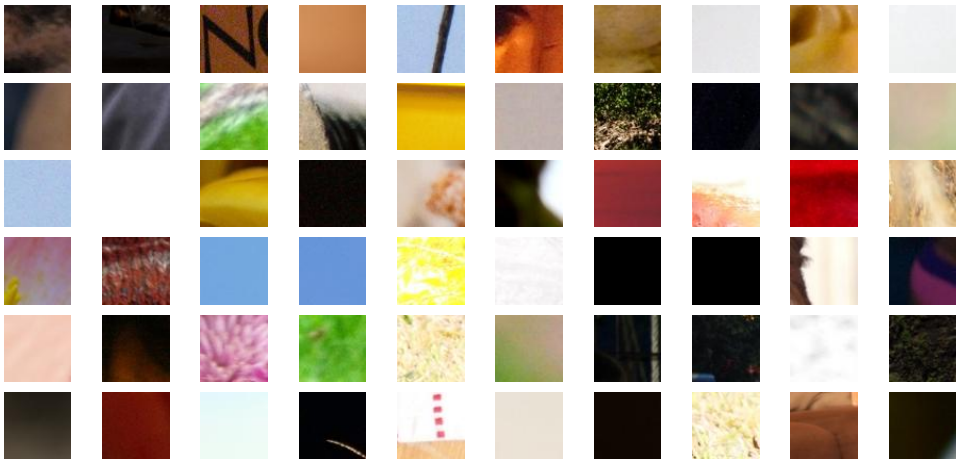


Figure 19: Patches marked as *not* indicative of being sampled from a high quality image.

References

[1] Oliver Wiedemann, Vlad Hosu, Hanhe Lin, and Dietmar Saupe. Disregarding the big picture: Towards local image quality assessment. In *10th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2018.

[2] Vlad Hosu, Franz Hahn, Oliver Wiedemann, Sung-Hwan Jung, and Dietmar Saupe. Saliency-driven image coding improves overall perceived JPEG quality. In *Picture Coding Symposium (PCS), 2016*, pages 1–5. IEEE, 2016.

[3] Hamid R Sheikh. Live image quality assessment database. *http://live.ece.utexas.edu/research/quality*, 2003.

[4] Hamid R. Sheikh. Live image quality assessment database release 2. *http://live.ece. utexas. edu/research/quality*, 2005.

[5] Hamid R. Sheikh, Muhammad F Sabir, and Alan C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. In *IEEE Transactions on Image Processing*, volume 15. IEEE, 2006.

[6] Patrick Le Callet and Florent Autrusseau. Subjective quality assessment IRCCyN/IVC database. 2005.

[7] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti. TID2008 - A database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radioelectronics 10: 30–45, 2009.*

[8] Nikolay Ponomarenko, Federica Battisti, Karen Egiazarian, Jaakko Astola, and Vladimir Lukin. Metrics performance comparison for color image database. *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2009.

[9] Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, et al. Image database TID2013: Peculiarities, results and perspectives. In *Signal Processing: Image Communication*, volume 30, pages 57–77. Elsevier, 2015.

[10] Eric Cooper Larson and Damon Michael Chandler. Categorical image quality (CSIQ) database, 2010.

[11] Eric Cooper Larson and Damon Michael Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. In *Journal of Electronic Imaging*, volume 19. International Society for Optics and Photonics, 2010.

[12] Domonkos Varga, Tamas Szirányi, and Dietmar Saupe. DeepRN: A content preserving deep architecture for blind image quality assessment. In *International Conference on Multimedia and Expo (ICME)*. IEEE, 2018.

[13] Dinesh Jayaraman, Anish Mittal, Anush K Moorthy, and Alan C. Bovik. Objective quality assessment of multiply distorted images. In *Conference Record of the 46th Asilomar Conference on Signals, Systems and Computers*, pages 1693–1697. IEEE, 2012.

[14] Toni Virtanen, Mikko Nuutinen, Mikko Vaahteranoksa, Pirkko Oittinen, and Jukka Häkkinen. CID2013: a database for evaluating no-reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 24(1):390–402, 2015.

[15] D. Ghadiyaram and A. C. Bovik. Live in the wild image quality challenge database. *Online: http://live.ece.utexas.edu/research/ChallengeDB/index.html*, 2015.

[16] Hanhe Lin, Vlad Hosu, and Dietmar Saupe. KonIQ-10K: Towards an ecologically valid and large-scale IQA database. *arXiv preprint arXiv:1803.08489*, 2018.

[17] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo Exploration Database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, Feb. 2017.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[20] Cuong T. Vu, Thien D. Phan, and Damon M. Chandler. S3: A spectral and spatial measure of local perceived sharpness in natural images. *IEEE Transactions on Image Processing*, 21(3):934–945, 2012.

[21] Thomas Young. The Bakerian Lecture. On the theory of light and colours. *Philosophical Transactions of the Royal Society of London*, 92:12–48, 1802.

[22] David Salomon. *Data Compression: The Complete Reference*. Springer Science & Business Media, 2006.

[23] ITU-R Recommendation BT.601 - Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, International Telecommunication Union. 2011.

[24] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008.

[25] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.

[26] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. In *IEEE Transactions on Image Processing*, volume 13, pages 600–612. IEEE, 2004.

[27] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. FSIM: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.

[28] Linda Henriksson, Aapo Hyvärinen, and Simo Vanni. Representation of cross-frequency spatial phase relationships in human visual cortex. *Journal of Neuroscience*, 29(45), 2009.

[29] Anuj Srivastava, Ann B. Lee, Eero P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33, 2003.

[30] Anush Krishna Moorthy and Alan C. Bovik. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17(5):513–516, 2010.

[31] Anush Krishna Moorthy and Alan C. Bovik. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Transactions on Image Processing*, 20(12):3350–3364, 2011.

[32] Michele A Saad, Alan C Bovik, and Christophe Charrier. Blind image quality assessment: A natural scene statistics approach in the dct domain. *IEEE Transactions on Image Processing*, 21(8):3339–3352, 2012.

[33] Anish Mittal, Anush Krishna Moorthy, and Alan C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

[34] Lixiong Liu, Bao Liu, Hua Huang, and Alan C. Bovik. No-reference image quality assessment based on spatial and spectral entropies. *Signal Processing: Image Communication*, 29(8):856–863, 2014.

[35] Yuming Fang, Kede Ma, Zhou Wang, Weisi Lin, Zhijun Fang, and Guangtao Zhai. No-reference quality assessment of contrast-distorted images based on natural scene statistics. *IEEE Signal Processing Letters*, 22(7):838–842, 2015.

[36] Andrej Karpathy. What I learned from competing against a convnet on imagenet. *Blog post, online: http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet*, 2014.

[37] Le Kang, Peng Ye, Yi Li, and David Doermann. Convolutional neural networks for no-reference image quality assessment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1733–1740, 2014.

[38] Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek. A deep neural network for image quality assessment. In *International Conference on Image Processing (ICIP)*, pages 3773–3777. IEEE, 2016.

[39] Simone Bianco, Luigi Celona, Paolo Napoletano, and Raimondo Schettini. On the use of deep learning for blind image quality assessment. *arXiv preprint arXiv:1602.05531*, 2016.

[40] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36, 2012.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Computing Research Repository*, arXiv: abs/1512.03385, 2015.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[43] ITU-T Recommendation P.910 - Subjective video quality assessment methods for multimedia applications, International Telecommunication Union. 2008.

[44] Zhou Wang, Hamid R Sheikh, and Alan C. Bovik. No-reference perceptual quality assessment of jpeg compressed images. In *2002 International Conference on Image Processing*. IEEE, 2002.

[45] Gregory K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1), 1992.

[46] Zhou Wang and Alan C. Bovik. Modern image quality assessment. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 2(1):1–156, 2006.

[47] Vlad Hosu, Hanhe Lin, and Dietmar Saupe. Expertise screening in crowdsourcing image quality. In *10th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2018.

[48] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[49] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, MIT Press Cambridge, 1995.

[50] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press Cambridge, 2016.

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[52] Augustin Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[53] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[54] Léon Bottou. Online algorithms and stochastic approximations. In David Saad, editor, *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998. revised, oct 2012.

[55] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[56] Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

[57] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[58] Rudolf Kruse, Christian Borgelt, Christian Braune, Sanaz Mostaghim, and Matthias Steinbrecher. *Computational Intelligence: A Methodological Introduction*. Springer, 2016.

[59] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *27th International Conference on Machine Learning*, pages 807–814, 2010.

[60] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[61] François Chollet et al. Keras, Online: http://keras.io. 2015.

[62] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[63] Tom Fawcett. ROC graphs: Notes and practical considerations for researchers. *Machine Learning*, 31(1):1–38, 2004.

[64] Phong V. Vu and Damon M. Chandler. A fast wavelet-based algorithm for global and local image sharpness estimation. *IEEE Signal Processing Letters*, 19(7):423–426, 2012.

[65] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[66] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *Computing Research Repository*, arXiv: abs/1608.06993, 2016.

[67] ITU-T Recommendation T.871 - Information technology - Digital compression and coding for continuous-tone still images: JPEG File Interchange Format (JFIF), International Telecommunication Union. 2011.

[68] ITU-T Recommendation T.81 - Information technology - Digital compression and coding for continuous-tone still images: requirements and guidelines, International Telecommunication Union. 1992.

[69] L. Berc, W. Fenner, R. Frederick, S. McCanne, and P. Stewart. RTP payload format for JPEG-compressed video. RFC 2435, RFC Editor, October 1998.

[70] Joe Yuchieh Lin, Lina Jin, Sudeng Hu, Ioannis Katsavounidis, Zhi Li, Anne Aaron, and C-C Jay Kuo. Experimental design and analysis of JND test on coded image/video. In *Applications of Digital Image Processing XXXVIII*. International Society for Optics and Photonics, 2015.

[71] Haiqiang Wang, Ioannis Katsavounidis, Jiantong Zhou, Jeonghoon Park, Shawmin Lei, Xin Zhou, Man-On Pun, Xin Jin, Ronggang Wang, Xu Wang, Yun Zhang, Jiwu Huang, Sam Kwong, C. Kuo, Yun Zhang, Jiwu Huang, Sam Kwong, and C.-C. Jay Kuo. Videoset: A large-scale compressed video quality dataset based on jnd measurement. *Journal of Visual Communication and Image Representation*, 46:292–302, 2017.

[72] David E. Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. Backpropagation: The basic theory. *Backpropagation: Theory, Architectures and Applications*, pages 1–34, 1995.

[73] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *24th International Conference on Machine learning*, pages 759–766. ACM, 2007.

[74] Deepti Ghadiyaram and Alan C Bovik. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Transactions on Image Processing*, 25(1):372–387, 2016.

[75] Martin Meyer. *Signalverarbeitung: Analoge und digitale Signale, Systeme und Filter*. Springer-Verlag, 2006.