UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

A Bag of Words Description Scheme for Image Quality Assessment

Miguel Francisco Fidalgo Fernandes

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores
(2º ciclo de estudos)

Orientador: Prof. Doutor António Manuel Gonçalves Pinheiro

Covilhã, Outubro de 2016

Acknowledgements

Ao Orientador, António Manuel Gonçalves Pinheiros pela paciência, conhecimento, motivação e tempo e que disponibilizou para que este trabalho fosse feito.
Ao Marco V. Bernardo, o aluno de doutoramento que tirou muitas das minhas dúvidas (algumas delas óbvias ou até ridículas).
Aos professores, de todos os ciclos de ensino desde o primário ao universitário, que permitiram que chegasse até aqui.
A todos os colegas de curso que se tornaram amigos, que conheci durante estes 5 anos.
Aos colegas da minha banda, Semilunar, que me aturaram estes anos todos.
À família pelo seu apoio constante, avós, tios, primos e principalmente aos meus pais e à pirralha que chamo de irmã (ou talvez à irmã que chamo de pirralha).

Resumo alargado

No nosso dia-a-dia as imagens são obtidas, processadas, comprimidas, guardadas, transmitidas e reproduzidas. Em qualquer destas operações podem ocorrer distorções que prejudicam a sua qualidade. A qualidade destas imagens pode ser medida de forma subjectiva, o que tem a desvantagem de serem necessários vários testes, a um número considerável de indivíduos para ser feita uma análise estatística da qualidade perceptual de uma imagem. Foram desenvolvidas várias métricas objectivas, que de alguma forma tentam modelar a percepção humana de qualidade. Todavia, em muitas aplicações a representação de percepção de qualidade humana dada por estas métricas fica aquém do desejável, razão porque se propõe neste trabalho usar modelos de reconhecimento de padrões que permitam uma maior aproximação.

Neste trabalho, são dadas definições para imagem e qualidade e algumas das dificuldades do estudo da qualidade de imagem são referidas. É referida a importância da qualidade de imagem como ramo de estudo, e são estudadas diversas métricas de qualidade.

São explicadas três métricas, uma delas que usa a qualidade original como referência (SSIM) e duas métricas sem referência (BRISQUE e QAC). Uma comparação é feita entre elas, mostrando-se uma grande discrepância de valores entre os dois tipos de métricas.

Para os testes feitos é usada a base de dados TID2013, que é muitas vezes considerada para estudos de qualidade de métricas devido à sua dimensão e ao facto de considerar um grande número de distorções. Neste trabalho também se fez um estudo dos tipos de distorção incluidos nesta base de dados e como é que eles são simulados.

São introduzidos também alguns conceitos teóricos de reconhecimento de padrões e alguns algoritmos relevantes no contexto da dissertação, são descritos como o K-means, KNN e as SVMs. Algoritmos de agregação de descritores como o "bag of words" e o "fisher-vectors" também são referidos.

Esta dissertação adiciona métodos de reconhecimento de padrões a métricas objectivas de qualidade de imagem. Uma nova técnica é proposta, baseada na divisão de imagens em células, nas quais uma métrica será calculada. Esta divisão permite obter descritores locais de qualidade que serão agregados usando "bag of words". Uma SVM com kernel RBF é treinada e testada na mesma base de dados e os resultados do modelo são mostrados usando cross-validation.

Os resultados são analisados usando as correlações de Pearson, Spearman e Kendall e o RMSE que permitem avaliar a proximidade entre a métrica desenvolvida e os resultados subjectivos.

Este modelo melhora os resultados obtidos com a métrica usada e demonstra uma nova forma de aplicar modelos de reconhecimento de padrões ao estudo de avaliação de qualidade.

Abstract

Every day millions of images are obtained, processed, compressed, saved, transmitted and reproduced. All these operations can cause distortions that affect their quality. The quality of these images should be measured subjectively. However, that brings the disadvantage of achieving a considerable number of tests with individuals requested to provide a statistical analysis of an image's perceptual quality. Several objective metrics have been developed, that try to model the human perception of quality. However, in most applications the representation of human quality perception given by these metrics is far from the desired representation. Therefore, this work proposes the usage of machine learning models that allow for a better approximation. In this work, definitions for image and quality are given and some of the difficulties of the study of image quality are mentioned. Moreover, three metrics are initially explained. One uses the image's original quality has a reference (SSIM) while the other two are no reference (BRISQUE and QAC). A comparison is made, showing a large discrepancy of values between the two kinds of metrics.
The database that is used for the tests is TID2013. This database was chosen due to its dimension and by the fact of considering a large number of distortions. A study of each type of distortion in this database is made.
Furthermore, some concepts of machine learning are introduced along with algorithms relevant in the context of this dissertation, notably, K-means, KNN and SVM. Description aggregator algorithms like "bag of words" and "fisher-vectors" are also mentioned.
This dissertation studies a new model that combines machine learning and a quality metric for quality estimation. This model is based on the division of images in cells, where a specific metric is computed. With this division, it is possible to obtain local quality descriptors that will be aggregated using "bag of words". A SVM with an RBF kernel is trained and tested on the same database and the results of the model are evaluated using cross-validation.
The results are analysed using Pearson, Spearman and Kendall correlations and the RMSE to evaluate the representation of the model when compared with the subjective results. The model improves the results of the metric that was used and shows a new path to apply machine learning for quality evaluation.

Keywords

Image Quality Assessment, Machine Learning, Image Description, Bag of Words, SVM, SSIM

Contents

List of Figures

List of Tables

Lista de Acrónimos

ADCTC Advanced Discrete Cosine Transform-Based Image Coder

BIQA Blind Image Quality Assessment

BIQI Blind Image Quality Index

BOW Bag Of Words

BPNN Back Propagation feed-forward Neural Network

BRISQUE Blind/Referenceless Image Spatial QUality Evaluator

CBP Circular Back-Propagation

CSF Contrast Sensitivity Function

CWSSIM Complex Wavelet Structural SIMilarity index

DCT Discrete Cosine Transform

FFNN Feed-Forward Neural Network

FR Full Reference

FSIM Feature SIMilarity

HDTV High Definition Television

HVS Human Visual System

IEEE Institute of Electrical and Electronics Engineers

IPTV Internet Protocol Television

IQA Image Quality Assessment

MSCN Mean Subtracted Contrast Normalized

MOS Mean Opinion Score

MSE Mean Squared Error

KNN K-Nearest Neighbors

MATLAB MATrix LABoratory

NR No Reference

PSNR Peak Signal-to-Noise Ratio

QAC Quality Aware Clustering

QoE Quality of Service

QoMEX Quality Of Multimedia EXperience

RBF Radial-Basis Function

RMSE Root Mean Squared Error

RR Reduced Reference

SSIM Structural SIMilarity index

SVM Support Vector Machine

WSSIM Wavelet based Structural SIMilarity index

Chapter 1

Introduction

According to the dictionary[1] an image is a representation of a physical likeness or representation of a person, animal or thing. Digital images are the result of the "acquisition, processing, compression, storage, transmission and reproduction" [2] of that representation. The representations undergo all these forms of processing before they are ultimately displayed to a consumer. Every one of these processing steps alters the appearance of an image resulting in the need to assess impact on the final visual quality. Quality can be defined as "the perception, reflection about the perception and the description of an individual's comparison and judgment process" [3]. There is, however, some misunderstanding and confusion of this term with "Beauty".

"Beauty is in the eye of the beholder". This famous phrase first appeared in the 3rd century BC in Greece. Famous writers like John Lyly, William Shakespeare, Benjamin Franklin and David Hume with slight changes have used it several times since. This old phrase couldn't be more current and correct today. Its literal meaning is that the perception of beauty is subjective. However beauty is not quality. An individual could take a picture of a beautiful landscape with a camera of questionable performance and obtain a sub-par outcome. That photograph of something arguably pleasant to the human eye might have poor quality. The opposite could equally happen.

Knowing this, "can Image Quality be usefully quantified?" This line is the title of the first chapter of a book by Brian Keelan [4]. It summarizes the typical researchers' struggle for the quality subjective prediction. That perception and interpretation of image quality can easily change from any person to another. Trying to predict the quality the average individual will evaluate an image on is an arduous task. Every one of the processing steps mentioned earlier alter the appearance of an image resulting in a need to assess the impact of processing on final visual quality.

[1] http://www.dictionary.com/

Chapter 2

Objectives and Scope

These days, with the high amount of handheld devices, social media and other electronic devices that display visual information it is of the most importance that the end-user has a satisfactory Quality of Service (QoE). Image quality, has perceived by humans, can be measured in a subjective way through subjective tests. These tests have numerous disadvantages, namely long time sessions and the number of ratings per person required to achieve a reliable result. To avoid these disadvantages, objective metrics have been developed to simulate human behavior. These models objectively compute image quality and are later correlated to evaluate their performance. This dissertation addresses the need to use the knowledge about the human perceived quality. The goal of this work is to improve the typical values provided by metrics using the application of machine learning to image quality evaluation. A new technique is proposed based on the division of images into several cells where the mean of the SSIM metric is computed. A sliding window over a grid of cells that divide the image will define a set of image descriptors that are aggregated using a bag of words.
In Chapter 1 definitions for image and quality were given.
Chapter 3 considers why Image Quality Assessment (IQA) is important and gives a definition for "image metric" and how they are classified. A general framework for IQA systems is given. A classifications for Full Reference (FR) image metrics, are defined. Three metrics, SSIM, QAC and BRISQUE, are explained and a comparison between them is made. The image quality database that will be used is analyzed. Each distortion of the database is explained.
Chapter 4, introduces machine learning methods in the context of this dissertation and describes briefly algorithms like K-means clustering, K-Nearest Neighbors (KNN), Support Vector Machine (SVM)s and BOW.
The proposed approach will be described in the following chapter 5 along with the testing results using SSIM, and an analysis of the parameters influence.
Finally some conclusions and future research directions will be discussed in chapter 6.
The main conclusions of this work were published in the 2016 Eighth International Conference on Quality Of Multimedia EXperience (QoMEX) [5].

Chapter 3

Image Quality

3.1 The pivotal necessity of Image Quality Assessment

IQA has, generally speaking, three kinds of applications [6] namely, to monitor image quality control systems, benchmarking image processing systems and algorithms and it's embedding on image processing systems to optimize the algorithms and parameter settings.
An example of the first application could be an image and video acquisition system using a quality metric to monitor and automatically adjust itself to get the best image and video data quality possible. Another possible example would be a network video server. The digital video transmitted could have its quality examined and control the video streaming.
A second case example can be the continuous evaluation of several image processing systems, that require a certain image/video quality for a specific task.
Finally the third application can be exemplified by a visual communication system that uses a quality metric to improve the design of the pre-filtering and bit assignment algorithms at the encoder, and the post processing algorithms at the decoder.
With the proliferation of network handheld devices which can "capture, store, compress, send and display a variety of audiovisual stimuli like High Definition Television (HDTV), Internet Protocol Television (IPTV) and websites such as Youtube and Facebook, an enormous amount of visual data is making its way to consumers" [7]. Due to this, there has been a considerable effort to ensure that the end users will be presented with a satisfactory QoE. According to QUALINET's White Paper [3], "Quality of Experience is the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and or enjoyment of the application or service in the light of the user's personality and current state". Since "the human eyes are the ultimate receivers in most image processing environments" [8], a subjective quality measurement Mean Opinion Score (MOS) of the system is the best indicator of how distortions affect perceived quality. The big disadvantage of doing this is the impracticality and time consumption needed to pool a large number of people to evaluate one or more images. Hence, models that somehow compute objectively the image quality and simulate the human behavior are required. Nevertheless, subjective assessment of visual quality is still used in all Image Quality databases to correlate with image quality metrics to analise it's performances.

3.2 Definition of image quality metric

Image quality assessment aims to "use computational models (metrics) to measure the image quality consistently with subjective evaluations" [9].
Objective image quality metrics can be classified in three ways:

Full Reference - There is the assumption that an entire reference image is known, one of the inputs is a pristine reference image with respect to which the quality of the distorted image is assessed;

No reference or "blind" quality assessment - No reference image is available. The only information that the algorithm receive before making a prediction is the distorted image whose quality is being assessed;

Reduced reference - When there is only a partial sample of the reference image or when the algorithm possesses some information regarding the reference image, but not the actual reference image itself.

One of the first metrics was the Mean Squared Error (MSE), a full reference quality metric that consists in calculating the mean of the square root of the luminosity difference between the pixels of the distorted image and the pixels of the reference image. The MSE allows a comparison between the "true" pixel values for the original image and the noisy image representing the average of the squares of the "errors".

$$\text{MSE} = \frac{1}{mn} \sum_{0}^{m-1} \sum_{0}^{n-1} \|f(i,j) - g(i,j)\|^2 \quad (3.1)$$

Peak Signal-to-Noise Ratio (PSNR) is an expression for the ratio between the maximum possible value(power) of a signal and the power of distorting noise that affects the quality of its representation.

$$\text{PSNR} = 20log_{10}(\frac{MAX_f}{\sqrt{MSE}}) \quad (3.2)$$

It is usually expressed in terms of the logarithmic decibel scale due to the wide dynamic range (ratio between the largest and smallest possible values of a changeable quantity). The higher the PSNR of an image, the more quality the degraded image has in comparison to the original.

Common models like MSE and PSNR did not provide satisfying results when correlated with subjective results and the image quality researchers were forced to find other mechanisms that would enhance the outcome when compared to its subjective equivalents [10].

Modern IQA algorithms estimate quality using a variety of image analysis techniques. When a reference image is available, local differences between the reference and distorted images are measured in various domains, and such differences are mapped for quality estimation.

3.3 Image Quality Assessment based on Error Sensitivity

Computational models for IQA have been developed by exploring effective features that are consistent with the characteristics of a Human Visual System (HVS) for visual quality perception. An image signal that is being subjected to an image metric can be considered as the sum between an undistorted image signal and an error signal [1]. IQA models are necessary because two distorted images that have the same MSE can have distinct errors that can be more or less visible than others. This is due to the fact that the MSE can only quantify the intensity of the error signal [1]. We can conclude from this that a perceptual loss of image quality isn't at all related to the visibility of the error signal. Most IQA approaches are related with error sensitivity aspects like visibility.

3.3.1 Framework

In the image 3.1, adapted from [1], the steps of an IQA system based on error sensitivity can be seen.

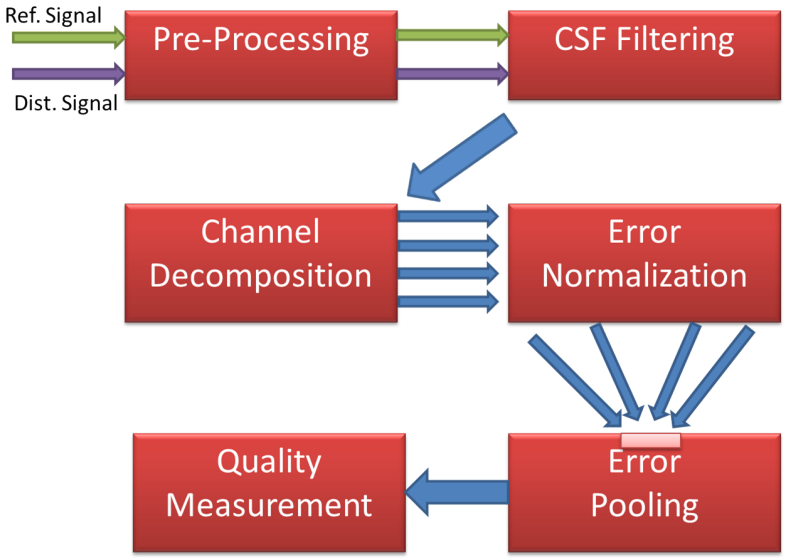


Figure 3.1: Prototype of an Image Quality Assessment System based on Error Sensitivity [1]

The system is comprised of Pre-Processing, Contrast Sensitivity Function (CSF) Filtering, Channel Decomposition, Error Normalization, Error Pooling and Quality measurement [1]. The CSF feature may be implemented separately, as displayed in the scheme, or in the Error Normalization stage.

3.3.2 Pre-Processing

This phase consists in basic operations devised to eliminate known distortions from the images that will be processed.
Firstly, if the images to be compared don't have the same size, they are scaled and aligned.
The used signals will, also, be beforehand converted from the RGB colour values to $YCbCr$ colour space. This transformation is done because it is perceived that $YCbCr$ is more appropriate for comparisons related with the HVS. RGB is an additive colour model in which red, green and blue light are added together to reproduce a broad array of colours. RGB signal are not efficient as a representation for storage and transmission, since they have a lot of redundancy. $YCbCr$ is a practical approximation to colour processing and perceptual uniformity, where the primary colours (red, green and blue) are processed into perceptually meaningful information. $YCbCr$ is used to separate out the luminance component (Y) and two Chroma components (Cb and Cr). With this conversion the Y channel is typically used as the metric input, and the Chroma components are in many cases just ignored.
The quality assessment metrics may need to convert the values of the digital pixels, stored in the computer memory in luminance values on the display device, using a linear point-wise transformation.
A low-pass filter simulating a point-spread function can be applied.
Finally, both images can be modified using a nonlinear point operation to simulate light adaptation.

3.3.3 Constrast Sensitivity Filtering

The CSF measures the threshold sensitivity of the HVS over a wide range of different spatial and temporal frequencies [11]. Some image quality metrics include a stage that analyses the signal according to this function, playing a central role in image processing techniques like compression. It is normally performed by applying a linear filter that estimates the response frequency of the CSF [1]. Nonetheless, other metrics also apply CSF after the channel decomposition as a base sensitivity normalization factor.

3.3.4 Channel Decomposition

In this stage images are divided into channels depending on their spatial frequencies, temporal frequencies or their orientation [1]. Whilst some methods of quality assessment implement sophisticated channel decomposition, others just apply a Discrete Cosine Transform (DCT) or separate the signal bands using wavelet transforms.

3.3.5 Error Normalization

This stage computes the difference between the reference and the distorted for each channel that was decomposed in the previous stage, followed by their normalization using a mask. If two or more image components have similar spatial frequencies, temporal frequencies or orientations, this process will decrease their visibility [1]. The difference is calculated by subtracting the intensity of the reference and the distorted coefficients of the same channel within a spatial neighborhood. Some methods also consider the effect of contrast response saturation.

3.3.6 Error Pooling

This final step is where a single value, that classifies the image quality objectively is obtained. All the different channels previously obtained, along with the normalized error signals, are pooled together. On most metrics, the pooling takes the shape of a Minkowsky norm [1],

$$E(\{e_{l,k}\}) = \left(\sum_{l} \sum_{k} |e_{l,k}|^{\beta} \right)^{1/\beta} \qquad (3.3)$$

where $e_{l,k}$ is the normalized error of the K-th coefficient of the l-th channel and β is a exponential constant typically chosen between 1 and 4. Minkowsky pooling can be done is space(index k) and after in frequency (index l) or vice-versa, with some non-linearity between them, or possibly with different β. A spatial map can indicate the relative importance of different regions and what can be used to obtain variant spatial weighting.

3.3.7 Evaluation of the Objective Models

To evaluate the performance of a metric, the Pearson correlation coefficient, Spearman rank order correlation, Kendall rank order correlation and the RMSE, between the subjective MOS and the quality estimation, are used.

3.3.7.1 Regression Model

Before using the measures, a logistic function is used as regression model of the estimated quality to fit the data,

$$\text{MOS}_p = b1 + \frac{b2}{1 + e^{(-b3(MR-b4))}} \quad (3.4)$$

The values of the MOS associated an image are normalized according to equation 3.5 where MR is the result of our methodology and $b1$, $b2$, $b3$ and $b4$ denote the regression parameters, initialized with MOS_{min}, MOS_{max}, MR_{min} and MR_{max} respectively,

$$\text{MOS}_n(i) = 5 \times \frac{\text{MOS}(i) - \text{MOS}_{min}}{\text{MOS}_{max} - \text{MOS}_{min}} \quad (3.5)$$

3.3.7.2 Pearson Linear Correlation

The Pearson linear correlation coefficients r_P measures the model prediction accuracy.

$$r_P = \frac{\sum_{j=1}^{N}(S_j - \bar{S})(O_j - \bar{O})}{\sqrt{\sum_{j=1}^{N}(S_j - \bar{S})^2(\sum_{j=1}^{N} O_j - \bar{O})^2}} \quad (3.6)$$

In this case, S_j denotes the subjective score MOS_n and O_j the objective MOS_p. $\bar{S}$ and $\bar{O}$ are the means of the respective data sets and N represents the total number of image samples considered in the analysis.

3.3.7.3 Spearman Rank-Order Correlation

The Spearman rank-order correlation coefficient r_S evaluates the model prediction monotonicity.

$$r_S = 1 - (\frac{6\sum_{j=1}^{N} d_j^2}{N(N^2-1)}) \quad (3.7)$$

where d is the difference between the ranked MOS_n and the ranked MOS_p. In practice, this is computed as the Pearson correlation, but instead of the variables the ranked variables are used. N represents the total number of image samples considered in the analysis.

3.3.7.4 Kendall Rank-Order Correlation

Kendall rank correlation r_K is a non-parametric test that measures the strength of dependence between two variables [12].

$$r_K = \frac{N_c - N_d}{\frac{1}{2}N(N-1)} \quad (3.8)$$

where N_c is the number of concordant pairs [12], N_d is the number of discordant pairs and N represents the total number of image samples.

3.3.7.5 Root Mean Square Error

RMSE measures the quality prediction error that is maximized when the RMSE is minimized.

$$RMSE = \sqrt{\frac{\sum_{j=1}^{N}(S_j - O_J)^2}{N}} \quad (3.9)$$

where N represents the total number of image samples considered in the analysis.

3.4 Image Metrics

Numerous objective methods for image evaluation have been introduced over the years from the first ones like MSE and PSNR (previously mentioned) to the more recent metrics. These metrics, as it has been previously noted, can be FR, No Reference (NR) or Reduced Reference (RR). The FR quality assessments can be divided in different categories: difference measures and statistical-oriented metrics, structural similarity measures, visual information metrics, information weighted metrics, HVS-inspired metrics and colour difference metrics.

Difference Measures and Statistical-Oriented Metrics - These metrics are based on pixel values differences and provide measures between the reference and the distorted image. Two examples of this category are the MSE and the PSNR.

Structural Similarity Measures- These metrics model the quality based on pixel statistics to model the luminance (using the mean), the contrast (variance), and the structure (cross-correlation).

Visual Information Measures - These metrics aim at measuring the image information by modeling the psycho-visual features of the HVS or by measuring the information fidelity. Then, the models are applied to the reference and distorted image, resulting in a measure of the difference between them.

Information Weighted Metrics - The metrics in this category are based on the modeling of relative local importance of the image information. As not all regions of the image have the same importance in the perception of distortion, the image differences computed by any metrics have allocated local weights resulting in a more perceptual measure of quality.

HVS Inspired Metrics - These metrics try to model empirically the human perception of images from real scenes.

Colour Difference Measures - The colour difference metrics were developed because the CIE1976 colour difference magnitude in different regions of the colour space did not appear correlated with perceived colours.

One of the most impactful FR quality metric SSIM. It is a method that improved upon earlier metrics such as PSNR and MSE. The 2004 paper in which it is described[1] is one of the most cited papers in image processing and was even awarded the Institute of Electrical and Electronics Engineers (IEEE) Signal Processing Society Best Paper Award[1]. This metric will be used later in the algorithm proposed in this dissertation.

[1] http://signalprocessingsociety.org/uploads/awards/Best_Paper.pdf

Other metrics, this time RR, are QAC, that uses unsupervised Machine Learning (K-Means), and BRISQUE, that uses supervised Machine Learning (SVM).
In the following sections are described the SSIM metric used in this work and QAC and BRISQUE as they are metrics that reveal between the most powerful NR metrics based on Machine Learning.

3.4.1 Structural Similarity Index Metric

The SSIM [1] is an objective metric based on the degradation of structural similarity. SSIM is based on that assumption that the HVS extracts structural information from the analyzed image textures. The SSIM index incorporates three representative features for luminance, contrast and structural information that are extracted by the average pixel intensity values, the standard deviation of the local image regions and the cross correlation values between two local image regions and the cross correlation values between two local image regions, respectively. Supposing x and y are two non-negative image signals, with one being the reference image and the other the distorted image, the luminance of each signal is compared:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (3.10)$$

The luminance comparison function $l(x, y)$ is a function of μ_x and μ_y. The signal contrast is calculated as described earlier:

$$\sigma_x = (\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2)^{1/2} \quad (3.11)$$

The contrast comparison $c(x, y)$ is then the comparison of σ_x and σ_y. The signal comparison is then normalized(divided) by its own standard deviation. The structure comparison is conducted on these normalized signals $(x - \mu_x)\sigma_x$ and $(y - \mu_y)\sigma_y$.
The three components are combined resulting is an overall similarity measure:

$$S(x, y) = f(l(x, y), c(x, y), s(x, y)) \quad (3.12)$$

These three components are relatively independent from each other.
For luminance comparison:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (3.13)$$

where the constant C_1 is included to avoid instability when $\mu_x^2 + \mu_y^2$ is approximately zero. C_1 is defined as

$$C_1 = (K_1 * L)^2 \quad (3.14)$$

where L is the dynamic range of the pixel values (255 for 8-bit grayscale images) and $K_1 << 1$ is a small constant. Weber's law says that the magnitude of a just-noticeable luminance change δ I is approximately proportionate to the background luminance I for a wide range of luminance values. This means that the HVS is sensitive to relative luminance changes and it is not to absolute luminance change. Applying Weber's law on the luminance equation,we can rewrite μ_y as $(1 + R) * \mu_x$, where R represents the size of luminance change relative to backgroung

luminance. Substituting in Eq. :

$$l(x,y) = \frac{2(1+R)}{1+(1+R)^2+C_1/\mu_x^2} \quad (3.15)$$

The contrast function is similar to the luminance function:

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2+\sigma_y^2+C_2} \quad (3.16)$$

where $C_2 = (K_2L)^2 and K2 << 1$. The structural comparison is done after luminance subtraction and variance normalization.

$$s(x,y) = \frac{\sigma_{xy}+C3}{\sigma_x\sigma_y+C3} \quad (3.17)$$

In the discrete form, σ_{xy} can be estimated as:

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y) \quad (3.18)$$

Combining the three equations:

$$SSIM(x,y) = [l(x,y)]^\alpha \times [c(x,y)]^\beta \times [s(x,y)]^\gamma \quad (3.19)$$

where $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ are parameters that adjust the relative importance of each component. To simplify the expression $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$ resulting in a specific form of the SSIM index:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (3.20)$$

3.4.2 Quality Aware Clustering

QAC [13] is a RR metric for general purpose Blind Image Quality Assessment (BIQA). The metric is built upon a set of some reference and distorted images (but without human score), divide the distorted images into overlapped cells and use a percentile pooling strategy to estimate the local quality level of each patch. Then the QAC method is proposed to learn a set of centroids on each quality level. These centroids are to obtain the quality of each cell in a given image, using a nearest neighbor algorithm, and subsequently a perceptual quality score of the whole image can be obtained.

3.4.2.1 Learning of Quality-Aware Centroids by QAC

First, there is a Random selection of 10 source images from the Berkeley Image Database [14] due to them having different scenes from the images in the databases. Each image is distorted to obtain the four of the most common types of distortions: Gaussian noise, Gaussian blur, JPEG compression and JPEG2000 compression. There are three quality levels for each of the four distortions to make sure that the quality distribution of the resulted samples that will be obtained is balanced.
The distortion level is controlled as in the following, standard deviation for Gaussian noise, blur kernel for Gaussian Blur, compression ratio for JPEG and JPEG2000.

To control each level of distortion for each of them,

A dataset of 120 distorted images is obtained from the 10 reference images. Each reference and distorted image is divided into overlapped cells.A cell of one reference image will be denoted by x_i and the corresponding distorted cell will be denoted by d_i. To assign a perceptual quality to d_i, a FR IQA similarity function such as SSIM [1] previously mentioned or Feature SIMilarity (FSIM) [9] to calculate the similarity between x_i and d_i. This way, there will be no dependency of human scores. The FSIM's formula is

$$s_i = S(x_i, d_i) = \frac{2PC(x_i)PC(d_i) + t_1}{PC(x_i)^2 + PC(d_i)^2 + t1} \times \frac{2G(x_i)G(d_i) + t_2}{G(x_i)^2 + G(d_i)^2 + t_2} \quad (3.21)$$

where $PC(x_i)$ and $G(x_i)$ refer to the phase congruency and gradient magnitude at the center of xi, respectively, and t_1 and t_2 are positive constants for numerical stability.

Each cell(d_y) will have a similarity score between 0 and 1. The s_i's are normalized using a percentile pooling procedure [13] to obtain an average close to the overall perceptual quality. In particular, s_i is divided by a constant C to define an average quality of all patches in an image will equal to the percentile pooling result. Denoting with Ω as the set of patch indices of an image, and by Ω_p the set of indices of the 10% percentile results with the lowest quality patches. The normalization factor C is calculated as:

$$C = \frac{\sum i\Omega s_i}{10 \sum i\Omega_p s_i} \quad (3.22)$$

After that, each s_i is normalized as: $c_i = s_i/C$.

With the cell quality normalization strategy, we acquire a set of cells d_i and their normalized scores c_i, on which the quality-aware clustering can be executed. Using c_i, d_i can be grouped into groups of similar quality obtaining different clusters based on local structures. Because c_i is a real-value number between 0 and 1, firstly c_i is uniformly quantized into L levels, denoted by $q_l = l/L, l = 1, 2, ..., L$. The cells that have the same quality are then grouped into the same group, denote by G_l.

$$G_l = \begin{cases} d_i | c_i \leqslant q_l, & \text{for } l = 1 \\ d_i | q_{l-1} < c_i < q_l, & \text{for } l = 2, ..., L \end{cases} \quad (3.23)$$

The clustering is then applied to each group G_l. To enhance the clustering accuracy, the QAC within each G_l should be based on some structural feature of d_i. The following high pass filter to extract the feature of cell d_i:

$$h_\sigma(r) = 1_{r=0} - \frac{1}{\sqrt{2\pi\sigma}} exp(-\frac{r^2}{2\sigma^2}) \quad (3.24)$$

where σ is the scale parameter to control the shape of the filter. By convolving h_σ with the image, the image's detailed structures will be enhanced. Three h_σ are used on different scales ($\sigma = 0.5, 2.0, 4.0$) to extract the feature of d_i. The filtering outputs of d_i on the three scales are concatenated into a feature vector, denoted by f_i. The acQAC of $d_i \varepsilon G_l$ is then performed by applying the K-mean clustering algorithm to f_i:

$$min_{m_{l,k}} \sum_{k=1}^{K} \sum_{d_i \varepsilon G_{l,k}} \|f_i - m_{l,k}\|^2 \quad (3.25)$$

where $G_{l,k}$ is the k^{th} cluster in Group G_l. The Euclidean distance is used as the similarity metric due to complexity costs.

3.4.2.2 Performing Blind Quality Estimation

To perform blind quality pooling, using the learned quality-aware centroids $m_{l,k}$, one must go through the following stages: patch partition and feature extraction, cluster assignment on multiple levels, patch quality estimation and final pooling with all patches' quality.

Patch partition and feature extraction- Each test image is partitioned into N overlapped patches y_i and use high pass filters h_σ to extract the feature vector, denoted by f_i^y, of each y_i, $i = 1, ..., N$.

Cluster assignment- Find the nearest centroid to the feature vector f_i^y of patch y_i by assuming that patches which have similar structural features will have similar visual quality.

Patch quality estimation- The distance between f_i^y and m_{l,k_i} is $\delta_{l,i} = \|f_i^y - m_{l,k}\|^2$. The shorter the distance $\sigma_{l,k}$ is, the more likely patch y_i should have the same quality level as that of centroid m_{l,k_i}. Therefore, we can use the following weighted average rule to determine the final quality score of y_i:

$$z_i = \frac{\sum_{l=1}^{L} q_l exp(-\delta_{l,i}/\lambda)}{\sum_{l=1}^{L} exp(-\delta_{l,i}/\lambda)} \quad (3.26)$$

where λ is a parameter to control the decay rate of weight $exp(-\delta_{l,i}/\lambda)$ with reference to distance $\delta_{l,i}$.

Final pooling Using the estimated quality z_i of all patches y_i, the final single quality score, denoted by z of the test image, y is obtained. The pooling is done using the following equation:

$$z = \frac{1}{N} \sum_{i=1}^{N} z_i \quad (3.27)$$

3.4.3 Blind/Referenceless Image Spatial QUality Evaluator

BRISQUE [7] is also a RR metric.
The metric starts by computing normalized luminances in a distorted image using local mean subtraction and divisive normalization.

$$(i, j) = (I(i, j) - \mu(i, j))/(\sigma(i, j) + C) \quad (3.28)$$

where, $i\varepsilon 1, 2...M$, $j\varepsilon 1, 2...N$ are spatial indices, M, N are the image height and width respectively, $C = 1$ is a constant that prevents divisions by 0 and

$$\mu(i,j) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} \omega_{k,l}(i,j) \quad (3.29)$$

and

$$\sigma(i,j) = \sqrt{\sum_{k=-K}^{K} \sum_{l=-L}^{L} \omega_{k,l}(I_{k,l}(i,j) - \mu(i,j))^2} \quad (3.30)$$

where $\omega_{k,l|k=-K,\ldots,K,l=-L,\ldots,L}$ is a $2D$ circularly-symmetric Gaussian weighting function sampled out to 3 standard deviations and rescaled to unit volume. K and L are constants equal to 3. The transformed luminances are refered as Mean Subtracted Contrast Normalized (MSCN) coefficients.

Also, the statistical relationships between neighboring pixels are modeled according to four orientation (horizontal (H), vertical (V), main-diagonal ($D1$) and secondary diagonal($D2$)).

$$H(i,j) = (i,j)(i,j+1) \quad (3.31)$$
$$V(i,j) = (i,j)(i+1,j) \quad (3.32)$$
$$D1(i,j) = (i,j)(i+1,j+1) \quad (3.33)$$
$$D2(i,j) = (i,j)(i+1,j-1) \quad (3.34)$$

for $i \,\varepsilon\, 1,2...M$ and $j \,\varepsilon\, 1,2...N$

For each of the four orientations, four parameters are computed (shape, mean, left variance and right variance) using a GGD [7] obtaining 16 values which will be part of the feature.

The feature has now 18 values. The original distorted image is filtered using a low-pass filter and is downscaled by a factor of 2. Another 18 values are extracted. The final BRISQUE feature has 36 values.

The metric uses a SVM with a Radial-Basis Function (RBF) kernel that was previously trained using the LIVE database to obtain a quality evaluation. The input for each image is the feature previously calculated.

3.4.4 Metrics Comparison

The three detailed metrics of previous section, SSIM, QAC and BRISQUE, were compared to evaluate the path to follow in this dissertation. The Pearson, Spearman, Kendall correlations, and also the RMSE between this three metrics and the MOS given in figures 3.2, 3.3, 3.4 and 3.5, reveal that the NR metrics are still far from providing an appropriate representation. The SSIM reveal always an improved estimation, apart the case of the simple and colour distortions classes. For this reason, during this work these metrics have been not used, and only SSIM was considered. Of course, other FR metrics could be used but from our preliminary tests no different conclusions should be defined.

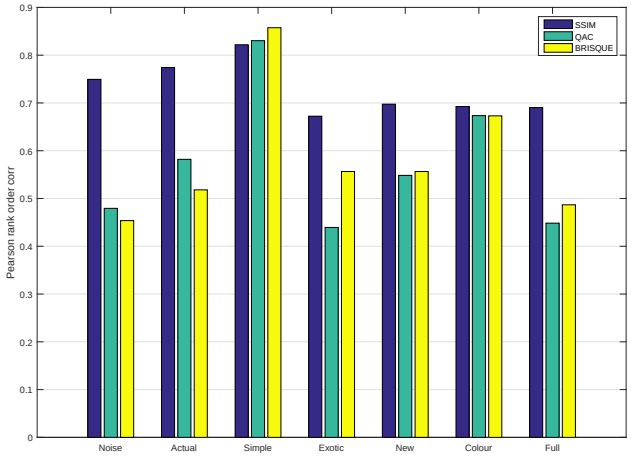


Figure 3.2: Comparison of SSIM, QAC and BRISQUE using the Pearson correlation coefficient

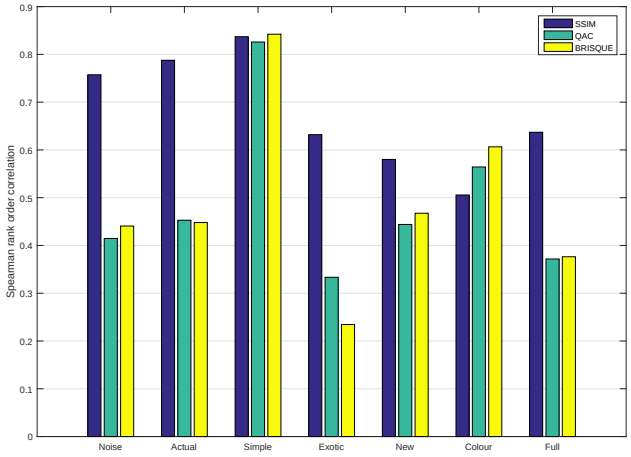


Figure 3.3: Comparison of SSIM, QAC and BRISQUE using the Spearman rank order Correlation

3.5 TID2013 Database

The database used for training and testing is the TID2013 [2]. This database contains images with 24 different distortion types. Each type has five levels of distortion. For all types of distortions the corresponding levels of PSNR are of about $33dB$, $30dB$, $27dB$, $24dB$ and $21dB$ (Lv1, Lv2, Lv3, Lv4 and Lv5). According to the creators of the database, this number of distortions for the 25 reference images is enough to reliably cover the full range of subjective quality. There were made subjective experiences in five countries to obtain a MOS. All the distorted images were obtained from the Kodak database[2] with the exception of one image (synthetic) that was artificially created.Each reference image originates 120 distorted images (five levels for each of twenty four types of distortions). The images have a fixed dimension of 384×512 pixels for unification purposes. The distortion types present in the database are Additive White Gaussian Noise

[2] r0k.us/graphics/kodak/

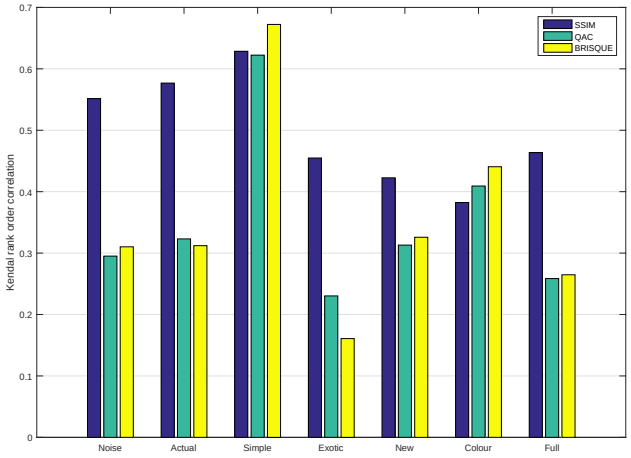


Figure 3.4: Comparison of SSIM, QAC and BRISQUE using the Kendall rank order Correlation

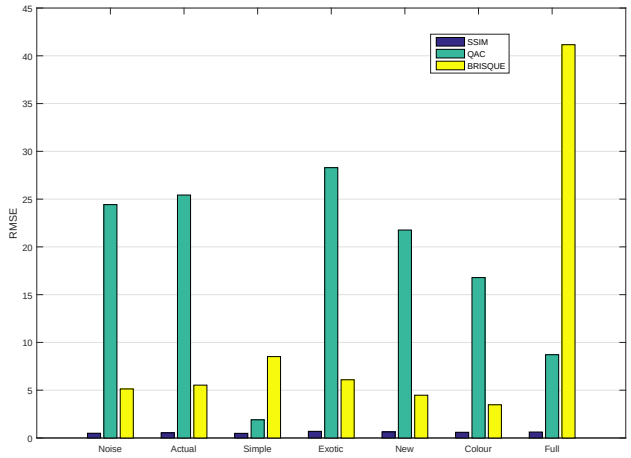


Figure 3.5: Comparison of SSIM, QAC and BRISQUE using the RMSE

applied to the luminance component (#1), Additive White Gaussian Noise applied to the colour components (#2), Spatially Correlated Additive Gaussian Noise (#3), Masked Noise (#4), High Frequency Noise (#5), Impulse Noise (#6), Quantization Noise (#7), Gaussian Blur (#8), Image Denoising (#9), JPEG Lossy Compression (#10), JPEG2000 Lossy Compression (#11), JPEG Transmission Errors (#12), JPEG2000 Transmission Errors (#13), Non-Eccentricity Pattern Noise (#14), Local Block-Wise Distortions of Different Intensity (#15), Mean Shift (#16), Contrast Change (#17), Change of Colour Saturation (#18), Multiplicative Gaussian Noise (#19), Comfort Noise (#20), Lossy Compression of Noisy Images (#21), Image Colour Quantization with Dither (#22), Chromatic Aberrations (#23), Sparse Sampling and Reconstruction (#24). The distortion types were divided in subsets, to allow an improved analysis of the data.

Table 3.1: Distortion subsets

No.	Type of Distortion	Noise	Actual	Simple	Exotic	New	Colour	Full
1	Additive White Gaussian Noise	+	+	+	-	-	-	+
2	Noise in Colour Comp.	+	-	-	-	-	+	+
3	Spatially Correl. Noise	+	+	-	-	-	-	+
4	Masked Noise	+	+	-	-	-	-	+
5	High Frequency Noise	+	+	-	-	-	-	+
6	Impulse Noise	+	+	-	-	-	-	+
7	Quantization Noise	+	-	-	-	-	+	+
8	Gaussian Blur	+	+	+	-	-	-	+
9	Image Denoising	+	+	-	-	-	-	+
10	JPEG Compression	-	+	+	-	-	+	+
11	JPEG2000 Compression	-	+	-	-	-	-	+
12	JPEG Transm. Error	-	-	-	+	-	-	+
13	JPEG2000 Transm. Errors	-	-	-	+	-	-	+
14	Non Ecc. Patt. Noise	-	-	-	+	-	-	+
15	Local Block-Wise	-	-	-	+	-	-	+
16	Mean Shift	-	-	-	+	-	-	+
17	Contrast Change	-	-	-	+	-	-	+
18	Change of Colour Saturation	-	-	-	-	+	+	+
19	Multiplicative Gaussian Noise	+	+	-	-	+	-	+
20	Comfort Noise	-	-	-	+	+	-	+
21	Lossy Compression Noisy Images	+	+	-	-	+	-	+
22	Image Colour Quantization w/ Dither	-	-	-	-	+	+	+
23	Chromatic Aberrations	-	-	-	+	+	+	+
24	Sparse Sampling and Reconstruction	-	-	-	+	-	-	+

3.6 Types of Image Distortions

This section gives a definition to each distortion, explains how each one is obtained and shows their effect on a specific reference image 3.31.

3.6.1 Gaussian Noise

In general, noise is an unwanted component in an image. Any degradation, such as a random variation of brightness or colour information that occurs in an image, can be called noise [15]. The most common form of noise is the so-called Additive noise, can be defined as:

$$f = g + q \quad (3.35)$$

where f is an image and g and q are the original image and the noise component, respectively. Some less common noises are multiplicative, instead of additive, resulting,

$$f = g \times q \quad (3.36)$$

Multiplicative noise can be changed into an additive model by using the logarithmic function:

$$e^f = e^{g+q} = e^f e^q \quad (3.37)$$


Figure 3.6: Reference Image 1


Figure 3.7: Reference Image 2


Figure 3.8: Reference Image 3


Figure 3.9: Reference Image 4


Figure 3.10: Reference Image 5


Figure 3.11: Reference Image 6


Figure 3.12: Reference Image 7


Figure 3.13: Reference Image 8


Figure 3.14: Reference Image 9


Figure 3.15: Reference Image 10


Figure 3.16: Reference Image 11


Figure 3.17: Reference Image 12


Figure 3.18: Reference Image 13


Figure 3.19: Reference Image 14


Figure 3.20: Reference Image 15


Figure 3.21: Reference Image 16


Figure 3.22: Reference Image 17


Figure 3.23: Reference Image 18


Figure 3.24: Reference Image 19


Figure 3.25: Reference Image 20


Figure 3.26: Reference Image 21


Figure 3.27: Reference Image 22


Figure 3.28: Reference Image 23


Figure 3.29: Reference Image 24

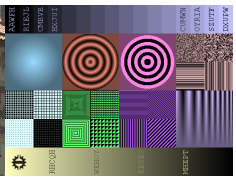


Figure 3.30: Reference Image 25

The opposite can also be done

$$\log(f) = \log(g \times q) = \log(g) + log(q) \quad (3.38)$$

Some noises can be described in an easier way using additive models, while others are better described as multiplicative models. Additive White Gaussian Noise (#1) is the most common occurring noise [15]. This type of noise follows a Gaussian distribution, meaning that it's probability density function given by the normal distribution (or Gaussian distribution).

$$P_q(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} for -\infty < x < \infty \quad (3.39)$$

where μ is the mean and σ is the standard deviation. A higher σ gives a bigger level of distortion.

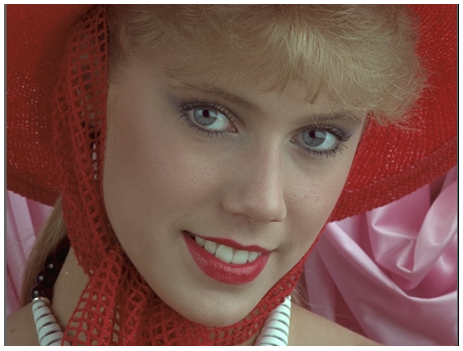


Figure 3.31: Reference Image for each distortion

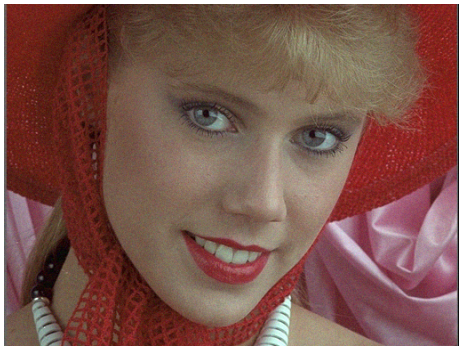


Figure 3.32: Additive Gaussian Noise.
PSNR $= 30dB$.

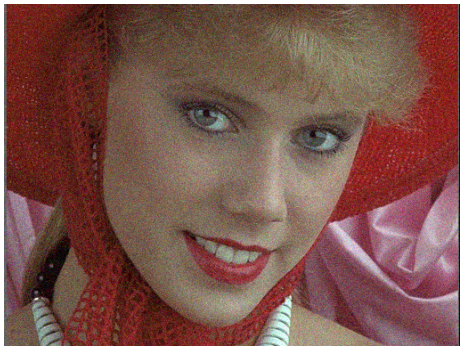


Figure 3.33: Additive Gaussian Noise.
PSNR $= 24dB$.

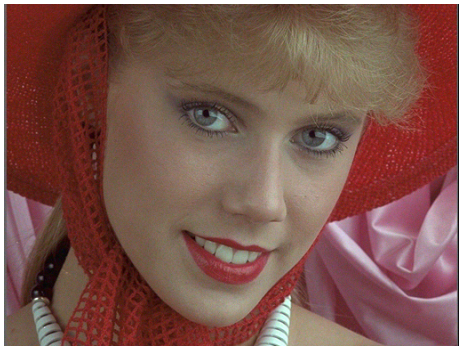


Figure 3.34: Additive White Gaussian noise added in colour components instead of in the luminance component.
PSNR $= 30dB$.

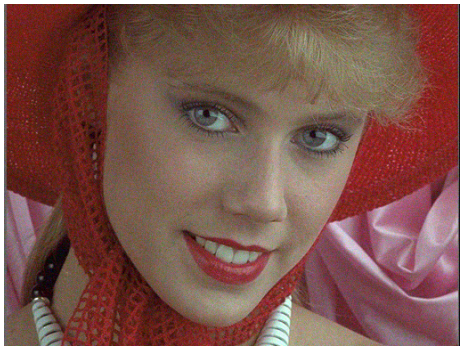


Figure 3.35: Additive White Gaussian noise added in colour components instead of in the luminance component.
PSNR $= 24dB$.

Figures 3.32 and 3.33 and shows how the distortion affects the reference image.
In the second distortion instead of adding the noise to the luminance component Y, the noise is added to the colour components $CbCr$ using a gaussian distribution [16]. An example with two different quality levels can be seen in 3.34 and 3.35. This distortion was added to test if quality metrics perceive brightness (luminance) and colour (chrominance) differently just like the HVS does.
The third distortion is also additive gaussian noise, but this time being a low-pass spatially correlated noise [2]. To obtain this distortion, noise is generated and then filtered using a low-pass filter. The result is added to the reference image.
Multiplicative Gaussian Noise (#19) is Gaussian noise that follows the multiplicative model explained above, instead of the additive one. In this database, before multiplying the Gaussian

Figure 3.36: Additive Gaussian Spatially Correlated Noise. PSNR $= 30dB$.

Figure 3.37: Additive Gaussian Spatially Correlated Noise. PSNR $= 24dB$.

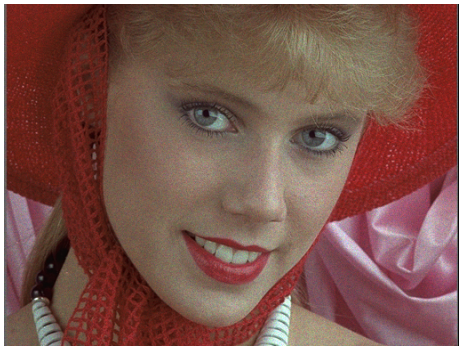


Figure 3.38: Multiplicative Gaussian Noise. PSNR $= 30dB$.

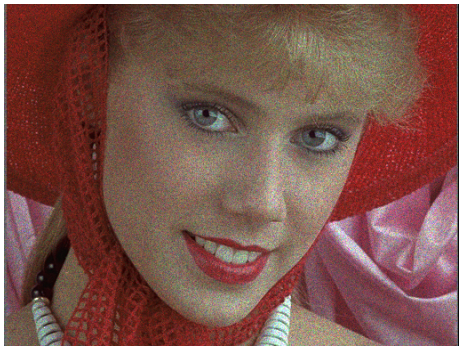


Figure 3.39: Multiplicative Gaussian Noise. PSNR $= 24dB$.

Figure 3.40: Masked noise. PSNR $= 30dB$.

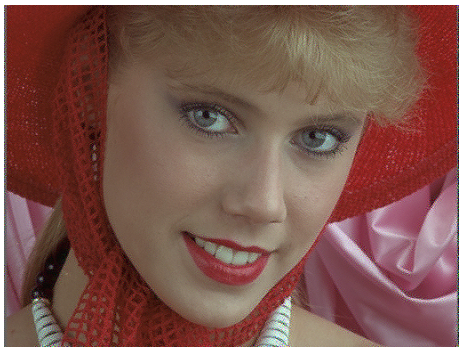


Figure 3.41: Masked noise. PSNR $= 24dB$.

noise with the reference image, the noise was simulated separately for each RGB colour with equal σ^2 (variance) [2]. The results can be observed in 3.38 and 3.39.

3.6.2 High Frequency Noise

Noise fluctuations can vary in spatial frequency[3]. High frequency images have a finer texture, as can be seen in figures 3.42 and 3.43, while a low frequency image has a coarser texture. High Frequency Noise(#5) is related to the spatial frequency of HVS [17]. The distorted image can be obtained by generating white noise followed by a high pass filter.

[3]http://www.cambridgeincolour.com/tutorials/image-noise-2.htm

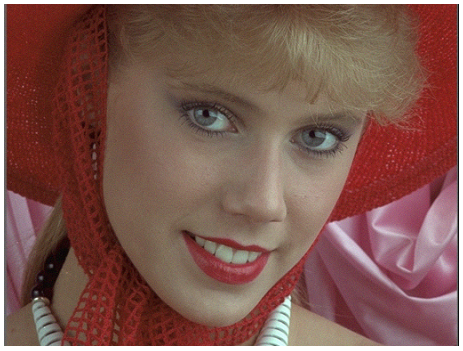


Figure 3.42: High Frequency Noises. PSNR $= 30dB$.

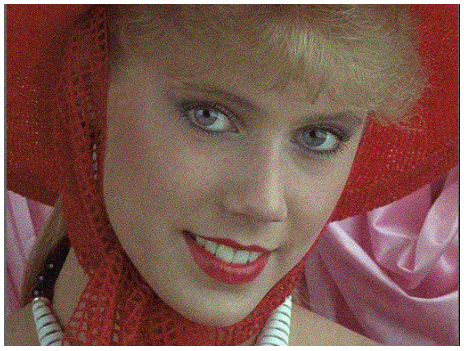


Figure 3.43: High Frequency Noises. PSNR $= 24dB$.

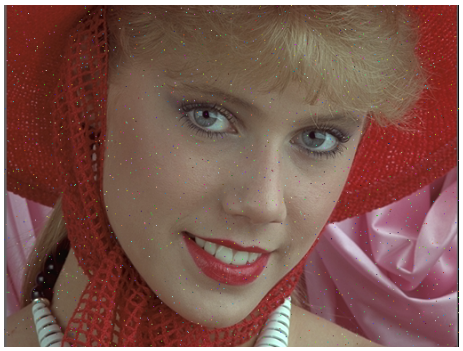


Figure 3.44: Salt-and-Pepper Noise. PSNR $= 30dB$.

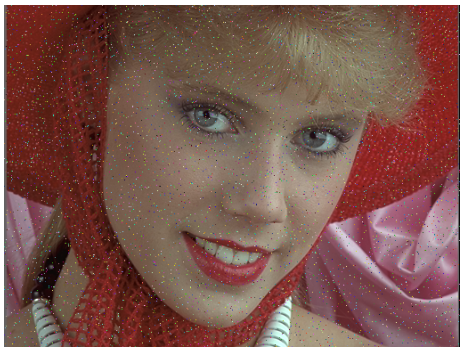


Figure 3.45: Salt-and-Pepper Noise. PSNR $= 24dB$.

3.6.3 Impulse Noise

The #6 distortion is Impulse Noise. Also called salt-and-pepper noise, images with this type of noise have dark pixels in brighter regions and bright pixels in darker regions. These distorted images appear to have black and white "dots" (salt-and-pepper). These outliers can be caused by bit errors in transmission or errors in analog-to-digital conversion [15]. When each pixel is quantized into B bits, it's value X can be written as,

$$X = \sum_{i=0}^{B-1} b_i 2^i \quad (3.40)$$

Assuming a binary symmetric channel with a crossover probability PR equal to ε, flipping each bit with the same probability and defining the received value as Y, the probabilities can be expressed as:

$$Pr(|X - Y| = 2^i) = \varepsilon \text{ for } i = 0, 1, ..., B - 1 \quad (3.41)$$

The pixels with the most changed bits should appear as black or white "dots".

3.6.4 Quantization Noise

Quantization Noise (#7) is caused when quantizing the pixels of a sensed image to a discrete number or number of discrete levels [15]. A continuous image signal is converted into a dis-

Figure 3.46: Quantization Noise. PSNR $= 30dB$.

Figure 3.47: Quantization Noise. PSNR $= 24dB$.

Figure 3.48: Gaussian Blur. PSNR $= 30dB$.

Figure 3.49: Gaussian Blur. PSNR $= 24dB$.

crete digital representation where a range of input values produces the same output producing discrete, stepped digital data resulting in a slight error. This distortion has an approximately uniform distribution and usually occurs in the acquisition process. The image's previous smooth gradations become regions separated by noticeable discontinuities.

3.6.5 Gaussian Blur

Gaussian Blur (#8) is a blur effect that occurs when an image is acquired during a motion/shaking period that smooths the image's sharpness (edges and boundaries) [18]. The visual effect of this blurring technique is a smooth blur similar to seeing the image through a translucent screen. It can be modeled as the result of blurring an image by a Gaussian filter. The MATrix LABoratory (MATLAB) function "imgaussfilt" can perform this transformation by applying the convolution kernel to the image:

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.42)$$

where x and y are the pixel's location and σ is the standard deviation of the Gaussian Blur. It is often used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales. Gaussian smoothing is very frequently used with edge detection.

3.6.6 Image Denoising

The #9th distortion is Image Denoising. There have been some efforts in trying to recover distorted images using denoising algorithms (filters) [19]. The resulting images may still contain

Figure 3.50: Image Denoising.
PSNR $= 30dB$.

Figure 3.51: Image Denoising.
PSNR $= 24dB$.

residual distortions that eventually look perceptually worse as bad quality [16]. The images in this database had Gaussian noise that was filtered using a acDCT [20]. The colour channels of the noisy image are descorrelated. The DCT technique decomposes the image into local cells of same size. The article [21] arques that a 16×16 window leads to the best results. A DCT transform is calculated for each cell, it's coefficients and thresholded (with a threshold equal to 3σ and an inverse DCT tranform is then calculated. Finally, the cells are averaged and aggregated to reconstruct the denoised image.

3.6.7 Distortions in JPEG and JPEG200

Compression is a data transformation followed by an enconding method [22] used to decrease the data file's size. While in lossless image compression, the goal is to represent an image using the least amount of bits without loss of information, in lossy image compression the goal is to achieve a faithful representation of the image using the least amount of bits [15]. It is clear that in lossy image compression there is some loss of information. The advantage of this method though, is the reduction of the image's size and bit-rate. The results of coding and decoding the image with JPEG (#10) and JPEG2000 (#11) with different degrees of compression can be seen in figures 3.52, 3.53, 3.54 and 3.55, respectively.

Moreover, and in the addition to the CODEC processes, transmission networks can add error to the streaming data. Distortions 12 and 13 show respectively the result of adding randomly bit errors to the encoded JPEG and JPEG2000 data stream of the images, resulting in decoding errors. The results can be observed in 3.56 and 3.57 (#12) and 3.58 and 3.59 (#13).

3.6.8 Non-Eccentricity Pattern Noise

Humans have difficulty in perceiving an image's distorted fragments if they appear similar to the original texture or the colour of the surrounding fragments. For this reason, the Non-Eccentricity Pattern Noise distortion (#14) was created and modeled for this database. Blocks of 15x15 pixels were randomly taken from the reference image and copied to locations of another blocks nearby. Without having the reference image to compare, it is not easy identifying this distortion on certain images.

Figure 3.52: JPEG lossy compression. PSNR $= 30dB$.

Figure 3.53: JPEG lossy compression. PSNR $= 24dB$.

Figure 3.54: JPEG2000 lossy compression. PSNR $= 30dB$.

Figure 3.55: JPEG2000 lossy compression. PSNR $= 24dB$.

Figure 3.56: JPEG lossy compression with transmission errors. PSNR $= 30dB$.

Figure 3.57: JPEG lossy compression with transmission errors. PSNR $= 24dB$.

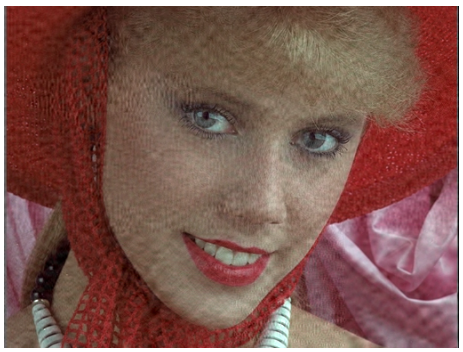


Figure 3.58: JPEG2000 lossy compression with transmission errors. PSNR $= 30dB$.

Figure 3.59: JPEG2000 lossy compression with transmission errors. PSNR $= 24dB$.

Figure 3.60: Non-Eccentricity Pattern Noise. PSNR $= 30dB$.

Figure 3.61: Non-Eccentricity Pattern Noise. PSNR $= 24dB$.

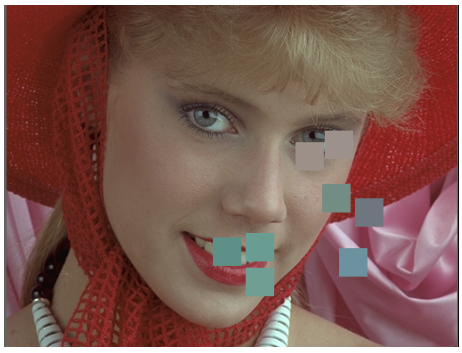


Figure 3.62: Block-Wise Distortions of Different Intensity. PSNR $= 30dB$.

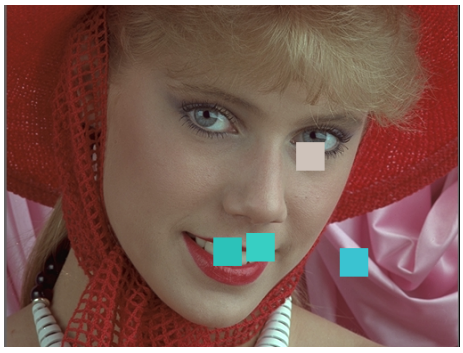


Figure 3.63: Block-Wise Distortions of Different Intensity. PSNR $= 24dB$.

3.6.9 Block-Wise Distortions of Different Intensity

Modeled for this database, "Block-Wise Distortions of Different Intensity" (#15) was based on the supposition that the HVS reacts to an area of pixels and ignores distortions on single pixels [16]. Blocks of 32x32 pixels randomly chosen and with an arbitrary colour are placed in important areas of an image. Depending on the distortion level 2, 4, 6, 8 and 10 blocks were replaced (for levels 1, 2, 3, 4 and 5 respectively) The colours and intensity of the blocks were adjusted to fit the PSNR levels defined earlier for all distortions.

3.6.10 Mean Shift

Mean Shifts (#16) cause lighting changes that may be unperceived by metrics using structural similarity approaches like SSIM [23]. These shifts can be simulated by replacing each pixel with the mean of the pixels in a range r neighborhood and whose value is within a distance d. The distance d is a distance function for measuring distances between pixels (usually Euclidean distance or Manhattan distance) and r is the radius (measured according the distance function chosen) that all pixels within it are accounted for the calculation.

$$g(i,j) = f(i,j) + \beta \quad (3.43)$$

The level of distortion is controlled with the PSNR by calculating it on the resulting image.

Figure 3.64: Mean Shift. PSNR $= 30dB$.

Figure 3.65: Mean Shift. PSNR $= 24dB$.

Figure 3.66: Contrast Change. PSNR $= 30dB$.

Figure 3.67: Contrast Change. PSNR $= 24dB$.

3.6.11 Contrast Changes

Contrast can be described as the difference in luminance or colour that makes the content of an image distinguishable [24]. Just like in mean shift distortions, lighting changes may cause contrast change distortions [23]. Gamma correction operation, image processing tools or operators at the visualization stage of an image [24] may also cause it. A common way to simulate this is with multiplication with a constant,

$$g(i,j) = f(i,j) \quad (3.44)$$

where $\alpha > 0$ is called the gain and f is the source image pixels and g is the output image pixels. The parameter α controls the contrast.

3.6.12 Masked Noise

Masked Noise (#4) is a distortion related to local contrast sensitivity of the HVS [17]. Contrast sensitivity is the ability to discern between luminances. Contrast masking is a technique used to fix images with high contrast changes (like blown out highlights or deep dark shadows)[4]. The technique normally consists in changing the colour model from RGB to grayscale using the acMATLAB function "rgb2gray", followed by the image's inverse (also called a B & W negative) and performing a Gaussian blur. The result is then overlayed with the original image and the level is opacity of the mask layer is changed (usually to aproximately 20%). The levels of the image

[4] http://www.photozone.de/contrast-masking

Figure 3.68: Change in Colour Saturation. PSNR $= 30dB$.

Figure 3.69: Change in Colour Saturation. PSNR $= 24dB$.

are adjusted to recover deep black and bright white. Sometimes the result still has distortions striking enough to be visible. This distortion can also occur by lossy image compression or digital watermarking.

3.6.13 Changes in Colour Saturation

Changes in Colour Saturation (#18) are changes in the intensity of colour of an image. This distortion may occur on stages of image acquisition and processing or in JPEG based compression during the colour components quantization [2]. To emulate this distortion, the RGB colour space of the image is changed to $YCbCr$ and maintaining the intensity of the Y channel, while shifting Cb and Cr according to the formulas $Cb = 128 + (Cb - 128) \times K$ and $Cr = 128 + (Cr - 128) \times K$ where, K is a variable parameter that controls the level of distortion [2].

3.6.14 Comfort Noise

Comfort Noise (#20) is based on the knowledge that humans in general do not pay much attention to the existence of noise present in a given image. Also, humans sometimes cannot distinguish texture changes if the texture fragments have the same parameters. These properties are already exploited in lossy compression of images to simultaneously attain larger compression ratio and natural appearance of decompressed data. A reference image is converted from RGB colour space to $YCbCr$. The Y channel is compressed lossly by a DCT-based coder Advanced Discrete Cosine Transform-Based Image Coder (ADCTC) [25] [26] proceeded by decompressing and deblocking, and the Y_r reconstructed image is obtained. A noisy part of the reference image is estimated as $Y_n = Y_r - Y$. The process is repeated for Cb and Cr.

3.6.15 Compression of Noisy Images

Lossy Compression of Noisy Images (#21) usually takes place in compressing images acquired in nonperfect conditions [2]. To model this, independent additive Gaussian noise with variance σ^2 was added to each colour component. The level of distortion is controlled by σ. The lossy compression is done by an ADCTC with the quantization step equal to 1.73σ. Decompression followed by deblocking leads to a distorted image.

Figure 3.70: Comfort Noise. PSNR $= 30dB$.

Figure 3.71: Comfort Noise. PSNR $= 24dB$.

Figure 3.72: Lossy Compression of Noisy Images. PSNR $= 30dB$.

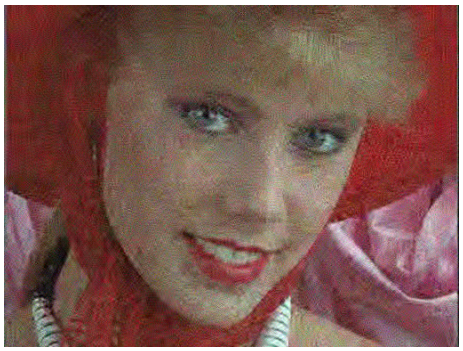


Figure 3.73: Lossy Compression of Noisy Images. PSNR $= 24dB$.

3.6.16 Colour quantization

"Colour quantization is the process of reducing number of colours used in an image while trying to maintain the visual appearance of the original image"[5]. Some quantized images have a problem displaying accurately colour because there might be insufficient bits to represent them, resulting in abrupt changes between shades of the same colour. This is called colour banding. To fix this, dither is used. Dither is pseudo-random noise, added before quantization, used to reduce the statistical dependence between the signal and quantization error [27]. Image colour quantization with dither correction (#22) is intentionally applied to produce noise that randomizes quantization error, preventing colour banding. A way to model this is using the MATLAB function rgb2ind. The image goes from RGB to an indexed image using dither. The number of quantization levels can be adjusted individually to provide a desired PSNR [2].

3.6.17 Chromatic Aberrations

Chromatic Aberrations (#23) normally take place in the acquisition or transformation stages of image processing. It can be modeled by shifting the R, G and B colour components of the image followed by the blur of the resulting image. The shift value and the blurring level control the distortion level. This distortion is particularly difficult to deal with, in places of high contrast and if a distortion level is high [2].

[5] rosettacode.org/wiki/Colour_quantization

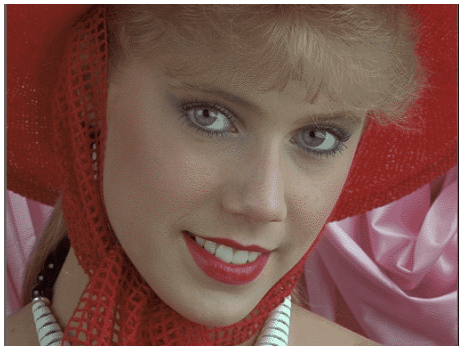


Figure 3.74: Image colour quantization with dither. PSNR $= 30dB$.

Figure 3.75: Image colour quantization with dither. PSNR $= 24dB$.

Figure 3.76: Chromatic aberrations. PSNR $= 30dB$.

Figure 3.77: Chromatic aberrations. PSNR $= 24dB$.

3.6.18 Compressive Sensing

Compressive Sensing (#24) is an approach that unifies signal sensing and signal compression. It consists on the "use of nonadaptive linear projections to acquire an efficient, dimensionally reduced representation of a sparse signal" [28]. With this approach distorted images are able to be reconstructed. It consists in separate modeling of the components Y, Cb and Cr of the image. Each component is subjected to a DCT that reduces the resolution of an image, removing some of it's higher frequencies by zeroing parts of them. The distortion level can be controlled by varying the number of zeroed DCT coefficients. After obtaining the DCT coefficients, the non-zeroed DCT coefficients are restored, and an inverse DCT transform is done followed by the BM3d filter [29] where a filtered image is acquired from the reconstructed one. The BM3D filter is a denoising method that finds image cells similar to a given image cell, groups them in a 3D block, does a 3D linear transform of that block followed by a shrinkage of the transform spectrum coefficients and finally computes a inverse 3D transformation [30]. This 3D filter filters all the 2D image cells in the 3D block. To obtain the figures in 3.78 and 3.79, a DCT was performed again, followed by the redoing of the last step. Ten iterations were done.

Figure 3.78: Compressive sensing.
PSNR $= 30dB$.

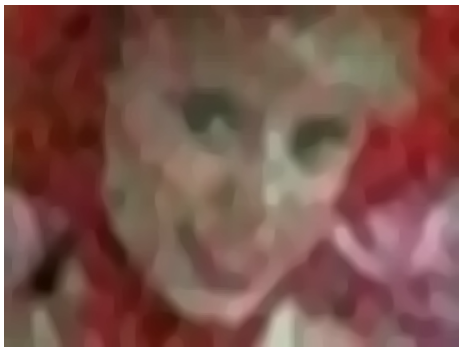


Figure 3.79: Compressive sensing.
PSNR $= 24dB$.

Chapter 4

Introduction to Machine Learning Concepts

In the context of this thesis some concepts, relevant to this study, are introduced in this chapter. Machine Learning studies how to automatically make accurate predictions based on knowledge given by past operations. Some classification problems would be text categorization (like spam filtering) or machine vision (like optical character recognition, face detection, face recognition). It classifies data samples into a given set of categories.
Most of the machine learning methods use pre-annotated samples to teach how to define the classification, in the so-called supervised classification. There are however, methods that do not use any annotated samples, and only use the statistical distribution of the information to be classified to separate it into classes. These are the unsupervised classification models, as the K-Means algorithm that is described in the following.
There are quite a number o supervised machine learning algorithms: decision trees [31], boosting methods [32], SVM [33], neural networks [34], nearest neighbor algorithms [35] and recently the convolutional networks with deep learning methods [36], among others. Most of these methods are binary, meaning that only made the classification as one of two classes. Even methods that do multiple class classification usually perform better when applied for binary classification.
Most of the classifiers need to have the information organized with appropriate descriptors. Descriptors are usually described as vectors with a set of measurements extracted form the information that is intended to classify. Typically, the classification system performance will result of a balance between the use of valuable descriptors that represent the information properly and the appropriate classifier. In practice the ideal classifier does not request any type of descriptors, while the ideal descriptors will require a basic classification.
In this work, the SVMwill be used. This method is a binary classifier known by its typical robustness in a set of multiple applications [37].

4.1 Descriptors and Aggregation of Descriptors

As previously defined descriptors are low level features that result of direct measures over the information. In the context of Image technology, descriptors are extracted globally or locally. Typically, in most of the cases it is better to extract multiple descriptors locally, representing certain local properties of the image. However, these processes might lead to a large number of local descriptors, making difficult the decision process defined by the classifiers. An option is to define descriptor aggregators that somehow represent the information given by the set of local descriptors. There are two main descriptors aggregators, the bag of words model [38] and the Fisher vectors [39].
Analyzing the images' content leads to image classification. It is often useful to reduce it's size and summarize the image's information that is needed to perform an objective evaluation [38]. In machine learning and pattern recognition, a descriptor or feature is "an individual measurable property of a phenomenon being observed" [40]. A set of numeric features is described as a

feature vector.
Choosing the right features and the way to extract them is of the most importance to the machine learning process effectivity.

4.1.1 Bag of Words

One way to do so is by using a BOW model, also known as a bag of features. It is simple to picture this has a book filled with words where order and syntax do not matter [41]. The only thing that matters is the number of each word on there (how many times they repeat).
To obtain a bag of words, first, the clustering of the data is performed to obtain the data's centroids. The data centroids are it's representative features [42]. After getting these features, the local image characteristics can be obtain by using a nearest search method. The local image characteristic is classified with certain word depending which centroid it is closest to. An histogram is formed on these words to count their frequency. The result of that histogram will be the descriptor (the bag of words) that represents the image.

4.1.2 Fisher Vectors

Fisher vectors [39] are image representations obtained by pooling local image features. It is frequently used as a global image descriptor in visual classification. The similarity between the local image features is measured using a function called fisher kernel on the basis of sets of measurements for each feature and a statistical model. A Gaussian Mixture Model [39] is used for the fitting the distribution of the features.
Fisher vectors result in a compact and dense representation, desirable for image classification.

4.2 Classifiers

4.2.0.1 K-means clustering algorithm

K-means is a unsupervised learning algorithm that classifies a given data set into a certain number of clusters (K clusters). It begins by randomly selecting a number of cluster centers (c). Considering $X = X_1, X_2, X_3, ..., X_n$ be the set of data points and $V = V_1, V_2, V_3, ..., V_c$ being c the number of clusters, are the set of cluster centers, with each V_i, $i \in 1, 2, ..., c$, is a vector with dimension n. The partitions are calculated using the distance between each data point X and the cluster centers. The most used distance measures are the squared Euclidean, Cityblock, Manhattan distance (the sum of absolute differences) and cosine (one minus the cosine of the included angle between points, treated as vectors).
The squared Euclidean distance is used in this work

$$d(x, c) = (x - c)(x - c)' \quad (4.1)$$

The data points will be assigned to the cluster center whose distance from the cluster center is the minimum of all the cluster centers distances. The cluster centers are then recalculated using,

$$v_i = \frac{1}{c_i} \sum j = 1^{c_i} x_i \quad (4.2)$$

where c_i represents the number of data points in i^{th} cluster. The distance between each data point is then recalculated to obtain new cluster centers. If no data point was reassigned the algorithm is stopped, otherwise the data points are reassigned just like it was previously described.

4.2.0.2 K-nearest neighbors

KNN is a supervised method that classifies objects based on the closest training examples in the feature space. It is a simple machine learning method used to predict labels of any type. An object is classified by a majority vote of its neighbors, being assigned to the most common class among its K nearest neighbors. For each data point to be scored, the algorithm uses the closest data for estimation, taking advantage of local information and making a decision. Closeness can be defined using any distance metric. The most common are the Euclidean distance, the Cityblock(or Manhattan) distance, Chebychev distance, cosine distance and Minkowsky distance.

Euclidean distance,

$$\sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \quad (4.3)$$

Cityblock or Manhattan,

$$\sum_{i=1}^{k} |x_i - y_i| \quad (4.4)$$

Minkowsky,

$$(\sum_{i=1}^{k} (|x_i - y_i|)^q)^{\frac{1}{q}} \quad (4.5)$$

4.2.1 Support Vector Machines

SVMs are supervised learning models (inferring a function from labeled training data) used for binary classification and regression analysis. Given a set of training examples, belonging to one of two categories(also called classes), a SVM builds a model that assigns new examples to one of the two categories. The similarity functions of the SVM use the so-called kernel functions for pattern analysis [33]. In practice the training descriptors are mapped by the Kernel functions in higher dimensions until a hyper-plane manages to separate the two classes. The same transformations are applied to any descriptor mapping it in the defined space. Then, the side of the hyper-plane where the descriptor was mapped defines the class. The most common kernel functions are the following: linear, polynomial, RBF and sigmoid.

4.3 Machine Learning applied to the Quality Evaluation

Multiple works use Machine Learning for Quality evaluation. Although some works have been developed on the domain of FR, most of the NR recent methods are included, as they compute a set of features that are followed by a classification of quality. Using machine learning these methods try to compensate with the training samples, the lack of a reference image. Objective quality assessment has had a limited success making accurate quality predictions when it comes to NR and RR IQA [7].

Algorithms like Blind Image Quality Index (BIQI) [8] and Distortion Identification based Image Verity and INtegrity Evaluation [43] use SVMs. They both consist in a 2-stage framework where the distortion type is identified in the first stage and image quality is evaluated in the second stage. A multiclass SVM with a RBF kernel is used to classify a given image into a distortion cathegory. After that another SVM computes a quality score. BRISQUE [7], mentioned earlier, also uses SVMs with a RBF kernel.

QAC, also mentioned earlier, despite not using human scores (it uses SSIM or FSIM as scores), uses K-means clustering on a training set to obtain centroids that will be used to classify the image.

BLIINDS is an NR IQA algorithm that uses natural scene statistics models of DCT coefficients that are applied in a probabilistic model for quality score prediction.

Other methods [44] use edge amplitude, edge length, background activity and background luminance as features to train a Feed-Forward Neural Network (FFNN).

Redi et al. used Color coreograms as image descriptors and Circular Back-Propagation (CBP) networks for the quality classification [45].

Other works have been presented that use the SSIM metric for in machine learning [46, 47, 48]. A variation of the SSIM called Complex Wavelet Structural SIMilarity index (CWSSIM) [49] followed by clustering of the results, is used in a training set to train a SVM [48]. Another version of SSIM called Wavelet based Structural SIMilarity index (WSSIM) [50] in [47] is used to train a Back Propagation feed-forward Neural Network (BPNN). In [46], a FFNN and a Principal Component Regression based algorithm is applied to create a feature vector for each image, concatenating the SSIM scalar features luminance, contrast and structure.

Convolutional Neural Networks with Deep Learning [51] has been tried, but research have been facing problems caused by the lack of large annotated database. To compensate this, apart the use of specific annotated databases, authors generate very large databases with multiple distortions and annotate them with the traditional FR metrics. However, because of the training methodology, it is not expected that the metrics used in the training can be overcome. Nevertheless, this kind of classification methodologies define a NR model that can approximate the traditional FR metrics performance.

Chapter 5

Bag of Words Model Description Scheme for Image Quality Assessment

The quality estimation model described in this dissertation is based on the spatial, location and content factors influence on the perception of quality. Basically the main content of an image is the area where impairments might have more influence on the definition of quality. Moreover, the way the impairments are distributed and their global location must have a strong influence. Hence, our approach divides an image in cells that are used to compute a set of quality descriptors. These quality descriptors are then grouped using a BOW model [52]. Finally this BOW is used as an identifier of quality and a final quality estimation is obtained using a supervised classification.

5.1 Local Image Quality Descriptors Computation

In this section the computation of the quality descriptors is described. For that, each image is divided into square cells with fixed size, with $DCell \times DCell$ pixels. The FR IQA metric result between each corresponding image cell of the distorted and the original images is computed (see Fig.5.3). In case of cells completely black or white, the IQA metric result will be set to 0 (avoiding divisions by zero).
Furthermore, the cells are further grouped into square windows of $NCell \times NCell$ cells. The set of metric result values in a window will define a local quality descriptor with size of $NCell \times NCell$ bins. A set of image descriptors is computed for each image by sliding the window by $IncCell$ cells until the image limits are reached. Obviously, if the value of $IncCell$ is smaller than $NCell$, there will be overlapping windows (see Fig. 5.1). With this local description it is expected that the model will adapt better to the influence of the local content and to the specific area of the image that might have a larger influence on the quality perception of observers. After this we will have a local quality descriptor for each image.

5.2 Local descriptors aggregation using a Bag of Words

Finally, the set of local descriptors of an image computed for each window location will be grouped using a BOW model [52]. Each*BOW* quality descriptor is determined for each quality level classification. As was decided to classify the image quality into 5 levels, 4 decision steps are needed, and hence, a *BOW* quality descriptor explicitly defined for each quality level classification is used, resulting in four quality descriptors for each image.
For the definition of each bag of words, an image training set is selected randomly, according with the training selection method explained in section 5.4. All the local quality descriptors of these training images are grouped into K clusters using the K-means clustering algorithm [53]. Then each local quality descriptor of an image is allocated to one of the clusters following the smaller Euclidean distance to each K-means cluster centroid. A histogram of the local quality

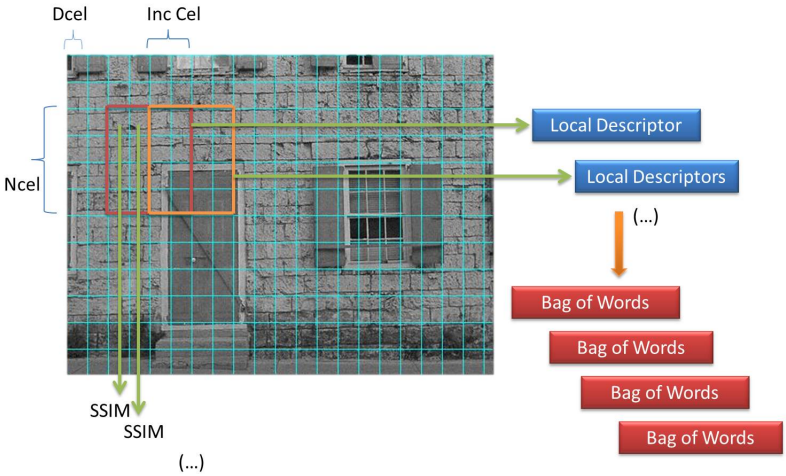


Figure 5.1: Description Scheme.

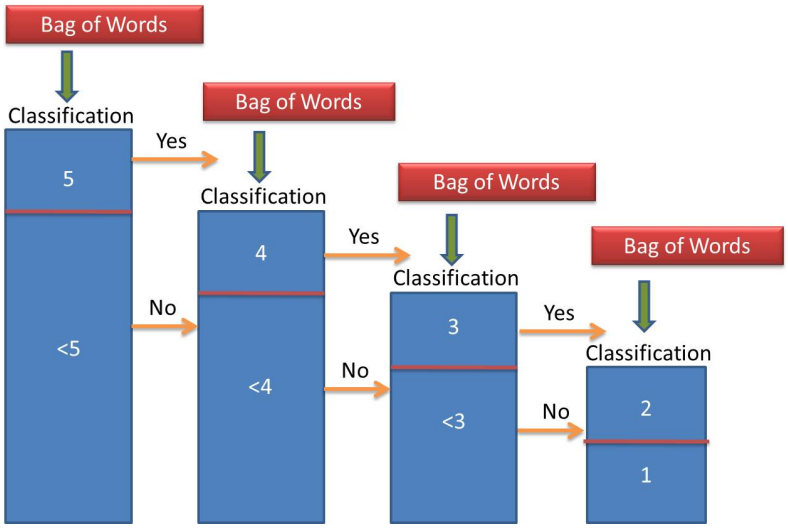


Figure 5.2: Image quality classification.

descriptor clusters will be used as a *BOW* quality descriptor. In practice, each *BOW* quality descriptor will have K bins and they count the number of local image quality descriptors that belong to each cluster previously defined by the K-means algorithm.

5.3 Classification of the quality level of an image

The final quality level assessment of an image is obtained classifying the *acBOW* quality descriptors. The classification is defined by a set of SVMs with RBF kernel, that classify the *BOW* quality descriptors until one of the quality levels is defined. Hence, the initial classification step classifies between the highest quality level (5) or not. If the classification is not the highest level, a new step classifies if the image has a quality level of 4, or not. The process is continued until a final classification of one level results, as it is shown in Fig. 5.2. Each classification step uses the appropriate *BOW* quality descriptor, obtained with the appropriate training set.

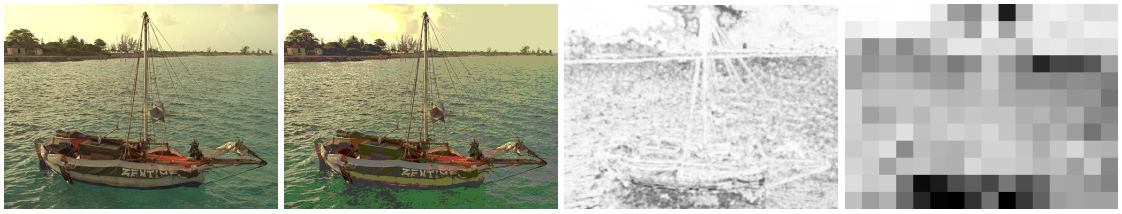
5.4 Training Selection

The defined algorithm requires a set of training data that will be used on the definition of the bag of words model and for training of the used SVM. This data is selected randomly between the 25 images plus impairments of the TID2013 database [2]. However, for the training set candidates of a given image can not be used the original image and none of the distorted versions. Hence, the training set is randomly selected between the remaining 24 images and respective versions with impairments. In the testing of the reported method, the same training set was defined for both *BOW* quality descriptor and respective SVM classifier. Initially, the training set subjective results defined by the MOS are ceiled between 1 and 5. Hence a classification of the quality into 5 different levels: 1 (Lower quality level) to 5 (Higher quality level), is sought by our model.

5.5 Classification of an image bow quality descriptor using a Binary Support Vector Machine

Since it is desired to have the same number of training images for the two classes in each SVM classifier, all the available local quality descriptors of the smaller class will be used. The same number of training samples will be selected randomly between the images of the class with larger number of available samples. Using this training set selection demands the need of multiple tests to evaluate the performance of the algorithm, as one of the training sets is obtained randomly. Hence, the method was tested using a ten fold cross-validation.

5.6 Analysis of Results for SSIM



(a) (b) (c) (d)

Figure 5.3: Example of Reference Image, Distorted Image, SSIM of entire image, and the mean SSIM outcome for the cell division ($DCell = 32$).

The IQA metric used here with the algorithm is SSIM. An example is shown in figure 5.3 where the outcome of the metric in each cell is shown.
To evaluate the reliability of each analyzed combination for the parameters $DCell$:$NCell$:$IncCell$, the Pearson and Spearman correlations, and RMSE were computed, between the MOS$_p$ obtained for each image and MOS$_n$. The results of the Kendall correlation are also shown, as they were used for the evaluation of different metrics with TID2013 database [2].
Several combinations of the parameters values $DCell$: $NCell$:$IncCell$ have been tested. In these plots are reported the combinations 16:4:1, 16:8:1, 24:4:1, 24:8:1, 32:4:1, 32:8:1, 48:4:1, 64:4:1, 32:8:2, 32:8:4. The results of the ten fold cross-validation experiment can be visualized in the graphs of Figs. 5.4, 5.5, 5.6 and 5.7 for the Pearson, Spearman and Kendall correlations,

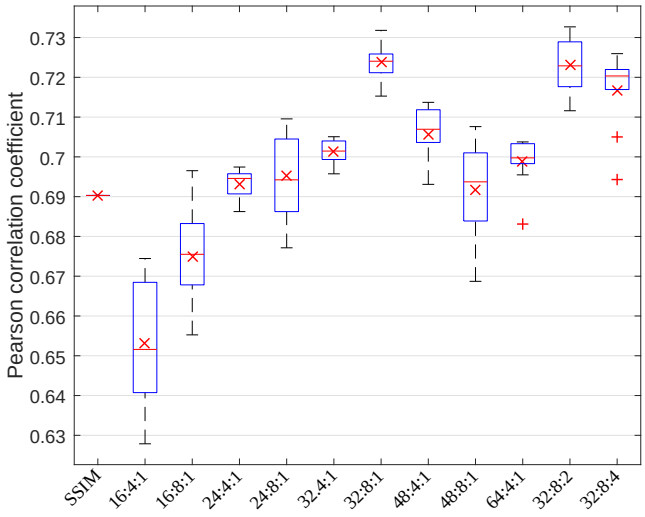


Figure 5.4: Pearson correlation coefficient for $2 \times NCell$ bins *BOW* boxplots for the ten fold cross-validation results. '$\times$' marks the mean result.

Table 5.1: Pearson correlation coefficient mean values.

Combination	Noise	Actual	Simple	Exotic	New	Colour	Full
SSIM Metric	0.7494	0.7741	0.8219	0.6723	0.6975	**0.6925**	0.6903
16:4:1	0.7041	0.7296	0.7714	0.6171	0.6556	0.6595	0.6530
16:8:1	**0.7619**	**0.7911**	0.8448	0.6260	**0.6990**	0.6784	0.6750
24:4:1	0.7361	0.7702	0.8100	0.6693	0.7037	0.6922	0.6931
24:8:1	0.7203	0.7549	0.8003	0.7126	0.6756	0.6601	0.6954
32:4:1	0.7251	0.7620	0.8018	0.6989	0.6962	0.6791	0.7012
32:8:1	0.7573	0.7828	**0.8482**	0.7511	0.6973	0.6707	**0.7237**
48:4:1	0.7191	0.7547	0.8019	0.7238	0.6868	0.6737	0.7055
48:8:1	0.7047	0.7352	0.7726	0.7250	0.6615	0.6404	0.6917
64:4:1	0.7158	0.7522	0.8029	0.7161	0.6744	0.6681	0.6987
32:8:2	0.7406	0.7733	0.8214	**0.7572**	0.6756	0.6634	0.7230
32:8:4	0.7344	0.7687	0.8200	0.7456	0.6732	0.6562	0.7160

and for the RMSE, respectively. In these cases, the *BOW* quality descriptors dimension was of $2 \times NCell$ bins (value of K in the K-means algorithm). In these plots the '$\times$' symbols in red represent the mean values and the red line represent the median values. '$+$' represents outlier values. Moreover, the SSIM result was also added to allow a comparison of the method improvement.

As can be seen the described method improves the SSIM quality estimation. The best result between the tested parameterization was obtained for values of $DCell$:$NCell$:$IncCell$ = 32:8:1. However, similar results are obtained for $DCell$: $NCell$:$IncCell$ = 32:8:2 that corresponds to less overlapping between the local descriptors, providing a faster implementation.

Tables 5.1, 5.2, 5.3 and 5.4 show a complete analysis for each type of distortion as designated in the TID2013 database [2]. The subset "Noise" contains different types of noise distortions common in image processing. The "Actual" subset has the most common types of distortion in compression among other practices of image processing. The subset "Simple" consists in

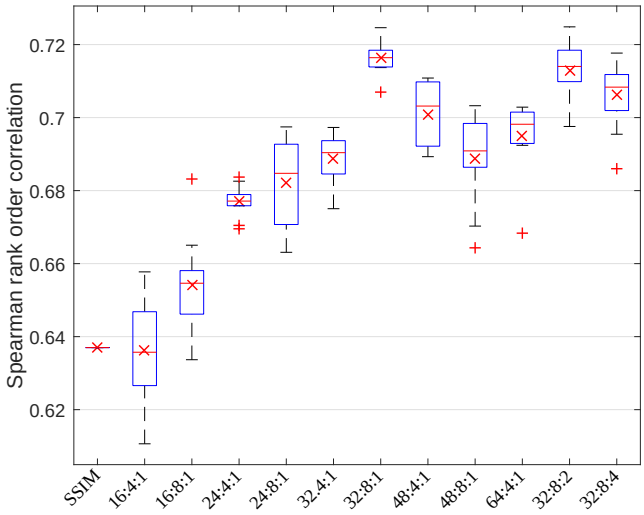


Figure 5.5: Spearman rank order correlation for $2 \times NCell$ bins *BOW* boxplots for the ten fold cross-validation results. '$\times$' marks the mean result.

Table 5.2: Spearman rank order correlation mean values.

Combination	Noise	Actual	Simple	Exotic	New	Colour	Full
SSIM Metric	0.7574	0.7877	0.8371	0.6320	0.5801	0.5057	0.6370
16:4:1	0.7105	0.7433	0.7778	0.5765	0.6243	0.6161	0.6363
16:8:1	**0.7688**	**0.8028**	0.8517	0.5867	0.6388	0.6297	0.6541
24:4:1	0.7435	0.7833	0.8202	0.6265	**0.6741**	**0.6519**	0.6771
24:8:1	0.7266	0.7620	0.8097	0.6935	0.6246	0.6134	0.6820
32:4:1	0.7308	0.7730	0.8112	0.6750	0.6640	0.6351	0.6888
32:8:1	0.7638	0.7946	**0.8482**	0.7419	0.6445	0.6303	**0.7165**
48:4:1	0.7271	0.7674	0.8165	0.7175	0.6488	0.6304	0.7010
48:8:1	0.7114	0.7466	0.7869	0.7271	0.6210	0.5933	0.6888
64:4:1	0.7231	0.7644	0.8183	0.7129	0.6368	0.6281	0.6951
32:8:2	0.7454	0.7827	0.8353	**0.7456**	0.6219	0.6195	0.7131
32:8:4	0.7385	0.7781	0.8353	0.7339	0.6147	0.6134	0.7063

three commonly standard types of distortion. The "Exotic" subset includes the rarest and most complex distortions in visual quality metrics. The subset "New" contains seven new types of distortions introduced to TID2013. The "Colour" subset corresponds to distortion types connected with changes of colour content. Finally, "Full" contains all types of distortions. Every result displayed in the tables 5.1 to 5.4 is the mean of the ten fold cross-validation results.

The Pearson correlation only shows a slight increase on every distortion with the exception of "Colour". For this reason, a further statistical significance analysis is required. For a further analysis of the results, a Shapiro-Wilk test [54] performed with significance level, α, equal to 0.05 was applied to the Pearson correlation values obtained in the cross-validation. It was concluded that these values are normally distributed, therefore an Analysis of Variance (ANOVA) [55] can be applied. The Kendall and Spearman correlations though, show a significant increase even in "Colour". The algorithm seems particularly effective on "Exotic" distortions. Fig. 5.8 represents the Pearson correlation boxplots for each distortion in particular. The SSIM result, the

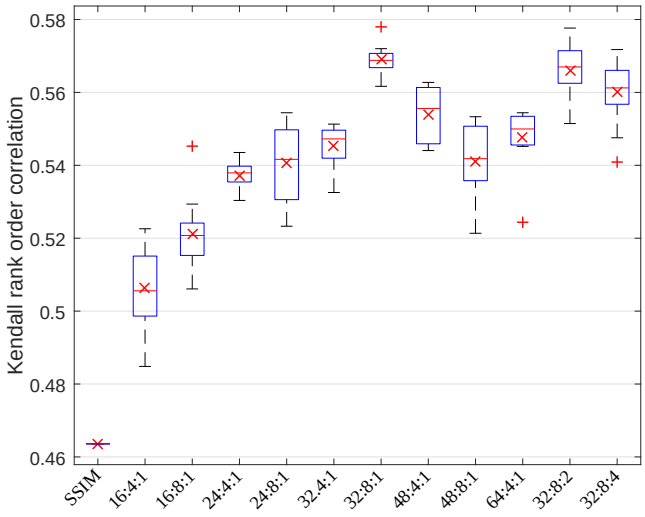


Figure 5.6: Kendall rank order correlation for $2 \times NCell$ bins *BOW* boxplots for the ten fold cross-validation results. '$\times$' marks the mean result.

Table 5.3: Kendall rank order correlation mean values.

Combination	Noise	Actual	Simple	Exotic	New	Colour	Full
SSIM Metric	0.5515	0.5768	0.6286	0.4548	0.4226	0.3823	0.4636
16:4:1	0.5658	0.5940	0.6274	0.4583	0.4914	0.4970	0.5065
16:8:1	**0.6230**	**0.6557**	**0.7074**	0.4590	0.5065	0.5080	0.5211
24:4:1	0.5880	0.6237	0.6569	0.4991	**0.5320**	**0.5243**	0.5373
24:8:1	0.5790	0.6099	0.6555	0.5506	0.4920	0.4914	0.5405
32:4:1	0.5762	0.6129	0.6460	0.5377	0.5206	0.5085	0.5452
32:8:1	0.6116	0.6392	0.6956	0.5890	0.5064	0.5056	**0.5692**
48:4:1	0.5728	0.6081	0.6557	0.5706	0.5061	0.5045	0.5540
48:8:1	0.5576	0.5877	0.6254	0.5765	0.4818	0.4696	0.5411
64:4:1	0.5682	0.6043	0.6566	0.5648	0.4965	0.5027	0.5478
32:8:2	0.5927	0.6254	0.6785	**0.5971**	0.4869	0.4947	0.5660
32:8:4	0.5862	0.6204	0.6784	0.5870	0.4800	0.4888	0.5600

best general combination 32:8:1, and the best parameter combination for each distortion subset is presented for each subset type (see Fig. 5.8).

As can be seen, combinations of $DCell$:$NCell$:$IncCell$ varies considering the type of distortion. Obviously, the number of images for each type of distortion is smaller than for the full analysis of Figs. 5.4 to 5.7, making the cross-validation results less reliable. Although the amount of data does not allow a reliable conclusion, it can be seen that the best combination for all subsets (32:8:1) provides a very competing result, and it is the best for the "Simple" distortions subset. However, in case of the "Colour" subset, the SSIM provides the best result, in opposition to the other distortion subsets where the developed method always manages to improve the SSIM estimation.

It was decided to do an analysis of the influence of the *bow* quality descriptors dimension, that is equal to the K parameter of the K-Means used in the bag of words determination. For that the 32:4:1 and 32:8:1 combinations were selected. The correlation graphs in Fig. 5.12, Fig. 5.13,

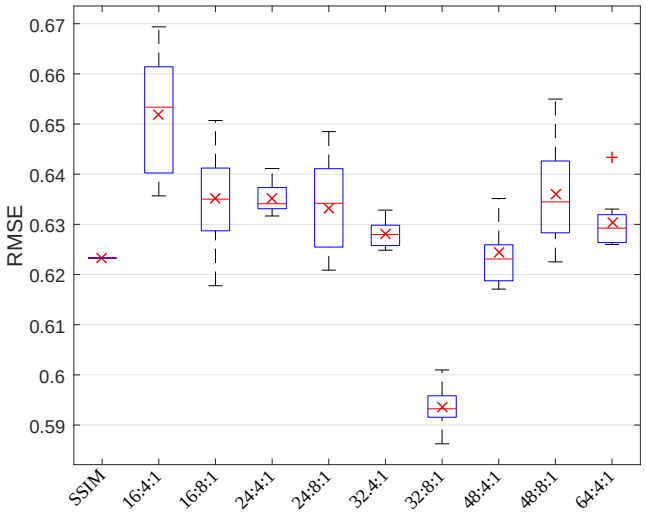


Figure 5.7: RMSE for $2 \times NCell$ bins *BOW* boxplots for the ten fold cross-validation results. '$\times$' marks the mean result.

Table 5.4: RMSE mean values.

Combination	Noise	Actual	Simple	Exotic	New	Colour	Full
SSIM Metric	0.4961	0.5334	0.4852	0.6970	0.6321	**0.5981**	0.6233
16:4:1	0.5323	0.5763	0.5429	0.7397	0.6650	0.6225	0.6519
16:8:1	**0.4857**	**0.5155**	**0.4564**	0.7327	**0.6295**	0.6295	0.6352
24:4:1	0.5245	0.5531	0.5193	0.7110	0.6356	0.6120	0.6353
24:8:1	0.5372	0.5686	0.5301	0.6708	0.6593	0.6370	0.6332
32:4:1	0.5337	0.5616	0.5291	0.6841	0.6421	0.6223	0.6282
32:8:1	0.4893	0.5240	0.4725	0.6198	0.6309	0.6149	**0.5935**
48:4:1	0.5382	0.5688	0.5287	0.6602	0.6502	0.6267	0.6243
48:8:1	0.5493	0.5877	0.5619	0.6578	0.6706	0.6508	0.6361
64:4:1	0.5409	0.5714	0.5278	0.6685	0.6606	0.6309	0.6304
32:8:2	0.5035	0.5339	0.4859	**0.6136**	0.6492	0.6200	0.5942
32:8:4	0.5086	0.5386	0.4875	0.6260	0.6512	0.6251	0.6000

Fig. 5.14, Fig. 5.15, Fig. 5.16, Fig. 5.17 show that $2 \times NCell$ bins dimension seems to provide the best value in both combinations due to a slightly better mean and a larger stability from the test results, defining a more reliable and stable outcome.

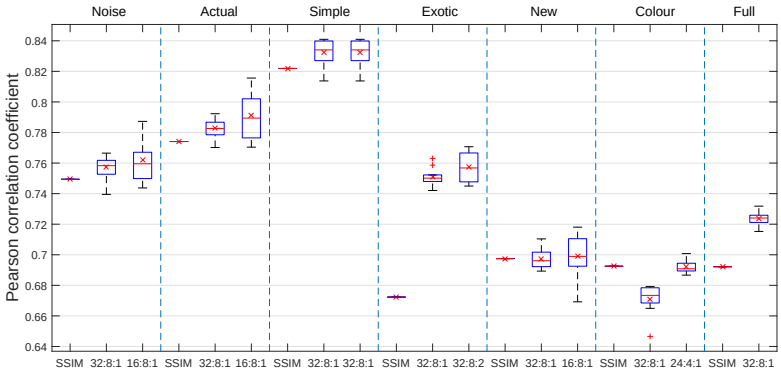


Figure 5.8: Pearson correlation coefficient comparing the SSIM, the 32:8:1 and the highest scoring combination for all distortion subsets for the ten fold cross-validation results (mean result signalized by the '×').

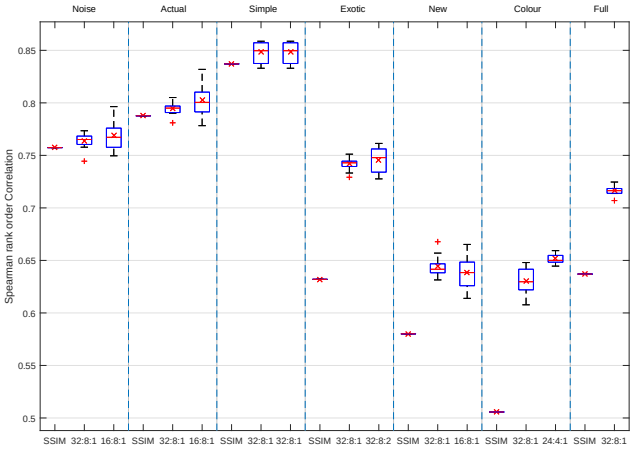


Figure 5.9: Spearman rank order correlation comparing the SSIM, the 32:8:1 and the highest scoring combination for all distortion subsets for the ten fold cross-validation results (mean result signalized by the '×').

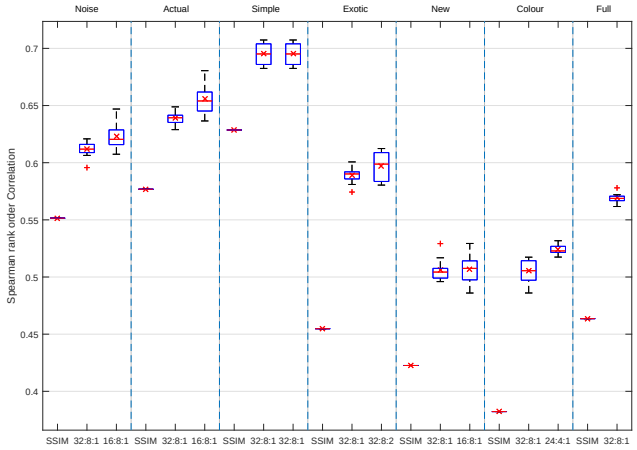


Figure 5.10: Kendall rank order correlation comparing the SSIM, the 32:8:1 and the highest scoring combination for all distortion subsets for the ten fold cross-validation results (mean result signalized by the '×').

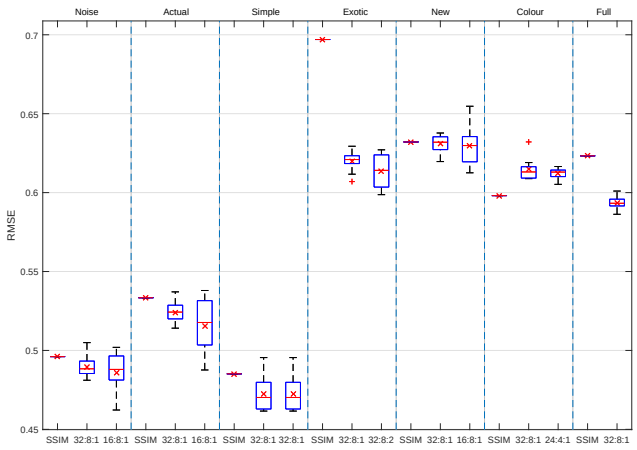


Figure 5.11: RMSE comparing the SSIM, the 32:8:1 and the highest scoring combination for all distortion subsets for the ten fold cross-validation results (mean result signalized by the '×').

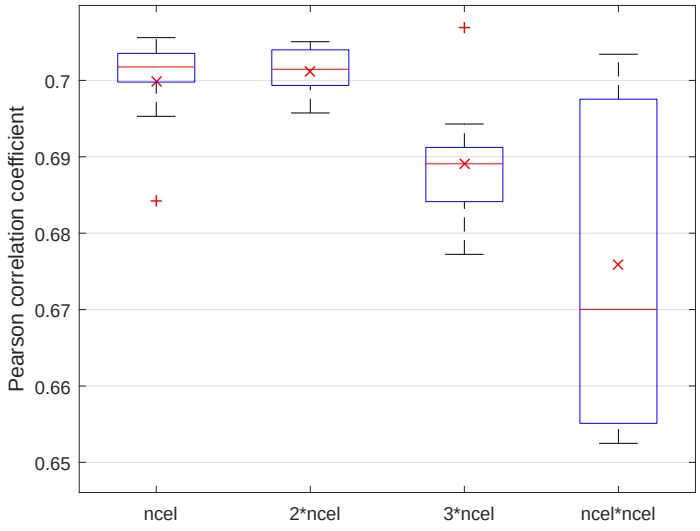


Figure 5.12: Pearson correlation coefficient for different *bow* dimension (K of K-means) for combinations 32:4:1 for the ten fold cross-validation results (mean result signalized by the '$\times$').

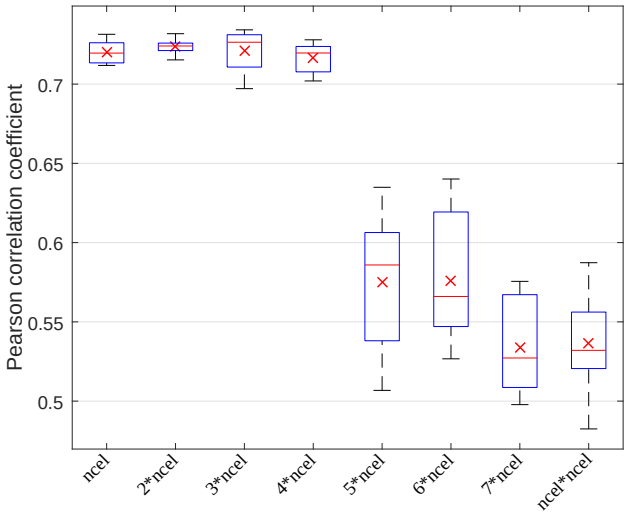


Figure 5.13: Pearson correlation coefficient for different *bow* dimension (K of K-means) for combinations 32:8:1 for the ten fold cross-validation results (mean result signalized by the '$\times$').

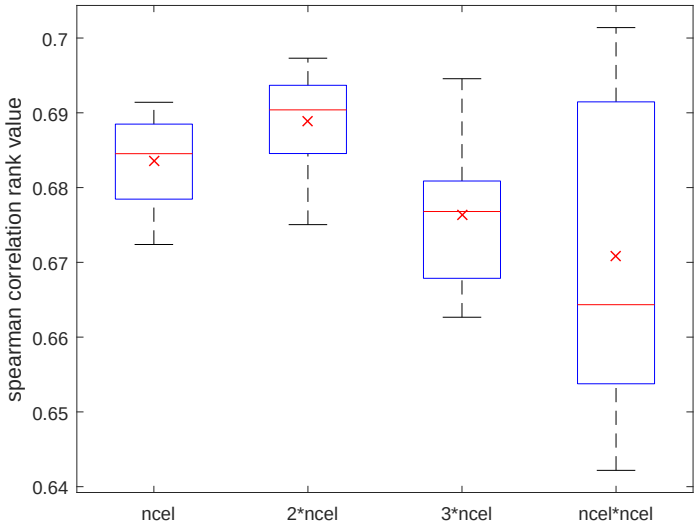


Figure 5.14: Spearman correlation rank value for different *bow* dimension (K of K-means) for combinations 32:4:1 for the ten fold cross-validation results (mean result signalized by the '$\times$').

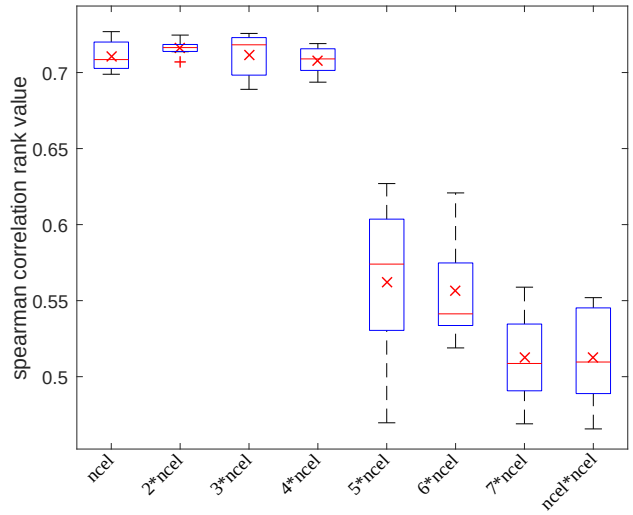


Figure 5.15: Spearman correlation rank value for different *bow* dimension (K of K-means) for combinations 32:8:1 for the ten fold cross-validation results (mean result signalized by the '$\times$').

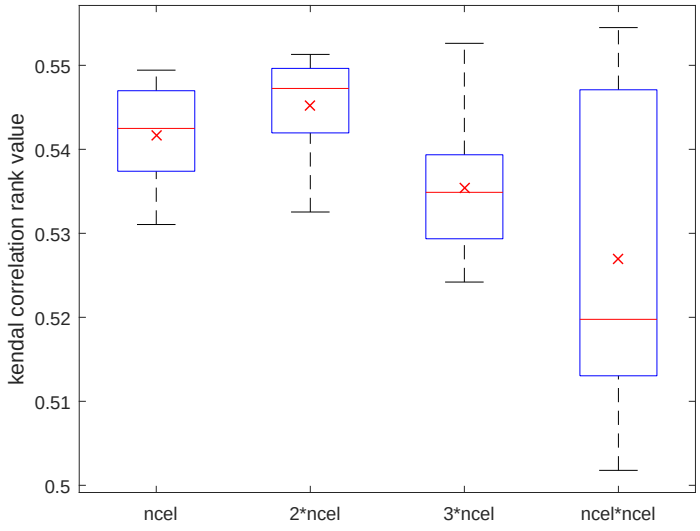


Figure 5.16: Kendall correlation rank value for different *bow* dimension (K of K-means) for combinations 32:4:1 for the ten fold cross-validation results (mean result signalized by the '$\times$').

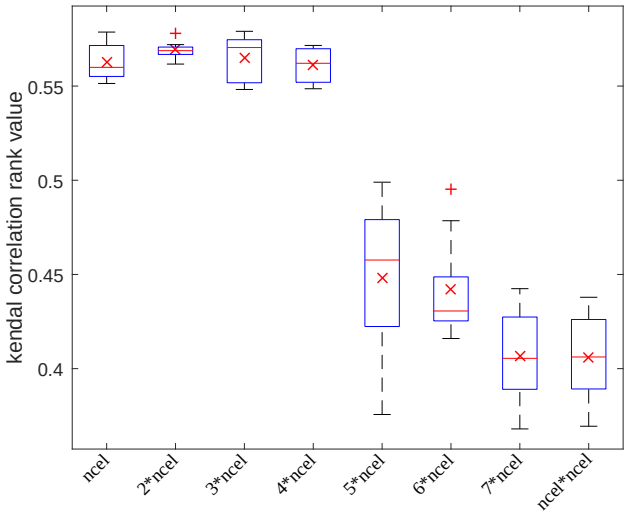


Figure 5.17: Kendall correlation rank value for different *bow* dimension (K of K-means) for combinations 32:8:1 for the ten fold cross-validation results (mean result signalized by the '$\times$').

Chapter 6

Comments and Future Work

In this work, a scheme for quality estimation using a machine learning method based on the FR metric SSIM was considered. The method computes the SSIM in a grid of cells that are further grouped to form a set of local descriptors. Those local descriptors are further aggregated for each image using BOW. Four BOW are extracted for each image allowing a classification using a SVM into the 5 different MOS levels.

With this approach it is expected that the different impairments location and also the affected content is considered in the quality estimation. The developed method was tested against the quality representation provided by the SSIM metric using the Pearson, Spearman and Kendal correlations, and also the RMSE. It was shown that the developed method outperforms the SSIM, and for that reason provides a very interesting path for research using machine learning methods.

It was observed that similar improvement can be obtained with other metrics, that are not reported as it would become very repetitive. However, in the future the use of metrics fusion can be considered. Moreover, the strategy followed can produce new insights to NR and RR metrics. development. Instead of computing the quality features over the image, might be interesting to do a similar approach, to enhance the local influence of the impairments.

This is an interesting approach using machine learning models. Using Convolutional Neural Networks with Deep Learning [51] has been tried recently, but results are very limited because of the lack of training samples with appropriate annotation.

Bibliography

[1] Z. W. A. C. B. H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600–612, 2004. xi, 6, 7, 8, 10, 11, 13

[2] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C.-C. J. Kuo, "Image database TID2013: Peculiarities, results and perspectives," *Signal Processing: Image Communication*, vol. 30, pp. 57–77, Jan. 2015. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01109219 1, 16, 20, 21, 28, 29, 39, 40

[3] P. L. Callet, S. Möller, and A. Perkis, Eds., *QUALINET White Paper on Definitions of Quality of Experience*. Lausanne, Switzerland: European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), June 2012. 1, 5

[4] B. Keelan, *Handbook of Image Quality: Characterization and Prediction*, ser. Optical Science and Engineering. CRC Press, 2002. 1

[5] M. Fidalgo-Fernandes, M. V. Bernardo, and A. M. G. Pinheiro, "A bag of words description scheme based on ssim for image quality assessment," in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, June 2016, pp. 1–6. 3

[6] Z. Wang, , Z. Wang, and A. C. Bovik, "Why is image quality assessment so difficult?" in *in Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, 2002, pp. 3313–3316. 5

[7] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process*, pp. 4695–4708, 2012. 5, 14, 15, 36

[8] A. K. Moorthy and A. C. Bovik, "A two-step framework for constructing blind image quality indices," *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 513–516, May 2010. 5, 36

[9] X. M. Lin Zhang, Lei Zhang and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Processing*, vol. 20, pp. 2378–2386, 2011. 5, 13

[10] B. Girod, "What's wrong with mean-squared error," *Digital Images and Human Vision (A. B. Watson, ed.)*, pp. 207–220, 1993. 6

[11] P. Hill, A. Achim, M. E. Al-Mualla, and D. Bull, "Contrast sensitivity of the wavelet, dual tree complex wavelet, curvelet, and steerable pyramid transforms," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2739–2751, June 2016. 8

[12] H. Abdi, "The kendall rank correlation coefficient," *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pp. 508–510, 2007. 9

[13] W. Xue, L. Zhang, and X. Mou, "Learning without human scores for blind image quality assessment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 995–1002. 12, 13

[14] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423. 12

[15] J. D. Gibson and A. Bovik, Eds., *Handbook of Image and Video Processing*, 1st ed. Orlando, FL, USA: Academic Press, Inc., 2000. 18, 19, 22, 24

[16] N. Ponomarenko, V. Lukin, A. Zelensky, and K. Egiazarian, "Tid2008-a database for evaluation of full-reference visual quality assessment metrics," vol. 10, pp. 30–45, 2009. 20, 24, 26

[17] N. Ponomarenko, V. Lukin, K. Egiazarian, J. Astola, M. Carli, and F. Battisti, "Color image database for evaluation of image quality metrics," in *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, Oct 2008, pp. 403–408. 21, 27

[18] J. Flusser, S. Farokhi, C. Höschl, T. Suk, B. Zitová, and M. Pedone, "Recognition of images degraded by gaussian blur," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 790–806, Feb 2016. 23

[19] D. Brunet, E. R. Vrscay, and Z. Wang, "The use of residuals in image denoising," in *International Conference Image Analysis and Recognition*. Springer, 2009, pp. 1–12. 23

[20] E. Vansteenkiste, D. V. der Weken, W. Philips, and E. Kerre, "Psycho-visual quality assessment of state-of-the-art denoising schemes," in *Signal Processing Conference, 2006 14th European*, Sept 2006, pp. 1–5. 24

[21] G. Yu and G. Sapiro, "DCT Image Denoising: a Simple and Effective Image Denoising Algorithm," *Image Processing On Line*, vol. 1, 2011. 24

[22] N. Akhtar, S. Khan, and G. Siddiqui, "A novel lossy image compression method," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*. IEEE, 2014, pp. 866–870. 24

[23] A. C. Brooks, X. Zhao, and T. N. Pappas, "Structural similarity quality metrics in a coding context: exploring the space of realistic distortions," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1261–1273, 2008. 26, 27

[24] A. Samani, K. Panetta, and S. Agaian, "Contrast enhancement for color images using discrete cosine transform coefficient scaling," in *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, May 2016, pp. 1–6. 27

[25] N. Ponomarenko, V. Lukin, K. Egiazarian, and E. Delp, "Comparison of lossy compression performance on natural color images," in *Picture Coding Symposium, 2009. PCS 2009*, May 2009, pp. 1–4. 28

[26] N. Ponomarenko, V. Lukin, K. Egiazarian, and J. Astola, "Adctc: Advanced dct-based image coder," *Department of Transmitters, Receivers and Signal Processing, National Aerospace University*, 2008. 28

[27] R. Hadad and U. Erez, "Dithered quantization via orthogonal transformations," *IEEE Transactions on Signal Processing*, vol. 64, no. 22, pp. 5887–5900, Nov 2016. 29

[28] J. L. Paredes and G. R. Arce, "Compressive sensing signal reconstruction by weighted median regression estimates," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2585–2601, June 2011. 30

[29] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, "Spatially adaptive filtering as regularization in inverse imaging: Compressive sensing, super-resolution, and upsampling," pp. 123–153, 2010. 30

[30] M. Lebrun, "An analysis and implementation of the bm3d image denoising method," *IPOL Journal*, vol. 2, pp. 175–213, 2012. 30

[31] L. Rokach, *Introduction to Decision Trees*. World Scientific, 2008. 33

[32] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1619–1630, Oct 2006. 33

[33] X. Li and Y.-y. Wang, *Advances in Computer Science and Information Engineering: Volume 2*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. Image Quality Assessment Method Based on Support Vector Machine and Particle Swarm Optimization, pp. 353–359. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30223-7_55 33, 35

[34] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4–22, Apr 1987. 33

[35] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418–435, May 1992. 33

[36] Y. LeCun, "Deep learning convolutional networks," in *2015 IEEE Hot Chips 27 Symposium (HCS)*, Aug 2015, pp. 1–95. 33

[37] Vladimir Vapnik, *The nature of statistical learning theory*. Springer-Verlag, 1995. 33

[38] N. M. Ali, S. W. Jun, M. S. Karis, M. M. Ghazaly, and M. S. M. Aras, "Object classification and recognition using bag-of-words (bow) model," in *2016 IEEE 12th International Colloquium on Signal Processing Its Applications (CSPA)*, March 2016, pp. 216–220. 33

[39] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International journal of computer vision*, vol. 105, no. 3, pp. 222–245, 2013. 33, 34

[40] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, 2006. 33

[41] L. Zhi-Jie, "Image classification method based on visual saliency and bag of words model," in *2015 8th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, June 2015, pp. 466–469. 34

[42] N. Passalis and A. Tefas, "Entropy optimized feature-based bag-of-words representation for information retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1664–1677, July 2016. 34

[43] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3350–3364, Dec 2011. 36

[44] S. Suresh, R. V. Babu, and H. Kim, "No-reference image quality assessment using modified extreme learning machine classifier," *Applied Soft Computing*, vol. 9, no. 2, pp. 541 – 552, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494608001075 36

[45] P. Gastaldo and J. A. Redi, "Machine learning solutions for objective visual quality assessment," in *Sixth international workshop on Video Processing and Quality Metrics (VPQM)*, 01/2012 2012. [Online]. Available: http://enpub.fulton.asu.edu/resp/vpqm/vpqm12/Papers/vpqm12_p17.pdf 36

[46] A. Hines, P. Kendrick, A. Barri, M. Narwaria, and J. A. Redi, "Robustness and prediction accuracy of machine learning for objective visual quality assessment," in *22nd European Signal Processing Conference, EUSIPCO 2014, Lisbon, Portugal, September 1-5, 2014*, 2014, pp. 2130–2134. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6952766 36

[47] A. Hu, R. Zhang, D. Yin, and W. Hu, "Machine learning-based multi-channel evaluation pooling strategy for image quality assessment," in *2013 IEEE International Conference on Image Processing*, Sept 2013, pp. 427–430. 36

[48] Y. Gao, A. Rehman, and Z. Wang, "Cw-ssim based image classification," in *2011 18th IEEE International Conference on Image Processing*, Sept 2011, pp. 1249–1252. 36

[49] Z. Wang and E. Simoncelli, *Translation insensitive image similarity in complex wavelet domain*, 2005, vol. II. 36

[50] A. Hu, R. Zhang, X. Zhan, and D. Yin, "Image quality assessment incorporating the interaction of spatial and spectral sensitivities of hvs," in *Proc. of 2011 the 13th IASTED International Conference on Signal and Image Processing*, 2011. 36

[51] L. Krasula, K. Fliegel, P. L. Callet, and M. Klíma, "On the accuracy of objective image and video quality models: New methodology for performance evaluation," in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, June 2016, pp. 1–6. 36, 49

[52] Z. Yang and T. Kurita, "Improvements to the descriptor of SIFT by bof approaches," in *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*, Nov 2013, pp. 95–99. 37

[53] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall PTR, 2011. 37

[54] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 3, no. 52, 1965. 41

[55] R. V. Hogg and J. Ledolter, *Engineering statistics*. Macmillan Pub Co, 1987. 41