Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea

Effective features for No-Reference Image Quality Assessment on Mobile Devices

Laureando

Simone Brazzo

Relatore

Prof. Carlo Fantozzi

Anno Accademico 2016/2017

A Fabiana, che ogni giorno mi sta accanto con la sua forza e coerenza.
Senza di te non sarei mai arrivato a questo traguardo.

Alla mia famiglia, senza la quale non sarei la persona che sono.
Non finirò mai di ringraziarvi perché avete sempre fatto
in modo che potessi inseguire i miei sogni.

Abstract

Questa tesi ha come obiettivo l'analisi e lo sviluppo di un algoritmo per la valutazione senza riferimenti della qualità delle immagini. Questa tipologia di algoritmi è parte integrante di qualsiasi applicazione multimediale, con lo scopo di fornire una crescente qualità per i servizi proposti. Un algoritmo allo stato dell'arte è stato scelto come punto di partenza per lo sviluppo del lavoro di tesi.
L'algoritmo proposto è stato sviluppato tenendo conto di un futuro impiego in sistemi a bassa disponibilità di risorse, come smarthphone e tablet. A questo proposito, il tempo di esecuzione è stato misurato tramite un applicazione sviluppata per smartphone. Infine, le performance ottenute in termini di valutazione della qualità sono state confrontate con quelle dell'algoritmo di partenza, attraverso una tecnica largamente utilizzata.

Abstract

The goal of this thesis is the analysis and development of a no-reference image quality assessment algorithm. Algorithms of this kind are increasingly employed in multimedia applications with the aim of delivering higher quality of service. In order to achieve the goal, a state-of-art no-reference algorithm was used as a groundwork to improve.
The proposed model is intended to be deployed in low-resources mobile devices such as smartphones and tablets. A mobile application employing the new algorithm was developed in order to measure its running time. Moreover, the quality assessment improvements were proved with respect to the groundwork algorithm using a broadly accepted comparison technique.

Acknowledgments

First and foremost, I would like to sincerely thank my advisor Prof. Carlo Fantozzi. He gave me guidance and direction that was needed to produce this work. Without his generous help, this thesis would not have been possible. Last but not least, I would like to thank Fabiana, my family and friends for their constant encouragement during my studies.

Contents

1 Introduction **1**

2 Image Quality Assessment **5**
2.1 Image Distortions 6
2.2 Reference-based Approaches 9
2.2.1 Full-Reference 9
2.2.2 Reduce-Reference 11
2.3 No-Reference Approach 13

3 No-Reference IQA algorithms **17**
3.1 BRISQUE Algorithm 17
3.1.1 Model 17
3.1.2 Quality prediction and Performances 20
3.2 BLIINDS-2 Algorithm 22
3.2.1 Model 22
3.2.2 Quality prediction and Performances 26

4 BRISQUE Weaknesses and Issues **29**
4.1 Quality Assessment of Downsampled Images 30
4.2 Content Dependency 36
4.3 Sensitivity With Respect to Filter Techniques 38
4.4 Feature Analysis 40
4.4.1 Sensitive features in different models 40

5 BRISQUE Quality assessment improvements **47**
5.1 Feature Selection 48
5.2 BRISQUE and BLIINDS-2 50
5.2.1 Multiple feature sets 51
5.3 Features for Quantized DCT Coefficients 56
5.3.1 The JPEG compression algorithm 56
5.3.2 Feature selection 57

6 Performance improvements **61**
6.1 BRISQUE Performance Analysis 61
6.1.1 Computer performances 61
6.1.2 Mobile device performances 63
6.2 BRISQUE Improvements 64
6.3 BLIINDS-2 Improvements 69

7 Conclusions and Future Work **73**

A Portello Dataset **75**
A.1 Devices 75
A.2 Dataset 76

B Existing Datasets **81**
B.1 LIVE IQA Database 81
B.2 CID2013 database 82
B.3 TID2008 Database 83
B.4 MDID Database 84
B.5 Live In the Wild Database 85

List of Figures

1.1 Number of photos taken worldwide (a) and divided by device category (b) since 2010. Data after 2013 are forecasts. 2

2.1 Reference image (a) and its Gaussian Blur distorted version (b). 7
2.2 Reference image (a) and its JPEG distorted version (b). . . . 7
2.3 Reference image (a) and its JPEG2000 distorted version (b). . 8
2.4 Reference image (a) and its White Noise distorted version (b). 8

3.1 The four pixel directions involved in the computation of the BRISQUE neighbor features 19
3.2 Block structure used in the computation of the BLIINDS-2 features. 23
3.3 Block division used in the computation of the BLIINDS-2 subband-energy features 24
3.4 Block division used in the computation of the BLIINDS-2 orientational features 25

4.1 Quality scores provided by BRISQUE with respect to filtered and downsampled versions of the D7000-01 image. 32
4.2 Quality score comparison of filtered and downsampled versions of images contained in the second Portello subset and D7000-04 image 34
4.3 Quality scores with respect to filtered and downsampled versions of images in the second Portello subset. 35
4.4 Quality scores computed on images of the first Portello subset. Cross points represent quality scores of undownsampled images. 37
4.5 Quality scores computed on five downsampled versions of the D7000-09 image. Different downsampling techniques were applied before score evaluation. 39

4.6 Comparison of feature values computed on the D7000-09 image with respect to the two BRISQUE implementations. Features values are extracted before (a) and after (b) the rescaling process. 43
4.7 Values contained in the matrix 4.3 after the splitting process into positive (a) and negative (b) values. 44

5.1 SROCC values of BRISQUE (black charts) VS the two proposed best models (green charts) across 60 train-test iterations on the LIVE IQA database. 55
5.2 Encoding and decoding processes performed by the JPEG algorithm. 58

6.1 Image of dimension 3286x2432 used in the process of time improvement. It was taken with a Google Nexus 5 in normal conditions. 62
6.2 Screenshots of the Android application deploying the BRISQUE algorithm. 64
6.3 Processes performed by the BRISQUE algorithm in order to extract features from adjacent pixels. 66

A.1 First Portello subset. Images taken with a Nikon D7000 SLR digital camera 77
A.2 Second Portello subset. Images taken with both the Nikon D7000 and other mobile devices. 78
A.3 Third Portello subset. Images taken with the Nikon D7000 SLR. 79

List of Tables

3.1 List of features extracted by the BRISQUE algorithm from the input image (i.e. i = 0) and its downsampled version (i.e. i = 1). 20
3.2 Median SROCC with respect to the BRISQUE algorithm and other IQA algorithms across 1000 train-test iterations on the LIVE IQA database. 21
3.3 Informal complexity analysis of the BRISQUE algorithm and other IQA algorithms on a 1.8-GHz Single-Core PC with 2GB of RAM. 21
3.4 List of features extracted by the BLIINDS-2 algorithm from the input image (i.e. i=0) and its i-th (i=1,2) downsampled versions. 26
3.5 Median SROCC across 1000 train-test iterations of the BLIINDS-2 algorithm (SVM and Probabilistic models) on the LIVE IQA database 27

4.1 Quality score ranking for images contained in the second Portello subset and the D7000-04. Lower is the score value better is the perceived quality. 34
4.2 Quality scores computed on the D7000-09 image with respect to five dimension and downsampling techniques. The original quality score was -8,77023 (image dimension of 4928x3264). . . 39
4.3 Threshold analysis of feature values contained in the Matrix 4.3. 45

5.1 Median SROCC across 60 train-test iterations on the LIVE IQA database with respect to the original BRISQUE feature set and BRISQUE18. 49
5.2 Median SROCC across 60 train-test iterations on the LIVE IQA database. The feature subsets [11-18,29-36], [7-10,25-28], [13-16,29-32] and [17-20,33-36] were removed from the BRISQUE feature set before training and testing. 49

5.3 Median SROCC across 60 train-test iterations on the LIVE IQA database. The feature subsets [3,21], [4,22], [5,23], [6,24], [9,27] and [10,28] were removed from the BRISQUE feature set before training and testing. 49
5.4 Median SROCC across 60 train-test iterations on the LIVE IQA database. The features 23, 24, 26, 27 were removed from the BRISQUE original set before training and testing. 50
5.5 Median SROCC across 60 train-test iterations on the LIVE IQA database. The features 29, 31, 32, 35 and 36 were removed from the BRISQUE model before each training and testing. 50
5.6 Median SROCC across 60 train-test iterations on the LIVE IQA database of BRISQUE18 and the BLIINDS-2 features computed on three different scales. 53
5.7 Median SROCC across 60 train-test iterations on the LIVE IQA database of BRISQUE18 and some BLIINDS-2 features computed on the first and third scales. 53
5.8 Median SROCC across 60 train-test iterations on the LIVE IQA database of BRISQUE18 and the BLIINDS-2 features computed on different DCT block sizes. 54
5.9 SROCC value of a single train-test iteration on the LIVE IQA database. Model trained on BRISQUE18, BLIINDS-2 (features from 1th and 3rd scales) and the parameters resulting from the model fitting of the 63th quantized coefficient. . . . 59

6.1 Original informal complexity analysis of the BRISQUE algorithm. Percentages for the three steps are computed with respect to the overall execution time upon a single input image (once the model trained) 63
6.2 Informal complexity analysis and execution time of the BRISQUE algorithm on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram. Input image in Figure 6.1. 63
6.3 Informal complexity analysis and execution time of the BRISQUE algorithm on a Google Nexus 5 with a 2.3 GHz Quad Core Qualcomm MSM8974 Snapdragon 800 processor and 2 GB of Ram. Input image in Figure 6.1. 65
6.4 Time required by the most consuming "Pairwise Products and AGGD" subprocesses on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram and input image in Figure 6.1 67

6.5 Time required by the most consuming "Pairwise Products and AGGD" subprocesses on a Google Nexus 5 with a 2.3 GHz Quad Core Qualcomm MSM8974 Snapdragon 800 processor and 2 GB of Ram. Input image in Figure 6.1. 67
6.6 Improved time for the "Pairwise Products and AGGD" subprocesses on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram. Input image in Figure 6.1. 68
6.7 Improved time for the "Pairwise Products and AGGD" subprocesses on a Google Nexus 5 with a 2.3 GHz Quad Core Qualcomm MSM8974 Snapdragon 800 processor and 2 GB of Ram. Input image in Figure 6.1. 68
6.8 Final improved BRISQUE execution time on both computer and mobile device 69
6.9 Execution time of the BLIINDS-2 Matlab version on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram. Input image in Figure 6.1. 70
6.10 Execution time of the BLIINDS-2 C++ version on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram. Input image in Figure 6.1. . . . 71
6.11 Execution time of the BLIINDS-2 C++ version without second scale and shape features. Analysis performed on both computer and mobile device. Input image in Figure 6.1. 71

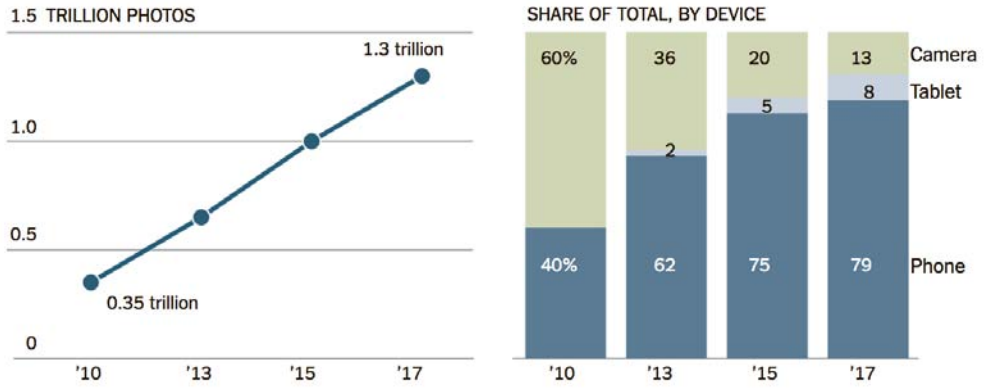
A.1 Devices involved in the Portello dataset building process. . . . 75

Chapter 1

Introduction

The last 20 years have been characterized by increasing improvements in multimedia applications. They take advantage of faster telecommunication channels and high performance computing which enable the possibility of transmitting and acquiring digital content.
The ubiquity of mobile devices such as smartphone and tablets have fueled the interest in digital applications since their introduction in the market. Indeed, one of the most important components of these technological products is the camera sensor. It provides the end-users with a digital image processing pipeline through the acquisition, storing and transmission of pictures.
In order to understand how mobile devices have revolutionized the way humans approach digital photos, a look to marked researches is needed. Back to 2010 the number of photos taken worldwide was 0.35 trillion [11]. 60% of them were captured with a specialized camera device and the remaining 40% with smartphones or tables. Almost seven years later the trend has been inverted and the role of mobile devices in digital photos is leading. In 2017 the number of photographs taken worldwide is expected to reach 1.3 trillion and 80% of them will be acquired with mobile devices [11].
These percentages are in part due to mobile social network applications which provide the ability to upload and share photos over the Internet. It is enough to think that the daily number of photos shared through social applications such as Facebook, WhatsApp and Snapchat is almost 3 billions [16].

With this plenty of images generated every day the need of delivering high quality content is on the rise. During a typical digital image processing pipeline the images are affected by many distortions that can compromise their quality. Despite the improvements in hardware camera components, new software algorithms have been developed in order to address the image quality evaluation. These algorithms belong to the realm of Image Qual-



(a) (b)

Figure 1.1: Number of photos taken worldwide (a) and divided by device category (b) since 2010. Data after 2013 are forecasts.

ity Assessment (IQA), a branch of the digital image processing which aims at evaluating quality as perceived by humans. It plays a key role in any multimedia application in order to deliver high Quality of Service (QoS) to end-users. The first studies on image quality date back to the end of 1940 during the first era of television and this research field is still growing today. In particular the area of No-Reference Image Quality Assessment is has gained increasing interests in the last decade. It is based on the fact that no reference high-quality image is available when evaluating a distorted image. The goal of this thesis is to enhance a state-of-art algorithm and develop a new no-reference model, mostly oriented towards low-resources mobile devices such as smartphone and tablets. A trade-off between execution time and quality evaluation effectiveness will be central in the development of the proposed work. In order to evaluate the results, a real mobile application will be developed and tested.

After this brief introduction, the remainder of this thesis will be organized as follow. In Chapter 2, the basic distortions occurring in digital image will be presented. Afterwords, a review of the techniques and tools in the area of Image Quality Assessment will be provided to the reader. In particular, the key concepts together with some evaluation models will be described.

In Chapter 3, two state-of-art algorithms will be described in depth because involved in the development of the proposed model.

In Chapter 4, issues and defects of one of the two previously described state-of-art algorithms will be analyzed. These will provide suggestions for the improvements that can be incorporated into the enhanced version. In Chapter 5 and Chapter 6 the improvements in terms of quality evaluation effective-

ness and execution time will be developed.
Finally, Chapter 7 will summarize the results obtained in this thesis before concluding with possible future work.

Chapter 2

Image Quality Assessment

Image quality is a characteristic of an image that measures the quality as perceived by humans. Techniques and metrics that allow the automatic assessment of perceived quality are included in the research area of Image Quality Assessment (IQA). This is an active research topic nowadays, because of the recent improvements in digital processing and telecommunication systems.
However, despite the large diffusion of IQA algorithms which is taking place now, this research field has a long history and its origins date back to the 40s. In fact, the early studies aiming to understand how assessing the image quality are related to optical systems, analog televisions and display systems.
Those research efforts have discovered the first factors which need to be involved in the image quality evaluation process, e.g., image definition, contrast, noise, brilliance and geometric distortion.
Since the first studies on quality, the need of involving characteristics from the Human Vision System (HVS) has emerged. Indeed, despite the innovations in past and recent image-based systems, the HVS is the only thing that will remain probably constant across all these revolutions. Nowadays the HVS has a key role in many algorithms which aim at evaluating image quality.
Despite that, modern IQA algorithms have yet to achieve the full comprehension of image quality acts with respect to the human beings. This is in part related to the difficulty in the HSV study which remains largely unknown. Alternative solutions such as Natural-Scene Statistics (NSS) have been recently and successfully applied in order to go beyond these limits.
This chapter firstly introduce the most relevant image distortions, which are responsible for the image quality deterioration. Afterwords, a resume of the most popular approaches used for perceptual quality evaluation will be provided. They are commonly referred as full-reference, reduce-reference and no-reference. For each class some of the most relevant algorithms will be

discussed. Moreover, basic techniques for distortion assessment will be also introduced in the no-reference section.

2.1 Image Distortions

Distortions are related to the concept of image degradation, which is a fundamental aspect for the process of perceived quality assessment. In ideal conditions, digital images are corrupted by a single kind of distortion. However, in other real-world scenarios such as images acquired by mobile devices many distortions occur.
The basic distortions occurring in digital images are well-known and they lead to structural changes that can be described in this manner: loss of spatial details and edge sharpness (blurring), discontinuity in adjacent blocks aspect (blocking), spurious changes near sharp transitions (ringing) or noisy areas (white noise). These distortions (also called artifacts) will be accurately described shortly, along with some examples.

The Blurring is the first distortion here described. It affects images in many steps of the digital image processing pipeline such as acquisition, compression and processing. For example, during the compression process it is caused by truncation of high-frequency components in the frequency domain.
Other reasons that can introduce the blur distortion in images are out-of-focus capturing, relative subjective motion or limitation due to low quality optical system. An comparison between a pristine image and its blurred version is provided in Figure 2.1.

The blocking distortion appears on images through discontinuities among adjacent pixels. These are largely caused by block-based processing techniques which work on small image areas.
The popular JPEG image compression method can be cited as the most relevant source of blocking distortion on digital images. Indeed, when the JPEG algorithm is applied with high compression rates, the removal of high frequency components occurs in the frequency space. This procedure leads to the previously cited discontinuities.
It should be noted that for images acquired with mobile devices this distortion is always present. A comparison between a pristine image and its JPEG compressed version is provided in Figure 2.2.

(a) Reference Image (b) Blurred version

Figure 2.1: Reference image (a) and its Gaussian Blur distorted version (b).

(a) Reference Image (b) JPEG compressed version

Figure 2.2: Reference image (a) and its JPEG distorted version (b).

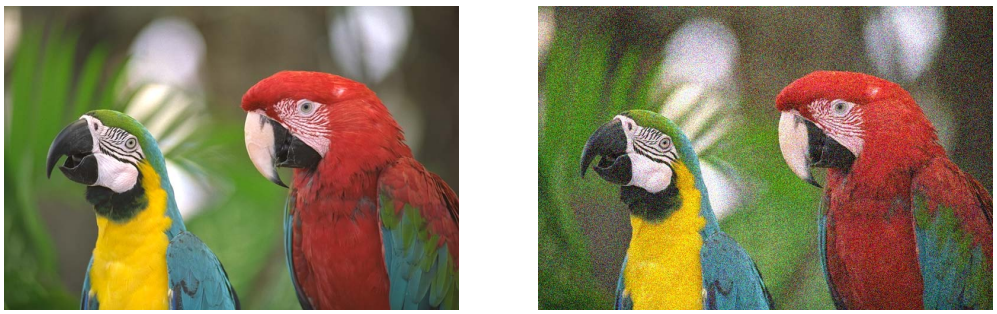
Another source of image degradation is due to the Ringing distortion. It is mostly caused by truncation of high frequency components in the frequency space and occurs in proximity of sharp image edges.
An example of ringing artifact can be caused by high compression rates in the JPEG and JPEG2000 algorithms on images containing textual components. Indeed, these kind of images are full of sharp edges, mainly located in the text borders. For these kind of elements it is difficult to achieve an effective compression.
A comparison between a pristine image and its JPEG2000 distorted version is provided in Figure 2.3.

The last artifacts here described are generally called Noise distortions. Depending on the characteristics through which it occurs, the noise distortion is referred to with different names. For example, the “salt-and-pepper”

(a) Reference Image (b) JPEG2000 compressed version

Figure 2.3: Reference image (a) and its JPEG2000 distorted version (b).

noise arises during the transmission through digital channels. It is characterized by few but very noisy pixels.
Instead, the "quantization noise" occurs during the acquisition process. Each image can be initially modeled as a continuous variable, however, the digitisation process produces a discrete representation which intrinsically contains the noise component.
Finally, in many real-world applications the "White Noise" is the widely used noise representation. It is modeled as an additive Gaussian component and occurs in many phases of the digital image processing pipeline such as acquisition, storing, processing and transmission. A comparison between a pristine image and its White Noise distorted version is provided in Figure 2.4.



(a) Reference Image (b) White Noise distortion

Figure 2.4: Reference image (a) and its White Noise distorted version (b).

2.2 Reference-based Approaches

Depending on the degree of available information from a reference image (i.e. without distortions), image quality assessment algorithms can be categorized in three different techniques: full-reference (FR), reduce-reference (RR) and no-reference (NR).

2.2.1 Full-Reference

Full-reference algorithms take full advantage of the distortion-free pristine image. Indeed, they usually take as input both the reference image and the distorted image, yielding as output an estimation of the perceived quality. Naturally, the quality evaluation is performed with respect to the distorted image.
A basic approach to evaluate the quality of an image with respect to its pristine version is to compute pixelwise differences. Such a methods are based on the simple idea that pixels in the distorted image are a corrupted version of those contained in the reference image. A quality factor is usually provided in order to quantify how many differences hold between these two images. The most common quality factors are obtained from the mean-squared error (MSE) and the peak-signal to noise ratio (PSNR) metrics.
However, many other full-reference approaches to quality assessment were proposed in the past. They involve more complex models in comparison with the previously cited MSE and PSNR.
The first algorithms here described are based on the HVS models. In particular, they take advantage of the primary visual cortex (V1) model as broadly accepted reference model for the HVS. In this cortex, the images are typical processed through a bank of filters. They aim at replicating the linear decomposition applied by the neurons in the V1 cortex. Subsequently, the Contrast Sensitivity Function (CSF) is involved in the model. The purposes of the CSF are twofold. The first is to adjust the effects of the filter banks and the second is to provide a further prefiltering stage. The simulated neural responses can be then processed through a divisive normalization.
The quality score is then obtained as result of the difference between the adjusted responses of the reference image and the adjusted responses of its distorted version.
HVS-based algorithms have also been deployed in order to determine if the distortions are visible or not with respect to a certain threshold. For these kind of applications a certain maturity level has been reached. In particular, they have demonstrated high performances with respect to the evaluation of near-threshold artifacts.

Despite these advances how the HVS acts in presence of more complex distortions and supra-threshold effects remains an open question. Moreover, research efforts will be further needed in order to understand how the cortices beyond the V1 work in the human image processing pipeline.
Another category of full-reference algorithms is based on the concept of image structure. The idea upon which these methods are developed is related to the fact that HVS has evolved to extract structures from the natural environment. As a consequence, an image that contains structural similarities with respect to the reference image can be evaluated as a high-quality image. Instead, if similarities are less marked then the distorted image is evaluated with lower quality.
The methods based on structural similarities have shown high correlation with human judgments since the early studies, date back to 1970s. A widely used and popular full-reference method is the Structural Similarity Index (SSIM) [35] which is developed on the concepts of correlation, luminance and contrast measures. Since its publication (2004) numerous versions of this algorithms were developed, e.g., the same authors have proposed a multi-scale version named MS-SSIM [33]. It applies the SSIM basic concepts to filtered and downsampled versions of the input image. Indeed, it is broadly accepted that distortions have a natural multiscale behavior, i.e., the humans perceive distortions across different image dimensions.
Other full-reference image quality methods were developed using machine learning techniques. They are deeply based on statistical distributions and have demonstrated high correlation with subjective quality scores. In this field natural-scene statistics have earned enough popularity, they are based on the fact that natural images represent only a small portion of the entire image space. Many statistical densities such as Gaussian or Laplacian have been successfully deployed in these algorithms in order to perform the quality evaluation.
In other cases a more direct learning-based approach has been applied through a training step. In [15] for example, a feature mixture is proposed which involves multiple IQA methods.
Machine learning techniques applied to IQA algorithms are related to the processes of feature extraction with respect to the reference image. When these features are extracted from a given image, a regression module is usually deployed. It permits to project the feature values from the feature space into a scalar which represent the image quality score. This approach is also commonly deployed in many other IQA algorithms.

2.2.2 Reduce-Reference

Reduce-Reference (RR) IQA methods aim at predicting image quality having partial information with respect to the reference image. These methods are based on the concept of image features. They represent useful information used in the process of quality evaluation of the distorted image. There are many applications that can take advantage from the IQA methods.
With the spread and improvements in telecommunication channels many multimedia applications have started delivering high-quality images to end-users. These channels are commonly characterized by a limited bandwidth which doesn't permits the transmission of the reference image or video in addition to the distorted versions. At the transmitter, features are computed from the reference content, then they are sent to the receiver which performs the extraction process again. A quality score is yields as a output from the comparison between the transmitted features and extracted features.
The design of a RR IQA algorithm needs to consider the limits of the transmission channels. If the extracted features are a burdensome amount of data to be transmitted over a channel the quality evaluation process can't take place. On the other side, if the feature transmission can easily take place over a low-data-rate channel but the features are perceptual irrelevant then the quality evaluation is unfair.
In literature can be found three common categories of RR methods. They are are based on models of the reference image, HVS and artifacts.
In the first category many successful models have been developed on the reference image structure. The RR general-purpose method proposed in [34] exploits the wavelet domain which provides a framework for localized representation of signals in space and frequency. This method extracts wavelet coefficients from a wavelet subband considering these as features. The goal is to compare the histograms of the extracted features of the reference and distorted images through a distance metric. However, the histogram transmission is a heavy burden for the transmission channel. As a consequence, the histogram is fitted with a generic Gaussian distribution at the sender side, and only the two distribution parameters are transmitted over the channel. The same technique is unfeasible at the receiver side because of the presence of artifacts in the distorted image. In order to select which wavelet subband is involved in the method a steerable pyramid has been used. It process an image as a multi-scale, multi-orientation band-pass filter bank.
The wavelet transform has been successfully deployed in other RR methods. In [13] the authors have proposed an approach built upon the Divisive Normalization Transform (DNT). That is, a two-stage technique which involves a wavelet linear decomposition followed by a normalization process. Roughly

speaking, the normalization process means dividing a wavelet coefficient by a factor which calculates the energy of the wavelet coefficient neighbours with respect to space, scale and orientation. With the aim at computing the normalization factor, the Gaussian mixture model has demonstrated to be an effective statistical-based approach. Finally, the proposed RR algorithm is validate with different datasets showing good results in terms of correlation with human judgements. In physiology that proves that the DNT is a perceptually relevant technique. In fact, the divisive normalization step has proved to be revealing with regards to some aspects of the visual cortex behaviour.

As in other cases, the popular SSIM FR method has been used in order to develop a enhanced version [20] of the previous work. This take advantage of the structural difference analysis across different wavelet subbands which is a principle in the SSIM method.

Even HVS-based methods have demonstrated to be effective in the case of quality evaluation when only partial informations about the reference image are available. In [7] the authors have proposed a method which is deeply based on descriptions from physiology literature. This take advantage of two visual pathways starting in the eyes and ending in the V4 and V5 visual cortices. In particular, the method has been developed on the perceptual concept of image annoyance. A two-phase process is deployed for the image quality evaluation. The first phase applies the developed method extracting the perceptual features. Instead, the second phase involves a comparison process between the features extracted from the distorted and reference images. Finally, at the end of the comparison process a quality score is yield as output.

The process of selecting an effective feature such that correlates well with respect to humans judgements is a challenge. In [6] a guideline scheme to select HVS-based features in the RR framework has been presented. It works on a three components space of a luminance transformed RGB input image. In the first stage this image is processed in order to obtain a perceptual representation of the HVS behaviour. Then characteristic points are extracted from different image subbands. These points are located on concentric ellipses centered on the image center and are used as information sources for perceptual features. The results shown good correlation with subjectives scores.

2.3 No-Reference Approach

No-reference Image Quality Assessment (NR-IQA) is nowadays the most active research field if compared with the full-reference and reduce-reference approaches. The last decade has been characterized by deep improvements in telecommunication systems and large diffusion of multimedia services. Along with these, the interests in no-reference algorithms have rapidly increased in order to enhance the quality of delivered services.
In literature three main no-reference quality assessment techniques can be found: No-Reference Pixel-based (NR-P) methods, No-Reference Bitstream-based methods (NR-B) and hybrid methods of these two techniques. In the remainder of this section, only the first two methods will be described. However, an extensive study comprising all these methods is provided in [25].
The NR-B methods are mainly based on the extraction of relevant features from the coded bitstream data. This approach can lead to several advantages such as reducing the computational complexity of processing the full data. A large amount of information are directly provided by the coded bitstream, which are directly employable for perceptual quality assessment.
However, the NR-B methods highly depend on the used bitstream coding technique. Indeed, different coding techniques entail different levels of data access. Two main models derive from this concept: the Parametric Models (PM) and Bitstream Layer Models (BLM). Since these two techniques deal with video streaming rather than images, to be coherent with the purpose of this thesis it has been decided to focus on the NR-P-based method.
The NR-P-based methods address the quality assessment process in terms of single or multiple artifacts. However, in presence of single distortions the no-reference quality evaluation is an easier task. In order to explain how no-reference methods act on images affected by a single distortions, some example will be presented shortly. In particular, the basic no-reference methods for evaluation and assessment of distortions in section 2.1 will be discussed.
NR-P-based algorithms perform the Blurring artifact evaluation using a two-process technique. It starts with the measurement of pixel spreads along image edges, which can be performed applying some well-know digital image operators such as Sobel or Canny. The second and last process aims at evaluating the edge distortions. In some cases, this process involves characteristics from the human vision system such as edge perception. Finally, at the end of these two processes, the assessment of blurring intensity can be delivered.
The Blocking artifact is commonly evaluated by no-reference methods using a three-process technique. Before describing each of them, it is important to emphasise that some of these processes are mutually independent. As a consequence, the overall evaluation process can take advantage of a parallel

execution in order to achieve the final goal.
The first process takes as input the distorted image and yields as output a set of small images (also called patches) which are composed by subregions of the input image. Afterwords, the detection process of blocking artifact on vertical and horizontal directions (two of three processes) is performed. If the result is positive, then an intensity estimation process is carry out in order to understand if the distortion is enough strong to be perceived by a human observer. Intensities of masking effects like luminance and texture can be deployed to address this estimation. Finally, after these three processes the blocking artifact can be evaluated. Many solutions starting from this basic scheme were exploited which involve detectors based on DCT or radial basis functions.
The basic Ringing artifact evaluation technique will be now described. The first process requires to map the image edges to a specific structure. In particular this process involves the fundamental phase of edge detection, which is performed using operators such as Sobel (already deployed in the blocking evaluation technique). The second process aims at extracting relevant features, in order to classify image areas based on distortion intensity. Similarly to the previous described technique, the characteristics of the human vision system (such as masking effects) can be involved to understand if ringing artifact is really perceived.
Finally, the noise distortion estimation process will be now described. A simplification assumes this artifact to be white noise, i.e. it is modeled as Independent Identically Distributed (IID) zero-mean Gaussian signal. The noise estimation methods can be categorized in smoothing-based or block-based. Methods included in the first category try to evaluate noise using a smoothed version of the image. Instead, noise variance among low-contrast (i.e. homogeneous) adjacent image blocks is the discriminant feature in block-based approaches. Strong noise signal affecting homogeneous blocks is broadly accepted to be highly discriminant for human visual perception. Finally, a feature extraction process is commonly used in block-based methods. It is subsequently followed by threshold techniques in order to evaluate noise intensity.

The previously cited evaluation techniques are based on the idea that no more than one distortion at a time affects images. However, this hypothesis is not a good representation of real-world digital images, such as those acquired by mobile devices. Indeed, the images taken with mobile devices usually involve multiple artifacts. These are introduced by low-quality sensors, environmental conditions or artifacts related to human behaviour (e.g. movements during the acquisition).

The use of multimedia applications is nowadays widespread. It is proved that these applications usually distort digital contents with many artifacts. In order to address this problem the research has moved towards the development of new algorithms, acting on images affected by multiple artifacts. They represent the key to deliver an higher quality of service (QoS) to the end-user.
Evaluating the image perceptual quality in presence of multiple artifacts is a tough task. All the previously described methods are based on handcrafted features which is an acting method inherited by the multiple-artifact-based algorithms. However, the deployment of such a features can't be effective in presence of multiple artifacts because each feature is only predictive for a specific kind of distortion. New algorithms then commonly deploy generics features which are predictive for different artifacts.
A feature category that has been extensively adopted in the multiple-artifact-based algorithms acts on the idea of Natural Scene Statistics (NSS). This take advantage of the fact that natural images represent a small set of the entire image space. They not only represent natural images the HVS has evolved to process, but they also contain unique statistical characteristics which are proven to be modified in presence of multiple artifacts. Many probabilistic models have been proposed in order to extract relevant statistical features. They basically perform a fitting process which yields as output predictive parameters such as scale or shape.
Statistical models broadly deployed in recent years are based on popular distributions such as Gaussian distribution ([22]), Weibull distribution ([9]), Cauchy distribution ([32]), Generalized Gaussian Distribution ([3],[26]) and Asymmetric Generalized Gaussian Distribution ([3]). The last two have been largely used by multiple-artifact-based algorithms because of their property of encompassing a large set of distortions.
Many NR-P-based methods have been proposed upon the NSS framework, they take advantage of features from both time and frequency domain. However, those methods are not suitable for images containing non-natural elements such as text or artificial graphics. The quality evaluation on such a kind of images is better achieved through learning-based features. Methods developed on those feature usually require a large codebook to obtain good performance in terms of correlation with quality perception. As a consequence, their are computationally expensive and not exploitable in real-world system equipped with low resources. In order to take advantage of the learning-based and handcrafted statistical-based feature properties, new algorithms have been recently proposed such as [37].

Chapter 3

No-Reference IQA algorithms

In the previous chapter many no-reference techniques for both single and multiple artifacts have been presented. The focus of this chapter will be on two state-of-art no-Reference algorithms upon which the entire thesis will be developed. Their working principles and performances will be described in order to provide a reference chapter for future comparisons.

3.1 BRISQUE Algorithm

Spatial-based referenceless algorithms are relatively under exploited for image quality evaluation. In this section the BRISQUE [3] algorithm will be deeply described. It is a simple spatial-based referenceless algorithm which has demonstrated high performances in terms of quality evaluation. Moreover, its low execution time is a key advantage in order to deploy it on low embedded systems such as mobile devices.

3.1.1 Model

The BRISQUE prediction model is developed on the concept of Natural Scene Statistics for the image's spatial domain. In particular, only the image luminance is processed by the algorithm and no consideration is made on chrominance.
Having as input a grayscale image of dimension MxN, the algorithm computes a formula of transformed luminance. It is called local normalized luminance and is explained by 3.1.

$$\hat{I}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + C}, i \in 1, 2...M, j \in 1, 2...N \tag{3.1}$$

A two-phase process is needed in order to compute the values deriving from the transformed luminance. The first phase involves the local-mean computation which is described by the Formula 3.2 whereas the second phase (called divisive normalization) is computed using the Formula 3.3. In both cases the $w_{\mathrm{k,l}}$ is a circularly-symmetric Gaussian weighting function, with $K = L = 3$.

$$\mu(i,j) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{\mathrm{k,l}} I_{\mathrm{k,l}}(i,j) \tag{3.2}$$

$$\sigma(i,j) = \sqrt{\sum_{k=-K}^{K} \sum_{l=-L}^{L} (I_{\mathrm{k,l}}(i,j) - \mu(i,j))^2} \tag{3.3}$$

The values $\hat{I}(i,j)$ are computed for the entire image and named Mean Subtracted Contrast Normalized (MSCN) coefficients.
In [21] the MSCN coefficients are observed to have a Gaussian distribution for natural image. This property is exploited by BRISQUE in order to model the contrast-gain masking which is an early HSV characteristic.
The hypothesis upon which the BRISQUE algorithm was developed is related to the statistics properties of the MSCN coefficients. They are supposed to change in presence of image artifacts which is a prerequisite to become predictive features. Indeed, in [21] is shown that when a reference image is affected by an artifact its MSCN coefficients exhibit a non-gaussian behavior.
In order to capture these statistical changes, BRISQUE adopt a Generalized Gaussian Distribution (GGD). This distribution has the property to encompass a large set of statistics such as Gaussian and Laplacian. The shape and scale parameters are then extracted from the GGD fitting on the MSCN coefficient histograms. The shape and scale, generated by the fitting process, are the first two BRISQUE features.
Another main contribute to the algorithm effectiveness dealing with neighbor pixels. The basic idea is that signs of adjacent pixels exhibit a defined structural behaviour which changes in presence of distortions.
Four transformed luminance-based coefficients are computed in order to take advantage of the neighbor pixel properties. They are related to four different directions with respect to a central pixel, as explained in Figure 3.1.

From this scheme a formula for the horizontal (3.4), vertical (3.5), main diagonal (3.6) and secondary diagonal (3.7) directions is obtained.

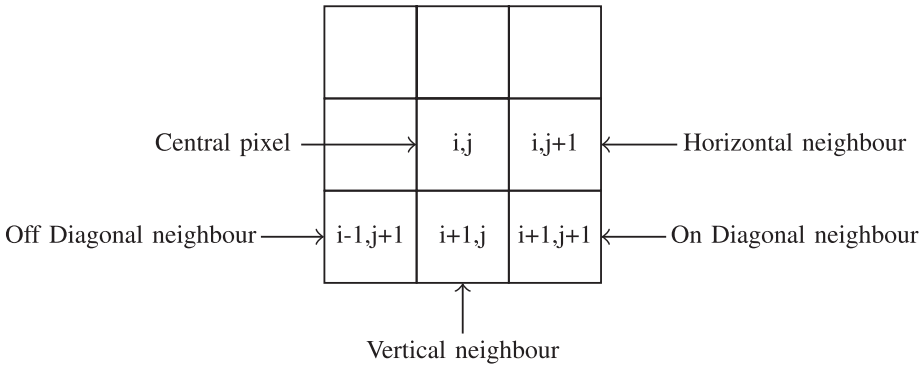


Figure 3.1: The four pixel directions involved in the computation of the BRISQUE neighbor features

$$H(i,j) = \quad \hat{I}(i,j) \times \hat{I}(i,j+1) \tag{3.4}$$

$$V(i,j) = \quad \hat{I}(i,j) \times \hat{I}(i+1,j) \tag{3.5}$$

$$D1(i,j) = \quad \hat{I}(i,j) \times \hat{I}(i+1,j+1) \tag{3.6}$$

$$D2(i,j) = \quad \hat{I}(i,j) \times \hat{I}(i+1,j-1) \tag{3.7}$$

With the aim of modeling structural changes in empirical distributions of adjacent normalized pixels, the Asymmetric Generalized Gaussian Distribution (AGGD) was deployed. It is a generalization of the previously cited GGD which introduces two more parameters ($\sigma^2{}_{\mathrm{l}}$,$\sigma^2{}_{\mathrm{r}}$). These control the left and right distribution tails spread which are supposed to change in presence of artifacts. The AGGD converges to a GGD when the left and right tails are equals.
Adjacent normalized pixels are fitted with the AGGD and four parameters are yields as output for each directions (η, ν, $\sigma^2{}_{\mathrm{l}}$, $\sigma^2{}_{\mathrm{r}}$), respectively shape, mean, left-variance and right-variance. A total of eighteen parameters are obtained from the first BRISQUE iteration on the input image.
The algorithm then recomputes all the features on a filtered and downsampled version (by a factor 2) of the input image. It is broadly accepted that artifacts are transferred across scales and humans are sensitive to these.
Finally, from an input image 36 features are extracted by BRISQUE. They will be used in order to develop a prediction model described in the next

section. The features list is fully reported in Table 3.1.

Feature ID	Feature Description	Computational Procedure
f_{1+18i}-f_{2+18i}	Shape and variance	Fit GGD to MSCN coefficients
f_{3+18i}-f_{6+18i}	Shape, mean, left variance, right variance	Fit AGGD to H coefficients
f_{7+18i}-f_{10+18i}	Shape, mean, left variance, right variance	Fit AGGD to V coefficients
f_{11+18i}-f_{14+18i}	Shape, mean, left variance, right variance	Fit AGGD to D1 coefficients
f_{15+18i}-f_{18+18i}	Shape, mean, left variance, right variance	Fit AGGD to D2 coefficients

Table 3.1: List of features extracted by the BRISQUE algorithm from the input image (i.e. i = 0) and its downsampled version (i.e. i = 1).

3.1.2 Quality prediction and Performances

In order to produce quality scores from a set of image features, a regression module has been deployed. The regression package of the popular LIBSVM [8] software library has been used for this task. In the training phase, subjective quality scores and image features are associated together. Then on these associations the training process is launched. A mathematical model is yields as output which permits to transform a set of features in the feature space to a scalar quality score.
In order to prove correlation performances with human judgements and performing the training phase, the widely used LIVE IQA [27] image database has been involved. As described in Chapter 2, it contains 779 images affected by five kind of distortions: JPEG, JP2K, Gaussian Blur (BLUR), Rayleigh Fast Fading Channel Simulation (FF) and White Noise (WN). Moreover, the DMOS of human subjectives with regard to database images are provided.
In order to obtain an unbiased performance evaluation, multiple training and testing processes have been performed. For each evaluation process the LIVE database has been divided in 80% (training) and 20% (testing) not overlapping randomly chosen subsets. This random training-test procedure has been repeated 1000 times and the median Pearson's linear coefficients (LCC) and Spearman Rank Order Correlation Coefficient (SROCC) produced as output.

These metrics are obtained with respect to the comparisons of predictive scores and subjectives DMOS. The resulted SROCC scores in comparison with other popular IQA algorithms are proposed in Table 3.2.
The BRISQUE model has been trained upon the LIVE database which contains images affected by 5 kind of artifacts. Despite that fact good results have been shown by this model with respect to the Challenge Live In the Wild database as described in [37]. This database contains images naturally distorted, i.e. the distortion are not generated artificially such as those of the LIVE IQA database.

	JP2K	JPEG	WN	Blur	FF	All
PSNR	0.8646	0.8831	0.9410	0.7515	0.8736	0.8636
DIIVINE [17]	0.9123	0.9208	0.9818	0.9373	0.8694	0.9250
SSIM	0.9389	0.9466	0.9635	0.9046	0.9393	0.9129
MS-SSIM	0.9627	0.9785	0.9773	0.9542	0.9386	0.9535
BRISQUE	0.9139	0.9647	0.9786	0.9511	0.8768	0.9395

Table 3.2: Median SROCC with respect to the BRISQUE algorithm and other IQA algorithms across 1000 train-test iterations on the LIVE IQA database.

Temporal performances have been stated to be very promising for runtime applications. The execution time has been acquired with an unoptimized Matlab Release [2] developed by the authors. That software implementation permits to evaluate an image of dimension 768x512 in about one second, leading BRISQUE to be one of the fastest referenceless IQA algorithms. Table 3.3 lists temporal comparisons among BRISQUE and other popular IQA algorithms.

Algorithm	Time [s]
PSNR	0.05
DIIVINE [17]	149
BLIINDS-II	70
BRISQUE	1

Table 3.3: Informal complexity analysis of the BRISQUE algorithm and other IQA algorithms on a 1.8-GHz Single-Core PC with 2GB of RAM.

3.2 BLIINDS-2 Algorithm

The Discrete Cosine Transform (DCT) is a largely used time-to-frequency transformation for two-dimensional signals. It has enjoyed great popularity in Digital Image processing with regards to different applications such as the JPEG compression standard.
The algorithm here described is based on DCT transformation and is named BLind Image Integrity Notator using DCT Statistics (BLIINDS-2) [23]. It was developed on the ideas of his predecessor BLIINDS-1 which is based on both the DCT transformation and the properties of frequency domain. Nevertheless, the set of features extracted by BLIINDS-2 are different from those of BLIINDS-1 and they take advantage of a novel statistical model based on natural scene statistics.

3.2.1 Model

As introduced before the algorithm's model is based on the frequency image domain. Given a grayscale image as input, BLIINDS-2 extracts two-dimensional DCT coefficients through blocks (called patches) of dimension 3x3. A patch shares a boundary pixel with the adjacent patches resulting in a final dimension of 5x5 as shown in Figure 3.2. This structural choice comes from the fact that HVS applies a local spatial frequency decomposition of the image. Patches are then used to retrieve statistics relevant for natural scene images.
Statistical models are involved with respect to the entire patch and their subparts. On the remainder of this section the extracted statistics will be deeply described along with their conceptual meaning.

The first statistic feature is the shape parameter provided by fitting an entire DCT patch with a Generalized Gaussian Distrubution (GGD) which has been already exploited in the previous section by the BRISQUE algorithm. With the exception of the DC coefficient (located in the top-left corner of a patch), all the remainder patch coefficients are involved in the fitting process. The DC coefficient has been proved to carry out no structural information about an image patch, resulting useless for the quality evaluation task.

The second statistical feature is called Frequency Variation Coefficient (FVC). It is derived by the random variable X which models the histogram of DCT coefficient magnitudes. Mean and standard deviation of X are involved in order to compute the FCV feature as described by the Formula 3.8.

DC	C_{12}	C_{13}	C_{14}	C_{15}
C_{21}	C_{22}	C_{23}	C_{24}	C_{25}
C_{31}	C_{32}	C_{33}	C_{34}	C_{35}
C_{41}	C_{42}	C_{43}	C_{44}	C_{45}
C_{51}	C_{52}	C_{53}	C_{54}	C_{55}

Figure 3.2: Block structure used in the computation of the BLIINDS-2 features.

$$\zeta = \frac{\sigma_{|\mathrm{X}|}}{\mu_{|\mathrm{X}|}} \tag{3.8}$$

A mathematical proof is provided by the authors which states that the FVC feature can be approximated using together the gamma function and the previous first shape feature. The mathematical procedure leads to the formula in 3.9.

$$\zeta = \sqrt{\frac{\Gamma(1/\gamma)\Gamma(3/\gamma)}{\Gamma^2(2/\gamma)} - 1} \tag{3.9}$$

The third statistical feature is the Energy Subband Ratio Measure (ESRM), it involves DCT patch coefficients through non-overlapping bands at different frequencies. In fact, each patch is divided into three bands following the main diagonal direction, i.e. from the top-left to the bottom-right coefficient as described in Figure 3.3.

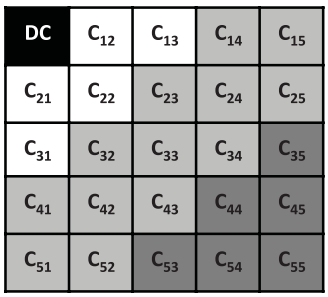


Figure 3.3: Block division used in the computation of the BLIINDS-2 subband-energy features

For each band the energy is defined as the variance of DCT band coefficients. Resulting in three band energy coefficients per patch namely E_i (i = 1,2,3), they are used to calculate the formula 3.1.

$$R_n = \frac{|E_n - \frac{1}{n-1}\sum_{j<n} E_j|}{E_n + \frac{1}{n-1}\sum_{j<n} E_j}, n = 1, 2 \tag{3.10}$$

Large values for R_1 and R_2 in formula 3.1 means that the current patch has pronounced differences between the current band and the previous, that are one band for R_1 and two bands for R_2. The final ESRM statistic parameter is computed as the mean value of R_1 and R_2.

The fourth statistical feature is based on the concept of human vision sensitivity to directional changes in images. Generally speaking their presence is stronger when the image is highly distorted. In order to capture information related to orientation BLIINDS-2 uses the Orientation Model-Based Feature (OMBF).
The OMBF feature is obtained through the patch division in three non-overlapping regions. These regions follow the main-diagonal direction from the top-left corner to the bottom-right corner. The DC coefficients is not involved in the first region because of the same reason explained for the previously features.
In order to compute the orientational FCV coefficient, each region is fitted with a generalized Gaussian distribution. Then, the shape parameter is ap-

plied within the Formula 3.9.
The variance across FCV orientation coefficients for a patch is considered as a features for the BLIINDS-2 model.

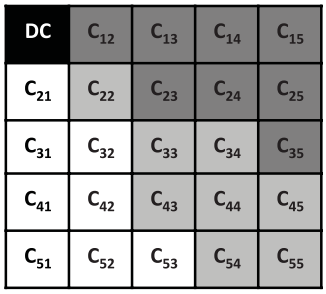


Figure 3.4: Block division used in the computation of the BLIINDS-2 orientational features

The four statistical features involved in the BLIINDS-2 have been yet described. Each of them is computed for an image patch, leading in total to a large numbers of features.
With the aim at reducing this amount of data, the pooling technique has been used. The features are clustered by type, then the 10th percentile and the 100th percentile (sample mean) are computed. This approach permits to obtain 8 features in total per image.

In the previous sections has been introduced the behavioural aspect of distortions at different scales. This hypothesis has lead to the inclusion of statistical features within the BRISQUE predictive model. In particular these features have been computed with respect to a filtered and downsampled version of the input image.
The same concept has been also deployed by the BLIINDS-2 algorithm which involve an additional image scale. In fact, the algorithm uses two filtered and downsampled versions of the input image in order to compute the same features at different scales. As a consequence, 24 features in total are computed for the input image. The Table 3.4 lists all the features used by the BLIINDS-2 algorithm.

Feature ID	Feature Description	Computation Procedure
f_{1+8i}-f_{2+8i}	The lowest 10th percentile of the shape parameter, the mean of the shape parameter	Fit GGD to DCT patch
f_{3+8i}-f_{4+8i}	The highest 10th percentile of the coefficient of frequency variation, the mean of the coefficient of frequency variation	Compute frequency variation coefficient from shape parameter
f_{5+8i}-f_{6+8i}	the highest 10th percentile of the energy subband ratio measure R_n, the mean of the energy subband ratio measure	Compute mean value of R_2 and R_3 in formula 3.1
f_{7+8i}-f_{8+8i}	The highest 10th percentile of the variance of FVC across the three orientations, the mean of the orientation feature	Compute frequency variation coefficient variance across all patch orientations

Table 3.4: List of features extracted by the BLIINDS-2 algorithm from the input image (i.e. i=0) and its i-th (i=1,2) downsampled versions.

3.2.2 Quality prediction and Performances

With the aim at evaluating the BLIINDS-2 performances, a simple probabilistic model inherited from the algorithm's predecessor has been adopted. Given a feature set $X_i = [x_1, ...x_{24}]$ extracted from a image and its subjective DMOS, the model's goal is to fit the distribution $(X_i, DMOS)$. This goal is achieved through a multivariate GGD model. A training phase on a subset of the LIVE IQA database is performed and the model parameters (mean and covariance) are then yields as output of the fitting process upon the test set empirical data.
The model aims not only to fit the sample data as well as possible, but to achieve high correlations between predicted and subjective DMOS.
The algorithm quality assessment performances are evaluated in the same manner described for the previous algorithm. Multiples training and testing iterations are performed in order to obtain an unbiased median correlation

coefficient. This is computed with respect to the correlation between predictive scores and subjective judgements.
In Table 3.5 are shown the median Spearman correlation coefficients (SROCC) across 1000 training (80%) and testing (20%) evaluations on the LIVE IQA database. Besides, a comparison between the described probabilistic model and its SVM counterpart is provided. It should be noted that the latter outperforms the probabilistic model on the overall assessment but not for all the distortion categories.

LIVE subset	BLIINDS-2 (SVM)	BLIINDS-2 (Probabilistic)
JPEG2000	0.9285	0.9506
JPEG	0.9422	0.9419
White Noise	0.9691	0.9783
GBlur	0.9231	0.9435
Fast Fading	0.8893	0.8622
ALL	0.9306	0.9202

Table 3.5: Median SROCC across 1000 train-test iterations of the BLIINDS-2 algorithm (SVM and Probabilistic models) on the LIVE IQA database

Temporal performances for BLIINDS-2 are listed in Table 3.3 of previous section. The algorithm doesn't show a reasonable run-time performance for real-time applications. By the way a lot of time is wasted by DCT computation on image patches and the original implementation is not optimized. There is then still room for improvement using efficient DCT implementations.

Chapter 4

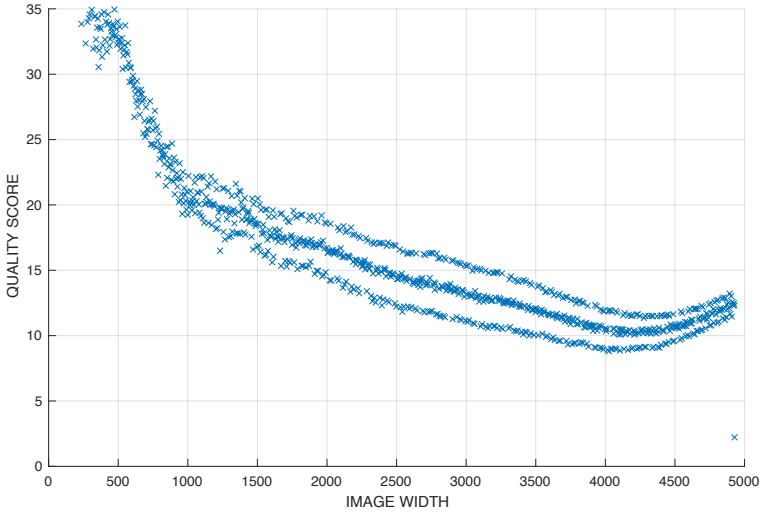
BRISQUE Weaknesses and Issues

Despite the properties of high execution speed and high correlation with human judgements, the BRISQUE algorithm is not free of defects. The goal of this chapter is to point out limits and issues of this algorithm.
In order to achieve this goal, a new photographic dataset named "Portello dataset" will be used. It was exclusively developed for the purposes of this thesis and fully described in Appendix A.
This chapter is organized as follow. In the first section, a subset of images contained in the "Portello dataset" will be evaluated by the BRISQUE algorithm. In particular, each image will go through a process of multiple downsampling which aims at discovering if any relationship exists among the quality scores of an image and its downsampled versions.
In the second section, the quality score dependency on the image content will be analyzed. The strength of the BRISQUE algorithm will be proved on a subset of images taken with a single reference device. These images are characterized by small variations introduced by the photographer with respect to translation and rotation .
In the third section, a unique quality score behaviour emerged in the previous sections will be pointed out. In particular, it shows the algorithm sensibility with respect to filter techniques, which are deployed in many phases of the digital image processing pipeline.
Finally, the last section will take advantage of the two proposed BRISQUE implementations. They will be used to compare feature values on the same input image. Indeed, as will be seen, a subset of features demonstrates high variability with respect to the two implementations. This fact will be exploited in chapter 5 with the aim of strengthening the overall quality evaluation performance.

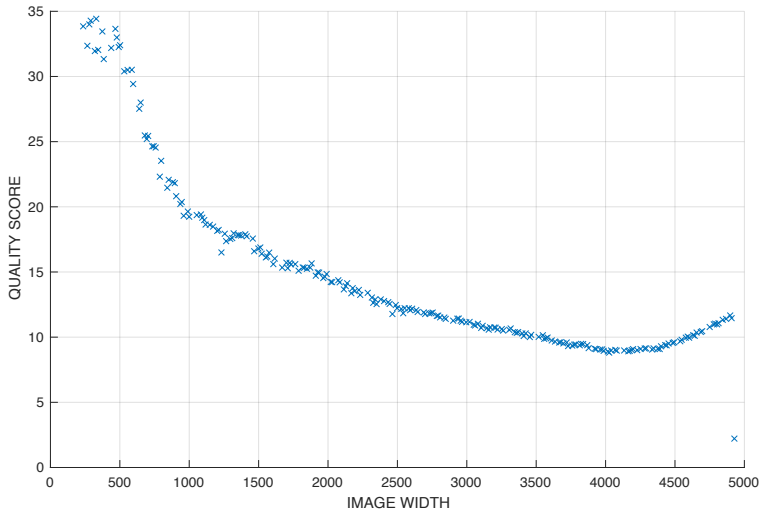
4.1 Quality Assessment of Downsampled Images

In order to gain information about the process of quality evaluation performed by the BRISQUE algorithm, a series of experiments based on quality scores of images contained in the second Portello subset (A.2) will be presented shortly.
It is important to point out that two implementations of the BRISQUE algorithm have been proposed by the authors. These were developed in Matlab and C++ [1]. However, in this section only the C++ version will be considered.
The first experiment aims at revealing how the BRISQUE algorithm evaluate the perceived quality at different image scales. If exists a relationship between the quality scores of the original image and its downsampled versions, then a mobile application could take advantage of this. Obviously, the downsampled versions are computationally less expensive with respect to the original image.
With the aim at evaluation the perceived quality of an image at different scales, the original BRISQUE implementation was integrated with an additional feature. It permits to obtain the quality scores of multiple downsampled versions of the same image on-the-fly. The dowsampling process is performed through the standard bicubic interpolation method provided by the OpenCV [5] computer vision library. This interpolation method was chosen because already involved in the original BRISQUE code for the computation of the last eighteen BRISQUE features.
In order to obtain as much score evaluations as possible, the experiments were performed on one-thousand downsampled versions of the input image. A downsampling step of 0.001 was used. It means a 0.1% reduction withe respect to width and height per iteration. However, this downsampling approach has involved odd image dimensions (width, height or both).
It was discovered that odd image sizes always alter the BRISQUE quality evaluation process. An example is provided in Figure 4.1a which represent one-thousand quality score evaluations of the Photograph A.1 a. It should be appreciated how three different score trends are shown. The upper line plot corresponds to images with both width and height odd. Instead, the middle line plot is related to images with odd width or height, not both occurring together.
Keeping in mind that decreasing values for the quality score means higher perceived quality, it can be stated that the overall behaviour of odd image sizes decrease the image quality. Those values were then discarded during

the downsampling process, resulting in a smaller score set per image.
Behind this action is the consideration that commercial digital cameras have photosensors developed with grid arrays with even number of rows and columns. The Figure 4.1a was then modified with the proposed changes, which can be appreciated in Figure 4.1b.



(a) One-thousand downsampled versions



(b) Downsampled versions with even width and height

Figure 4.1: Quality scores provided by BRISQUE with respect to filtered and downsampled versions of the D7000-01 image.

According to the problems introduced before, the quality scores of images contained in the second Portello subset were computed with respect to even dimensions. They are shown in Figure 4.3.

Despite the photographs were taken with different devices, the score trends appear very similar. Indeed, they are characterized by a gap between the quality scores of the original image and its first downsampled version. Subsequently, the trends decrease before starting to increase again.
The decreasing score phase, which characterizes image dimension in the range [1700,2500) suggests that spurious details or artifacts have been removed in the downsampling process. However, this statement can be valid up to a certain image dimension (approximately 1700). After that decreasing the image dimension means information loss and worse perceived quality.
It should be note that the trend in Figure 4.3 a manifests a slightly different behaviour with respect to the others. It decreases until width values around 1400 before starting to increase again. This can be justified by an aggressive impact of the noise removal algorithm occurring in Photograph A.2 (b) on which the score trend is computed.
The experiments performed so far had the goal to find some relationship between quality scores at different scales which unfortunately not occurred. However, an issue for the BRISQUE algorithm has been pointed out. Indeed, each quality score trend derived from the second Portello subset contains a marked gap betweem the score of the original image and its first downsample version. It is obvious that the BRISQUE algorithm is sensitive to any action performed in the downsampling process. This is an algorithm weakness because the first downsampled image version is almost equal to the original image. From a perceptual point of view, the quality of the two images is the same.
Despite this negative but important characteristic, the algorithm seems to evaluate images in the right way. Indeed, it should be natural that images photographed with a high-quality devices are evaluated better then those photographed with a low-quality devices. The higher hardware quality in image sensors lead to a better perceived quality by a human observer.
This fact is confirmed by the BRISQUE algorithm with regard to the second Portello subset. The score trends were compared and plotted in Figure 4.2. It can be appreciated how the D7000-04 overcomes the other quality score trends. This is natural because the photograph was taken with the high-quality reference device. Moreover, from the chart it can be seen how the gap is evident from the original images and their first downsapled versions. The quality ranking results for the original images are listed in Table 4.1.

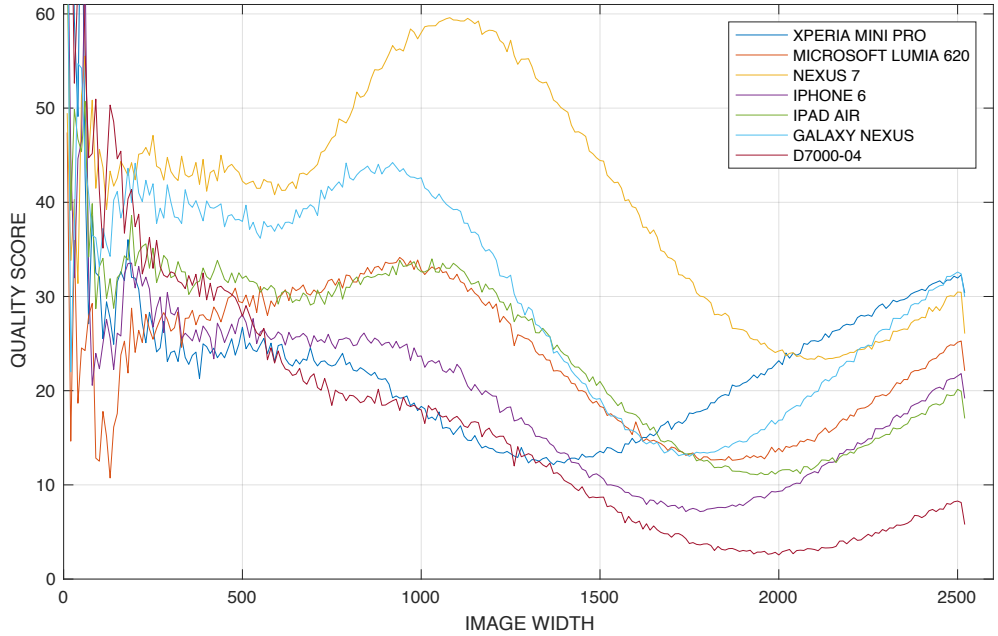
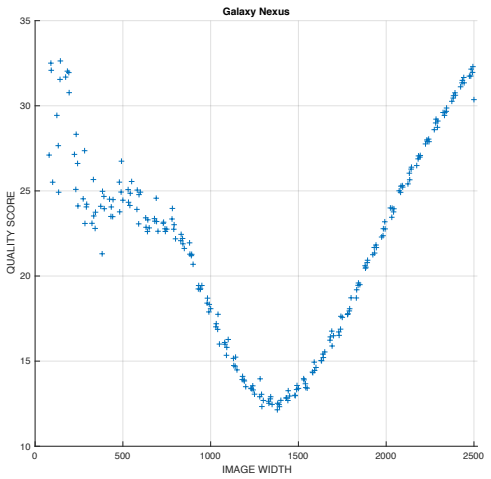


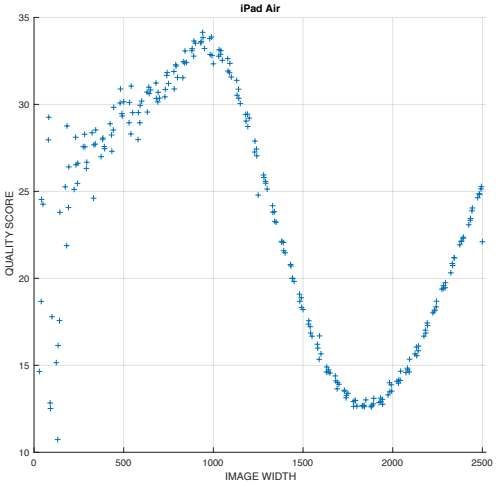
Figure 4.2: Quality score comparison of filtered and downsampled versions of images contained in the second Portello subset and D7000-04 image

device model	minimum score [%]
D7000	5.7909
iPad Air 2	17.0552
iPhone 6	19.1796
Microsoft Lumia 620	22.1045
Nexus 7 2013	26.0624
Galaxy Nexus	29.9447
Xperia Mini Pro	30.3615

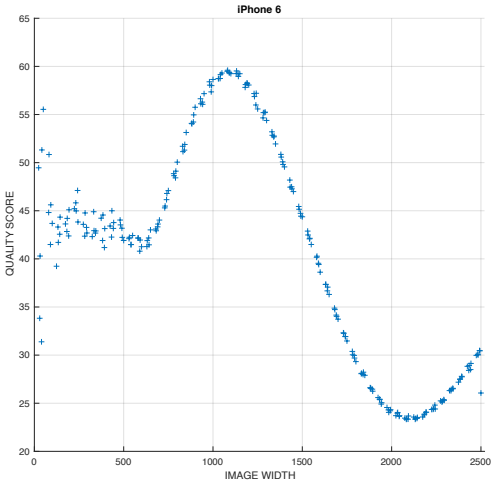
Table 4.1: Quality score ranking for images contained in the second Portello subset and the D7000-04. Lower is the score value better is the perceived quality.



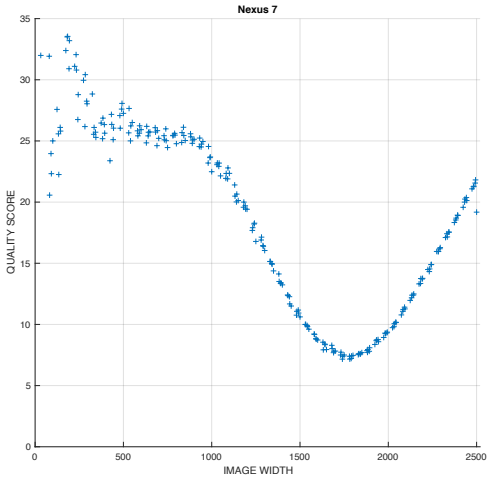
(a) Galaxy Nexus



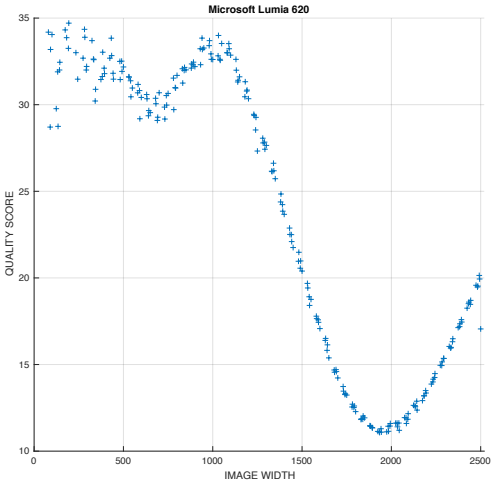
(b) iPad Air



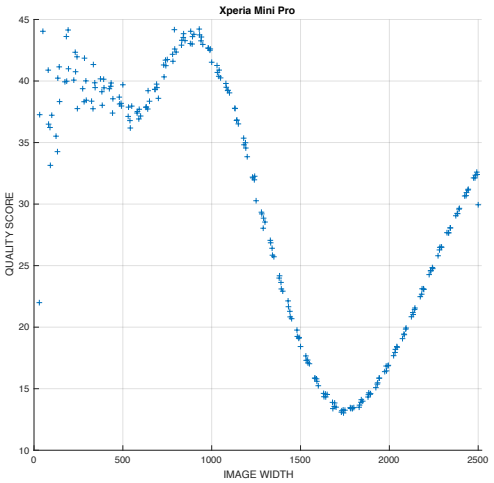
(c) iPhone 6



(d) Nexus 7



(e) Microsoft Lumia 620



(f) Xperia Mini Pro

Figure 4.3: Quality scores with respect to filtered and downsampled versions of images in the second Portello subset.

4.2 Content Dependency

The previous section is started with the quality score trend analysis of images at different scales. Despite this process has permitted to discover a BRISQUE weakness, it should be further employed in order to analyze how the score changes with respect to variations in the image content.
Measuring the BRISQUE score strength to slightly variation in image content is the goal of this section. In order to achieve it, the first Portello subset will be involved in the score trend analysis. According to the first section of this chapter, this subset contains photographs taken with the reference device (Nikon D7000) which differ from each other for small (sometimes imperceptible) variations.
However, these variations were deliberately introduced by the end-user with the aim at evaluating the quality score in presence of translational and rotational changes. Thinking about the human behaviours, this is a typical situation that can arise when multiple users photographing a scene with their devices.
In Figure 4.4 the overlapping results of quality evaluations on the entire first Portello subset are shown in order to provide an handle for the considerations which will be described shortly. The figure x-axis is characterized by a restricted width range because the necessity to provide the focus on the first score values.
The images A.1 (b and c) which are affected by vertical variations shown different quality score trends. In particular, meanwhile the quality score trend for the figure A.1 (c) is near to that of the group-reference image, the same cannot be said for the trend related to the figure A.1 (b).
The latter case (plotted using the orange line) highly differs from the reference image trend (dark blue line). Probably this behaviour is due to the presence of more sky and less monument reflections in the river with respect to the other images. These variation lead to fewer details in the image scene. As a consequence, the overall quality trend is the higher which means worse perceived quality.
The same behaviour occurs for the images affected by vertical changes which are related to the photographs A.1 (d and e). The score trend for the photograph A.1(d) plotted with a purple line is near the reference image trend. Instead, the photograph A.1(e) is represented by the lowest trend, in other words higher perceived quality. This fact can be justified by the gray rightmost building which not occurs in the image A.1 d and is almost imperceptible in the reference image. The building absence leaves more space for the sky content which is a component with few details.
Differences in score trends of rotationally-changed photographs can be also

appreciated by the figure. However, they are less evident with respect to the cases previously described.
In conclusion, the score trends confirm the BRISQUE sensitivity to slight content variations in photographs taken with the same device. It can be state that the perceptual quality evaluation performed by the algorithm is not independent from translational and rotational variations. Moreover, this aspect holds for the original images which leave out any alterations due to the downsampling process.
Finally, this is a weighty issue in the case of evaluating images photographed by different users with respect to the same scene (i.e. a monument). For example if two users share the same device and they take the photograph side by side then it could be difficult to choose which photos is better in terms of quality because the score dependence from the image content.

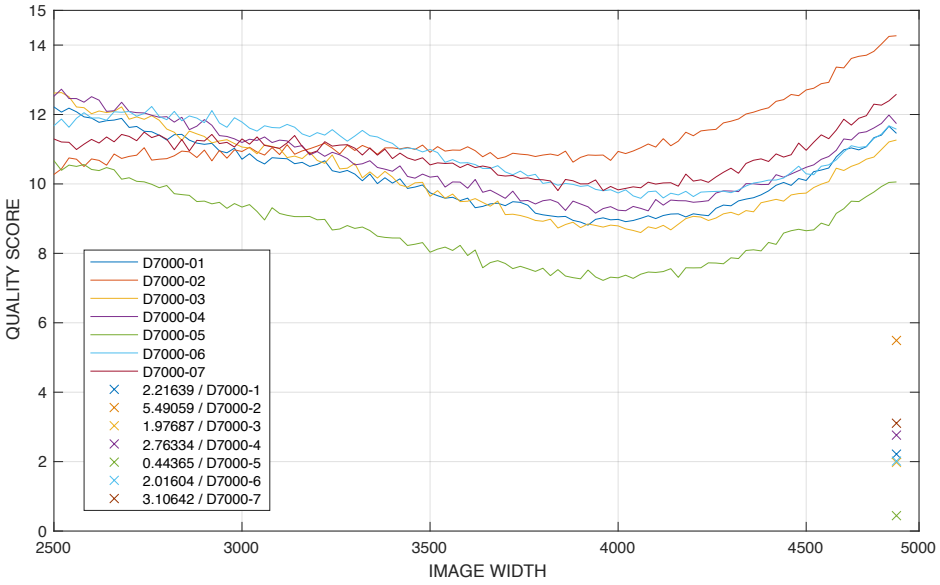


Figure 4.4: Quality scores computed on images of the first Portello subset. Cross points represent quality scores of undownsampled images.

4.3 Sensitivity With Respect to Filter Techniques

Another important aspect that deserves to be studied has been pointed out during the previous downsampling processes. Indeed, all the perceived quality trends presented before are characterized by a gap between the score of the original image and the score of its first downsampled version.
The reasons of this behavior can be explained by the bicubic interpolation (over 4x4 neighboring pixels) which was introduced twice in the BRISQUE algorithm. It is present in the original algorithm and in the additional downsampling procedure introduced for this thesis.
This interpolation technique is characterized by a low-pass filtering process as many other downsampling methods. If this is responsible of the score degradation then the previously described gap behaviours could be motivated and the BRISQUE weakness to filter technique raised.
In order to prove this statement, quality score of many interpolation methods were compared with respect to the OpenCv bicubic interpolation.
The widely used Photoshop editing program was involved in the comparison process. It provides a rich set of downsampling methods which include also the same bicubic interpolation. All that methods have been applied to the image in Figure A.3 a which was scaled to specific dimensions. Outcomes of this experiment are listed in Table 4.2 and plotted in Figure 4.5.

The score trends manifest a similar behaviour for four out of five downsampled methods. These are Bicubic Sharper, Bilinear and the Photoshop and BRISQUE Bicubic Interpolation. Small differences between the last two methods underline the BRISQUE sensitivity to different implementations of the same downsampling procedure.
However the special case comes from the Photoshop Nearest Neighbor, which shows a very different behaviour respect to the others. The first downsampled version has a quality score value similar to that of the original images. Instead, in the other cases a marked score loss occurs.
As explained in some experiments before, score losses for the first downsampled versions is not justified by differences in the perceptive quality. Indeed a human observer could assess the original image and the first downsampled version in the same manner.
The only reason which justify the Nearest Neighbor score trend is the absence of any filtering technique. Indeed, if compared with the other methods this is the only which does not include a filter.
Finally, the previous experiments have demonstrated the BRISQUE weak-

nesses with respect to filter techniques. This fact is very relevant because filters are involved in many steps of the digital image processing pipeline. Moreover, considering that a filtering technique is performed by the BRISQUE algorithm in the downsampled image generation, it is reasonable to think that each final quality score is internally modified by this presence.

Width	Height	BRISQUE Bicubic	Photoshop Bicubic	Photoshop Bilinear	Photoshop Bicubic Sharper	Photoshop Nearest Neighbor
4908	3250	4,94596	4,82997	9,03155	7,03995	-8,31694
4602	3048	3,25529	5,08893	7,36883	7,49256	-3,46657
3986	2640	2,98456	6,13542	6,0302	6,99707	6,06778
2808	1860	13,8697	8,70872	11,1723	8,50228	17,045
2084	1380	23,3335	12,6964	15,4659	11,8333	21,9118

Table 4.2: Quality scores computed on the D7000-09 image with respect to five dimension and downsampling techniques. The original quality score was -8,77023 (image dimension of 4928x3264).

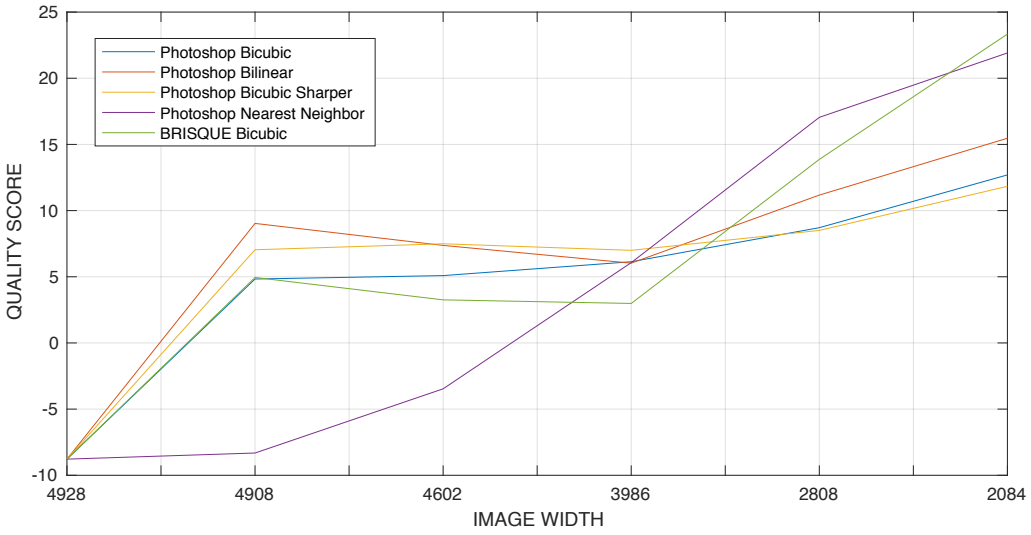


Figure 4.5: Quality scores computed on five downsampled versions of the D7000-09 image. Different downsampling techniques were applied before score evaluation.

4.4 Feature Analysis

This section aims at evaluating the robustness of the BRISQUE features. Some doubts were pointed out in the previous sections, when different downsampling methods were deployed before the process of quality evaluation.
The first experiment has the purpose of comparing features extracted from the image in Figure A.3 (b). In particular, using the Matlab and C++ algorithm versions, two clusters of features were computed. The values are shown in Figures 4.6 (a) and 4.6 (b), which represent features before and after the rescaling process, respectively. As described in Chapter 3, the rescaling process deploys a low pass filter followed by a downsampling phase of the input image (by a factor of 2).
In the first figure, it should be noted how differences are mainly concentrated in the range of features from 19 to 36, which represent the second half of the BRISQUE feature set. It is important to remember that the two BRISQUE implementations represent the same algorithm. No implementation error has been discovered, thus, the feature disparities can be due to inner implementations related to inner function of Matlab or C++.
The previously cited inequalities are further emphasized by the rescaling process, as can be seen in second figure. This is obviously due to the BRISQUE *allrange* file. It is used in the rescaling process and contains the upper and lower limits of each feature computed on the undistorted LIVE database images. Probably, each image has different features with respect to the two implementations.

This first experiment represents a starting point in the BRISQUE feature-strength analysis. Indeed, the previously cited feature differences can be considered as weaknesses of the BRISQUE algorithm. With the aim at validating this process, in the following subsection the comparison will be extended and analyzed to the entire LIVE IQA database [12].

4.4.1 Sensitive features in different models

Feature analysis started with value comparisons on two feature sets. However, that experiment included features from a single input image. In order to obtain an unbiased result, the process has been replicated for the entire LIVE database. Some code has been developed for the Matlab implementation in order to provide the training model procedure. Indeed, this procedure has been provided by the authors only for the C++ BRISQUE version. Basically, it allows to compute features for all the images contained in the LIVE

database.
Given two feature sets computed on the LIVE database with the two BRISQUE implementations, their similarity has been studied using the following method. Feature sets are grouped in matrices 4.1 and 4.2. In these matrices lines represent images (779 in total) and columns represent feature values (36 in total) for a given image.

$$Matlab_{image,feature} = \begin{pmatrix} Mat_{1,1} & Mat_{1,2} & \cdots & Mat_{1,36} \\ Mat_{2,1} & Mat_{2,2} & \cdots & Mat_{2,36} \\ \vdots & \vdots & \ddots & \vdots \\ Mat_{779,1} & Mat_{779,2} & \cdots & Mat_{779,36} \end{pmatrix} \tag{4.1}$$

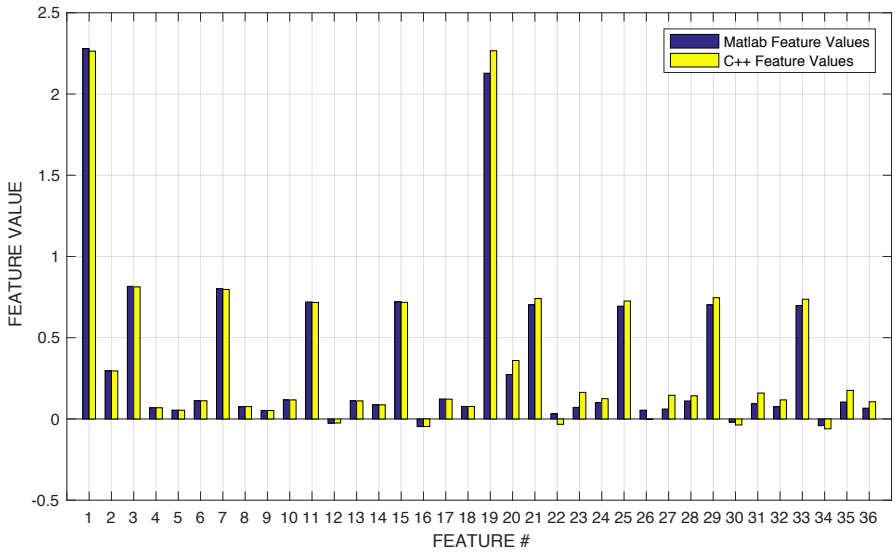
$$Cpp_{image,feature} = \begin{pmatrix} Cpp_{1,1} & Cpp_{1,2} & \cdots & Cpp_{1,36} \\ Cpp_{2,1} & Cpp_{2,2} & \cdots & Cpp_{2,36} \\ \vdots & \vdots & \ddots & \vdots \\ Cpp_{779,1} & Cpp_{779,2} & \cdots & Cpp_{779,36} \end{pmatrix} \tag{4.2}$$

Then matrix 4.3 was computed with the purpose of discovering any feature differences, through an empirical-based approach. In fact, if the matrix values are equal to zero then features are implementation-independent. On the other side, if values are different from zero then even small implementation details affect the features.

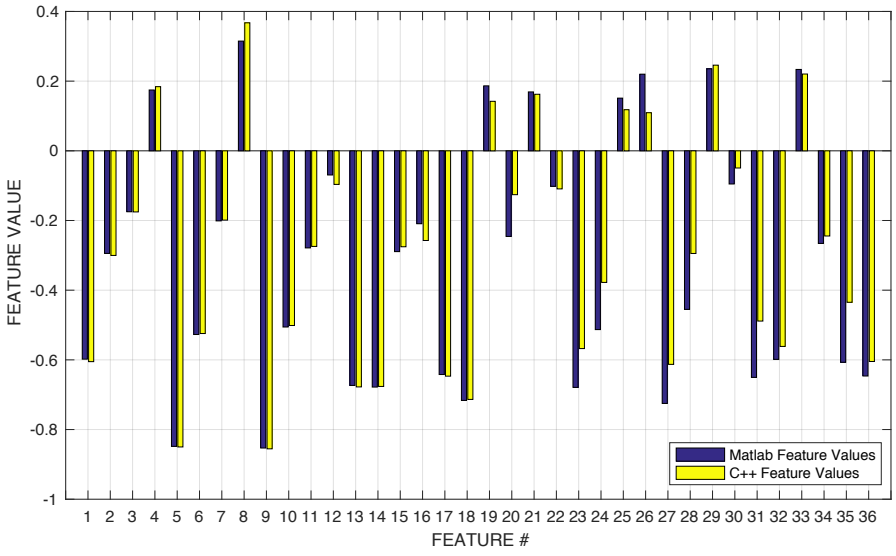
$$Diff_{image,feature} = \begin{pmatrix} \frac{Cpp_{1,1}-Matlab_{1,1}}{Matlab_{1,1}} & \cdots & \frac{Cpp_{1,36}-Matlab_{1,36}}{Matlab_{1,36}} \\ \vdots & \ddots & \vdots \\ \frac{Cpp_{779,1}-Matlab_{779,1}}{Matlab_{779,1}} & \cdots & \frac{Cpp_{779,36}-Matlab_{779,36}}{Matlab_{779,36}} \end{pmatrix} \tag{4.3}$$

To better understand the matrix 4.3, a splitting process was performed. It has lead to the matrix division into positive and negative values, respectively contained in two matrices of the same dimension. Reasons for this actions is that the matrix 4.3 has been observed to contain high but infrequent positive and negative values. Comparing these values all together results in a loss of details and the major differences cannot be appreciated.
The values contained in these two matrices are shown in Figures 4.7 (a) and 4.7 (b) for positive and negative values, respectively.

For the sake of clarity, these figures contain the feature values extracted by the BRISQUE algorithm with respect to the distorted images of the LIVE IQA database. The features of a image are positives or negatives and their signs can be further changed by the filling process employed in the matrix 4.3. As a consequence, the features related to an image could be contained in the positive and negative matrices.
Despite that, these figures point out a common characteristic related to the feature values. That is, the high-value features are mostly concentrated in the second half of the x-axis, i.e. for feature numbers in the range from 21 to 36. As can be seen this statement is not absolutely true for all features in this range. However, the comparison with respect to the first half of the x-axis in both figures highlights this general aspect.
In order to quantify which feature cluster demonstrates high variability, a threshold study regarding positive and negative values was performed. It has been shown that eight out of eighteen features related to the second image scale group, differ at least 20 times in the two implementations. Moreover, three features (23,26,28) differ at least 500 times in the two implementations. The results of this study are shown in Table 4.3.
Keeping in mind the considerations described in the previous section, it can be stated that the BRISQUE algorithm sensitivity to filter techniques affects mostly a subset of the second half feature group. This result will be applied in the next section in order to develop a new ensemble of predictive features.

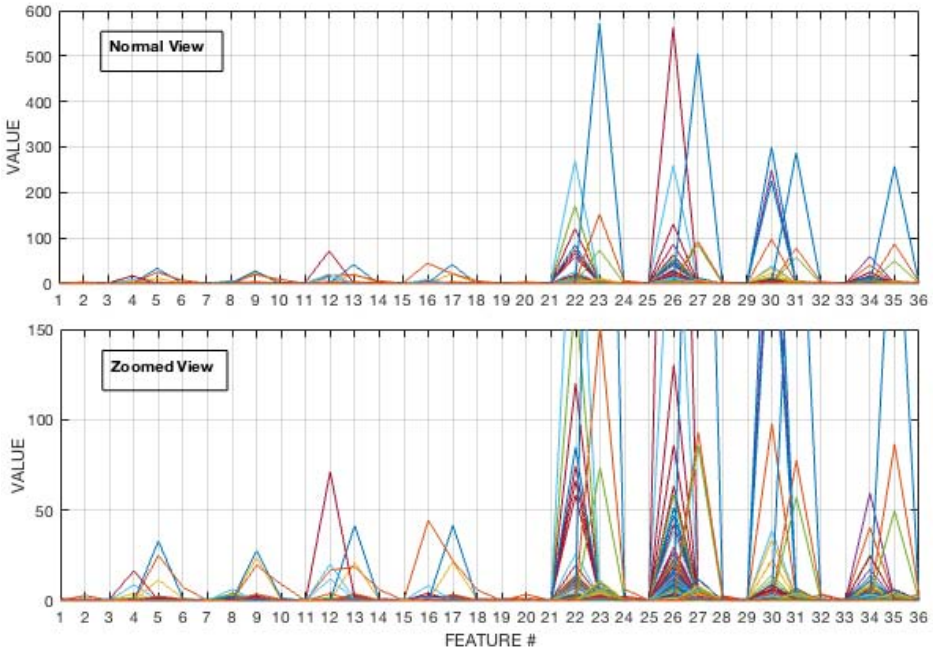


(a) Feature values before rescaling

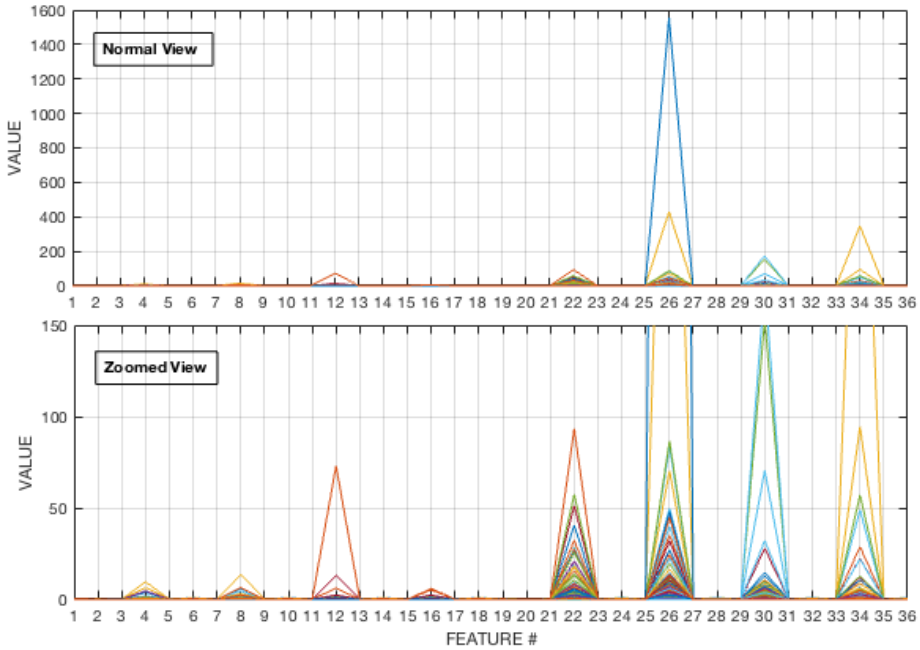


(b) Feature values after rescaling

Figure 4.6: Comparison of feature values computed on the D7000-09 image with respect to the two BRISQUE implementations. Features values are extracted before (a) and after (b) the rescaling process.



(a) Positive matrix values



(b) Negative matrix values

Figure 4.7: Values contained in the matrix 4.3 after the splitting process into positive (a) and negative (b) values.

Feature #	> 20	> 50	> 100	> 200	> 300	> 400	> 500
5	2	0	0	0	0	0	0
9	2	0	0	0	0	0	0
12	3	0	0	0	0	0	0
13	2	2	0	0	0	0	0
16	1	0	0	0	0	0	0
17	3	0	0	0	0	0	0
22	24	11	4	1	0	0	0
23	3	3	2	1	1	1	1
26	38	13	5	4	3	3	1
27	3	3	1	1	1	1	1
30	13	7	5	3	0	0	0
31	3	3	1	1	0	0	0
34	11	4	1	1	1	0	0
35	3	2	1	1	0	0	0

Table 4.3: Threshold analysis of feature values contained in the Matrix 4.3.

Chapter 5

BRISQUE Quality assessment improvements

The goal of this chapter is to improve the performances of BRISQUE in terms of correlation with human judgements. In order to achieve this goal, many subsets of features will be removed from the BRISQUE feature set and their effectiveness proved through multiple train-test iterations on the LIVE IQA database. In particular, the feature selection process will start with the removal of entire groups of features. Afterwards, the most promising subsets will be further reduced, achieving groups composed by two features. These experiments will terminate with the removal of the last-eighteen features from the BRISQUE algorithm, which are computed on the filtered and downsampled version of the input image.
Following an ensemble-based approach, a new set of frequency-based features will be included in the reduced BRISQUE model. These will be selected from the BLIINDS-2 algorithm as a result of performance evaluations.
The effectiveness of the proposed model will be proved with a comparison to the original BRISQUE feature set.
Finally, to further increase the overall quality evaluation performance for mobile devices, features based on the JPEG compression process will be analyzed.

5.1 Feature Selection

According to the BRISQUE paper [3], the performance of an IQA algorithm can be measured using the SROCC correlation coefficient computed on the quality scores and the human judgements.
A high-value correlation coefficient depends naturally by the choice of predictive features. The BRISQUE algorithm exhibits a high median SROCC coefficient (0.9395) on the LIVE database and its features are to be considered effective.
Despite this fact, from the previous chapter sensitive features with respect to implementation details have been discovered. In order to understand their real impact on the final BRISQUE performance, the SROCC coefficient was computed without them.
Many experiments of SROCC evaluation were performed without some features from the original BRISQUE feature set (Table 3.1). Each experiment has required the training and testing phases on the entire LIVE IQA database. Quality scores predicted by algorithm were then compared with the subjective judgements (DMOS).
The first experiment was performed discarding the range of features from 19 to 36, thus, only those in the range from 1 to 18 were used for training and testing. In the remainder of this thesis, the predictive model computed on the first-eighteen BRISQUE features will be called "BRISQUE18".
Results of the first experiment are shown in Table 5.1. In the first column, it should be noted that the SROCC is different from that presented by the authors (0.9395). This fact is due to the median SROCC across 60 train-test iterations on the LIVE IQA database. The original BRISQUE correlation represents the median SROCC across 1000 train-test iterations on the LIVE IQA database. Despite that, the SROCC presented in the first column of Table 5.1 will be considered as reference for future comparisons.
In the previous table, the most important result comes from the SROCC of BRISQUE18 (0.9327). Despite this value is reasonably lower with respect to the BRISQUE full feature set, their difference is not particularly high. This behaviour needs to be further investigated in order to understand how the BRISQUE features obtained from the filtered and downsampled image version, i.e. in the range from 19 to 36, affect the perceptual quality assessment.

The second set of experiments was performed with the removal of groups of features shared between the original image and its downsampled version. Results are presented in Table 5.2. The first column shows the lower SROCC in the table. This is due to the high number of features removed (7 per scale, probably some of them highly predictive). Except for the last two columns,

	BRISQUE	BRISQUE18
SROCC	0.9522	0.9327

Table 5.1: Median SROCC across 60 train-test iterations on the LIVE IQA database with respect to the original BRISQUE feature set and BRISQUE18.

the other correlation values are very close to the reference SROCC. These results confirm the presence of weak features in the BRISQUE feature set.

Feature set removed	[11-18, 29-36]	[3-6, 21-24]	[7-10, 25-28]	[13-16, 29-32]	[17-20, 33-36]
SROCC	0.8934	0.9508	0.9483	0.9450	0.9395

Table 5.2: Median SROCC across 60 train-test iterations on the LIVE IQA database. The feature subsets [11-18,29-36], [7-10,25-28], [13-16,29-32] and [17-20,33-36] were removed from the BRISQUE feature set before training and testing.

In order to discover if some subsets from the previous table contain other weak features, a further reduction was performed. Sets composed by two features were used for the median SROCC analysis, the results are shown in Table 5.3. It should be noted how, surprisingly, all the SROCC values are greater than 0.95 and very close to the reference coefficient. This can be considered as another proof of the presence of weak features.

Feature set removed	[3,21]	[4,22]	[5,23]	[6,24]	[9,27]	[10,28]
SROCC	0.9514	0.9501	0.9515	0.9505	0.9510	0.9509

Table 5.3: Median SROCC across 60 train-test iterations on the LIVE IQA database. The feature subsets [3,21], [4,22], [5,23], [6,24], [9,27] and [10,28] were removed from the BRISQUE feature set before training and testing.

All the previous experiments have been characterized by the use of some features from the second image scale. Keeping in mind the study performed in the previous chapter, it is reasonable to think that SROCC values obtained so far are highly affected by some of those features.
In order to understand to what extent the single features from the [19-36] group are related with the final SROCC, training and testing phases were

perfomed without some of them. The results are presented in Tables 5.4 and 5.5.
The SROCC values are further increased with respect to the previous experiment and two of them the are very close to the reference value (features 23 and 27). These results suggest that features extracted from the filtered and downsampled version of the input image do not provide any robust contribution to the predictive power of the BRISQUE model. This fact will be used later in order to substitute part of the original feature set.

Feature removed	23	24	26	27
SROCC	0.9521	0.9516	0.9501	0.9521

Table 5.4: Median SROCC across 60 train-test iterations on the LIVE IQA database. The features 23, 24, 26, 27 were removed from the BRISQUE original set before training and testing.

Feature removed	29	31	32	35	36
SROCC	0.9516	0.9520	0.9518	0.9517	0.9517

Table 5.5: Median SROCC across 60 train-test iterations on the LIVE IQA database. The features 29, 31, 32, 35 and 36 were removed from the BRISQUE model before each training and testing.

5.2 BRISQUE and BLIINDS-2

In the previous chapter (section 4.4), the BRISQUE features in the range from 19 to 36 were proved to be sensitive to filter techniques. Moreover, the experiments performed in the previous section have demonstrated that the removal of some features from the set [19-36] does not entail a severe drop in the SROCC.
Based on these principles, the features computed on the filtered and downsampled version of the input image were eliminated from the model. In this section, a new feature set will be proposed with the aim of replacing this weak group and improving the overall performance.
An ensemble-based approach has been adopted in order to find this new set of highly-predictive features. According to the BRISQUE model, the features seen so far are exclusively obtained from spatial-domain statistics. However,

the possibility to enhance the algorithm using features computed in other domains is clear.
Other no-reference algorithms such as [18], [38] and BLIINDS-2 have obtained excellent performances with features developed in the frequency domain. In particular, the latter was chosen as candidate algorithm for the substitution of the last-eighteen BRISQUE features.
This choice comes from the BLIINDS-2 model simplicity and from its temporal performances rather than its frequency-based features. A temporal comparison beetween BRISQUE and BLIINDS-2 is shown in Table 3.3. Despite the fact that BLIINDS-2 is slower, there is still room for improvements. They will be described in the next chapter.

In this section the BLIINDS-2 Matlab implementation [24] will be used because of the absence of any C++ version.
Unfortunately, during the process of code analysis some errors were discovered with respect to what stated in the paper.
The first error is located in the file `dct_freq_bands.m`, where the `r2` coefficient is computed. In fact the formula 3.10, regarding the energy band coefficient number 1, is wrongly applied. The denominator should be $(var\_band2 + var\_band1 + eps)$ instead of $(var\_band3 + var\_band1 + eps)$.
The second error is located in the file `bliinds2_feature_extraction.m`. Indeed, before any computational process the red channel is extracted from the input image. It is not clear the reason why the algorithm works on this channel. It should be more reasonable to execute the algorithm on the grayscale image version. In order to be coherent with the BRISQUE implementation, the grayscale version will be used in future comparisons.

5.2.1 Multiple feature sets

The first step towards the inclusion of the BLIINDS-2 algorithm was the insertion (before the training procedure) of its frequency-based features. Thus, the new feature set is composed by the first-eighteen BRISQUE features and the features from the BLIINDS-2 algorithm which were merged all together in a single vector.

As described in Section 3.2, the BLIINDS-2 algorithm computes eight features per image, with a maximum of twenty-four features. In particular, the last sixteen are extracted from two filtered and downsampled versions of the input image. However, as stated in [23], not all the 24 features of the BLIINDS-2 algorithm contribute equally to the perceived quality evaluation.

As a consequence, in order to evaluate the proposed feature model and understand to which subset of BLIINDS-2 features is more predictive, many experiments were performed. Similarly to the previous section, they are represented by train-test iterations on the LIVE IQA database which yield as output the SROCC values.
The goal of the first experiment is to evaluate the new model effectiveness with respect to the BLIINDS-2 features at different image scales. Outcomes of this experiment are shown in Table 5.6.
The results show a better quality evaluation performance with respect to the BLIINDS-S algorithm (0.9232). Furthermore, the SROCCs in the last two columns overcome that of the BRISQUE reference (0.9522).
Since the results are computed as median SROCC across multiple train-test evaluations on the LIVE database, they are to be considered reliable. Moreover, it should be also noted how the introduction of features from different image scales leads to a score performance increase. The small gap between the SROCC values of the second and third scales is justified by the results stated in [23] with regard to the third scale introduction.
In the BLIINDS-2 algorithm, no analysis has been performed by the authors with regard to inter-scale feature relationships. However, the small gap in the SROCC between the second and third scales could be due to overlapping features.
It is not clear why the authors selected exactly three different scales for the BLIINDS-2 model. Despite that fact, features computed on the third scale could be as effective as those computed on the second scale. Keeping in mind the thesis goal, a low-resources mobile device could computationally take advantage of this aspect.
With the aim of proving the above cited hypothesis, some experiments were performed taking into consideration features from the first and third image scales rather than some of their subsets.
Outcomes of these experiments are presented in Table 5.7. They are divided in four columns, each one regarding different feature sets. In the first column the SROCC is obtained involving all the features related to the first and third scales which are shape (γ), frequency variation (ζ), sub-band energy (R_n) and the variance of frequency variation across orientation features (ζ_{or}).
It should be appreciated how the median SROCC value seems to sustain the hypothesis of overlapping features. In fact, the correlation value in the first column is even greater if compared with the three scores presented in Table 5.6.
In conclusion, the predictive power of the model containing the first-eighteen BRISQUE features along with the features from the first and third BLIINDS-2 scales overcomes that containing also the second scale.

The second experiment aims at discovering the importance of the previously introduced γ parameter. In the BLIINDS-2 Matlab implementation, the shape feature extraction is performed through a very computationally expensive task. If good performances in terms of SROCC can be obtained without this feature then the algorithm could be strongly speeded up. Again, this aspect is crucial in order to deploy the model in a low-resources mobile device.
The SROCC value in the second column of Table 5.7 seems to confirm the possibility of removing the shape parameter. Surprisingly, this result slightly overcomes the median SROCC of the model including the shape feature.
Finally, the last two columns are related to experiments which have further removed features from the first and third scales. However, the prediction power of the model seems to be slightly more sensitive to the removal of these features.

Features added to BRISQUE18	BLIINDS-2 One scale	BLIINDS-2 Two scales	BLIINDS-2 Three scales
SROCC	0.9477	0.9536	0.9552

Table 5.6: Median SROCC across 60 train-test iterations on the LIVE IQA database of BRISQUE18 and the BLIINDS-2 features computed on three different scales.

Features added	BLIINDS-2 1th and 3rd scales			
to BRISQUE18	γ,ζ,R_n,ζ_{or}	ζ,R_n,ζ_{or}	ζ,R_n	R_n,ζ_{or}
SROCC	0.9554	0.9558	0.9513	0.9529

Table 5.7: Median SROCC across 60 train-test iterations on the LIVE IQA database of BRISQUE18 and some BLIINDS-2 features computed on the first and third scales.

Another bottleneck in the BLIINDS-2 execution time is due to the dimension of DCT blocks. These have dimension of 5x5, with one pixel of overlapping border between adjacent blocks. In our hypothesis, if the block dimension increases then the algorithm could be speeded up.
In order to analyze the algorithm performance with respect to the block dimensionality parameter, other two SROCC experiments were performed. In

particular these take advantage of DCT block dimensions multiple of that involved in the JPEG algorithm which is typically 8x8.
Results are shown in Table 5.8. These SROCC values suggest that small increases with respect to the block dimension do not entail a severe drop in terms of performance.
Finally, in Figure 5.1 the best models discovered in this thesis are compared to the original BRISQUE algorithm. The comparison on each iteration is possible because a seeded pseudo-random image division was applied. That is, using the Matlab environment the image training set (and consequently the testing set) was created with a random permutation preceded by a random generator seeding. The used seed is the number of the current iteration, e.g., if the current iteration is the number five then the generation procedure was preceded by $rng(5)$. It should be noted how, in almost each iteration, the proposed models overcome the BRISQUE algorithm. The only iterations where it does not occur are characterized by bars ending with black color.

Features added	BLIINDS-2 1th and 3rd scales, no shape (γ)	
to BRISQUE18	8x8 block size	16x16 block size
SROCC	0.9504	0.9493

Table 5.8: Median SROCC across 60 train-test iterations on the LIVE IQA database of BRISQUE18 and the BLIINDS-2 features computed on different DCT block sizes.

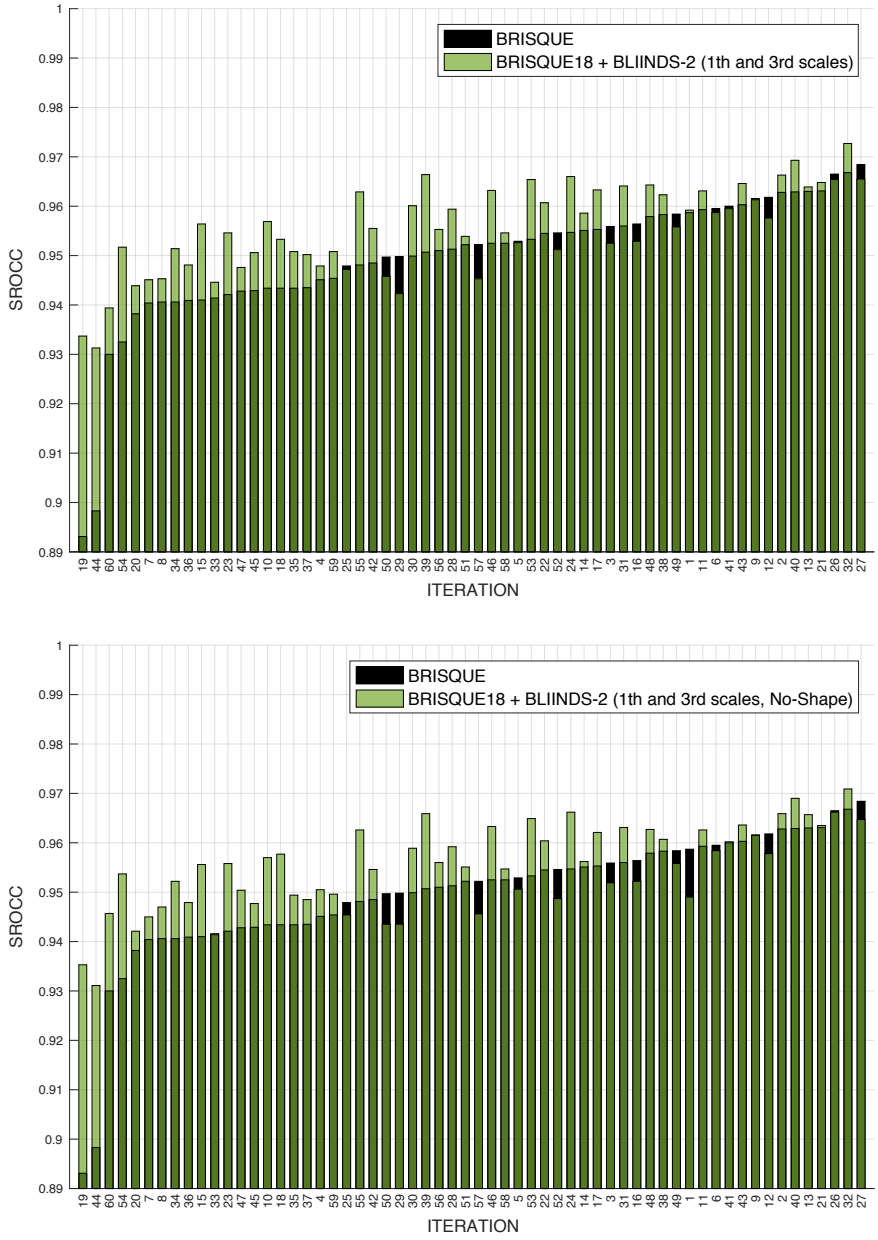


Figure 5.1: SROCC values of BRISQUE (black charts) VS the two proposed best models (green charts) across 60 train-test iterations on the LIVE IQA database.

5.3 Features for Quantized DCT Coefficients

The images acquired with mobile devices follow a common pipeline. Indeed, they are acquired, processed and finally stored. Before the latter procedure, in the processing step, each image is typically compressed in order to save space.
The most popular algorithm employed for digital image compression is commonly refereed to as JPEG. It is employed in many other applications outside the domain of mobile devices such as transmission between web applications. Despite it is commonly used for lossy compression, it also provides a lossless version.
In its lossy version, the JPEG algorithm involves a quantization step, the feature here introduced are developed upon the results of this process.
A short description of the JPEG algorithm will be provided in the first subsection. Then in the second subsection features related to quantized coefficients will be involved to improve the SROCC performance achieved so far.

5.3.1 The JPEG compression algorithm

Despite the JPEG algorithm can be employed with small changes, its process pipeline is well defined. The encoding procedure can be summarized with the following steps:

- The input image is converted from the RGB space to the YC_BC_R, consisting of one luma component (Y), representing brightness, and two chroma components, (C_B and C_R), representing color. This color space allows for greater compression without losing in perceptual quality. In particular, the compression effectiveness is due to the brightness channel which is considered more relevant in the human quality assessment.

- Given the YC_BC_R image representation, a downsampling process is performed. This takes advantage of the observations made in the previous step, which highlights the Y channel key role. Indeed, the downsampling process typically acts on the chrominance channels (C_B and C_R) without modifying the brightness channel. The ratios at which the downsampling process is typically employed are 4:2:2 (reduction by a factor of 2 in the horizontal direction), or 4:2:0 (reduction by a factor of 2 in both the horizontal and vertical directions).

- The block splitting process take place on the input image. It is commonly performed with respect to block dimension of 8x8, however other dimensions such as 16x16 can be employed. If image dimension is not

a multiple of the block size, the blocks computed at the image borders are typically filled with dummy data.

- The block channels, in a typical 8-bit per channel representation, contain non-negative values in the range from 0 to 255. They are converted to a zero-centered representation, in the range from [-128,127], in order to reduce the dynamic range required by the following process. This convert each channel to the frequency domain, using the Discrete Cosine Transform (DCT). The converted blocks contain in the top-left corner the direct component (DC), which defines the basic hue for the entire block. The remaining components are called Alternating Components (AC). The AC components from the top-left to the bottom-right corner represent increasing frequencies.

- A quantization process is performed in order to reduce the high frequency components occurring in the image blocks. Each block component is quantized through a division by a constant value, and then rounding to the nearest integer. If the DCT involved in the previous step is made with high precision, then the quantization is the only lossy process in the JPEG algorithm.

- The entropy coding step is then performed. It leads to a lossless image compression employing the run-length encoding (RLE), often refereed as the "zig-zag" block division. This permits to group similar frequencies components among different blocks. At the end of this process the input image is JPEG encoded.

The decoding step follows the above steps in reverse. Both encoding and decoding JPEG processes are presented in Figure 5.2.

5.3.2 Feature selection

Another experiment performed in order to increase the overall SROCC was the introduction of features related to the JPEG format. These are yields as output of a statistical model of quantized DCT coefficients, which is employed for discovering hidden information in images produced by a typical digital camera [30]. These kind of methods are grouped in the Steganalysis research area.
In our opinion these features could be representative for quality evaluation of images acquired with mobile devices. They should carry out information about the delivered quality of digital camera sensors.
The algorithm employing the cited statistical model takes advantage of a

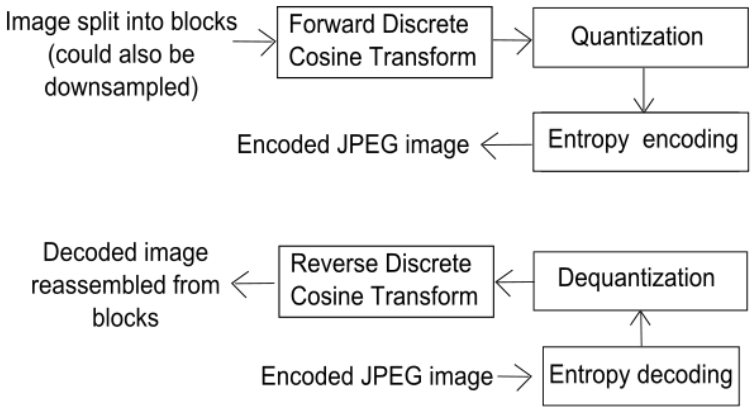


Figure 5.2: Encoding and decoding processes performed by the JPEG algorithm.

block splitting process, which is similar to that involved in the JPEG algorithm. Similar frequency coefficients are grouped among image blocks and then fitted by the proposed model. The grouping procedure does not involve the DC coefficient because it does not provide any structural information.
Two features are yields as output of the fitting process. They are defined as ν and η and are obtained for each frequency group, resulting in a final matrix with 63 rows and 2 columns.
The proposed statistical model is provided with a Matlab implementation [30]. In order to fit the quantized coefficients, the source code takes as input a single-channel JPEG image which contains the unquantized components and the quantization tables. They are extracted using the LIBJPEG [14] software library, and provided to the statistical model.
However, the images contained in the LIVE database are bitmaps and the LIBJPEG does not provide any useful functionality to extract quantization coefficient from such a kind of format. The proposed implementation was then modified. Given a bitmap image, the algorithm computes the quantization coefficients directly. The hypothesis here made is the lossless representation of the LIVE database images. The quantization table is then a matrix of ones and the process of extracting unquantized components follows that described in the previous section.
Using an empirical-based approach, it was noted that the 63th coefficient histograms demonstrate high variability with respect to a pristine image and its distorted versions. This behaviour should be relevant for quality prediction. Only the ν and η parameters related to the 63th coefficient were then added to the models proposed in the previous chapter.

In order to prove the quality evaluation performances of this last model, 60 train-test iterations on the LIVE database were performed and listed in Table 5.9.

Features added	BLIINDS-2 first and third scales	
to BRISQUE18	γ,ζ,R_n,ζ_{or}	γ,ζ,R_n,ζ_{or},ν_{63},η_{63}
SROCC	0.9554	0.7648

Table 5.9: SROCC value of a single train-test iteration on the LIVE IQA database. Model trained on BRISQUE18, BLIINDS-2 (features from 1th and 3rd scales) and the parameters resulting from the model fitting of the 63th quantized coefficient.

Unfortunately, the SROCC of the proposed model is much lower than that in the first column. Reasons of such a low correlation between predicted scores and human judgements are related to the white noise distortion. Indeed, this class of images presents high density of high-frequency components. The parameters ν and η are excessively high for these images if compared with other kind of distortion classes. This affect the training of the model during the rescaling process, which scales each feature in the range from -1 to 1.
Basically, the features ν and η are always equal to -1 for images not distorted with white noise. This behaviour generates the model ineffectiveness.

Chapter 6

Performance improvements

An important characteristic of the BRISQUE algorithm is the short execution time. Indeed, if compared with other referenceless IQA algorithms, it manifests the best time performances. This property makes the algorithm a good candidate to be deployed in a mobile device.
The goal of this chapter is to measure the BRISQUE execution time on a mobile device.
A smartphone application will be developed in order to provide a framework to BRISQUE algorithm. Moreover, the time enhancements
achieved for the mobile device will be always compared to those of a computer.

6.1 BRISQUE Performance Analysis

The time analysis will be performed with regard to a computer and a smartphone. In particular, the computer involved is a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram. Instead, the mobile device is a Google Nexus 5 equipped with the Android operative system, with a 2.3 GHz Quad Core Qualcomm MSM8974 Snapdragon 800 processor and 2 GB of Ram.
In order to perform temporal comparisons between the previously described devices, a unique photograph was involved. This was captured using the same Google Nexus 5 smartphone and it is shown in Figure 6.1.

6.1.1 Computer performances

With the aim at obtaining the computer temporal performances, the C++ BRISQUE version [1] was used. The advantages of this choice are twofold.

Figure 6.1: Image of dimension 3286x2432 used in the process of time improvement. It was taken with a Google Nexus 5 in normal conditions.

The first is the use of a non-interpreted programming language such as C++, which to some extent is generally faster with respect to the BRISQUE Matlab version. The second, is the possibility to deploy the C++ program in the future Android application through the native code interface provided by the mobile operative system.
The time analysis follows the study proposed in [3], where the algorithm execution time has been divided into three sections. For simplicity this study is proposed in Figure 6.1.
The first step performed in this section is the checking process of the compliance with the informal complexity 6.1. That is, given as input the image in Figure 6.1, the total time required by the quality evaluation was computed along with the "MSCN", "GGD" and "Pairwise Products and AGGD" steps. Outcomes of this experiment are shown in table 6.2.
As it can be seen from the table, about 70% of the execution time is due to the "Pariwise Products and AGGD" step. It is a considerably different result in comparison with the BRISQUE original informal complexity analysis. The reason of this percentage imbalances could be due to the image's dimension, i.e., the input image has a dimension of 3286x2432 which substantially different from the LIVE database images (typically 768x512).

The first time evaluation has provided some insights on which the improvements should be performed. Moreover, the total execution time of 2.23s is encouraging for the future plan of algorithm deployment in a realtime mobile application.

step	percentage of time [%]
MSCN	50.9
GGD	8.6
Pairwise Products and AGGD	40.6

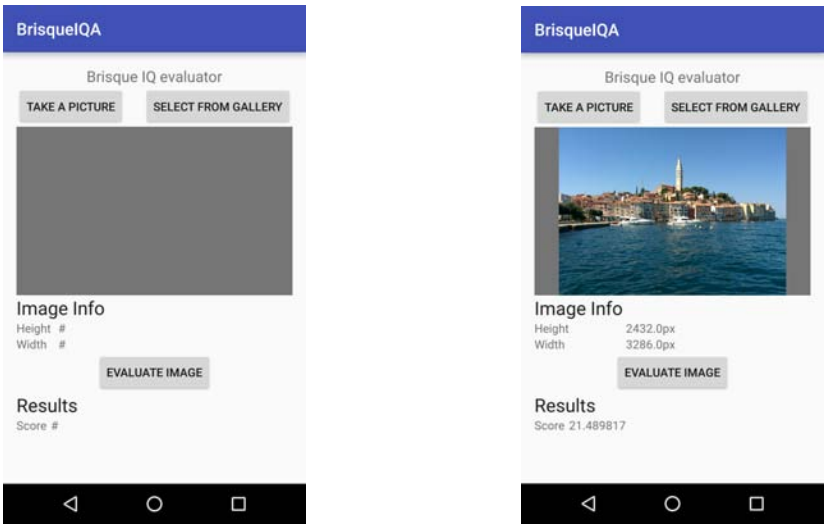
Table 6.1: Original informal complexity analysis of the BRISQUE algorithm. Percentages for the three steps are computed with respect to the overall execution time upon a single input image (once the model trained)

step	percentage of time [%]	time [s]
MSCN	20.0	0.45
GGD	8.2	0.18
Pairwise Products and AGGD	71.8	1.60
total time		2.23

Table 6.2: Informal complexity analysis and execution time of the BRISQUE algorithm on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram. Input image in Figure 6.1.

6.1.2 Mobile device performances

As described before in this chapter, the algorithm execution time analysis has a key role in the path to mobile deployment. With the aim of achieving this goal, a mobile application was developed using the Android Studio IDE. Again, the BRISQUE C++ version was involved in the mobile application development. Since the standard programming language for Android applications is JAVA, the Android Development Kit (NDK) was used in order to export the C++ algorithm code inside the mobile application. Moreover, a graphical interface was developed to permit the user selection of photographs or pictures. Two ways of selecting an input image have been provided, i.e., “take a picture” and “Select from gallery” with their obvious meanings. Screenshots of the mobile application are shown in Figure 6.2.



(a) Before image evaluation (b) After image evaluation

Figure 6.2: Screenshots of the Android application deploying the BRISQUE algorithm.

The application performances were measured in the same points and steps described in the previous section. During the image quality evaluation no background applications or processes have been launched. So as to obtain as much as possible unbiased measurements. Outcomes of this experiment are shown in Table 6.3.
No substantial differences are pointed out by the informal complexity with respect to those presented in the previously computer analysis. Furthermore, the "Pairwise Products and AGGD" step demonstrates to be a very time consuming process even in the mobile environment.
The time results shown in the table 6.3 exhibit a total execution time of $6.40s$. This is not a considerable execution speed for a realtime mobile application. However, since no improvements have been performed on the code so far, it is worth to state that there is still room to enhance the algorithm.

6.2 BRISQUE Improvements

As seen in the previous sections, on both the mobile and computer devices a significant amount of time is required by the "Pairwise Products and AGGD" step. It has been found that this step is the best candidate in order to improve the algorithm time performances.

step	percentage of time	time [s]
MSCN	27.5	1.76
GGD	5.7	0.36
Pairwise Products and AGGD	66.8	4.27
total time		6.40

Table 6.3: Informal complexity analysis and execution time of the BRISQUE algorithm on a Google Nexus 5 with a 2.3 GHz Quad Core Qualcomm MSM8974 Snapdragon 800 processor and 2 GB of Ram. Input image in Figure 6.1.

To better understand how the computational cost is distributed in this step, an in-deep time analysis was performed. However, before going into details, a review about how the the original C++ BRISQUE code works is needed.
The BRISQUE algorithm take advantage of the neighbor pixels as described in Figure 3.1. To apply that scheme, the MSCN coefficients are computed over the entire input image. This process yields as output a new matrix (same dimension of the input image) which will be used for the extraction of the first two model features. However, this is only a preprocessing required by the "Pairwise Products and AGGD" step.
The MSCN matrix is involved in a loop executed four times, one for each adjacent pixel direction. The goals of these iterations are many. First of all, a shifted version of the MSCN matrix is computed with respect to the actual direction. This process accesses to the overall input matrix in order to read the MSCN coefficients and write them into the shifted version. A pair-wise multiplication is then performed between the original matrix and its shifted version. The results are then saved using an in-place technique into the shifted version.
Secondly, the fitting process is acting inside the loop. It uses the modified shifted matrix in order to compute the features related to the adjacent pixel directions (four features each).
All the operations performed in the original "Pairwise Products and AGGD" implementation are well explained by the flowchart in Figure 6.3. It should be remarked that the BRISQUE model needs all these operations to be executed twice because of the two-scale image framework. Although the second image scale has been discarded in the model proposed by this thesis, for comparison purposes it will be here considered.

To achieve some improvements with regards to the execution speed, the

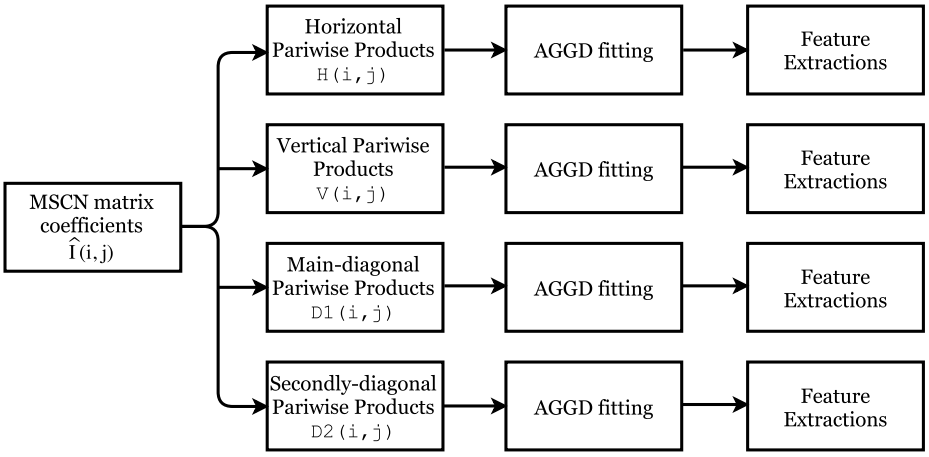


Figure 6.3: Processes performed by the BRISQUE algorithm in order to extract features from adjacent pixels.

time analysis of the "Pairwise Products and AGGD" original implementation is needed. As a consequence, each of its subprocesses was analyzed and their execution times acquired.
Two subprocesses have pointed out to be the most time consuming. They are the "shifting" process and the "multiply and fitting" process, both involved in the first bank blocks (i.e. Pairwise Products) presented in Figure 6.3.
The "shifting" procedure is responsible for creating the shifted matrices from the MSCN matrix. Instead, the "multiply and fitting" performs the multiplication between the original MSCN matrix and its shifted counterparts other than performing the fitting process. Both execution times are listed in Tables 6.4 and 6.5, for computer and smartphone, respectively. It should be noted that in these tables the total time for both processes is almost the 100% of the original time analysis for the same step. This is a proof that the processes are the most time consuming.
The tables point out the almost equal distribution of the total time required by the step between the two processes. This is particularly true on the table related to the computer where the unique difference is a 0.01 for the first image scale.
Furthermore, the time execution boost of the second scale with regards to the first scale is remarkable. In Table 6.4 the time required is 1.19 and 0.32 for the first and second scale, respectively. Instead, in table 6.5 the times with respect to the same scales are 3.39 and 0.87. Both cases show how different platforms could benefit from a filtered and downsampled input im-

age. Despite this fact, as explained in the previous chapter, no quality score relationships exist between an image and its downsampled counterparts.

Image scale	Shifting [s]	Multiply and Fitting [s]	Both processes [s]
1	0.64	0.65	1.29
2	0.16	0.16	0.32
total time	0.80	0.81	1.62

Table 6.4: Time required by the most consuming "Pairwise Products and AGGD" subprocesses on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram and input image in Figure 6.1

Image scale	Shifting [s]	Multiply and Fitting [s]	Both processes [s]
1	1.97	1.42	3.39
2	0.46	0.41	0.87
total time	2.43	1.83	4.26

Table 6.5: Time required by the most consuming "Pairwise Products and AGGD" subprocesses on a Google Nexus 5 with a 2.3 GHz Quad Core Qualcomm MSM8974 Snapdragon 800 processor and 2 GB of Ram. Input image in Figure 6.1.

The previous analysis also suggests the improvements that could be performed on the "Pairwise Products and AGGD" step. Indeed, the "Multiply and Fitting" subprocess can't be easily changed because of the multiplication is needed and the speed up of the fitting process means loosing quality evaluation accuracy. In particular, the latter aspect leads to a reduction in the overall algorithm performances which is in contrast with the thesis goal. Consequently, the possible improvements are focused on the "Shifting" step. The idea behind the enhancement process is related to the fact that shifted MSCN versions share some pairwise coefficients.
All the times a pairwise product matrix is computed (four loop iterations), read operations on the MSCN matrix and write operation on the shifted space are performed. However, the four loop iterations could be reduced to a unique iteration.
Indeed, the improvement process acts filling the shifted versions with the

shared components and terminating the procedure treating the non-shared components.
The positive aspect is that non-shared components are located along the matrix borders which simplify the filling process.
In Figure are shown the MSCN components shared among the four shifted matrix versions, along with the border components. It should be noted that the this procedure permits to access the MSCN matrix only one time (originally were four).

After the previously described improvement has been applied, the time analysis for the "Pairwise Products and AGGD" step was performed again. This process is needed in order to prove the improvement effectiveness.
The results are shown in Tables 6.6 and 6.7 for computer and smartphone, respectively. The enhancements are affective, in fact the overall time in smartphone has been reduced of almost 47%, from $4.26s$ to $2.89s$. Instead, in the computer case the improvement achieve the 27%, from $1.62s$ to $1.19s$. As can be expected, the "Shifting" subprocess contribute entirely to the overall time speed up because the only which has been changed.

Image scale	Shifting [s]	Multiply and Fitting [s]	Both processes [s]
1	0.28	0.67	0.95
2	0.07	0.17	0.23
total time	0.35	0.84	1.19

Table 6.6: Improved time for the "Pairwise Products and AGGD" subprocesses on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram. Input image in Figure 6.1.

Image scale	Shifting [s]	Multiply and Fitting [s]	Both processes [s]
1	0.99	1.36	2.36
2	0.16	0.37	0.53
total time	1.15	1.73	2.89

Table 6.7: Improved time for the "Pairwise Products and AGGD" subprocesses on a Google Nexus 5 with a 2.3 GHz Quad Core Qualcomm MSM8974 Snapdragon 800 processor and 2 GB of Ram. Input image in Figure 6.1.

Finally, a resume of the time improvements for computer and smartphone is shown in Table 6.8 along with its informal complexity analysis.

	Computer		Smartphone	
step	percentage of time	time [s]	percentage of time	time [s]
MSCN	22.40	0.42	33.60	1.59
GGD	11.10	0.19	8.70	0.40
Pairwise Products and AGGD	64.70	1.13	57.70	2.72
total time		1.74		4.73

Table 6.8: Final improved BRISQUE execution time on both computer and mobile device

6.3 BLIINDS-2 Improvements

The most promising models obtained in the previous chapter are characterized by features from the BRISQUE and BLIINDS-2 algorithms. Despite the latter has been provided by the authors with a Matlab implementation, its C++ version is required in order to perform temporal analysis on mobile devices. The BLIINDS-2 algorithm was then rewritten in C++ and optimized. Before discussing the results, a brief introduction about the major defects of the original version and their solutions will be provided.

The Matlab version of the BLIINDS-2 algorithm contains a major inefficiency. Indeed, it recomputes the image patches (DCT transformed) for each kind of feature involved in the final probabilistic model. This task along with the shape fitting process represent the most expensive computational burden.
In the new C++ version, all the BLIINDS-2 DCT patches were initially computed and pooled in a vector. It is involved in all the feature computation processes, avoiding the problem explained before. Moreover, some initial steps are shared among feature computation processes. As a consequence, they can be performed jointly leading to a reduction in the overall temporal cost.
For example, the shape (α) and rho (ζ) features share the initial patch decomposition technique. It transforms each DCT patch into a single column

vector composed by the patch columns one above the other (from left to right).
After the improvements were described, it is necessary to provide a reference starting time for the BLIINDS-2 algorithm. Despite in the BRISQUE paper [3] the BLIINDS-2 performance is stated to be 70 seconds for a single image evaluation, this time is computed with respect to an image from the LIVE IQA database. Accordingly to the technique used in this chapter, the computational costs of the Matlab implementation were computed again for the image in Figure 6.1.
The results of the BLIINDS-2 original Matlab implementation [24] are presented in Table 6.9. It should be noted that the major computational cost is due to the original image (98% of the total evaluation). The *blockproc* and *dct*2 are the two Matlab functions which require the greatest amount of time. As described before, they have been optimized in the C++ version.

step	percentage of time	time [s]
BLIINDS-2 1th scale	98.4	8662.3
BLIINDS-2 3rd scale	1.6	138.9
total time		8801.2

Table 6.9: Execution time of the BLIINDS-2 Matlab version on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram. Input image in Figure 6.1.

The original BLIINDS-2 implementation is too time consuming to be included in a mobile device application. However, the results obtained for the C++ version are encouraging. They are presented in Table 6.10. As it can be seen, the time required by the BLIINDS-2 original image has been largely reduced. Despite being far from employable in a real-time mobile application, the code can be further optimized.

In table 6.10, the most expensive task is due to the four shape features. Indeed, they are computed in the two BLIINDS-2 implementations through an expensive fitting process.
Keeping in mind that the model with the first-eighteen BRISQUE features and the BLIINDS-2 features from the original and third scale (without shape) has demonstrated high correlation with human observers, its computational analysis has been further performed. The results are presented in Table 6.11. It is not a surprise that the temporal cost has been highly reduced. This is a confirmation of the shape features cost. Moreover, the temporal costs with

step	percentage of time	time [s]
BLIINDS-2 1th scale	93.5	286
BLIINDS-2 3rd scale	6.5	20
total time		306

Table 6.10: Execution time of the BLIINDS-2 C++ version on a MacBook Pro late 2011, with a 2,4 GHz Intel Core i5 processor and 4 GB DDR3 1333 MHz of Ram. Input image in Figure 6.1.

respect to the last model on a mobile device are provided in the same table.

	Computer		Smartphone	
step	percentage of time	time [s]	percentage of time	time [s]
BLIINDS-2 1th scale (No-Shape features)	93.0	3.87	93.6	27.55
BLIINDS-2 3rd scale (No-Shape features)	7.0	0.29	6.4	1.87
total time		4.16		29.42

Table 6.11: Execution time of the BLIINDS-2 C++ version without second scale and shape features. Analysis performed on both computer and mobile device. Input image in Figure 6.1.

Chapter 7

Conclusions and Future Work

This thesis has been focused on the development of a No-Reference Image Quality Assessment (NR-IQA) algorithm. In order to achieve this goal, the state-of-art BRISQUE [3] algorithm was taken as starting point to improve upon. An in-deep analysis pointed out many weaknesses and issues related to this algorithm. Based on this study, many experiments were performed which have demonstrated the existence of subsets of predictive features. The last-eighteen features were then removed from the BRISQUE algorithm because they were considered weak.
Following an ensemble-based approach, a new set of predictive features were included in the reduced BRISQUE model in order to improve IQA performances. In particular, these new features were first analyzed and then extracted from the BLIINDS-2 algorithm [23]. They are calculated in the frequency-space with respect to the original image and its downsampled version (reduced by a factor of 4).
The thesis goal was achieved because the proposed algorithm overcomes the quality evaluation performances of the original BRISQUE model. This result was validated through the median SROCC between predictive scores and human judgments, across 60 train-test iterations on the LIVE IQA database [12]. The ensemble-based approach of features from the spatial-domain and the frequency-domain has proved effective for the quality assessment of digital images.
Finally, the time analysis of BRISQUE and BLIINDS-2 was performed on both computer and mobile environments, which has led to an improvement of the overall execution time. This analysis has involved the deployment of the proposed algorithm into a mobile application which has demonstrated to be capable of real-time performances for the quality evaluation of images taken with mobile devices.

Future work will aim at improving two different aspects of the proposed work: effectiveness of quality assessment and time performances.
In the first case, the quality assessment could be enhanced using new features related to distortion introduced by mobile devices. Instead, some improvements with respect to execution time could be achieved with a multithreding version of the proposed algorithm.
Finally, the mobile application developed for this thesis could be part of a collaborative learning platform, where images of heritage sites are mapped and ranked. The platform could be in the Cloud in order to take advantage of scalability, performances and availability characterizing this kind of architecture.

Appendix A

Portello Dataset

A.1 Devices

Remembering that the thesis goal is to develop a NR-IQA algorithm optimized for mobile devices, a small subset of them were involved in the building of a new photographic dataset. In particular, seven devices with different image sensors, focal lengths and resolutions were used in order to capture many photographs. Only a device differ from the others in terms of overall delivered quality, it is a Nikon D7000 Single Lens Reflex (SLR) which represent the high-performance reference device. A summary of the used devices along with their image sensor characteristics is listed in Table A.1.

Device model	Photo Resolution [ppi]	Focal Length [35mm equivalent]	Image Width [px]	Image Height [px]
Nikon D7000 SLR	300	24.0	4928	3264
Galaxy Nexus	72	3.4	2608	1960
iPad Air 2	72	3.3	3264	2448
iPhone 6	72	4.2	3264	2448
Nexus 7 2013	72	3.0	2592	1944
Microsoft Lumia 620	72	0.0	2592	1936
Xperia Mini Pro	72	3.5	2592	1944

Table A.1: Devices involved in the Portello dataset building process.

A.2 Dataset

The photographs included in this dataset were acquired near a touristic site located in Padova (Italy). This site is commonly referred as Portello because of a small port located near an ancient door dated 1519. The dataset will be then refereed as "Portello dataset" in the remainder of this thesis.
According to what has been introduced before, the photographs contained in this dataset were taken with both the reference SLR digital camera and the other low-end mobile devices.
Three main subsets can be derived with respect to the used devices. Indeed, the first and third photographic subsets, respectively in Figures A.1 and A.3 were taken with the Nikon SLR, whereas the second photographic subset in Figure A.2 was taken with both the SLR and the low-end mobile devices.
Despite the fact the first subset contains all photographs from the same scene, it should be note that small differences distinguish those photographs. In order to understand these differences the photograph in Figure A.1 (a) is to consider as reference image for this subset. It was taken with the photographer in standard position which is ideally without any camera rotation.
The photographs A.1 (a - b - c) were taken with sensitive upward and downward vertical translations with respect to the reference image.
Instead, the photographs A.1 (d - e) were taken with sensitive horizontal translations, respectively rightward and leftward. Finally, rotational changes have been introduced in the photographs A.1 (f -g) which are more evident and can be better appreciated with respect to the other distortions.
The second Portello subset was realized using both the reference device and the low-end mobile devices. This approach has lead to many photographs with different dimension and aspect ratios. Indeed, each mobile device has a different photosensor in terms of dimension and resolution.
In order to remove these differences, a post-acquisition cropping and resizing process was performed. We are aware that this procedure can affect the image quality evaluation. However, the main priority for the images in this subset was to achieve the same aspect ratio.
A dimension of 2500x1900 has been obtained for each image at the end of the previously described procedure. Moreover, the photograph in Figure A.1 (a) was also involved in this crop and resize process for comparison purposes.
The last Portello subset contains only two photographs taken exclusively with the Nikon D7000. They will be used with the aim at evaluating the algorithm sensibility to filter techniques.

(a) Reference image

(b) Upward translation

(c) Downward translation

(d) Rightward translation

(e) Leftward translation

(f) Leftward rotation

(g) Rightward rotation

Figure A.1: First Portello subset. Images taken with a Nikon D7000 SLR digital camera

(a) Xperia Mini Pro

(b) Galaxy Nexus

(c) iPad Air

(d) iPhone 6

(e) Nexus 7 2013

(f) Microsoft Lumia 620

Figure A.2: Second Portello subset. Images taken with both the Nikon D7000 and other mobile devices.

(a) (b)

Figure A.3: Third Portello subset. Images taken with the Nikon D7000 SLR.

Appendix B

Existing Datasets

B.1 LIVE IQA Database

The first database here described is the popular LIVE Image Quality Assessment [12] database, in the remainder simply LIVE. Despite the fact that its first release [36] date goes back to 2005, this database is nowadays widely used. Moreover, it plays a key role in this thesis because involved in numerous training and comparison procedures.
The first LIVE release contains images afflicted by two kind of distortions, i.e. ,the JPEG and JPEG2000. These were introduced in a set of 29 high-quality 24 bits/pixel RGB images. Each image (original or distorted) has a typical dimension of 768x512.
A total of 175 JPEG images and 169 JPEG2000 images at different quality levels were created as a result of the generation phase. The bit rates were in the range of 0.150 to 3.336 and 0.028 to 3.150 bits/pixel for the JPEG and JPEG2000, respectively. These were chosen nonuniformly in order to obtain approximately uniform subjective quality score distributions over the entire range.
In order to obtain subjective judgements, a linear scale of values "Bad", "Poor", "Fair", "Good", and "Excellent" was provided to participants. Each image has been evaluated in terms of quality using a score within this scale. Human subjects were mostly college male students, they were seated at a fixed distance from the screen displaying the images. The images afflicted by JPEG and JPEG2000 distortions was vied by 13 to 20 and 25 subjects, respectively.
With the aim of obtaining unbiased results, raw scores of each subject were normalized by their mean and variance. The entire dataset then has been rescaled to fill the range from 1 to 100 and the Differential Mean Opinion

Scores (DMOSs) for each image produced.
In the evaluation process some observers were afflicted by annoyance or fatigue. As a consequence, outlier judgements have been discarded before the DMOS computation. The average standard deviations of the subjective scores for JPEG, JPEG2000, and all images were 6.00, 7.33, and 6.65, respectively.
In the second LIVE release [28] three more distortions were introduced, they are White Noise, Gaussian Blur and Simulated Fast Fading (wireless) channel.
The white noise distortion was introduced to the RGB components of the images after scaling each component between 0 and 1. This distortion was characterized by a standard deviation between 0.012 and 2.0. The three components were clipped and rescaled to the range of 0 to 255.
In the case of gaussian blur a filtering process was adopted. The image components were in fact filtered using a circular-symmetric 2-D Gaussian kernel having standard deviation in the range of 0.42 to 15 pixels.
Finally, the wireless channel distortion was inserted in images already afflicted by JPEG2000 compression. That is, images bitstream was modified by bit errors over a simulated wireless channel. In order to change the distortion's strength, the receiver Signal to Noise Ration (SNR) was varied between 15.5 dB an 26.1 dB. The data rate was fixed to 2.5 bits per pixel for all images at the source.
This database contains a total of 779 distorted images were created by these five distortions and they were evaluated using a single-stimulus process. As in the first version a post-processing was performed in order to align human judgements and producing DMOS scores.

B.2 CID2013 database

Differently from the previous database, the CID2013 [31] database is mainly oriented to address the problem of no-reference image quality assessment. It consists of six datasets of images taken with 79 different low-end cameras or image signal processing pipelines. As a consequence, the reference images have not been included in these datasets.
The CID2013 is intended to represents real-world photographs, e.g., captured with low-end consumer mobile devices. With the aim of evaluating the human quality judgements, a new dynamic reference method was proposed. It combines the properties of both the pairwise comparison method and the single-stimulus method. The test images were presented in a dynamic ref-

erence slideshow consisting of images showed one at a time and delayed by 1000 ms each. In addition to the slideshow, for some datasets an anchor image was displayed on a second monitor. This provides the observers a physical reference to the scale (i.e. score) in each cluster, allowing for the data combinations among image sets.
At the end of the slideshow, observers had a small amount of time for the overall quality scale evaluation of displayed images. Then, the slideshow was shown a second time and observers were asked to evaluate single preferences with regard to each image. The factors to be judged were: sharpness and graininess with a scale ranged in 0-100 and lightness and color saturation with a scale between -100 to 100.
MOS was computed for each image in the datasets, after a process of outliers elimination. Along with MOS coefficients, answers of observers with respect to some of their behavioural aspects were included in the database.

B.3 TID2008 Database

Another widely used reference database is the TID2008 [19]. It is mainly involved in full-reference IQA algorithms and was developed using human observers from three different countries (Italy, Ukraine and Finland).
The images in this database have dimension of 512x384 pixels, in some cases produced after a cropping procedure. The reference images are in total 25, mostly contained also in the LIVE database. A synthesised image was included in the reference group in order to provide adequate testing for metrics intended to work with such kind of images.
The distortions addressed by the TID2008 database are 17, with four intensity levels for each distortion. These have been introduced in the reference images in order to obtain a total of 1685 distorted images.
A multi-tour experimental approach was adopted by the authors to obtain the Mean Opinion Score (MOS) for a given image. Each observer was asked to evaluate 306 pairwise of images generated by only one reference image. This experiment was performed in 9 tours, each consisting in 34 random pairwise images generated from the distorted versions (68 in total) of the reference. When a pair of images was shown to an observer, it had to choose which image was better in terms of quality. The selected image got one point and in the next step only representatives images of the same point groups were used for a comparison.
At the end of the process, images had obtained a score in the range between 0 and 9 which represents the perceptual score. The final results for each

image are presented as MOS computed across human judgments.

B.4 MDID Database

The MDID [29] is another database where distortions are artificially introduced. It contains 1600 distorted images, which are generated from a set of 20 reference images.
The images yield as output are affected by five kind of distortions: Gaussian Noise, Gaussian Blur, Contrast Change, JPEG and JPEG2000. Each distortion has been justified because occurring in some phases of a typical digital image processing pipeline.
Distortions have been introduced using an algorithm which takes into consideration four intensity levels. It takes as input a reference image and it yields as output 80 distorted images. This is the order followed by the algorithm to introduce distortions: GB and CC first, JPEG or JPEG2000 second, and GN last. The JPEG and JPEG2000 compressions have never been introduced together because supposed not to occur simultaneously during the digital image processing pipeline.
In order to acquire human judgements, a pair-comparison sorting scheme was proposed. It is based on the idea that images have an inherited quality score, then it is possible to sort them with respect to that score.
Only graduate students were involved in the experiment. They were 192 and each student was instructed before the test about the experimental setting used to collect human opinions. It consists of a screen with three images, i.e., the reference image and its two distorted versions. The observer was asked which distorted image was perceptually better with respect to the other distorted image. If some difficulties were meet during this evaluation, then the observer had to choose the "equal" judgement.
Since the human judgements were acquired as ranking sequences, a normalization procedure was performed in order to transform the image ranks in the range from 0 to 9. After this step, the processes of outlier detection and subject rejection were applied with the same method proposed in the LIVE database. Finally, the MOS was computed for each image in the database, using the transformed human judgements.

B.5 Live In the Wild Database

With the aim of providing a ground truth database focused on images acquired by real-world digital cameras, the LIVE In the Wild Image Quality Challenge Database [10] has been proposed. Almost all the previously described databases (except for the MDID database) were developed in restricted experimental conditions. Moreover, distortions have been introduced using artificial procedures which is obviously different from the real-world conditions that end-users face.
All these problems have been addressed in [10], which contains images taken with low-end mobile device cameras. These are affected by distortions difficult to analyze and replicate. In fact, the image environments are not restricted to a single laboratory room.
A total of 1162 images are contained in this database. Each of them was acquired with different low-end mobile devices and no limitations were imposed to the end users. As a consequence of this experimental setting, no pristine reference images are available.
A crowdsourced framework for gathering subjective quality scores was developed using the Amazon Mechanical Turk [4] system. This process has involved 8100 unique human observers collecting about 350000 opinion scores. The MOSs were then computed on the images along with experiments about the results consistency.

Bibliography

[1] A. K. Moorthy A. Mittal and A. C. Bovik. *BRISQUE C++ Software Release*. http://live.ece.utexas.edu/research/quality/BRISQUE_C++release.zip.

[2] A. K. Moorthy A. Mittal and A. C. Bovik. *BRISQUE Matlab Software Release*. http://live.ece.utexas.edu/research/quality/BRISQUE_release.zip.

[3] A.C. Bovik A. Mittal A.K. Moorthy. "No-reference image quality assessment in the spatial domain". In: *IEEE Trans. Image Process.* 21.12 (2012), pp. 4695–4708. DOI: https://doi.org/10.1109/TIP.2012.2214050.

[4] *Amazon Mechanical Turk*. https://www.mturk.com/mturk/.

[5] G. Bradski. "OpenCv Computer Vision Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[6] M. Carnec, P. Le Callet, and D. Barba. "An image quality assessment method based on perception of structural information". In: *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*. Vol. 3. Sept. 2003. DOI: 10.1109/ICIP.2003.1247212.

[7] M. Carnec, P. Le Callet, and D. Barba. "Visual features for image quality assessment with reduced reference". In: *IEEE International Conference on Image Processing 2005*. Vol. 1. Sept. 2005. DOI: 10.1109/ICIP.2005.1529777.

[8] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27.

[9] Jan-Mark Geusebroek and Arnold W. M. Smeulders. "A Six-Stimulus Theory for Stochastic Texture". In: *Int. J. Comput. Vision* 62.1-2 (Apr. 2005), pp. 7–16. ISSN: 0920-5691. DOI: 10.1007/s11263-005-4632-7. URL: http://dx.doi.org/10.1007/s11263-005-4632-7.

[10] D. Ghadiyaram and A. C. Bovik. "Massive Online Crowdsourced Study of Subjective and Objective Picture Quality". In: *IEEE Transactions on Image Processing* 25.1 (Jan. 2016), pp. 372–387. ISSN: 1057-7149. DOI: 10.1109/TIP.2015.2500021.

[11] Stephen Heyman. *Photos, Photos Everywhere*. https://www.nytimes.com/2015/07/23/arts/international/photos-photos-everywhere.html?_r=0.

[12] L. Cormack H.R. Sheikh Z.Wang and A.C. Bovik. *LIVE Image Quality Assessment Database Release 2*. http://live.ece.utexas.edu/research/quality.

[13] Q. Li and Z. Wang. "Reduced-Reference Image Quality Assessment Using Divisive Normalization-Based Image Representation". In: *IEEE Journal of Selected Topics in Signal Processing* 3.2 (Apr. 2009), pp. 202–211. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2009.2014497.

[14] *LIBJPEG*. http://ijg.org.

[15] M. Liu and X. Yang. "A New Image Quality Approach Based on Decision Fusion". In: *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*. Vol. 4. Oct. 2008, pp. 10–14. DOI: 10.1109/FSKD.2008.469.

[16] Mery Meeker. *KPCB - Internet Trends 2016 - Code Conference*. http://www.kpcb.com/internet-trends.

[17] A. K. Moorthy and A. C. Bovik. "Blind Image Quality Assessment: From Natural Scene Statistics to Perceptual Quality". In: *IEEE Transactions on Image Processing* 20.12 (Dec. 2011), pp. 3350–3364. ISSN: 1057-7149. DOI: 10.1109/TIP.2011.2147325.

[18] Anush Krishna Moorthy and Alan Conrad Bovik. "Blind Image Quality Assessment: From Natural Scene Statistics to Perceptual Quality". In: *Trans. Img. Proc.* 20.12 (Dec. 2011), pp. 3350–3364. ISSN: 1057-7149. DOI: 10.1109/TIP.2011.2147325. URL: http://dx.doi.org/10.1109/TIP.2011.2147325.

[19] Nikolay Ponomarenko et al. "TID2008-a database for evaluation of full-reference visual quality assessment metrics". In: *Advances of Modern Radioelectronics* 10.4 (2009), pp. 30–45.

[20] A. Rehman and Z. Wang. "Reduced-Reference Image Quality Assessment by Structural Similarity Estimation". In: *IEEE Transactions on Image Processing* 21.8 (Aug. 2012), pp. 3378–3389. ISSN: 1057-7149. DOI: 10.1109/TIP.2012.2197011.

[21] Daniel L Ruderman. "The statistics of natural images". In: *Network: Computation in Neural Systems* 5.4 (1994), pp. 517–548. DOI: 10.1088/0954-898X_5_4_006. eprint: http://www.tandfonline.com/doi/pdf/10.1088/0954-898X_5_4_006. URL: http://www.tandfonline.com/doi/abs/10.1088/0954-898X_5_4_006.

[22] Daniel L. Ruderman, Thomas W. Cronin, and Chuan-Chin Chiao. "Statistics of Cone Responses to Natural Images: Implications for Visual Coding". In: *Journal of the Optical Society of America A* 15 (1998), pp. 2036–2045.

[23] M. A. Saad, A. C. Bovik, and C. Charrier. "Blind Image Quality Assessment: A Natural Scene Statistics Approach in the DCT Domain". In: *IEEE Transactions on Image Processing* 21.8 (Aug. 2012), pp. 3339–3352. ISSN: 1057-7149. DOI: 10.1109/TIP.2012.2191563.

[24] M.A Saad and A. C. Bovik. *BLIINDS-2 Matlab Software Release*. http://live.ece.utexas.edu/research/quality/BLIINDS2_release.zip.

[25] Muhammad Shahid et al. "No-reference image and video quality assessment: a classification and review of recent approaches". In: *EURASIP Journal on Image and Video Processing* 2014.1 (2014), p. 40. ISSN: 1687-5281. DOI: 10.1186/1687-5281-2014-40. URL: http://dx.doi.org/10.1186/1687-5281-2014-40.

[26] K. Sharifi and A. Leon-Garcia. "Estimation of Shape Parameter for Generalized Gaussian Distributions in Subband Decompositions of Video". In: *IEEE Trans. Cir. and Sys. for Video Technol.* 5.1 (Feb. 1995), pp. 52–56. ISSN: 1051-8215. DOI: 10.1109/76.350779. URL: http://dx.doi.org/10.1109/76.350779.

[27] H. R. Sheikh, M. F. Sabir, and A. C. Bovik. "A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms". In: *IEEE Transactions on Image Processing* 15.11 (Nov. 2006), pp. 3440–3451. ISSN: 1057-7149. DOI: 10.1109/TIP.2006.881959.

[28] H. R. Sheikh, M. F. Sabir, and A. C. Bovik. "A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms". In: *IEEE Transactions on Image Processing* 15.11 (Nov. 2006), pp. 3440–3451. ISSN: 1057-7149. DOI: 10.1109/TIP.2006.881959.

[29] Wen Sun, Fei Zhou, and Qingmin Liao. "MDID: A multiply distorted image database for image quality assessment". In: *Pattern Recognition* 61 (2017), pp. 153–168. ISSN: 0031-3203. DOI: http://dx.doi.org/10.1016/j.patcog.2016.07.033. URL: http://www.sciencedirect.com/science/article/pii/S0031320316301911.

[30] T. H. Thai, R. Cogranne, and F. Retraint. "Statistical Model of Quantized DCT Coefficients: Application in the Steganalysis of Jsteg Algorithm". In: *IEEE Transactions on Image Processing* 23.5 (May 2014), pp. 1980–1993. ISSN: 1057-7149. DOI: 10.1109/TIP.2014.2310126.

[31] T. Virtanen et al. "CID2013: A Database for Evaluating No-Reference Image Quality Assessment Algorithms". In: *IEEE Transactions on Image Processing* 24.1 (Jan. 2015), pp. 390–402. ISSN: 1057-7149. DOI: 10.1109/TIP.2014.2378061.

[32] An Vo, Soontorn Oraintara, and Nha Nguyen. "Vonn Distribution of Relative Phase for Statistical Image Modeling in Complex Wavelet Domain". In: *Signal Process.* 91.1 (Jan. 2011), pp. 114–125. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2010.06.014. URL: http://dx.doi.org/10.1016/j.sigpro.2010.06.014.

[33] Z. Wang, E. P. Simoncelli, and A. C. Bovik. "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003.* Vol. 2. Nov. 2003, 1398–1402 Vol.2. DOI: 10.1109/ACSSC.2003.1292216.

[34] Zhou Wang and Eero P. Simoncelli. "Reduced-reference image quality assessment using a wavelet-domain natural image statistic model". In: vol. 5666. 2005, pp. 149–159. DOI: 10.1117/12.597306. URL: http://dx.doi.org/10.1117/12.597306.

[35] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.

[36] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.

[37] J. Xu et al. "Blind Image Quality Assessment Based on High Order Statistics Aggregation". In: *IEEE Transactions on Image Processing* 25.9 (Sept. 2016), pp. 4444–4457. ISSN: 1057-7149. DOI: 10.1109/TIP.2016.2585880.

[38] Yi Zhang et al. "C-DIIVINE: No-reference Image Quality Assessment Based on Local Magnitude and Phase Statistics of Natural Scenes". In: *Image Commun.* 29.7 (Aug. 2014), pp. 725–747. ISSN: 0923-5965. DOI: 10.1016/j.image.2014.05.004. URL: http://dx.doi.org/10.1016/j.image.2014.05.004.