UNIVERSITY OF CAMBRIDGE

Visibility metrics and their applications in visually lossless image compression

Nanyang Ye

Gonville&Caius College

This dissertation is submitted on August, 2019 for the degree of Doctor of Philosophy

Abstract

Visibility metrics and their applications in visually lossless image compression

Nanyang Ye

Visibility metrics are image metrics that predict the probability that a human observer can detect differences between a pair of images. These metrics can provide localized information in the form of visibility maps, in which each value represents a probability of detection. An important application of the visibility metric is visually lossless image compression that aims at compressing a given image to the lowest fraction of bit per pixel while keeping the compression artifacts invisible at the same time.

In previous works, most visibility metrics were modeled based on largely simplified assumptions and mathematical models of human visual systems. This approach generally fits well into experimental data measured with simple stimuli, such as Gabor patches. However, it cannot predict complex non-linear effects, such as contrast masking in natural images, particularly well. To predict visibility of image differences accurately, we collected the largest visibility dataset under fixed viewing conditions for calibrating existing visibility metrics and proposed a deep neural network-based visibility metric. We demonstrated in our experiments that the deep neural network-based visibility metric significantly outperformed existing visibility metrics.

However, the deep neural network-based visibility metric cannot predict visibility under varying viewing conditions, such as display brightness and viewing distances that have great impacts on the visibility of distortions. To extend the deep neural network-based visibility metric to varying viewing conditions, we collected the largest visibility dataset under varying display brightness and viewing distances. We proposed incorporating white-box modules, in other words, luminance masking and viewing distance adaptation, into the black-box deep neural network, and we found that the combination of white-box modules and black-box deep neural networks could generalize our proposed visibility metric to varying viewing conditions.

To demonstrate the application of our proposed deep neural network-based visibility metric to visually lossless image compression, we collected the visually lossless image compression dataset under fixed viewing conditions and significantly improved the deep neural

network-based visibility metric's accuracy of predicting visually lossless image compression threshold by pre-training the visibility metric with a synthetic dataset generated by the state-of-the-art white-box visibility metric—HDR-VDP [1]. In a large-scale study of 1000 images, we found that with our improved visibility metric, we can save around 60% to 70% bits for visually lossless image compression encoding as compared to the default visually lossless quality level of 90.

Because predicting image visibility and predicting image quality are closely related research topics, we also proposed a trained perceptually uniform transform for high dynamic range images and videos quality assessments by training a perceptual encoding function on a set of subjective quality assessment datasets. We have shown that when combining the trained perceptual encoding function with standard dynamic range image quality metrics, such as peak-signal-noise-ratio (PSNR), better performance was achieved compared to the untrained version.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface or specified in the text. This dissertation does not exceed the prescribed limit of 60 000 words.

Nanyang Ye
August, 2019

Acknowledgements

After spending these truly wonderful years, I truly believe that I have become a better person in my life. This would not have been even possible without the huge support of the most amazing and generous people I have ever met.

My PhD life would not be so smoothly without the guidance of my supervisor—Dr. Rafał K. Mantiuk. He leads me through this journey with full support. Without his generous help, I cannot see through the fog and reach our final goal. Limited vocabulary cannot express my gratitude for him.

Besides Rafał, over the course of PhD, I was also very lucky to be lead by Professor. Peter Robinson at the early stage, who raise me to a higher altitude to see PhD life.

Over these years, I was also lucky to have wonderful people around who have insights and makes our lives more interesting—Minjung Kim, Maryam Azimi, Gyorgy Denes, Maria P. Ortiz, Aliaksei Mikhailiuk, Fangcheng Zhong, Param Hanji, Akshay Jindal and Dingcheng Yue ...

Last but not least, I want to thank my parents, who brought me to this wonderful world and inspire me with their unconditional love.

Contents

List of Abbreviations

CNN Convolutional Neural Network.

CRT Cathode-Ray Tube.

CSF Contrast Sensitivity Function.

DCT Discrete Cosine Transform.

FR-IQM Full-Reference Image Quality Metric.

HDR-VDP High Dynamic Range Visible Differences Predictor 2.

IQM Image Quality Metric.

IVM Image Visibility Metric.

JFIF JPEG file interchange format.

JPEG Joint Photographic Expert Group.

LCD Liquid Crystal Display.

LocVis Local Visibility maps of artifacts and distortions in images dataset.

LocVisVC LocVis with varying Viewing Conditions.

MOS Mean Opinion Score.

ppd Unit:pixels per degree.

PSNR Peak Signal-to-Noise Ratio.

SSIM Structural Similarity Index Metric.

VDP Visible Differences Predictor.

VLIC Visually Lossless Image Compression dataset.

VLT Visually Lossless compression Threshold.

List of Tables

List of Figures

CHAPTER 1

Introduction

The visibility metric is an image metric measuring whether introduced changes in images are visible or not. It plays a key role in visually lossless image compression because it predicts whether compression distortions in images are visible or not.

1.1 Motivation

Measuring visible image differences accurately is essential for many image processing applications, such as image compression, 3D rendering in computer games, and invisible watermarking. For example, in image compression, inventing more efficient lossless image encoders would require a tremendous amount of effort. However, with a visibility metric measuring whether compression artifacts are visible or not, we can easily achieve a much lower bit rate without the need for changing the current image communication standards by setting a suitable parameter for the compression encoder to produce visually lossless distortions. In this thesis, we will first introduce our preparation work (dataset collection), and then propose a visibility metric based on machine learning, and finally suggest how to improve the visibility metric for visually lossless image compression. The findings indicate a new direction to compress images much more efficiently.

1.2 Publication and presentation list

This section includes publications, presentations, and demonstrations.

Publications:

1. **Langevin Dynamics with Continuous Tempering for Training Deep Neural Networks** *Nanyang Ye*, Zhanxing Zhu, Rafał K. Mantiuk. Published at **Conference on Neural Information Processing Systems 2017 (NIPS)**.

Paper contribution: To tackle the challenge of minimizing non-convex and high-dimensional objective functions, I proposed a learning process that uses Bayesian sampling for better generalization performance. In the Bayesian sampling phase, I proposed a novel method in which temperature was adjusted automatically with continuously tempered Langevin dynamics (CTLD). In the optimization phase, I employed stochastic gradient optimization for fine-tuning of the neural network. The CTLD has been proven to converge to the true posterior distribution and has bounded estimation bias that vanishes with the increasing of the number of steps.

2. **Dataset and Metrics for Predicting Local Visible Differences** Krzysztof Wolski*, Daniele Giunchi*, *Nanyang Ye**, Piotr Didyk, Karol Myszkowski, Radosaw Mantiuk, Hans-Peter Seidel, Anthony Steed, and Rafał K. Mantiuk. Published at **ACM Transaction on Graphics (TOG)**. (* indicates equal contribution)

   Paper contribution: Proposed a deep learning-based visibility metric that achieved state-of-the-art performance on our visibility dataset. Tested the deep learning-based visibility metric on a visually lossless compression dataset and achieved state-of-the-art results. The deconvolutional neural network architecture was proposed instead of the fully-connected network architecture to avoid over-fitting, an approach which made the visibility metric generalized more effectively.

3. **Predicting visible image differences under varying display brightness and viewing distance**. *Nanyang Ye*, Kysztof Wolski and Rafał K.Mantiuk. Published at **IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019 (CVPR).**

   Paper contribution: I extended the work of Wolski *et al.* so that the proposed visibility metric could take account of a range of display brightness levels and angular resolutions. I achieved this by combining white-box models of luminance masking and spatial resampling with a black-box CNN-based model, based on the architecture from Wolski *et al.* .

4. **Trained Perceptual Transform for Quality Assessment of High Dynamic Range Images and Video** *Nanyang Ye*, María Pérez-Ortiz and Rafał K.Mantiuk. Published at **IEEE International Conference on Image Processing 2018 (ICIP)**.

   Paper contribution: Proposed a trained perceptually transform for quality assessment of high dynamic range (HDR) images and video. The transform was used to convert absolute luminance values found in HDR images into perceptually uniform units, which could be used with any standard-dynamic-range metric. The new transform was derived by fitting the parameters of a previously proposed perceptual encoding

function to 4 different HDR subjective quality assessment datasets using Bayesian optimization. The new transform combined with a simple peak signal-to-noise ratio measure achieved better quality prediction performance in the cross-dataset validation than existing transforms.

5. **Visibility Metric for Visually Lossless Image Compression**. *Nanyang Ye*, María Pérez-Ortiz and Rafał K.Mantiuk. Published at **Picture Coding Symposium 2019 (PCS).**

   Paper contribution: I collected a visually lossless compression dataset consisting of 50 JPEG- or WebP- compressed images. I analyzed the training process of the deep learning-based visibility metric and improved the generalization performance significantly. I proposed a visually lossless compression method based on improved visibility metrics. The improved visibility metrics have achieved state-of-the-art results by a large margin on our dataset.

Presentations:

1. **Langevin Dynamics with Continuous Tempering for Training Deep Neural Networks** Nanyang Ye presented at Long Beach, California, USA 2017.

2. **Trained Perceptual Transform for Quality Assessment of High Dynamic Range Images and Video** Nanyang Ye presented at Athens, Greece 2018.

3. **Visibility Metrics and Their Application in Visually Lossless Compression** Second-year research presentation in the Department of Computer Science and Technology.

4. **Reading Group: Deep Learning of Human Visual Sensitivity in Image Quality Assessment** Nanyang Ye presented at the machine learning and imaging reading group, Cambridge, UK 2018.

5. **Seminar: Visually Lossless Compression** Nanyang Ye presented at Cambridge-Oxford Seminar on Image Processing, Cambridge, UK 2018.

6. **Predicting visible image differences under varying display brightness and viewing distance** Nanyang Ye presented at IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, California, USA 2019.

7. **Invited talk: Visibility metric and visually lossless image compression** Nanyang Ye presented at Hamlyn Symposium, Imperial College, London, UK 2019.

Demonstrations:

1. Visually lossless image compression demonstration for **ARM**, Cambridge, UK, 2018

2. Visually lossless image compression demonstration for **Display Link**, Cambridge, UK, 2018

3. Visually lossless image compression demonstration for **HUAWEI**, Cambridge, UK, 2018

CHAPTER 2

Background

In this chapter, we will introduce relevant concepts in visibility metrics, visually lossless image compression, and machine learning.

2.1 Image quality metrics vs. Image visibility metrics

Image metrics evaluate the effects of distortions in image processing and can be divided into image quality metrics (IQMs) and image visibility metrics (IVMs), both addressing different applications. As shown in Figure 2.1, IQMs predict a single global quality score for the entire image. IQMs are usually trained and evaluated on mean opinion scores (MOSs) that are obtained in user experiments for each distorted image. In contrast, IVMs predict the probability that a human observer will detect differences between a pair of images. They provide localized information in the form of a visibility map or a probability of detection map, in which each pixel represents the probability of detecting the difference between a pair of images at the pixel's location. IVMs tend to be more accurate for

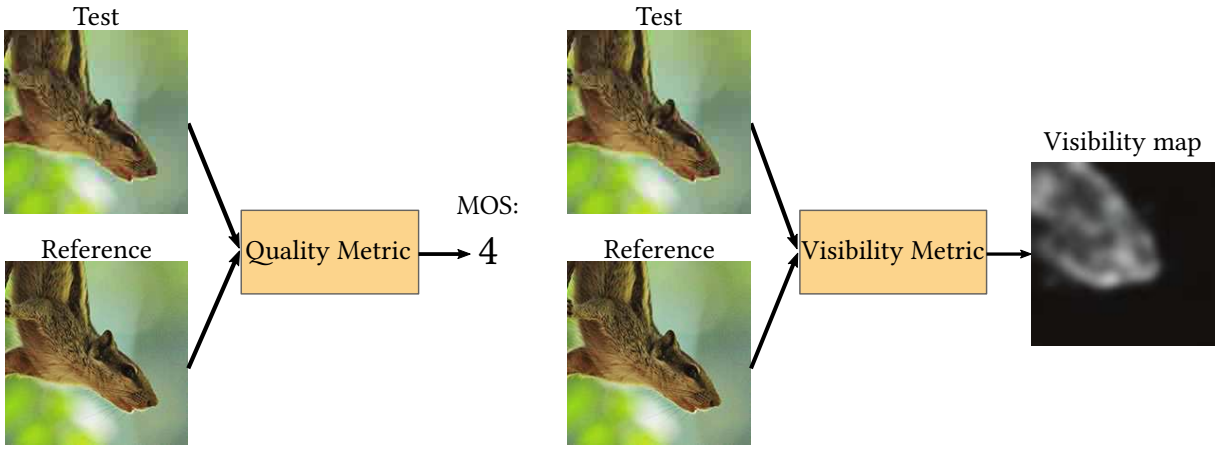


Figure 2.1: Comparison between image quality metrics and image visibility metrics. Images with high MOS scores may still have visible distortions.

Metrics	Superathreshold	Near-threshold	Interpretability	Output
IQMs	✓	✗	✗	MOS
IVMs	✗	✓	✓	Visibility map

Table 2.1: Comparison between IQMs and IVMs.

measuring small and barely noticeable distortions but are unable to assess the severity of distortion. IVMs are often more relevant for graphics applications. The goal of IVMs is to maximize performance without introducing any visible artifacts. Besides, different from IVMs, IQMs rarely consider display brightness and viewing distance, both of which can significantly impact the visibility of barely noticeable distortions, which will be discussed in Section 3.2.

The most related type of IQM to this thesis is the *full-reference* IQM (FR-IQM) that takes a pair of images as the input. Simple statistics-based FR-IQMs include the Peak Signal-to-Noise Ratio (PSNR). For more complex quality metrics, the Structural Similarity Index Metric (SSIM, [3]) considers the spatial variation information. The Visual Saliency-Induced Index (VSI, [4]) and the Feature Similarity Index (FSIM, [5]) have a similar framework as the SSIM but employ other information, such as saliency maps. However, these IQMs do not take account of different absolute luminance levels or viewing distance. High Dynamic Range Video Quality Measure (HDR-VQM, [6]) addresses the change of physical luminance in the images and videos. HDR-VQM employs the perceptual uniform transformation [7] to convert the physical luminance to the perceptual uniform values and use log-Gabor filters [8] to compute the subband differences in a pair of images. Because FR-IQMs uses the magnitude of image distortion as a single mean opinion score value, FR-IQMs are often more accurate in terms of strong distortions than IVMs. On the other hand, IVMs are trained for near-threshold distortions to predict the probability of detecting the difference. This property makes IVMs more suitable for applications where the accurate detection of near-threshold distortions is important, such as visually lossless image compression. We will show this with experiments in Section 6.2. We recommend that readers refer to more complete surveys on quality metrics [9, 10] further.

Another difference between IQMs and IVMs is that IVMs are generally easier to interpret than IQMs. These MOS scores measure subjective opinions. The MOS scores are sometimes related to the aesthetics of images that are difficult to interpret, whereas the visibility of image differences are more related to the perception ability of human beings that can be interpreted as the human ability to perceive differences directly. A summary of differences between IQMs and IVMs can be found in Table 2.1. In the following sections, we will focus on IVMs.

2.2 The fundamentals of visibility metrics

In this section, we will introduce the fundamentals of IVMs research. These metrics are aimed at modeling human visual systems. We will focus on the computational modeling of IVMs in the following section. For clarity, we summarize the main parts of the human visual system involved in detecting image differences in Figure 2.2 [11]. Whereas this figure provides a comprehensive description of the detection process, most IVMs only use parts of it. However, actual models may cover only part of the elements. It is also worth noting that the computational modeling of the human visual system does not necessarily reflect the anatomic structure of the human visual system but instead focuses on approximately predicting visibility maps. We will introduce the main elements in the following sections. In this thesis, only the display model and the luminance masking module are used in our proposed IVMs, and the optics of the eye element are particularly considered in the state-of-the-art IVM [1] and are explained in Section 2.3.2 later.

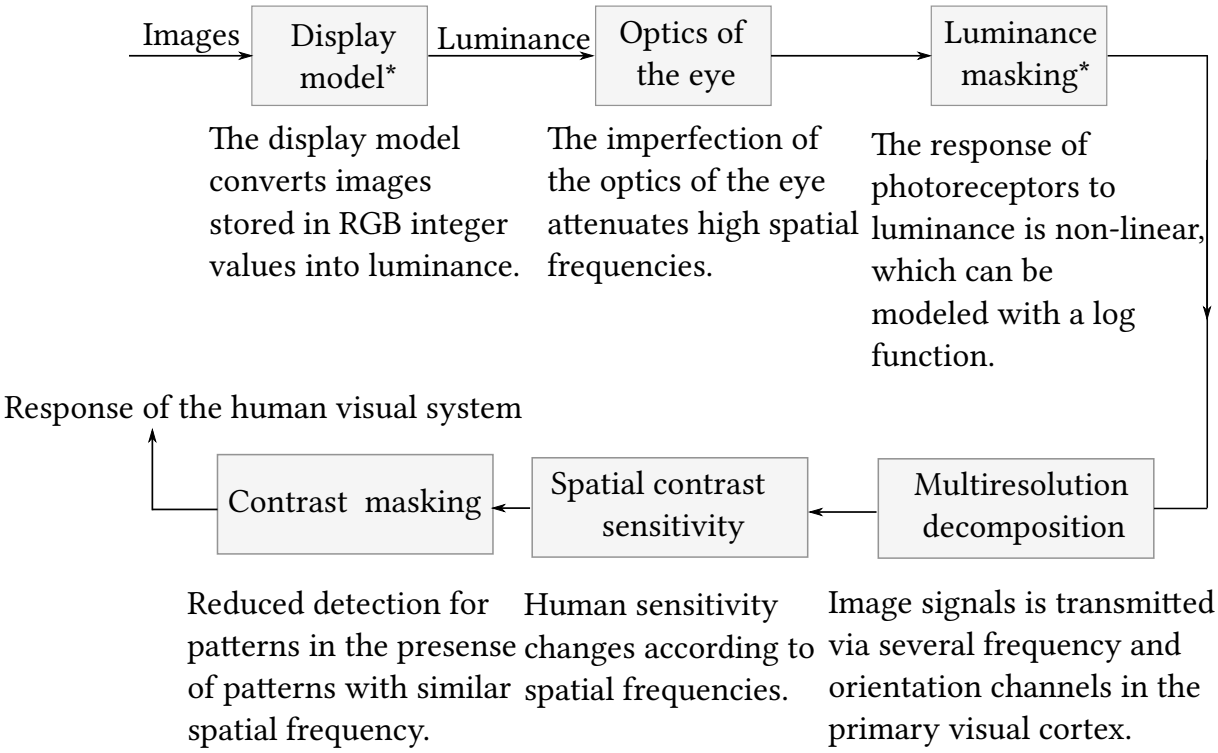


Figure 2.2: Computational model of the human visual system. Components with $*$ are used in this thesis.

2.2.1 Display model

Compared to high dynamic range images, standard dynamic range images are still significantly more popular. Different from high dynamic range images' formats that can represent the physical luminance of images, standard dynamic range images are usually

represented in the standard RGB (sRGB) color space where each pixel is represented with an RGB triplet integer ranging from [0-255]. However, the visibility of differences between a pair of images is dependent on the display brightness. Thus, it is necessary to convert the RGB triplet integers to physical luminance values for IVMs (Figure 2.3). The display models enable IVMs to consider the effects of different display brightness, such as a desktop display put in an office or a cell phone display at night. This procedure is highly different from IQMs that take RGB values directly as the input.

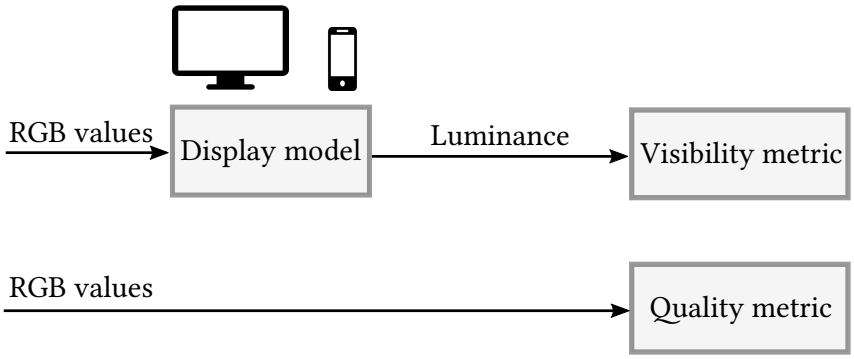


Figure 2.3: Role of display models in IVMs.

Cathode-ray tube (CRT) displays were widely used before the 2000s, and CRT displays have a non-linear relationship between the driving voltage and intensity. The transformation from RGB triplets to output light intensities is termed as the "gamma correction". Due to this historical reason, modern liquid crystal displays (LCDs) also adopt this procedure. We use the sRGB transfer function to transform the RGB triplets to physical luminance:

$$l = (l_{\text{peak}} - l_{\text{black}})(v/255)^{2.2} + l_{\text{black}} \tag{2.1}$$

where l is the emitted luminance of the image, l_{peak} is the peak luminance of the display, l_{black} is the luminance of the black color, and v is the RGB triplet value.

2.2.2 Angular resolution

The viewing distance can have a large impact on the visibility of image differences. However, different displays have varying resolutions, thus viewing distance alone is not sufficient. We typically use the angular resolution instead of the viewing distance to measure the effect of viewing distance and display resolution. The angular resolution used in visibility research is defined as the number of pixels per degree of visual angle (Figure 2.4).

The angular resolution of an image on the display can be computed as:

$$\rho = \frac{N_y}{h_{\text{deg}}} \quad [ppd] \ , \tag{2.2}$$

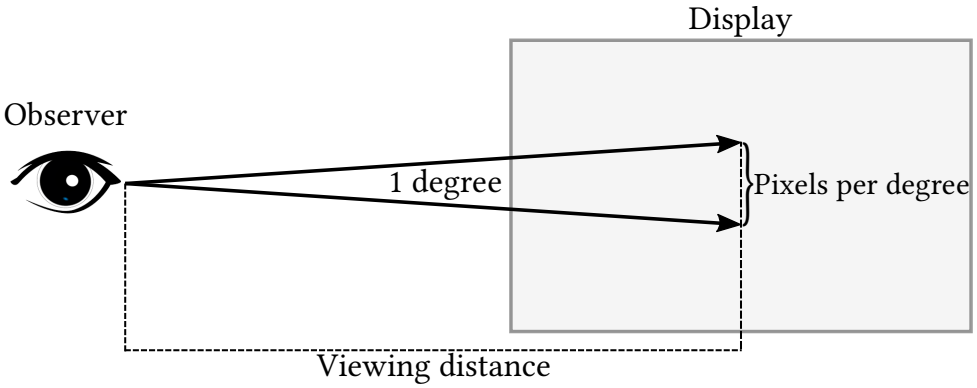


Figure 2.4: Angular resolution of images on a display.

where N_y is the vertical display resolution expressed in pixels, and h_{deg} is the display height in degrees of visual angle. h_{deg} is given by:

$$h_{\text{deg}} = 2 \arctan\left(\frac{h}{2d}\right) , \tag{2.3}$$

where h is the display height, and d is the viewing distance. The display height can be found from:

$$h = \sqrt{\frac{(s_{\text{diag}})^2}{1 + \left(\frac{N_x}{N_y}\right)^2}} , \tag{2.4}$$

where s_{diag} is the display diagonal length. The unit of the angular resolution is pixels per degree (ppd). The angular resolution is also limited by the Nyquist frequency which is two times the maximum spatial frequency the display can show to the observer. For example, for an image viewed at 60 ppd, the maximum spatial frequency the image can reach is 30 cycles per degree. The angular resolution provides a principled way to consider the effects of viewing distance and the display resolution on the visibility of image differences.

2.2.3 Luminance masking

Luminance is a photometric measurement of the luminous intensity per unit area. The relationship between the perceived intensity of light and the intensity of light is non-linear. The eye is more sensitive to the relative luminance (the ratio of stimulus luminance and the background luminance) than the absolute luminance [11, 12], and the sensitivity decreases under high luminance conditions. This effect is referred to as "luminance masking". A simple luminance masking model is based on the Weber-Fechner law in psychophysics [13]:

$$\mathrm{d}P = \frac{1}{L}\mathrm{d}L \tag{2.5}$$

where P is the perceived luminance, and L is the luminance. Integrating Equation 2.5 results in a log luminance masking function:

$$P(L) = \log(L) \tag{2.6}$$

Note that this is only a simple approximation that is inaccurate under low-luminance conditions. Particularly, the luminance masking can also be modeled as an S-shaped function, which is used in the visual difference predictor [14]:

$$P(L) = \frac{L}{L + c_1 L^b} \tag{2.7}$$

where c_1 is 12.6 and $b = 0.63$. We will compare different models of luminance masking in Section 5.2.3 and attempt to combine luminance masking models with deep neural networks to predict visibility maps under different peak display brightnesses. We will also train a data-driven perceptual luminance masking transform for high dynamic range image quality assessment in Chapter 7. The luminance masking model describes the sensitivity of the human eye to different luminance levels. However, luminance alone provides an incomplete description of visibility because the spatial frequency also has a large impact on visibility.

2.2.4 Contrast sensitivity function

Whereas visual acuity measures the ability to identify objects, such as letters from a certain distance, contrast sensitivity is defined as the inverse of the quantity of minimum contrast required for people to detect a target image against a homogeneous background [15] [1]. The function that models contrast sensitivity is the contrast sensitivity function (CSF). There are many different types of CSFs, each one modeling a different aspect of the human visual system. CSF can be further divided into the spatial CSF and the temporal CSF. We focus on the spatial CSF, which characterizes the contrast sensitivity in the spatial domain because it is more relevant to images. CSF is often measured with sinusoidal test patterns because such patterns can be analyzed easily with Fourier transforms. We plot an example of spatial CSF and sine-wave grating patterns in Figure 2.5. The red line is a CSF that characterizes the threshold of detecting the sine-wave gratings against a uniform background. From Figure 2.5, we can see that humans are less sensitive to higher and lower spatial frequencies. The anatomical reason behind this phenomenon is that the middle spatial frequency stimuli correspond better to the size of a neuron [2]. For clarity, we will not attempt to explain each CSF in detail but instead focus on one of the most popular CSFs—Barten's CSF.

[1] Contrast for sine-wave gratings is defined as the Michelson contrast ranging from 0-1.

[2] Details can be found in [12] pp 136.

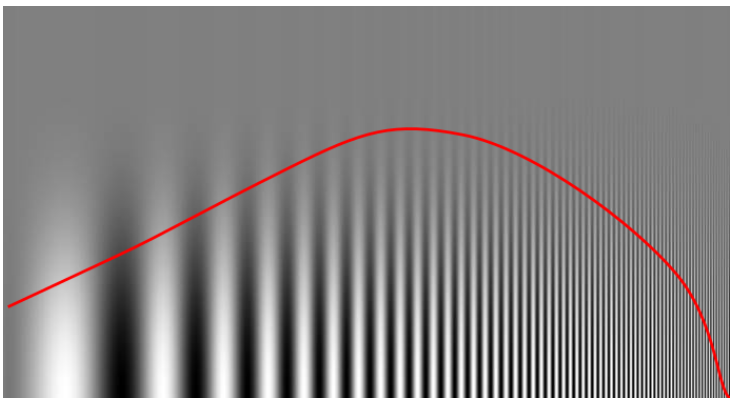


Figure 2.5: An approximate plot of contrast sensitivity function on the Campbell-Robson contrast sensitivity pattern [2].

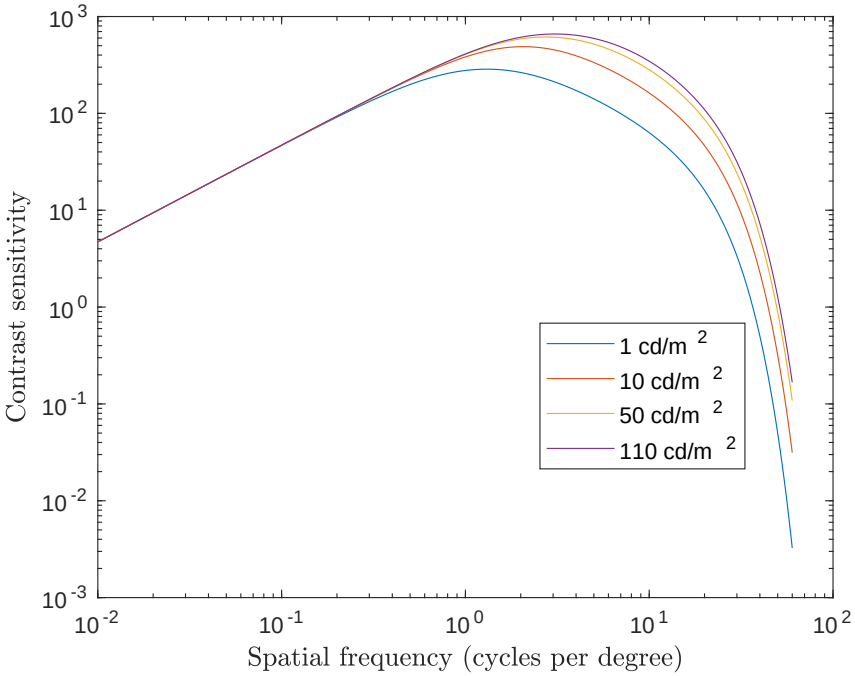


Figure 2.6: Barten's CSF at different luminances.

Barten's CSF is perhaps the most popular CSF in visibility research. In Barten's CSF, contrast sensitivity is defined as the inverse of the modulation threshold of a sinusoidal luminance pattern where parameters are fitted with human experimental data [16]:

$$S(\rho, L) = \frac{5200 \exp\left(-0.0016\rho^2(1+100/L)^{0.08}\right)}{\sqrt{(1+\frac{144}{X_0^2}+0.64\rho^2)(\frac{63}{L^{0.83}}+\frac{1}{1-\exp(-0.02\rho^2)})}} \tag{2.8}$$

where ρ is the spatial frequency of pattern, L is the luminance, X_0^2 is the area of the stimulus. We plot the Barten's CSF in Figure 2.6. Barten's CSF shows that contrast sensitivity is lower at lower luminance levels.

There are many different formulas and measurements for CSFs [2, 15, 17–24]. Each

CSF is measured under different experimental conditions, such as the number of observers, with or without limitation on the viewing time. For a more complete review of CSFs, readers are recommended to read Barten's review on CSFs [15]. The CSF can consider the effects of luminance and spatial frequency. However, the interactions between patterns with different spatial frequencies are not considered in the CSF but could influence the visibility of patterns to a significant degree.

2.2.5 Contrast masking

Contrast masking refers to the change in contrast sensitivity for test patterns presented on a non-uniform background. Contrast masking typically leads to reduced sensitivity, as illustrated in Figure 2.7. Contrast masking has been studied with relatively simple stimuli, such as static sine-wave gratings. Legge analyzed contrast masking in human vision by measuring the contrast threshold for sine-wave gratings in the presence of other masking sine-wave gratings [25]. Foley *et al.* found that the contrast masking model in [25] did not fit experimental data well, and non-linear functions had to be used [26]. Sometimes, the interaction of different spatial frequencies can enhance contrast sensitivity, an issue which is referred to as facilitation. Georgeson *et al.* found that facilitation of detection would diminish if the masking signal occurs asynchronously, and the phenomenon indicated that there were links between contrast masking and luminance masking, making it more difficult to model contrast masking correctly [27]. For natural images, facilitation rarely exists and popular IVMs do not model this effect [11]. Although contrast masking has been measured relatively well on simple static stimulus such as sine-wave gratings, the contrast masking in natural images is particularly problematic in terms of being modeled and predicted accurately [1]. The phenomenon of contrast masking clearly indicates that using a CSF alone cannot model the visibility of distortions correctly. Next, we will explain the multiresolution decomposition that enables the contrast masking to be included in the IVM.

2.2.6 Multiresolution decomposition

According to the multiresolution theory of vision [12], the human visual system transmits image information through multiple channels, where each channel represents different frequency and orientation information. Thus, it is natural to decompose the image into multiple channels and use the CSF to compute the contrast sensitivity of each channel separately. In addition, decomposing the image into multiple channels also allows us to model the contrast masking by selectively suppressing activity from channels with different spatial frequencies. Image pyramids representations are commonly used in IVMs for multiresolution decomposition and are computed as follows. We vectorize the input

Figure 2.7: An example of contrast masking. The right image is the distorted version of the left original image. The distortion noise is visible against the sky but more difficult to see against the high frequency parts of the image, such as grass and trees. From [11].

signal as a column vector I_1. Then, the convolution operation in image pyramids as a matrix C, where the rows of C contain the convolution kernels, the subsampling operation as a matrix S with only zero or one entries. Then, the basic pyramid operation can be written as $P = S\,C$. The generation of image pyramids can be formally written as:

$$I_{i+1} = P_i\,I_i = S_i\,C_i\,I_i,\ \ i = 1, \cdots, N \tag{2.9}$$

where N is the maximum number of image pyramids. Each pyramid can represent the image's information at a particular resolution or spatial frequency. If there are multiple convolution filters C with different orientations at each step, the multiresolution decomposition can also consider orientation information, which is used in steerable pyramids. There are different kinds of pyramids, such as Gaussian, Laplacian, and steerable ones. The main differences between different pyramids are the type and the number of convolution kernels used in each step [12]. Among these methods, steerable pyramids that can consider orientation information are used in the state-of-the-art white-box IVM [1]. It is also worth noting that the multiresolution decomposition was previously popular in computer vision research before the invention of more flexible models, such as deep neural networks. The multiresolution decomposition allows us to model visibility in multiple channels.

2.2.7 Probability summation

To summarize the overall information from multiple channels, the probability summation is used and therefore is an important step in IVMs [1, 28]. This summation assumes that the detection of visible differences is caused by several independent channels at the same time. For example, assuming that the probability of detecting differences at the spatial frequency f is P_f, the overall probability of detecting the difference P_{det} can be written as:

$$P_{det} = 1 - \prod_{i=1}^{N}(1 - P_{f_i}) \tag{2.10}$$

where N is the total number of spatial frequency channels. This probability summation assumes that the interactions of different factors in human visual systems for detecting the differences are independent, which may not hold in natural color images.

2.3 Image visibility metrics

Studies on IVMs are relatively sparse and mainly focus on the white-box modeling of data due to the extremely limited size of training datasets available. Existing IVMs can be generally divided into statistical IVM and human visual system model-based IVM (HVS-IVM), which we will explain in the following sections:

2.3.1 Statistical image visibility metrics

Statistical IVMs are simple metrics that use image statistics. The simplest statistical IVM might be the absolute difference metric, which computes the absolute pixel-by-pixel difference of reference and distorted image. Another statistical IVM is Spatial CIELAB (sCIELab) [29]. The sCIELab metric attempts to model the effects of spatial frequency by simply pre-filtering pixels in CIELab colorspace with a spatial-chromatic contrast sensitivity function prior to computing the visibility map. This simple approach has limited success in predicting visibility in natural color images, as shown in Section 4.6.

2.3.2 Human visual system model-based visibility metrics

To consider more complex visual phenomena, some human visual system model-based IVMs have been proposed, such as the Visual Discrimination Model (VDM) [30], the Visible Differences Predictor (VDP) [14], and the High Dynamic Range Visible Differences Predictor 2 (HDR-VDP-2) [1]. Those metrics consider luminance masking, contrast sensitivity, contrast masking, and frequency-selective channels [10]. Among these metrics, HDR-VDP-2 is the state-of-the-art IVM and is used in this thesis. We will first briefly introduce the high level ideas of HDR-VDP-2 based on the fundamentals of IVMs described in Section 2.2. Then, we will introduce HDR-VDP-2 in more detail. Because HDR-VDP-2 has several slightly different implementations on-line [3], in the rest of this thesis, we will refer to the up-to-date version of HDR-VDP-2 as HDR-VDP for simplicity.

[3] http://hdrvdp.sourceforge.net/wiki/

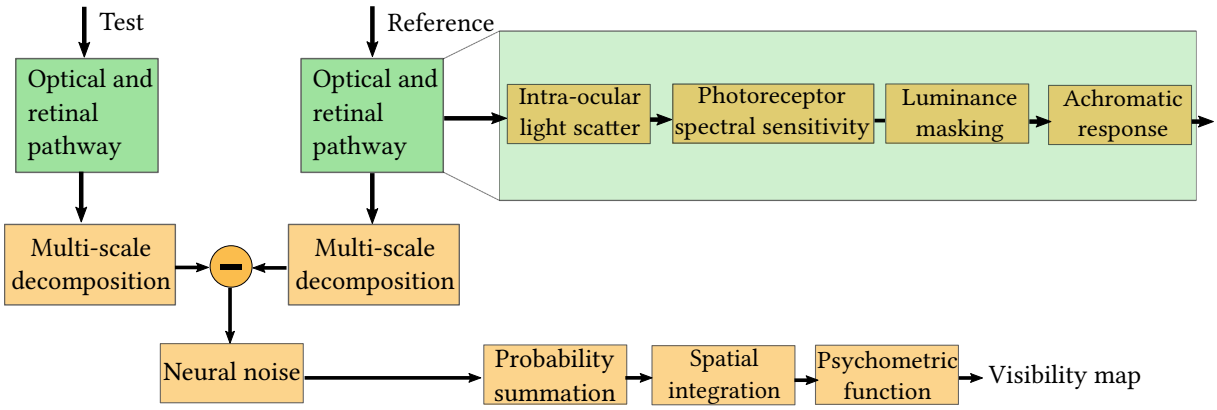


Figure 2.8: Architecture of HDR-VDP.

High dynamic range visual difference predictor (HDR-VDP [1]): The architecture of HDR-VDP is shown in Figure 2.8. Inspired by the physiology of the human visual system, HDR-VDP models the optics of the eyes attenuating the high spatial frequency information. Then, HDR-VDP models luminance masking with photoreceptor sensitivity data. Then, HDR-VDP uses the steerable pyramid transform for multi-resolution decomposition and uses a modified version of the Barten CSF to compute the contrast sensitivity at different channels. Finally, HDR-VDP uses probability summation to obtain the final prediction of the visibility map. Next, we will explain each step of HDR-VDP in detail [4].

Optical and retinal pathway:

(1) Intra-ocular light scatter. A small portion of the light traveling through the eye will scatter on its way to the retina [31]. This phenomenon attenuates high spatial frequencies and causes light pollution that reduces contrast. HDR-VDP models the intra-ocular light scattering as a modulation transfer function (MTF):

$$\mathcal{F}\left(L_o\right)[c] = \mathcal{F}(L)[c]\,\mathrm{MTF} \tag{2.11}$$

where L_o is the output image radiance map, $\mathcal{F}$ is the Fourier transform operator, [.] is the index of image channel, and L is the input image radiance map. The MTF function used in HDR-VDP is a generic model proposed by Ijspeert [32]:

$$\mathrm{MTF}(\rho) = \sum_{k=1}^{4} a_k \exp\left(-b_k \rho\right) \tag{2.12}$$

where a_k and b_k are the MTF's coefficients.

(2) Photoreceptor spectral sensitivity. The photoreceptor spectral sensitivity

[4]Readers without related backgrounds can jump the details of HDR-VDP that are irrelevant to the main content of this thesis. The details are described here only for understanding the state-of-the-art IVM.

curves describe the probability of sensing a photon at different wavelengths. As shown in Figure 2.9, L-, M-, S-cones, and rods exhibit different spectral sensitivities [33]. The probability of detecting a photon can be computed by multiplying photoreceptor spectral sensitivity and integrating out the wavelength term:

$$v_{L|M|S|R}[c] = \int_{\lambda} \sigma_{L|M|S|R}(\lambda)\, f[c](\lambda) d\lambda \tag{2.13}$$

where σ is the spectral sensitivity of L-, M-, S-cones, or rods, c is the index of input radiance map ranging from 1-3 for RGB color images, and $f[c]$ is the input light spectrum that is related to the display. Then, the total amount of light sensed by each photoreceptor type is:

$$R_{L|M|S|R} = \sum_{c=1}^{C} L_o[c]\, v_{L|M|S|R}[c] \tag{2.14}$$

where C is the total number of input spectral maps (for color images, $C = 3$). This step is equal to color space transformation in the real implementation, making HDR-VDP more accurate with different types of displays.

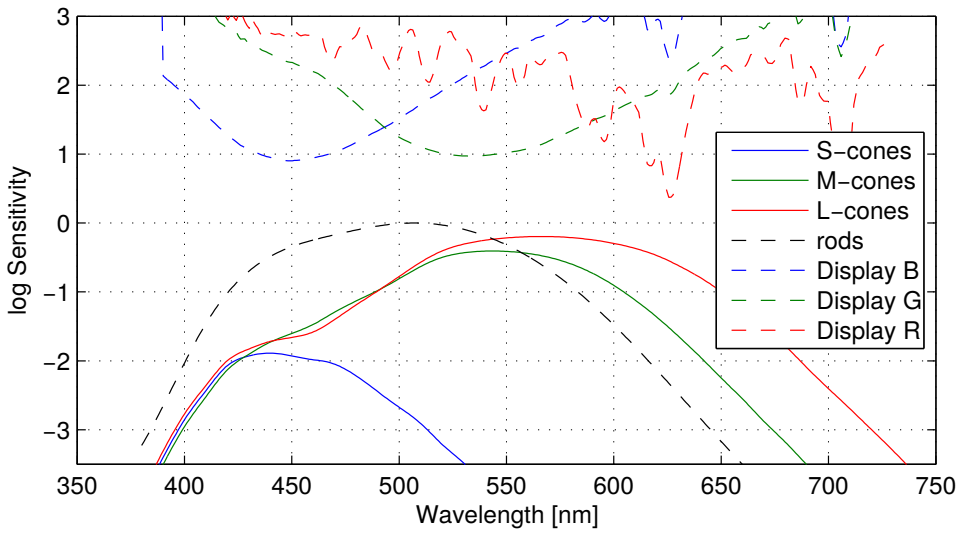


Figure 2.9: Photoreceptor spectral sensitivity curve. The curve in the upper part shows the measured emission spectra for a CRT display. From [1].

(3) Luminance masking. HDR-VDP models luminance masking with non-linear transducer functions $t_{L|M|R}$:

$$P_{L|M|R} = t_{L|M|R}(R_{L|M|R}) \tag{2.15}$$

where $P_{L|M|R}$ is the photoreceptor response for L-, M-cones, and rods, $R_{L|M|R}$ is the corresponding photoreceptor input. HDR-VDP omits the modeling of S-cones as they

have almost no effect on the luminance perception. The transducer function is:

$$t_{L|M|R} = s_{\text{peak}} \int_{r_{min}}^{r} \frac{s_{L|M|R}(\mu)}{\mu} d\mu \tag{2.16}$$

where r is the photoreceptor absorbed light ($R_{L|M|R}$), r_{min} is the minimum detectable luminance (10^{-6} cd/m^2), and $s_{L|M|R}$ is the adjustment for the peak sensitivity of visual system. To derive $s_{L|M|R}$, we first considered the combined sensitivity of all types of photoreceptors are:

$$s_A(l) = s_L(r_L) + s_M(r_M) + s_R(r_R) \tag{2.17}$$

where l is the photoreceptor luminance, r_L, r_M, and r_R are absorbed luminance by L-cones, M-cones, and rods respectively. Although the combined sensitivity of luminance $s_A(l)$ is captured in [34], measuring isolated photoreceptor luminance sensitivity for each type is difficult because cones and rods are hard to separate in human experiments. However, data exist for measurements of a person without cone vision. Given photopic luminance $l = r_L + r_M$, HDR-VDP assumes that $r_L = r_M = 0.5l$, which means that L-cones and M-cones have the same sensitivity function. Then, The L-cone's and M-cone's sensitivity function can be obtained by subtracting the overall sensitivity with the rod's sensitivity:

$$s_{L|M}(l) = 0.5(s_A(2l) - s_R(l)) \tag{2.18}$$

The resulting transducer function is shown in Figure 2.10.

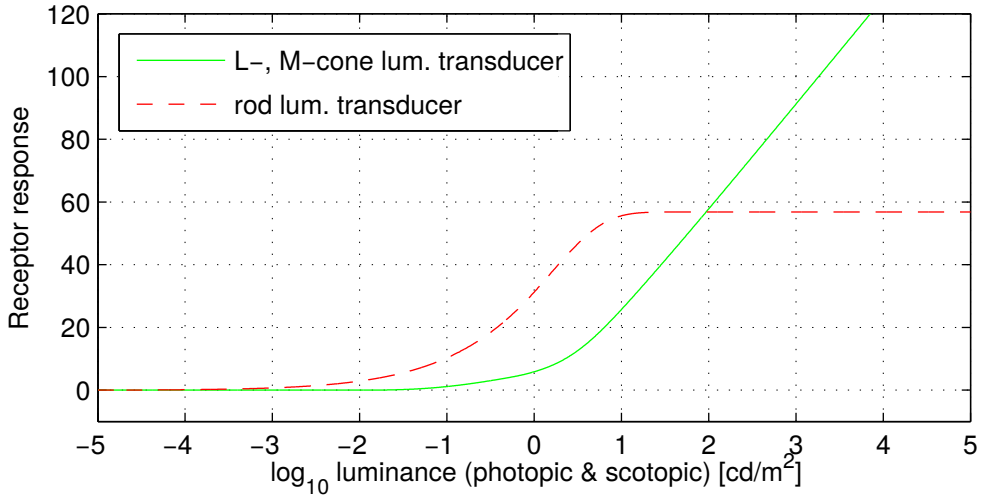


Figure 2.10: Luminance transducer function for cones and rods. From [1].

In this step, HDR-VDP omits the modeling of S-cones here because S-cones contribute a relatively small amount in the total response. However, S-cones are found to play important roles in sensing color contrast at low brightness [35]. This simplification makes

HDR-VDP less accurate. The achromatic response is then computed as:

$$P = P_L + P_M + P_R \tag{2.19}$$

Neural noise: This step models effects, such as contrast masking. HDR-VDP uses the steerable pyramid transform [36] to decompose input achromatic response into orientation and spatial frequency bands. There are four orientation bands, and the number of spatial frequency bands is limited by the image angular resolution. HDR-VDP assumes that the differences in the contrast detection of different frequencies and orientations are due to the sum of several sources of noise. This process is modeled as the sum of the signal-independent noise (neural CSF) and signal-dependent noise (contrast masking) [5]. At the f-th spatial frequency band and the o-th orientation, the steerable pyramids $B_{T|R}[f, o]$ are computed for the test and the reference image, which is referred to as "Multi-scale decomposition" in Figure 2.8. Then, the noise-normalized channel difference is:

$$D[f, o] = \frac{|B_T[f, o] - B_R[f, o]|^p}{\sqrt{N^{2p}_{\text{nCSF}}[f, o] + N^2_{\text{mask}}[f, o]}} \tag{2.20}$$

where p is the hyper-parameter set as 3.5 in HDR-VDP.

The signal-independent noise N_{nCSF} in the neural system is constructed by dividing the CSF function by the MTF function and the joint photoreceptor luminance sensitivity s_A:

$$N_{\text{nCSF}}[f, o] = \frac{1}{\text{nCSF}[f, o]} = \frac{\text{MTF}(\rho) s_A(l)}{\text{CSF}(\rho, l)} \tag{2.21}$$

where ρ is the spatial frequency, l is the photopic luminance after intra-occular scatter $l = R_L + R_M$, CSF is the CSF for HDR-VDP, and it is fitted with experimental data based on a simplified version of Barten's CSF model, ρ is the spatial frequency at the f-th spatial frequency band, which can be computed as:

$$\rho = \frac{n_{\text{ppd}}}{2^f} \tag{2.22}$$

where n_{ppd} is the input image's angular resolution in pixel per degree. The signal-dependent part of the neural noise models the masking effect (described in Section 2.2.5). The contrast

[5] The term "signal-independent" and "signal-dependent" are used in the original paper but may better be described as "without channel interactions" and "with channel interactions". We will use the original terms for consistency.

masking in HDR-VDP consists of three parts:

$$N_{\text{mask}} = \frac{k_{\text{self}}}{n_f}(n_f B_M[f,o])^q + \frac{k_{\text{xo}}}{n_f}\left(n_f \sum_{i=O/o} B_M[f,i]\right)^q + \frac{k_{\text{xn}}}{n_f}(n_{f+1}B_M[f+1,o] + n_{f+1}B_M[f-1,o])^q) \tag{2.23}$$

where the first term models self-masking, the second models masking across orientations, and the third is the masking due to two signals of neighboring frequency bands [6]. k_{self}, k_{xo}, and k_{xn} are the weights controlling the influence of masking. O is the set of all orientations. n_f is the normalization factor $n_f = 2^{-(f-1)}$. $B_M[f,o]$ is the activity in the band f and orientation o. B_M is implemented as a Gaussian blurred version of B_T. We recommend that readers refer to the original paper of HDR-VDP [1] for the exact mathematical formula of B_M and CSF.

Probability summation: To summarize the overall information from multiple channels, after obtaining the noise-normalized difference signal $D[f,o]$ from the above steps, we can compute the probability of detecting a difference at frequency f and orientation o with the psychometric function, which can be written as:

$$\psi[f,o] = 1 - \exp\left(\log(0.5)D^{\beta}[f,o]\right) \tag{2.24}$$

where β is the slope of the psychometric function. Then, the overall probability of a detection map considering all orientations and frequencies can be written as:

$$\begin{aligned} P_{map} &= 1 - \Pi_{(f,o)}(1 - \psi[f,o]) \\ &= 1 - \exp\left(\log(0.5)\sum_{(f,o)} D^{\beta}[f,o]\right) \end{aligned} \tag{2.25}$$

This probability map is in the steerable pyramid representation. To obtain the probability map in the pixel domain, HDR-VDP uses the inverse steerable pyramid prior to modeling the psychometric function and adds spatial summation:

$$P_{map} = 1 - \exp\left(\log(0.5)\text{SI}\left(\mathcal{P}^{-1}(D^{\beta})\right)\right) \tag{2.26}$$

where $\mathcal{P}^{-1}$ is the steerable pyramid reconstruction operator, and SI is the added spatial integration. The spatial integration function describes the phenomenon that larger distortion patterns are easier to detect. HDR-VDP applies the ratio of the summation of pixel

[6] The self-masking term can adjust the strength of effects of mutual masking

values over the entire image and the maximum of the pixel value for spatial integration:

$$\mathrm{SI}(S) = \frac{\sum_{i,j} S_{ij}}{\max_{i,j} S_{ij}} \tag{2.27}$$

where S is the input image matrix. HDR-VDP has achieved tremendous success for visibility predictions. However, there are several limitations that prevent HDR-VDP from being more accurate in predicting visibility maps:

Limitations of HDR-VDP

1. Complex non-linear interactions exist between components. For white-box models to be accurate, there are many components to consider. With the increasing number of components, improving one component may have advert effects on another. Thus, it becomes harder to improve the components.

2. Assumptions of the human visual system. For computational purposes, HDR-VDP has to make many assumptions. For example, HDR-VDP assumes that contrast masking only happens between neighboring frequency bands. However, this assumption does not necessarily hold.

3. Stimuli used in psychophysical experiments may be too simple to apply to natural color images. HDR-VDP uses data from psychophysical experiments. However, these psychophysical experiments are usually conducted using simple stimuli such as log Gabor patches. The result might not be the same for complex natural color images.

The limitations of HDR-VDP suggest that improving IVM predictions by modeling the human visual system may be rather difficult. Later, we will discuss how to use black-box deep neural networks based on machine learning to improve the accuracy of visibility prediction in Chapter 4 and Chapter 5.

2.4 Image compression

A large amount of image and video data on the Internet poses a significant challenge for data transmission and storage. Image compression techniques typically rely on the fact that the human visual system is less sensitive to high-frequency distortions and color differences than low-frequency distortions and luminance differences. Image compression methods can be divided into lossy image compression methods and lossless image compression methods. Lossless image compression is a reversible image compression method that can recover all of the original image information from the compressed data. However, lossless image compression methods generally produce much larger image files than lossy

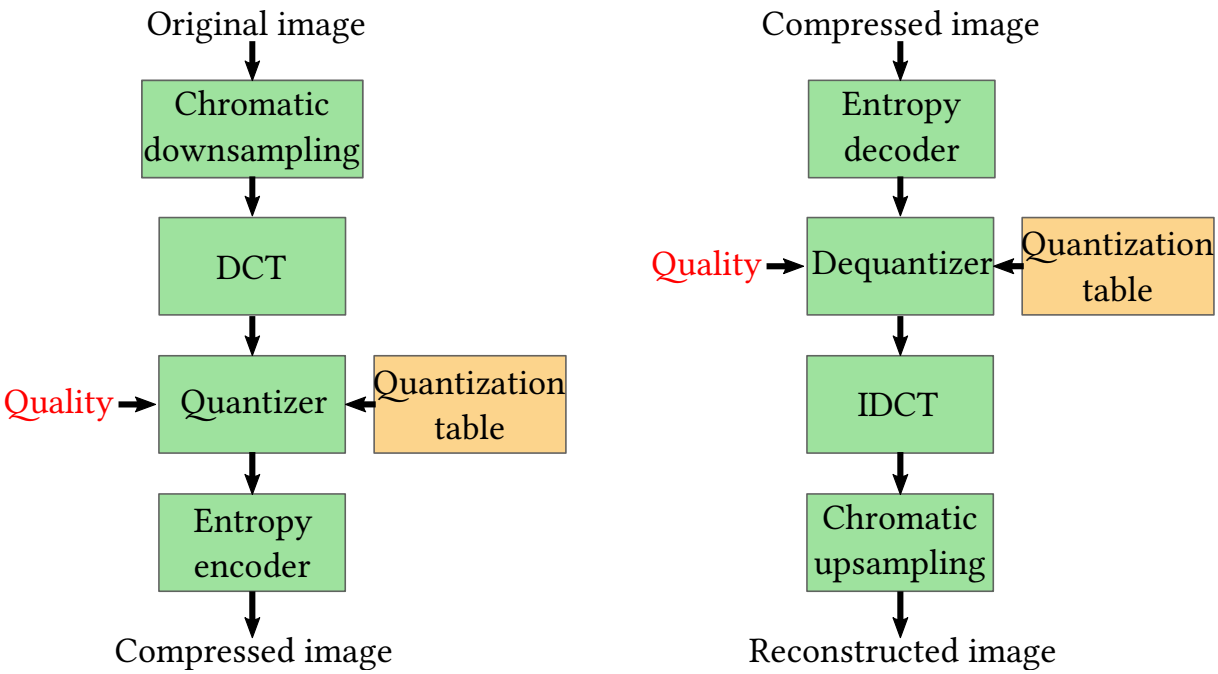


Figure 2.11: Flow of JPEG image compression

image compression methods. This issue has restricted their applications for compressing large-scale images on the Internet. This thesis will focus on lossy image compression methods. In the following section, we will explain the JPEG compression in detail, and then we will briefly discuss WebP, which is another image compression method used in this thesis.

2.4.1 Standard JPEG image compression

Joint Photographic Experts Group (JPEG) image compression is the most popular image compression format on the Internet [37]. In this thesis, we use the JPEG file interchange format (JFIF) format of JPEG and the widely-used open source implementation of the format—libjpeg[7] from the independent JPEG group (IJG). The flow of JPEG image compression is shown in Figure 2.11. JPEG first transforms the input images in the RGB color space to the Y′CbCr color space. In the Y′CbCr color space, images are represented with the luma component (Y′), blue-yellow chroma component (Cb), and red-green chroma component (Cr). The reason behind the color space transformation is that red, green, and blue coordinates of color are correlated with each other, and the principle component analysis shows that the three main components in natural images are luminance, red-green, and blue-yellow color channels [11]. Then, the chroma components of the original images are spatially subsampled to reduce the data. The subsampling exploits the fact that the human eyes' spatial resolution for color is lower than for luminance. Next, JPEG uses

[7]https://github.com/LuaDist/libjpeg

the discrete cosine transform (DCT) [38] to transform images to the frequency domain and quantifies the coefficients controlled by a parameter-quality. The higher the quality, the more precise the coefficients are quantified. Finally, the entropy encoder stores the coefficients in zig-zag order. The reconstruction of images from the compressed image data follows the same steps as in compression, but in the reverse order.

The following paragraphs will introduce each JPEG compression step in more detail because many image compression algorithms follow similar design principles as JPEG.

Color space transformation To compress images with fewer distortions, JPEG first transforms the images from the linear color space to the Y′CbCr color space. The Y′ channel represents the luma that is related to image brightness. The other channels represent chromaticity [39]. The following steps of JPEG compression occur independently in each channel of Y′CbCr. The conversion from RGB color space to Y′CbCr color space is linear:

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 & 0 \\ -0.168736 & -0.331264 & 0.5 & 128 \\ 0.5 & -0.418688 & -0.081312 & 128 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \\ 1 \end{bmatrix} \tag{2.28}$$

Original Y′ component Cb component Cr component

Figure 2.12: RGB color space to Y′CbCr color space transformation.

Downsampling This step attempts to utilize the fact that the human spatial resolution for color is lower than for luminance. Instead of using all of the pixels in chromatic channels, JPEG provides different subsampling rates for chromaticity channels, such as 4:4:4 or 4:2:0. The first number denotes the number of samples for the luma channel that is usually 4, the second number denotes the number of chroma samples following the luma samples, and the last number denotes the number of changes of the chroma samples in the subsequent row. [8].

[8] 4:4:4 means no subsampling in chromaticity channels.

DCT Transform In this step, JPEG splits images into 8×8 patches and runs the DCT on each patch to obtain image representations in the frequency domain. The DCT transform approximates continuous signals with the weighted sums of cosine functions at different frequencies [39]. For example, to decompose a one-dimensional discrete real-valued function $f(x)$:

$$f(x) = \frac{1}{N}c_0 + \frac{2}{N}\sum_{k=1}^{N-1} c_k \cos\left(\frac{\pi}{N}k(x+\frac{1}{2})\right) \tag{2.29}$$

The coefficient c_k is the weight for the k-th frequency:

$$c_k = \sum_{x=0}^{N-1} f(x) \cos\left(\frac{\pi}{N}k(x+\frac{1}{2})\right) \tag{2.30}$$

The one-dimensional DCT can be extended to a two-dimensional DCT to decompose a two-dimensional discrete function $f(x, y)$:

$$f(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C_u C_v c_{u,v} \cos\left(\frac{(2x+1)u\pi}{16}\right)\cos\left(\frac{(2y+1)v\pi}{16}\right) f(u,v) \tag{2.31}$$

where C_u or C_v is $\frac{1}{\sqrt{2}}$ when u or v is 0, otherwise $C_u = C_v = 1$. $c_{u,v} = C_{u+8v}$ [39]. From this formula, there are 8x8 base cosine functions for decomposing image patches.

Quantization The quantization step is where JPEG achieves most of its compression. In this step, the DCT coefficients obtained from the DCT transform are quantized by elementwisely dividing a quantization matrix. Remainders are rounded to the nearest integers. The amount of compression is determined by a single parameter—compression quality ranging from [0-100] in the libjpeg. The default quantization matrix for luma channel is:

$$Q_{\text{luma}} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{2.32}$$

The quantization matrix for chroma channel is different, which is:

$$Q_{chroma} = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix} \tag{2.33}$$

Many entries in the chroma channel quantization matrix are larger than the luma channel quantization matrix to take the advantage of fact that the human visual system is more tolerable to the difference of color than luminance [40]. These two quantization tables are used as examples in the Annex K.1 of the ISO/IEC JPEG standard [37]. It is also worth noting that some commercial software, such as Adobe Photoshop, has its own quantization tables that are usually more conservative than the ones shown here [9]. According to the IJG's formula [10], the quantization scaling factor S at the compression quality q can be computed as:

$$S = \begin{cases} 5000/q, & q <= 50 \\ 200 - 2 * q\,, & q > 50 \end{cases} \tag{2.34}$$

Then, the quantization matrix is scaled according to the scaling factor S to provide different precision of quantization. Details can be found in the libjpeg implementation[11]. It is also worth noting that some software, such as Photoshop CS 6, employs a different quality range in the user interface, but actually stores the quality ranging from 0-100. The JPEG compression implemented in Photoshop CS 6 is more conservative than common settings of libjpeg because of the quantization matrices used and the high default quality (85-99). Due to this reason, the percentage of savings reported in the later chapters compared with the Photoshop setting is a lot more conservative. However, for simplicity, we will use the libjpeg implementation of JPEG and the recommended quantization matrices.

Entropy encoding Entropy encoding is a lossless step that combines the run-length encoding (RLE) and Huffman encoding to help encode zero coefficients more efficiently.

From the JPEG compression algorithm, we can see that by adjusting the quality

[9] Different quantization tables can be found in https://www.impulseadventure.com/photo/jpeg-quantization-lookup.html?src1=255

[10] Request for Comments 2435 and the code in line 130-145 in https://github.com/LuaDist/libjpeg/blob/6c0fcb8ddee365e7abc4d332662b06900612e923/jcparam.c

[11] Line 49-56 in https://github.com/LuaDist/libjpeg/blob/6c0fcb8ddee365e7abc4d332662b06900612e923/jcparam.c

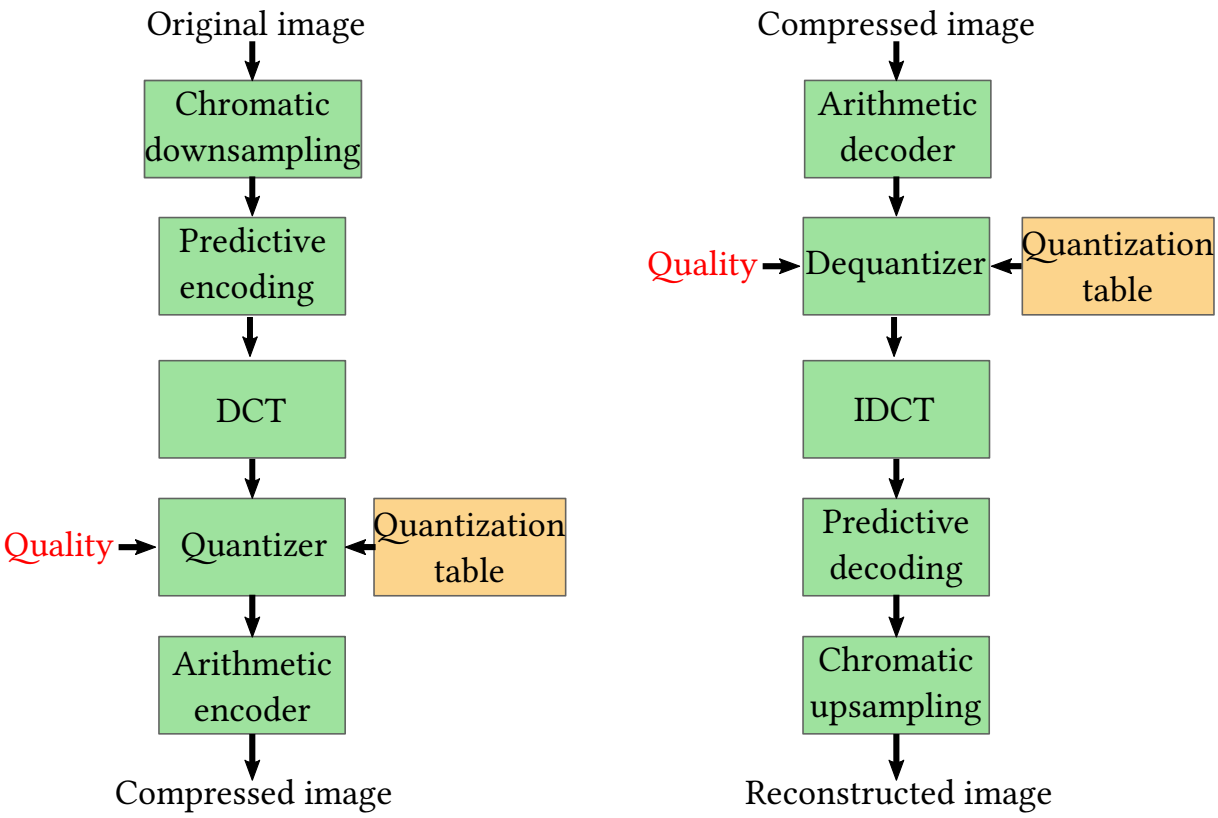


Figure 2.13: Flow of WebP image compression.

parameter, users can easily control the trade-off between the visual quality of JPEG compressed images and the storage space required.

2.4.2 WebP image compression

Created by Google in 2012, WebP is becoming increasingly popular in web image compression [41]. WebP is based on the VP8 video codec, in which predictive coding is an important feature. The flow of WebP image compression is shown in Figure 2.13. The main difference between WebP and JPEG is that WebP encodes the differences between neighboring image blocks rather than image blocks themselves and uses arithmetic encoding instead of the entropy encoding to compress quantized DCT coefficients. By encoding image block differences, WebP can represent images with local similarities more efficiently. The Huffman encoding used in JPEG represents each unique DCT coefficient with a different symbol (a sequence of binary numbers). This process could waste some space if some bit symbols are unused. WebP uses arithmetic encoding to represent DCT coefficient sequences with a fraction ranging from 0 to 1. For further details, see [42]. Designed for compressing images on the web, WebP is better at compressing images at lower bit rates, as we will demonstrate later in Section 6.4.

2.4.3 Other lossy image compression methods

Despite the wide use of JPEG, a significant amount of research has been conducted on proposing new lossy image compression encoding for different applications. For example, JPEG 2000 is a relatively new standard that is widely used in some fields such as digital cinema.

Recently, the neural network[12]-based image compression has become a popular topic. Toderici *et al.* proposed a recurrent neural network[43]-based encoder and decoder [44], which is the first neural network architecture that is able to outperform JPEG at image compression. Rippel *et al.* proposed an autoencoder neural network[45] image compression [46]. The autoencoder neural network is more efficient than the recurrent neural network at image compression as it utilizes the similarities of all neighboring pixels. Other methods have been proposed [47–54] mainly by either improving the neural network architecture or the form of the loss for training the neural network. To illustrate how these methods work, we take the widely cited Rippel's compression algorithm as an example. Rippel's compression algorithm compresses 128×128 blocks from images independently, making it able to compress images with arbitrary large resolutions. The general flow of this compression algorithm is shown in Figure 2.14. The main difference between this algorithm and the classic image compression algorithms such as JPEG is that deep neural networks are used for feature extraction. The original image is transformed into the feature space and transformed back to the reconstructed image. The differences between the original image and the reconstructed image are used as the loss function for training neural networks.

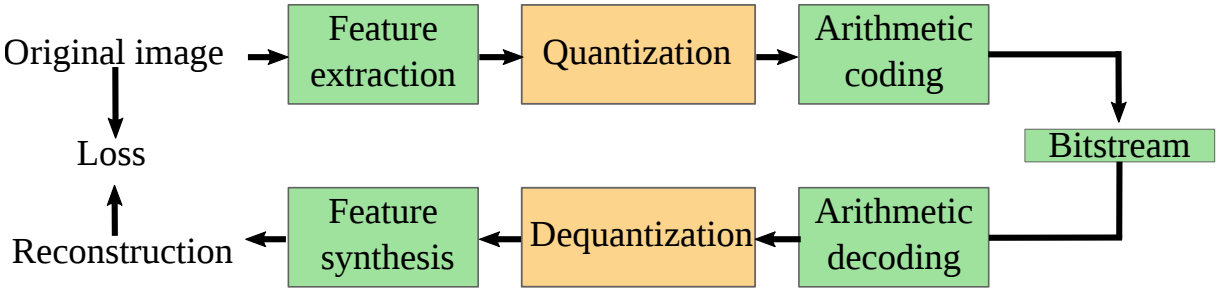


Figure 2.14: Flow of image compression based on deep neural networks.

The neural network-based image compression methods have demonstrated better compression performance than classic image compression algorithms, such as JPEG and WebP. However, these methods have to use the graphic processing unit (GPU) to achieve similar speeds as JPEG, and therefore it is difficult to use these methods for mobile devices that have limited power storage. In addition, although the deep neural network's

[12]We refer readers to Section 2.6 for background knowledge on neural networks.

weights do not take too much storage space (probably less than 60 MB) [13], installing the necessary software takes a large amount of storage space [14]. While all these methods indicate a possible new direction and demonstrate better performance than the JPEG standard, JPEG will possibly remain the most popular lossy image compression format on the Internet in the near future.

2.4.4 Problem with distortion measurements

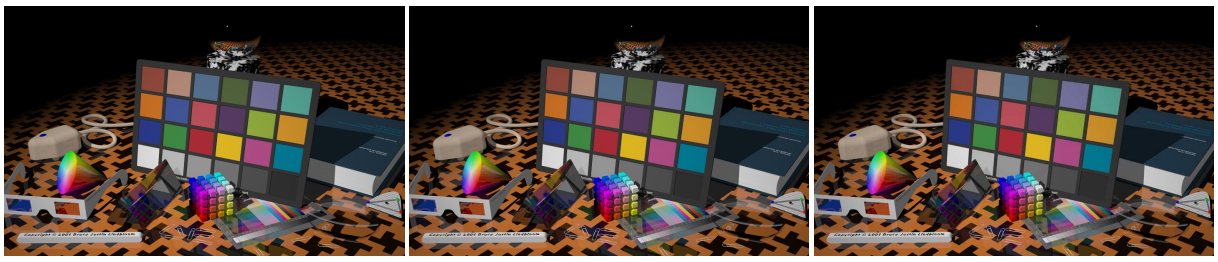
While significant effort has been invested in better image and video coding methods, aligning these methods with visual performance has received much less attention. Besides, it is difficult to compare these methods because conducting large-scale user experiments is too expensive and time consuming. However, widely used simple image quality metrics, such as the PSNR or SSIM, do not correlate well with human perceptions of image quality. For visually lossless image compression, we need highly accurate metrics to determine whether compression artifacts are visible or not. Although limited in number, there are still some previous works on visually lossless image encoding that are relevant to the IVMs study. Next, we will introduce the previous research on visually lossless image compression below:

2.5 Visually lossless image compression

Image compression methods are traditionally categorized as lossy or lossless. Whereas lossless methods retain the original information up to the bit-precision of the digital representation, they are many times less efficient than lossy methods. The Shannon source encoding theorem states that how much we can compress information in a lossless way depends on the variance in information sources [55]. Given the high variance of contents of natural color images, improving existing lossless image compression methods to a large extent can be relatively difficult. However, humans are not sensitive to small changes in images. Thus, it may not be necessary to retain all the information in images to make the differences between compressed images and reference images invisible. A natural question is: Can we control parameters of lossy image compression methods to make the difference between the compressed image and the original image invisible to observers (*i.e.*, visually lossless image compression)? To conceptually demonstrate this possibility, we show an example of an image compressed in a visually-lossless manner in Figure 2.15. By controlling the JPEG compression quality setting, we can use the lossy image compression method to achieve visually lossless image compression. Such visually lossless compression

[13]The exact parameters of Rippel's architecture is not available in the paper. The number is based on this implementation— https://github.com/rgerd/adaptive-image-compression.

[14]Deep learning frameworks, such as Tensorflow-GPU, typically take more than 1 Gigabytes.



Reference JPEG Quality 90 JPEG Quality 52

Figure 2.15: Image named "artificial" is compressed at the compression quality of 90 or 52 using JPEG encoder. The default visually lossless JPEG compression quality for Photoshop is around 90 [15]. We measured the visually lossless JPEG compression quality for an average observer to be 52 under the viewing condition of 110 cd/m^2 display peak brightness and 60 ppd. The measurements come from the VLIC dataset in Section 3.3.

methods are positioned in-between lossy and lossless image compression: they introduce compression distortions but ensure that those distortions are unlikely to be visible. Another line of research worth noting here is perceptual lossless image compression based on just noticeable distortion (JND) models [56, 57]. Perceptual lossless image compression belongs to the category of supra-threshold image compression methods that aims at compressing images to the same quality, which is different as the same MOS score does not mean no visible differences as in visually lossless image compression. (This will be shown in Section 6.2 with experiments.)

Visually lossless image compression was first introduced to compress medical images to handle the increasing amount of data in clinics' picture archiving and communication systems [58]. By selecting a fixed compression parameter or modifying the compression encoders, visually lossless compression has been shown to be effective in compressing medical images in the gray-scale domain. However, previous research on visually lossless compression is largely content- and encoder-dependent [59–63], which means that we cannot apply the same algorithm for medical images as for everyday pictures. Previous research in this area can be divided into three categories: (1) constant visually lossless compression threshold (VLT) compression, (2) visual optimization for image coding, and (3) metric-based visually lossless compression. Predicting the visibility of distortions is the key in many visually lossless image compression methods, and one of the methods is HDR-VDP as we discussed in Section 2.3.2. Among these methods, visual optimization for image coding requires that the existing standards for image communication change, making it less applicable in practice.

2.5.1 Constant VLT compression

Constant VLT methods use a single number to determine compression quality. Kocsis *et al.* [59] found that VLT of JPEG2000 compression for micro-calcification in mammography

corresponds to a compression ratio (CR) of 40:1. Lee *et al.* estimated VLT CR as 5:1 for abdominal computed tomography (CT) images by alternatively displaying compressed and original images on the same screen and asking observers whether they could see the difference [60]. Constant VLT methods work well for specific types of gray-scale medical images, but these findings do not translate to other types of content. Note that the constant VLT compression has already been widely applied in commercial software. For example, Amazon.co.uk uses a default visually lossless JPEG compression quality of 75.

2.5.2 Visual optimization for image coding

Another line of research is focused on improving the visual performance of image coders for specific types of images to make the compression visually lossless. Zeng *et al.* [62] introduced several additional stages in JPEG 2000 coding that took account of frequency-dependent contrast sensitivity and visual masking. Wu *et al.* proposed a visually lossless medical image coder with an additional stage of visual pruning, controlled by a visual system model[63]. The encoder, however, required manual calibration of the visual model parameters for each type of encoded medical image.

2.5.3 Metric-based visually lossless image compression

There are many metric-based visually lossless image compression methods [56, 64–66]. Watson *et al.* proposed to measure the visibility of quantization noise in wavelet image compression methods. However, Watson *et al.* measured with simple Gabor patches, and their results do not generalize well to natural color images. Eckert *et al.* summarized the observations when using IQMs for visually lossless image compression [67]: Simple metrics, such as MSE, can have problems dealing with images with different textures. In addition, the lack of contrast masking modeling might be a significant reason for the failure of many quality metrics across a range of image contents. In addition, Eckert *et al.* noted that there was no general consensus about how contrast masking models should be designed. Chandler *et al.* proposed a visually lossless compression method for CT scans images, which predicts the maximum contrast that wavelet subband quantization distortions can achieve in the compressed image in order to be remain visually undetectable [61]. Kim *et al.* proposed a method for predicting visible artifacts in JPEG 2000 using PSNR and HDR-VDP [66].

Recently, Alakuijala *et al.* proposed an IVM named "Butteraugli" to find the VLT for JPEG images [68]. Butteraugli transforms an input image pair into a set of feature maps, such as an edge detection map, which are then combined to predict differences between a compressed image and a reference image. The maximum value in the difference map is taken as the indicator of compression artifact visibility.

The performance of metric-based visually lossless image compression relies on the accuracy of the IVM.

2.6 Machine learning

Machine learning is generally defined as a set of methods for automatically learning functions from data to complete tasks without explicit human instructions [69, 70]. Machine learning has been developing rather rapidly in the past decade. It has achieved superior performance in tasks that were previously impossible, such as large-scale image classification. For example, Alex *et al.* proposed a deep neural network-based image classification method, which might be the first important task for deep learning methods to demonstrate significantly improved performances [71]. In recent years, deep learning has also been successful in image generative modeling [72–74] and image quality assessment [75, 76, 76–78]. However, very few machine learning related studies can be found for IVMs and visually lossless image compression. Machine learning methods can be divided into three types; (1) supervised learning, (2) unsupervised learning, and (3) semi-supervised learning, depending on whether true labels are provided. In supervised learning, all data have labels. In unsupervised learning, no labels are available. In semi-supervised learning, only a part of the data has labels. Next, we will give a brief introduction to the machine learning methods used in this thesis.

2.6.1 Supervised learning

In the setting of supervised learning, the goal is to automatically infer a system capable of mapping inputs $\mathbf{x}$ to predict labels $\mathbf{y}$ based on a labeled training dataset $\mathcal{D} = (\mathbf{x}_i, \mathbf{y}_i)_{i=1}^{N}$. Supervised learning methods can be further divided into classification and regression methods depending on the type of the label. When $\mathbf{y}$ is categorical data, this type of supervised learning is termed classification. When $\mathbf{y}$ is continuous data, this type of supervised learning is termed regression. In the context of this thesis, $\mathbf{x}_i$ is the distorted image and the reference image. $\mathbf{y}_i$ is the visibility map, the values of which are continuous. Thus, our problem can be categorized as a regression problem.

2.6.2 Feedforward neural network

The feedforward neural network (multi-layer perceptron) was initially proposed in the 1950s [79, 80]. Although the neural network has become quite a hot topic in recent years, it was not regarded as a promising method for machine learning in the beginning [81]. The feedforward neural network is often referred to as the fully connected neural network to distinguish it from the convolutional neural network. The architecture of a fully connected

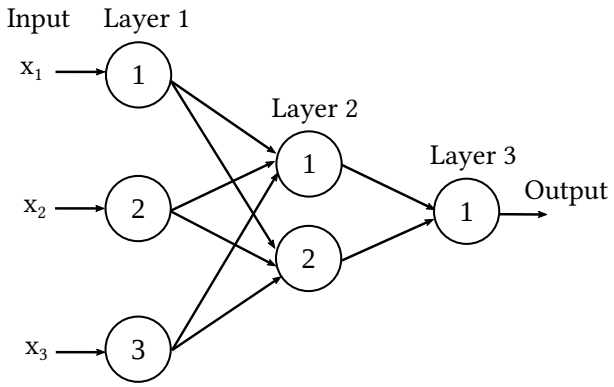


Figure 2.16: Architecture of a 3-layer fully connected neural network.

neural network is shown in Figure 2.16. In this form of a neural network, each node (neuron) accepts the inputs from all nodes in the previous layer and generates output to each neuron in the next layer based on a non-linear function that is generally referred to as the activation function.

We denote the connection weights from the i-th neuron to the j-th neuron at the l-th layer as $w_{i,j,l}$, the connection bias from the i-th neuron to the j-th neuron at the l-th layer as $b_{j,l}$, the activation from the i-th neuron to the j-th neuron is given by:

$$y_{j,l} = \Phi\left(\sum_i w_{i,j,l}\, y_{i,l-1} + b_{j,l}\right) \tag{2.35}$$

where Φ is the activation function that can provide flexibility for fitting complex non-linear functions. Popular choices of activation functions include the sigmoid function, the rectified linear units function (ReLU) [82], the exponential linear units (ELU) [83], and the scaled exponential linear units (SeLU) [84]. Recently, automatic searching for optimal non-linear activation function method has been proposed [85]. However, due to its complexity, this thesis will only focus on fixed non-linear activation functions. ReLU function is the most popular choice in practice due to its simplicity and efficiency:

$$\Phi(\mathbf{x}) = \max(\mathbf{0}, \mathbf{x}) \tag{2.36}$$

To train the fully connected neural network, the backpropagation algorithm is used [86]. For simplicity of notations, we denote the weights of the l-th layer as a matrix $\mathbf{w}_l$, the bias of the l-th layer as a vector $\mathbf{b}_l$, $\mathbf{y}_l$ as the output activation vector of l-th layer, $\odot$ as the Hadmard product operator. Then, the backpropagation algorithm is similar to the chain rule in calculus and can be explicitly written in Algorithm 1.

By running Algorithm 1 until convergence, a fully connected neural network can be trained to learn any continuous function. This property is referred to as the universal approximation theorem [87]. From Algorithm 1, we can also observe that as long as we

Algorithm 1 One step of backpropagation algorithm

Input: $\mathbf{x}$: the input data; $\mathbf{y}$: the input label; L: the number of layers; η: the learning rate; $\mathbf{w}_l$: the weights matrix at l-th layer, $\mathbf{b}_l$: the bias vector at l-th layer, Φ_l: the activation function at l-th layer, Cost is the loss function.

Feedforward computation:

for $l = 1 \dots L$ **do**

compute outputs of l-th layer:

$$\mathbf{y}_l = \Phi_l \left(\mathbf{w}_l \mathbf{y}_{l-1} + \mathbf{b}_l \right) \tag{2.37}$$

end for

Compute backpropagated error of prediction of the last layer:

$$\delta_L = \frac{\delta \text{Cost}(\mathbf{y}_L, \mathbf{y})}{\delta \mathbf{y}_L} \odot \Phi_L^{'} \tag{2.38}$$

Backpropagate the error:

for $l = L - 1 \dots 1$ **do**

compute the backpropagated error at l-th layer:

$$\delta_l = ((\mathbf{w}_{l+1})^T \delta_{l+1}) \odot \Phi_l^{'} \tag{2.39}$$

end for

for $l = L \dots 1$ **do**

Compute the backward gradient:

$$\frac{\partial \delta_L}{\partial \mathbf{w}_l} = \delta_l \mathbf{y}_{l-1} \tag{2.40}$$

$$\frac{\partial \delta_L}{\partial \mathbf{b}_l} = \delta_l \tag{2.41}$$

Compute the updated parameters:

$$\mathbf{w}_l = \mathbf{w}_l - \eta \frac{\partial \delta_L}{\partial \mathbf{w}_l} \tag{2.42}$$

$$\mathbf{b}_l = \mathbf{b}_l - \eta \frac{\partial \delta_L}{\partial \mathbf{b}_l} \tag{2.43}$$

end for

can compute the gradients of the activation functions, the backpropagation algorithm can be used to train our neural networks. In practice, we can even estimate the gradients when the real gradients are not available [88]. Although the fully connected neural network is theoretically guaranteed to approximate any continuous functions, it generally requires a large number of parameters in training, an issue which may cause over-fitting. It was not until the invention of the convolutional neural network that deep neural networks became widely used for image processing.

2.6.3 Convolutional neural network

Inspired by the fact that in the visual cortex, certain neurons only respond to a small visual field and the receptive fields overlap with each other [89–91], the convolutional neural network (CNN) was proposed [92]. Compared with the fully connected neural network, CNN reduces the number of parameters to train because only the activations of nearby neurons' are summed together. This operation is usually referred to as convolution [93]. To understand the convolution operation, we start with a simple one-dimensional example. Given two functions $x(t)$ and $f(t)$, the convolution operation can be written as follows:

$$s(t) = \int x(\omega) f(t - \omega) d\omega \tag{2.44}$$

Since we deal with discrete data (pixels), the index t is an integer in convolution operations:

$$s(t) = (x * w)(t) = \sum_{\omega} x(\omega) f(t - \omega) \tag{2.45}$$

Extending the convolution operation to two-dimensional space is straightforward. We illustrate a convolution operation process for image data in Figure 2.17.

Similar to the feedforward neural network, a bias vector and a non-linear activation function are applied after the convolution operation to form the convolution layer. We can write this formally by rewriting Equation 2.37 as:

$$\mathbf{y}_l = \Phi_l \left(\mathbf{w}_l * \mathbf{y}_{l-1} + \mathbf{b}_l \right) \tag{2.46}$$

where $*$ is the convolution operation. From this equation, we can observe that the convolution layer is similar to the adaptive filtering in signal processing where the weights of the convolution filters are learned by the backpropagation algorithm and the number of convolution filters are usually much larger.

Another common operation in CNN is pooling, which usually occurs after the convolution layer. The pooling operation replaces the original signal with a downsampled version. For example, the max-pooling operation extracts the maximum output within a rectangular neighborhood [94]. The intuition behind pooling is to subsample the signal for reducing the complexity of the system while maintaining the main information of the signal. By stacking convolutional layers and pooling operations, we can obtain different CNN architectures.

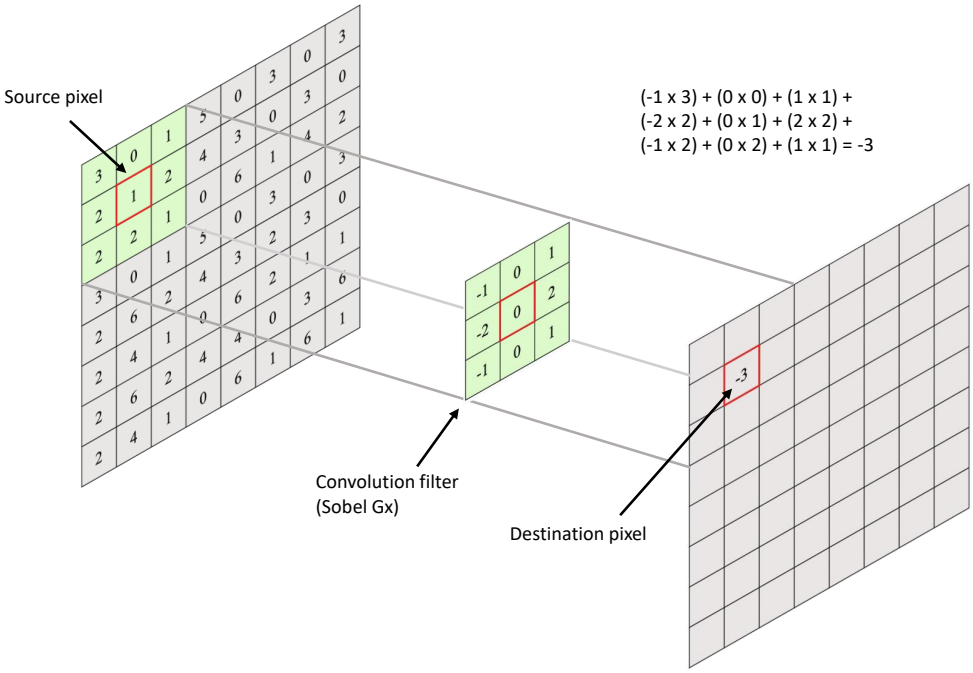


Figure 2.17: Convolution operation.

2.6.4 Batch normalization

Batch normalization is a method for normalizing inputs to layers to accelerate neural network training [95]. It has become almost a default choice for implementing deep neural networks. Recently, theoretical and empirical studies have shown that batch normalization can avoid the problem of diverging loss for particularly deep neural networks by constantly correcting layer inputs to be zero-mean and of unit standard deviation [96].

Although batch normalization enables larger gradient steps resulting in faster convergence rate, it is not suitable to train datasets that contain images with diverse luminance ranges. We will use batch normalization in Chapter 4 for training IVMs with the fixed display brightness. However, it will not be used in Chapter 5 when training IVMs with varying viewing conditions because batch normalization may shift the peak luminance of images and prevent the neural network from predicting visibility maps correctly at different display brightnesses.

Algorithm 2 Batch normalization algorithm

1: **Input:** $\mathbf{x}$: a mini-batch of input data; γ: scale parameter to be learned; β: shift parameter to be learned; ϵ: a small number to avoid dividing by zero.
2: Compute the minibatch mean and variance:

$$\mu = \frac{1}{m}\sum_{i}^{m} x_i \tag{2.47}$$

$$\sigma^2 = \frac{1}{m}\sum_{i}^{m}(x_i - \mu)^2 \tag{2.48}$$

3: Normalize data:

$$\mathbf{x} = \frac{\mathbf{x} - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{2.49}$$

4: Scale and shift:

$$\mathbf{y} = \gamma\mathbf{x} + \beta \tag{2.50}$$

2.7 Summary

In contrast to previous works on IVMs and visually lossless image compression, we will conduct research in the following directions:

1. Existing IVMs are based on a simplified version of human visual system models and are calibrated with small datasets. Different from using this white-box modeling approach, we plan to explore using machine learning methods in a black-box or hybrid way to improve the prediction performances of IVMs because machine learning has become the driving force for image quality assessment. Details will be discussed in Chapter 4 and Chapter 5.

2. The majority of previous research on visually lossless image compression is content and encoder dependent. Different from this content and encoder dependent approach, we plan to improve machine learning-based IVMs to automatically determine the compression setting to achieve visually lossless image compression, making our proposed visually lossless image compression method content and encoder independent. Details will be discussed in Chapter 6.

CHAPTER 3

Data collection

To study visibility and visually lossless image compression, we first need to collect datasets for training and evaluating our methods. There are two kinds of datasets related to our research: (1) visibility datasets that measure the visibility map and, (2) a visually lossless image compression dataset that measures the visually lossless compression threshold. Collection of visibility datasets (LocVis and LocVisVC) was done at the Max Planck Institute and the University of Cambridge. We also generated a large visibility dataset, marked by the traditional visibility metric HDR-VDP, to increase the number of training samples in the University of Cambridge (details will be explained later). Collection of the visually lossless compression dataset (VLIC dataset) was done at the University of Cambridge. We will introduce the collection of the visibility datasets and the visually lossless image compression dataset respectively. The LocVis dataset is available at `https://doi.org/10.17863/CAM.21484`. The LocVisVC dataset is available at `https://doi.org/10.17863/CAM.37996`.

3.1 Visibility dataset with fixed viewing conditions (LocVis dataset)

In this section, I will describe the visibility dataset that we used for training and evaluating visibility metrics. The previously largest visibility metric collected contains only 30 images with 216x216 pixels each [97]. Besides, the experimental procedure in [97] is not very efficient and takes several hours. We refine the procedure of collecting data from [98] to obtain the largest dataset of locally visible distortions. To have more images for training, we also include the TID2013 quality dataset [99] with automatically generated markings.

3.1.1 Stimuli

The dataset consists of 557 images with 170 unique scenes (261 images are marked by humans and 296 images are automatically generated). Many of them are generated for up to 3 distortion levels, for example, different quality settings of image compression. The scenes are selected to cover many common and specialized computer graphic artifacts such as noise, image compression, shadow acne, peterpanning, warping artifacts from image-based rendering methods, and deghosting due to HDR merging. This variety makes our data challenging for existing visibility metrics. The images used in our dataset come from many previous studies. We organize them into the following subsets. MIXED (59 images) is an extended version of the localized computer graphics artifacts dataset (LOCCG) from [98] where we generate images at several distortion levels by blending or extrapolating the difference between the distorted and the reference images. The distortions include high-frequency and structured noise, virtual point light (VPL) clamping, light leaking artifacts, local changes of brightness, aliasing and tone mapping artifacts. PERCEPTION (34 images) from [100] is artificial patterns designed to expose well known perceptual phenomena, such as luminance masking, contrast masking, and contrast sensitivity. Datasets ALIASING (22 images), PETERPANNING (10 images), SHADOWACNE (9 images), DOWNSAMPLING (27 images) and ZFIGHTING (10 images) are derived from [101] and contain real-time rendering artifacts. Those images were created using popular game engines (i.e. Unreal Engine 4, Unity) and they contain both near-threshold (e.g. aliasing) and supra-threshold distortions (e.g. z-fighting, peter-panning). COMPRESSION (71 images) contains distortions due to experimental low-complexity image compression, operating at several bit-rates. This set is an important source of near-threshold distortions. DEGHOSTING (12 images) contains artifacts due to HDR merging, which exposes the shortcomings of popular deghosting methods [102]. IBR (36 images), and CGIBR (6 images) contain artifacts produced by view-interpolation and image-based rendering methods, which come from [68]. TID2013 (261 images) contains a subset of images from TID2013 image quality dataset [99] in which images were selected so that the distortions are visible in the entire image (the entire marking map set to 1), or are invisible (the entire marking map set to 0).

3.1.2 Generating TID2013 visibility dataset

In addition to 296 newly marked images, we added 261 images from the TID2013 image quality dataset [99], for which we could automatically generate marking. We selected from that dataset a subset of images that did not contain noticeable differences and assigned them marking maps set to 0s (no user markings). Then we selected another subset with well-noticeable distortions and set corresponding marking maps to 1s (distortions visible in the entire image). To ensure that both subsets were correctly selected, we

compared the four least severe distortion levels with the reference images in an additional pairwise comparison experiment (comparisons missing in the original dataset) and scaled the original (per-observer) pairwise data together with additional measurements using methods described in [103] and assuming Thurstone Case V observer model. Then, we selected for the first subset the images with the score of less than 0.2 just-objectionable-difference (JOD) to the reference, and for the second subset the images with the difference larger than 3 JODs. We also excluded the distortion types that affected only small image regions, such as JPEG transmission errors, and left the distortions that affected all pixels.

The aim of the experiment is to collect data on distortion visibility in each image location. This serves to distinguish between distortions that are below the visibility threshold and cannot be detected and those that are well visible. Such visibility thresholds are typically collected in threshold experiments, using constant stimuli, adjustment or adaptive methods, which can measure a single image location at a time, making such procedures highly inefficient. For example, the largest dataset collected using such methods [97] contains just 30 images, 216x216 pixels each, and it required tens of experiment hours to collect it. Instead, we refined the procedure from [98] to obtain the largest dataset of local visible distortions. In addition, we also included images from the TID2013 quality datasets with automatically generated markings, as described in the supplemental material. The summary of the dataset is shown in Table 3.1 and examples of selected images are shown in Figure 3.1.

More details about all dataset categories can be found in the supplemental material of our paper published in ACM Transaction on Graphics [104]. Next, we will introduce the experiment procedure and apparatus.

Subset name	**Scenes**	**Images**	**Distortion levels**	**Level generation method**	**Peak luminance**	**ppd**
MIXED	20	59	2-3	blending	110 cd/m^2	40
PERCEPTIONPATTERNS	12	34	1,3	blending	110 cd/m^2	40
ALIASING	14	22	1-3	varying sample number	110 cd/m^2	40
PETERPANNING	10	10	1	n/a	110 cd/m^2	40
SHADOWACNE	9	9	1	n/a	110 cd/m^2	40
DOWNSAMPLING	9	27	3	varying shadow map resolution	110 cd/m^2	40
ZFIGHTING	10	10	1	n/a	110 cd/m^2	40
COMPRESSION	25	71	2-3	varying bit-rates	110 cd/m^2	60
DEGHOSTING	12	12	1	n/a	100 cd/m^2	60
IBR	18	36	1,3	varying key frame distances	110 cd/m^2	40
CGIBR	6	6	1	n/a	110 cd/m^2	40
TID2013	25	261	n/a	n/a	100 cd/m^2	40

Table 3.1: The subsets of LocVis dataset used for training.

3.1.3 Experimental procedure and apparatus

In this section, I will present our experimental procedure for marking visible distortions.

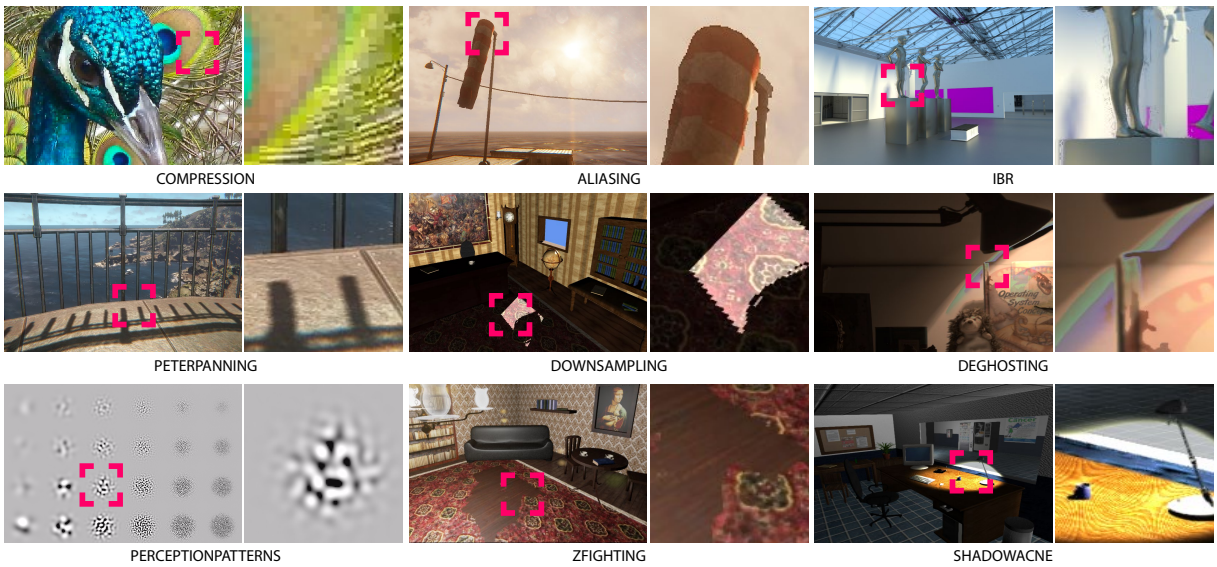


Figure 3.1: The figure presents examples of stimuli from our dataset. The insets show the closeup of the artifacts. For the full preview of the image collection please refer to the supplemental materials.

Comparison method Visibility of image differences can be measured using different presentation methods, such as flickering between distorted and reference images, the same with a short blank screen in between, a side-by-side presentation, or no-reference presentation [98]. Different presentation methods will result in different levels of sensitivity to distortions. Observers are extremely sensitive to differences in flicker presentation, resulting in overly conservative estimates of visible differences for most applications, in which a reference image is rarely presented or available. For that reason, we opted for side-by-side presentation, which is also more relevant to many graphics applications such as visually lossless image compression as achieving the visually lossless difference in side-by-side comparison is already enough for most scenarios in real practice.

Experiment software For the purpose of collecting training data, we designed a web application for marking visible distortions. To increase the comfort and accuracy of marking, we provided the ability to change brush size, erase, clear all marking. Figure 3.2 depicts the application layout.

Figure 3.3 shows a sample scene with three distortion levels and the corresponding observer markings.

Viewing conditions The experimental room had dimmed lights, and the monitor was positioned to minimize screen reflections. The observers sat 60 cm from a 23″, 1920×1200 resolution Acer GD235HZ display, resulting in the angular resolution of 40 ppd. The measured peak luminance of the display was 110 cd/m^2 and the black level was 0.35 cd/m^2. For COMPRESSION and DEGHOSTING sets, the distance was changed to the

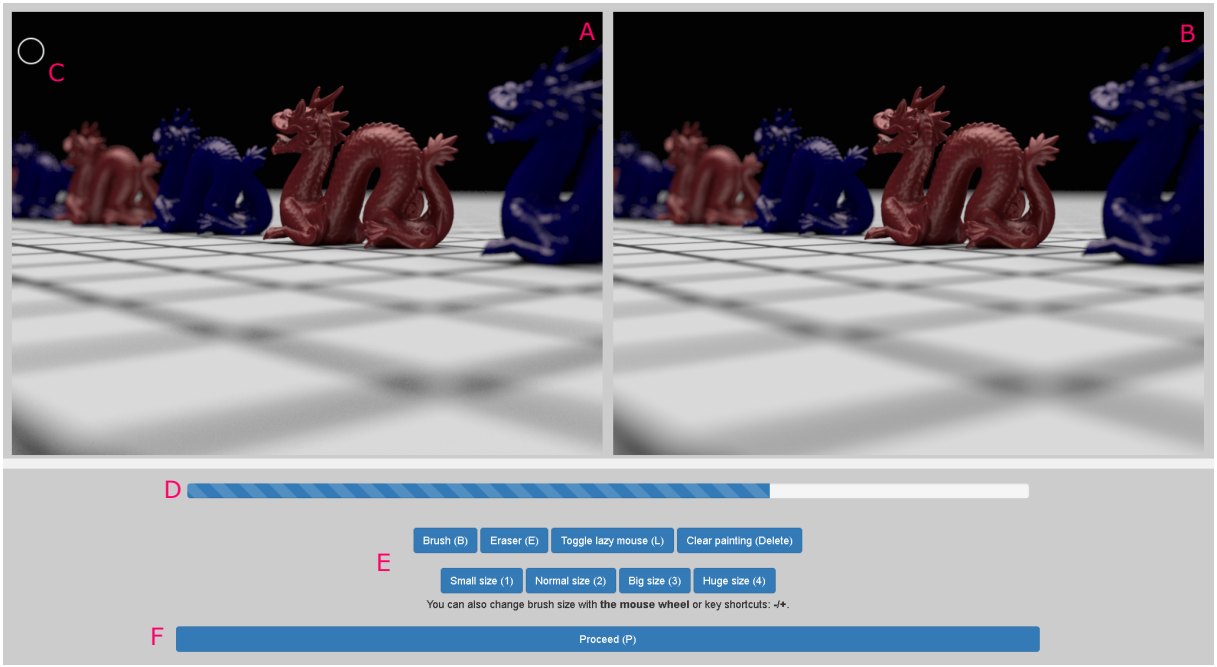


Figure 3.2: Layout of the custom application for marking visible distortions: A) distorted and B) reference images, C) brush cursor, D) progress bar, E) setting buttons, and F) proceed button.

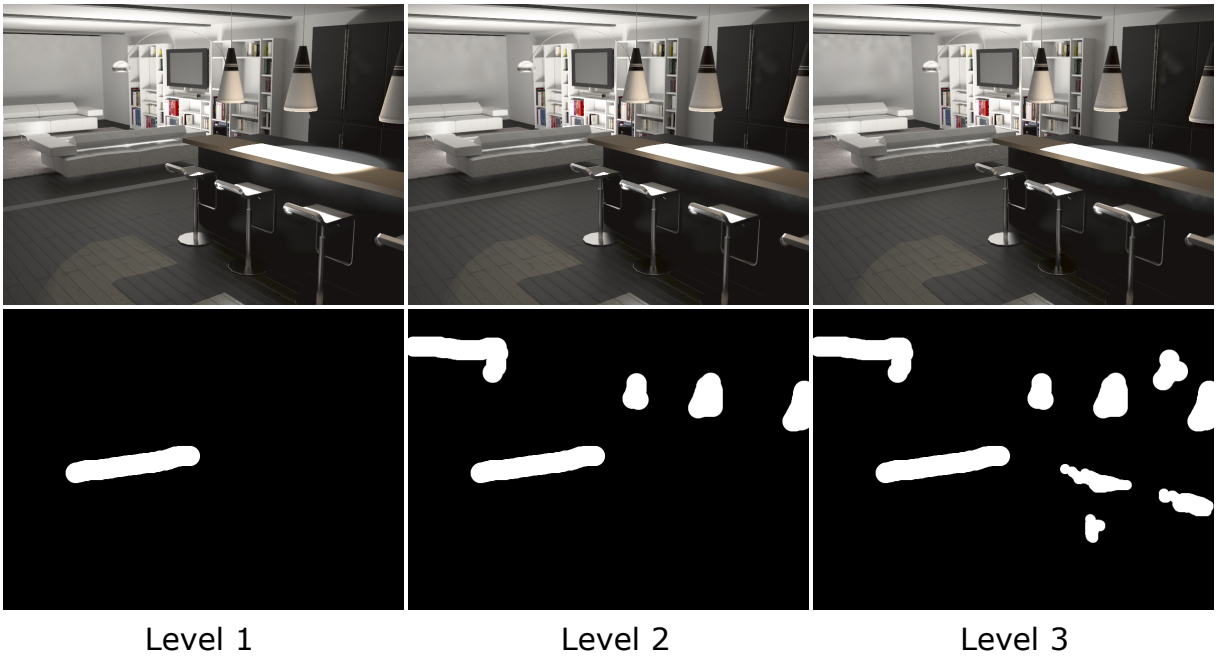


Figure 3.3: An example scene with three levels of distortion magnitude (top row), and the corresponding distortion markings (bottom row). The distortion level increases from left to right, which results in adding newly marked regions.

one corresponding to 60 ppd to reduce the visibility of distortions.

Observers Different groups of observers were asked to complete each subset of the dataset. At least 15 and at most 20 observers completed each subset. In total, 46 observers

(age 23 to 29) were recruited from computer science students and researchers. The observers were paid for their participation. All observers had normal or corrected-to-normal vision and were also naive to the purpose of the experiment. To reduce the effect of fatigue, the experiment was split into several sessions, where each session lasted less than one hour. The post-experiment interviews indicated that the session length was acceptable and did not cause excessive fatigue.

3.2 Visibility dataset with varying viewing conditions (LocVisVC dataset)

Visibility changes greatly with varying viewing conditions. Figure 3.4 illustrates the trends of the visibility change under different luminance and distance conditions. With the increase of luminance and the decrease of ppd (decreasing ppd is equivalent to decreasing distance), distortions become more visible, which agrees with empirical observations and previous research [1]. This confirms the need for a visibility dataset that takes account of both absolute luminance and a viewing distance. To collect LocVisVC dataset, we used the same experiment software and protocol as for collecting LocVis dataset but under different viewing conditions:

3.2.1 Stimuli

We selected parts of LocVis dataset's stimuli and measured their visibility under varying viewing conditions. As the manually labeled datasets were insufficient for training, we also prepared a dataset with synthetic labels, generated with the HDR-VDP visibility metric. We used 200 high-quality photographs obtained directly from camera RAW files. All photographs were resized to the maximum resolution of 1920×1080. The images were then distorted by encoding and decoding using JPEG[1] and WebP[2] image compression at the quality settings of 20, 50 and 90. We then randomly selected 50 images as the base scenes for our dataset. Each of these images was converted into linear colorimetric units using the display model assuming the peak luminance of 10 cd/m^2, 110 cd/m^2, and 220 cd/m^2. The visibility map for these images was then predicted for the angular resolutions of 30, 40, 50 and 60 ppd, producing in total 600 labeled images. A summary of the dataset can be found in Table 3.2.

[1] https://github.com/LuaDist/libjpeg
[2] https://developers.google.com/speed/webp

Subset name	Scenes	Images	Distortion levels	Level generation method	Peak luminance	ppd
MIXED	20	59	2-3	blending	110 cd/m^2	40
PERCEPTIONPATTERNS	12	34	1,3	blending	110 cd/m^2	40
ALIASING	14	22	1-3	varying sample number	110 cd/m^2	40
PETERPANNING	10	10	1	n/a	110 cd/m^2	40
SHADOWACNE	9	9	1	n/a	110 cd/m^2	40
DOWNSAMPLING	9	27	3	varying shadow map resolution	110 cd/m^2	40
ZFIGHTING	10	10	1	n/a	110 cd/m^2	40
COMPRESSION	25	71	2-3	varying bit-rates	110 cd/m^2	60
DEGHOSTING	12	12	1	n/a	100 cd/m^2	60
IBR	18	36	1,3	varying key frame distances	110 cd/m^2	40
CGIBR	6	6	1	n/a	110 cd/m^2	40
TID2013	25	261	n/a	n/a	100 cd/m^2	40
VIEWCOND	26	**264**	1-3	n/a	10, 200 cd/m^2	30, 60
PRETRAIN	200	**600**	3	JPEG and WebP compression	10, 110, 200 cd/m^2	30,40,50,60

Table 3.2: The subsets of LocVisVC dataset used for training. VIEWCOND is the newly measured LocVisVC dataset. PRETRAIN is the HDR-VDP generated synthetic dataset for pre-training. The other sets are from the original LocVis dataset.

3.2.2 Experimental procedure and apparatus

For collecting this dataset, we used the side-by-side presentation as in the process of collecting LocVis dataset. Observers were asked to paint freely all the visible distortions using a custom painting interface. To speed up the process and to increase the coherency of collected data, multiple levels of distortion magnitude proposed in [104] were used.

Display and viewing conditions The experiment took place in a room with dimmed lights. The display was positioned to minimize screen reflections. The images were shown on a 23″, 1920 × 1200 pixels resolution Acer GD235HZ display set the sRGB color profile. The screen was calibrated using a Minolta LS100 luminance meter to two different peak luminance conditions: 10 cd/m^2 and 220 cd/m^2. To achieve the luminance of 10 cd/m^2, the display was dimmed and a 0.6 Neutral Density (ND) filter, reducing the light by a factor of 4, was put on the screen. These two setups cover the luminance range found in most of the displays [3]. The observers viewed the display at two distances, 40 cm and 86 cm, which correspond to angular resolutions of 30 and 60 ppd.

Observers In total, 46 observers, aged between 23 and 29 years old, were recruited among computer science and other field students. All observers were paid for their participation and had normal or corrected-to-normal vision. They were naïve about the purpose of the experiment. To reduce the effect of fatigue, the experiment was split into several sessions, where each session lasted less than one hour.

[3] https://www.laptopmag.com/benchmarks/display-brightness

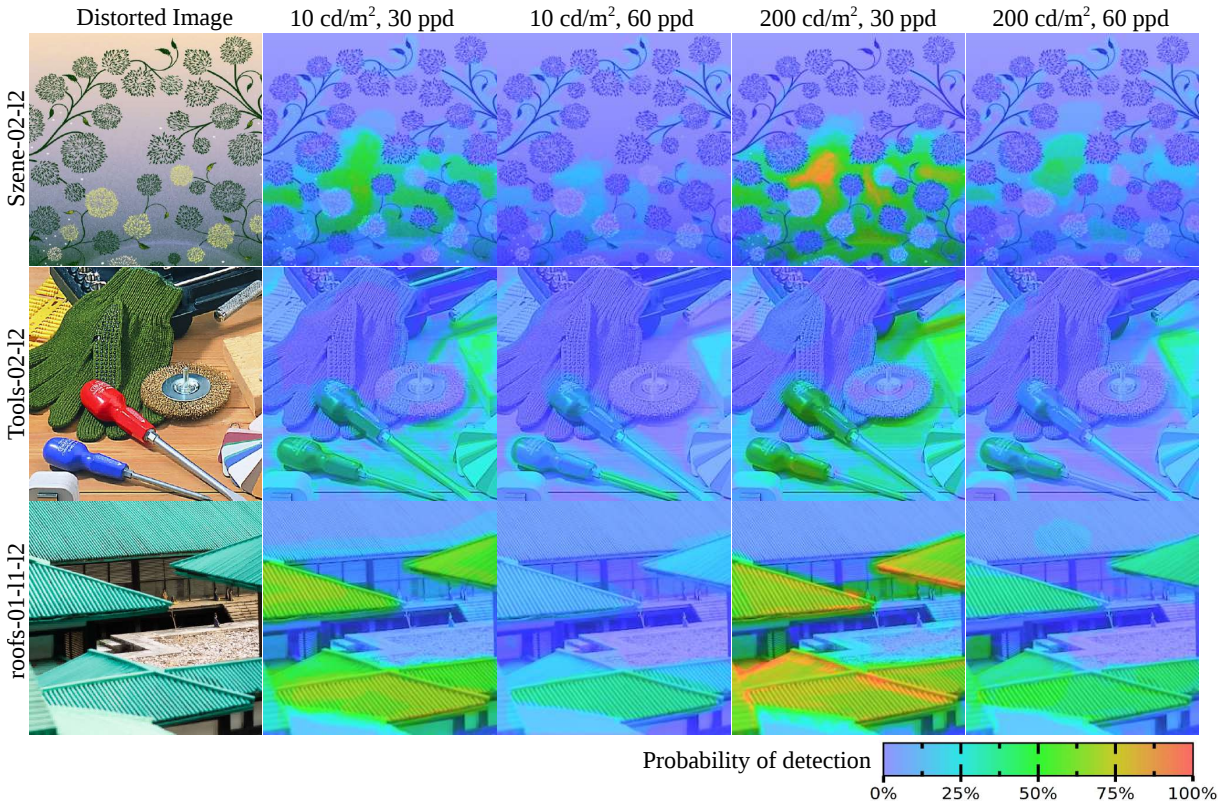


Figure 3.4: Examples of images and subjective data from LocVisVC dataset. Decreasing ppd (decreasing distance between the observer and the display) for the same luminance condition increases the visibility of artifacts. When luminance is increased keeping the same ppd condition the visibility of artifacts also increases.

3.3 Visually lossless image compression dataset with fixed viewing conditions (VLIC dataset)

To evaluate the visibility metric's performance on visually lossless image compression, we collected a visually lossless image compression (VLIC) dataset, containing images encoded with JPEG and WebP[4] codecs.

3.3.1 Stimuli

The VLIC dataset consists of 50 reference scenes obtained from previous studies of image compression and image quality. The stimuli were taken from the Rawzor's free dataset (14 images)[5], CSIQ dataset (30 images) [105], and the subjective quality dataset in [106] (where we randomly selected 6 images from the 10 images in the dataset). For Razor's dataset, images were cropped to 960x600 pixels to fit within our screen. These images provided a variety of contents, including portraits, landscapes, and images of daylight and

[4] `https://developers.google.com/speed/webp/`
[5] `http://imagecompression.info/test_images/`

night scenes. 'For JPEG compression, we used the standard JPEG codec (libjpeg[6]). For WebP compression, we used the WebP codec (libwebp[7]). Half of the reference scenes were compressed using JPEG and the other half using WebP, each into 50 different compression levels.

3.3.2 Experiment Procedure and apparatus

The experimental task was to find the compression level at which observers could not distinguish between the reference images and the compressed images.

Experiment stages The experiment consisted of two stages as shown in Figure 3.5. In the first stage, observers were presented with reference and compressed images side-by-side and asked to adjust the compression level of the compressed image until they cannot see the difference (method-of-adjustment). A 0.5 second long blank with the middle-gray background was displayed when changing compression levels so that observers could not use temporal changes to guide their choice. The compression level found in the first stage was used as the initial guess for the more rigorous 4-alternative-forced-choice procedure (4AFC), used in the second stage. In the second stage, observers were shown 3 reference images and 1 distorted image and asked to select the distorted one (see Figure 3.5). The adaptive Bayesian method—QUEST was used for sampling of compression levels and to find the VLT [107]. We collected between 20 and 30 4AFC trials per participant for each image.

Viewing Condition The experiments were conducted in a dark room. The screen was positioned to minimize screen reflections. The experiment set is shown in Figure 3.6. Observers sat 90 cm from a 24 inch, 1920x1200 resolution NEC MultiSync PA241W display, which corresponded to the angular resolution of 60 ppd. The viewing distance was controlled with a chinrest.

Observers Observers were recruited from the University of Cambridge with normal or corrected to normal vision. All observers were paid for their participation and had normal or corrected-to-normal vision. They were naïve about the purpose of the experiment. We collected data from 19 people aged between 20 and 30 years old. z

Distribution of visually lossless threshold To analyze observers' differences in VLT for image compression, we plot the distribution of VLT of all 50 images in the VLIC

[6] `https://github.com/LuaDist/libjpeg`
[7] `https://github.com/webmproject/libwebp`

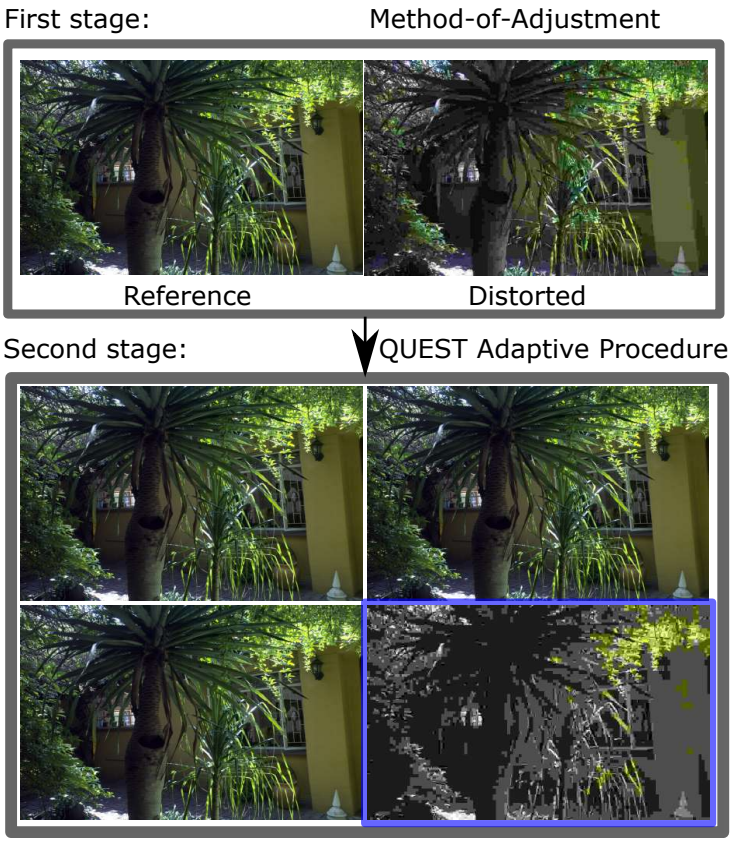


Figure 3.5: Visually lossless compression experiment procedure.

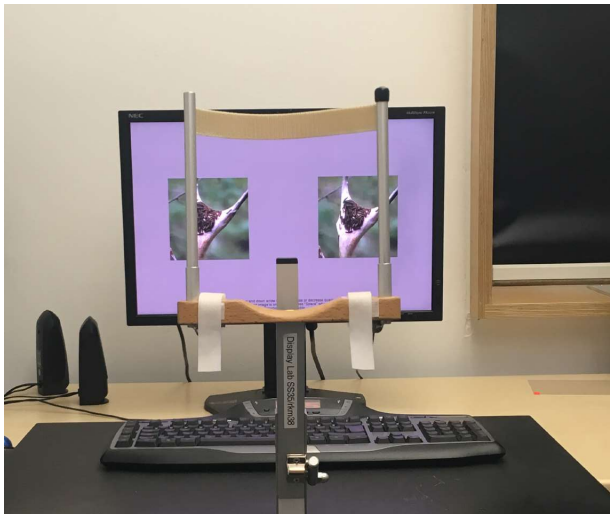


Figure 3.6: Visually lossless compression experiment apparatus (Taken in an environment with adequate lighting for clarity).

dataset in Figure 3.7 and Figure 3.8. From Figure 3.7 and Figure 3.8, we have three observations:

1. We find that VLT varies substantially among different images. This phenomenon indicates that simply setting a fixed compression quality for all images is not ideal. However, this method is widely used in practice. For example, Amazon.co.uk

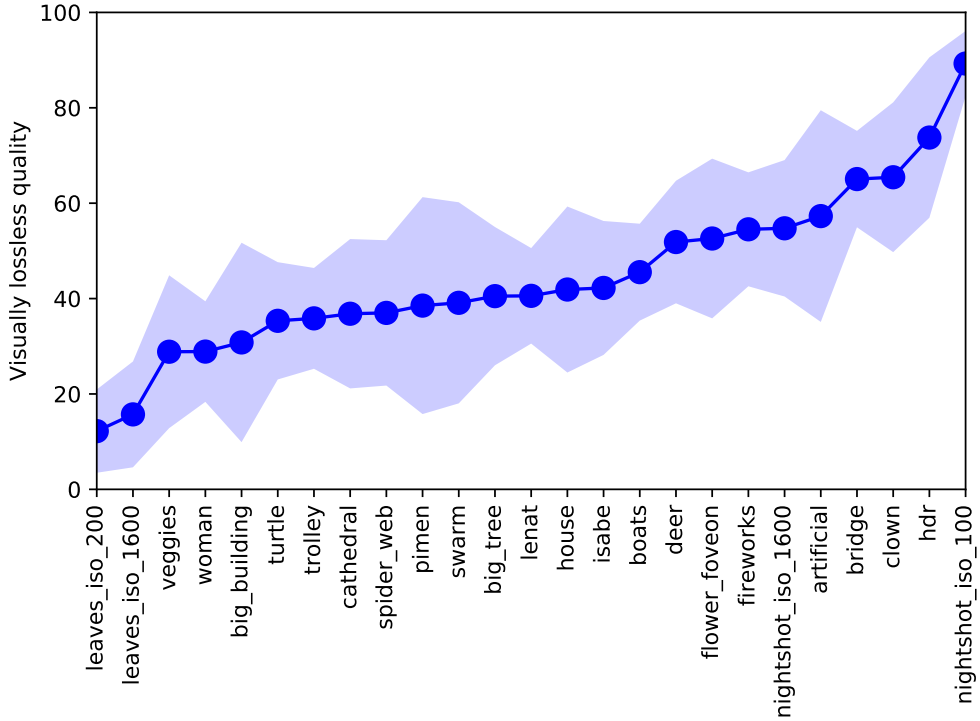


Figure 3.7: Distribution of VLT for JPEG compression. The solid line is the mean of observers' results. Shaded area indicates the range between 20 percentile and 80 percentile of observers' results.

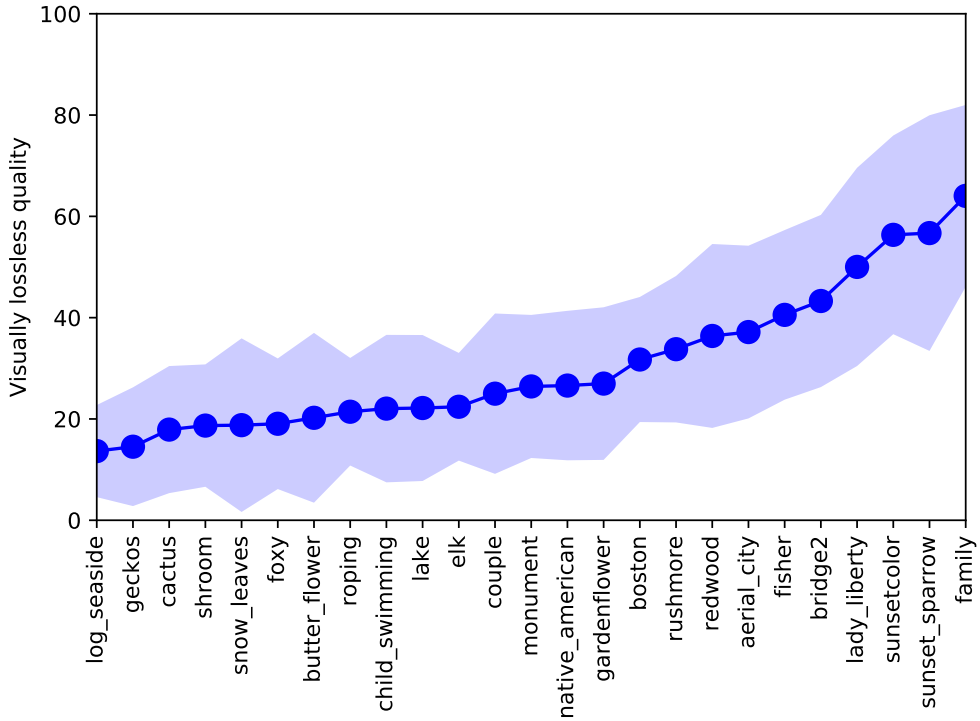


Figure 3.8: Distribution of VLT for WebP compression. The solid line is the mean of observers' results. Shaded area indicates the range between 20 percentile and 80 percentile of observers' results.

compresses all images displayed with JPEG with a quality setting of 75. This setting will result in visible compression artifacts. However, compressing all images at quality 90 will be too conservative and consumes much more transmission bandwidth and storage space than needed.

2. We also observe that for a single image, the VLT for different people ranges greatly, this indicates that different people have distinctive perception thresholds for compression artifacts in images. In this research, we use the mean of VLT as we find out in practice that it gives enough quality for general-purpose visually lossless image compression.

3. The VLT for WebP compression can be much lower than JPEG. This is because WebP uses a predictive encoding technique that is more efficient for compressing more uniform images. However, this phenomenon does not necessarily indicate WebP can save more space than JPEG. We will further compare the performance between JPEG and WebP in Chapter 6.

CHAPTER 4

Predicting visibility under fixed viewing conditions

4.1 Introduction

In this chapter, we will introduce how to derive the visibility metric under fixed viewing conditions using the LocVis dataset. The visibility metric is key in many applications, such as visually lossless image compression, determining the maximum subsampling level for a single image super-resolution and adjusting content-dependent watermarks so that their intensity can be maximized while remain imperceptible. The code of our proposed visibility metric is available at `https://github.com/Chuudy/CNN_visibility_metric`.

My contribution in this chapter This chapter is the result of cooperation between the University of Cambridge and the Max Planck Institute. My contribution is as follows:

1. Proposed using the probability density function of binomial distribution as the loss function which was later improved to the statistical loss function.

2. Substituted the original fully-connected neural network architecture with the deconvolutional neural network, an approach that significantly reduced the number of parameters and made the proposed neural network achieve the state-of-the-art performance for applications. The new architecture was later improved by a coauthor by combining the downsampling layer and the convolutional layer to further improve the performance.

3. Determined the key parameters for training, such as the batch-size, which improved generalization performance.

4. Proposed a framework for using the CNN visibility metric for visually lossless image compression, a framework that would be extended and explained later in Chapter 6.

5. Improved HDR-VDP's performance by introducing a Gaussian filter for the final output.

For clarity, I will provide a full description of the work in the following sections.

4.2 Probability loss

Before deriving new visibility metrics, we need to have a loss function to correctly reflect how well a visibility metric can perform on predicting visibility maps. This is important both in training visibility metrics and evaluating of visibility metrics.

Firstly, to have a reasonable loss function, we first consider the process of how human mark the distorted areas of images in our experiment. As the experimental data is the result of a stochastic process that is affected by noise, it is not ideal to use the experimental data as the ground truth directly without probability modeling. Besides, during the experiment, observers are also likely to make some mistakes and they sometimes do not pay attention to areas where distortions are clearly visible. Thus, we model the stochastic process of human marking in Figure 4.1. We assume the probability of mistakenly marking an area is p_{mis}, the probability of attending an image location is p_{att}, the probability of detecting a difference is p_{det}. p_{det} is our ground truth that we want to predict accurately. Thus, we have the statistic model for a single observer marking an image location.

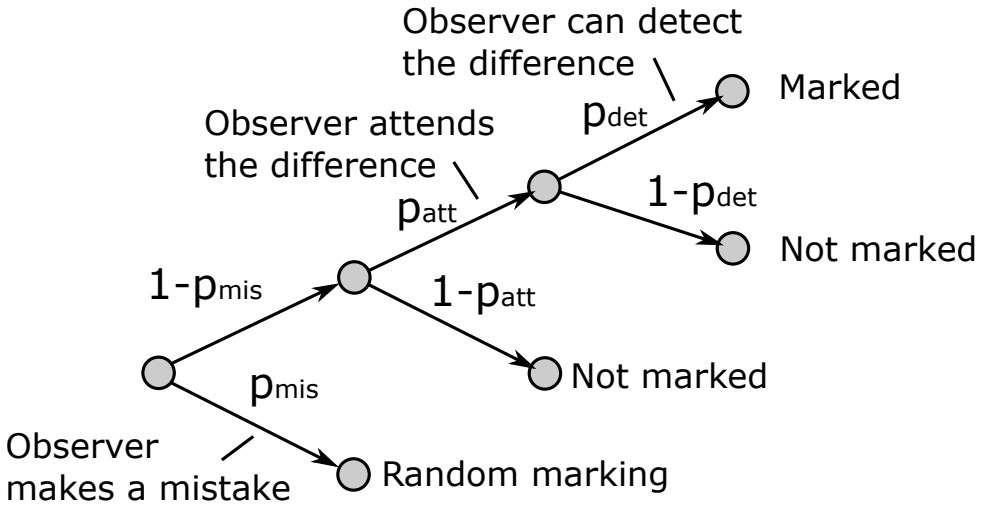


Figure 4.1: The statistical process modeling observed data, given the probability of observer marking a mistake (p_{mis}), the probability of attending (p_{att}) and detecting (p_{det}) differences in images.

As in our experiment, we typically have 10-14 observers for a single image, we need to extend the stochastic modeling to multiple observers. When we have multiple observers

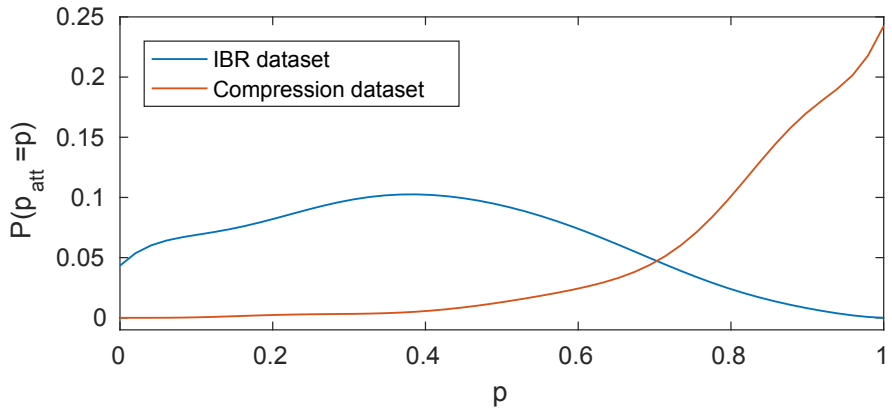


Figure 4.2: The probability that the probability of attending a difference is equal to p, plotted separately for two subsets of LocVis.

marking an image location, we assume that stochastic process of each individual observer is independent and has the same distribution. Then, we can model multiple persons (in which k out of N mark a location) marking based on Bernoulli process with an adjustment for the mistakes:

$$\begin{aligned} P(data) &= p_{mis} + (1 - p_{mis}) \binom{N}{k} (p_{att} \cdot p_{det})^k \, (1 - p_{att} \cdot p_{det})^{n-k} \\ &= p_{mis} + (1 - p_{mis}) \, Binomial(k, N, p_{att} \cdot p_{det}) \, . \end{aligned} \tag{4.1}$$

From the above equations, to infer p_{det}, we need to know p_{att}. However, p_{att} is different across different observers, distortion types, and images. Therefore, p_{att} is a random variable rather than a constant. Thus, we need to estimate the distribution of p_{att} to get p_{det}. We found p_{att} is mostly dependent on the type of distortion. We estimate the distribution of p_{att} for each subset of LocVis dataset. In every subset of LocVis dataset, there are some largely distorted areas that are not marked by observers. We assume that if the difference is large enough (20/255 in our experiment), the difference is definitely observable if observers attend to the difference. Since this corresponds to $p_{det} = 1$, the p_{att} is distributed as:

$$P(p_{att} = p) = p_{att}(p) = \frac{1}{|\Omega|} \sum_{(x,y) \in \Omega} Binomial(k(x, y), N, p) \, , \tag{4.2}$$

where Ω is a set of all pixels (x,y) with large pixel value differences and $|\Omega|$ is the cardinality of that set. For simplicity, we ignore p_{mis} in the above estimate. An example distribution of p_{att} for two datasets is plotted in Figure 4.2.

Next, we can insert the distribution of $p_{att}(p)$ into our statistical model and aggregate the probability of each location to get the log-likelihood for the whole image (note that

this step is similar to the probability summation for HDR-VDP in Section 2.3.2):

$$L = \sum_{(x,y)\in\Theta} \log(p_{mis} + (1 - p_{mis}) \cdot \int_0^1 p_{att}(p) \cdot Binomial\left(k(x,y), N, p_{att}(p) \cdot p_{det}(x,y)\right)\, dp\Big), \tag{4.3}$$

where Θ is the set of all pixels with coordinates x, y. The second line of the equation is the expected value of observing the outcome given the distribution of p_{att}. Equation 4.3 gives a probabilistic loss function, which we use when fitting the visibility models.

To understand the importance of probability of attending modeling (p_{att}), we computed the distribution of p_{att} for IBR dataset and compression dataset and simulated the expected likelihood (integral in Equation 4.3) for a single-pixel location instead of the whole image for simplicity. We assume the total number of observers is 20 and the expected likelihood with regard the number of observers marking are shown in Figure 4.3. From the upper plot in Figure 4.3, we can see that for IBR dataset, if only $k = 10$ out of 20 observers mark the pixel, the probability of detection can range from around 0.5 to 1 and it is very likely that the probability of detection is above 0.8. This means that for this dataset, the differences are highly likely to be detectable if observers attended to the difference. On the other hand, for compression dataset (lower plot), the probability for $k = 10$ is concentrated around $p_{det} = 0.55$, which means that the differences are well attended by observers but still hard to detect.

The probability likelihood function from Equation 4.3 provides a principle way for us to model the experimental data and training and evaluation visibility metrics on large datasets. With the probability likelihood function, we can also take account of uncertainty in the data given a limited number of observers. Next, we will describe the architecture of the proposed deep neural network visibility metric.

4.3 Metric architecture

CNNs have achieved tremendous success in many applications, such as image classification, object detection, and instance segmentation. Inspired by this, we used the basic block of CNNs—convolutional layer (described in section 2.6.3) to construct our visibility metric. As previously mentioned in Section 2.1, full-reference image quality metrics are mostly related to visibility metrics. We found that the Siamese CNN performed well for image quality metrics. For example, Bosse *et al.* . [108] use a Siamese CNN to transform the reference patch and distorted patch into latent space and then using the difference of latent space with the fully-connected layers to predict per-patch quality. Siamese CNN consists of two identical branches that share weights during training. However, in the experiment,

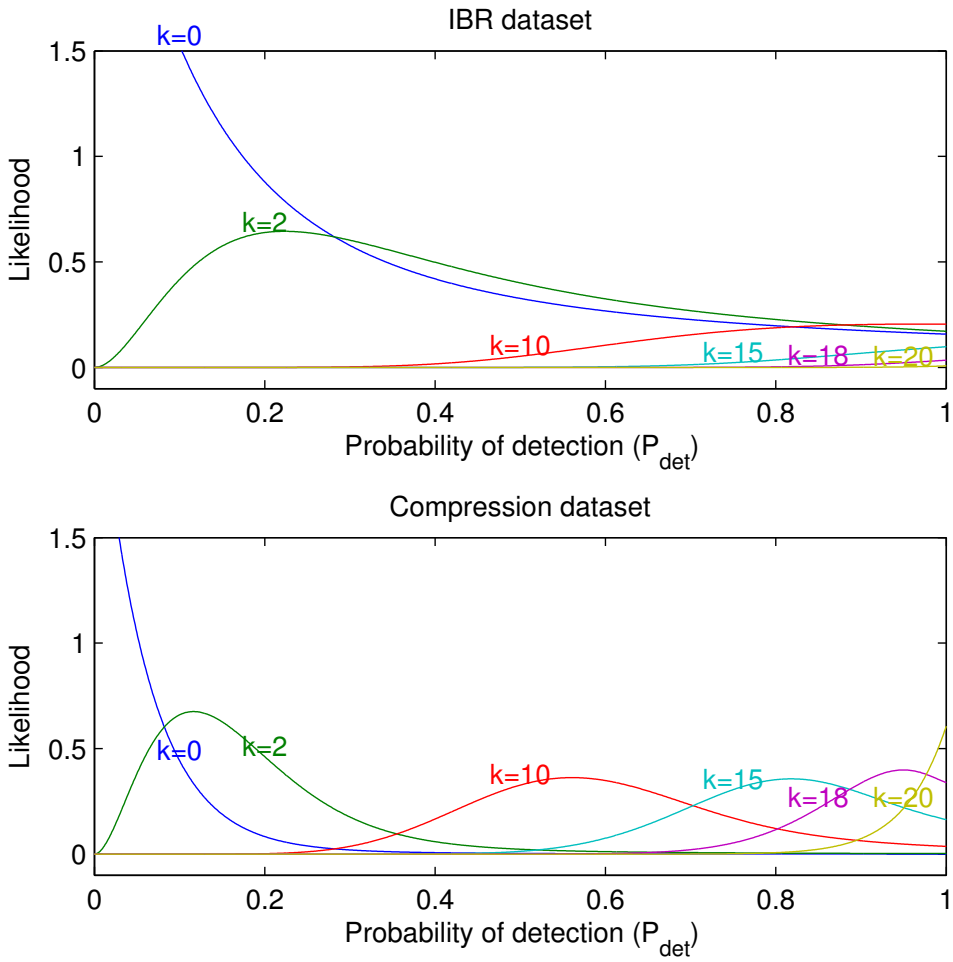


Figure 4.3: The probability of detecting the difference for two datasets.

we find that computing the difference between distorted and reference patch in image space rather than feature space is better for generalization performance. The reason is that different from image quality assessment, the distortions in the visibility dataset are much smaller and closer to thresholds. Computing the difference in feature space will cause information loss and make it harder to predict visibility. Thus, we adopt a non-Siamese architecture for the visibility metric. Besides, we find that the fully-connected layer is not suitable for visibility map prediction. Different from image quality metrics that predict a single score, visibility metrics predict a probability map that has high dimensions. To reduce the number of neural network parameters to train, we use convolutional layers instead of fully-connected layers.

Our metric's architecture is shown in Figure 4.4. Different from Siamese architecture, each branch's weights are not shared with each other. In the upper branch, the difference of distorted patch and reference patch is taken as the input. In the lower branch, the reference patch is taken as the input. After going through convolutional layers' operations, the extracted features of both branches are concatenated together to preserve all features' information. Then, we construct the visibility map from the latent-space representations

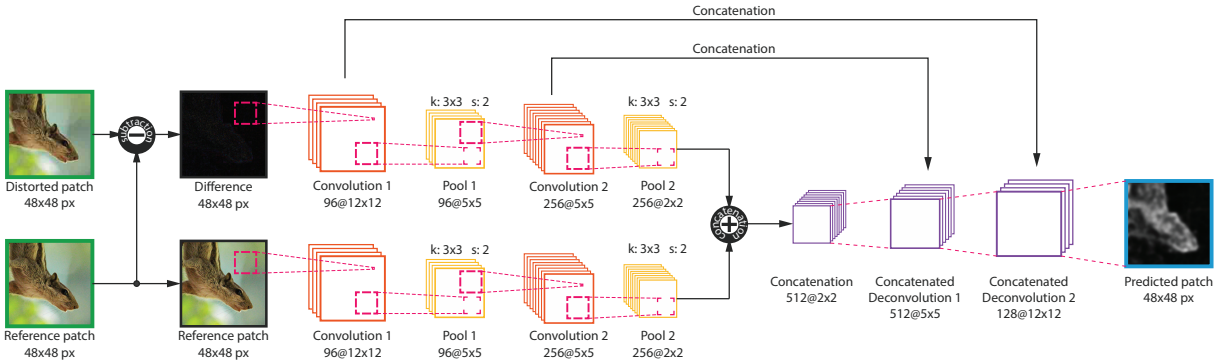


Figure 4.4: Two-branch fully convolutional CNN architecture with the difference branch. The difference branch takes a difference between the distorted and reference images as the input, while the other branch accepts the reference image. The output is a visibility map, achieved by regression, with the same size as the input patch. Each branch contains two convolution layers with 11×11 kernel and stride 4 followed by another layer with 5×5 kernel and stride 1. The deconvolution section uses convolution layers with 3×3 kernel and stride 1.

using 3 deconvolutional layers. For each deconvolutional layer, we use an upsampling operation plus a convolutional layer. Using deconvolutional layer rather than a fully-connected layer improved the ability of the metric to be generalized to different datasets.

To describe this architecture formally, we denote R as the reference patch, D as the distorted patch. The prediction of our metric $M_w(D, R)$ is formulated as:

$$M_w(D, R) = F_{w_{dec}}(Concatenate(F_{w^d_{conv}}(D - R), F_{w^r_{conv}}(R))) \tag{4.4}$$

where $F_{w^d_{conv}}$, $F_{w^r_{conv}}$ and $F_{w_{dec}}$ are mapping functions of two convolutional branches and deconvolutional layers, w^d_{conv}, w^r_{conv}, and w_{dec} are the parameters for these branches. In the following, we will give a more detailed description of convolutional layers and deconvolutional layers:

Convolutional layers Our convolutional layers' architecture is modified from AlexNet implementation [71]. AlexNet has been proven to be an effective architecture for complex tasks, such as ImageNet classification. As the size of our training dataset is limited, we initialize our neural network with AlexNet's weights and then finetune our neural network on the LocVis dataset. In the experiment, we find that the original 5 layer implementation of AlexNet achieved similar results as the first 2 convolutional layers. We then remove the last 3 convolutional layers to reduce the number of parameters to train. In our implementation, the two convolution layers alternate with pooling layers. Rectified linear unit (ReLU) is used as the activation function. Besides, for accelerating the training, we use batch-normalization (described in Section 2.6.4) at the end of pooling layers. To avoid overfitting, we set the dropout value to be 0.5.

Deconvolutional layers We reconstruct the visibility map from the concatenation of features by the deconvolution layers. This can prevent checker-board patterns that appear when we use the transposed operation of convolution. We further improve the performance of our neural network by adding highway connections between the convolutional layers and deconvolutional layers. Such highway connections are known to improve the performance as they can mitigate the problem of vanishing gradient in deeper neural network [109, 110].

4.4 Training

For training the metric, the negative log-likelihood from Equation 4.3 is taken as the loss function. We split images into patches of 48×48 pixels without overlapping. The patch size is determined to preserve high-frequency details but also to be kept small to avoid the curse of dimensions in later layers.

To increase the size of the dataset and prevent overfitting, we do data augmentation by horizontal and vertical flipping the rotations of 90, 180, 270 degrees. We also ignore all the patches for which there is no difference between their distorted and reference versions. The total number of patches is approximately 400,000. To speed up the training process, we use a mini-batch technique with the batch size of 48. We find in the experiment that the batch size matters for better generalization performance, we will explain this phenomenon with a MNIST example in Section 4.5. We use the Adaptive Momentum Estimation (Adam) method for training the neural network. We set the total number of iterations to be 50,000, learning rate to be 0.00001 and decay the learning rate every 2500 steps using the exponential decay function in Tensorflow at a factor of 0.9.

The CNN architecture is implemented in TensorFlow 1.4 [1]. We perform training and testing exploiting Tensorflow GPU support on an NVIDIA GeForce GTX 980 Ti.

To predict a visibility map for a full-size image, we split it into 48×48 patches with 42-pixel overlap, infer a visibility map for each patch and finally assemble the complete map by averaging each pixel shared by the overlapping patches. Prediction for an 800×600 pixel distorted image takes approximately 3.5 seconds.

4.5 Determining the batch-size

In this section, we will analyze the effects of batch-sizes through the lens of stochastic non-convex optimization. Minimizing non-convex error functions over continuous and high-dimensional spaces has been a primary challenge because of a large number of local minima [111]. To optimize the parameters of deep neural networks, the stochastic gradient

[1] `https://www.tensorflow.org/`

descent (SGD) method is widely used for optimizing highly non-convex functions, such as deep neural networks [111, 112]. The SGD computes gradients on random samples of the dataset—a mini-batch instead of computing gradients on the entire dataset. The gradient computed on the mini-batch is often referred to as the stochastic gradient [111, 113] because it is a noisy estimation of the gradient computed on the entire dataset. However, the stochastic noise of the gradient computed on the mini-batch has been shown to improve the neural network generalization performance by helping the optimization process to jump out of local minima [113]. Because this stochastic noise is from mini-batch computation, it is important to determine the right batch-size to achieve better generalization performance.

Intuitively, when batch-sizes become smaller, the gradient will be a noisier estimation of the gradient on the entire dataset because the smaller batch-size gives less information about the entire dataset. To provide some theoretical evidence for this intuition, we denote the parameter to be learned as θ and the loss function of a random sample of the dataset as $f(\theta)$. We denote the batch-size as B and the SGD algorithm can be written as follows [112]:

$$\theta_{t+1} = \theta_t - \eta \nabla \frac{1}{B} \sum_{i=1}^{B} f_i(\theta) \tag{4.5}$$

where t is the number of iterations, η is the learning rate, and $f_i(\theta)$ is the loss computed on i-th data in the mini-batch. $f_i(\theta)$ is a random variable because it is randomly drawn from the dataset. We can approximate the distribution of $f_i(\theta)$ as a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ as in the previous work [113], where μ is the mean and σ is the variance. The variance of the mini-batch loss $\frac{1}{B}\sum_1^B f_i(\theta)$ then can be re-scaled by a factor of B according to the sample variance law for independently identically distributed random variables:

$$\text{Var} = \frac{1}{B}\sigma \tag{4.6}$$

From Equation 4.6, we can conclude that with the decrease of batch-size B, the variance of the stochastic gradients will increase. Zhang *et al.* has shown that the increased magnitudes of stochastic noise in the gradients can help the optimization process to jump out of local minima in non-convex optimization and thus improve the generalization performance [113].

To validate our approximate computation empirically, we compute the test error of the MNIST dataset—a well-known dataset for benchmarking machine learning algorithms with regard to different batch-sizes shown in Figure 4.5. For the experiment, we use the LeNet architecture [92] and set the stochastic gradient descent method with a learning rate of 0.01 and a momentum of 0.5. We run the optimization for 10 epochs and obtain the error on the test dataset of MNIST. From Figure 4.5, we can observe that when the batch-size becomes larger, the test accuracy decreases almost exponentially fast. This example demonstrates

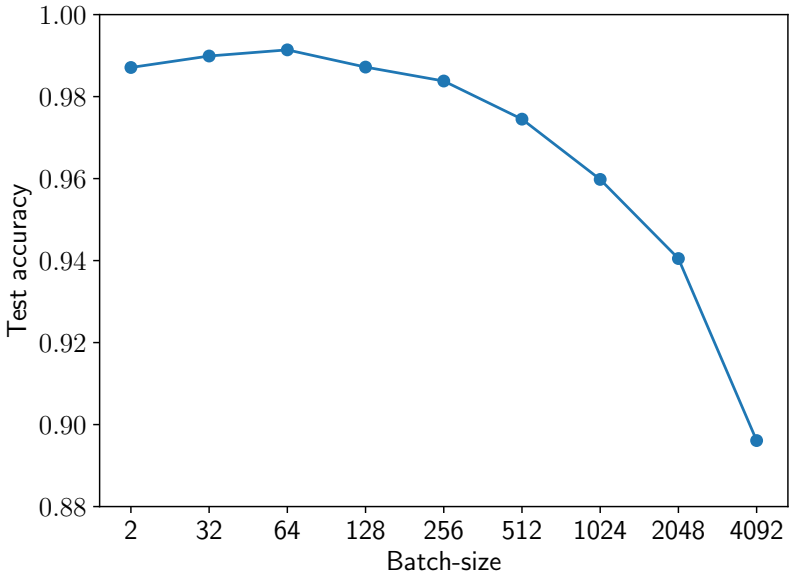


Figure 4.5: Test accuracy of varying batch-sizes.

that the choice of the batch-size has large effects on generalization performances. When the batch-size is excessively large, the magnitude of the stochastic noise will be too small for the optimization process to jump out of local minima. Besides, too large batch-sizes will be impossible because such sizes exceed the graphics processing unit's memory limit. When the batch-size is too small, it will be too noisy for the optimization process to converge to a better solution and too time-consuming because the smaller batch-size requires a larger number of iterations to go over the same dataset. We use the same experiment protocol in Section 4.4 and search for the optimal batch-size. The cross-fold validation results are shown in Figure 4.6. The decreasing trend of LocVis dataset is not as monotonic as MNIST dataset because our neural network is much more complex and harder to optimize. We find the batch-size 48 generally gives us good empirical performance.

4.6 Results

We use 5-fold cross-validation for comparing metrics' performances. For training-test split, we ensured that the testing set did not contain any of the scenes used for training, regardless of the distortion level.

We customized the OpenTuner[2] optimization software to run on cluster computation facilities to train different metrics. In our experiment, we use the absolute difference (ABS), structural similarity index (SSIM), visual saliency induced index (VSI), feature similarity index (FSIM), CIEDE2000, sCIELab, Butteraugli and HDR-VDP metric for

[2] `http://opentuner.org`

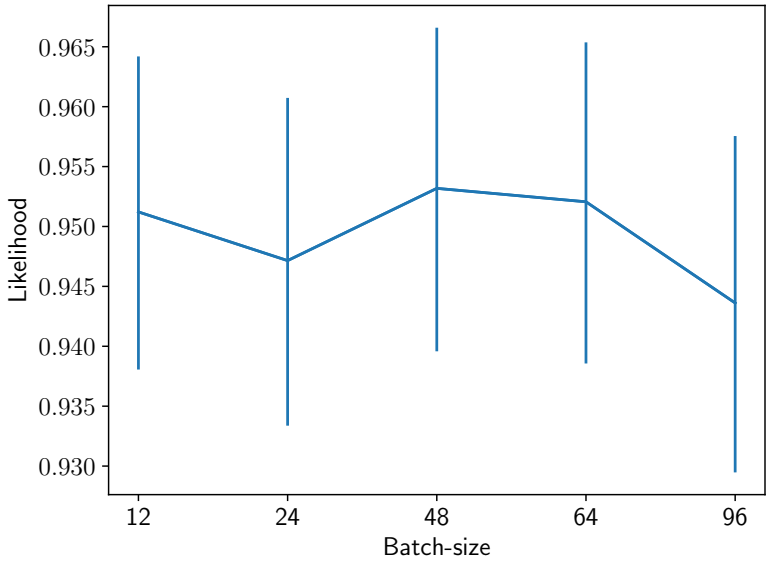


Figure 4.6: Likelihood of varying batch-sizes on the LocVis dataset. (The higher the better)

comparison. For re-scaling the values of metrics' predictions to be in the range of 0-1, we use a psychometric function. We use the probability loss function to train all metrics.

[**ABS**] ABS computes the absolute differences (D) between pixel values. Then, we divide D by the threshold value t and then re-scale it with a psychometric function:

$$p_{det}(x,y) = 1 - \exp\left(\log(0.5)\cdot\left(\frac{D(x,y)}{t}\right)^{\beta}\right) \ , \tag{4.7}$$

where x, y are pixel coordinates. The two parameters to be optimized were t and β. The absolute difference D was computed between luma values of distorted and reference images.

[**SSIM**] As SSIM metric's predictions are negatively correlated with image quality, we have the following constructions to use SSIM to predict for visibility:

$$D_{SSIM}(x,y) = \frac{1}{\epsilon}\left(\log\left(1 - M_{SSIM}(x,y) + \exp(-\epsilon)\right) + \epsilon\right) \ , \tag{4.8}$$

where $\epsilon = 10$ and M_{SSIM} is the original SSIM difference map. The transformation makes the D_{SSIM} values positive, in the range 0–1 and increasing with higher image differences. The $D_{SSIM}(x,y)$ values are then processed by the psychometric function from Equation 4.7.

[**VSI, FSIM**] After transforming the difference maps D_{VSI} and D_{FSIM} into increasing values in the range 0–1, We also apply equation 4.7 for re-scaling. The parameters to be

optimized were psychometric function's parameters t, β, and three parameters of the VSI metric, C_1, C_2, and C_3 (Equations 4–6 in [4]), or respectively two parameters of the FSIM metric, T_1 and T_2 (refer to Equations 4-5 in [5]).

[CIEDE2000] The distorted and reference images were transformed into linear XYZ space assuming Rec. 709 color primaries and using a gain-gamma-offset display model simulating our experimental display. The predicted ΔE were transformed into probabilities using the psychometric function (Equation 4.7). The parameters to be optimized were t and β.

[sCIELab] Our adaptation of sCIELab was identical to the one we used for CIEDE2000, except that the metric was also supplied with the image angular resolution in pixels per visual degree.

[Butteraugli] Butteraugli is an image quality metric proposed by Google [68] based on combining image frequency and luminance features. In the original Butteraugli implementation the threshold for visible distortions is determined by a constant "good_quality". However, we found that this constant does not correlate well with human experiment results, and better results can be achieved if the map is transformed by the psychophysical function from Equation 4.7.

[HDR-VDP] We modified HDR-VDP (v2.2) for better performance. Firstly, we found that orientation-selective bands did not improve predictions for any of our datasets; therefore, we simplified the multi-scale decomposition to all-orientations spatial-frequency bands. Secondly, we improved the spatial probability pooling. The original HDR-VDP was calibrated to detection datasets in which one distortion was visible at a time. This enabled using a simplified spatial pooling, in which all differences in an image were added together. However, this resulted in inaccurate results for our datasets, in which distortions vary in their magnitude across an image. The original pooling was replaced with spatial probability summation

$$P_{sp}(x,y) = \mathbf{1} - \exp\left(\log(1 - P(x,y) + \epsilon) * g_\sigma\right)(x,y) \ , \tag{4.9}$$

where $P(x,y)$ is the original probability of detection map (Equation 20 in [1]), ϵ is a small constant, and g_σ is the Gaussian kernel. The fitted parameters were the peak sensitivity, a self masking factor (`mask_self`), a cross-band masking factor (`mask_xn`), the p-exponent of the band difference (`mask_p`), and the standard deviation of the spatial pooling kernel (`si_sigma`) in visual degrees.

In the following parts, we use the prefix "T-", e. g., T-Butteraugli, to distinguish between the metrics trained by us or in their original version.

The results for all metrics are shown in Figure 4.7. CNN, T-Butteraugli, and T-HDR-VDP are top 3 best performing visibility metrics with CNN outperforms all other metrics. Besides, predicting visibility on some datasets is more difficult (lower likelihood). Notably, COMPRESSION was the most difficult dataset, followed by PERCEPTIONPATTERNS.

We also show a few interesting examples in Figure 4.8 to compare different metrics.

Image *uncorrelated noise* contains three noisy circular patterns modulated by a Gaussian envelope, presented on the background of a lower amplitude noise. T-Butteraugli and CNN performed better as they can ignore the difference in background noise to some extent.

The *gorilla* image was distorted by image compression. As mentioned before, COMPRESSION is the hardest one. The compression distortion contains complex masking patterns, which largely affect the visibility of the distortions. Only more advanced metrics, CNN, T-Butteraugli, and T-HDR-VDP can predict the visibility of the gorilla's face and CNN is even more accurate in predicting the chest area.

The *peter panning* image contains distortion caused by shadow mapping where the shadow is detached from an object casting it [101]. The images also contain small differences in pixel values due to shading and post-processing effects in the game engine. T-FSIM metric failed to mask such small differences (best seen in the electronic version), while other metrics correctly ignore the visibility of those small differences. T-Butteraugli and T-HDR-VDP tend to excessively expand the region with the difference. This is because a simple contrast masking function cannot model human's perception ability correctly. The general approximation ability of CNN makes it able to model complex masking function automatically from data.

The *car* image contains distortion on the body of the car, but there is also a readily visible noise pattern in the bottom right corner. Though the distortions are well visible in the corner, very few people marked the noise pattern as more attention is paid to the car. This is an example that observers' markings cannot be treated as ground truth and we need the statistical model from Section 4.2 to correctly model uncertainty in the data. In this case, all metrics predicted the visibility of the noise pattern on the body of the car correctly.

The *classroom* image contains a rendering of the same scene, but from a slightly different camera position in the distorted and reference images. While the observers could not notice any differences, such pixel misalignment triggered a lot of false positives for most metrics. CNN could only partially compensate for pixel misalignment. However, for applications such as visually lossless image compression, we always have pixels aligned before predicting visibility, which is a common assumption in image quality assessment.

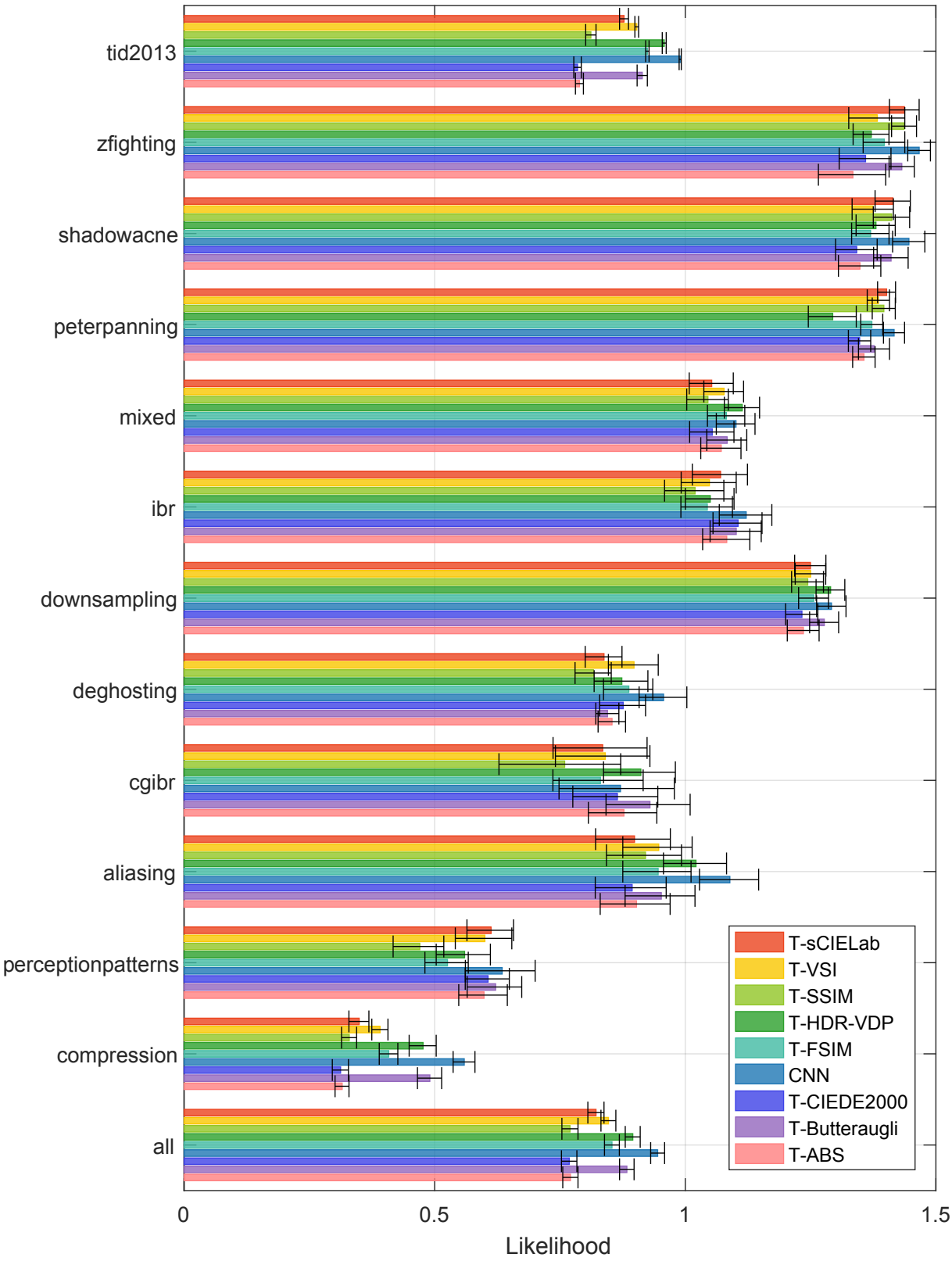


Figure 4.7: The results of cross-validation of the quality metrics for each dataset and for all datasets together. The error bars denote standard error when averaging across images in the dataset.

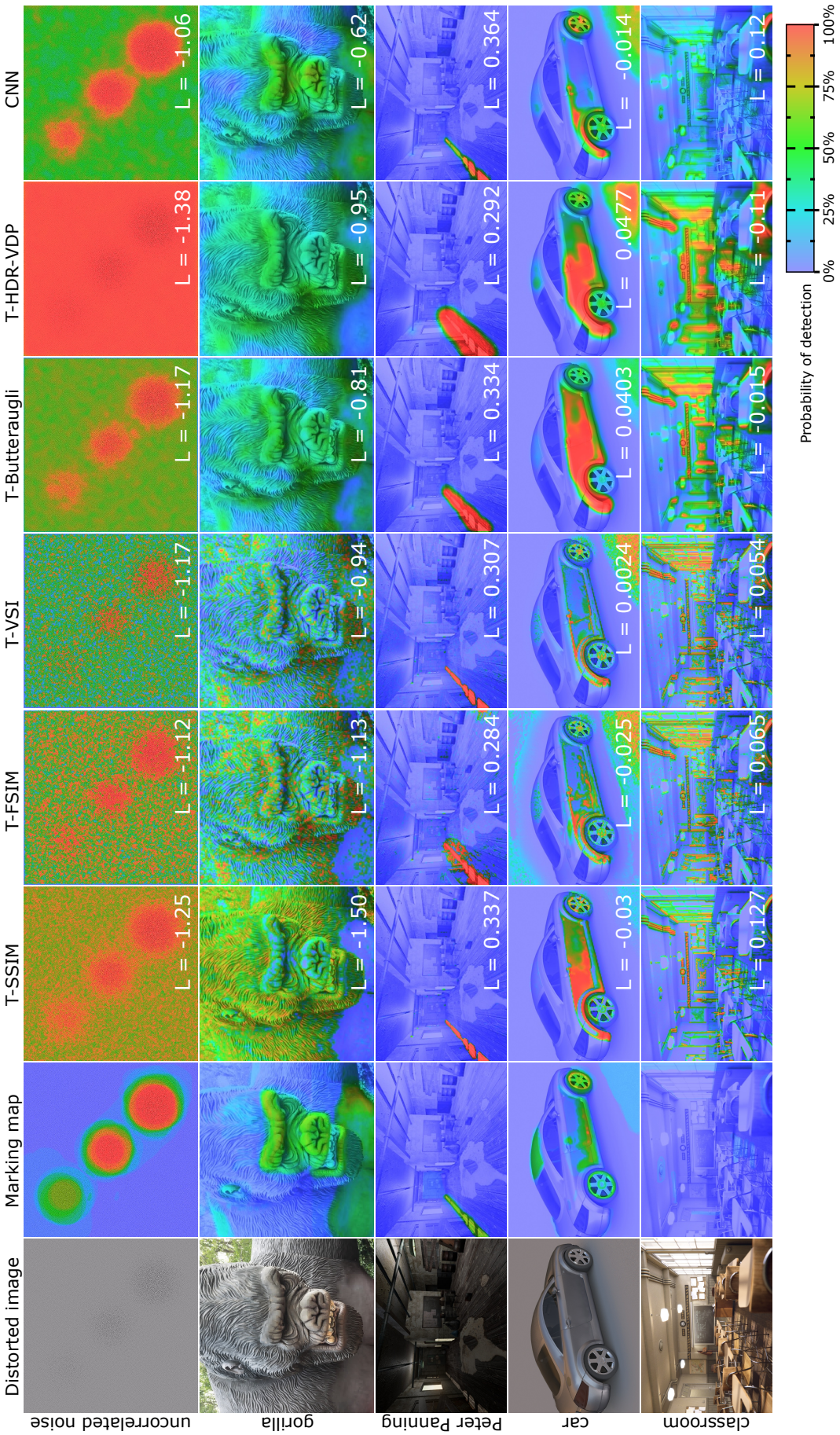


Figure 4.8: Distorted images, observers' markings and metric predictions for a few selected images from the dataset. Metric predictions must be viewed in color.

4.7 Visually lossless image compression

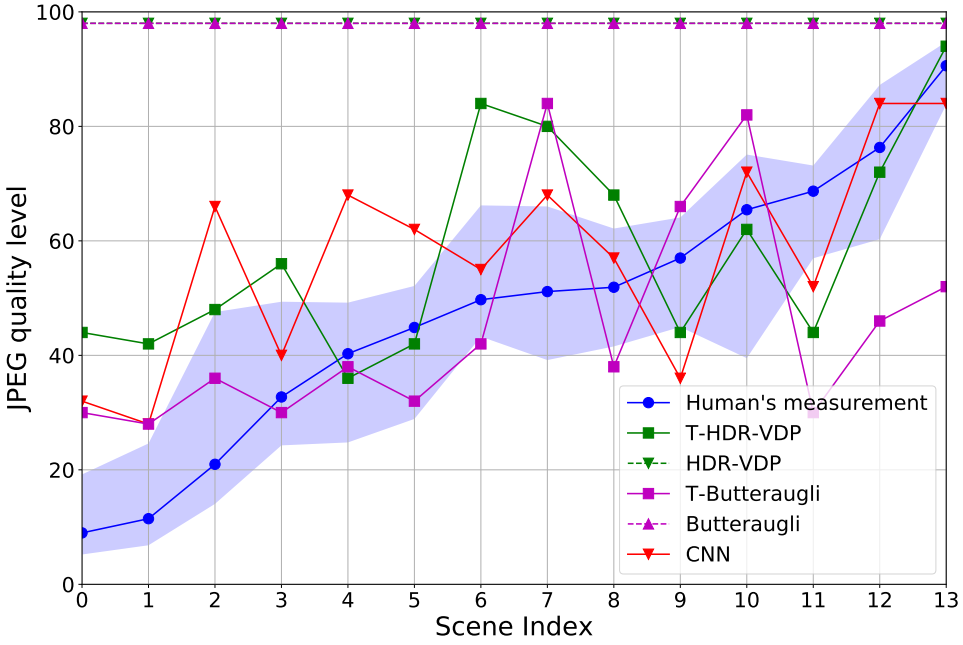


Figure 4.9: Results for visual lossless image compression. The blue line is the median and the blue shaded region is 20th and 80th percentile of manual adjustment.

In this research, we also tried using the visibility metric for visually lossless image compression. To validate metrics' performance in this application, we conducted an additional experiment, in which observers indicated the lowest JPEG quality setting for which distortions remained invisible in a side-by-side presentation. In the experiment, we use the standard JPEG codec (libjpeg[3]). To avoid using the same images for training, we used Rawzor's free dataset[4] which contains a rich set of image content. The images are cropped to 960×600 pixels to fit on our screen. The images are distorted by compression with a standard JPEG codec using a range compression qualities. The experimental procedure involved selecting a distorted image from 4 presented, where only one image was distorted (four-alternative-forced-choice protocol). The quality setting was adaptively adjusted using the QUEST method. Between 20 and 30 trials were collected per image to find the quality settings at which an observer could select a distorted image with 75% confidence probability level in the QUEST procedure. 10 observers completed the experiment. To predict the visually lossless quality setting for JPEG compression, we take the maximum value of the metrics' predicted visibility map, which gives a conservative estimate. A similar approach was used in [68]. We search the quality settings from 0 to 98 and choose the lowest quality setting that produces the visibility map of maximum value

[3] `https://github.com/LuaDist/libjpeg`
[4] `http://imagecompression.info/test_images/`

less than 0.5, which corresponds to 50% of observers spotting the difference. We select the best three metrics, CNN, T-HDR-VDP, and T-Butteraugli, and their original versions, HDR-VDP and Butteraugli, for evaluation.

The results of the experiment and the predictions of the top-performing visibility metrics are shown in Figure 4.9. The blue line denotes the median value computed across observers and the blue shaded area represents the range between the 20th and 80th percentiles. CNN, T-HDR-VDP, and T-Butteraugli correlate reasonably well with the experiment results, although the distortions in scenes 9 and 11, were under-predicted by the trained metrics. The most visible distortions in those images are due to contouring in smoothly shaded regions. Such distortion types were missing in our training set, which could lead to the worse-than-expected performance.

We quantify the accuracy of metrics' predictions as the mean squared error (MSE) between the predictions and levels found in the experiment. Among the top three metrics, CNN's performance is the best with an MSE of 367.7 followed by T-HDR-VDP with an MSE of 467.5 and T-Butteraugli with an MSE of 479.4. The original (untrained) versions of HDR-VDP and Butteraugli resulted in strongly over-predicted visibility of JPEG artifacts. This result confirms that, with our proposed dataset, the trained metrics could generalize well to different distortion types (we did not include JPEG distortions in the dataset) and different content. Compared with the common practice of setting the quality to a fixed value of 90, the best CNN metric could help to reduce file size on average by 60% for the selected set of images. This example demonstrates the potential of using the CNN visibility metric for visually lossless image compression. We will later extend this work in Chapter 6 and much improve the prediction performance on large datasets.

4.8 Summary

In this chapter, we have proposed a deep neural network-based visibility metric that can work very well under the fixed display brightness and viewing distance. The cross-fold evaluation of the proposed visibility metric on the current largest dataset shows that the proposed CNN metric outperforms other state-of-the-art visibility metrics.

In the next chapter, we will introduce how to extend the deep neural network-based visibility metric to different viewing conditions, such as display brightness and viewing distance.

CHAPTER 5

Predicting visibility under varying display brightness and viewing distances

5.1 Introduction

Predicting visibility under viewing conditions is important in many applications. For example, when a user is browsing images on a dimmed display of a mobile phone, image compression distortions are much less visible than when a user is browsing images on a bright mobile phone display. However, the accuracy of existing white-box visibility metrics that can predict visibility under varying viewing conditions, such as HDR-VDP, is often not sufficient. CNN-based black-box visibility metrics have proven to be more accurate, but they cannot take account of differences in viewing conditions, such as display brightness and viewing distance. In this chapter, we propose a CNN-based visibility metric, which maintains the accuracy of deep network solutions and takes viewing conditions into consideration. To achieve these aims, we use the extended version of the LocVis dataset with a new set of measurements, collected considering the aforementioned viewing conditions—LocVisVC dataset (`https://doi.org/10.17863/CAM.37996`). Then, we develop a hybrid model that combines white-box processing stages, modeling the effects of luminance masking and contrast sensitivity, with a black-box deep neural network. We will demonstrate that the novel hybrid model can handle the change of viewing conditions correctly and outperforms state-of-the-art metrics.

Most existing image visibility metrics, such as the Sarnoff Visual Discrimination Model (VDM) [30], VDP [14], and HDR-VDP [1] are white-box models that are designed to model the low-level perception mechanisms of the human visual system. Because of their white-box nature, these models can generalize well to new conditions, such as different

viewing distances or absolute luminance levels. However, because of the limited number of trainable parameters and their complexity, these models cannot be trained to fit complex multi-modal data distributions as effectively as black-box machine learning-based models. In Chapter 4, we have already demonstrated that a CNN-based visibility predictor achieves higher performance than the existing white-box metrics. However, this deep learning solution was trained for and could predict visibility only for a fixed viewing condition: a display with a peak luminance of 110 cd/m^2 and an angular resolution of 40 ppd.

In this chapter, we extend the visibility in Chapter 4 so that the proposed visibility metric can take a range of display brightness levels and angular resolutions into account. We achieve this by combining white-box models of luminance masking and angular resolution resampling with a black-box CNN-based model, based on the architecture from [104]. In the following, we will refer to our proposed visibility metric as deep photometric visibility metric (DPVM). The code of this chapter is available at `https://www.cl.cam.ac.uk/research/rainbow/projects/dpvm/`.

To obtain sufficient data for training our DPVM, we use HDR-VDP[1], an existing white-box visibility metric, to generate predictions for a large number of images affected by JPEG and WebP image compression under different absolute luminance levels and viewing distances. This generated dataset is used to pre-train DPVM. Then, we use a human-labeled dataset to fine-tune the DPVM and validate the results. The human-labeled dataset consists of both existing local visibility dataset (LocVis [1]) and a newly-collected dataset of 264 images labeled under different viewing conditions. The details of the dataset are shown in Section 3.2.

5.2 Metric architecture

Most neural network-based metrics rely on existing architectures, which are trained in an end-to-end manner. In our case, both the viewing distance and the display peak brightness are significant factors that affect predictions. Both parameters could be fed to the network in a standard manner, hoping that the network will learn the correct relationships. However, such a solution requires a large quantity of subjective data, which cannot be easily collected for our task in a reasonable time. To address this challenge, we design a hybrid architecture, in which the viewing distance and the display peak luminance are modeled explicitly as a pre-processing stage of the CNN-based metric. The architecture of the proposed metric and the data pre-processing are illustrated in Figure 5.1 and described in the following sections.

[1] https://www.repository.cam.ac.uk/handle/1810/274368

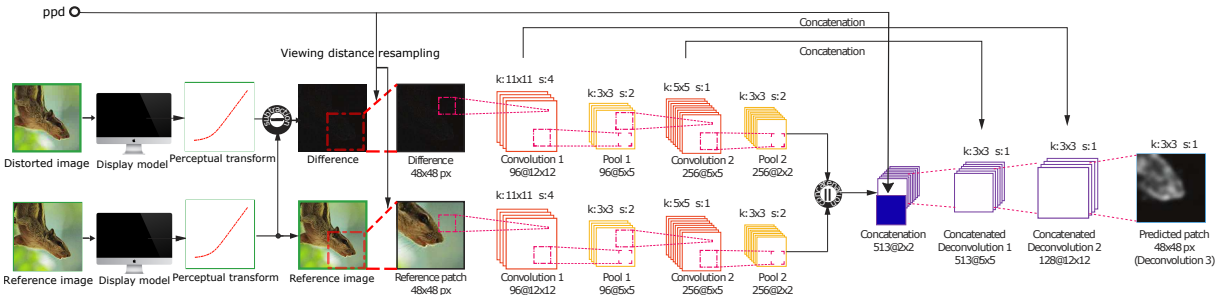


Figure 5.1: CNN visibility metric architecture.

5.2.1 Display model

Since modern displays differ substantially in their peak brightness, it is important to model how much light they emit. As an example, some mobile displays can reach the peak luminance of 900 cd/m^2 and can be dimmed to as low light levels as 3 cd/m^2. The visibility of image distortions is very different between both cases. To model the amount of the emitted light, we use the standard gain-gamma-offset display model:

$$L = (L_{\text{peak}} - L_{\text{black}}) \left(\frac{I}{255} \right)^{2.2} + L_{\text{black}}, \tag{5.1}$$

where I is the input pixel value, L_{peak} is the peak luminance of the display, and L_{black} is the luminance of black level (light emitted from pixels set to black). Each image provided to the metric is first transformed from pixel values to colorimetric red, green and blue values using the display model from the equation above.

5.2.2 Viewing distance

An intuitive way to take account of the viewing distance is to provide to the model an image with the fixed angular resolution. As the contrast sensitivity of visual system is mostly dependent on the spatial frequency content in cycles per visual degree (cpd), the constant angular resolution ensures that spatial frequencies remain the same regardless of the viewing distance. From Section 2.2.2, we already know how to compute the angular resolution of an image on the display. Once we know the angular resolution of the input image, we resample it so that it has the angular resolution of 60 ppd. 60 ppd is the highest resolution in our dataset and also a reasonable limit for most visual task, since the sensitivity of visual system drops rapidly below 30 cpd [15]. An example of this step is shown in Figure 5.2. In Figure 5.2, the same image is cropped with different patch sizes under varying ppds. For example, 60 ppd's image patch is two times larger than the 30 ppd's image patch in each dimension. Then, we resize the cropped image patches into the same size (48X48) and use the patches as the inputs for deep neural networks in our architecture.

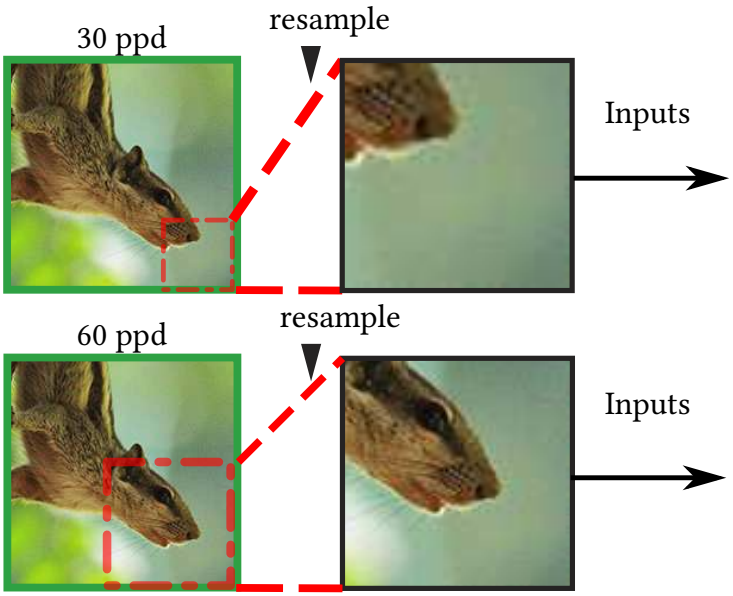


Figure 5.2: The resampling step based on the angular resolution.

Since resampling alone cannot take account of all frequency-dependent effects, such as the shift of peak sensitivity with luminance, we also introduce the *ppd* parameter to the latent code. This is achieved by concatenating a slice with replicated *ppd* values to the feature maps generated by the encoders (see Figure 5.1).

5.2.3 Luminance masking

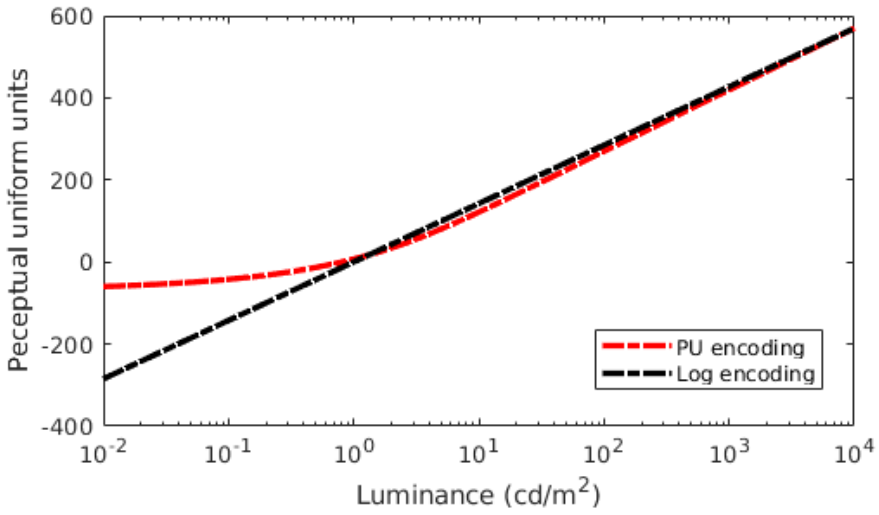


Figure 5.3: PU and logarithmic transform functions, for converting absolute light levels into approximately perceptually uniform values, which could be input to a CNN.

Since differences are less visible at lower absolute luminance levels, we need to take account of this drop of visual system sensitivity. Luminance masking can be modeled by a transfer function derived from the contrast sensitivity function of visual system [1, 7].

The transfer function we use is also known as Perceptually Uniform (PU) encoding [7], as it transforms physical luminance into approximately perceptually uniform units. The PU encoding is defined as an integral of inverse of detection thresholds:

$$P(L) = \int_{L_{min}}^{L} \frac{1}{T(l)} dl \tag{5.2}$$

where L_{min} is the minimum luminance to be encoded. The detection thresholds $T(L)$ are modeled as a function of absolute luminance L:

$$T(L) = S \cdot \left(\left(\frac{C_1}{L} \right)^{C_2} + 1 \right)^{C_3} \tag{5.3}$$

Where S is the absolute sensitivity constant, L is the luminance, and C_1, C_2, C_3 are parameters obtained by fitting to contrast sensitivity measurements. We use the parameters from [1] which is ($C_1 : 4.0627$, $C_2 : 1.6596$, $C_3 : 0.2712$).

For comparison, we also experiment with the logarithmic encoding of luminance, as it is the first-order approximation of the visual system response, which takes account of the Fechner law. We show both perceptual encoding functions in Figure 5.3.

5.2.4 CNN architecture

The CNN architecture of the proposed metric is based on the one proposed in Section 4.3. Although image metrics are often modeled using Siamese architectures [114], the CNN we employ has two independent branches, which encode different information: the first branch encodes the difference between test and reference images (after pre-processing steps) and the second branch encodes the reference image. Such independent branches, shown in Figure 5.1, are used to improve the detection of small image differences. In contrast to CNN architectures used for classification or detection tasks, which need to be robust to noise, our model needs to be particularly sensitive to small variations in input.

Each branch of the encoder uses two convolutional layers of the AlexNet [71]. Two branches and the *ppd* value are concatenated together, as explained in Section 5.2.2. The patch with the predicted probability of detection map is generated by two deconvolution layers. More formally, we denote the perceptually encoded color images of the difference and reference patches as D and R, respectively. We also define mapping functions F_{wconv^d} and F_{wconv^r} to represent the convolutional operations for two branches, in which $wconv^d$ and $wconv^r$ are weights for the difference and reference encoding branches, respectively. We also denote the w_{dec} as the weights for deconvolutional operations with skip connections.

Our metric can then be expressed as:

$$P_w(D,R) = F_{w_{dec}}\left(F_{w_{conv^d}}(D) \textcircled{\scriptsize ||} F_{w_{conv^r}}(R) \textcircled{\scriptsize ||} r\right), \tag{5.4}$$

where $\textcircled{\scriptsize ||}$ represents the concatenation operation of the output of the difference branch, reference branch, and the slice with the replicated *ppd* values r. Note that we do not use batch normalization as batch normalization will normalize data with different peak luminance.

To predict a visibility map for an image of arbitrary size, we slice the image into 48×48 pixel patches with 42-pixel overlap, infer visibility of each patch and compute the final visibility map by averaging the predictions from the overlapping patches. Predicting a visibility map usually takes 2-4 seconds for 1920×1080 image using NVidia GTX 1080Ti GPU.

5.3 Training

For training deep visibility metrics, we use the probabilistic loss function from Section 4.2, as it provides a principled way of modeling the experimental data. The probabilistic loss function models the marking task as a stochastic process taking accounting of the mistakes, lack of attention and a limited number of observations. This allows us to capture the uncertainty in the human-labeled dataset. After the pre-processing steps, we split images into 48×48 pixel non-overlapping patches. We remove the patches where there is no difference between their distorted and reference versions. We implement the CNN in Tensorflow 1.10.1 [2]. We use the adaptive momentum optimizer (Adam) with a learning rate $1e^{-5}$ and a batch size of 48 is used for optimization.

We split the training process into two stages.

Stage 1: Pre-training with HDR-VDP As the collected dataset contains only limited variation in viewing distance and display peak luminance levels, we supplement our training with over 13 million patches that have been automatically labeled by a white-box visibility metric — HDR-VDP. The generation of this PRETRAIN dataset was explained in Section 3.2. The idea is inspired by the work of Kim *et al.* [76], who demonstrated that PSNR scores can be used to pre-train CNN-based quality metrics. Similarly, we run 20000 iterations of training on the PRETRAIN dataset, which is followed by fine-tuning in Stage 2. Although the labels generated by HDR-VDP can be inaccurate, they capture the general relationship between input and output patches and therefore prime the CNN to capture the relationships, which could be missing in manually labeled data.

[2]https://www.tensorflow.org

Stage 2: Fine-tuning At this stage, we initialize the neural network with weights from the first stage and use the manually labeled datasets for training.

5.4 Results

To validate prediction performance, we randomly split the LocVisVC dataset into 5 folds, ensuring that each scene is in a single fold, and run 5-fold cross-validation. We report the mean and standard error of the likelihood used for the loss function (the higher likelihood indicated the higher accuracy).

PU vs. logarithmic encoding First, we compare the performance when either a PU encoding or a logarithmic function is used to take account of luminance masking. The likelihood of the PU encoding (0.877 ± 0.015) was substantially higher than of logarithmic function (0.705 ± 0.02). This suggests that luminance masking is a significant effect in our dataset, which cannot be easily learned by black-box CNN. Given sufficient data, we could expect similar performance for both luminance encodings. This result demonstrates that when the data is limited, the combination of white-box preprocessing and black box learning is a more efficient strategy.

HDR-VDP pre-training Next, we investigate the effect of pre-training on metric performance. We run pre-training for the number of iterations ranging from 10,000 to 50,000, followed by fine-tuning of 50,000 steps, and report the results in Figure 5.4. The figure shows that pre-training always results in higher accuracy, but the performance dropped after about 20,000 iterations. This shows that the amount of pre-training needs to be carefully controlled to retain the ability of the network to effectively learn from the human-labeled data. In the following experiments, we use 20,000 iteration for pre-training. Note the variations of likelihood is high but similar in different folds' validation. This is because, for the training-test split in cross-fold validation, we ensure that the testing set does not contain any of the scenes used for training, regardless of the distortion level.

Metric comparison Finally, we compare the proposed metric to the HDR-VDP, which is the state-of-the-art visibility metric that can take account of the viewing conditions. The result of the cross-validation is shown individually for each subset in Figure 5.5. The likelihood of the proposed DPVM is significantly higher in each subset, demonstrating the CNN-based metric can be trained for higher accuracy.

From Figure 5.5, we can also see that on almost all dataset, the proposed metric outperforms traditional white-box visibility metric-HDR-VDP. Besides, the proposed metric achieves good performance at TID2013 dataset and perceptionpatterns dataset.

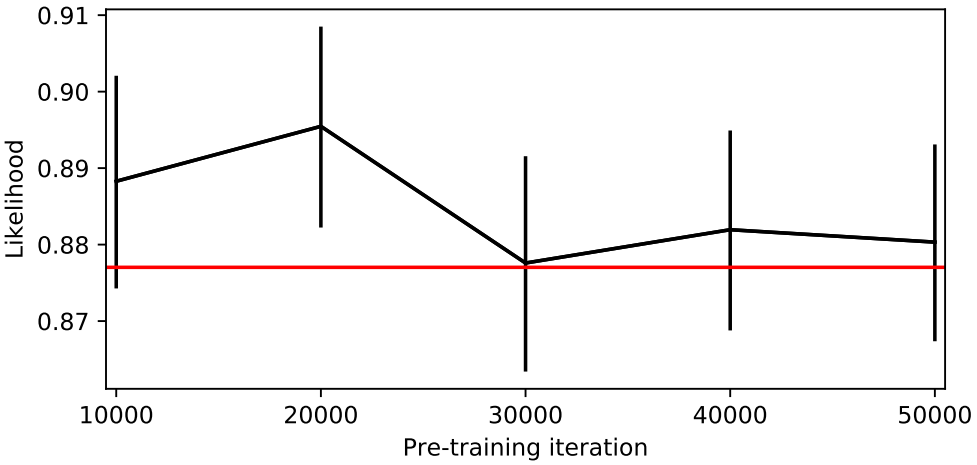


Figure 5.4: The effect of pre-training iterations on the performance. The red line denotes the result without pre-training. The error bars denote standard errors. The higher likelihood, the better is accuracy.

Examples of metric predictions and user markings are shown in Figure 5.6. We can observe there that similar to HDR-VDP, DPVM can take account of the change of viewing distance and absolute luminance as shown in row 1–3. Pre-training with HDR-VDP also helps improve the generalization performance in most cases.

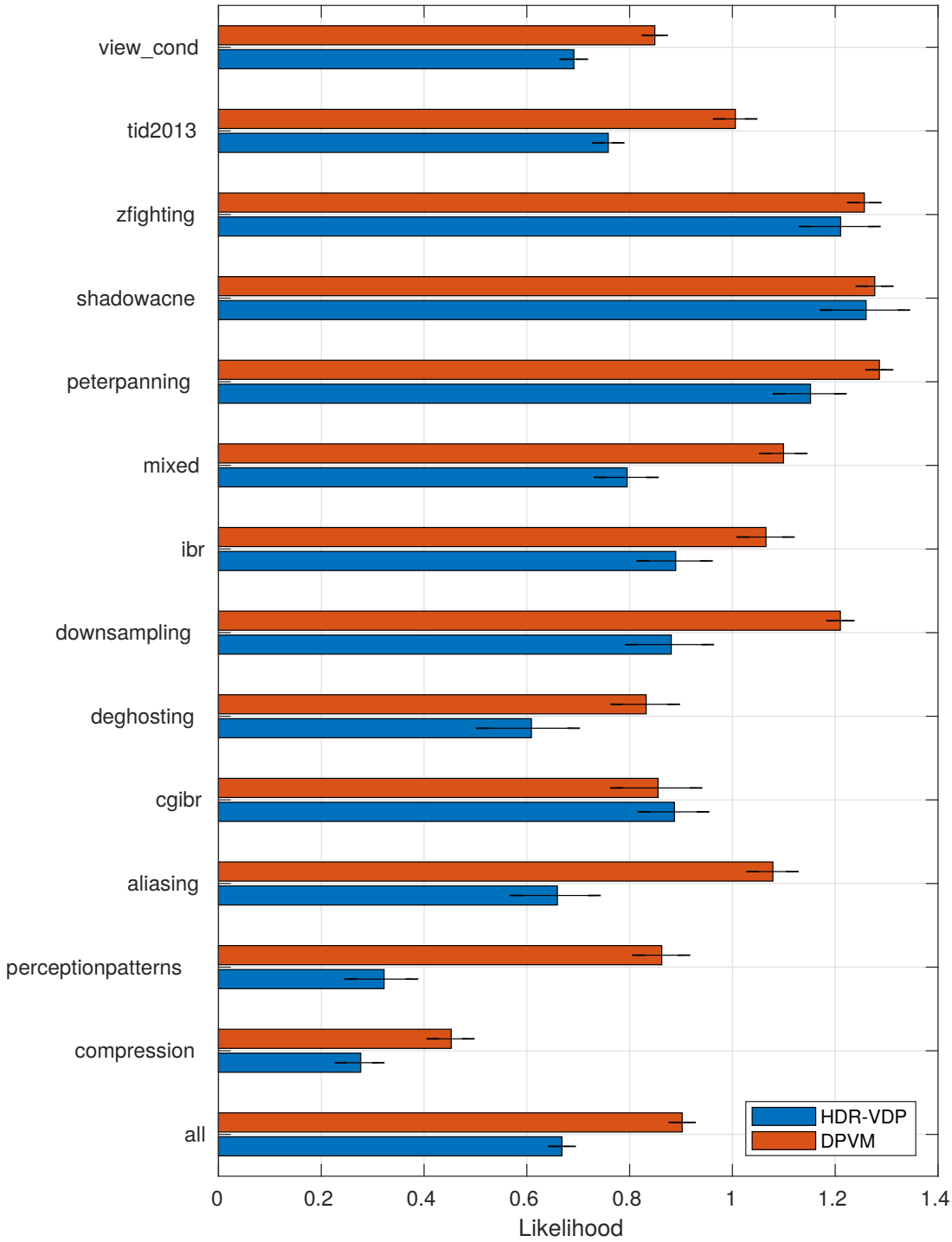


Figure 5.5: Metric cross-validation results for each subset and for the entire dataset.

Generalizing performance under viewing conditions To demonstrate how the proposed DPVM visibility metric can generalize under different display brightness and viewing distances, we show the DPVM's predictions in Figure 5.7.

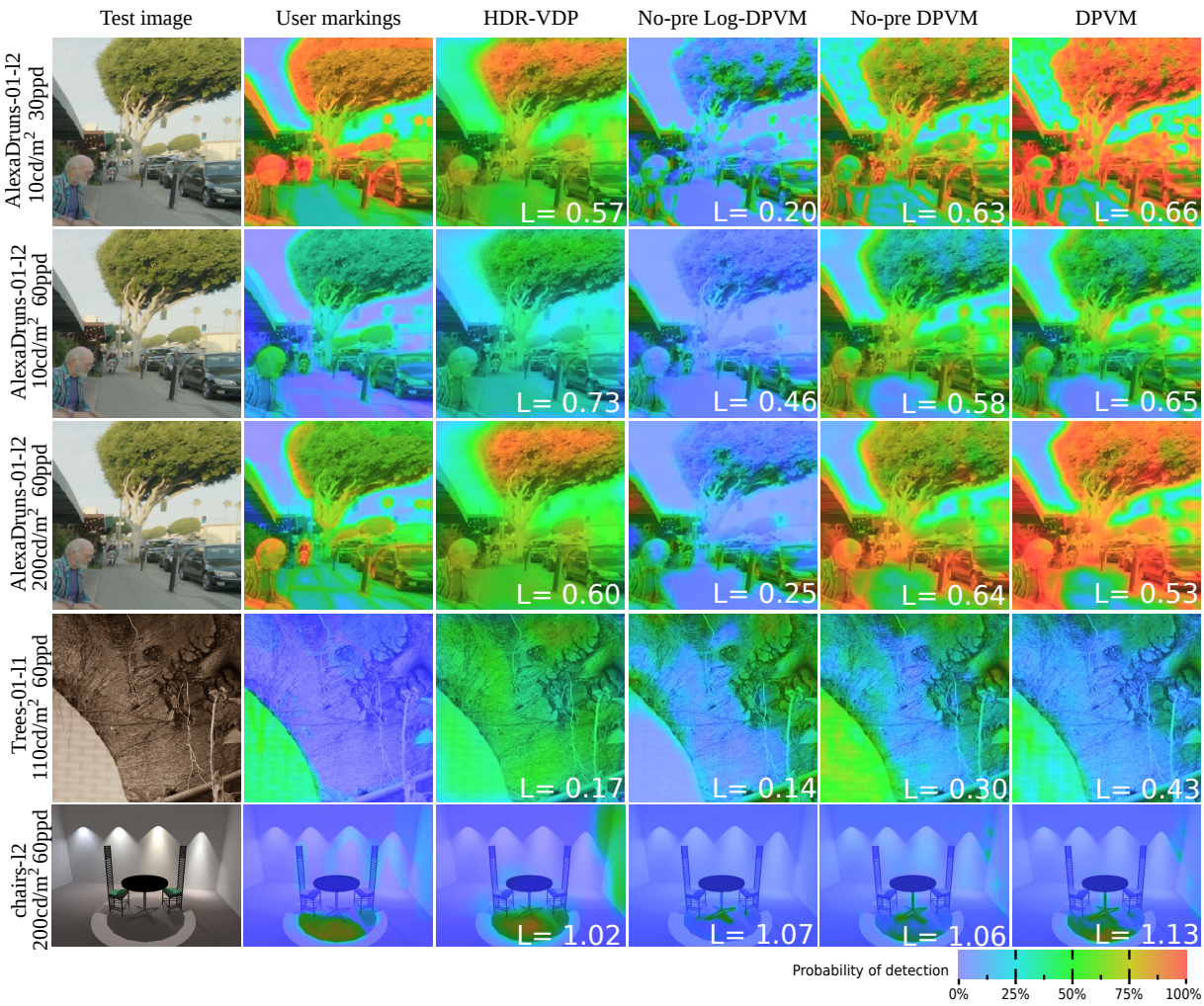


Figure 5.6: Distorted images, users' markings and metrics' predictions examples from the dataset. L is the likelihood, the higher the better. No-pre prefix means without HDR-VDP pre-training.

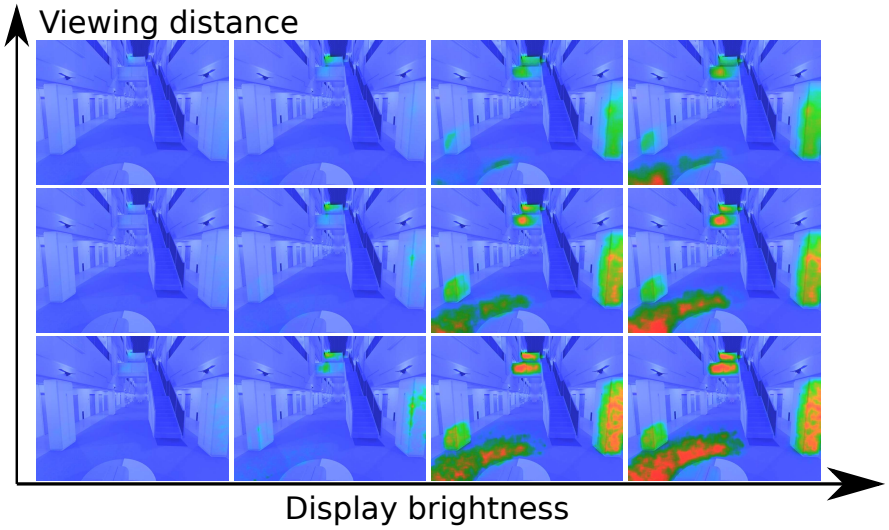


Figure 5.7: Generalization performance of DPVM under varying viewing conditions.

5.5 Summary

In this chapter, we have extended the black-box CNN visibility metric to consider different viewing conditions with white-box models. The proposed gray-box model outperforms the state-of-the-art visibility metric HDR-VDP that can work under different display brightness and viewing distances. Next, we will introduce how to improve the CNN visibility metric for visually lossless image compression. For simplicity, we focus on the case with fixed viewing conditions. Similar methods can be applied to the case with varying viewing conditions.

CHAPTER 6

Visually lossless image compression

6.1 Introduction

To achieve the best trade-off between image compression performance and image quality, we may want to encode images in a visually lossless manner, so that any compression artifacts are invisible to the majority of users. This can be achieved by manually adjusting the compression quality parameter of existing lossy compression methods, such as JPEG or WebP. Visibility metrics can be used to automatically determine the optimal compression quality parameter. The visually lossless threshold (VLT) is the encoder's parameter setting that produces the smallest image file while ensuring visually lossless quality. In this paper, we propose to train a visibility metric to determine the VLT. The proposed flow is shown in Figure 6.1. The original image is compressed at several quality levels by a lossy compression method, such as JPEG or WebP. Then, decoded and original images are compared by the visibility metric to determine the quality level at which the probability of detecting the difference (p_{det}) is below a predetermined threshold.

However, creating an accurate visibility metric is a challenging task because of the complexity of the visual system and the effort needed to collect the required data. In this chapter, we investigate how to train a more accurate visibility metric for visually lossless compression with a relatively small dataset. More specifically, we find that the use of pre-training techniques can significantly improve the accuracy of the CNN visibility metric. The experiments show that we can reduce the prediction error by 40% compared with the state-of-the-art method. In addition, with our proposed method, we can potentially save between 25%-75% of storage space compared with a fixed quality parameter setting of 90 that is similar to the default quality parameters (92-99) used in Photoshop for visually

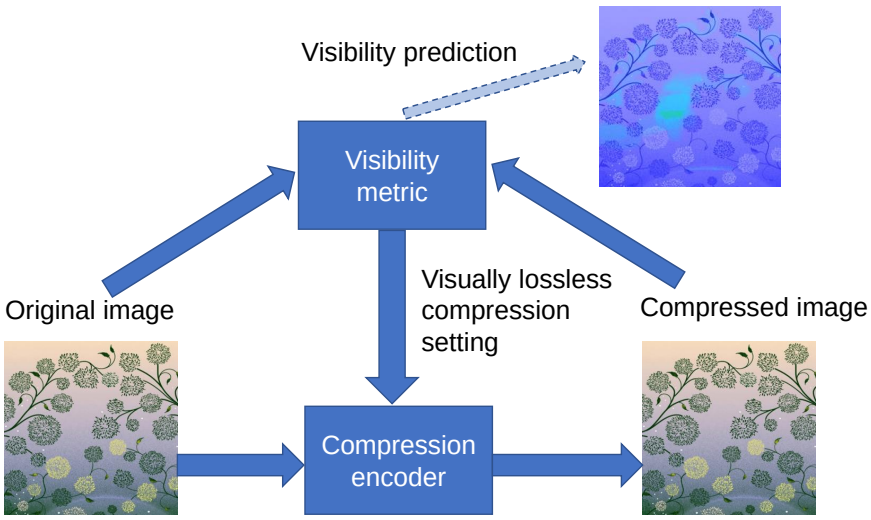


Figure 6.1: Proposed flow of our method for visually lossless compression based on visibility metrics.

lossless image compression [1]. We also demonstrate how the visibility metric can be used to compare the performance of image compression methods.

6.2 Image quality metric for visually lossless image compression

Image quality metrics (IQMs) are very successful at assessing image quality for suprathreshold distortions. However, for visually lossless image compression that requires accurate measurements of near-threshold distortions, IQMs may not be suitable for this task. To explore whether IQMs can predict the visually lossless image compression threshold, we use the state-of-the-art image quality metric, including full-reference IQMs and non-reference IQMs, and demonstrate that both types of IQMs are not suitable for this task in the following experiments. We take the state-of-the-art full-reference IQM—**W**eighted **a**verage **D**eep **I**mage **Qua**lity **M**easure for **F**ull-**R**eference image quality assessment (WaDIQaM-FR) from [115] trained on largest publicly available subjective image quality dataset (TID2013) [99]. We use the WaDIQaM-FR [2] to predict the mean opinion scores (MOSs) of 5 randomly-selected images in the VLIC dataset(collected in Section 3.3). The results are shown in Figure 6.2.

As trends are similar for all images, we randomly select 5 images for clarity of plots. From Figure 6.2, we can observe that the WaDIQaM-FR is not sensitive to the changes in

[1]Commercial software may has different quantization tables, some wide-used software, such as Photoshop, may use a more conservative quantization table and the default quality setting may vary across different versions (https://www.impulseadventure.com/photo/jpeg-quantization-lookup.html?src1=255). Here we use the quality 90 using the libjpeg for benchmarking purposes.

[2]Implementation can be found in `https://github.com/dmaniry/deepIQA`

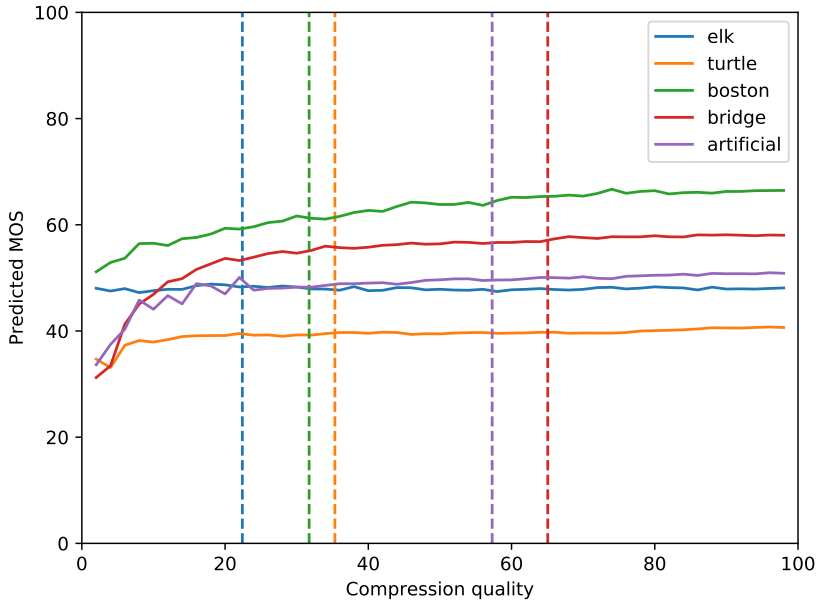


Figure 6.2: The WaDIQaM-FR(full-reference IQM) predictions on images compressed with increasing compression quality. The higher the predicted MOS, the better the visual quality. The vertical lines denotes the VLTs from human experiment. Each color represents a different image.

compression quality for most images. It is also impossible to set a fixed MOS threshold for visually lossless compression because the lowest predicted MOS values of some images are even higher than those of other images. In addition, we also test a widely-used simple full-reference IQM—SSIM that is trained with the LocVis dataset (details in Section 4.6). The results are shown in Figure 6.3. Similar to Figure 6.2, the predictions are noisy flat lines, making it impossible to set a fixed threshold for visually lossless image compression as well.

Besides, we test the state-of-the-art non-reference IQM—**N**eural **IM**age **A**ssessment (NIMA) on the same set of images. As the range of NIMA's predictions are from 0-10 and the higher the better, we re-scale NIMA's prediction score to the same range 0-100 as the previous IQMs. We use an open source implementation for NIMA in our experiment [3].The result is shown in Figure 6.4. The result for this non-reference IQM is even worse as the MOS predictions are almost straight lines. This is because non-reference IQM do not have the information for reference images, and it is very hard to predict the visibility of near-threshold distortions without reference images to compare.

We demonstrate an interesting example of the image named "elk" for analysis as shown in Figure 6.5. From Figure 6.5, we can see that the distortions in the image are definitely visible. However, IQMs are not sensitive to the near-threshold distortions, making IQMs not readily suitable for visually lossless image compression.

[3]Implementation can be found in `https://github.com/kentsyx/Neural-IMage-Assessment`

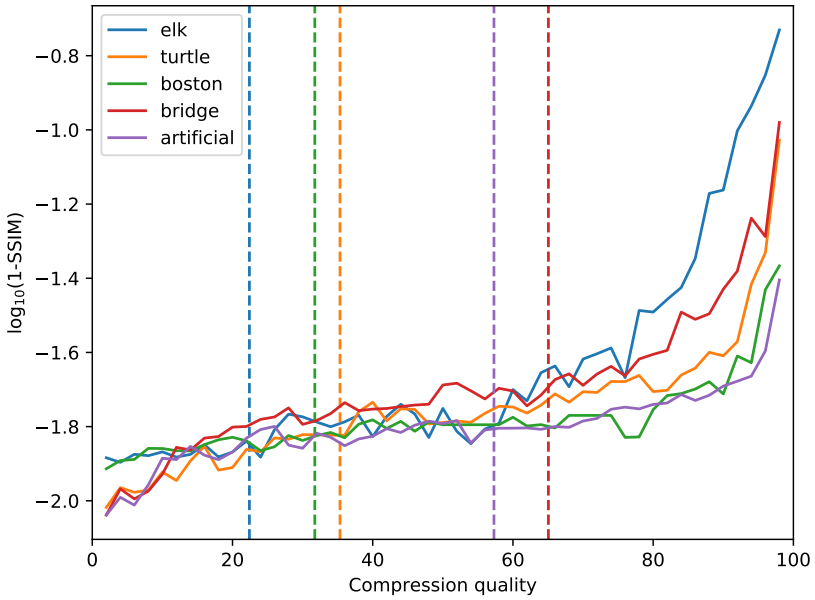


Figure 6.3: SSIM(full-reference IQM) predictions on images compressed with increasing compression quality. For clarity, we transform the Y axis into the log domain. The higher the value of the vertical axis, the better the visual quality. The vertical lines denotes the VLTs from human experiment. Each color represents a different image.

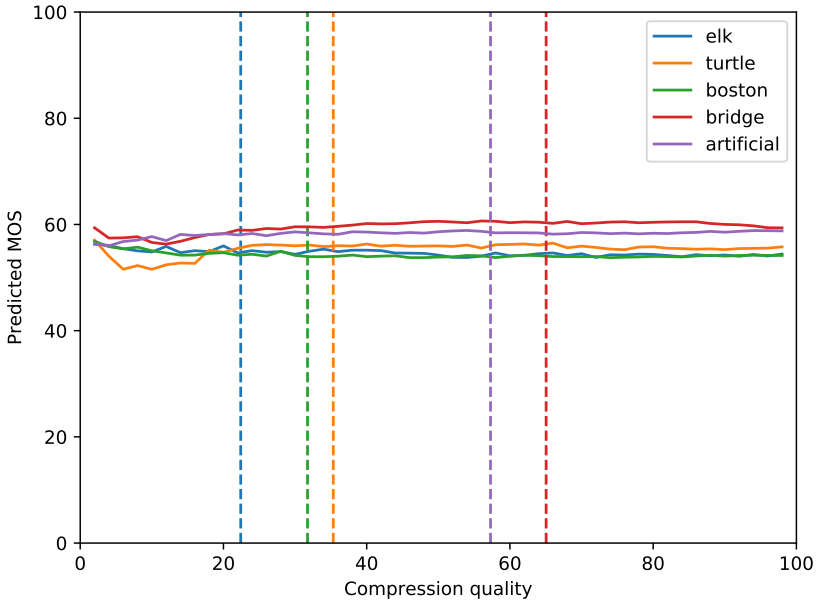
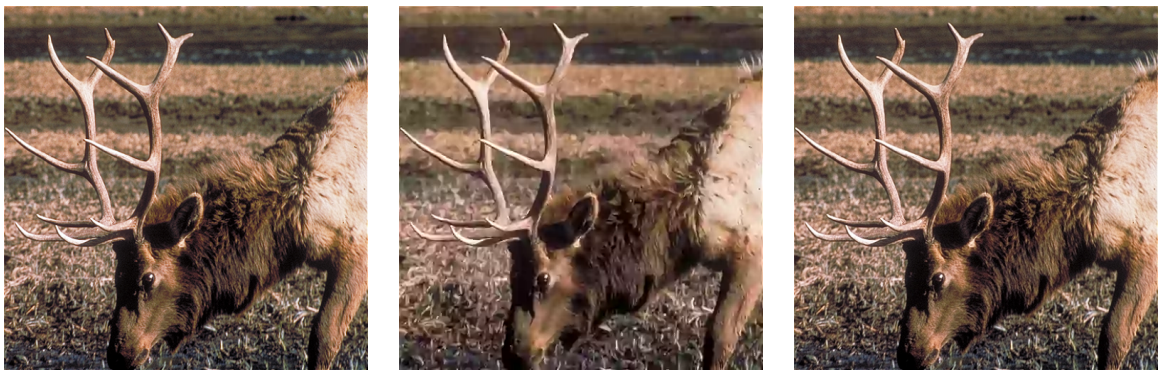


Figure 6.4: NIMA(Non-reference IQM) predictions on images compressed with increasing compression quality. The higher the predicted MOS, the better the visual quality. The vertical lines denotes the VLTs from human experiment. Each color represents a different image.

6.3 Training the network

We train our network on the LocVis dataset[4], which consists of test images, reference images, and maps with the probability of detection experimentally determined for each

[4]LocVis dataset: `https://doi.org/10.17863/CAM.21484`



Reference Compress quality 2 Compress quality 98

Figure 6.5: Image "elk" is compressed at compression quality of 2 or 98 using the Webp encoder. When the compression quality is very low at 2, the distortion is well visible at the elk's fur and grassland. However, IQMs cannot give different predictions when visible distortions are presented as the majority part of the compressed image's appearance is similar to the original uncompressed image.

location in the image. The network is trained using the probabilistic loss function, taking account of the measurement noise, as explained in Section 4.2. To minimize random effects and allow for more rigorous testing, we divide the LocVis dataset into 5 parts and repeat training 5 times for each test, using the leave-one-out approach. Moreover, we do not validate the metric on the left-out part but use the newly collected VLIC dataset instead, as explained in the next section.

To give more insights, we compare a number of training strategies, which we will discuss in the following sections. To make the task computationally feasible, we restrict our experimentation to a subset of possible combinations of strategies. For comparison, we use the ablation study method in the following parts to determine the training method for our proposed visually lossless compression algorithm. In the following experiments, we use the batch size of 48 with 50000 iterations and Adam optimizer. We implemented our experiments in Tensorflow 1.8.

6.3.1 Validation measure

We use the LocVis dataset for training. The validation error is computed on the newly collected VLIC dataset. This ensures that the generalization ability of the proposed metric is tested not only on different images, containing different distortions but also on a different task.

To find the VLT of compressing a particular image, the prediction of p_{det} is computed for 50 quality levels. The prediction, shown as a blue line in in Figure 6.6 (for *big_building* image), often results in non-monotonic function. For that reason, we cannot rely on a binary search or any fast root-finding procedure. Instead, we search from high to low quality to find the quality level ($q1$) at which p_{det} raises above the predetermined threshold

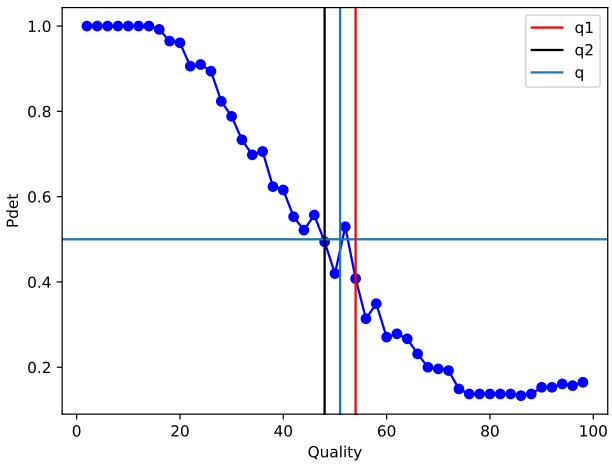


Figure 6.6: The procedure used to determine the visually lossless threshold using the visibility metric.

(0.5 in our case). Then, we search from the low quality to the high quality to find the quality level ($q2$) at which p_{det} drops below the threshold. Then, we compute the mean of these two levels as the metric's prediction for scene i. The validation error reported in the following sections is computed as a root-mean-squared-error (RMSE) between the metric's prediction and the threshold determined in the experiment and averaged across all observers.

6.3.2 Pre-training

Kim *et al.* demonstrated that PSNR scores could be used to pre-train DNN quality metrics, which later fine-tuned on a smaller, human-labeled dataset [76]. Inspired by this idea, we use existing visibility metrics, HDR-VDP and Butteraugli, to generate the additional set of 3000 images with local visibility marking, which greatly increases the amount of training data. The images were taken from TID2013 image quality dataset [99], which consists of 25 scenes affected by 24 different types of distortion at several distortion levels. We first pre-train the visibility metric on the newly generated dataset and then fine-tune the CNN weights on the LocVis dataset with accurate human markings.

In Table 6.1 we compare the performance of the metric trained without pre-training, pre-trained using the dataset generated with HDR-VDP, and with Butteraugli. From Table 6.1, we find that both pre-training datasets can reduce RMSEs and the standard deviations. This suggests that pre-training improved accuracy and enhanced the generalizing ability of the neural network. However, we also observe much larger improvement for HDR-VDP pre-training dataset. The statistical significance of the difference is confirmed by a two-sample t-test and illustrated as underlined in Table 6.1.

Examples of the metric-generated markings are shown in Figure 6.7. We can observe in this figure that Butteraugli can predict sharper results but tends to underestimate

RMSE of VLT Prediction		
No-pretraining	Butteraugli-pretraining	HDR-VDP-pretraining
24.82 ± 5.42	22.48 ± 3.35	**12.62 ± 0.56**

Table 6.1: Pre-training cross-fold validation result (Results that do not have statistically significant differences are underlined)

the effects of Gaussian blur compared with HDR-VDP (column 2 and column 4). The better pre-training performance could be potentially explained by higher complexity of HDR-VDP, which takes account of a larger set of visual phenomena. Even as the metric may appear to be less accurate in terms of actual predictions, it can capture important functional relationships, which are later fine-tuned on the human-labeled LocVis dataset.

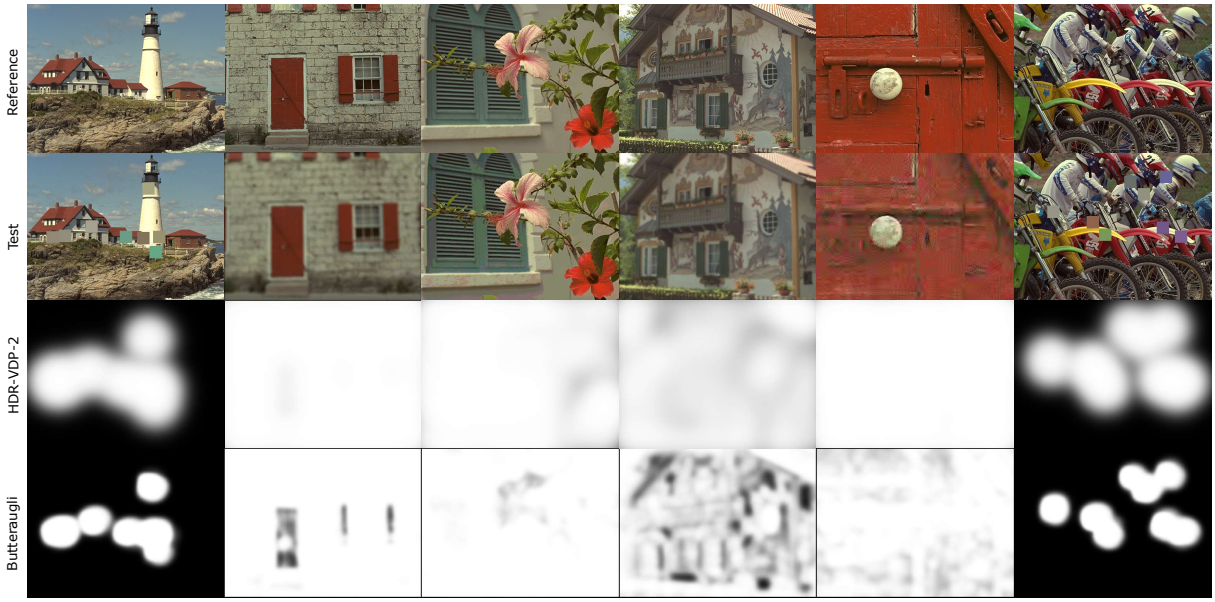


Figure 6.7: Proxy labels generated by HDR-VDP and Butteraugli. Note that the Butteraugli's predictions have boundary artifacts. (Best viewed on the screen when enlarged.)

6.3.3 Data oversampling

To further increase the number of samples for training, we can augment the human-labeled dataset by different techniques. However, many commonly used image-based methods for data augmentation are unsuitable for our task. For example, though adding Gaussian noise or changing image contrast is a common practice in classification problems, it would drastically change the visibility of distortions and thus render marking maps invalid. For instance, it has been shown that one can improve the generalization performance of neural networks by adding synthetic data through simple perturbations to input and output training data, encouraging the model to assign similar outputs to the set of artificial inputs derived from the same training point [116]. To fully test several data augmentation

Figure 6.8: Distortion level interpolation ($r = 0.5$). The first row shows the reference images; the second row shows the distorted images; the third row is the marking images. The interpolated version images are interpolated between neighbouring distortion levels.

methods, we try different combinations of simple data augmentation methods, such as random rotation and flipping with other oversampling approaches based on the previously mentioned idea of generating local perturbations of training data. The data augmentations techniques we experimented with were as follows:

ROF: During training, we randomly flip and/or rotate training images and markings by 0, 90, 180 or 270 degrees.

DI: Inspired by the success of data interpolation for oversampling [117, 118], we propose to interpolate images and markings between neighboring distortion levels. We use the uniform distribution in the range of $[0, 1]$ to determine the interpolation ratio. An example of DI is shown in Figure 6.8.

MIX: The common way of training deep neural networks is to minimize the loss on a finite training dataset, which is referred to as empirical risk minimization (ERM). ERM shows good performance in generalizing the predictions to unseen data. However, ERM-based trained neural networks often show degenerated performance for test data close to the training data because the functions learned by neural networks are often not smooth [119]. For predicting the visibility maps for visually lossless compression, we need neural networks to be accurate when predicting small distortions. Thus we also consider using the mix-up oversampling method to generate training samples from our data distribution [118]. This reduces to using data interpolation in an agnostic way, i.e. independent of

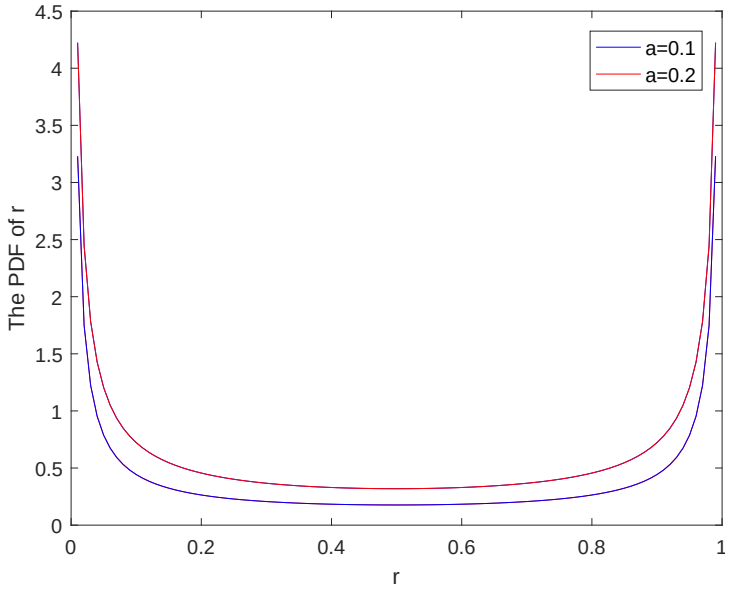


Figure 6.9: The probability distribution function (PDF) of r.

content and distortion level. We sample from the beta distribution to determine the mixup ratio of r. It is described by the probability density function (PDF):

$$p(r) = \frac{1}{B(a,a)} r^{a-1}(1-r)^{a-1}, 0 \leq r \leq 1, \tag{6.1}$$

where B is the Bernoulli distribution, and a is the parameter controlling the skewness. The PDF for two parameters used in the experiments is shown in Figure 6.9. The PDF with high values near 0 and 1 indicates that most newly generated samples are dominated by one or the other image from the sampled pair, as shown in Figure 6.10. The reason to use the beta distribution's PDF is that the beta distribution's PDF has demonstrated superior performance in the previous publication [118]. We find that the best performance can be achieved when mixup samples are introduced every 100 iterations and decide to use this strategy in the following experiments.

6.3.4 Ablation study

We test different combinations of the above data oversampling techniques, with and without pre-training, to find potential interactions between them. The experiment results are shown in Table 6.2. When no pre-training is used (index 13–17), data augmentation helps the neural network to better generalize by reducing the mean or the standard deviation of RMSE results. However, when pre-training is used (index 1–12 in the table), there is little improvement in the performance. Nonetheless, given that data interpolation and mixup training have been shown to help the generalization performance in previous research with many different datasets [118], we decide to use the data interpolation and mixup training

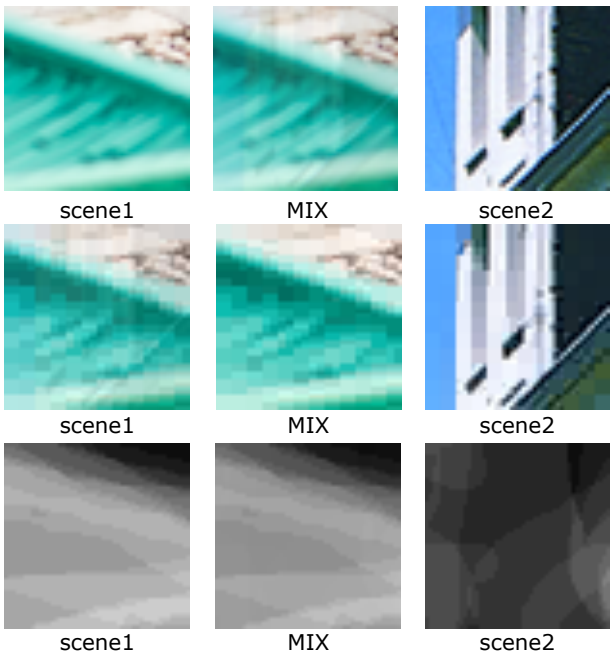


Figure 6.10: Mixup ($r = 0.9$). The first row is the reference images; the second row is the distorted images; the third row is the marking images. The mixed version images are interpolated between neighbouring distortion levels.

Pre-training	Index	ROF	DI	MIX	RMSE
HDR-VDP	1	no	no	no	12.62 ± 0.56
	2	no	yes	no	12.75 ± 1.39
	3	no	no	$a = 0.1$	*12.59 ± 0.77*
	4	no	no	$a = 0.2$	*12.59 ± 0.77*
	5	no	yes	$a = 0.1$	**12.36 ± 1.25**
	6	no	yes	$a = 0.2$	*12.41 ± 1.18*
	7	yes	no	no	14.93 ± 2.02
	8	yes	yes	no	14.29 ± 1.46
	9	yes	no	$a = 0.1$	14.7 ± 1.44
	10	yes	no	$a = 0.2$	14.7 ± 1.44
	11	yes	yes	$a = 0.1$	13.38 ± 0.49
	12	yes	yes	$a = 0.2$	15.13 ± 1.2
none	13	no	no	no	24.82 ± 5.42
	14	no	yes	$a = 0.1$	20.69 ± 6.23
	15	yes	no	no	23.22 ± 3.70
	16	only rotate	no	no	23.04 ± 2.00
	17	only flip	no	no	24.95 ± 2.87

Table 6.2: Data augmentation experiment results.

as the baseline, i.e., the experiment setting of index 5 in Table 6.2. It is worth-noting that rotation and flipping (ROF) results in degenerated performance in all cases. One reason may explain that is human perception ability changes with regard to the orientation of distortions, and ROF for data augmentation is not accurate with visibility data.

6.3.5 Comparison with other methods

In this section, we use the best training strategy obtained from the previous experiments and train our neural network on the entire LocVis dataset. Then, we compare our results with other best performing visibility metrics [104]. The results are shown in Table 6.3. The proposed method reduces RMSE by 40% compared with the CNN-based metric from our previous work in Chapter 4. It should be noted that this result is computed for the newly collected VLIC dataset, which was not used in training. Figure B.3 in the appendix shows a number of images encoded at the VLT, together with the corresponding reference images and the achieved saving in file size.

RMSE of VLT Prediction			
Proposed	CNN[104]	Butteraugli [68]	HDR-VDP [1]
12.38	20.33	20.91	43.33

Table 6.3: Experiment results on the visually lossless image compression dataset.

6.4 Applications

We demonstrate the utility of our metric in two applications: visually lossless image compression and benchmarking of lossy image compression. For this demonstration, we randomly selected 1000 high-quality images from the Flicker8K dataset[5], which were encoded with JPEG quality of 96.

6.4.1 Visually lossless image compression

For visually lossless compression, we used the procedure from Section 6.3.1 to find the VLT with the probability of detection 0.25. This ensured that only 25% of the population had a chance of spotting the difference between the compressed and original images. The threshold was found separately for each image. Then, we computed the saving in file size between our visually lossless coding and JPEG or WebP, both set to quality 90. We chose the quality of 90 as many applications have often used it as a default setting. We plotted the histogram of per-image file size saving in Figure 6.11. The figure shows that our metric can save between 25% to 75% of file size for most images in the dataset for both compression methods. Note that the negative number in the plots indicates that some images need to be compressed with higher quality than 90.

[5] `http://nlp.cs.illinois.edu/HockenmaierGroup/8k-pictures.html`

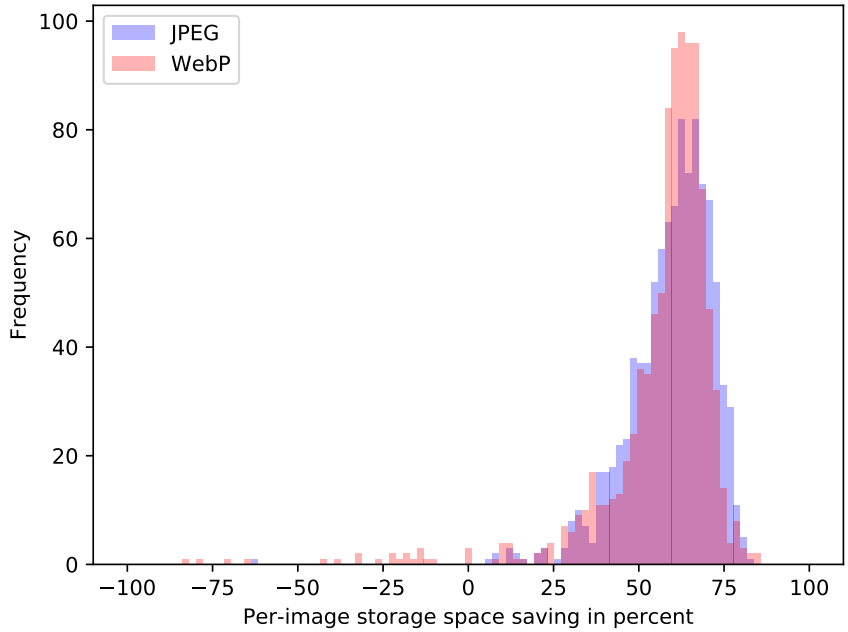


Figure 6.11: Histogram of per-image storage saving as compared to quality 90 setting.

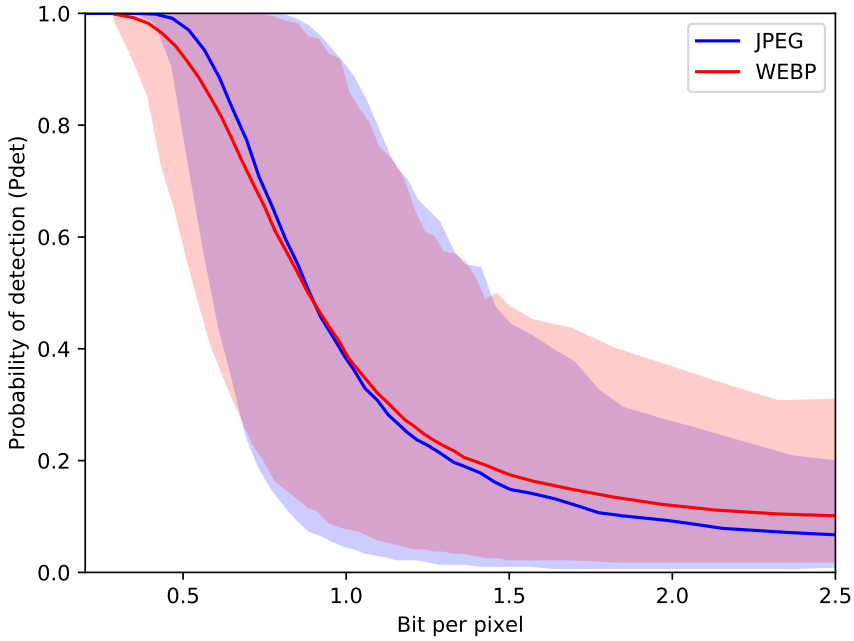


Figure 6.12: Relationship between probability of detection and file size. We use bits-per-pixel because of different size of images in the dataset. The shaded region marks the range between 2.5th to 97.5th percentile.

6.4.2 Benchmarking lossy image compression

To demonstrate that our metric can be utilized for benchmarking lossy image compression methods, we encoded images from Flicker8K dataset at different bit-rates with JPEG and

WebP image compression. Then, for each decoded image, we used our metric to predict the probability of detection and plotted the results in Figure 6.12. The figure shows that WebP can encode low bit-rate images with less noticeable artifacts than JPEG. However, the advantage of WebP is lost at higher bit-rates. This observation is confirmed by several examples shown in Figures B.1 and B.2 in the appendix. The coding artifacts are more visible for JPEG at lower bit-rates as shown in Figure B.1. It is more difficult to spot the difference at higher bit-rates, but examples in Figure B.2 show slightly richer textures (pay attention to the fur and the hair) and more saturated colors in JPEG images. Such analysis can provide useful insights into image compression methods without the need to run tedious quality assessment experiments. Furthermore, such analysis can be performed on thousands of images rather than several dozens, which can be realistically tested in a subjective study. Compared to quality metrics, our visibility metric can express the difference in terms of the probability of detecting artifacts, rather than in terms of an arbitrarily scaled quality value. It was also shown that quality metrics were much less accurate in predicting the visibility of artifacts [104].

6.5 Summary

In this chapter, we have demonstrated that hand-crafted metrics could help to train the CNN-based metric: we used HDR-VDP to generate a large dataset for pre-training the CNN-based metric. This showed that a large dataset with possible inaccurate labels was helpful to initialize the network so that it could capture the main relations between the input and output data. Then, a smaller but accurately labeled dataset was used to fine-tune the weights. This approach seemed to be much more effective than oversampling or data augmentation techniques, and it demonstrated the synergy between white-box metrics and black-box, learning-based metrics.

This work also showed that a CNN-based visibility metric trained on a locally marked dataset (LocVis) could well generalize to the task of predicting visually lossless thresholds. All models were validated on a newly collected dataset (VLIC), containing different images and distortion types, and measured using different procedure than that of the dataset used for training. We demonstrated that the metric could be used to encode images with JPEG and WebP in a visually lossless manner, providing a substantial saving in bandwidth and data storage as compared to a fixed (usually conservative) compression quality setting. Such a visually lossless compression could be applied, for example, to web-caches to reduce the amount of data sent to end-devices, especially through low-bandwidth wireless networks. Moreover, the same metric can be used to compare the performance of image compression methods, which is usually delegated to quality metrics. We argue that predicting visibility (probability of detection) rather than quality (mean-opinion-

score) provides a more accurate and meaningful measurement of visually lossless image compression performance.

CHAPTER 7

Perceptual quality transform for high dynamic image quality assessment

7.1 Introduction

Although the majority of this thesis is about visibility, image quality assessment is an important research field. Quality metrics are necessary to develop robust compression and processing algorithms for high dynamic range (HDR) imaging.

In Chapter 5, we have explored combining perceptual uniform transform into black-box neural networks for luminance masking. However, given that the perception of linear red, green, blue, or luminance values, found in HDR content, is strongly non-linear, standard low-dynamic-range quality metrics, such as peak signal-to-noise ratio (PSNR) or structural similarity index (SSIM), cannot be directly used with HDR images or videos. Linear HDR pixel values can become more perceptually uniform by transforming them into the logarithmic domain [120, 121]. However, such a logarithmic transform does not take account of the absolute brightness of the HDR display. Content shown on a brighter display will reveal more distortions than the same content shown on a darker display. Therefore, most widely used HDR metrics are dependent on displays and require HDR values to be adjusted by a display model so that they represent absolute luminance values (in cd/m^2) emitted from the HDR display. Such adjustment usually involves multiplying pixel values by a constant and clipping the values above or below the dynamic range of a particular display.

The display-referred HDR quality metrics include those metrics that were specifically designed to handle HDR content, such as HDR-VDP [1, 122], HDR-VQM [6], or DRIQM [123], as well as metrics that were adapted from standard-dynamic-range (SDR) metrics

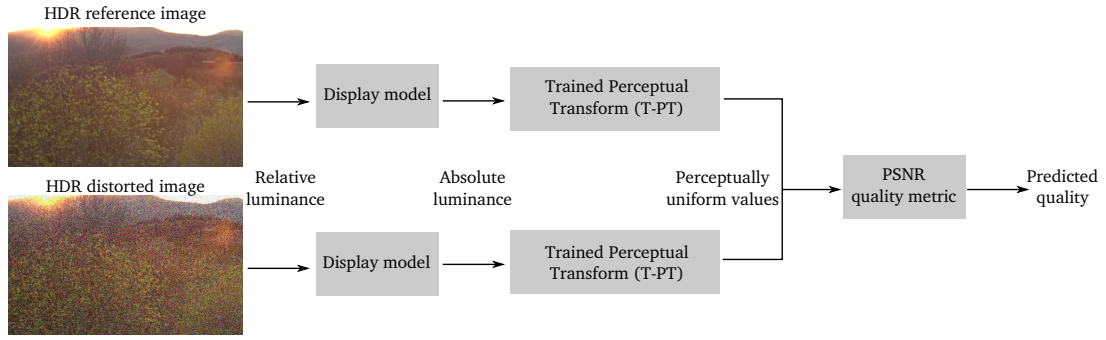


Figure 7.1: The existing SDR quality metrics, such as PSNR, can be adapted to handle HDR content by transforming display-referred color values into perceptually uniform values.

to process HDR content. The adaptation involves a perceptual transform (PT) that converts linear HDR pixel values into perceptually uniform units, which can be directly used with SDR metrics [7], such as PSNR or SSIM. Figure 7.1 illustrates the typical processing blocks of PT-based metrics. The original HDR images are first transformed by a display model to simulate the HDR display and to obtain absolute display-referred color values. Then, a perceptual transform converts both distorted and reference images into perceptually uniform units, which could be input directly into SDR quality metrics. Such an approach may not provide the best predictive performance but it leads to a simple, fast and differentiable quality metric, which could be easily used as a loss function in training image processing algorithms. The property of being differentiable is particularly important when the metric is used as a loss function in optimization-driven problems.

In this chapter, we extend previous work on PT-based metrics [7], proposing a new version that improves the predictive performance of the PU-PSNR metric. Instead of deriving PT from contrast detection models (contrast sensitivity function), we use existing HDR subjective image quality datasets [124–127] to fit the parameters of a new PT function.

7.2 Related work

Quality metrics for HDR images are traditionally based on the models of low-level vision, taking account of the limitations of the visual system. The very first metric, HDR-VDP [128] was designed to predict a map representing visibility of differences between a pair of images, rather than a quality score that would be correlated with mean opinion scores (MOS). The prediction of quality was added in HDR-VDP-2 [1] and then improved in HDR-VDP-2.2 [6] by calibrating metric parameters on HDR quality datasets. The dynamic range independent quality metric (DRIQM) [123] extended HDR-VDP with a set of rules for predicting loss, amplification and reversal of visible contrast to predict objectionable changes between images of different dynamic range, for example, a tone-mapped image and its HDR counterpart. The metric for high dynamic range video, HDR-VQM [6], simplified

spatial processing but added temporal pooling to offer quality predictions for video.

Aydin et al. [7] proposed a perceptually uniform (PU) transform to convert absolute display-referred HDR color values into perceptually uniform units[1], which could be used with existing SDR metrics, such as PSNR or SSIM. The transform is derived to ensure that the change in PU units is relative to just-noticeable differences in luminance, as predicted by the contrast sensitivity function. The transform is further constrained so that the range of luminance values typically reproduced on SDR monitors (0.8—80 cd/m^2) is mapped to a range 0—255 so that the resulting quality values for SDR images corresponded to those values produced by SDR quality metrics. Some authors have started to apply PQ EOTF [129] to achieve similar goals as the PU transform. The main difference between PU and PQ transforms is that the former approach was derived from the HDR-VDP-2 CSF function, whereas the latter strategy was taken from Barten's CSF [16]. It should be noted that a perceptual transform is one of the first processing steps of all advanced HDR quality metrics, including HDR-VDP, HDR-VDP-2, DRIQM, and HDR-VQM.

Although simple quality metrics based on a perceptual uniform transform do not achieve as high predictive performance as more advanced HDR quality metrics, they offer many benefits. Such metrics are significantly less complex, fast to compute, and differentiable, making them a suitable candidate for a perceptual loss function in optimization problems. The obvious limitation of the PU transform is that it does not take account of more complex visual phenomena, such as contrast masking. In this chapter, we explore whether such more complex effects can be partially taken account of by training the PU transform on HDR quality datasets.

7.3 Trained perceptual uniform encoding

Perceptual transform functions (PT), such as PU and PQ, were derived and optimized from contrast detection models intended for simple patterns, such as sinusoidal gratings and Gabor patches. This derivation and optimization results in some of the drawbacks found with the existing PT encodings: i) the used models predict visibility, but visibility may not be directly related to quality and ii) those models do not take complex semantic information in images into account; thus, they may not perform well on complex scenes. These two reasons motivate us to consider fitting the PT using HDR image quality datasets with real-world complex images.

To train such a PT, we first need to determine which transform to use. In practice, both PU [7] and PQ [129] PT encodings have highly similar function shapes. However, after analyzing their results, we can see that the PU function's performance is better than

[1]The code for the PU transform can be found at `https://sourceforge.net/projects/hdrvdp/files/simple_metrics/`

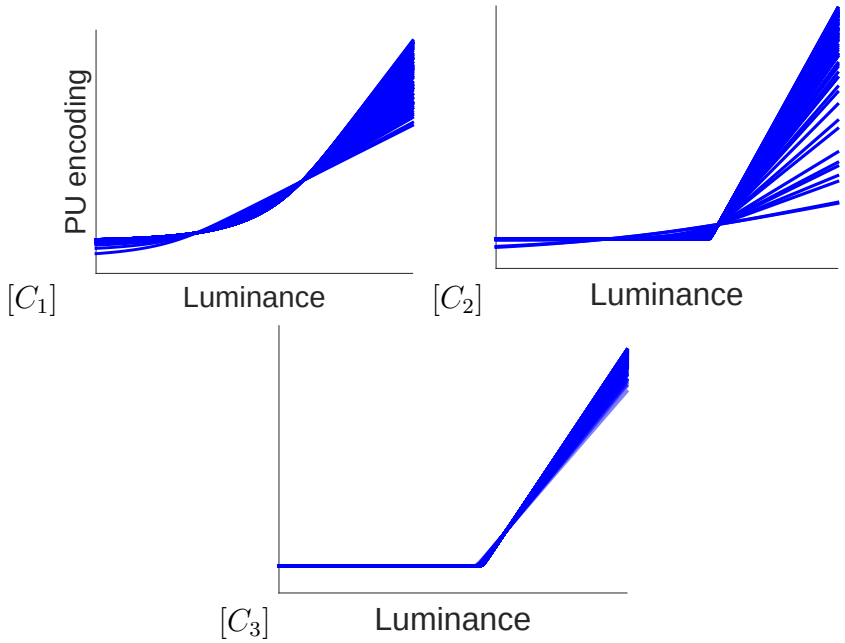


Figure 7.2: Plausible PT encoding functions, where C_1, C_2, and C_3 follow uniform distributions in [0.1—10] with 100 samples.

the PQ function. This improvement is shown in Table 7.3, where it can be seen that the Spearman rank correlation coefficient (SROCC) of PU-PSNR is higher than that of PQ-PSNR. Hence, we use the function used for the PU encoding. The PU transform is defined as an integral of the inverse of detection thresholds:

$$P(L) = \int_{L_{min}}^{L} \frac{1}{T(l)} dl \tag{7.1}$$

where L_{min} is the minimum luminance to be encoded. The detection thresholds $T(L)$ are modeled as a function of absolute luminance L:

$$T(L) = S \cdot \left(\left(\frac{C_1}{L} \right)^{C_2} + 1 \right)^{C_3} \tag{7.2}$$

Where S is the absolute sensitivity constant, L is the luminance, C_1, C_2, and C_3 are scaling parameters. We further linearly rescale the $P(L)$ values so that $P(0.8) = 0$ and $P(80) = 255$. Because of the rescaling, the parameter S does not influence the shape of the function, and the only three adjustable parameters are C_1, C_2 ,and C_3. To illustrate how the curve changes with regard to C_1, C_2 and C_3, in Figure 7.2 we plot the PU curves when each parameter is varied individually in the range of [0.1—10].

A major challenge when using multiple image quality datasets is that each dataset represents quality scores using a different scale. For example, the quality score for two images in two different datasets could be highly similar, but the actual quality of both images may be particularly different. To use all datasets together, those datasets have to

be realigned and put into a common quality scale. Zerman at el. [130] realigned four HDR image quality datasets by using multiple well-known quality metrics. We argue, however, that this may not be an appropriate approach when attempting to test the performance of quality metrics because the realignment may bias the quality scores to offer better predictions for the metrics used in the alignment.

To address the alignment problem we optimize the PT by maximizing SROCC between the predicted quality and dataset's MOS scores. SROCC is computed individually for each dataset, and the values are averaged. SROCC is invariant to any monotonic scaling function and thus avoids the need for quality realignment.

We use a Bayesian optimization method to optimize the parameters that was demonstrated to be effective in hyper-parameter fine-tuning in many applications [131]. The Bayesian optimization method uses Gaussian process regression to estimate the landscape of the loss function and determine the next parameter set for evaluation.

7.4 Results

This section presents the experiments performed to test the behavior of the new trained PU encoding function.

Dataset	Observers	Conditions	Scenes	Distortion type
#1 Narwaria2013[124]	27	140	10	JPEG
#2 Narwaria2014[125]	29	210	6	JPEG 2000
#3 Korsunov2015[127]	24	240	21	JPEG-XT
#4 Zerman2017 [130]	15	100	11	JPEG, JPEG2000, JPEG-XT

Table 7.1: Summary of characteristics of the datasets used in the experiments.

7.4.1 HDR image quality datasets

The available number of subjectively annotated image quality HDR datasets is limited. For our experiments, we selected Narwaria's 2013 dataset[2] [124] and 2014 dataset [125], the dataset by Korshunov[3] [127], and the latest HDR image quality assessment dataset [130] by Zerman et al.[4]. These are, to the best of our knowledge, all the datasets that can be found for HDR image quality assessment. A summary of the main characteristics of these datasets can be found in Table 7.1, including the number of observers, the subjective quality measurement method, the number of conditions and scenes, the distortion type, and display type. All datasets contain images in absolute display-referred units, which correspond to physical luminance and color emitted from the display used in the original experiments. However, due to the differences in implementation of Radiance HDR format,

[2] `http://ivc.univ-nantes.fr/en/databases/JPEG_HDR_Images/`

[3] `http://mmspg.epfl.ch/jpegxt-hdr`

[4] `http://webpages.l2s.centralesupelec.fr/perso/giuseppe.valenzise/download.htm`

the values from Narwaria2013 and Narwaria2014 datasets need to be multiplied by 179 when reading images with *pfstools* software [5].

7.4.2 Trained perceptually uniform encoding

To derive the trained PT encoding function and validate the consistency of the results on different datasets, we train our metric using three datasets and test it on the remaining dataset, repeating the procedure four times. Parameters are initialized in our optimization procedure to the original parameters for the PU function. The results are shown in Table 7.2, which includes the final trained parameter C_1, C_2, and C_3 (T-C_1, T-C_2 and T-C_3) and our optimized result (T-PT-PSNR), the original PQ encoding's result (PQ-PSNR), and the original PU encoding's result (PU-PSNR).

Test dataset	T-C_1	T-C_2	T-$C3$
#1 Narwaria2013[124]	0.10568	4.7378	0.10824
#2 Narwaria2014[125]	0.10078	8.8794	4.405
#3 Korsunov2015[127]	0.1135	3.3663	0.22871
#4 Zerman2017 [130]	0.10054	9.794	2.2137

Table 7.2: The trained T-C_1 – T-C_3 parameters.

Test dataset	T-PT-PSNR	PQ-PSNR	PU-PSNR	HDR-VDP2.2	HDR-VQM
#1 Narwaria2013[124]	**0.6024**	0.58478	0.5898	*0.8911*	*0.8874*
#2 Narwaria2014[125]	**0.4887**	0.38043	0.3605	*0.5727*	*0.8126*
#3 Korsunov2015[127]	**0.8908**	0.8751	0.8833	*0.9503*	*0.9572*
#4 Zerman2017 [130]	**0.8673**	0.81347	0.8249	*0.9298*	*0.9193*

Table 7.3: SROCC results for cross-dataset validation. Each row corresponds to a different test dataset. Bold font indicates the best result excluding complex metrics (HDR-VDP2.2 and HDR-VQM).

Resulting T-PT encoding functions are shown in Figure 7.3. The name of the dataset in the legend indicates the test dataset. From this figure, we can observe that despite training on different datasets, the curves show a similar trend. The biggest difference in the shape of curves can be noted for low luminance, where the T-PT curves have steeper slopes. This result suggests that the visibility of distortions is higher than predicted by the simple detection models (CSFs) used to derive PU and PQ. From Table 7.2, we can also conclude that T-PT functions achieved better results than PU and PQ functions on all datasets. Note that despite an improved performance, the new T-PT-PSNR metric is still worse than HDR-VDP2.2 and VQM metrics. However, the new T-PT-PSNR can compute quality in a fraction of the time required by these two complex metrics.

[5] `http://pfstools.sourceforge.net/`

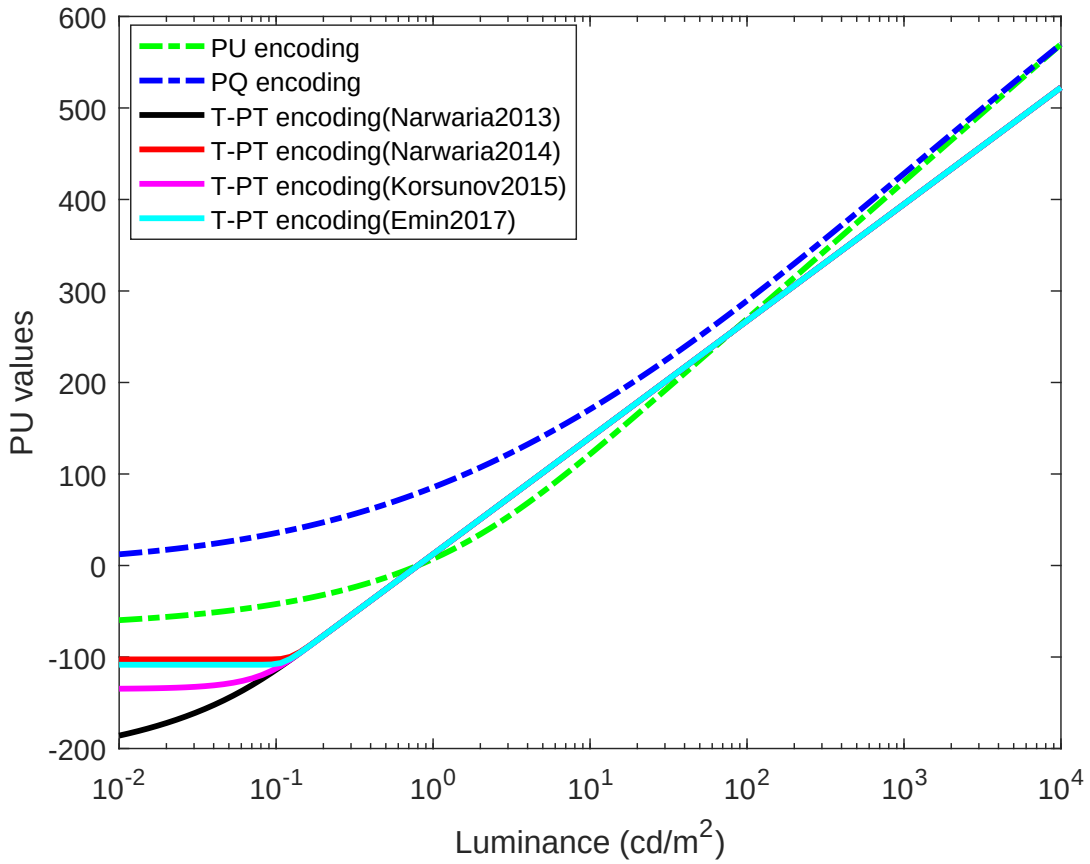


Figure 7.3: T-PT encoding function results from the cross-dataset validation experiment, where the dataset name indicates the one dataset missing in training. The PQ curve is rescaled to have the same maximum value as the PU encoding curve.

Test Dataset	T-PT-PSNR
#1 Narwaria2013[124]	0.6186
#2 Narwaria2014[125]	0.5230
#3 Korsunov2015[127]	0.8906
#4 Zerman2017 [130]	0.8669

Table 7.4: T-PT-PSNR SROCC results when training on all datasets.

To derive our final proposed PT encoding function, we use all the datasets for training. In the training, we optimize the mean of the SROCC values for each dataset to achieve better performance on all datasets. The proposed function curve is shown in Figure 7.4. The final T-C_1, T-C_2, and T-C_3 on all datasets are 0.14249, 2.192 and 0.30499. The SROCC results for this final PT encoding, shown in Table 7.4, indicate further improvement in prediction performance. To further evaluate the results, we take the final PT encoding function and use it as a transfer function for SSIM. The results in Table 7.5, indicates that T-PT-SSIM offers better performance than PU and PQ alternatives for datasets # 3 and # 4, but not for datasets # 1 and # 2, for which PQ-SSIM provides better predictions. We are not sure what could be causing this difference, but we also observe that PQ-SSIM

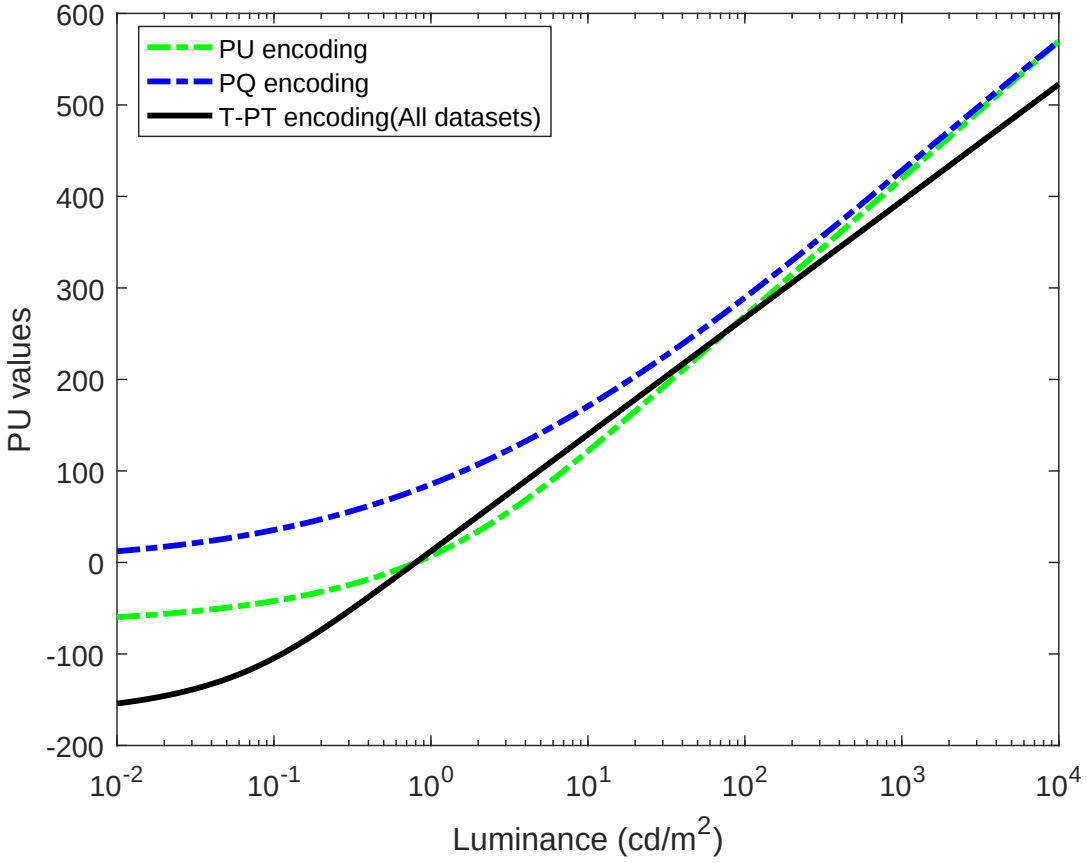


Figure 7.4: T-PT encoding function trained on all datasets. The PQ curve is rescaled to have the same maximum value as PU encoding.

outperforms PU-SSIM for this pair of datasets. Because T-PT is based on the PU function, it is also likely to share worse performance for that particular combination of metric and datasets. It must be noted that T-PT-SSIM was not trained using the SSIM metric, and it is likely that a transfer function needs to be trained separately for each metric.

Test Dataset	T-PT-SSIM	PU-SSIM	PQ-SSIM
#1 Narwaria2013[124]	0.6838	0.6969	**0.7348**
#2 Narwaria2014[125]	0.6145	0.5149	**0.8292**
#3 Korsunov2015[127]	**0.9268**	0.9239	0.8728
#4 Zerman2017 [130]	**0.8864**	0.8430	0.8022

Table 7.5: T-PT-SSIM, PU-SSIM, and PQ-SSIM results.

7.5 Summary

In this chapter, we have proposed a trained perceptually uniform transform for fast quality assessment for HDR images and videos by fitting a perceptual encoding function to a set of subjective quality assessment datasets. We have shown that when combined with

SDR metrics, such as PSNR and SSIM, better performance can be achieved compared to original perceptually uniform transforms. The new transfer function offers a better alternative for low-complexity HDR quality metrics, which are used in the applications for which computational cost is a significant factor.

CHAPTER 8

Conclusion

This section summarizes the contributions this thesis has made for visibility metrics and visually lossless image compression, as well as discusses some potential future research.

8.1 Contribution

1. In Chapter 3, we have discussed the process of collecting datasets for visibility and visually lossless image compression under fixed and varying viewing conditions. I was involved in collecting visually lossless image compression datasets, and the visibility datasets were collected in the MPI and the University of Cambridge jointly.

2. In Chapter 4, as the result of collaborative work with MPI, we have proposed a visibility metric that works well under the fixed display brightness and viewing distance.

3. In Chapter 5, I incorporated white-box modules, such as luminance masking and viewing distance accommodations, in the black-box CNN to allow the proposed visibility metric to be able to work under varying display brightness and viewing distances.

4. In Chapter 6, I improved the performance of the visibility metric in Chapter 4 by almost 40% for visually lossless image compression. I tested the improved visibility metric on 1000 high-quality images and found that this improved visibility metric could save 50—75 % of the storage space compared with the default setting in commercial software to achieve visually lossless image compression.

5. In Chapter 7, I improved the performance of the perceptual uniform transform for HDR image quality assessment by training the transform with HDR image quality datasets.

8.2 Future work

Although we have achieved tremendous performance improvement for visibility metrics and visually lossless image compression with machine learning, there are clearly some questions left to be explored later.

1. Interpretable machine learning for visibility metrics and visually lossless image compression. Although black-box machine learning methods, such as deep learning, can provide us with significant performance improvements, it would be interesting to determine how black-box machine learning methods provide such a prediction.

2. Because videos are playing a more important role in our daily lives, would it be possible to extend our visibility metrics to temporal domains to ensure a good visibility metric for videos. Eventually, we can achieve visually lossless video compression with video visibility metrics, potentially by combining our visibility metric with the white-box temporal contrast sensitivity function model to provide a spatial-temporal CNN visibility metric. How to collect video visibility datasets would then be an open question because there is no experimental protocol of video visibility data collection readily available.

3. On-line learning for visually lossless image compression. In real applications, sometimes, we may not have a perfect visually lossless image compression system at first due to the limited size of the training dataset. However, users can often provide a significant amount of feedback on the quality of compression. If we can use the feedback effectively, a practically useful visually lossless image compression system can be developed despite not having a substantial amount of initial training data.

Bibliography

[1] Rafał K. Mantiuk, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Trans. Graph.*, 30(4):40:1–40:14, July 2011.

[2] Frank W. Campbell and John Robson. Application of fourier analysis to the visibility of gratings. *The Journal of physiology*, 197:551–66, 09 1968.

[3] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[4] Lin Zhang, Ying Shen, and Hongyu Li. VSI: A visual saliency-induced index for perceptual image quality assessment. *IEEE Transactions on Image Processing*, 23(10):4270–4281, 2014.

[5] 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.

[6] Manish Narwaria, Matthieu Perreira Da Silva, and Patrick Le Callet. HDR-VQM: An objective quality measure for high dynamic range video. *Signal Processing: Image Communication*, 35:46 – 60, 2015.

[7] Tunc O Aydin, Rafał K. Mantiuk, and Hans-Peter Seidel. Extending quality metrics to full luminance range images. *Proceedings of SPIE*, 6806:68060B–68060B–10, 2008.

[8] David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America*, 4(12):2379–2394, Dec 1987.

[9] Weisi Lin and C.-C. Jay Kuo. Perceptual visual quality metrics: A survey. *J. Visual Communication and Image Representation*, pages 297–312, 2011.

[10] Damon M Chandler. Seven challenges in image quality assessment: Past, present, and future research. *ISRN Signal Processing*, 2013.

[11] Rafał K. Mantiuk. *High-fidelity imaging : the computational models of the human visual system in high dynamic range video compression, visible difference prediction and image processing*. PhD thesis, Saarland University, 2006.

[12] Brian A. Wandell. *Foundations of Vision*. Sinauer Associates, Sunderland, Massachusetts, 1995.

[13] Donald M. MacKay. Psychophysics of Perceived Intensity: A Theoretical Basis for Fechner's and Stevens' Laws. *Science*, 139(3560):1213–1216, 1963.

[14] Scott J. Daly. Visible differences predictor: an algorithm for the assessment of image fidelity. In Andrew B. Watson, editor, *Digital Images and Human Vision*, volume 1666, pages 179–206. MIT Press, 1993.

[15] Peter G.J. Barten. *Contrast sensitivity of the human eye and its effects on image quality*. SPIE Press, 1999.

[16] Peter. G.J. Barten. Formula for the contrast sensitivity of the human eye. In Y. Miyake and D. R. Rasmussen, editors, *Image Quality and System Performance*, volume 5294, pages 231–238, December 2003.

[17] James. J. Depalma and Earl. M. Lowry. Sine-wave response of the visual system. ii. sine-wave and square-wave contrast sensitivity. *Journal of the Optical Society of America*, 52(3):328–335, 03 1962.

[18] Arpit S. Patel. Spatial resolution by the human visual system. the effect of mean retinal illuminance. *Journal of the Optical Society of America*, 56(5):689–694, May 1966.

[19] John Robson. Spatial and temporal contrast-sensitivity functions of the visual system. *Journal of The Optical Society of America*, 56, 08 1966.

[20] Floris L. Van Nes and Maarten A. Bouman. Spatial modulation transfer in the human eye. *Journal of the Optical Society of America*, 57(3):401–406, Mar 1967.

[21] A. Watanabe, T. Mori, Shojiro. Nagata, and K. Hiwatashi. Spatial sine-wave responses of the human visual system. *Vision Research*, 8(9):1245 – 1263, 1968.

[22] Murray B. Sachs, Jacob Nachmias, and John G. Robson. Spatial-frequency channels in human vision∗. *Journal of the Optical Society of America*, 61(9):1176–1186, Sep 1971.

[23] A. van Meeteren and Johannes J. Vos. Resolution and contrast sensitivity at low luminances. *Vision Research*, 12(5):825 – IN2, 1972.

[24] Edwin R. Howell and Robert F. Hess. The functional area for summation to threshold for sinusoidal gratings. *Vision Research*, 18(4):369–374, 1978.

[25] Gordon E. Legge and John M. Foley. Contrast masking in human vision. *Journal of the Optical Society of America*, 70(12):1458–1471, Dec 1980.

[26] John M. Foley. Human luminance pattern-vision mechanisms: masking experiments require a new model. *Journal of the Optical Society of America*, 11(6):1710–1719, Jun 1994.

[27] Mark Georgeson and Janet Georgeson. Facilitation and masking of briefly presented gratings: Time-course and contrast dependence. *Vision research*, 27:369–79, 02 1987.

[28] Scott J. Daly. Digital images and human vision. chapter The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity, pages 179–206. MIT Press, Cambridge, MA, USA, 1993.

[29] Xuemei Zhang and Brian A. Wandel. A spatial extension of CIELAB for digital color-image reproduction. *Journal of the Society for Information Display*, 5(1):61, 1997.

[30] Jeffrey Lubin. *Vision models for target detection and recognition*, chapter A Visual Discrimination Model for Imaging System Design and Evaluation, pages 245–283. World Scientific, 1995.

[31] Tobias Ritschel, M. Ihrke, Jeppe R. Frisvad, J. Coppens, Karol Myszkowski, and H.-P. Hans P. Seidel. Temporal glare: Real-time dynamic simulation of the scattering in the human eye. *Computer Graphics Forum*, 28(2):183–192.

[32] Jan K. Ijspeert, T.J.T.P. Van Den Berg, and Henk Spekreijse. An improved mathematical description of the foveal visual point spread function with parameters for age, pupil size and pigmentation. *Vision Research*, 33(1):15 – 20, 1993.

[33] Andrew Stockman and Lindsay T. Sharpe. The spectral sensitivities of the middle- and long-wavelength-sensitive cones derived from measurements in observers of known genotype. *Vision Research*, 40(13):1711 – 1737, 2000.

[34] Rafał K. Mantiuk, Scott J. Daly, Karol Myszkowski, and Hans-Peter Seidel. Predicting visible differences in high dynamic range images: model and its calibration, 2005.

[35] Peter Gouras. The role of S-cones in human vision. *Documenta Ophthalmologica*, 106(1):5–11, Jan 2003.

[36] Eero P. Simoncelli and William T. Freeman. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *Proceedings., International Conference on Image Processing*, volume 3, pages 444–447 vol.3, Oct 1995.

[37] ITU JPEG standard. ISO/IEC 10918-1 : 1993(E) CCIT Recommendation T.81, 1993.

[38] Gilbert Strang. The discrete cosine transform. *SIAM Rev.*, 41(1):135–147, March 1999.

[39] Nico Schertler. Improving JPEG Compression with Regression Tree Fields. 2014.

[40] William B. Pennebaker and Joan L. Mitchell. *JPEG Still Image Data Compression Standard*. Kluwer Academic Publishers, Norwell, MA, USA, 1st edition, 1992.

[41] WebP image compression. "https://developers.google.com/speed/webp". Accessed: 2019-02-25.

[42] Khalid Sayood. *Introduction to Data Compression, Fourth Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2012.

[43] Michiel Hermans and Benjamin Schrauwen. Training and analysing deep recurrent neural networks. In *Advances in Neural Information Processing Systems 26*, pages 190–198. Curran Associates, Inc., 2013.

[44] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. *arXiv preprint*, August 2016.

[45] Geoffrey E. Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[46] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2922–2930, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[47] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. *arXiv preprint*, August 2018.

[48] Shibani Santurkar, David Budden, and Nir Shavit. Generative compression. In *2018 Picture Coding Symposium (PCS)*, pages 258–262, June 2018.

[49] Feng Jiang, Wen Tao, Shaohui Liu, Jie Ren, Xun Guo, and Debin Zhao. An end-to-end compression framework based on convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):3007–3018, Oct 2018.

[50] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. 03 2017.

[51] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Semantic perceptual image compression using deep convolution networks. In *2017 Data Compression Conference (DCC)*, pages 250–259, April 2017.

[52] David Minnen, Johannes Ballé, and George D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10794–10803. Curran Associates, Inc., 2018.

[53] I. Schiopu and A. Munteanu. Residual-error prediction based on deep learning for lossless image compression. *Electronics Letters*, 54(17):1032–1034, 2018.

[54] Johannes Ball, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[55] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 7 1948.

[56] Yuting Jia. Just noticeable distortion model and its application in image processing. Master's thesis, National University of Singapore, 2006.

[57] Kai Liu. A Just-Noticeable-Distortion based perceptually lossless image compression codec. Master's thesis, University of Stuttgart, 2012.

[58] Bradley J. Erickson. Irreversible compression of medical images. *Journal of Digital Imaging*, 15(1):5–14, Mar 2002.

[59] O. Kocsis, L. Costaridou, L. Varaki, E. Likaki, C. Kalogeropoulou, S. Skiadopoulos, and G. Panayiotakis. Visually lossless threshold determination for microcalcification detection in wavelet compressed mammograms. *European Radiology*, 13(10):2390–2396, 2003.

[60] Kyoung Ho Lee, Young Hoon Kim, Bo Hyoung Kim, Kil Joong Kim, Tae Jung Kim, Hyuk Jung Kim, and Seokyung Hahn. Irreversible JPEG 2000 compression of abdominal CT for primary interpretation: Assessment of visually lossless threshold. *European Radiology*, 17(6):1529–1534, 2007.

[61] Sheila S. Hemami Damon Michael Chandler, Nathan L. Dykes. Visually lossless compression of digitized radiographs based on contrast sensitivity and visual masking. In *Proc. of SPIE Medical Imaging: Image Perception, Observer Performance, and Technology Assessment*, pages 5749–14, 2005.

[62] Wenjun Zeng, Scott J. Daly, and Shawmin Lei. An overview of the visual optimization tools in JPEG 2000. *Signal Processing: Image Communication*, 17(1):85–104, 2002.

[63] David Wu, Damian M. Tan, Marilyn Baird, John DeCampo, Chris White, and Hong Ren Wu. Perceptually lossless medical image coding. *IEEE Transactions on Medical Imaging*, 25(3):335–344, 2006.

[64] Andrew. B. Watson, Gloria. Y. Yang, Joshua. A. Solomon, and John. Villasenor. Visibility of wavelet quantization noise. *IEEE Transactions on Image Processing*, 6(8):1164–1175, Aug 1997.

[65] H. H. Y. Tong and Anastasios N. Venetsanopoulos. A perceptual model for jpeg applications based on block classification, texture masking, and luminance masking. In *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, pages 428–432 vol.3, Oct 1998.

[66] Bohyoung Kim, Kyoung Ho Lee, Kil Joong Kim, Rafał K. Mantiuk, Seokyung Hahn, Tae Jung Kim, and Young Hoon Kim. Prediction of perceptible artifacts in JPEG 2000 compressed chest ct images using mathematical and perceptual quality metrics. *American Journal of Roentgenology*, 190(2):328–334, Feb 2008.

[67] Michael P. Eckert and Andrew P. Bradley. Perceptual quality metrics applied to still image compression. *Signal Process.*, 70(3):177–200, November 1998.

[68] Jyrki Alakuijala, Robert Obryk, Ostap Stoliarchuk, Zoltan Szabadka, Lode Vandevenne, and Jan Wassenberg. Guetzli: Perceptually guided JPEG encoder. *arXiv preprint*, June 2017.

[69] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2013.

[70] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, Berlin, Heidelberg, 2006.

[71] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[72] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[73] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[74] Yujun Shen, Ping Luo, Junjie Yan, Xiaogang Wang, and Xiaoou Tang. Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[75] Da Pan, Ping Shi, Ming Hou, Zefeng Ying, Sizhe Fu, and Yuan Zhang. Blind predicting similar quality map for image quality assessment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[76] Jongyoo Kim, Hui Zeng, Deepti Ghadiyaram, Sanghoon Lee, Lei Zhang, and Alan C. Bovik. Deep convolutional neural models for picture-quality prediction: Challenges and solutions to data-driven image quality assessment. *IEEE Signal Processing Magazine*, 34(6):130–141, Nov 2017.

[77] Jangyoo Kim and Sanghoon Lee. Deep learning of human visual sensitivity in image quality assessment framework. *Computer Vision and Pattern Recognition (CVPR)*, pages 1969–1977, 2017.

[78] Ekta Prashnani, Hong Cai, Yasamin Mostofi, and Pradeep Sen. Pieapp: Perceptual image-error assessment through pairwise preference. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[79] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[80] Bohdan Macukow. Neural Networks – State of Art, Brief History, Basic Models and Architecture. In Khalid Saeed and Wladyslaw Homenda, editors, *15th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM)*, volume LNCS-9842 of *Computer Information Systems and Industrial Management*, pages 3–14, Vilnius, Lithuania, September 2016. Springer International Publishing. Part 1: Invited Paper.

[81] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, expanded edition, 1988.

[82] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[83] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). *CoRR*, abs/1511.07289, 2015.

[84] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *arXiv preprint*, June 2017.

[85] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. *arXiv preprint*, October 2017.

[86] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2018.

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

[88] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2008–2026, 2017.

[89] David H. Hubel and Torsten N. Wiesel. Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology*, 148:574–591, 1959.

[90] Guy Wallis and Edmund T. Rolls. Invariant face and object recognition in the visual system. *Progress in Neurobiology*, 51(2):167 – 194, 1997.

[91] Masakazu Matsugu, Katsuhiko Mori, Yusuke Mitari, and Yuji Kaneda. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5-6):555–559, 2003.

[92] Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

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

[94] Yi Ting Zhou and Rama Chellappa. Computation of optical flow using a neural network. *IEEE 1988 International Conference on Neural Networks*, pages 71–78 vol.2, 1988.

[95] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015.

[96] Johan Bjorck, Carla P. Gomes, and Bart Selman. Understanding batch normalization. *arXiv preprint*, June 2018.

[97] Md Mushfiqul Alam, Kedarnath P. Vilankar, David J. Field, and Damon M. Chandler. Local masking in natural images: A database and analysis. *Journal of Vision*, 14(8):22, 2014.

[98] Martin Čadík, Robert Herzog, Rafał K. Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. New measurements reveal weaknesses of image quality metrics in evaluating graphics artifacts. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 31(6):147, 2012.

[99] Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and C.-C. Jay Kuo. Image database tid2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, 30:57 – 77, 2015.

[100] Martin Čadík, Robert Herzog, Rafał K. Mantiuk, Radosław Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. Learning to predict localized distortions in rendered images. In *Computer Graphics Forum*, volume 32, pages 401–410, 2013.

[101] Rafał Piórkowski, Radosław Mantiuk, and Adam Siekawa. Automatic detection of game engine artifacts using full reference image quality metrics. *ACM Transactions on Applied Perception (TAP)*, 14(3):14, 2017.

[102] Kanita Karaduzovic-Hadziabdic, Jasminka Hasic Telalovic, and Rafał K. Mantiuk. Assessment of multi-exposure hdr image deghosting methods. *Computers and Graphics*, 63, 01 2017.

[103] Maria Perez-Ortiz and Rafał K. Mantiuk. A practical guide and software for analysing pairwise comparison experiments. *arXiv preprint*, December 2017.

[104] Krzysztof Wolski, Daniele Giunchi, Nanyang Ye, Piotr Didyk, Karol Myszkowski, Radosław Mantiuk, Hans-Peter Seidel, Anthony Steed, and Rafał K. Mantiuk. Dataset and metrics for predicting local visible differences. *ACM Trans. Graph.*, 37(5):172:1–172:14, November 2018.

[105] Eric C. Larson and Damon M. Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *J. Electronic Imaging*, 19(1):011006, 2010.

[106] Clément. Strauss, Francois. Pasteau, Florent. Autrusseau, Marie. Babel, Laurent. Bedat, and Olivier. Deforges. Subjective and objective quality evaluation of LAR coded art images. In *2009 IEEE International Conference on Multimedia and Expo*, pages 674–677, June 2009.

[107] Andrew B. Watson and Denis G. Pelli. Quest: A Bayesian adaptive psychometric method. *Perception & Psychophysics*, 33(2):113–120, Mar 1983.

[108] Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek. Neural network-based full-reference image quality assessment. In *Proceedings of the Picture Coding Symposium (PCS)*, pages 1–5, 2016.

[109] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[110] Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5:1–11, 02 2017.

[111] Nanyang Ye, Zhanxing Zhu, and Rafał K. Mantiuk. Langevin dynamics with continuous tempering for training deep neural networks. In *Advances in Neural Information Processing Systems 30*, pages 618–626. Curran Associates, Inc., 2017.

[112] Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2101–2110. PMLR, Auguest 2017.

[113] Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In *Proceedings of the 2017 Conference on Learning Theory*, volume 65, pages 1980–2022. PMLR, July 2017.

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

[115] Sebastian. Bosse, Dominique. Maniry, Klaus. Mller, Thomas. Wiegand, and Wojciech. Samek. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Transactions on Image Processing*, 27(1):206–219, Jan 2018.

[116] Chris M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Comput.*, 7(1):108–116, January 1995.

[117] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002.

[118] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[119] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2553–2561. Curran Associates, Inc., 2013.

[120] Rafał K. Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. *High Dynamic Range Imaging*. John Wiley and Sons, Inc., 1999.

[121] Rafał K. Mantiuk. Practicalities of predicting quality of high dynamic range images and video. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 904–908, Sept 2016.

[122] Manish Narwaria, Rafał K. Mantiuk, Mattheiu Perreira Da Silva, and Patrick Le Callet. HDR-VDP-2.2: a calibrated method for objective quality prediction of high-dynamic range and standard images. *Journal of Electronic Imaging*, 24(1):010501, jan 2015.

[123] Tunç Ozan Aydin, Rafał. K Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. Dynamic range independent image quality assessment. *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, 27(3):69, 2008.

[124] Manish Narwaria, Matthieu Perreira Da Silva, Patrick Le Callet, and Romuald Pepion. Tone mapping based hdr compression: Does it affect visual experience? *Signal Processing: Image Communication*, 29(2):257 – 273, 2014. Special Issue on Advances in High Dynamic Range Video Research.

[125] Manish Narwaria, Da Silva Matthieu Perreira, Le Callet Patrick, and Pepion Romuald. Impact of tone mapping in high dynamic range image compression. In *Proceedings of the eighth international workshop on video processing and quality metrics for consumer electronics (VPQM)*, 2014.

[126] Giuseppe Valenzise, Francesca De Simone, Paul Lauga, and Frederic Dufaux. Performance evaluation of objective quality metrics for HDR image compression. In *Applications of Digital Image Processing XXXVII*, San Diego, United States, August 2014. SPIE.

[127] Pavel Korshunov, Phillippe Hanhart, Thomas Richter, Alessandro Artusi, Rafał. K. Mantiuk, and Touradj Ebrahimi. Subjective quality assessment database of hdr images compressed with jpeg xt. In *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 1–6, May 2015.

[128] Rafał. K Mantiuk, Scott. J Daly, Karol Myszkowski, and Hans-Peter Seidel Seidel. Predicting visible differences in high dynamic range images: model and its calibration. In *Human Vision and Electronic Imaging*, pages 204–214, 2005.

[129] Scott Miller, Mahdi Nezamabadi, and Scott Daly. Perceptual signal coding for more efficient usage of bit codes. In *The 2012 Annual Technical Conference Exhibition*, pages 1–9, Oct 2012.

[130] Emin Zerman, Giuseppe Valenzise, and Frederic Dufaux. An extensive performance evaluation of full-reference HDR image quality metrics. *Quality and User Experience*, 2(5), 2017.

[131] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc., 2012.

[132] Alexander Keller. Instant Radiosity. *SIGGRAPH '97 (Proceedings of the 24th annual conference on Computer graphics and interactive techniques)*, 1997.

[133] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A K Peters, Ltd, 2001.

[134] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. *Computer Graphics*, 22(4), 1988.

[135] Jaroslav Křivánek, Pascal Gautron, Sumanta N. Pattanaik, and Kadi Bouatouch. Radiance caching for efficient global illumination computation. *IEEE Trans. Vis. Comput. Graph.*, 11(5):550–561, 2005.

[136] Kanita Karauzović-Hadžiabdić, Jasminka Hasić Telalović, and Rafał K Mantiuk. Assessment of multi-exposure hdr image deghosting methods. *Computers & Graphics*, 63:1–17, 2017.

[137] Vamsi K. Adhikarla, Marek Vinkler, Denis Sumin, Rafał K. Mantiuk, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. Towards a quality metric for dense light fields. In *Computer Vision and Pattern Recognition*, 2017.

APPENDIX A

Summary of stimuli in LocVis dataset

The LocVis dataset has 557 low dynamic range (LDR) images with 170 unique scenes. Many of them are generated for up to 3 distortion levels. The distortion types covered many common and specialized computer graphic artifacts such as noise, image compression, shadow acne etc. This variety makes our data challenging for even the state-of-the-art image metrics. During the selection process that aimed to generate a large database collecting images from different sources, we gathered scenes with multiple artifact types in a specific category named mixed. In this way we avoided ambiguities in all the other sets or possible hiding effects of one artifact over another one. In the following, the newly-collected stimuli subsets are marked with *. At the end of this section, we present all images from the dataset except TID2013 subset due to its big volume. In this part of the work, I proposed the idea of using TID2013 dataset and generated the initial version of TID2013 dataset for visibility predictions. Then, a pairwise comparision experiment was conducted in the Max Planck Institute to clean some images in TID2013 subset.

A.1 Mixed dataset

The mixed dataset has 59 images from LOCCG data set [98] and contains more than one distortion type. The distortions used in this dataset include high-frequency and low-frequency noise, structured noise, virtual point lights (VPL) artifacts, clamping, downsampling, blurring and light leaking artifacts. One example of those artifacts is presented in Figure A.1.

High frequency noise is a common artifact in many global illumination methods e.g. ray tracing, path tracing, radiosity etc. It is caused by low number of samples and appears usually in shadowed areas. Structured noise is a distortion that results from correlated

pixel errors. Both noise and bias are showed. Instant radiosity [132], photon mapping [133] and radiance caching algorithm [134] [135] can exhibit interpolation and caching artifacts.

VPL is one of many global illumination methods. VPL causes local brightness changes and low-frequency noise, that spoil the overall look of the image. Due to its high computational complexity, this method is not used in real time computer graphics but in production renderings (movies, animations, architectural visualizations).

Light leaking is one of the photon mapping artifacts and appears like an area clearly brighter than normal. It depends on illumination of particular geometries in the scene, like corners in a room, or related to specific attributes of a material like smoothness, showing reflected light even if the object is closed off by other geometries.

These last two techniques (VPL and photon mapping) are classified as approximate global illumination algorithms. Locally they inject errors, sometimes deliberately to camouflage more evident artifacts. Artifacts like VPL clamping in instant radiosity, light leaking in photon mapping and irradiance caching belong to this set.

Figure A.1: Example of artifacts in VPL method.

A.2 Perception patterns

Contrast-Luminance-Frequency-Masking (CLFM) dataset consists of 34 images from [100] that are artificial patterns designed to expose well-known perceptual phenomena, such as luminance masking, contrast masking and contrast sensitivity. The images are generated in the luminance domain (linear) and converted to gray scale images (luma) using the sRGB color space. Different from other sets in our collection this one includes abstract patterns like blobs or stripes with different contrast values. For those scenes we prepared three distortion levels by blending the linearly reference and the distorted image. One of the scenes is shown in Figure A.2.

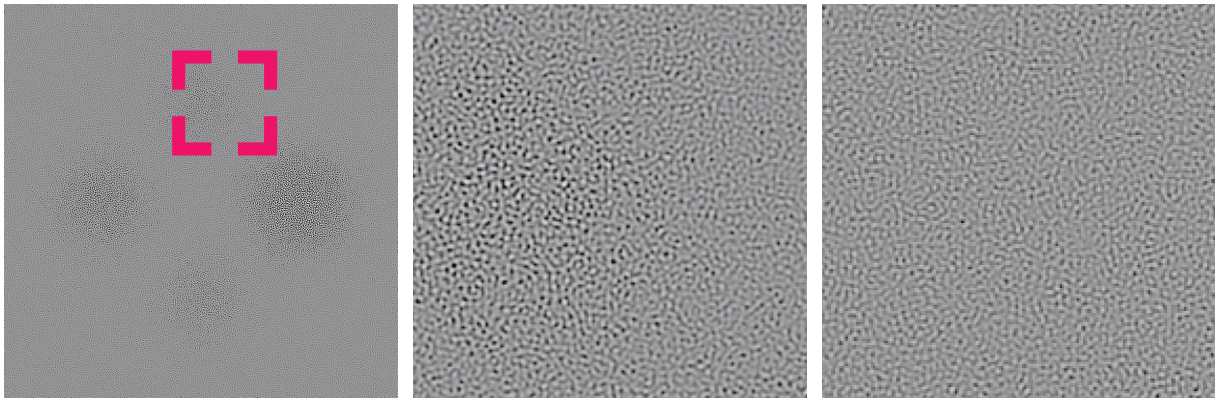


Figure A.2: Example of perception patterns dataset.

A.3 Aliasing dataset

Generally aliasing is a phenomenon which happens when sampling frequency of a signal is too low to reproduce high frequency details accordingly to the Nyquist criterion. In the image domain, aliasing appears as an effect that include jagged profiles, improperly rendered details, and stair step artifacts on the edges. All these images are rendered starting from 3D scenes of interior rooms or outdoor environments. In this category we created from one up to three distortion levels by using different sample numbers for multi-sampling anti-aliasing method. Images source: [101]. Figure A.3 shows an example of aliasing artifact.

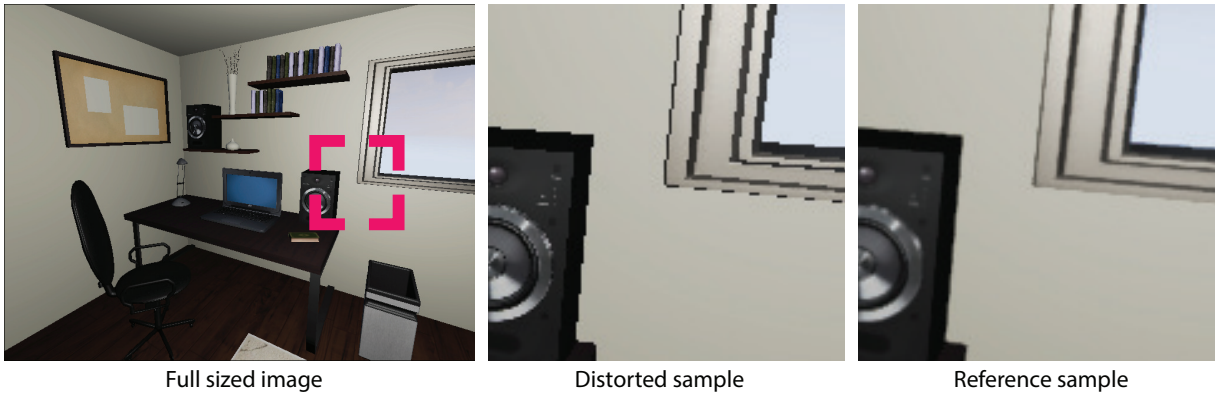


Figure A.3: Example of aliasing artifact.

A.4 Shadow acne*

Shadow acne is an effect caused by the discrete nature and limited resolution of the shadow map. During depth map generation the angle between surface and ray of light has to be taken into account. Tilted depth texel can cross the surface with a part above and a part

below it. The resulting effect is a striped Moire pattern. This supra-threshold type of artifact is commonly seen in computer games and can be the reason of unnatural looking image. Images come from [101]. An example of this kind of artifact is presented in Figure A.4.

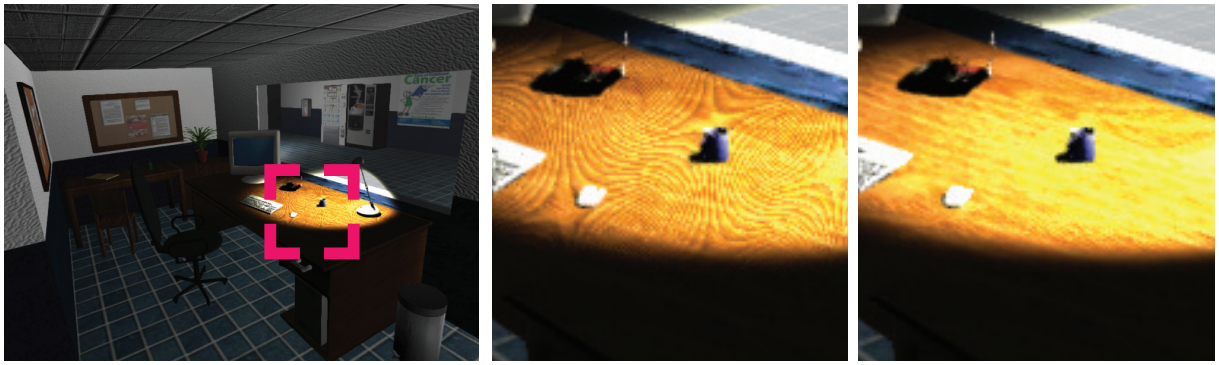


Figure A.4: Example of Moire pattern known as shadow acne artifact.

A.5 Peter panning*

This artifact appears clearly as supra-threshold distortion and is related to objects with missing shadows or part of it, which look like detached from the surface, conveying the illusion of floating above the surface. Peter Panning arises from a correction of another problem. Since adding a depth offset is a technique for removing shadow acne (Section A.4), this increment is related to pixel position in light space. Peter Panning results arises from too large depth offset which causes errors in the depth test. Like shadow acne, peter panning is aggravated when there is insufficient precision in the depth buffer. Calculating near planes and far planes also helps avoid peter panning. Figure A.5 shows an example of peter panning artifact.

Figure A.5: Example of peter panning artifact - typical 'detached' shadows.

A.6 Shadow map downsampling*

Downsampling of an image is the process of information reduction. Using lower resolution of shadow maps results in a loss of shadows accuracy but improves the computational performance. Game rendering benefits a lot from this technique. In order to maintain a fast refresh rate during game rendering, it is common practice to use the possibly smallest shadow maps. If the map used for generating shadows is too small, it appears on the screen as the jagginess of shadows' edges. This is similar to a supra-threshold artifact, but since it is localized only on the shadows' edges it is quite difficult to notice, especially without any reference image. An example of artifact caused by too low shadow map resolution is shown in Figure A.6.

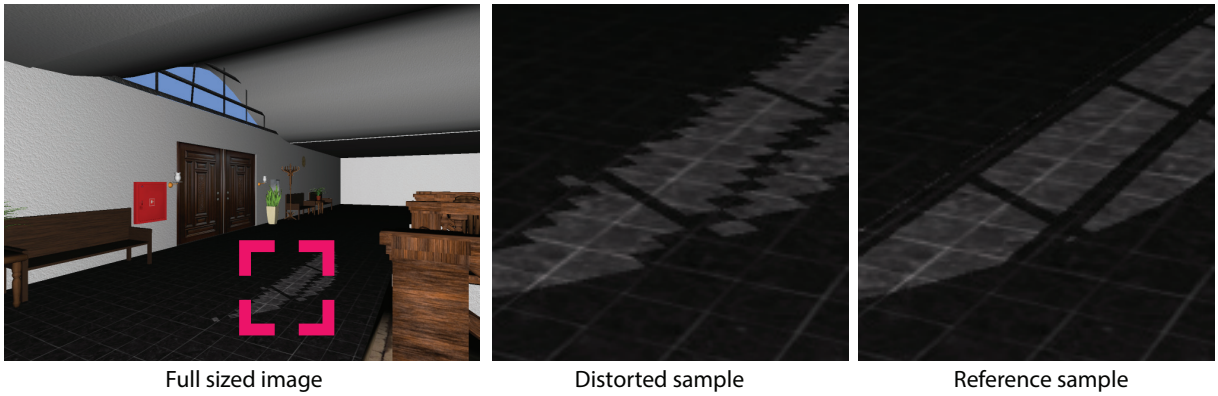


Figure A.6: Example of jagged shadows' edges as typical artifact that appears when the shadow map resolution is too low.

A.7 Z-fighting

Z-fighting (or stitching) is a 3D rendering effect that happens where two or more primitives have close values in the z-buffer. This causes annoying flickering issues since one primitive can be displayed in front of or behind the other inconsistently. Several techniques as increasing depth buffer resolution or changing slightly the position of the objects can mitigate the problem. Since it is usual in game engines to deal with very complex scenes with many objects, it is quite common to have this kind of artifact. All 10 images were render in Unity or CryEngine and come from [101]. Figure A.7 shows an example of z-fighting artifact.

Figure A.7: Example of Z-fighting artifact caused by small precision of the depth buffer.

A.8 Compression*

Compression dataset consists of 71 images and contains distortions due to experimental low-complexity image compression, operating at several bit-rates. Compression artifacts are the most common ones in computer graphics. Too low quality settings of compression result in very well known blockiness or mosaic artifact (Figure A.8) which has a great impact on the overall image quality. The distortion appears globally on the whole image and its visibility depends on the local image content. Usually it is well seen on gradients and can be easily masked by some specific frequencies. This set is an important source of near-threshold distortions.

Figure A.8: Example of blockiness effect - compression artifact.

A.9 Deghosting

High dynamic range (HDR) images become very popular in the recent years. Merging multiple exposures is a common method for generating HDR images. During acquisition, in the presence of a dynamic scene, non static objects can cause ghosting artifacts. Usually

deghosting algorithms replace detected the motion pixels either with pixels from only one exposure, or from multiple exposures. The main drawback of these methods is the reduction of high dynamic range of the moving object and local color or brightness deviation (Figure A.9). Some other common artifacts that could be introduced by the deghosting process are motion artifacts and noise. This set of artifacts consists of either supra-threshold and near-threshold distortions. Images come from [136].

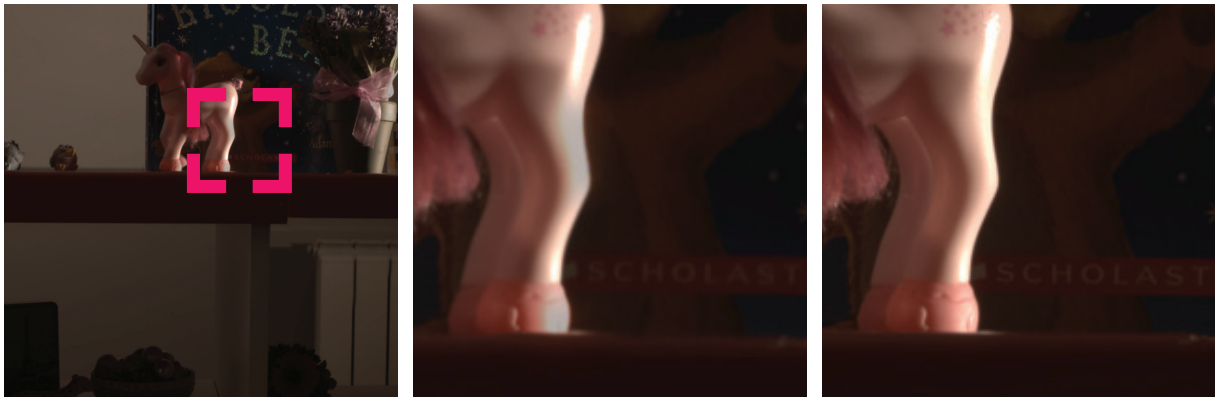


Figure A.9: Example shows local color and brightness deviations.

A.10 IBR

This set contains typical optical flow warping artifacts (Figure A.10) and small shifts caused by nearest neighbor warping methods. Optical flow warping usually results in deformation of the objects, slight ghosting and discontinuities. The images created with NN method do not contain any artifacts, but they are slightly shifted according to the reference image. This effect is almost unnoticeable for a human, even when he compares the testing image with the reference one. This kind of distortions were prepared to make our metric invulnerable to slightly misaligned images. The images come from dense light-field camera acquisition, followed by a process of images reduction and subsequently a reconstruction process with an optical flow or a nearest neighbor (NN) policy. NN images have only one distortion level and optical flow ones have three of them. All the images originate from [137].

Full sized image | Distorted sample | Reference sample

Figure A.10: Double edges and discontinuities caused by optical flow warping method.

A.11 CGIBR

This set consists 5 rendered images and one photography that contains optical flow and linear warping artifacts (ghosting and discontinuity presented in Figure A.11). Ghosting artifact results in the image as the objects with double edges and semi-transparent areas between those edges. In this case objects closer to the camera have stronger ghosting effect than objects in background. Each image has only one distortion level. All images of this set come from [137].

Full sized image | Distorted sample | Reference sample

Figure A.11: Example of ghosting as typical artifact of linear warping.

A.12 TID2013*

In addition to the 296 newly marked images, we added 261 images from the TID2013 image quality dataset [99], for which we could automatically generate marking. We selected from that dataset a subset of images that did not contain noticeable differences and assigned them marking maps set to 0s (no user markings). Then we selected another subset with well-noticeable distortions and set corresponding marking maps to 1s (distortions visible in the entire image). To ensure that both subsets were correctly selected, we

compared the four least severe distortion levels with the reference images in an additional pairwise comparison experiment (comparisons missing in the original dataset) and scaled the original (per-observer) pairwise data together with additional measurements using methods described in [103] and assuming Thurstone Case V observer model. Then, we selected for the first subset the images with the score of less than 0.2 just-objectionable-difference (JOD) to the reference, and for the second subset the images with the difference larger than 3 JODs. We also excluded the distortion types that affected only small image regions, such as JPEG transmission errors, and left the distortions that affected all pixels.

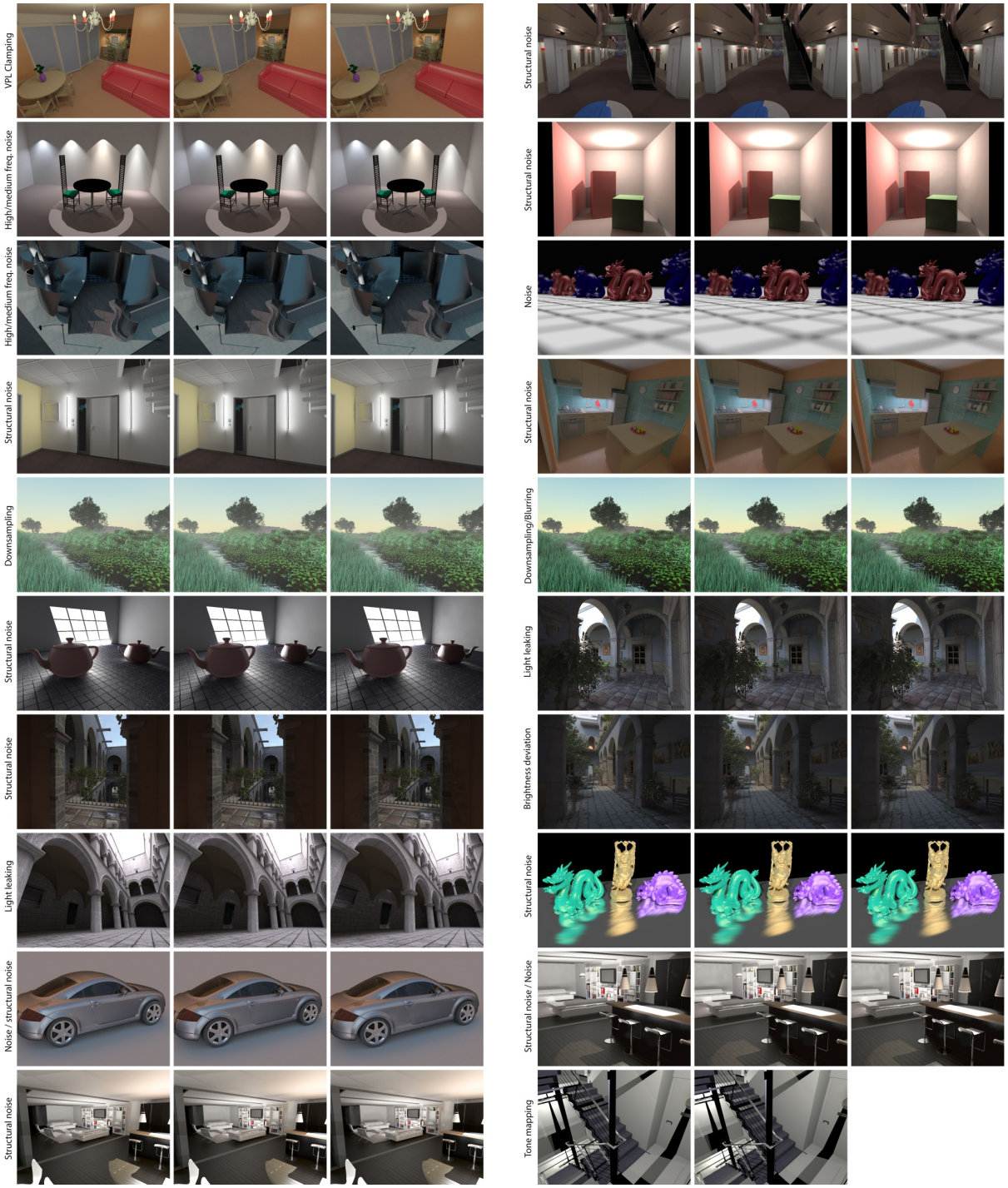


Figure A.12: Mixed subset.

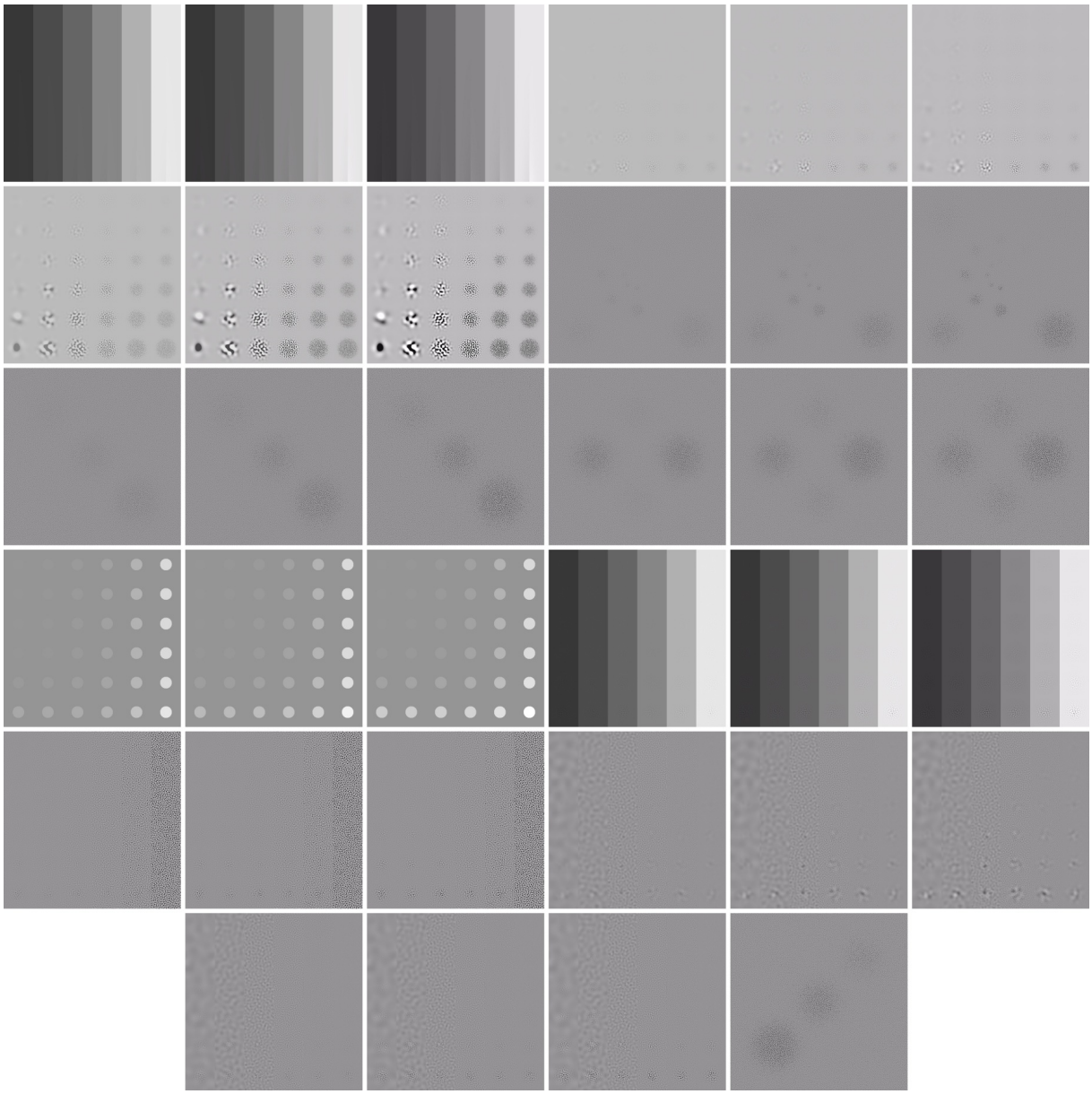


Figure A.13: Perception patterns subset.

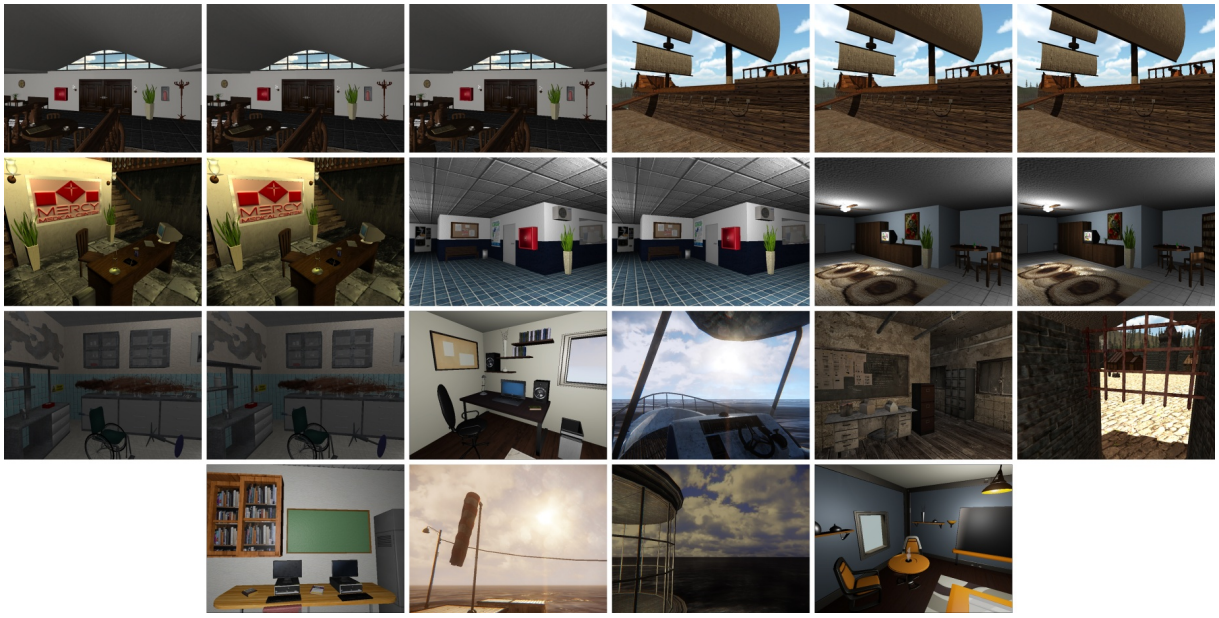


Figure A.14: Aliasing subset.

Figure A.15: Shadow acne subset.

Figure A.16: Peter panning subset.

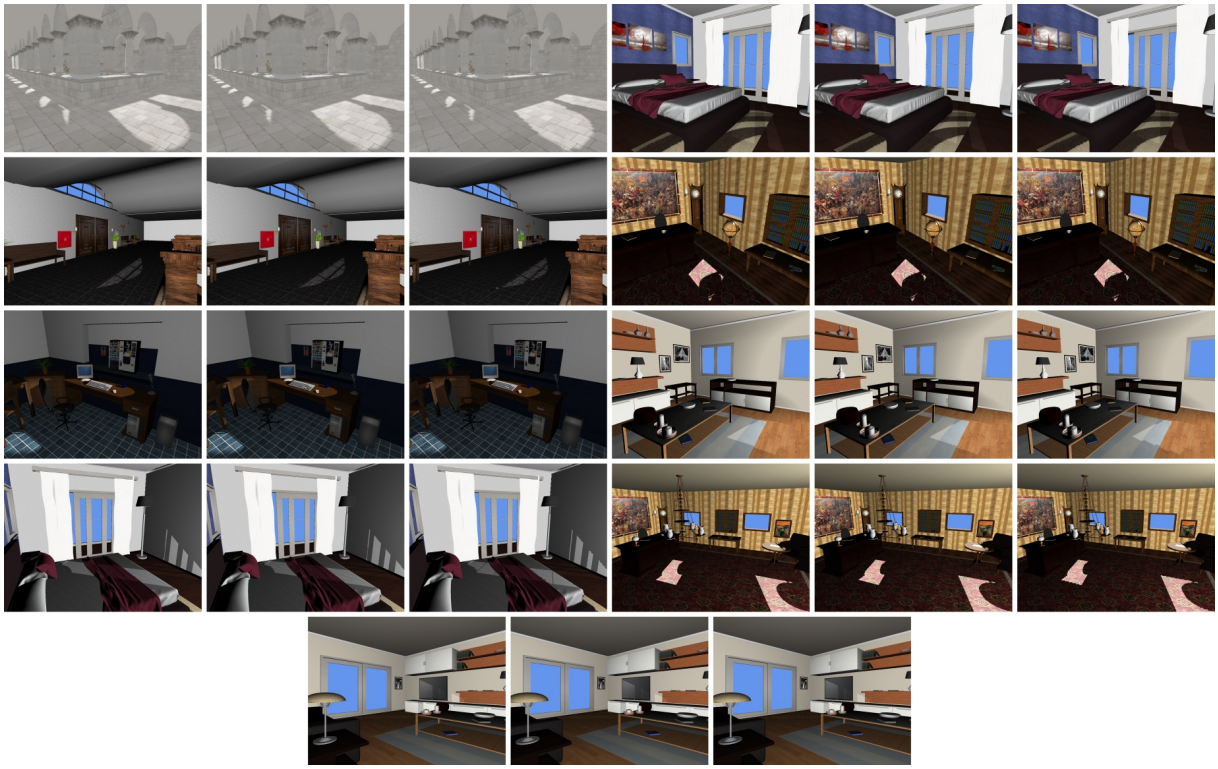


Figure A.17: Shadow map downsampling subset.

Figure A.18: Z-fighting subset.

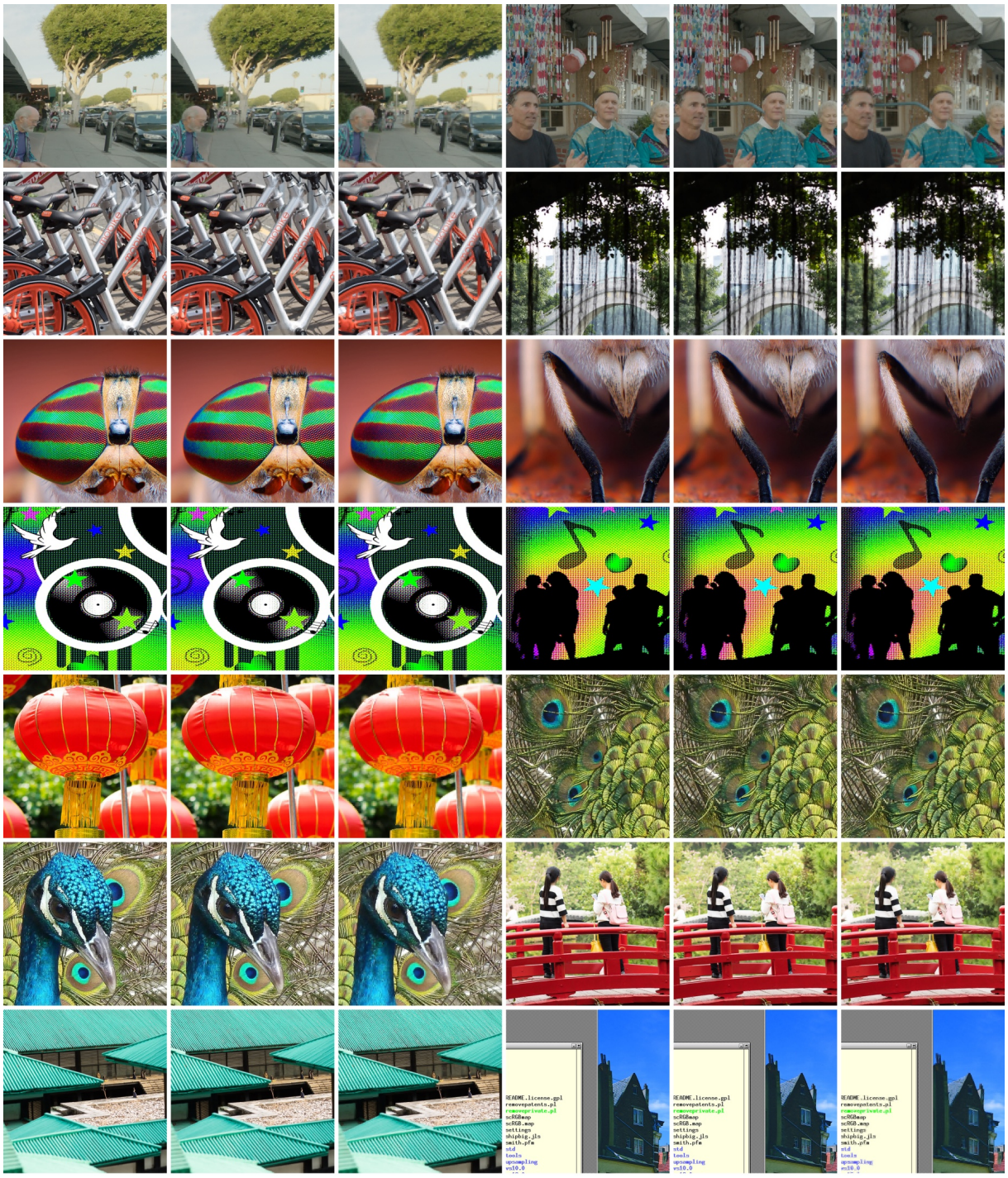


Figure A.19: Compression subset (continued on the next page).

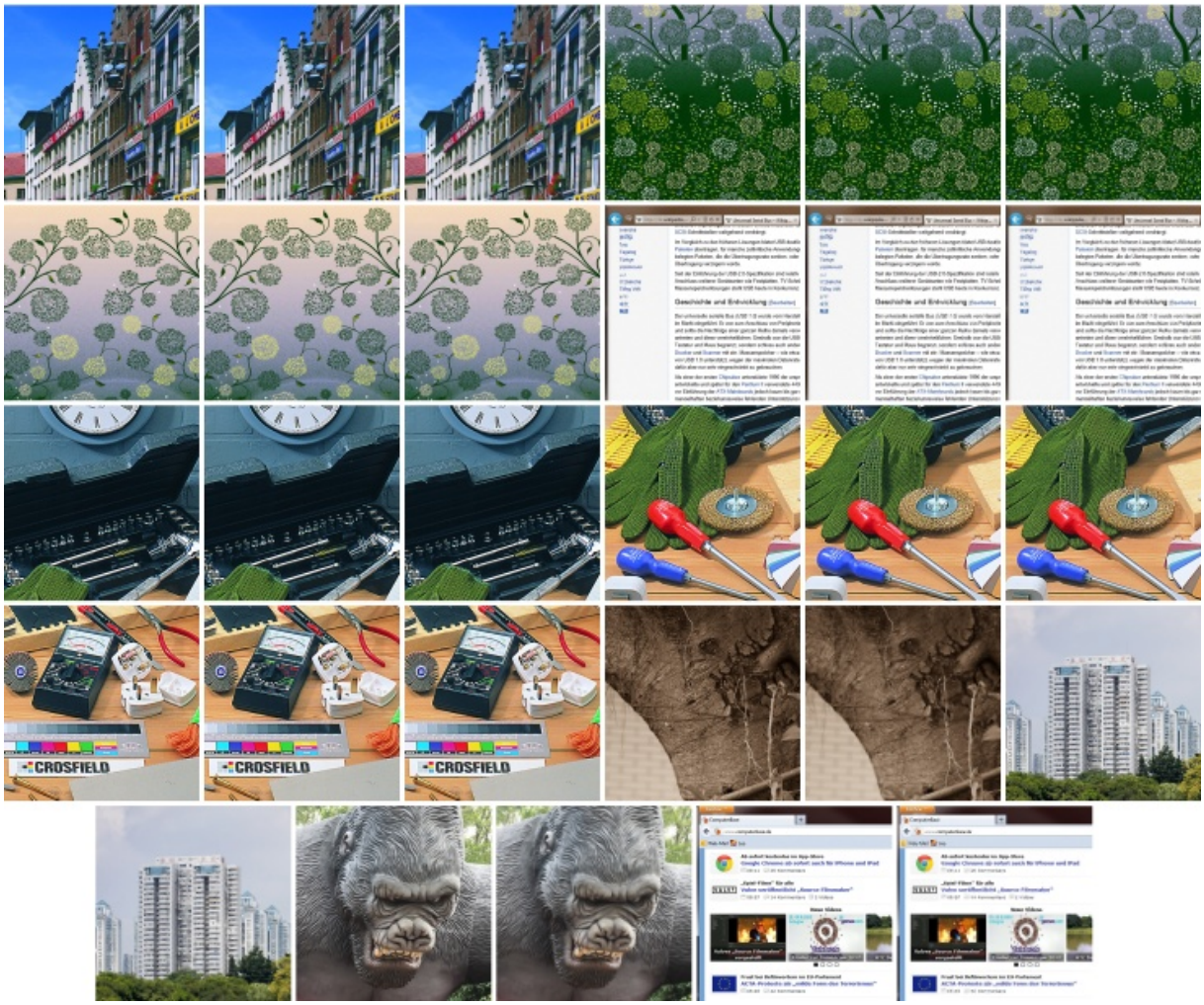


Figure A.19: Compression subset. (cont.)

Figure A.20: Deghosting subset.

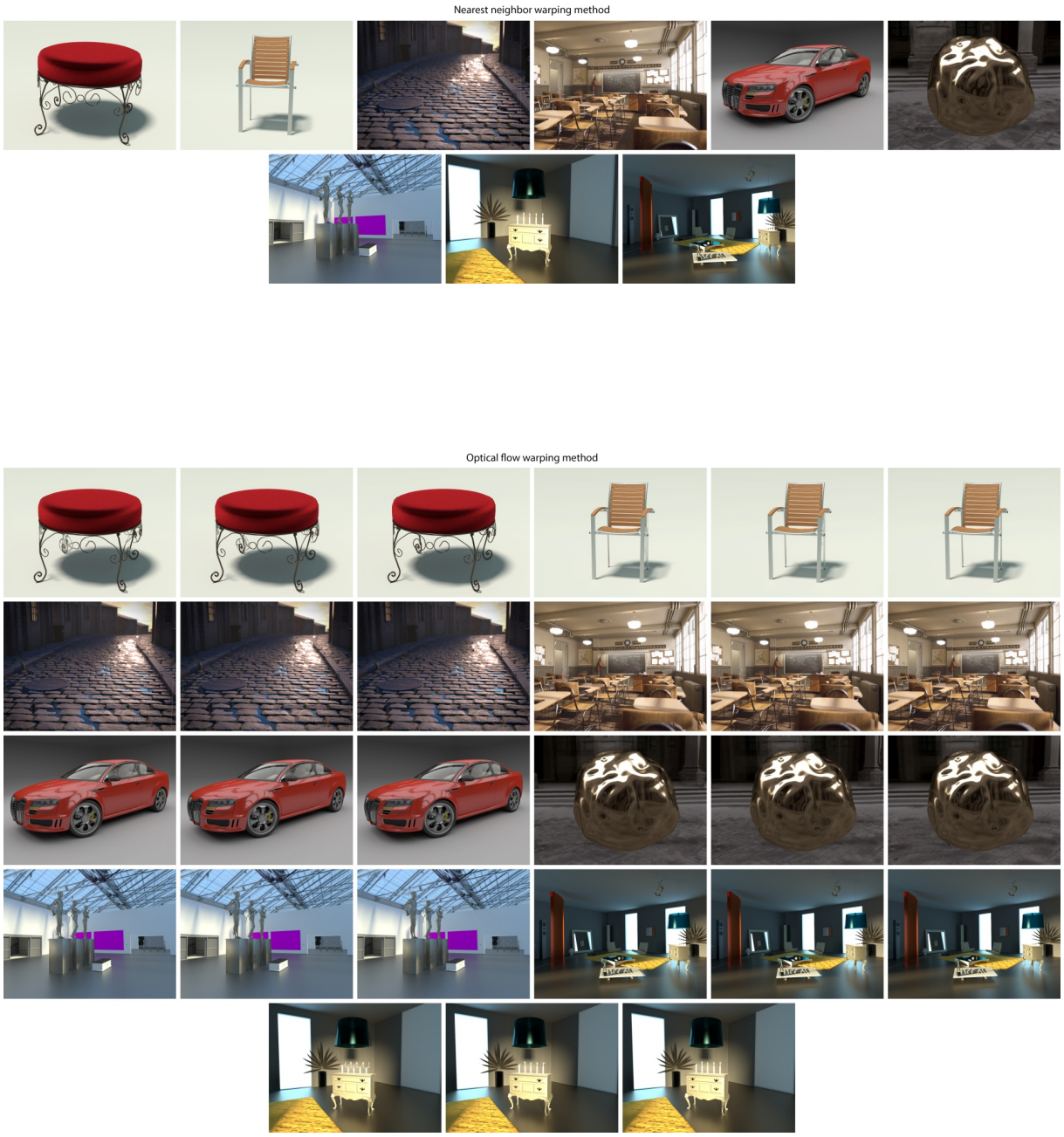


Figure A.21: IBR subset.

Figure A.22: CGIBR subset.

appendix B

Visually lossless image compression demonstrations

We demonstrate the utility of the CNN visibility metric in two applications: benchmarking of lossy image compression and visually lossless image compression. In Figure B.1, the figure shows that WebP can encode low bit-rate images with less noticeable artifacts than JPEG. However, the advantage of WebP is lost at higher bit-rates. The coding artifacts are more visible for JPEG at lower bit-rates as shown in Figure B.1. For example, in the first row, the background compression artifacts are more visible in the JPEG compressed image than in the WebP compressed image. However, it is more difficult to spot the difference at higher bit-rates, but examples Figure B.2 show slightly richer textures (pay attention to the fur and the hair) and more saturated colors in JPEG images.

Another application is visually lossless image compression as shown in Figure B.3. This example demonstrates that with more accurate visibility metrics, we can achieve visually lossless image compression with lossy image compression methods and save a lot of storage space compared with their default settings that set a fixed compression parameter for visually lossless image compression.

A JPEG Pdet:0.99 BPP:0.47 A WebP Pdet:0.41 BPP:0.46

B JPEG Pdet:0.99 BPP:0.47 B WebP Pdet:0.45 BPP:0.45

C JPEG Pdet:0.96 BPP:0.46 C WebP Pdet:0.38 BPP:0.46

Figure B.1: Comparison between JPEG and WebP for lower bit-rates. Best seen on the screen.

D JPEG Pdet:0.02 BPP:1.97 D WebP Pdet:0.15 BPP:1.96

E JPEG Pdet:0.18 BPP:1.98 D WebP Pdet:0.34 BPP:1.99

F JPEG Pdet:0.05 BPP:1.95 F Pdet:0.34 WebP BPP:1.97

Figure B.2: Comparison between JPEG and WebP for higher bit-rates. Best seen on the screen.

Figure B.3: The pairs of reference and compressed images in which the compression quality was adjusted using the proposed metric to be at the visually lossless level. The values in parenthesis denote saving as compared to JPEG and WebP with the fixed quality of 90. Best seen on the screen.