Image quality assessment using artificial neural networks

Alexander Havstad
Edith Cowan University

Follow this and additional works at: https://ro.ecu.edu.au/theses

Part of the Engineering Commons

Recommended Citation

This Thesis is posted at Research Online.
https://ro.ecu.edu.au/theses/664
Edith Cowan University

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorize you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

- Copyright owners are entitled to take legal action against persons who infringe their copyright.

- A reproduction of material that is protected by copyright may be a copyright infringement. Where the reproduction of such material is done without attribution of authorship, with false attribution of authorship or the authorship is treated in a derogatory manner, this may be a breach of the author’s moral rights contained in Part IX of the Copyright Act 1968 (Cth).

- Courts have the power to impose a wide range of civil and criminal sanctions for infringement of copyright, infringement of moral rights and other offences under the Copyright Act 1968 (Cth). Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
IMAGE QUALITY ASSESSMENT USING ARTIFICIAL NEURAL NETWORKS

By
Alexander Havstad

A thesis submitted for the degree of
Master
at
School of Engineering and Mathematics
Edith Cowan University

Principal Supervisor : Abdesselam Bouzerdoum

March 2005
USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.
DECLARATION

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Signature: 

[Redacted]

1/6/05
Acknowledgements

I would like to thank Professor Abdesselam Bouzerdoum for an interesting project and for all his time spent on this project.

I would also like to thank all the people from the lab, who helped me and gave me feedback on my project.
Abstract

Image quality assessment (IQA) is an important component of image processing and vision algorithms. Image quality is subjective and difficult to assess due to the complex interactions that take place within the human visual system (HVS). The most commonly used measures, such the mean square error (MSE) and the peak-signal-to-noise ratio (PSNR), are too simple to account for the complex characteristics of the HVS. Some subjective image assessment methods do exist; however, these are too complex, nondeterministic, and time consuming to compute. Therefore, it is desirable to find reliable objective measures that can approximate perceptual image quality with high fidelity and low computation burden.

The goal of this study is to find an objective IQA method that reflects well the way humans perceive image quality. To this end, neural networks are developed to approximate two different types of image quality measures: an objective measure and a subjective measure. Furthermore, two types of input encoding schemes are investigated: one uses statistical features and the other uses raw pixel values as inputs to the neural network. Several metrics were used to assess the performance of the trained networks; they include linear correlation, rank-order correlation and ANOVA. The performances of the neural networks are compared to those of existing objective IQA techniques, in particular the mean structural similarity (MSSIM) index. Experimental results show that the predicted image quality with neural networks is more consistent with perceptual image quality than existing objective measure, such as the MSSIM.
## Contents

Acknowledgements iv

Abstract v

1 Introduction 1
   1.1 Thesis Objective and Contributions 2
   1.2 Thesis Structure 3

2 Image Quality Assessment Methods 4
   2.1 Introduction 4
   2.2 Subjective Versus Objective Measures 5
   2.3 Pixel-Difference Based Measures 6
   2.4 Correlation-Based Measures 8
   2.5 Edge-Based Measure 10
   2.6 Spectral Distance-Based Measures 11
   2.7 Context-Based Measures 11
   2.8 Human Visual System-Based Measures 12
   2.9 Mean Opinion Score 13
   2.10 Structural Similarity-Based Measures 14
   2.11 Conclusion 17

3 Analysis of Objective Image Quality Measures 18
   3.1 Introduction 18
   3.2 Objective Measures versus Perceptual Quality 18
   3.3 Comparison of Objective Image Quality Measures 20
      3.3.1 Comparison using Linear Correlation 21
      3.3.2 Comparison using Polynomial Fitting 22
      3.3.3 Comparison using Analysis of Variance 22
4 Neural Networks: A Review and Application to IQA

4.2 An Overview of Artificial Neural Networks

4.3 Multilayer Feedforward Neural Networks

4.4 Training the MLP

4.4.1 Gradient-Descent and Error-Backpropagation Algorithm

4.4.2 Conjugate Gradient Algorithms

4.4.3 Levenberg-Marquardt Algorithm

4.5 Estimating the MSSIM Using Neural Networks

4.5.1 Data Preparation

4.5.2 Statistical Features-based Neural Network

4.5.3 Pixel-based Neural Network

5 Predicting the MOS Using Neural Networks

5.2 Data Preparation

5.2.1 Data for Feature-based Approach

5.2.2 Data for Pixel-based Approach

5.3 Feature-based Neural Network Approach

5.3.1 Selection of Feature Vector

5.3.2 Selection of Network Size

5.3.3 Analysis and Comparison

5.4 Pixel-based Neural Network Approach

5.5 Comparison of NN Approaches and MSSIM

5.5.1 Results and Analysis for Set 1

5.5.2 Results and Analysis for Set 2

5.6 Conclusion
### List of Figures

2.1 Classification of image quality assessment methods. .................. 6  
2.2 Original and distorted images used for image quality assessment. . 9  
2.3 Contrast sensitivity function. ........................................... 13  
2.4 Original and distorted images and their IQ scores. .................... 16  
3.1 Original and distorted images. ........................................... 19  
3.2 Third-order polynomial fitting of MOS and objective IQ measures. 23  
3.3 Box plots of six image quality measures: MOS, MSSIM, UIQI, PSNR, SNR, MSE. .................................................. 26  
4.1 Block diagram of a single artificial neuron. ............................. 30  
4.2 Some common activation functions. .................................... 31  
4.3 Multilayer Feedforward Neural Network, known as the MLP. ........ 32  
4.4 Test error versus the number of neurons. ............................. 38  
4.5 Actual MSSIM versus the network-prediction for images added with noise. ........................................... 39  
4.6 Actual versus predicted MSSIM scores (feature-based network). .... 40  
4.7 Training error versus the number of neurons. .......................... 41  
4.8 Actual MSSIM versus the network-prediction for images added with noise. ........................................... 42  
4.9 Actual versus predicted MSSIM scores (pixel-based network). ....... 43  
5.1 Mean and maximum error based on the test set for different feature vectors. ........................................... 51  
5.2 Correlation coefficients based on the test set for different feature vectors. 51  
5.3 Mean and max error for the test set depending on different network sizes. ........................................... 54
5.4 Correlation coefficients for the test set depending on different network sizes. ........................................ 54
5.5 Comparison of the network-predicted MOS and the actual MOS for the training set. ........................................ 56
5.6 Comparison of the network-predicted MOS and the actual MOS for the test data. ........................................ 56
5.7 Comparison of the actual MOS and the network-predicted MOS using ANOVA. ........................................ 57
5.8 Pixel-based ANN approach: mean and maximum errors on the test set for different network sizes. ........................................ 58
5.9 Comparison of the network-predicted MOS and the actual MOS for the training data. ........................................ 59
5.10 Comparison of the network-predicted MOS and the actual MOS for the test data. ........................................ 59
5.11 Comparison of the actual MOS and the network-predicted MOS using ANOVA. ........................................ 61
5.12 Comparison of polynomial-approximated MOS and the actual MOS. ........................................ 62
5.13 Different image quality groups according to the MOS and polynomial MSSIM scores. ........................................ 63
5.14 Four distorted images. ........................................ 65
List of Tables

2.1 Image quality scores from experiment with MSE, MAD, PSNR and SNR. .................................. 8
2.2 Image quality scores from experiment with correlation-based measures based on the images shown in Figure 2.2. ................. 10
3.1 Objective IQ scores of test images. .............................................. 20
3.2 Correlation coefficients between objective IQ measures and the MOS. 21
3.3 Residual error of third-order polynomial fitting of the MOS. .......... 22
3.4 Means and standard deviations of image quality groups for six IQ measures. ............................................. 25
3.5 F-values from the ANOVA simulation. ........................................... 27
4.1 MSSIM prediction of features-based neural network for Test Set 1 (Gaussian, salt & pepper noise) .................................. 39
4.2 MSSIM prediction performance of feature-based neural network. . 40
4.3 MSSIM prediction of pixel-based network for Test Set 1 (Gaussian and salt & pepper noise). ............................................. 42
4.4 MSSIM prediction performance of pixel-based neural network. ..... 43
5.1 Feature vectors for the feature-based ANN approach. ................. 48
5.2 Performance of feature vector 5: $[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, \text{MSE}]$. ................. 49
5.3 Performance of feature vector 6: $[\sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y, \text{MSE}]$. ................. 49
5.4 Performance of feature vector 7: $[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y]$. ................. 50
5.5 Performance of feature vector 8: $[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y, \text{MSE}]$. ................. 50
5.6 Performance of network size (6-6-6-1). ................................. 53
5.7 Performance of network size (6-10-5-1). ................................. 53
5.8 Performance indicators of the statistical feature-based network. ..... 55
5.9 Performance indicators of pixel-based network. ......................... 60
5.10 Comparison of five image quality measures: MOS, MSSIM, polynomial MSSIM (p-MSSIM), feature-based NN (FBNN), and pixel-based NN (PBNN) .................................................. 63

5.11 Five image quality measures: MOS, MSSIM, polynomial MSSIM (p-MSSIM), feature-based NN (FBNN), and pixel-based NN (PBNN) for Test Set 2 ........................................... 64

A.1 Performance of feature vector 1: \([\sigma_{xy}, \sigma_x, \sigma_y]\) ........................................ 74
A.2 Performance of feature vector 2: \([\sigma_{xy}, \sigma_x, \sigma_y, \text{MSE}]\) ......................... 75
A.3 Performance of feature vector 3: \([\sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y]\) .......................... 75
A.4 Performance of feature vector 4: \([\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y]\) ..................... 76
A.5 Performance of network size (6-4-4-1) ................................................................. 76
A.6 Performance of network size (6-5-3-1) ................................................................. 77
A.7 Performance of network size (6-8-6-1) ................................................................. 77
A.8 Performance of network size (6-8-8-1) ................................................................. 78
A.9 Performance of network size (6-6-8-1) ................................................................. 78
Chapter 1

Introduction

Image quality assessment (IQA) has always been an integral part of image processing. Many different approaches for IQA have been developed in the last three decades, each with a different level of complexity. Some approaches rely on simple mathematical formulae, while others attempt to include aspects of the human visual system (HVS) in the IQA model. Image and video coding are increasingly being used to minimize the amount of data needed to store or transmit images. Often, however, compression algorithms change the information in the image, and therefore a measure to quantify the amount of distortion or degradation is required.

Many measures that rely on simple mathematical formulae have been employed to quantify image degradation. However, if the end user of visual information is the human observer, simple mathematical measures are inadequate for IQA because they do not capture the essential characteristics of the HVS and can become unstable in the presence of significant image degradation. The perceived image quality depends on many factors, including spatial frequency, colour, contrast and luminance. For this reason, the perceived quality does not often correlate well with the amount of degradation in the image. Since it is not practicable to incorporate a human observer inside image processing algorithms, there is a real need for objective image quality measures that correlate well with the subjective IQA.

The aim of this research is to develop an objective measure that approximates the subjective image quality, without having recourse to complex mathematical models of the HVS. The human visual system relies on various types of information, some of which is not well understood, to quantify image quality. Many researchers have tried to model the complex process of visual perception, then incorporate the model into an image quality measure. However, the implementation of HVS models
is often a complex and time consuming process. Moreover, HVS-based measures require further development before they become consistent for all types of image degradations.

In this thesis, the potential of using artificial neural networks (ANNs) for IQA is investigated. Neural networks (NNs) have the capability to learn complex tasks from examples and generalize to previously unseen data; they have been introduced to emulate how the human brain functions [Haykin and Zurada]. Similar to a human brain, a NN needs to be trained before it can be employed to perform a given task; therefore, it requires a set of training data and a training algorithm to adjust the parameters of the network. Once the network has learnt the training data, it can be used to solve tasks related to the data used in the training process. Many different neural network architectures have been developed, each of which is suitable for a particular task. In this research, the feedforward network architecture, known as the multilayer perception (MLP), is used to emulate the process of image quality assessment by the HVS.

1.1 Thesis Objective and Contributions

The main objective of this thesis is to develop an objective image quality assessment method that can serve as a correlate to the HVS. To this end, neural networks are employed to learn image quality as perceived by human observers. The main contributions of this thesis are:

- A survey of existing IQA approaches and an analysis of their effectiveness, or otherwise, in providing consistent measures of image quality. The different approaches are compared in order to identify the most consistent with human assessment of image quality.

- A review of feedforward neural networks and an evaluation of their capability to measure image quality.

- Development of a systematic method for image quality assessment using neural networks. Two alternative methods are proposed: the first uses computable image attributes and the second uses raw pixel values as inputs to the neural network. The performance of the proposed NN quality measure is evaluated and compared with existing approaches.
1.2 Thesis Structure

This thesis comprises 6 chapters and one appendix, in addition to the bibliography. It is organized as follows:

- Chapter 1 presents a general introduction, the thesis objectives and contributions, and its general structure.

- Chapter 2 reviews existing IQA methods. The chapter provides a description of each method, together with its advantages and disadvantages. The discussion is illustrated by examples of original and distorted images and their IQA scores.

- Chapter 3 presents a statistical comparison of the various IQA methods. The goal of this comparison is to identify an objective measure that is consistent with the human perception of image quality. This comparative study is based on two types of correlation coefficients and the analysis of variance (ANOVA).

- Chapter 4 gives an introduction to NNs. An explanation of how they work, what they can be used for, and how to train them is given. NNs are a broad subject to cover; therefore, we focus only on the network architectures and training algorithms used in this project. In this chapter, neural networks are developed to predict two objective IQA measures; their performance is evaluated on an image database. The purpose of this is to establish whether or not a NN can learn an IQA measure from a limited set of examples.

- In Chapter 5, we develop several neural networks to approximate a subjective IQA measure. Two different approaches are presented: one approach uses statistical features as network inputs, and the other uses pixel values as network inputs. Both approaches are evaluated and compared with existing approaches. The purpose here is to test whether a neural network can learn, from a limited set of examples, the perceived image quality by human observers, without having recourse to complex modelling of the HVS.

- In Chapter 6, a summary of the results is presented, and ideas of further work are given.
Chapter 2

Image Quality Assessment Methods

2.1 Introduction

Image quality assessment (IQA) plays a very crucial role in image and video processing. The aim is to replace human judgment of perceived image quality with a machine evaluation. As a consequence, over the past three decades a large effort has been devoted to developing IQA measures that try to mimic human perception [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. While many methods and models still rely on simple measures, such as the peak-signal-to-noise-ratio (PSNR) and the mean-squared error (MSE), many others use sophisticated signal processing techniques, such as multi-channel filtering [4, 5], discrete cosine transform [7, 8], multi-scale Wavelet decompositions [9, 10], and Wigner-Ville distribution [11]. To date, however, it has been very difficult to find a reliable objective measure that correlates very highly with human perception [12].

Measuring image quality is an important, but also difficult task [12]. Many methods have been developed to solve this problem. They can be categorized into three approaches: full-reference IQA, no-reference IQA, and reduced-reference IQA. In the full-reference IQA, a copy of the original image is available, with which the distorted image is compared. In contrast, in the no-reference approach image quality is assessed based solely on the information content of the test image; that is, there is no reference image with which the test image can be compared. In the reduced-reference approach, only partial information about the original image is available. This thesis deals only with the full-reference approach, where the fidelity of a test image is computed based on features extracted from the reference and test images.

In this chapter we introduce several IQA methods, and explain the difference
between objective and subjective methods. Furthermore, we will illustrate the objective measures and highlight their inconsistencies with the subjective measures. This is done by showing original and distorted images, and comparing the image quality scores produced by different IQA methods.

2.2 Subjective Versus Objective Measures

There are two main classes of IQA metrics: objective and subjective methods. While objective methods attempt to quantify the amount of degradation present in the image using a well-defined mathematical model, subjective measures are based on evaluation by human observers. Since invariably the end user of visual information is the human observer, it is generally recognized that subjective IQA methods are the ultimate solution. However, subjective measures are difficult to design and time consuming to compute; furthermore, they cannot be readily incorporated into the design and optimization of image and video processing algorithms, such as compression and image enhancement. For this reason, there has been an increasing interest in objective IQA techniques that can automatically predict or approximate the perceived image quality.

Objective measures are explicitly defined according to computable image features. There are six classes of objective quality or distortion assessment methods [13]:

- Pixel difference-based measurement: peak signal-to-noise ratio (PSNR) and the mean-squared error.
- Correlation-based measures: correlation of pixels, or of the vector angular directions.
- Edge-based measures: displacement of edge positions or their consistency across resolution levels.
- Spectral distance-based measures: measuring the magnitude and/or phase spectral discrepancies.
- Context-based measures: penalties based on various functions of the multidimensional context probability.
• Human Visual System (HVS) based measures: measure image quality by incorporating aspects of the human visual system characteristics. The quality of an image, as perceived by a human, depends on many factors, such as contrast, color, spatial frequency and masking effects.

By far the most common objective IQA methods are the pixel difference-based metrics because they have low computational complexity, and can easily be incorporated into other image processing algorithms. They are also independent of the viewing conditions and the individual observers. However, such simple measures, which do not take into account the HVS characteristics, are not adequate for describing perceptual image quality. Other more sophisticated measures do exist, such as the Universal Image Quality index (UIQI) [14] and the Structural Similarity (SSIM) Index [15], which are better correlated with subjective image quality.

Figure 2.1 illustrates the different IQA classes; these will be described in more details in the remainder of this chapter.

![Figure 2.1: Classification of image quality assessment methods.](image)

### 2.3 Pixel-Difference Based Measures

These measures are defined at the pixel-level; they are fast and easy to compute. They include the mean square error (MSE), the signal-to-noise ratio (SNR) and the peak signal-to-noise ratio (PSNR). Pixel difference-based measures are by far the most common because they have lower computational complexity. However, it is generally recognised that energy-based measures do not correlate very well with the perceived subjective image quality.
The signal-to-noise ratio (SNR) is an IQA measure based on the pixel differences between the reference and distorted images [13][16]. It is the ratio of the signal energy to the noise energy. Let $f(m,n)$ be the reference image, and $\hat{f}(m,n)$ be the test or distorted image, and suppose that the two images are of size $M \times N$. The signal energy $E_s$ is the energy of the reference image, which is given by

$$E_s = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^2(m,n)$$ (2.1)

The noise energy is defined as the energy of the error signal, where the error signal $e(m,n)$ is the difference between the distorted image $\hat{f}(m,n)$ and the reference image $f(m,n)$:

$$e(m,n) = \hat{f}(m,n) - f(m,n)$$ (2.2)

Thus, the noise energy $E_n$ is given by

$$E_n = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e^2(m,n) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [\hat{f}(m,n) - f(m,n)]^2$$ (2.3)

The signal-to-noise ratio (SNR) is then given by

$$SNR = \frac{E_s}{E_n}$$ (2.4)

The SNR is usually measured in decibels (dB):

$$SNR_{dB} = 10 \log_{10} SNR = 10 \log_{10} \frac{E_s}{E_n},$$ (2.5)

where $\log$ denotes the logarithm base 10 function.

The mean square error (MSE) is defined as

$$MSE = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e^2(m,n)$$ (2.6)

Another measure that is related to the previous two measures is the peak-signal-to-noise ratio (PSNR), which is defined as

$$PSNR_{dB} = 10 \log_{10} \left[ \frac{L^2}{MSE} \right]$$ (2.7)

where $L$ is the dynamic range of the pixel values (e.g. $L = 255$ for an 8-bit image).

Sometimes, however, image quality is measured using the mean absolute difference (MAD) defined as

$$MAD = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |e(m,n)| = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |\hat{f}(m,n) - f(m,n)|$$ (2.8)
It is well known, however, that the MSE, MAD, PSNR, and SNR do not correlate very well with perceptual distortion. Furthermore, these measures are not consistent amongst themselves. Figure 2.2 shows three original images and three distorted images all of which were corrupted with the same additive white Gaussian noise (AWGN). The four image quality measures described above are presented in Table 2.1. This table shows that while the MSE and MAD, which depend only the amount of noise added, are the same in all three cases, the SNR and PSNR vary depending on the image content.

Table 2.1: Image quality scores from experiment with MSE, MAD, PSNR and SNR.

<table>
<thead>
<tr>
<th>Test Image</th>
<th>MSE</th>
<th>MAD</th>
<th>SNR</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.2(b)</td>
<td>24.95</td>
<td>3.98</td>
<td>29 db</td>
<td>34 db</td>
</tr>
<tr>
<td>Figure 2.2(d)</td>
<td>24.95</td>
<td>3.98</td>
<td>27 db</td>
<td>34 db</td>
</tr>
<tr>
<td>Figure 2.2(f)</td>
<td>24.95</td>
<td>3.98</td>
<td>16 db</td>
<td>22 db</td>
</tr>
</tbody>
</table>

2.4 Correlation-Based Measures

Correlation is often used to measure the similarity between two signals. In IQA, correlation is used to quantify the similarity between the reference and test images [13] [17]. The cross-correlation (at zero lag) between two signals \( x \) and \( y \) is defined as

\[
r_{xy} = \sum_{n=-\infty}^{\infty} x(n)y(n)
\]  

(2.9)

The autocorrelation is the correlation of a signal with itself:

\[
r_{xx} = \sum_{n=-\infty}^{\infty} x(n)x(n)
\]  

(2.10)

There are three common correlation-based IQA measures:

- The ratio of autocorrelations of original and distorted images:

\[
C_1 = \frac{r_{ff}}{r_{jj}} = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{f}(m,n)^2}
\]  

(2.11)

- The ratio of original-distorted cross-correlation to autocorrelation of original image:

\[
C_2 = \frac{r_{j\hat{f}}}{r_{ff}} = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\hat{f}(m,n)}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)^2}
\]  

(2.12)
Figure 2.2: Original and distorted images used for image quality assessment.
• The Czekanowski distance:

\[
C_3 = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left( 1 - \frac{2 \times \min(f(m, n), \hat{f}(m, n))}{(f(m, n) + \hat{f}(m, n))} \right)
\]  

(2.13)

The same pairs of original and distorted images as used in Figure 2.2 are tested with correlation measures. Table 2.2 shows the quality scores given by these measures. It shows that the Czekanowski distance and the ratio of the autocorrelation for the original and distorted images are highly dependent on the image content.

Table 2.2: Image quality scores from experiment with correlation-based measures based on the images shown in Figure 2.2.

<table>
<thead>
<tr>
<th>Test Image</th>
<th>(C_1)</th>
<th>(C_2)</th>
<th>(C_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.2(b)</td>
<td>0.9986</td>
<td>0.9979</td>
<td>0.9413</td>
</tr>
<tr>
<td>Figure 2.2(d)</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.9999</td>
</tr>
<tr>
<td>Figure 2.2(f)</td>
<td>0.9772</td>
<td>0.9781</td>
<td>0.8617</td>
</tr>
</tbody>
</table>

### 2.5 Edge-Based Measure

This measure is based on the displacement of edge points, or their consistency across different resolutions. Edges usually correspond to object or region boundaries, and therefore can be used for image quality assessment [13] [18]. Edge discontinuities, decreases in edge sharpness, offsets of edge positions, missing edge points and falsely detected edge points are some examples of edge degradations. Edge-based measures consist of two steps: finding the edges and measuring the image quality based on those edges. Image edges are found by applying edge operators, such as the Canny or Sobel operators, and then thresholding the edge magnitude [16]. Once the edges are found, the image quality can be computed using the following expression [19]:

\[
E = \frac{1}{\max\{n_d, n_o\}} \sum_{i=0}^{n_d} \frac{1}{1 + a d_i^2}
\]

(2.14)

where \(n_d\) and \(n_o\) are the respective numbers of edge pixels in the distorted image and the original image, \(d_i\) is the distance from the \(i^{th}\) edge pixel in the distorted image to the nearest edge pixel in the original image, and \(a\) is a constant.
2.6 Spectral Distance-Based Measures

Spectral distance-based measures use the Discrete Fourier Transform (DFT) to quantify the difference between the original and the distorted image [13] [20]. The 2-D DFT of an image \( f(m, n) \) is defined as

\[
F(u, v) = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \exp[-2\pi i \left( \frac{mu}{M} + \frac{nv}{N} \right)]
\]

(2.15)

The magnitude and phase spectra are given by, respectively,

\[
M(u, v) = |F(u, v)|
\]

(2.16)

and

\[
\varphi(u, v) = \tan^{-1} \frac{I(u, v)}{R(u, v)}.
\]

(2.17)

where \( F(u, v) = R(u, v) + iI(u, v) \). Both spectra can be used for IQA. The magnitude and phase distortions are defined, respectively, as follows:

\[
S_1 = \frac{1}{M \times N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |M(u, v) - \hat{M}(u, v)|^2
\]

(2.18)

and

\[
S_2 = \frac{1}{M \times N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |\varphi(u, v) - \hat{\varphi}(u, v)|^2.
\]

(2.19)

An image quality score can be defined based on a weighted sum of the phase and magnitude spectral distortions:

\[
S_3 = \frac{\lambda}{M \times N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |\varphi(u, v) - \hat{\varphi}(u, v)|^2 + \frac{(1-\lambda)}{M \times N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |M(u, v) - \hat{M}(u, v)|^2
\]

(2.20)

where \( \lambda \) is a constant (0 \leq \lambda \leq 1).

2.7 Context-Based Measures

In all the methods described so far, a measure of image quality is determined through pixel-wise comparison. Instead of comparing the reference and test images pixel-wise, pixel neighborhoods are compared. Using context-based measures, the degradation of information within the pixel neighbourhood is taken into consideration.
[13]. Statistical information of the context probabilities, also called probability mass function \((pmf)\) of pixel neighbourhood, gives a good indication of the image quality. However, context-based measures require huge amount of computation, and hence have limited applications. Rate-distortion-based measures are a well known method where the changes in the context probabilities are quantified in the relative entropy, defined as follows:

\[
D(p||\hat{p}) = \sum_{X \in X^s} p(X) \log \frac{p(X)}{\hat{p}(X)}
\]  
(2.21)

where \(X^s\) is a pixel neighborhood, \(X = [x_1, \cdots, x_s]\) is a random vector, and \(p\) and \(\hat{p}\) are the pmf’s of the original and the distorted image contexts. The entropy for two different images has two different values, and therefore a reference is needed. To get a reference, two distorted images from the same original image are used; a comparison of the difference between the distorted images gives us the image quality. This is defined as

\[
Q = D(p||\hat{p}_1) - D(p||\hat{p}_2)
\]  
(2.22)

### 2.8 Human Visual System-Based Measures

Another approach is to use the characteristics of the Human Visual System (HVS) to measure image quality [21], [22], [23], [13], [24], [25]. This approach uses the HVS as a reference — through the eye, the human visual system typically looks at contrast, colour, frequency changes and masking effects when it measures the image quality. The two images are normally transformed into the frequency domain — the HVS is more sensitive to noise in some frequencies than others. Two techniques are normally used to transform the images into the frequency domain: Discrete Fourier transform (DFT) and Wavelet transform. After the images are transformed into the frequency domain, a filter — the contrast sensitivity function (CSF) — is applied to the original and the distorted images. The CSF has a band-pass characteristic similar to the spectral response of the HVS. An example of band-pass filter is defined in the frequency domain as follows:

\[
H(w) = \begin{cases} 
0.05e^{0.554w} & w < 7 \\
e^{-9\log_{10}w - \log_{10}9} & w \geq 7 
\end{cases}
\]  
(2.23)

where \(w = (u^2 + v^2)^{1/2}\) and \(u\) and \(v\) are the spatial frequencies. Figure 2.3 shows a plot of \(H(w)\). The filter \(H(w)\) removes the frequencies to which the human eye
is less sensitive. Different IQA measures can be defined based on the filter output. The simplest measure is the MSE. Among the objective measures discussed so far, the HVS-based measures are the closest to subjective measures. Furthermore, many other aspects of the HVS, such as masking effects, remain not very well understood and their modelling needs to be improved.

2.9 Mean Opinion Score

The mean opinion score (MOS) is the most common subjective measure [12]. A group of people is asked to visually compare an original image with a degraded image and estimate the image quality of the degraded image; the mean score of the group is taken as the image quality index. Normally, the MOS score is ranged between [1 – 5] or [1 – 10], but in this work a scaling between [1 – 100] is used to compare this work with the results given by Wang. While MOS reflects more faithfully human perception, it is time consuming and impractical to use in conjunction with other image processing algorithms. For this reason, there is strong interest in developing objective methods that correlate very well with the subjective assessment.
2.10 Structural Similarity-Based Measures

In 2000, Wang and Bovik proposed a new image quality measure, called the universal image quality index (UIQI) [14], where the comparison between the reference and test images is broken down into three similarity measures: luminance, contrast, and structural comparisons. The luminance comparison \( l(x, y) \) between a reference image \( X \) and a test image \( Y \) is described by

\[
l(x, y) = \frac{2\mu_x \mu_y}{\mu_x^2 + \mu_y^2}
\]  

(2.24)

where the mean values of the images \( X \) and \( Y \) are defined as

\[
\mu_x = \frac{1}{T} \sum_{i=1}^{T} x_i
\]

(2.25)

\[
\mu_y = \frac{1}{T} \sum_{i=1}^{T} y_i
\]

(2.26)

where \( T \) is the total number of pixels in image \( X \) or \( Y \). The contrast comparison \( c(x, y) \) is defined as

\[
c(x, y) = \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2},
\]

(2.27)

where the standard deviations of \( X \) and \( Y \) are defined as

\[
\sigma_x^2 = \frac{1}{T-1} \sum_{i=1}^{T} (x_i - \bar{x})^2
\]

(2.28)

\[
\sigma_y^2 = \frac{1}{T-1} \sum_{i=1}^{T} (y_i - \bar{y})^2.
\]

(2.29)

The structural comparison is given by

\[
s(x, y) = \frac{\sigma_{xy}}{\sigma_x \sigma_y}
\]

(2.30)

where the covariance of \( X \) and \( Y \) are defined as

\[
\sigma_{xy} = \frac{1}{T-1} \sum_{i=1}^{T} (x_i - \bar{x})(y_i - \bar{y}).
\]

(2.31)

Based on these three comparison measures, the UIQI is defined as

\[
UIQI(x, y) = l(x, y)c(x, y)s(x, y) = \frac{4\mu_x \mu_y \sigma_{xy}}{(\mu_x^2 + \mu_y^2)(\sigma_x^2 + \sigma_y^2)}.
\]

(2.32)

To compute the overall Universal Image Quality Index (UIQI)[14], the reference and test images are divided into overlapping square blocks (a common block size...
is \( T = 8 \times 8 \) pixels). For each two corresponding blocks, the UIQI is computed according to (2.32). The overall index is found by averaging the indices of the individual blocks.

\[
\text{UIQI} = \frac{1}{P} \sum_{j=1}^{P} Q_j,
\]

(2.33)

where \( P \) is the number of \( 8 \times 8 \) blocks, and \( Q_j \) is the UIQI of the \( j \)th block. The dynamic range of UIQI is \([-1, 1]\), with 1 representing a perfect quality.

The UIQI is a simple measure, which depends solely on first and second order statistics of the reference and test images. It is independent of the viewing conditions, and its computation is straightforward. However, it is somewhat unstable, especially at uniform areas, where the denominator term is very small. Furthermore, the UIQI doesn’t correlate well with subjective assessment. For example, if the distorted image is a blank image (all pixels = 0), the UIQI is zero (Figure 2.4(b)). If the distorted image is the pixel-wise inversion of the original image, the UIQI becomes -0.7630 for Figure 2.4(c) and 1 for Figure 2.4(d). This may not be acceptable in applications where the inverted image is considered to have more information (e.g., edges) compared to the blank image, or have the same quality scores equally to the original.

In order to alleviate the problem of stability and improve the correlation between the objective and subjective measures, Wang et al. [15] proposed the structural similarity index (SSIM) as an improvement to the UIQI. The SSIM has been defined as follows [15]:

\[
\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}
\]

(2.34)

The constants \( C_1 \) and \( C_2 \) are given by \( C_1 = (K_1 L)^2 \) and \( C_2 = (K_2 L)^2 \) where \( L \) is the dynamic range of the pixel values (\( L = 255 \) for 8-bit images), and \( K_1 \) and \( K_2 \) are two small positive constants. \( K_1 \) and \( K_2 \) are used to avoid instability when \( \mu_x^2 + \mu_y^2 \) is very close to zero.

At every pixel \((i, j)\), a local SSIM index, \( SSIM(i, j) \), is defined by evaluating the mean, standard deviation and covariance on a local neighborhood \( N_{ij} \), around that pixel. The overall image quality is measured by the mean SSIM (MSSIM) index given by

\[
\text{MSSIM} = \frac{1}{P} \sum_i \sum_j SSIM(i, j),
\]

(2.35)

where \( P \) is the total number of local SSIM indexes.
Figure 2.4: Original and distorted images and their IQ scores.
Wang et al. compared the MSSIM and the MOS of human assessors, using a database of JPEG and JPEG2000 compressed images at various bit rates. They found that although the MSSIM does not exhibit a linear relationship with the MOS, it is well correlated with it when the MOS is estimated from the MSSIM using nonlinear regression. Furthermore, a comparison with other IQA methods, using different metrics, showed that the MSSIM predicts the MOS better than existing IQA methods [15].

2.11 Conclusion

An introduction to image quality measures has been given in this chapter. Both objective and subjective measures are presented. It was shown that estimating image quality is a complex task. Furthermore, it is difficult to define an image quality measure that is suitable for all applications. The objective measures such as MSE, SNR, PSNR are computationally fast, and hence widely used. However, these measures are not always the best choice. This literature review revealed a lack of a comparative study of IQA methods. Therefore, the next chapter is devoted to a quantitative comparison of the various IQA measures.
Chapter 3

Analysis of Objective Image Quality Measures

3.1 Introduction

The chapter presents an analysis of objective image quality measures, which have been described in the last chapter. The goal is to find an objective measure close to the human perception of the image quality. First, we examine the objective measures of several test images to see how well these measures indicate the perceptual image quality. Second, we perform a comparison among the objective measures, using the mean opinion score (MOS) as the reference, to identify the objective measure that is the closest to a subjective measure. This is done by computing and line the IQ scores from several image quality methods, based on images with a known MOS, and then perform a comparison of the quality scores against each other.

3.2 Objective Measures versus Perceptual Quality

In this section, we analyze visually five objective image measures, namely MSSIM, UIQI, PSNR and SNR. In our experiment, an original image was used. From this image, three images were generated by adding three types of noises, namely salt & pepper, Gaussian, and speckle; one image was JPEG compressed. The original images and the derived images are shown in Figure 3.1; the computed IQ scores for these images are presented in Table 3.1.
Figure 3.1: Original and distorted images.
Table 3.1: Objective IQ scores of test images.

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>MSSIM</th>
<th>UIQI</th>
<th>PSNR</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Original</td>
<td>1</td>
<td>1</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>2</td>
<td>Salt &amp; pepper noise</td>
<td>0.7481</td>
<td>0.7451</td>
<td>20.6</td>
<td>15.3</td>
</tr>
<tr>
<td>3</td>
<td>Gaussian noise</td>
<td>0.6465</td>
<td>0.6835</td>
<td>20.7</td>
<td>15.4</td>
</tr>
<tr>
<td>4</td>
<td>Speckle noise</td>
<td>0.6972</td>
<td>0.7264</td>
<td>20.7</td>
<td>15.4</td>
</tr>
<tr>
<td>5</td>
<td>JPEG compression</td>
<td>0.6588</td>
<td>0.6128</td>
<td>20.6</td>
<td>15.3</td>
</tr>
</tbody>
</table>

The results in Table 3.1 illustrate an issue with the objective IQ scores. For example, images 2 and 5 have the same PSNR scores (20.6dB), the same SNR scores (15.3dB). However, Figure 3.1 shows clearly that the JPEG-compressed image (ID = 5) has lower visual quality due to the blocking artifacts, compared to the image with salt & pepper noise (ID = 2). Table 3.1 shows that the MSSIM and UIQI are better at quantifying the visual distortion than the PSNR and SNR.

3.3 Comparison of Objective Image Quality Measures

In this section, we compare quantitatively different objectives IQ measures, using the MOS as the reference. The same objective IQ methods as used in previous section are used in several comparisons presented in this section. The data were extracted from a database of images with known MOS scores created by Dr Zhou Wang [26]. We used a total of 168 distorted images, where only images with significant distortions are used. This is because some of the objective measures have an unstable formula when the original and the test image are almost identical. For example, the PSNR measure becomes unstable for similar images because the denominator of (2.7) goes to zero.

For each distorted image, the objective IQ measures were computed. These IQ measures were compared in three different aspects:

- linear correlation with the MOS. The normal and ranked correlation coefficients are used;
- polynomial fitting of the MOS. Third-order polynomials are used to approximate the nonlinear relationship between the objective measures and the MOS;
• consistency with the MOS. The analysis-of-variance (ANOVA) tool is used to study the objective measures for different groups of MOSs.

3.3.1 Comparison using Linear Correlation

Because the MOS is considered as one of the measures that are closest to the perceptual quality, a good objective IQ measure must be strongly correlated to the MOS. The correlation between an objective IQ measure and the MOS can be quantified by the correlation coefficient or the Spearman rank-order correlation coefficient [27, 28].

The correlation coefficient between two variables \( x \) and \( y \) is defined as:

\[
r = \frac{\sum_{i=1}^{N} (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{N}(x_i - \mu_x)^2} \sqrt{\sum_{i=1}^{N}(y_i - \mu_y)^2}},
\]

where \( x = [x_1, x_2, ..., x_N] \) and \( y = [y_1, y_2, ..., y_N] \) are two sample sets, \( \mu_x \) and \( \mu_y \) are the respective means of the two sets. A coefficient of 1 indicates a perfect correlation between \( x \) and \( y \), a coefficient of 0 means no correlation, and a coefficient of \(-1\) means perfect negative correlation. The Spearman rank-order correlation coefficient is defined similarly with an exception that the two sets are ordered before (3.1) is applied.

The different correlation coefficients between the UIQI, MSSIM, PSNR, SNR and MSE measures and the MOS are presented in Table 3.2. Among the objective measures tested, the MSSIM has the strongest correlation with the MOS (corr. coef. = 0.911). In term of correlation coefficients, the PSNR is better than the UIQI and the MSE. The MSE has negative correlation coefficients with the MOS because a higher MSE value means a lower-quality image.

<table>
<thead>
<tr>
<th>IQ Measure</th>
<th>Correlation Coefficient</th>
<th>Rank Correlation Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSIM</td>
<td>0.911</td>
<td>0.950</td>
</tr>
<tr>
<td>UIQI</td>
<td>0.834</td>
<td>0.837</td>
</tr>
<tr>
<td>PSNR</td>
<td>0.888</td>
<td>0.908</td>
</tr>
<tr>
<td>SNR</td>
<td>0.857</td>
<td>0.880</td>
</tr>
<tr>
<td>MSE</td>
<td>-0.732</td>
<td>-0.908</td>
</tr>
</tbody>
</table>
3.3.2 Comparison using Polynomial Fitting

It is unlikely that a simple objective measure, such as the UIQI or the MSSIM, would capture all the characteristics of the MOS, especially its nonlinearity. In fact, there exists a nonlinear relationship between the MOS and the MSSIM and between the MOS and the UIQI, as is evident from Figure 3.2. This nonlinear relationship can easily be modelled using a third order polynomial, or a logistic function. Therefore, for a given image quality measure $x$ (e.g., MSSIM), we can write the predicted MOS $y_x$ as

$$y_x = f(x),$$

(3.2)

where $f(x)$ is the nonlinear relationship between the objective measure and the MOS. Here, we used $f(x)$ is a third-order polynomial whose coefficients are estimated using the least-square method [27]. The residual error between the predicted and the actual MOSs can be used to compare the performances of the different objective measures. The plots of the actual MOS versus objective measures with the best-fit third-order polynomial superimposed are shown in Figure 3.2. The residual errors are presented in Table 3.3. The results show that the MSSIM had the lowest residual error (77.15), compared to the other four measures. However, there are still significant deviations between the MSSIM and the MOS for low-quality images (ie. low MOSs). The UIQI had the highest residual error; its best-fit polynomial is almost linear. Figure 3.2 shows that the relationships between the PSNR, SNR, MSE and the MOS are clearly nonlinear.

<table>
<thead>
<tr>
<th>Objective IQ Measure</th>
<th>Residual Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSIM</td>
<td>77.15</td>
</tr>
<tr>
<td>UIQI</td>
<td>139.07</td>
</tr>
<tr>
<td>PSNR</td>
<td>104.66</td>
</tr>
<tr>
<td>SNR</td>
<td>121.90</td>
</tr>
<tr>
<td>MSE</td>
<td>118.66</td>
</tr>
</tbody>
</table>

3.3.3 Comparison using Analysis of Variance

Analysis of variance (ANOVA) is a statistical tool that can be used to compare the differences among image quality measures [29] [30]. The ANOVA tool finds
Figure 3.2: Third-order polynomial fitting of MOS and objective IQ measures.
the common mean and variance of groups of data. We use the one-way analysis of variance (ANOVA) tool to assess the consistency of the objective IQ measures and the MOS.

The one-way ANOVA aims to determine if groups of data have the same mean \[ [29] [30] [28]. \] This would tell us the characteristics of groups of data, and make it possible to compare the image quality scores against one another. A one-way ANOVA can be considered as a special case of the linear model, defined by

\[
x_{ij} = a_{j} + \varepsilon_{ij}
\]

where \( x_{ij} \) is an observation in which each column represents the different groups, \( a_{j} \) is a matrix with the groups mean, where dot \( j \) notation means that all the values in row \( j \) has the same value, and \( \varepsilon_{ij} \) is a matrix of random disturbance. This model says that the observation \( x \) is a constant, plus a random disturbance.

The steps of using the ANOVA tool are as follows. The images are divided into four groups: images in group 1 have MOS values between \([0, 21.25]\), images in group 2 have MOS values between \((21.25, 45.5]\), images in group 3 have MOS values between \((45.5, 63.75]\), images in group 4 have MOS values between \((63.75, 85]\). The objective IQ measures for images in each group are collected. Clearly, if an objective measure is consistent with the MOS, the objective scores for each group must be similar and there are little overlaps in the objective scores between different groups.

Let \( x \) represent an objective IQ measure, ie. \( x \) can be MSSIM, UIQI, PSNR, SNR, or MSE. Let \( K \) be the number of groups, \( K = 4 \) in our case. Let \( N_{i} \) be the number of images in group \( i \), and \( N \) be the total number of images in all groups. Let \( x_{ij} \) be the objective IQ score for image \( j \) in group \( i \). Let \( \bar{x}_{i} \) be the average of the scores in group \( i \). An estimate of the within-group variance is given as:

\[
S_{W} = \frac{\sum_{i=1}^{K} \sum_{j=1}^{N_{i}} (x_{ij} - \bar{x}_{i})^{2}}{N - k}.
\]

An estimate of the between-group variance is defined as:

\[
S_{B} = \frac{\sum_{i=1}^{K} N_{i}(\bar{x}_{i} - \bar{x})^{2}}{K - 1},
\]

where \( \bar{x} \) is the mean across all groups. The ratio between the between-group variance and the within-group variance, \( \frac{S_{B}}{S_{W}} \), is known as the F-value. Clearly, if the samples within each group are similar and the groups are well separated, the F-value is high. The F-value is distributed within \((K - 1)\) and \((N - K)\).
The mean and standard deviation for each image quality group across different IQ measures are presented in Table 3.4. The box plots are used to show the mean and standard deviation graphically and describes the consistency of the image quality approaches. In Figure 3.3 is several box plots shown for different types of image quality assessment approaches. The figure shows significant overlaps between different quality groups for the UIQI, SNR, PSNR and MSE. There is no overlap for the reference MOS and the MSSIM. The F-values computed for the six different IQ measures are presented in Table 3.5. It is clear from the above discussion that an objective IQ measure that is more consistent to the MOS will have a higher F-value. Table 3.5 shows that the reference MOS has highest F-value, as we would expect. The MSSIM has the second highest F-value whereas the MSE has the lowest F-value. This result confirms our finding that, among the objective IQ measures tested, the MSSIM is the closest to the perceptual MOS.

3.4 Conclusion

In this chapter, we have analyzed several objective measures, namely the MSSIM, UIQI, PSNR, SNR and MSE. We have identified a problem with the PSNR, SNR and MSE measures in that they give almost the same scores for images of very
Figure 3.3: Box plots of six image quality measures: MOS, MSSIM, UIQI, PSNR, SNR, MSE.
different visual quality. We have conducted a detailed comparison of the five objective measures using three different tools: linear correlation, polynomial fitting and ANOVA. The experimental results show clearly that among the five objective measures, the MSSIM is the closest and most consistent to the mean opinion score, which is generally considered as the ground-truth of perceptual image quality. Therefore, the MSSIM will be used in later chapters to evaluate different neural network-based approaches that we propose for image quality assessment.
Chapter 4

Neural Networks: A Review and Application to IQA

4.1 Introduction

Modern computers can perform with tremendous efficiency in computation tasks that have well-defined algorithmic solutions. However, they still cannot match humans in cognitive tasks such as speech and handwriting recognition. It is clear that many applications are possible if computers can perform these tasks, which are normally attributed to the human brain. Artificial neural networks are simplified mathematical models of the human brain. They are developed to better understand how the human brain works, and more importantly how to realize some of its capabilities in computers. In this study, artificial neural networks are developed to perform automatic assessment of the image quality.

This chapter presents a brief introduction to neural networks and investigate their ability to approximate an objective image quality measure. In the previous chapter, a comparative study of several objective measures showed that the mean structural similarity index (MSSIM) is the most consistent with human assessment of image quality. Therefore, in this chapter ability of neural networks to approximate the MSSIM is investigated. The final goal, of course, is to use neural networks to predict directly the human perceptual quality; this will be dealt with in Chapter 5.

This chapter is organised as follows. Section 4.2 presents a general introduction to artificial neural networks (ANNs). Section 4.3 gives a detailed description of an ANN architecture, the multilayer feedforward neural network (MLP). Section 4.4 revises algorithms for training MLPs to perform pattern classification and function
approximation tasks. Section 4.5 focuses on the design of MLPs to estimate MSSIM index. Section 4.6 is the chapter conclusion.

4.2 An Overview of Artificial Neural Networks

An artificial neural network consists of a large number of basic processing units called neurons [31]. The ANN architecture specifies how neurons in the network are connected together. There are mainly three ANN architectures: feedforward, recurrent, and time-delayed [31, 32]. It is also possible to create a hybrid model combining the above architectures. In a feedforward architecture, information flows only in one direction, from the inputs towards the network outputs, whereas in a recurrent architecture information also flows in the opposite direction. Neurons are connected together via synaptic weights. Each neuron receives inputs from several other neurons (see Figure 4.1). The received inputs are multiplied by the synaptic weights, and the weighted sum of the inputs is passed through an activation transfer function to generate the neuron output signal, which is then propagated to other neurons. Although this is a simplified model of the biological neuron\(^1\), it has proven to be a powerful computational tool, which has been used extensively for classification, nonlinear regression, speech recognition, hand-written character recognition and many other applications.

Let \( x = [x_1, x_2, \ldots, x_p]^T \) be the input vector, \( w = [w_1, w_2, \ldots, w_p]^T \) be the synaptic weight vector, \( \theta \) be a scalar known as the bias term, and \( f \) be the activation transfer function. The neuron output \( y \) is given by

\[
y = f \left( w^T x + \theta \right). \tag{4.1}
\]

Note that, the bias term \( \theta \) may be viewed as the weight of a fixed input equal to 1. In this case, both the input and weight vectors can be augmented as follows: \( x = [x_1, x_2, \ldots, x_p, 1]^T \) and \( w = [w_1, w_2, \ldots, w_p, \theta]^T \). In this case the neuron output can be expressed as

\[
y = f \left( w^T x \right). \tag{4.2}
\]

There exist many different activation functions, which can be used in an ANN; the most common activations functions are defined below (see also Figure 4.2):

---

\(^1\)The human brain has over ten billion neurons, whereas an artificial neural network consists of orders-of-magnitudes fewer neurons.
• The hyperbolic tangent sigmoid (tansig)

\[
f(x) = \tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}}.
\]  (4.3)

• The logistic sigmoid (logsig)

\[
f(x) = \frac{1}{1 + e^{-x}}.
\]  (4.4)

• The linear function (lin)

\[f(x) = x.
\]  (4.5)

• The saturating linear function (satlins)

\[
f(x) = \begin{cases} 
  x, & -1 \leq x \leq 1 \\
  1, & x > 1 \\
  -1, & x < -1 
\end{cases}
\]  (4.6)

### 4.3 Multilayer Feedforward Neural Networks

The multilayer feedforward neural network, also called the multilayer perception (MLP), is one of the most common neural network architectures [31, 32]. It has been used extensively in applications that require function approximation and pattern classification. A MLP consists of many layers, each having a number of neurons; Figure 4.3 illustrates the architecture of a feedforward ANN. There are two fixed-size layers called the input layer and output layer. Between these two layers, there is one or more hidden layer(s). The input layer simply receives data from the environment.
and distributes them to the succeeding hidden layer. The output layer generates the network outputs and passes them to the environment. Hidden layers are so called because they have no direct connection with the outside environment. All processing in the MLP is done in the hidden layers and the output layer. Neurons in each layer receive inputs from the preceding layer, calculate their outputs and transmit them to the following layer. Connections between neurons are weighted, and it is the connection weights that determine the operation of a network. The process of adjusting the network parameters (weights and biases) so that it produces the desired outputs is called training; some training algorithms for the MLP are discussed next. The performance of the network is usually assessed in terms of its ability to generalize to previously unseen patterns. There is usually a danger of overtraining a neural network, which results in overfitting; a network is said to overfit the training data when its performance is very accurate on the training patterns, but very inaccurate on data which it hasn’t seen during training.
4.4 Training the MLP

Multilayer perceptrons are trained using *supervised learning* [31, 32]. In a supervised learning scheme, the network is given a set of training exemplars. Each training sample consists of an input-output pair. The adjustable parameters of the network (i.e., the weights and biases) are first initialised (e.g., with small random values). Training is often an iterative process. At each iteration, the network output is compared with the expected output, and the difference (i.e., the error function) is computed. This error function is used to modify the trainable parameters so that the error is reduced in the next training iteration. Training ends when one of several stopping criteria is met.

The following tasks are important in designing an MLP:

- finding a representative training set for the given application;
- selecting a suitable network structure: number of layers, number of neurons in each layer, and activation function for each layer;
- finding an efficient training algorithm with respect to initialization, error function, weight update rule, and stopping criteria.

One common error function that is often used is the *mean squared error*, which is defined as

$$E = \frac{1}{2N} \sum_{n=1}^{N} ||t_n - y_n||^2,$$

(4.7)
where \( N \) is the number of training exemplars, \( y_n \) is the vector of network outputs due to the \( n \)th input pattern, and \( t_n \) is the corresponding desired output. A main objective of training, besides improving generalization capability, is to minimize this error function. Clearly, the network has learnt the training data when \( E \) becomes small. Common stopping criteria for training are:

- the error function is below a threshold;
- the number of iterations exceeds a given limit;
- the network performance on a validation set starts to degrade.

Training usually involves presenting the input patterns, one by one, and computing the prediction error according to (4.7). Then the free parameters are adjusted according to a learning rule of the form

\[
w(k + 1) = w(k) + \alpha d(k),
\]

where \( w(k + 1) \) is the new weight vector, \( w(k) \) is the previous weight vector, \( \alpha \) is the learning rate, and \( d(k) \) is the search direction. The different training algorithms differ in the way the search direction is computed. In the next three subsection, three common training algorithms are discussed in more details.

### 4.4.1 Gradient-Descent and Error-Backpropagation Algorithm

The error function \( E \) is expressed as a differentiable multivariate function of the network free parameters, or weight vector \( w \). It is a fact that at any point on the error surface, the negative gradient of the error function points towards the direction of the steepest descent \([32, 33]\). Therefore, in order to reduce the error, it is sufficient to choose the search direction along the negative gradient direction. In this case a single free parameter \( w \) can be updated as follows

\[
w(k + 1) = w(k) - \alpha \frac{\partial E}{\partial w},
\]

where \( k \) represents the iteration number and \( \alpha \) is the learning rate. This update equation is known as the gradient-descent weight update rule.

An important discovery in the neural network field is the error-backpropagation algorithm for computing the error gradient \( \nabla E \) with respect to the free parameters. Let \( y_{l,i} \) denote the output of neuron \( i \) in layer \( l \), \( c_{l+1,j} \) denote the weighted input...
sum to neuron \( j \) in layer \( l + 1 \), and \( w_{l,i,j} \) denote the connection weight from the \( i \)th neuron in layer \( l \) to \( j \)th neuron in layer \( l + 1 \).

\[
 c_{l+1,j} = \sum_i w_{l,i,j} y_{l,i} \tag{4.10}
\]

The output of the \( j \)th neuron in layer \( l + 1 \) is

\[
 y_{l+1,j} = f(c_{l+1,j}) \tag{4.11}
\]

We define the error sensitivity \( s_{l,i} \) of neuron \( i \) in layer \( l \) as follows:

\[
 s_{l,i} = \frac{\partial E}{\partial c_{l,i}} \tag{4.12}
\]

The error sensitivity is the rate of change of the error with respect to the weighted input sum to a neuron. If all error sensitivities are known, the local gradient \( \frac{\partial E}{\partial w_{l,i,j}} \) can be computed by applying the chain rule of differentiation:

\[
 \frac{\partial E}{\partial w_{l,i,j}} = \frac{\partial E}{\partial c_{l+1,j}} \times \frac{\partial c_{l+1,j}}{\partial w_{l,i,j}} = y_{l,i} s_{l+1,j} \tag{4.13}
\]

It turns out that the error sensitivities can be computed recursively, starting from the output layer:

\[
 s_{l,i} = \frac{\partial E}{\partial c_{l,i}} = \sum_j \frac{\partial E}{\partial c_{l+1,j}} \frac{\partial c_{l+1,j}}{\partial c_{l,i}} = \sum_j s_{l+1,j} \times w_{l,i,j} f'(y_{l,i}) \tag{4.14}
\]

Taking out the common term \( f'(y_{l,i}) \), the above equation can be written as

\[
 s_{l,i} = f'(y_{l,i}) \sum_j w_{l,i,j} s_{l+1,j} \tag{4.15}
\]

Equation (4.15) shows that error sensitivity for a neuron in layer \( l \) can be computed by propagating backwards error sensitivities of layer \( l + 1 \) through the synaptic weights \( w_{l,i,j} \). Thus, using the backpropagation procedure in (4.15) we can compute all error sensitivities, and thereby obtain the gradient vector from (4.13).

### 4.4.2 Conjugate Gradient Algorithms

The gradient-descent algorithm updates weights along only one direction, which is the negative gradient of the error function [29, 34, 35, 36]. In comparison, conjugate gradient (CG) algorithms update the weights along different directions which are expected to yield faster convergence. The weight update \( \Delta w(k) \) at iteration \( k \) can be expressed as

\[
 \Delta w(k) = \alpha \times d(k), \tag{4.16}
\]
where $\alpha$ is the learning rate and $d(k)$ is the search direction. The initial search direction is usually the negative gradient. Subsequently, the new search direction is a combination of the current search direction and the negative gradient:

$$d(k + 1) = -\nabla_w(k) + \beta(k) \times d(k)$$  \hspace{1cm} (4.17)

where $\beta(k)$ is a scalar and $\nabla_w(k)$ is the error gradient at the $k$th iteration. Conjugate gradient algorithms differ in the choice of $\beta(k)$. Some common CG algorithms are listed below:

- **Fletcher-Reeves algorithm:**

  $$\beta(k) = \frac{\|\nabla_w(k)\|^2}{\|\nabla_w(k - 1)\|^2}$$  \hspace{1cm} (4.18)

- **Polak-Ribière algorithm:**

  $$\beta(k) = \frac{(\nabla_w(k) - \nabla_w(k - 1))^T \nabla_w(k)}{\|\nabla_w(k - 1)\|^2}$$  \hspace{1cm} (4.19)

- **Powell-Beale algorithm:** The search direction is reset to the negative gradient when there is little orthogonality between the current and the previous gradient:

  $$|\nabla_w^T(k - 1) \nabla_w(k)| \geq 0.2 \|\nabla_w(k)\|^2$$  \hspace{1cm} (4.20)

  Between resets, the search direction is computed as follows:

  $$d(k + 1) = -\nabla_w(k) + \beta(k)d(k) + \gamma(k)d(r),$$  \hspace{1cm} (4.21)

  where $r$ is the iteration number of the previous reset, $\beta(k)$ and $\gamma(k)$ are scalar learning rates. The learning rates are computed as

  $$\beta(k) = \frac{\nabla_w^T(k)\{\nabla_w(k) - \nabla_w(k - 1)\}}{d^T(k)\{\nabla_w(k) - \nabla_w(k - 1)\}},$$  \hspace{1cm} (4.22)

  and

  $$\gamma(k) = \frac{\nabla_w^T(r)\{\nabla_w(r) - \nabla_w(r - 1)\}}{d^T(r)\{\nabla_w(r) - \nabla_w(r - 1)\}}.$$  \hspace{1cm} (4.23)

In this project, we use the Powell-Beale algorithm to train networks with a large number of parameters (e.g, networks in Section 5.4.1)
4.4.3 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt (LM) algorithm is a quasi-Newton algorithm [29, 34]. It involves the computation of Jacobian matrix $J$, which is the first derivative of the error function $E$ with respect to weights $w$. The Jacobian matrix can be computed through the standard error-backpropagation algorithm, and it is used to approximate the Hessian matrix. The Levenberg-Marquardt weight update rule can be summarized as follows:

$$w(k + 1) = w(k) - (J^T J + \mu I)^{-1} J^T E$$  \hspace{1cm} (4.24)

where $\mu$ is an adaptive parameter. When $\mu$ is small, the LM update is similar to a second-order optimization method (i.e., Newton method). When $\mu$ is large, the LM update is similar to the gradient descent method. The LM algorithm can approach the convergence speed of Newton method, without computing the Hessian matrix. However, it still requires large memory and hence is suitable only for small or medium-sized networks.

4.5 Estimating the MSSIM Using Neural Networks

In this section, we develop MLP neural networks to estimate the MSSIM score of image quality. We will examine two different approaches. The first approach uses statistical features as inputs to the network. The features are the mean, standard deviation and covariance of the original and the distorted image. The second approach uses direct pixel values as network inputs. This approach involves more data, but it does not require any preprocessing. Our objective is to evaluate the suitability of these ANN-based approaches to predict the MSSIM image quality score. Evaluation is based on comparison between the predicted and the actual MSSIM scores.

4.5.1 Data Preparation

Two training sets were prepared for the two neural network approaches: feature-based and pixel-based. For training Set 1, we used 7 original images. Different noises were added to these images to create distorted images. The images were divided into non-overlapping $8 \times 8$ blocks. For each pair of blocks (one from the original image and the other from the distorted image), the following five features are computed:
• mean of original block,
• mean of distorted block,
• variance of original block,
• variance of distorted block,
• covariance of original and distorted blocks.

The reader is referred to Section 2.10 for the computation of these features. The five features form a feature vector which is used as input to the network. The actual SSIM score for the pair is computed as described in Section 2.10 (parameters $K_1 = 0.01$ and $K_2 = 0.03$), and is used as the network target. Overall, training Set 1 has 3,704 training samples.

Training Set 2 was prepared in a similar fashion. We used 18 original images and 32 distorted images. However, now each feature vector has 128 elements: 64 pixel values from the original block and 64 pixel values from the distorted block. Overall, training Set 2 contains 15,200 samples.

Two test sets were prepared to evaluate the trained networks. Test Set 1 consists of images distorted by Gaussian and salt-pepper noise. Test Set 2 has 190 images distorted with JPEG compression. None of the test images has been used in the training.

4.5.2 Statistical Features-based Neural Network

The networks were trained with the Levenberg-Marquardt algorithm. This algorithm was selected because it performs very well for small-sized networks with small training sets. During training, the network was evaluated on the training set and "hard" training samples were identified. These training samples were presented with higher frequency in subsequent training iterations.

Different network sizes were experimented with. We began with small network sizes, and increased the number of neurons step-by-step. Each trained network was evaluated on the two test sets.

The overall prediction error for test Set 1 is shown in Figure 4.4 as a function of network sizes. The figure shows that the prediction error decreases as the number of neurons is increased. However, there was little improvement in the prediction error
for network sizes above 5-8-8-1 (8 neurons in hidden layer 1, 8 neurons in hidden layer 2). Hence, we selected the network of size 5-8-8-1 to be tested further.

![Graph showing test error versus the number of neurons.](image)

**Figure 4.4:** Test error versus the number of neurons.

The results for test Set 1 are shown in Table 4.1 and Figures 4.5(a) and 4.5(b). Table 4.1 shows the network prediction and the MSSIM score, where the PSNR values indicate the amount of noise. The results indicate that the neural network approximates the MSSIM index with very small error. The absolute error between the predicted and the actual MSSIM scores has a maximum value of 0.022 and a mean value of 0.0041. We should note that the network prediction performance is better for images with salt & pepper noise compared to images with Gaussian noise. One explanation is that Gaussian noise lead to large shifts in the local SSIM scores making it harder for the network to assess this type of distortion. Nevertheless, our results show that NNs have the ability to approximate the MSSIM index, and hence further experimentation with the neural network approach is justified. The neural network was also evaluated on Test Set 2. The network-predicted MSSIM scores are plotted against the actual MSSIM scores in Figure 4.6. This figure shows that the network performed well for images with low distortion (high MSSIM scores); it did not perform so well for images with high distortion. Nevertheless, there is a strong correlation between the predicted and the actual MSSIM scores. Table 4.2 shows performance indicators of the feature-based network. It shows clearly that the network provided a very good approximation of the MSSIM score. There was a strong linear correlation between the network output and the actual MSSIM scores.
Table 4.1: MSSIM prediction of features-based neural network for Test Set 1 (Gaussian, salt & pepper noise).

<table>
<thead>
<tr>
<th>Type of noise</th>
<th>PSNR (db)</th>
<th>MSSIM value</th>
<th>Network output</th>
<th>Absolute error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>51.44</td>
<td>0.999</td>
<td>1.000</td>
<td>0.00101</td>
</tr>
<tr>
<td>Gaussian</td>
<td>41.63</td>
<td>0.997</td>
<td>0.997</td>
<td>0.000022</td>
</tr>
<tr>
<td>Gaussian</td>
<td>29.69</td>
<td>0.948</td>
<td>0.946</td>
<td>0.00227</td>
</tr>
<tr>
<td>Gaussian</td>
<td>27.64</td>
<td>0.913</td>
<td>0.911</td>
<td>0.00197</td>
</tr>
<tr>
<td>Gaussian</td>
<td>24.62</td>
<td>0.811</td>
<td>0.813</td>
<td>0.00244</td>
</tr>
<tr>
<td>Gaussian</td>
<td>23.58</td>
<td>0.753</td>
<td>0.759</td>
<td>0.00579</td>
</tr>
<tr>
<td>Gaussian</td>
<td>22.37</td>
<td>0.665</td>
<td>0.676</td>
<td>0.01068</td>
</tr>
<tr>
<td>Gaussian</td>
<td>21.41</td>
<td>0.581</td>
<td>0.596</td>
<td>0.01514</td>
</tr>
<tr>
<td>Gaussian</td>
<td>20.84</td>
<td>0.531</td>
<td>0.548</td>
<td>0.01791</td>
</tr>
<tr>
<td>Gaussian</td>
<td>20.32</td>
<td>0.489</td>
<td>0.511</td>
<td>0.02203</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>42.97</td>
<td>0.997</td>
<td>0.997</td>
<td>0</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>35.85</td>
<td>0.986</td>
<td>0.986</td>
<td>0.00020</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>35.15</td>
<td>0.985</td>
<td>0.985</td>
<td>0.00017</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>31.24</td>
<td>0.962</td>
<td>0.961</td>
<td>0.00037</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>26.75</td>
<td>0.903</td>
<td>0.902</td>
<td>0.00073</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>24.98</td>
<td>0.853</td>
<td>0.853</td>
<td>0.00092</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>19.79</td>
<td>0.639</td>
<td>0.639</td>
<td>0.00031</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>18.50</td>
<td>0.559</td>
<td>0.558</td>
<td>0.00011</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>17.44</td>
<td>0.493</td>
<td>0.493</td>
<td>0.00023</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>16.66</td>
<td>0.438</td>
<td>0.439</td>
<td>0.00080</td>
</tr>
</tbody>
</table>

(a) salt & pepper noise  
(b) Gaussian noise

Figure 4.5: Actual MSSIM versus the network-prediction for images added with noise.
Table 4.2: MSSIM prediction performance of feature-based neural network.

<table>
<thead>
<tr>
<th>Performance Indicator</th>
<th>Test set 1</th>
<th>Test set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean error</td>
<td>0.0041</td>
<td>0.0109</td>
</tr>
<tr>
<td>Maximum error</td>
<td>0.0220</td>
<td>0.0890</td>
</tr>
<tr>
<td>Correlation of actual and predicted values</td>
<td>0.9996</td>
<td>0.9973</td>
</tr>
<tr>
<td>Ranked correlation</td>
<td>0.9985</td>
<td>0.9908</td>
</tr>
</tbody>
</table>

with a correlation coefficient of 0.9973.

4.5.3 Pixel-based Neural Network

In this approach, the feature vector has 128 elements; this requires a large training set to learn the network parameters. Hence, we decided to use the Conjugate Gradient algorithm for training. Training Set 2 was divided into several groups, which were then presented to the network sequentially. This technique improves the training speed and memory utilization. Again, we experimented with different network sizes.

The trained networks were evaluated on the two test sets describe above. The errors on test Set 1 for different network sizes are shown in Figure 4.7. The figure
shows that the mean error fluctuated slightly for different network sizes. We selected the network of size 128-25-15-1 for final testing because it had a reasonable performance in terms of the maximum error and the mean error.

![Graph](image)

**Figure 4.7:** Training error versus the number of neurons.

The results for Test Set 1 are shown in Table 4.3 and Figures 4.8(a) and 4.8(b). The results indicate that this neural network approximates the MSSIM index with slightly higher errors than the feature-based network presented in Section 4.5.2. The absolute error between the predicted and the actual MSSIM scores has a maximum value of 0.0854 and a mean value of 0.0246.

The pixel-based neural network was also evaluated on Test Set 2. The predicted and the actual MSSIM scores are plotted in Figure 4.9. The figure shows a strong linear correlation between the network and the MSSIM. This is also confirmed by the results in Table 4.4. In addition to strong correlation, Table 4.4 shows that the network has a small error between the predicted MSSIM and the actual MSSIM for both test sets. The maximum and mean error for test set 2 has improved from 0.089 and 0.0109 to 0.0617 and 0.0095 respectfully. The improvement can be explained by the difference in the input-coding scheme and the larger training set. Clearly, some information about the image quality is lost when a pair of 8 × 8 blocks are represented by only five features.
Table 4.3: MSSIM prediction of pixel-based network for Test Set 1 (Gaussian and salt & pepper noise).

<table>
<thead>
<tr>
<th>Type of noise</th>
<th>PSNR (dB)</th>
<th>MSSIM value</th>
<th>Network output</th>
<th>Absolute error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>51.44</td>
<td>1.000</td>
<td>0.992</td>
<td>0.00811</td>
</tr>
<tr>
<td>Gaussian</td>
<td>41.63</td>
<td>0.997</td>
<td>0.988</td>
<td>0.00898</td>
</tr>
<tr>
<td>Gaussian</td>
<td>29.69</td>
<td>0.948</td>
<td>0.938</td>
<td>0.00985</td>
</tr>
<tr>
<td>Gaussian</td>
<td>27.64</td>
<td>0.913</td>
<td>0.908</td>
<td>0.00486</td>
</tr>
<tr>
<td>Gaussian</td>
<td>24.62</td>
<td>0.811</td>
<td>0.825</td>
<td>0.01433</td>
</tr>
<tr>
<td>Gaussian</td>
<td>23.58</td>
<td>0.753</td>
<td>0.779</td>
<td>0.02590</td>
</tr>
<tr>
<td>Gaussian</td>
<td>22.37</td>
<td>0.665</td>
<td>0.710</td>
<td>0.04484</td>
</tr>
<tr>
<td>Gaussian</td>
<td>21.41</td>
<td>0.581</td>
<td>0.641</td>
<td>0.06020</td>
</tr>
<tr>
<td>Gaussian</td>
<td>20.84</td>
<td>0.531</td>
<td>0.602</td>
<td>0.07181</td>
</tr>
<tr>
<td>Gaussian</td>
<td>20.32</td>
<td>0.489</td>
<td>0.574</td>
<td>0.08537</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>46.57</td>
<td>0.999</td>
<td>0.991</td>
<td>0.00842</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>36.67</td>
<td>0.986</td>
<td>0.978</td>
<td>0.00835</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>35.16</td>
<td>0.984</td>
<td>0.975</td>
<td>0.00913</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>31.59</td>
<td>0.965</td>
<td>0.955</td>
<td>0.00925</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>27.15</td>
<td>0.908</td>
<td>0.894</td>
<td>0.01367</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>25.28</td>
<td>0.864</td>
<td>0.846</td>
<td>0.01800</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>20.01</td>
<td>0.633</td>
<td>0.613</td>
<td>0.02008</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>18.69</td>
<td>0.567</td>
<td>0.541</td>
<td>0.02603</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>17.50</td>
<td>0.486</td>
<td>0.465</td>
<td>0.02043</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>16.82</td>
<td>0.456</td>
<td>0.432</td>
<td>0.02356</td>
</tr>
</tbody>
</table>

![Figure 4.8](image1.png)  
(a) salt & pepper noise  
(b) Gaussian noise

Figure 4.8: Actual MSSIM versus the network-prediction for images added with noise.
Figure 4.9: Actual versus predicted MSSIM scores (pixel-based network).

Table 4.4: MSSIM prediction performance of pixel-based neural network.

<table>
<thead>
<tr>
<th>Performance Indicator</th>
<th>Test set 1</th>
<th>Test set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean error</td>
<td>0.0258</td>
<td>0.0095</td>
</tr>
<tr>
<td>Maximum error</td>
<td>0.0854</td>
<td>0.0617</td>
</tr>
<tr>
<td>Correlation coefficient</td>
<td>0.9860</td>
<td>0.9954</td>
</tr>
<tr>
<td>Ranked correlation coefficient</td>
<td>0.9895</td>
<td>0.9884</td>
</tr>
</tbody>
</table>
4.6 Conclusion

An introduction of artificial neural networks has been given in this chapter. We described the artificial neuron and the multilayer feedforward neural network. We then present in details several training algorithms including the gradient-descent with error-back propagation, the conjugate-gradient, and the Levenberg-Marquardt. In this chapter, we also presented two neural network approaches for approximating an objective image quality index, called the mean structural similarity index. The first approach uses statistical image features as network inputs, and the second approach uses pixel values directly as network inputs. For both approaches, the predicted MSSIM scores exhibit strong correlation with the actual MSSIM scores although the first approach is found to have better approximation. We also found that the errors are higher for images corrupted with white Gaussian noise. Experiments in this chapter show that artificial neural networks can be trained to approximate an objective image quality measure. The question now is: can neural networks predict the perceptual quality? This question will be answered in the next chapter.
Chapter 5

Predicting the MOS Using Neural Networks

5.1 Introduction

In Chapter 3, we presented a comparison between objective and subjective image quality assessment methods. The comparison results showed that no objective measure can match with high fidelity the subjective measure. In Chapter 4, we trained neural networks to approximate an objective IQA measure, called the mean structural similarity index; not surprisingly, the neural network was able to approximate the MSSIM with very high accuracy. In this chapter, neural networks are employed to predict the subjective image quality, called the mean opinion score (MOS). Two neural network approaches are developed: the first uses statistical features as inputs to the network, whereas the second approach uses direct pixel values as inputs to the network. Furthermore, a comparison between the neural network approaches and the MSSIM index is conducted. The chapter is organised as follows. Next section describes the process of preparing the data for training and testing the neural networks. In Section 5.3, the feature-based neural network approach is described and the developed neural networks are analysed in terms of their ability to predict the MOS. Section 5.4 presents the pixel-based approach. A comparison between the neural network approaches and the MSSIM is conducted in Section 5.5, followed by some concluding remarks in Section 5.6.
5.2 Data Preparation

The data used in our work are taken from an online image database created by Zhou Wang for research in image quality assessment [26]. The database consists of 460 images (233 JPEG-compressed images and 227 JPEG2000-compressed images). Each image in this database has a MOS value associated with it; the MOS values were obtained from a group of human observers. We used only distorted images with noticeable degradation (i.e., images with MSSIM values less than 1) in Wang's database. However, the images we used still have MOS values in the range [15, 95]. Our dataset consists of 343 distorted images with known MOS values with respect to the corresponding original images, and nine images with no distortion. The nine images have their MOS values set to 100; they were used to test IQA algorithms on images with perfect quality.

5.2.1 Data for Feature-based Approach

Only non-overlapping features are used to save computing time for the networks. A changing from a overlapping to a non-overlapping has only a small impact on the MSSIM. Within the 341 test images, was the maximum difference only 0.008 on the MSSIM index. Each distorted image was partitioned into non-overlapping 8 × 8 blocks. For each block, eight statistical measures were computed:

- mean of the original block $\mu_x$, Eq. (2.25),
- mean of the distorted block $\mu_y$, Eq. (2.26),
- standard deviation of the original block $\sigma_x$, Eq. (2.28),
- standard deviation of the distorted block $\sigma_y$, Eq. (2.29),
- covariance of the original and distorted blocks $\sigma_{xy}$, Eq. (2.30),
- mean square error of the original and distorted block MSE, Eq. (2.6),
- contrast of original block $c_x$, Eq. (5.1), and
- contrast of distorted block $c_y$.

The following Michelson contrast formula is used to compute $c_x$ and $c_y$:

$$c_x = \frac{\max\{f(m,n)\} - \min\{f(m,n)\} + c}{\max\{f(m,n)\} + \min\{f(m,n)\} + c'}$$  \hspace{1cm} (5.1)
where $f(m, n)$ is an image block and $c$ is a constant.

The training set was prepared as follows. For each distorted image, an overall MSSIM score was computed. A distorted image block was selected only if its SSIM (structural similarity) score is within ±0.01 of the overall MSSIM score. This filtering technique ensured that only blocks that were consistent with the overall image quality assessment were used for training. The SSIM score was used in filtering because it was found to be the closest to the mean opinion score, among the objective measures tested in Chapter 3. To further reduce the amount of training data, a limit of 80 training samples per image was imposed. The target value for each training sample was the MOS of the corresponding image.

The test set was generated in a similar way using images which had not been used for training. However, only blocks whose SSIM scores are within ±0.04 of the corresponding MSSIM score were selected, and a limit of 400 test samples per image was imposed.

### 5.2.2 Data for Pixel-based Approach

The training and test sets for the pixel-based approach were extracted from the same data used for the feature-based approach. Each distorted image and the corresponding original image were partitioned into non-overlapping 8 × 8 blocks. For each pair of original and distorted blocks, a feature vector consisting of 128 pixel values was formed. The network target for the feature vector was the MOS of the corresponding distorted image.

The SSIM-based filtering technique described above was used to select representative training samples. The same limits on the number of samples per distorted image were also imposed.

### 5.3 Feature-based Neural Network Approach

This section focuses on the ANN approach of predicting the MOS using statistical features as inputs to the network. The section is organised as follows. First, networks using different statistical features are developed and compared to find the most suitable feature vector for estimating the MOS. Next, the network with the best feature vector is analysed in terms of correlation and error estimates. Finally, the network is analysed using the ANOVA tool in order to evaluate the consistency of
the network-predicted MOS with the actual MOS.

5.3.1 Selection of Feature Vector

We trained several neural networks, using different feature vectors (see Table 5.1). These feature vectors are formed from the eight statistical measures described in Section 5.2.1. For each feature vector, a group of ten networks were trained using the Levenberg-Marquardt algorithm. Each network had two hidden layers with 6 neurons in each layer. For each image, the average of the network outputs, for all non-overlapping $8 \times 8$ blocks, was taken as an estimate of the MOS score. This estimated MOS was compared with the actual MOS to evaluate the network performance.

Table 5.1: Feature vectors for the feature-based ANN approach.

<table>
<thead>
<tr>
<th>Class</th>
<th>Feature Vector</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$[\sigma_{xy}, \sigma_x, \sigma_y]$</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>$[\sigma_{xy}, \sigma_x, \sigma_y, \text{MSE}]$</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>$[\sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y]$</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>$[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y]$</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>$[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, \text{MSE}]$</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>$[\sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y, \text{MSE}]$</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>$[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y]$</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>$[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y, \text{MSE}]$</td>
<td>8</td>
</tr>
</tbody>
</table>

The performances of some feature vectors are shown in Tables 5.2 to 5.5. Results for other feature vectors are shown in Tables A.1 to A.4 in the Appendix. The performances were measured in terms of the mean error, maximum error, linear correlation coefficient and rank-order correlation coefficient between the actual and the predicted MOS. The comparative performance indicators for the eight feature vectors are shown in Figures 5.1 and 5.2.

Results in Figures 5.1 and 5.2 show that feature vectors 5, 6 and 8 have the best performances in estimating the MOS. Among the eight feature vectors, feature vector 8 has the lowest error, and slightly better correlation coefficient. However, networks using feature vector 8 require more weights compared to networks using other feature vectors. Feature vector 5 has a low error rate and a high correlation with the actual MOS; it also requires fewer network weights. Therefore, feature vector 5 will be used in our further experiments.
Table 5.2: Performance of feature vector 5: $[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, \text{MSE}]$.

<table>
<thead>
<tr>
<th>Network</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.865</td>
<td>16.795</td>
<td>0.979</td>
<td>0.954</td>
</tr>
<tr>
<td>2</td>
<td>3.347</td>
<td>14.490</td>
<td>0.984</td>
<td>0.956</td>
</tr>
<tr>
<td>3</td>
<td>3.919</td>
<td>18.407</td>
<td>0.977</td>
<td>0.948</td>
</tr>
<tr>
<td>4</td>
<td>4.184</td>
<td>19.599</td>
<td>0.975</td>
<td>0.949</td>
</tr>
<tr>
<td>5</td>
<td>3.846</td>
<td>16.982</td>
<td>0.979</td>
<td>0.952</td>
</tr>
<tr>
<td>6</td>
<td>3.995</td>
<td>17.330</td>
<td>0.978</td>
<td>0.952</td>
</tr>
<tr>
<td>7</td>
<td>4.082</td>
<td>18.220</td>
<td>0.977</td>
<td>0.953</td>
</tr>
<tr>
<td>8</td>
<td>3.906</td>
<td>17.865</td>
<td>0.978</td>
<td>0.951</td>
</tr>
<tr>
<td>9</td>
<td>3.750</td>
<td>16.465</td>
<td>0.980</td>
<td>0.955</td>
</tr>
<tr>
<td>10</td>
<td>3.928</td>
<td>17.869</td>
<td>0.978</td>
<td>0.953</td>
</tr>
<tr>
<td>Mean value</td>
<td>3.882</td>
<td>17.402</td>
<td>0.978</td>
<td>0.952</td>
</tr>
</tbody>
</table>

Table 5.3: Performance of feature vector 6: $[\sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y, \text{MSE}]$.

<table>
<thead>
<tr>
<th>Network</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.043</td>
<td>17.463</td>
<td>0.977</td>
<td>0.956</td>
</tr>
<tr>
<td>2</td>
<td>4.023</td>
<td>20.122</td>
<td>0.976</td>
<td>0.949</td>
</tr>
<tr>
<td>3</td>
<td>4.064</td>
<td>20.582</td>
<td>0.976</td>
<td>0.950</td>
</tr>
<tr>
<td>4</td>
<td>3.930</td>
<td>18.805</td>
<td>0.977</td>
<td>0.951</td>
</tr>
<tr>
<td>5</td>
<td>4.045</td>
<td>18.328</td>
<td>0.977</td>
<td>0.947</td>
</tr>
<tr>
<td>6</td>
<td>4.184</td>
<td>18.149</td>
<td>0.976</td>
<td>0.950</td>
</tr>
<tr>
<td>7</td>
<td>3.747</td>
<td>17.661</td>
<td>0.980</td>
<td>0.956</td>
</tr>
<tr>
<td>8</td>
<td>3.755</td>
<td>17.665</td>
<td>0.980</td>
<td>0.953</td>
</tr>
<tr>
<td>9</td>
<td>3.874</td>
<td>19.851</td>
<td>0.978</td>
<td>0.955</td>
</tr>
<tr>
<td>10</td>
<td>4.054</td>
<td>18.820</td>
<td>0.976</td>
<td>0.951</td>
</tr>
<tr>
<td>Mean value</td>
<td>3.972</td>
<td>18.745</td>
<td>0.977</td>
<td>0.952</td>
</tr>
</tbody>
</table>
Table 5.4: Performance of feature vector 7: $[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y]$.

<table>
<thead>
<tr>
<th>Network</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.509</td>
<td>26.014</td>
<td>0.960</td>
<td>0.935</td>
</tr>
<tr>
<td>2</td>
<td>5.762</td>
<td>26.397</td>
<td>0.956</td>
<td>0.924</td>
</tr>
<tr>
<td>3</td>
<td>5.642</td>
<td>20.274</td>
<td>0.960</td>
<td>0.931</td>
</tr>
<tr>
<td>4</td>
<td>5.558</td>
<td>22.233</td>
<td>0.960</td>
<td>0.934</td>
</tr>
<tr>
<td>5</td>
<td>6.070</td>
<td>25.066</td>
<td>0.953</td>
<td>0.917</td>
</tr>
<tr>
<td>6</td>
<td>5.726</td>
<td>25.631</td>
<td>0.956</td>
<td>0.919</td>
</tr>
<tr>
<td>7</td>
<td>5.901</td>
<td>24.763</td>
<td>0.954</td>
<td>0.928</td>
</tr>
<tr>
<td>8</td>
<td>5.439</td>
<td>21.829</td>
<td>0.960</td>
<td>0.943</td>
</tr>
<tr>
<td>9</td>
<td>5.994</td>
<td>21.992</td>
<td>0.954</td>
<td>0.925</td>
</tr>
<tr>
<td>10</td>
<td>5.690</td>
<td>24.737</td>
<td>0.958</td>
<td>0.937</td>
</tr>
<tr>
<td>Mean value</td>
<td>5.729</td>
<td>23.894</td>
<td>0.957</td>
<td>0.929</td>
</tr>
</tbody>
</table>

Table 5.5: Performance of feature vector 8: $[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y, \text{MSE}]$.

<table>
<thead>
<tr>
<th>Network</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.675</td>
<td>15.666</td>
<td>0.981</td>
<td>0.957</td>
</tr>
<tr>
<td>2</td>
<td>3.811</td>
<td>16.478</td>
<td>0.980</td>
<td>0.956</td>
</tr>
<tr>
<td>3</td>
<td>4.278</td>
<td>16.268</td>
<td>0.975</td>
<td>0.953</td>
</tr>
<tr>
<td>4</td>
<td>3.958</td>
<td>16.301</td>
<td>0.979</td>
<td>0.953</td>
</tr>
<tr>
<td>5</td>
<td>3.783</td>
<td>15.834</td>
<td>0.980</td>
<td>0.956</td>
</tr>
<tr>
<td>6</td>
<td>4.020</td>
<td>17.615</td>
<td>0.977</td>
<td>0.951</td>
</tr>
<tr>
<td>7</td>
<td>3.905</td>
<td>18.006</td>
<td>0.978</td>
<td>0.950</td>
</tr>
<tr>
<td>8</td>
<td>3.832</td>
<td>17.515</td>
<td>0.979</td>
<td>0.955</td>
</tr>
<tr>
<td>9</td>
<td>3.890</td>
<td>16.153</td>
<td>0.979</td>
<td>0.953</td>
</tr>
<tr>
<td>10</td>
<td>3.916</td>
<td>14.959</td>
<td>0.979</td>
<td>0.958</td>
</tr>
<tr>
<td>Mean value</td>
<td>3.907</td>
<td>16.479</td>
<td>0.979</td>
<td>0.954</td>
</tr>
</tbody>
</table>
Figure 5.1: Mean and maximum error based on the test set for different feature vectors.

Figure 5.2: Correlation coefficients based on the test set for different feature vectors.
These results indicate that feature vectors based on the standard deviations, the covariance and the MSE lead to better performance. The mean and contrast measures have less effects on the prediction results, which is surprising.

5.3.2 Selection of Network Size

In the previous subsection, we found that the feature vector $[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, MSE]$ has a small size and gives good performance. In this section, we aim to find a suitable network size for this feature vector. In our experiment, the Levenberg-Marquardt training algorithm was again used. The original training set was divided into several parts, which were fed to the network sequentially. This was done to improve the speed and memory usage of training. During training, the "hard" training samples were identified and their frequency of presentation was increased in subsequent training iterations.

Networks of seven different sizes were trained and evaluated in order to find a suitable network size. For each network size, ten networks were used, and four performance indicators were recorded: mean error, maximum error, linear correlation coefficient and rank-order correlation coefficient. The average performance indicator for the ten networks was also computed. The results are shown in Tables 5.6 and 5.7. The group average scores are shown in Figures 5.3 and 5.4; more results are available in Section A.2 of the Appendix.

These results show the performance differences between different network sizes. The best group performance was obtained with a network of size 6-10-5-1. However, similar results were obtained with a network of size 6-6-6-1 (i.e. 6 inputs, 6 neurons in hidden layer 1, 6 neurons in hidden layer 2, and one output), which has fewer neurons and weights. Hence, we selected this network size for further analysis.

5.3.3 Analysis and Comparison

From the experiments presented in Sections 5.3.1 and 5.3.2, we have identified a suitable feature vector and network size. The selected network has the following properties:

- *Input feature vector* $[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y, MSE]$.
- *Network size* 6-6-6-1.
Table 5.6: Performance of network size (6-6-6-1).

<table>
<thead>
<tr>
<th>Network</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.114</td>
<td>20.805</td>
<td>0.975</td>
<td>0.951</td>
</tr>
<tr>
<td>2</td>
<td>3.850</td>
<td>14.992</td>
<td>0.980</td>
<td>0.952</td>
</tr>
<tr>
<td>3</td>
<td>3.892</td>
<td>17.299</td>
<td>0.978</td>
<td>0.948</td>
</tr>
<tr>
<td>4</td>
<td>3.732</td>
<td>17.635</td>
<td>0.980</td>
<td>0.956</td>
</tr>
<tr>
<td>5</td>
<td>4.166</td>
<td>20.511</td>
<td>0.975</td>
<td>0.944</td>
</tr>
<tr>
<td>6</td>
<td>4.073</td>
<td>19.052</td>
<td>0.977</td>
<td>0.948</td>
</tr>
<tr>
<td>7</td>
<td>3.678</td>
<td>15.214</td>
<td>0.981</td>
<td>0.955</td>
</tr>
<tr>
<td>8</td>
<td>3.761</td>
<td>17.473</td>
<td>0.980</td>
<td>0.951</td>
</tr>
<tr>
<td>9</td>
<td>3.745</td>
<td>16.608</td>
<td>0.980</td>
<td>0.955</td>
</tr>
<tr>
<td>10</td>
<td>3.768</td>
<td>16.234</td>
<td>0.979</td>
<td>0.951</td>
</tr>
<tr>
<td>Mean value</td>
<td>3.878</td>
<td>17.582</td>
<td>0.979</td>
<td>0.951</td>
</tr>
</tbody>
</table>

Table 5.7: Performance of network size (6-10-5-1).

<table>
<thead>
<tr>
<th>Network</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.987</td>
<td>15.024</td>
<td>0.978</td>
<td>0.956</td>
</tr>
<tr>
<td>2</td>
<td>3.922</td>
<td>16.862</td>
<td>0.978</td>
<td>0.953</td>
</tr>
<tr>
<td>3</td>
<td>3.797</td>
<td>18.366</td>
<td>0.979</td>
<td>0.956</td>
</tr>
<tr>
<td>4</td>
<td>3.626</td>
<td>17.157</td>
<td>0.981</td>
<td>0.955</td>
</tr>
<tr>
<td>5</td>
<td>3.987</td>
<td>15.024</td>
<td>0.978</td>
<td>0.956</td>
</tr>
<tr>
<td>6</td>
<td>3.922</td>
<td>16.862</td>
<td>0.978</td>
<td>0.953</td>
</tr>
<tr>
<td>7</td>
<td>3.797</td>
<td>18.366</td>
<td>0.979</td>
<td>0.956</td>
</tr>
<tr>
<td>8</td>
<td>3.626</td>
<td>17.157</td>
<td>0.981</td>
<td>0.955</td>
</tr>
<tr>
<td>9</td>
<td>3.891</td>
<td>20.320</td>
<td>0.978</td>
<td>0.948</td>
</tr>
<tr>
<td>10</td>
<td>3.814</td>
<td>16.571</td>
<td>0.979</td>
<td>0.958</td>
</tr>
<tr>
<td>Mean value</td>
<td>3.837</td>
<td>17.171</td>
<td>0.979</td>
<td>0.955</td>
</tr>
</tbody>
</table>
Figure 5.3: Mean and max error for the test set depending on different network sizes.

Figure 5.4: Correlation coefficients for the test set depending on different network sizes.
Here, we present a detailed analysis in terms of prediction error and correlation between the network-predicted MOS and the actual MOS. The ten-fold cross-validation technique [37] was used in this analysis. The entire dataset was divided into ten equal subsets; and ten networks were constructed. For each iteration, a neural network was trained with a new combination of nine subsets and then tested on the remaining subset. Finally, the overall error rate was evaluated across all networks. The N-fold cross-validation technique is useful in determining reliable estimates of network generalization capability; it has been used extensively in the neural network field.

Figure 5.5(a) shows the predicted MOS versus the actual MOS for the training data; Figure 5.5(b) shows the error histogram between the network-predicted MOS and the actual MOS. The same types of plots for the test data are shown in 5.6(b) and 5.6(a). These figures show that the network approximates the MOS with low errors. In addition, there is a strong correlation, as illustrated by the straight lines in Figures 5.5(a) and 5.6(a), between the network-predicted MOS and the MOS. These observations are also confirmed by the results shown in Table 5.8. We can conclude that the neural network produces image quality scores that are very closely related to human perception of image quality. It is interesting to note that the mean error on the test set (3.74) is lower than the mean error on the training set (4.09), and the correlation coefficient between the predicted MOS and the actual MOS is higher on the test set (0.9744) than on the training set (0.9631). These results show that the trained network generalised well from the training set.

5.3.4 Analysis Using the ANOVA Tool

To determine the consistency of neural network prediction, we use the ANOVA tool, which has been described in Section 3.3.3. The actual MOS values between 0 and 100 are divided into five groups, each group spans an interval of length 20. For each

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Training Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Error</td>
<td>4.09</td>
<td>3.74</td>
</tr>
<tr>
<td>Maximum Error</td>
<td>14.52</td>
<td>15.16</td>
</tr>
<tr>
<td>Correlation Coefficient</td>
<td>0.9631</td>
<td>0.9744</td>
</tr>
<tr>
<td>Ranked Correlation Coefficient</td>
<td>0.9716</td>
<td>0.9691</td>
</tr>
</tbody>
</table>
Figure 5.5: Comparison of the network-predicted MOS and the actual MOS for the training set.

Figure 5.6: Comparison of the network-predicted MOS and the actual MOS for the test data.
group, two vectors are formed: one consisting of the actual MOS values, and the other consisting of the corresponding network-predicted MOS values. Each vector is analyzed to find the group mean and standard deviation. The results are shown in Figure 5.7. The similarity between the box plots of the actual and network-predicted MOS values indicates that the neural network produces MOS values that are very close to the subjective MOS. However, Figure 5.7 shows that there is a larger variance in the predicted MOS for images with high MOS (i.e., high quality). The ANOVA tool also gives F-value, which is high if the overlaps between groups are small and vice versa. In our experiment, the actual MOS has an F-value of 1051.6, and the network-predicted MOS has an F-value of 589.0.

![Box plots of actual and network-predicted MOS values](image)

(a) actual MOS  (b) network-predicted MOS

Figure 5.7: Comparison of the actual MOS and the network-predicted MOS using ANOVA.

### 5.4 Pixel-based Neural Network Approach

In this section we develop a network that produces an estimate of the MOS directly from raw pixel values. This means no preprocessing is needed, but the network must handle a large feature vector (128 pixel values in our case). In this section, we first identify an appropriate network size, and then analyze in detail the network performance in terms of estimation error and correlation between the predicted and the actual MOS.
5.4.1 Network Size Selection

We used the dataset described in Section 5.2.2, and the Conjugate Gradient algorithm for training. This algorithm is more efficient in memory usage than the Levenberg-Marquardt algorithm where the network has a large size and the training set must contain many samples, as in our case. Similar steps as described in Section 5.3.2 were taken in this experiment.

We examined a number of network sizes, the mean and maximum errors on the test set for different numbers of hidden neurons are shown in Figure 5.8. The figure shows that in general the error decreases as the number of hidden neuron increases. However, the change in error is very small for network sizes beyond 128-45-25-1. Based on this result, we selected a network of size 128-45-25-1 in all subsequent analysis.

![Figure 5.8: Pixel-based ANN approach: mean and maximum errors on the test set for different network sizes.](image)

5.4.2 Error and Correlation Analysis of Network Output

The ten-fold cross-validation technique, as described in Section 5.3.3, was used. The differences between the network-predicted MOS and the actual MOS on the training set are shown in Figure 5.9. The differences between the network-predicted MOS and the actual MOS on the test set are shown in Figure 5.10.

The performance indicators (mean error, maximum error, correlation coefficients) of the trained network are shown in Table 5.9. Figure 5.10(a) and Table
Figure 5.9: Comparison of the network-predicted MOS and the actual MOS for the training data.

Figure 5.10: Comparison of the network-predicted MOS and the actual MOS for the test data.
5.9 shows a strong correlation between the predicted and the actual MOSs. Their normal correlation coefficient over the test set is 0.9784. The network achieved a mean absolute error of 3.45.

Table 5.9: Performance indicators of pixel-based network.

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Training Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Error</td>
<td>3.10</td>
<td>3.45</td>
</tr>
<tr>
<td>Maximum Error</td>
<td>11.13</td>
<td>12.43</td>
</tr>
<tr>
<td>Correlation Coefficient</td>
<td>0.9840</td>
<td>0.9784</td>
</tr>
<tr>
<td>Ranked Correlation Coefficient</td>
<td>0.9796</td>
<td>0.9733</td>
</tr>
</tbody>
</table>

5.4.3 Analysis Using the ANOVA Tool

To evaluate the consistency between the predicted and the actual MOS, we used the ANOVA statistical tool; the same steps as in Section 5.3.4 were applied. The box plots produced by the ANOVA tool are shown in Figure 5.11. The box plots of the actual MOS and network-predicted MOS have a similar shape, which means the network has produced a consistent score compared to the actual MOS. However, there is small deviation between the actual MOS and the network output for the group with low image quality. In addition, the group variances of the network output is always larger than those of the actual MOS. The network-predicted MOS has an F-value of 697.1, whereas the actual MOS has an F-value of 1051.6.

5.5 Comparison of NN Approaches and MSSIM

In Chapter 3, we have seen that, compared to other objective measures, the MSSIM is the most consistent with the subjective measure MOS. In this section, we present a performance comparison of the two neural-network approaches and the MSSIM. The comparisons were done using two sets of images:

- Set 1 is the test set described in Section 5.2, extracted from Wang's database;
- Set 2 consists of Lena image corrupted with different types of noise.
Figure 5.11: Comparison of the actual MOS and the network-predicted MOS using ANOVA.

5.5.1 Results and Analysis for Set 1

The first performance comparison involved comparing the MSSIM with the NN approaches using the MOS as a reference. The comparison involved the following five image quality scores:

- mean opinion score (MOS), which is used as the reference score;
- mean structural similarity index (MSSIM);
- approximation of the MOS by a third-order polynomial of the MSSIM;
- feature-based neural network output;
- pixel-based neural network output.

Because the MSSIM has a nonlinear relationship with the MOS, it is suggested that to improve the performance a nonlinear regression fit be used to predict the MOS values using the MSSIM scores. To determine the nonlinear mapping between the MSSIM scores and the MOS, a third order polynomial fit is applied; note that we could have used logistic regression fit as well, but this does not change the results drastically. The approximation of the MOS by a third-order polynomial can be written as follows:

\[
y_{MOS} = a_3x^3 - a_2x^2 + a_1x + a_0 ,
\]  

(5.2)
where $y_{MOS}$ is the predicted MOS value, $x$ is the value of the MSSIM, and $\{a_0, a_1, a_2, a_3\}$ are the polynomial coefficients. The coefficients are computed as the least-square solution to a system of linear equations, defined on a training set. The best-fit polynomial on our training set was found to be:

$$y_{MOS} = 284.00x^3 - 348.52x^2 + 152.60x.$$  \hfill (5.3)

Figure 5.12(a) shows the MSSIM and the MOS values for the test set, with the best-fit polynomial function superimposed; Figure 5.12(b) shows the error histogram for the polynomial-approximated MOS and the actual MOS. The box plots for different image quality groups, according to the MOS and polynomial MSSIM scores, are shown in Figure 5.11(a). The box plots of MSSIM have different shapes compared to those of the predicted MOS values.

Figure 5.12: Comparison of polynomial-approximated MOS and the actual MOS.

Four performance indicators including the mean error, correlation coefficient, rank-order correlation coefficient, and F-value are presented in Table 5.10. With the MOS being the reference, it has perfect performance indicators (mean error = 0, correlation coefficient = 1, and F-value = 1051.6). Compared to the MSSIM, there is a significant improvement in the performance of the polynomial MSSIM (p-MSSIM), in terms of F-value and correlation.

Among the four image quality measures, MSSIM, p-MSSIM, PBNN, and FBNN, the pixel-based neural network (PBNN) has the best performance indicators. It has the highest F-value (697.1) and the highest correlation coefficients (0.9784 and
Figure 5.13: Different image quality groups according to the MOS and polynomial MSSIM scores.

Table 5.10: Comparison of five image quality measures: MOS, MSSIM, polynomial MSSIM (p-MSSIM), feature-based NN (FBNN), and pixel-based NN (PBNN).

<table>
<thead>
<tr>
<th>Performance Indicator</th>
<th>MOS</th>
<th>MSSIM</th>
<th>p-MSSIM</th>
<th>FBNN</th>
<th>PBNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean error</td>
<td>0</td>
<td>N/A</td>
<td>5.12</td>
<td>3.74</td>
<td>3.45</td>
</tr>
<tr>
<td>Correlation coefficient</td>
<td>1</td>
<td>0.9008</td>
<td>0.9551</td>
<td>0.9744</td>
<td>0.9784</td>
</tr>
<tr>
<td>Ranked corr. coef.</td>
<td>1</td>
<td>0.9557</td>
<td>0.9538</td>
<td>0.9691</td>
<td>0.9733</td>
</tr>
<tr>
<td>F-value</td>
<td>1051.6</td>
<td>357.0</td>
<td>519.2</td>
<td>589.0</td>
<td>697.1</td>
</tr>
</tbody>
</table>
Table 5.11: Five image quality measures: MOS, MSSIM, polynomial MSSIM (p-MSSIM), feature-based NN (FBNN), and pixel-based NN (PBNN) for Test Set 2.

<table>
<thead>
<tr>
<th>Image</th>
<th>(% Agreed) MOS Rank</th>
<th>MSSIM (Rank)</th>
<th>p-MSSIM (Rank)</th>
<th>FBNN (Rank)</th>
<th>PBNN (Rank)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>100 (1)</td>
<td>1.00 (1)</td>
<td>88.08 (1)</td>
<td>100.00 (1)</td>
<td>98.83 (1)</td>
</tr>
<tr>
<td>Salt &amp; pepper</td>
<td>100 (2)</td>
<td>0.87 (2)</td>
<td>56.77 (2)</td>
<td>56.16 (2)</td>
<td>57.49 (2)</td>
</tr>
<tr>
<td>Gaussian</td>
<td>64 (3)</td>
<td>0.27 (4)</td>
<td>21.27 (4)</td>
<td>22.30 (4)</td>
<td>44.30 (3)</td>
</tr>
<tr>
<td>Blurring</td>
<td>64 (4)</td>
<td>0.78 (3)</td>
<td>41.50 (3)</td>
<td>39.15 (3)</td>
<td>42.86 (4)</td>
</tr>
</tbody>
</table>

0.9733); this means that the pixel-based neural network output is the most consistent with the reference MOS. Both feature-based and pixel-based neural networks outperform the MSSIM and the polynomial MSSIM in terms of F-values and correlation with the actual MOS. This proves that our neural network approaches produce image quality scores that are more consistent with perceptual image quality.

5.5.2 Results and Analysis for Set 2

An experiment was conducted to test the ability of the different IQA measures to work with with other types of distortions; Wang’s database contains only JPEG compressed images, and hence it does not cover other types of distortions. The test set is constructed from an original image, the Lena image, corrupted with three different types of noise: salt & pepper, Gaussian, and blurring. The original image and the three distorted images are shown in Figure 5.14.

Twelve people were asked to rank these images in terms of their quality. Column 1 in Table 5.11 shows the majority ranking and the percentage of people agreeing with this ranking. According to the subjective assessment, the ranking, from the highest to the lowest, is as follows: original image → salt & pepper image → Gaussian image → blurring image. For each image in the set, four objective quality scores were computed: MSSIM, polynomial MSSIM, pixel-based NN output, and feature-based NN output. The scores are also presented in Table 5.11; the numbers in bracket in the last four columns are the image quality rankings according to each computed score (1 is the highest quality and 4 is the lowest quality). Among the four IQA methods, only the pixel-based neural network produced the same ranking as the subjective ranking.

Note that, the perceptual ranking between the image corrupted by Gaussian noise
(a) original image: Lena
(b) salt & pepper noise
(c) Gaussian
(d) blurring \((7 \times 7)\)

Figure 5.14: Four distorted images.
(Figure 5.14(c)) and the blurred image (Figure 5.14(d)) is not a clear-cut decision: eight out of twelve participants considered the Gaussian image to have higher quality, but the rest said otherwise. For these two images, the pixel-based neural network also produces very close scores (relative to the maximum quality score of 98.83). In contrast, the MSSIM, polynomial MSSIM, and feature-based neural network produce scores that are significantly far apart (relative to the corresponding maximum quality scores). Therefore, it appears that the pixel-based NN produces image quality scores that are more consistent with the subjective assessment.

5.6 Conclusion

In this chapter, two approaches of using neural network to predict the mean opinion score of image quality have been proposed. The approaches differ in the choices of network inputs: the first approach uses statistical features, and the second approach uses raw pixel values.

For the first approach, networks with different feature vectors were developed and evaluated. The best results were obtained with the feature vector consisting of two block means, two block standard deviations, block covariance, and block MSE. Using this feature vector, we evaluated several networks in order to find the most suitable network sizes. We found that a network consisting of two hidden layers with 6 neurons in each layer performed better compared to other network sizes. Comparison results of the best network's output and the actual MOS show that the network can approximate very closely the perceptual MOS. The feature-based network is found to outperform the MSSIM, which is the best among the five objective measures tested in Chapter 3.

In the second approach, neural networks are trained to generate the MOS directly from pixel values. Training networks becomes a harder problem because of the high dimensionality of the network inputs. We found that the network size of 45 neurons in the first hidden layer and 25 neurons in the second hidden layer gives the best results. Our experiments show that none of the tested objective measures has better correlation coefficients with the MOS than the trained neural network. We also found that the pixel-based ANN approach performs better than the feature-based ANN approach.
Chapter 6

Conclusions And Future Work

6.1 Thesis Summary

This thesis deals with automatic techniques for assessing the visual quality of digital images that have been degraded by artifacts or noise during image capture or compression. Chapter 2 presents a review of existing image quality assessment (IQA) methods, including the mean-squared error (MSE), the peak signal-to-noise ratio (PSNR), the universal image quality index (UIQI), the mean structural similarity index (MSSIM), and the mean opinion score (MOS); the advantages and disadvantages of these techniques were highlighted. In Chapter 3, a comparative study of the different IQA measures was conducted. This study showed that the MSSIM is the objective measure that is most consistent with human perceptual quality; it is the subjective measure which has the strongest correlation with the MOS. However, the MSSIM's performance degrades quickly for images with poor image quality.

The aim of this thesis is to develop objective IQA measures using neural network. Artificial neural networks are computational models of the human brain that have been used extensively in diverse applications, including computer vision and image processing. Chapter 4 presents an overview of artificial neural networks and some of their training algorithms. Furthermore, in this chapter neural networks are used to approximate the MSSIM measure of image quality. Two approaches were examined: one uses statistical feature as inputs, and the other uses raw pixel as the inputs to the network. Both approaches give promising results; the neural networks predicted values have very small errors, and the predicted values are highly correlated with the actual MSSIM values. However, it was found that the pixel-based approach outperforms slightly the statistical feature-based approach.
Based on these preliminary experiments, two neural network approaches were developed for the prediction of subjective image quality assessment. In Chapter 5, multilayer perceptions are designed to approximate the mean opinion score. The MOS is obtained by averaging the subjective IQA results of several human observers. Therefore, approximating if a neural network learns to approximate the MOS, it would have learnt some aspects of human perception of image quality. Again, two neural network approaches were investigated. The first approach uses statistical features, computed from the original and the distorted images, as inputs to the neural network; these features include mean, standard deviation, covariance, contrast, and mean square error. The second approach, on the other hand, uses raw pixel values directly as the network inputs.

Initially several network architectures were trained and tested on a database of JPEG and JPEG-2000 compressed images. For the feature-based approach, it was found that a neural network with two hidden layers and 6 neurons in each layer is adequate for predicting the MOS values. This architecture was then tested using ten-fold cross-validation. The results show that the network achieves a mean test error of 3.74, a linear correlation coefficient of 0.9744, and a rank-order correlation coefficient of 0.9691. For the second approach, which uses the pixel values as inputs, it has a mean test error of 3.45, a linear correlation coefficient of 0.9784, and a rank-order correlation coefficient of 0.9733. Overall its performance is slightly better than that of the feature-based neural network. However, training of the pixel-based neural network is more difficult because of the high dimensionality of the input vector. The findings of these experiments suggest that image quality assessment requires more information than that provided by simple image statistics.

Finally a comparison was conducted between the neural networks and the MSSIM. It was found that both neural network approaches outperform the MSSIM measure in terms of mean absolute error and correlation metrics. A statistical comparison using ANOVA also confirm the superiority of the neural network approaches over the MSSIM.

6.2 Future Work

In this thesis, we presented a new approach to image quality assessment using neural networks. The results show that neural networks have the potential to approximate
the subjective image quality with high accuracy. However, further research is re-
quired to refine the neural networks and find a universal image quality that works
with different types of distortions. To this end, we suggest a number of further
research directions:

• Investigate other statistical features besides the mean, standard deviation,
  mean square error, and covariance. Possible options include features extracted
  from the frequency-domain. Since the performance of the HVS varies with
  spatial frequency, using features from different frequency channels might help
  the learning process and improve the generalisation of the network.

• Develop a more comprehensive database of images together with accurate MOS
  values. The database can include other types of noise, and could use a more
  refined MOS scores.

• Investigating new techniques for selecting network training data in order to
  improve the generalization capability.

• Examine other supervised-learning algorithms.

• Investigate other neural network types such as convolutional neural networks
  and radial basis functions.

• In this work, Matlab has been used to develop a ANN-based IQA prediction
  methods. For future work, it would be more adequate to implement the tools
  in other elaborated languages like C++ to give an idea about the CPU time.
Bibliography


Appendix A

Further Results from Chapter 5

A.1 Additional Results for Subsection 5.3.1

Table A.1: Performance of feature vector 1: $[\sigma_{zy}, \sigma_{x}, \sigma_{y}]$.

<table>
<thead>
<tr>
<th>Network nr.</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net. 1</td>
<td>5.412</td>
<td>23.904</td>
<td>0.961</td>
<td>0.936</td>
</tr>
<tr>
<td>Net. 2</td>
<td>5.267</td>
<td>21.990</td>
<td>0.964</td>
<td>0.926</td>
</tr>
<tr>
<td>Net. 3</td>
<td>5.405</td>
<td>23.964</td>
<td>0.961</td>
<td>0.933</td>
</tr>
<tr>
<td>Net. 4</td>
<td>5.715</td>
<td>23.085</td>
<td>0.957</td>
<td>0.941</td>
</tr>
<tr>
<td>Net. 5</td>
<td>5.412</td>
<td>23.904</td>
<td>0.961</td>
<td>0.936</td>
</tr>
<tr>
<td>Net. 6</td>
<td>5.267</td>
<td>21.990</td>
<td>0.964</td>
<td>0.926</td>
</tr>
<tr>
<td>Net. 7</td>
<td>5.405</td>
<td>23.964</td>
<td>0.961</td>
<td>0.933</td>
</tr>
<tr>
<td>Net. 8</td>
<td>5.715</td>
<td>23.085</td>
<td>0.957</td>
<td>0.941</td>
</tr>
<tr>
<td>Net. 9</td>
<td>5.702</td>
<td>22.566</td>
<td>0.959</td>
<td>0.931</td>
</tr>
<tr>
<td>Net. 10</td>
<td>5.277</td>
<td>22.454</td>
<td>0.963</td>
<td>0.938</td>
</tr>
<tr>
<td>Mean value</td>
<td>5.458</td>
<td>23.091</td>
<td>0.961</td>
<td>0.934</td>
</tr>
</tbody>
</table>
Table A.2: Performance of feature vector 2: $[\sigma_{xy}, \sigma_x, \sigma_y, \text{MSE}]$.

<table>
<thead>
<tr>
<th>Network nr.</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net. 1</td>
<td>3.920</td>
<td>19.151</td>
<td>0.978</td>
<td>0.952</td>
</tr>
<tr>
<td>Net. 2</td>
<td>4.121</td>
<td>17.485</td>
<td>0.977</td>
<td>0.954</td>
</tr>
<tr>
<td>Net. 3</td>
<td>3.927</td>
<td>19.592</td>
<td>0.978</td>
<td>0.951</td>
</tr>
<tr>
<td>Net. 4</td>
<td>4.073</td>
<td>21.154</td>
<td>0.974</td>
<td>0.952</td>
</tr>
<tr>
<td>Net. 5</td>
<td>4.013</td>
<td>18.642</td>
<td>0.976</td>
<td>0.944</td>
</tr>
<tr>
<td>Net. 6</td>
<td>4.107</td>
<td>18.219</td>
<td>0.977</td>
<td>0.949</td>
</tr>
<tr>
<td>Net. 7</td>
<td>4.135</td>
<td>18.055</td>
<td>0.975</td>
<td>0.946</td>
</tr>
<tr>
<td>Net. 8</td>
<td>3.827</td>
<td>18.091</td>
<td>0.979</td>
<td>0.954</td>
</tr>
<tr>
<td>Net. 9</td>
<td>4.050</td>
<td>18.118</td>
<td>0.977</td>
<td>0.953</td>
</tr>
<tr>
<td>Net. 10</td>
<td>4.018</td>
<td>18.631</td>
<td>0.977</td>
<td>0.950</td>
</tr>
<tr>
<td>Mean value</td>
<td>4.019</td>
<td>18.714</td>
<td>0.977</td>
<td>0.951</td>
</tr>
</tbody>
</table>

Table A.3: Performance of feature vector 3: $[\sigma_{xy}, \sigma_x, \sigma_y, c_x, c_y]$.

<table>
<thead>
<tr>
<th>Network nr.</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net. 1</td>
<td>5.784</td>
<td>23.957</td>
<td>0.956</td>
<td>0.935</td>
</tr>
<tr>
<td>Net. 2</td>
<td>5.892</td>
<td>24.816</td>
<td>0.954</td>
<td>0.931</td>
</tr>
<tr>
<td>Net. 3</td>
<td>5.838</td>
<td>25.036</td>
<td>0.955</td>
<td>0.932</td>
</tr>
<tr>
<td>Net. 4</td>
<td>5.893</td>
<td>24.720</td>
<td>0.955</td>
<td>0.932</td>
</tr>
<tr>
<td>Net. 5</td>
<td>5.470</td>
<td>26.078</td>
<td>0.960</td>
<td>0.924</td>
</tr>
<tr>
<td>Net. 6</td>
<td>5.725</td>
<td>24.423</td>
<td>0.958</td>
<td>0.919</td>
</tr>
<tr>
<td>Net. 7</td>
<td>5.687</td>
<td>22.857</td>
<td>0.956</td>
<td>0.938</td>
</tr>
<tr>
<td>Net. 8</td>
<td>5.930</td>
<td>26.908</td>
<td>0.954</td>
<td>0.924</td>
</tr>
<tr>
<td>Net. 9</td>
<td>5.728</td>
<td>26.374</td>
<td>0.956</td>
<td>0.928</td>
</tr>
<tr>
<td>Net. 10</td>
<td>5.951</td>
<td>27.851</td>
<td>0.953</td>
<td>0.921</td>
</tr>
<tr>
<td>Mean value</td>
<td>5.790</td>
<td>25.302</td>
<td>0.956</td>
<td>0.928</td>
</tr>
</tbody>
</table>
Table A.4: Performance of feature vector 4: $[\mu_x, \mu_y, \sigma_{xy}, \sigma_x, \sigma_y]$.

<table>
<thead>
<tr>
<th>Network nr.</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net. 1</td>
<td>5.415</td>
<td>20.200</td>
<td>0.963</td>
<td>0.939</td>
</tr>
<tr>
<td>Net. 2</td>
<td>5.654</td>
<td>26.140</td>
<td>0.957</td>
<td>0.928</td>
</tr>
<tr>
<td>Net. 3</td>
<td>5.697</td>
<td>19.566</td>
<td>0.961</td>
<td>0.933</td>
</tr>
<tr>
<td>Net. 4</td>
<td>5.138</td>
<td>23.869</td>
<td>0.965</td>
<td>0.931</td>
</tr>
<tr>
<td>Net. 5</td>
<td>5.415</td>
<td>20.200</td>
<td>0.963</td>
<td>0.939</td>
</tr>
<tr>
<td>Net. 6</td>
<td>5.654</td>
<td>26.140</td>
<td>0.957</td>
<td>0.928</td>
</tr>
<tr>
<td>Net. 7</td>
<td>5.697</td>
<td>19.566</td>
<td>0.961</td>
<td>0.933</td>
</tr>
<tr>
<td>Net. 8</td>
<td>5.138</td>
<td>23.869</td>
<td>0.965</td>
<td>0.931</td>
</tr>
<tr>
<td>Net. 9</td>
<td>5.415</td>
<td>20.200</td>
<td>0.963</td>
<td>0.939</td>
</tr>
<tr>
<td>Net. 10</td>
<td>5.654</td>
<td>26.140</td>
<td>0.957</td>
<td>0.928</td>
</tr>
<tr>
<td>Mean value</td>
<td>5.488</td>
<td>22.589</td>
<td>0.961</td>
<td>0.933</td>
</tr>
</tbody>
</table>

A.2 Additional Results for Subsection 5.3.2

Table A.5: Performance of network size (6-4-4-1).

<table>
<thead>
<tr>
<th>Network nr.</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net. 1</td>
<td>4.151</td>
<td>18.291</td>
<td>0.976</td>
<td>0.947</td>
</tr>
<tr>
<td>Net. 2</td>
<td>4.199</td>
<td>18.829</td>
<td>0.975</td>
<td>0.950</td>
</tr>
<tr>
<td>Net. 3</td>
<td>4.043</td>
<td>17.998</td>
<td>0.977</td>
<td>0.948</td>
</tr>
<tr>
<td>Net. 4</td>
<td>3.962</td>
<td>17.938</td>
<td>0.978</td>
<td>0.953</td>
</tr>
<tr>
<td>Net. 5</td>
<td>4.076</td>
<td>17.072</td>
<td>0.977</td>
<td>0.956</td>
</tr>
<tr>
<td>Net. 6</td>
<td>3.717</td>
<td>16.608</td>
<td>0.980</td>
<td>0.954</td>
</tr>
<tr>
<td>Net. 7</td>
<td>3.992</td>
<td>18.017</td>
<td>0.978</td>
<td>0.951</td>
</tr>
<tr>
<td>Net. 8</td>
<td>3.900</td>
<td>18.389</td>
<td>0.978</td>
<td>0.954</td>
</tr>
<tr>
<td>Net. 9</td>
<td>4.093</td>
<td>17.275</td>
<td>0.976</td>
<td>0.943</td>
</tr>
<tr>
<td>Net. 10</td>
<td>4.293</td>
<td>20.249</td>
<td>0.974</td>
<td>0.947</td>
</tr>
<tr>
<td>Mean value</td>
<td>4.043</td>
<td>18.067</td>
<td>0.977</td>
<td>0.950</td>
</tr>
</tbody>
</table>
Table A.6: Performance of network size (6-5-3-1).

<table>
<thead>
<tr>
<th>Network nr.</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net. 1</td>
<td>4.213</td>
<td>17.457</td>
<td>0.975</td>
<td>0.956</td>
</tr>
<tr>
<td>Net. 2</td>
<td>4.184</td>
<td>18.491</td>
<td>0.976</td>
<td>0.953</td>
</tr>
<tr>
<td>Net. 3</td>
<td>3.611</td>
<td>17.333</td>
<td>0.981</td>
<td>0.954</td>
</tr>
<tr>
<td>Net. 4</td>
<td>4.085</td>
<td>15.770</td>
<td>0.978</td>
<td>0.957</td>
</tr>
<tr>
<td>Net. 5</td>
<td>3.839</td>
<td>17.015</td>
<td>0.978</td>
<td>0.951</td>
</tr>
<tr>
<td>Net. 6</td>
<td>4.009</td>
<td>18.218</td>
<td>0.977</td>
<td>0.945</td>
</tr>
<tr>
<td>Net. 7</td>
<td>4.144</td>
<td>17.784</td>
<td>0.977</td>
<td>0.955</td>
</tr>
<tr>
<td>Net. 8</td>
<td>3.828</td>
<td>16.593</td>
<td>0.980</td>
<td>0.950</td>
</tr>
<tr>
<td>Net. 9</td>
<td>4.332</td>
<td>18.372</td>
<td>0.974</td>
<td>0.939</td>
</tr>
<tr>
<td>Net. 10</td>
<td>4.083</td>
<td>19.697</td>
<td>0.975</td>
<td>0.950</td>
</tr>
<tr>
<td>Mean value</td>
<td>4.033</td>
<td>17.673</td>
<td>0.977</td>
<td>0.951</td>
</tr>
</tbody>
</table>

Table A.7: Performance of network size (6-8-6-1).

<table>
<thead>
<tr>
<th>Network nr.</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net. 1</td>
<td>3.876</td>
<td>17.681</td>
<td>0.979</td>
<td>0.957</td>
</tr>
<tr>
<td>Net. 2</td>
<td>4.229</td>
<td>13.037</td>
<td>0.977</td>
<td>0.947</td>
</tr>
<tr>
<td>Net. 3</td>
<td>3.967</td>
<td>19.511</td>
<td>0.977</td>
<td>0.951</td>
</tr>
<tr>
<td>Net. 4</td>
<td>3.951</td>
<td>17.298</td>
<td>0.978</td>
<td>0.950</td>
</tr>
<tr>
<td>Net. 5</td>
<td>3.960</td>
<td>17.870</td>
<td>0.978</td>
<td>0.952</td>
</tr>
<tr>
<td>Net. 6</td>
<td>3.972</td>
<td>18.907</td>
<td>0.977</td>
<td>0.949</td>
</tr>
<tr>
<td>Net. 7</td>
<td>3.907</td>
<td>20.588</td>
<td>0.978</td>
<td>0.948</td>
</tr>
<tr>
<td>Net. 8</td>
<td>4.001</td>
<td>18.403</td>
<td>0.977</td>
<td>0.952</td>
</tr>
<tr>
<td>Net. 9</td>
<td>3.961</td>
<td>16.893</td>
<td>0.978</td>
<td>0.953</td>
</tr>
<tr>
<td>Net. 10</td>
<td>3.833</td>
<td>15.665</td>
<td>0.980</td>
<td>0.954</td>
</tr>
<tr>
<td>Mean value</td>
<td>3.966</td>
<td>17.585</td>
<td>0.978</td>
<td>0.951</td>
</tr>
</tbody>
</table>
Table A.8: Performance of network size (6-8-8-1).

<table>
<thead>
<tr>
<th>Network nr.</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net. 1</td>
<td>4.002</td>
<td>17.390</td>
<td>0.978</td>
<td>0.951</td>
</tr>
<tr>
<td>Net. 2</td>
<td>4.030</td>
<td>19.207</td>
<td>0.977</td>
<td>0.952</td>
</tr>
<tr>
<td>Net. 3</td>
<td>4.085</td>
<td>18.245</td>
<td>0.977</td>
<td>0.949</td>
</tr>
<tr>
<td>Net. 4</td>
<td>3.705</td>
<td>16.007</td>
<td>0.980</td>
<td>0.954</td>
</tr>
<tr>
<td>Net. 5</td>
<td>4.002</td>
<td>17.390</td>
<td>0.978</td>
<td>0.951</td>
</tr>
<tr>
<td>Net. 6</td>
<td>4.030</td>
<td>19.207</td>
<td>0.977</td>
<td>0.952</td>
</tr>
<tr>
<td>Net. 7</td>
<td>4.085</td>
<td>18.245</td>
<td>0.977</td>
<td>0.949</td>
</tr>
<tr>
<td>Net. 8</td>
<td>3.705</td>
<td>16.007</td>
<td>0.980</td>
<td>0.954</td>
</tr>
<tr>
<td>Net. 9</td>
<td>3.867</td>
<td>19.904</td>
<td>0.977</td>
<td>0.954</td>
</tr>
<tr>
<td>Net. 10</td>
<td>3.883</td>
<td>15.464</td>
<td>0.979</td>
<td>0.954</td>
</tr>
<tr>
<td>Mean value</td>
<td>3.939</td>
<td>17.707</td>
<td>0.978</td>
<td>0.952</td>
</tr>
</tbody>
</table>

Table A.9: Performance of network size (6-6-8-1).

<table>
<thead>
<tr>
<th>Network nr.</th>
<th>Mean error</th>
<th>Max error</th>
<th>Correlation</th>
<th>Ranked correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net. 1</td>
<td>3.973</td>
<td>16.540</td>
<td>0.978</td>
<td>0.951</td>
</tr>
<tr>
<td>Net. 2</td>
<td>4.234</td>
<td>16.876</td>
<td>0.976</td>
<td>0.950</td>
</tr>
<tr>
<td>Net. 3</td>
<td>4.013</td>
<td>19.158</td>
<td>0.976</td>
<td>0.946</td>
</tr>
<tr>
<td>Net. 4</td>
<td>4.149</td>
<td>18.126</td>
<td>0.976</td>
<td>0.952</td>
</tr>
<tr>
<td>Net. 5</td>
<td>4.160</td>
<td>18.580</td>
<td>0.975</td>
<td>0.950</td>
</tr>
<tr>
<td>Net. 6</td>
<td>3.840</td>
<td>17.078</td>
<td>0.979</td>
<td>0.952</td>
</tr>
<tr>
<td>Net. 7</td>
<td>4.030</td>
<td>17.274</td>
<td>0.977</td>
<td>0.951</td>
</tr>
<tr>
<td>Net. 8</td>
<td>4.075</td>
<td>18.160</td>
<td>0.976</td>
<td>0.955</td>
</tr>
<tr>
<td>Net. 9</td>
<td>4.093</td>
<td>18.656</td>
<td>0.977</td>
<td>0.949</td>
</tr>
<tr>
<td>Net. 10</td>
<td>3.915</td>
<td>17.606</td>
<td>0.979</td>
<td>0.952</td>
</tr>
<tr>
<td>Mean value</td>
<td>4.048</td>
<td>17.806</td>
<td>0.977</td>
<td>0.951</td>
</tr>
</tbody>
</table>