POLITECNICO DI MILANO

Scuola di Ingegneria dei Sistemi
Corso di Laurea Magistrale in Ingegneria Matematica

An approach for detecting global image manipulations

Relatore: Prof. Stefano Tubaro
Correlatori: François Cayre, Kai Wang
(Gipsalab Grenoble)

Tesi di Laurea di:
Valeria Chiesa
Matr. 786455

Anno Accademico 2013/2014

Abstract

In the last years the large diffusion of digital photo/video-cameras and of image retouching software has completely changed how normal people judge the truthfulness of visual content available on Internet. In fact, currently the image forgery is so easy that any image that documents a real event is considered with some suspect. In the last twenty years many forensic methods able to identify doctored images in a blind way have been developed. In general these techniques are based on the principle that any forgery disturbs the intrinsic statistical properties of the considered image with respect to those associated to the original (unaltered) visual content. Often, when an image reveals that post-processing operations or compressions has been applied on it this is interpreted as an indication that some malicious tampering has been applied to it. This is not always true, but for sure who alter an image always applies some global post-process to mask the introduced modifications. For this reason tools able to discover that some global editing or compression has been applied to an image can be used to identify visual contents whose genuineness could be doubtful. Starting from this basic idea, this work, developed at GipsaLab in Grenoble, suggests a scheme to build a classifier able to recognize a set of possible (common) global image modifications. When no one of the considered modifications are discovered this can be considered as a strong hint regarding the fact that the considered image is in its original form. In this work the following global manipulations have been considered: JPEG-compression, Gaussian random noise addition, median filtering, average and sharpening filtering. In particular from the considered image a set of feature (calculated in the pixel and DCT domain) are evaluated. They are the input to a one class classifier targeted to distinguish original versus altered (in a malicious or innocent way) images. Simulations on a large set of real images show that almost all considered modifications are recognized with accuracy higher than 85%. Comparisons with other methods presented in the literature have shown the quality of the proposed approach.

The work described in this thesis was developed throughout a five months internship financed by GipsaLab, a research center in Grenoble, France. To attest the time spent working in this laboratory, a testimony from the professors involved in the project is reported.

Abstract

La vasta diffusione di macchine fotografiche e videocamere digitali e di software che permettono il ritocco di immagini ha diminuito la fiducia delle persone nei confronti delle informazioni visive. La contraffazione di immagini è oggi così semplice che qualunque documentazione visiva di un evento reale sarebbe da considerarsi dubbia. Negli ultimi vent'anni sono stati sviluppati diversi metodi forensi in grado di identificare falsificazioni di immagini senza avere alcuna informazione a priori. In generale queste tecniche sono basate su un principio che ogni contraffazione cambia le proprietà statistiche intrinseche all'immagine ripetto a quelle associate ai contenuti originali. Spesso il comprimere un'immagine o sottoporla a determinati processi viene considerato indice di alterazione per fini illeciti. Questo non è sempre vero, ma sicuramente l'alterazione dei contenuti di un'immagine è sempre seguita da una manipolazione globale per nascondere le modifiche apportate. Per questa ragione, strumenti capaci di individuare la presenza di una correzione o una compressione possono essere usati per verificare l'autenticità di un'immagine i cui contenuti siano stati messi in discussione. Il presente lavoro, sviluppato presso il GipsaLab di Grenoble, suggerisce uno schema per costruire un classificatore capace di riconoscere un insieme di possibili (comuni) alterazioni globali dell'immagine. Le modifiche prese in esame sono in particolare la compressione JPEG, l'aggiunta di rumore gaussiano, l'applicazione del filtro mediano, l'applicazione del filtro medio e lo sharpening; si può assumere che la loro assenza possa essere considerata un forte indizio a favore dell'originalità dei contenuti dell'immagine. In particolare, dalle immagini è estratto un insieme di features (calcolate nel dominio dei pixels e della DCT), utilizzato poi come input di un classificatore One-Class mirato a distinguere le immagini originali dalle altre. Le simulazioni fatte su un grande numero di immagini reali mostrano che quasi tutte le modifiche considerate sono riconosciute con un'accuracy superiore all'85%. Il confronto con altri metodi presenti in letteratura mostra la qualità dell'approccio proposto.

Il lavoro descritto in questa tesi è stato svolto durante uno stage finanziato dal centro di ricerca GipsaLab a Grenoble, in Francia. A testimonianza del periodo trascorso presso tale laboratorio viene riportata la valutazione da parte dei professori che hanno seguito questo progetto.

Grenoble
November 28, 2014

Subject: Evaluation of Valeria CHIESA's work.

To Whom it may concern,

Valeria CHIESA has been an intern in GIPSA-Lab during the last semester of the academic year 2013/2014. She worked under the supervision of Kai WANG and myself on the subject of universal forensics. The goal is to find a small set of relevant features allowing for tamper detection of digital images.

Valeria has been an outstanding student and her work is very valuable to us. She has shown many of the required qualities for a successful work in a lab: she is creative, imaginative, rigorous and she can handle quite a heavy load of work.

Valeria has not only been able to implement and test the initial ideas we had, she has also been quickly able to propose and design new features we did not think of in the first place. Each time, her proposal was backed both with theoretical justifications and experimental validation.

It has been an immense pleasure to work with her and she always had exciting results to show during our weekly meetings. We can only hope to have more students like her in the future.

Please note that we are looking forward to write a paper with her on the work she carried out during her internship.

Kind regards,

Dr. Kai WANG
CNRS Researcher
GIPSA-Lab

Dr. François CAYRE
Assistant Professor
Grenoble-INP – Phelma

GIPSA-Lab
11, rue des Mathématiques
Grenoble Campus – BP46
F-38402 St-Martin d'Hères CEDEX
France

☎ +33 4 76 82 63 78
✉ francois.cayre@grenoble-inp.fr
www balistic-lab.org
Office D1192

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the invention of photo cameras, images have assumed a leading role in information sharing and recording. People are encouraged to trust on the presumed objectivity of images as they can transfer a complete visual scene to any time in the future. As we thinks that images deteriorate their information much slower than an average person's memory, it becomes of crucial importance knowing if the delivered information is actually true. Therefore the authenticity of images has become a discussed issue of primary importance in many fields, like in political or juridical environments. In many occasions image forgeries have been applied to promote false information and, even worse, for illegal purposes. In the film era this kind of modifications was more complicated and not accessible to everyone. Nowadays modifying an image after acquisition is rather typical and accessible to everybody thanks to the mass diffusion of digital cameras and personal computers [1], [2].

As a direct consequence, the need for an authenticity test is continuously increasing to check and determine whether or not the content of an image can be still trusted to be as it was in its origin. Image forensics is the field of science whose goal is to investigate the history of an image using passive (blind) approaches.

Compared with the authentication based on digital watermarking, image forensic methods aim to assess the authenticity of an image in a passive and blind way, thus without prior information embedded in the image, i.e. the digital watermark. The term forensic clearly comes from the juridical environment; however the research in this field goes well beyond the mere problems strictly related to forgeries. As a matter of fact the entire image history could be of crucial importance as many innocent modifications can be adopted for hiding illegal ones.

This thesis is focused on the subject of global modifications with the aim of detecting if a RAW image has been changed since its acquisition. Global modifications are the ones which apply to each pixel or pixel group the same way, regardless of the information content of the image. Examples are filters or compression algorithms

and those kinds of transformations commonly adopted by the majority of image processing software. Apart from a few cases like in particular hiding by quality worsening, these transformations are not referred to as forgeries, because they are not characterized by some local and content dependent modification such as cut-and-paste or move-and-paste operations. Notwithstanding this fact we believe them of crucial importance in image history reconstruction, either for their widespread use or their potential application in forgery covering [1], [3].

Most of existing image forensic methods concerning global modifications can only detect a specific type of image manipulation, or reveal a specific type of processing operation that were applied [4, 5, 6, 7, 8, 9, 3, 10, 11, 12]. Because the recognition of all possible manipulation would be impossible, we focused on five different modifications: JPEG-compression, addition of random noise and application of median filters, average and sharpening filters. Starting from these processing operations, features are extracted and the decision fusion strategy is performed at feature level. The technique as presented is easily generalizable to a wider class of modifications since a one-class classifier is trained on original images only [13], [14]. The generality of the technique is clearly evident as the only definition needed for the training step regards pristine images and not the type of process inducing the modifications.

We believe that our technique has to main advantages. First the accurate and careful study of single modifications and second the adoption of a training scheme based on original images only. We always preferred to summarize the information of each modification in few, but very effective features in order to capture the essence of each process. Our approach to the problem, based on statistical analysis methods, has been able to produce a restricted set of features, that well separates original images from modified ones in feature space. This is the prerequisite for the second step where the volume space enclosing original images should be as distinct as possible from the volume of the modified ones. That is clearly the best condition where the one-class classifier can be trained with pristine images, targeting the only feature-region of interest. As consequence not only we rely on good identification for the single selected problems but also we allow features to interact together in characterizing original images at best, hopefully helping separating processed images beyond the purpose they were planned for.

With this goal, five characteristic values have been identified based on properties of considered modification. Classifiers able to process the five values are built and results are compared with the state of art.

The thesis is divided into four parts. In the first there is the description of the post-processing algorithms we want to distinguish. The second chapter is focused on the explanation of the most relevant up-to-date analysis methods and the new

features developed along the thesis work. Then in the third chapter the adopted decision functions are presented as the techniques used to organize the extracted features in a single classifier. Finally, the last part is dedicated to results and to a comparison with the state of the art methods.

Chapter 2

Digital image forensics analysis

2.1 Background

Thanks to the diffusion of cheap and affordable softwares for image processing, doctored images can be easily found in our daily life and have been used, for instance, in advertising, politics and personal attacking.

Today visual digital objects might go through several processing stages, from enhancement of quality to tampering of contents. This situation highlights the need for science to answer two main questions: which is the history and the origin of an image. The goals of digital image forensics and watermarking are to reveal the acquisition device and to investigate the post-acquisition processes applied.

For this purpose several solutions have been proposed by the research community in more than twenty years. The methods are divided in two categories, active and passive.

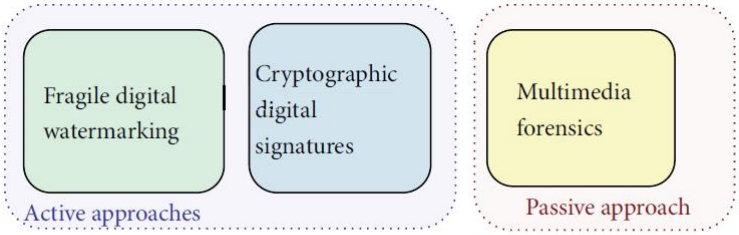


Figure 2.1: Possible approaches for the assessment of the history and credibility of a digital image. [2]

Active methods are less common than passive ones because they require trustworthy cameras. These devices should embed a digital watermark or digital signature in images at the time of the acquisition. Any later modifications alter the digital watermark or digital signature. Among the several drawbacks of active techniques we should mention that digital cameras have to be equipped with an appropriate chip. The implementation of that kind of camera requires the definition of a standard protocol that is not easy to achieve.

On the other hand, passive methods can be applied when no prior actively embedded information about an image is available. Forensic techniques are based on the idea that most, if not all, image-processing tools leave some traces onto the processed image. Hence the presence of these traces can be investigated in order to understand whether the image has undergone some kind of processing or not [2].

The life of a digital image can be divided into three main steps: acquisition, coding and editing. Both the acquisition and coding phases are implemented inside the camera, meanwhile image editing can be applied outside the camera thanks to software tools.

As said above, each process leaves, on the image, a trace that forensic analysts study. The fingerprints left by the acquisition phase are useful in order to trace the origin of image. In fact, although the pipeline is common for most devices, each step is performed according to specific manufacturer choices and each acquisition device is characterized by some imperfections and intrinsic properties.

This so called coding phase is, as mentioned, characteristic of each camera, but typically defaulted to end with a JPEG-compression before saving images on a non-volatile memory. Nevertheless some situations require that no lossy operations are to be done on an image and in this work we will start from pictures acquired by the camera without the further JPEG-compression step. In addition to the traces left during the acquisition phase there could be more inconsistencies or artefacts perhaps associated with tampering or forgeris.

Image editing can be of two different kinds: innocent or malicious. The most important instances and common studied malicious modifications set are the copy-move attacks and cut-and-paste attacks [15, 16]. Usually, the so-called innocent editing are enhancements computed in order to increase image quality. Despite the name "innocent", enhancement of images is not always allowed also because those kind of editing can be applied to hide tampering or malicious modifications, [17], [10] altering or reducing the visual information.

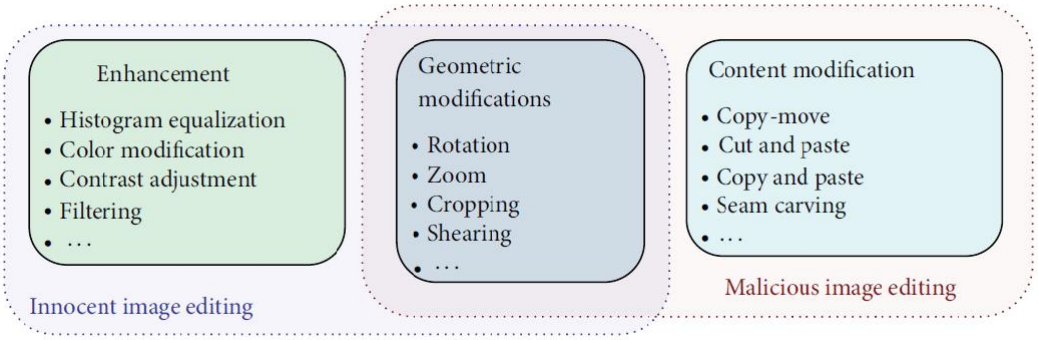


Figure 2.2: Type of editing operators. [2]

A field close to image forensics analysis is *steganography*. Steganography is defined as the art of hiding information in images. As opposed to watermarking, the aim is sending secret massages. Only someone instructed on the algorithm to read back the message would be able to make it visible again.

The hardest aim of steganalysis is to discover if an image contains hidden information. Blind steganalysis methods are often developed without directly targeting the hidden message, but rather, by approaching the problem without any hope of retrieving the embedding mechanism [18, 13, 19, 20]. Many techniques analyze certain statistical properties with the only purpose of identifying those images where a message is concealed. For this reason, algorithms elaborated for steganalysis can be used in forensic analysis in order to recognize if an image has been subjected to editing [21].

The statistical model found by Lyu and Farid [13] based on wavelet decomposition is an example, as the SPAM features elaborated by Pevny et al. [18], that we will use to compare our work with.

Meanwhile forensics algorithms are developed, techniques able to defect existed forensic methods are developed. In forensic analysis knowing the limit and pitfalls of methods is important for new improvements. Parallel to the study of forensic techniques, *Counter-forensics* or *anti-forensics* analysis are explored to attack and to show the limitation of forensics methods [21], [17] that stimulate further studies.

Usually the first step in creating a tamper detection tool is features extraction. Digital images are matrices of pixels, so they contain a huge quantity of data as big as the value of all the pixels. An RGB image with a size of 512x512 is composed of 786432 8-bit integer values. For this reason working directly on images is neither clever nor useful. Besides features extraction, there is a thorny procedure that

starts with the study of the problem under investigation. Depending on the goal of forensic analysis, the most significant properties capable of distinguishing original images from modified ones can be very few. Thus a feasible solution is to summarize each image with appropriate values so as to express its information through a vector far smaller in dimension than the original matrix.

The second step is the organization of the feature vector in a classifier. It is a structure that, conveniently trained, is able to discern between different categories, like in our case, between pristine images from others.

Very often, the creation of forgeries involves more than one processing tool [22]. This suggests researchers to analyze images with several tampering detection tools. This consideration entails that a structure capable of considering different tools should be defined. There are basically three kinds of approaches to tackle the problem. In the first case fusion is performed at the feature level: features extracted by each tool are concatenated in one space only and afterwords are used to train a global classifier. The second approach is to consider the output provided by tools and fuse them. The last case consists in fusing binary answers and is called fusion at the abstract level.

The first approach is the most used but it has drawbacks: usually this kind of analysis involves a large number of features, making feature level fusion computationally really expensive and a large number of features requires having a large dataset, which is not always available. Moreover the definition of the training set can be complicated because it should be representative of the whole dataset. On the other hand, fusion at the abstract level, discarding a lot of information, involves the opposite problem.

Usually, forensic methods treat detection of forgeries, i.e. manipulations that change image content [23], [24]. Nevertheless this work is devoted to identify a set of global image modifications that in principle could be considered "innocent" like coding or contrast enhancement but this type of modification can be also considered as something that could be used to mask some other "more malicious" manipulation of the considered image. By "pristine" or "original" images we mean images that was never subject to compression or to any processes.

Obviously in the case where the image is saved in JPEG format to investigate on its possible compression is not useful. Some format use a lossless data compression techniques as the Tagged Image File Format (TIFF) but include the header tags (size, definition, image-data arrangement, applied image compression): from this format knowledge about the previous history of the considered image can be extrapolated. However not always these information are correct or available. In particularly in this work, without losing in generality, we analyze 256-level grayscale

images saved in Portable Network Graphics (PNG) format, a kind of lossless compression.

We principally focused on five editing operations that we believe significant: JPEG compression, addition of Gaussian random noise and three kinds of filters, two low-pass and one high-pass. In fact, we want to range as much as possible in the set of all global editing modifications. We mention the most popular compression technique with the hope that our method can be expanded to others. Addition of noise is studied in the case of Gaussian distributed random noise, but it is a hypothesis that can be changed with other distributions. Filters have specifically different behavior: the average filter is a linear smoothing process, it leaves most unchanged low-frequencies and muffles high-frequencies. The median filter is not linear and represents an open challenge for forensic analysts. The sharpening filter has the opposite functionality and empathizes high frequencies to make edges more evident.

In order to build a detector capable of distinguishing these modifications from the original images, we analyze the considered processes.

2.2 JPEG compression

JPEG-compression is one of the most popular lossy-compression methods for digital images. This kind of encoding is referred to as lossy because during the compression some pieces of information are lost and it will never be possible to reconstruct the image as it was in origin. Most standard cameras have implemented JPEG-compression code in their software or they can even produce only JPEG-compressed images. Nevertheless our base scenario would assume RAW images as reproducing the data as acquired by devices and digitalized in 8-bit. This is the case of professional imaging where superior quality is required, but it can be applied to files that will undergo multiple edits in raw or bitmap format.

The effect of JPEG-compression can be deep and can easily interfere with forensic methods. The compression is able to destroy artefacts left by tampering or forgeries [24], to interfere with camera identification [25]. Moreover, the presence of double compression is often an evidence of deepest modifications. Therefore the study of the history of JPEG-compression applied on images is of primary interest. Aside from the degradation of information, some traceable artefacts come out leaving a detectable pattern on the picture. Thus many researchers have focused on forensic analysis and anti-forensics in this direction. This is, for instance, the case described by Fan et al. [17]. In fact starting from the algorithm proposed by Fan and De Queiroz [5] to detect JPEG-compression based on quantization of DCT coefficient

and blocking artefacts, Stamm et al. elaborate an anti-forensics method [3] able to fool Fan and De Queiroz's work adding a dithering signal. Later Valenzise at al. [10] propose a counter anti-forensic approach that studies the degradation of the image, made useless by Fan et al. [17], that improves image quality. Other authors work at the same time on Stamm et al. paper as Lai and Bohme [11].

2.2.1 JPEG-compression coding

JPEG-compression takes its name by *Joint Photographic Experts Group*, a commission created in the first years of the 90' to set standard for picture coding.

JPEG-compression is based on two fundamental ideas:

- Human eyes are more sensible to intensities that changes with low spatial frequency with respect to changes at high frequency.
- Quantization of distributions decreases the amount of information to save.

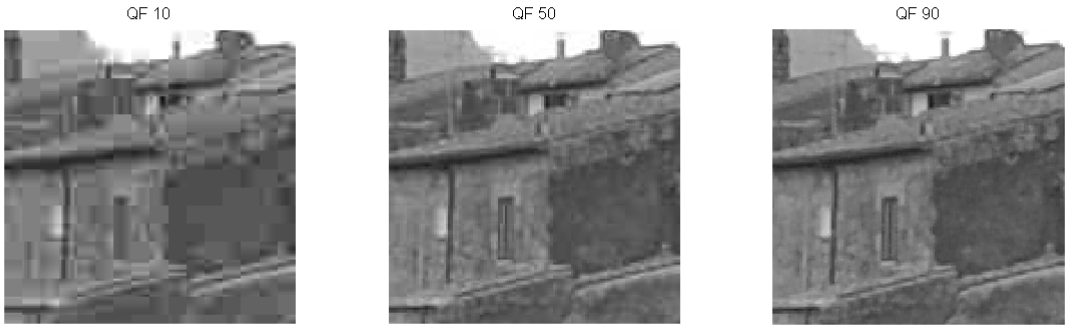
In JPEG-compression algorithm, an image is split into 8x8 pixel blocks and each block is converted into frequency domain representation using a two dimensional type-II Discrete Cosine Transform. The output of this transformation is a matrix whose first top left element is the DC component (Direct Current component) associated with the null frequency oscillation while all the other components are coefficients related to the periodic cosine in the 8x8 window. As the average of those cosines is null, in the 8x8 matrix the respective amplitudes of the transformation are called AC components (Alternating Current).

Typically for natural images most of information is stored in low frequencies, whose coefficients are placed around DC component. Edges and textures are represented in high frequencies, whose coefficients are placed in the opposite corner of matrix.

In the JPEG compression algorithm each component is divided by a different constant, depending on the quality factor and not defined by the standard quantization table t. In table 2.1 an example of quantization table and the operation needed to compute quantization are shown. The quality factor is an integer value between 1 and 100 that describes the level of compression: if the quality factor is equal to 100, the image is close to the original; if the quality factor is equal to 1, the image loses a huge amount of information leaving visually untouched large areas of the same colour only. The quantization represents the lossy operation of the algorithm, where the signal is reduced into fewer bits than the original. Obviously the image quality decreases, but often that is an acceptable price to pay for low dimension in terms of

bytes. Indeed it becomes possible to try and preserve the quality of low frequency amplitude while removing part of information contained in high frequencies, thus without impoverishing the image in a visible way.

(a) *Original image.*



(b) *Example of artifacts for different JPEG-compression quality factor.*

Figure 2.3: Example of artifacts for different JPEG-compression quality factors

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

Example of quantization table for a quality factor of 50. Quantization coefficients in bottom right corner are higher than those in top left corner.

$$Table_{(QF)} = \begin{cases} \lfloor t \times \frac{50}{QF} + \frac{1}{2} \rfloor, \ 1 \leq QF < 50 \\ \lfloor t \times (2 - \frac{QF}{50}) + \frac{1}{2} \rfloor, \ 50 \leq QF \leq 100 \end{cases} \tag{2.1}$$

Once the lossy operation is completed, quantized coefficients are then arranged and encoded with a lossless compression method, a variant of *Huffman Code*.

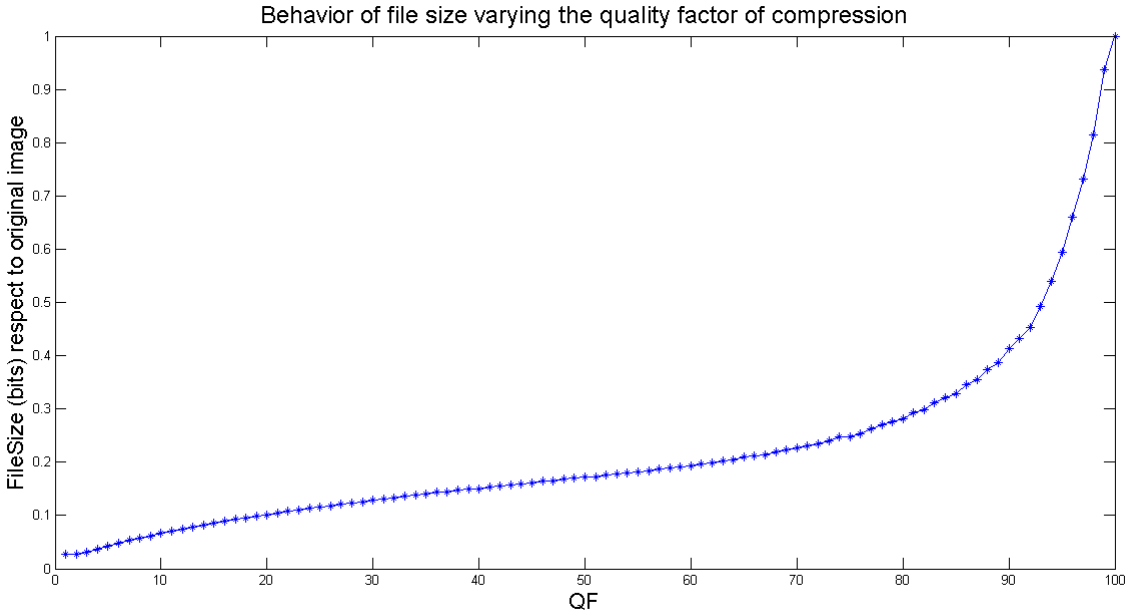
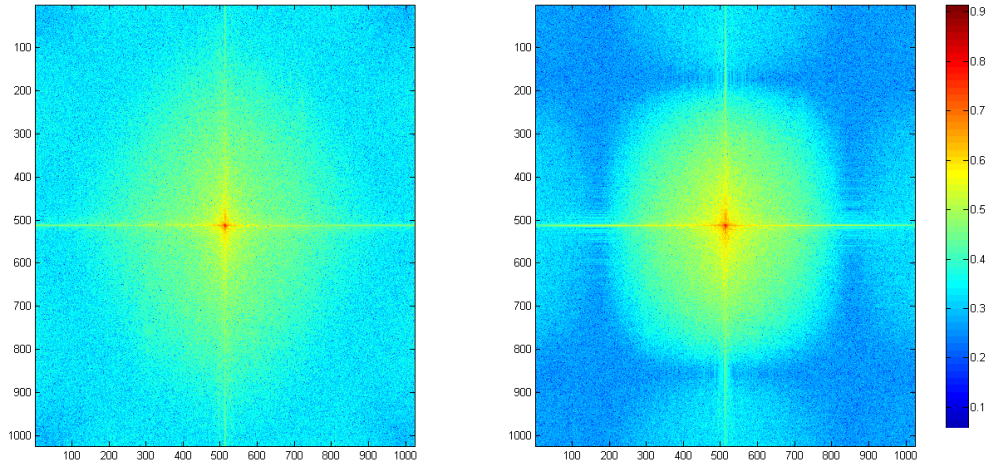


Figure 2.4: Number of bits needed with respect to size of original image to save a JPEG-compressed image for different quality factors.

2.3 Average filter

The acquisition of an image by a camera is often corrupted by random variation of intensity or illumination. The camera itself produces undesirable artefacts due to electronic or optical noise. For this reason many techniques are developed in image processing to improve image quality and to try to reduce random noise.

(a) *On the left original image, on the right the same image subjected to average filter 3x3.*



(b) *On the left normalised absolute value of 2-D Fourier transform amplitudes of original image, on the right normalised 2-D Fourier transform of filtered image (logarithmic scale)*

Figure 2.5: Example of average filtered image and its Fourier transformation. The low-pass effect is particularly evident in Fourier representation.

Linear smoothing filters are one solution to the problem of random noise reduction. Average filter (or mean filter) is one of the simplest smoothing techniques. This method is often adopted for its low computational cost even if it tends to blur sharp edges and to be influenced by outlier pixel values.

Computation of average filter is performed as follows. The value of each pixel is substituted by the value of the average of the pixel itself and the neighboring ones. Depending on how and which neighbors are weighted determines the shape of the so-called window function which is convolving the image. The average filter can be

implemented as a convolution of an image with a linear convolution mask. In this way, applying an average filter corresponds to evaluating translating convolution mask to each pixel in the image.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Table 2.1: Convolution mask for an average filter 3x3.

2.4 Gaussian random noise

The addition of random Gaussian noise can be used in order to hide some statistical properties due to image tampering or to deteriorate fingerprints [15],[3],[23]. In fact, many different sources of noise exist during the image acquisition process: poor illumination or high temperature, imperfection in electric circuits or material are all causes of random noise in digital images. Because of difference between various acquisition devices it is possible to consider random noise as a fingerprint that characterizes the camera.

Random noise transformed in Fourier space preserves the same characteristics and appears as random noise in frequency amplitudes. Moreover it leaves more traces in high frequency amplitudes due to the favourable signal to noise ratio. As information is concentrated in low frequencies, an additional random noise can be harder to distinguish as signal amplitudes are big, while it can be easier in high frequencies where the random value is added to a very tiny one. Since the grid of JPEG compression deals with the high frequencies of the image, we developed features for both JPEG and noise detection in parallel. This should not come as a surprise as noise could appear as additional information in the images, while at the same time our JPEG detection strategy base of information measurement required to be tested for robustness and improvements. Further details about these features and their relation with the information theory can be found in the following chapter.

The addition of Gaussian random noise in images is a simple procedure: to every pixel is added a value sampled from a Gaussian distribution. Let $x_{i,j}$ be the value of pixel in i, j position and $x^{noi}_{i,j}$ the same pixel after Gaussian random noise addition, then we have

$$x^{noi}_{i,j} = x_{i,j} + y \tag{2.2}$$

where

$$y \sim N(\mu, (\frac{\sigma}{255})^2)$$

Variance is normalized by a factor of 255 because pixels are described by 8 bit integer taking values between 0 and 255. In this way 1 is associated to variance corresponding to the maximum value.

(a) *Original image.*

(b) *Noisy image (variance 0.005).*

(c) *Noisy image (variance 0.0005).*

Figure 2.6: Example addition of Gaussian random noise in an image. The addition of Gaussian random noise that we want to consider is almost invisible to naked eye.

2.5 Median filter

Median filter is adopted for noise reduction, especially salt and pepper or impulse noise because of the robust independence from extreme values. The computation of the median value is a non-linear process as the sort algorithm cannot be represented in signal-independent matrix form. As opposed to local averaging operation, this filter does not tend to blur sharp discontinuities in intensity values of an image.

In the past a big variety of techniques has been developed in order to recognize the presence of the median filtering in an image. Many methods are based on statistical properties of differences between pixels values [6], [4]. Other scientists study the probability of zero values on one-order difference map of texture regions [8], or analyze the histogram of first order difference of image [9]. Often a Markov

Chain model seems to be helpful because of its ability to explain the relationships between adjacent pixels [18].

Figure 2.7: Example of median filtered image with a 5x5 window. The effect of median filtering is almost invisible to naked eye.

The *median* value of a distribution is the value that separates the higher half of a distribution from the lower half. This value is often preferred to the average for major stability in the presence of outliers. On the contrary, average value is influenced by all samples, especially from unusual ones. The median value is defined by the following formula:

$$med = \{x | P(X < x) = \frac{1}{2}\} \tag{2.3}$$

In discrete domain *median* value is the value that, after sorting all samples, is located at the central position.

As for average filtering each pixel is processed with the near ones. Together they constitute a sequence of numbers whose median value can be easily calculated and afterwords to the original value of the pixel is substituted by the median value. Typically the operation takes place on square windows with odd-side dimensions built around the pixel.

2.6 Sharpening filter

Sometime, in order to stress the edges in an image, a sharpening filter is applied. The aim of a sharpening filter is to enhance high frequencies so that edge and small irregularities become more evident. In fact, as information about big uniform areas is stored in low frequencies, the details are coded in high frequencies. The sharpening filter can be thought of as composed of the sum of the image itself together with the version of the image processed with a high-pass filter. The identity operation preserves the image content stored in the low frequency range, while the high pass filter is applied for the sharpening effect.

In this work the high pass filter window function is calculated as the identity filter minus the window function of a Gaussian filter.

The shape of a Gaussian curve is controlled by a parameter σ related to the variance of the curve. The expression for the filter is as follows

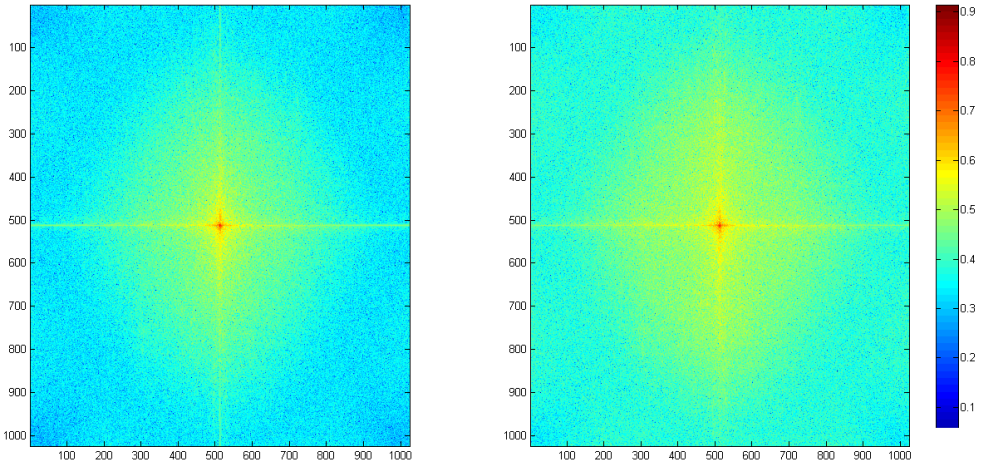
$$G(x,y) = -\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} + \delta(x^2+y^2-r^2) \tag{2.4}$$

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Table 2.2: Mask of a 5x5 window of Gaussian filter.

(a) *On the left original image, on the right the same image subjected to sharpening filter (variance 1.5).*



(b) *On the left normalised absolute value of 2-D Fourier transform amplitudes of original image, on the right normalised 2-D Fourier transform of filtered image (logarithmic scale)*

Figure 2.8: Example of sharpening filtered image and its Fourier transformation. The enhancement of high frequencies is particularly visible in Fourier domain.

Chapter 3

Features

This chapter deals with the first milestone of the thesis, which is dedicated to the description of strategies undertaken to detect modified image. This preliminary work paves the way for recognizing whether or not an image could be still considered as pristine, i.e. never compressed and never post-processed, with respect to a few isolated problems and the background strategies are presented as the fundamental building block of the second part of the thesis, where the implementation of global classifiers is actually realized. As the title suggests the core of detection relies on what in literature is referred to as features. A feature is the output of a function of pixel values of an image which can be used to reveal some properties or, in our case, the presence of modifications. The goal of the features extraction is to elaborate in a clever way the entire information coming from the pixel matrix in order to reduce it to a few but very effective numbers. These values are then considered and carefully studied to build the classifier, as if the entire information of the image could be summarized by those values in the analysis of selected problems.

In this chapter are described first the procedures followed to create the three feature sets proposed in [5], [18], [4] and their limitations are analyzed. Then, the five scalar features proposed by this thesis and the reasons for their choice are illustrated. The considered image alterations are analyzed and studied in order to understand the main characteristic that marks one particular modification. Once a distinctive trait has been found, which allows to distinguish pristine image from the others, it is possible to build a function of image whose output is a value that gives some information about the characteristics under investigation.

3.1 Feature sets in literature

Before to describe the new elaborated features, three feature sets from literature are presented. The first one is composed of only one feature and it is focused on JPEG

detection [5]. The second is a huge dimensional feature set used in steganalysis to recognize the presence of embedded messages. The third is based on feature extraction in difference domain and it is created to detect median filtered images. We believe that all these approaches can be compared with ours, the first only in the specific case of JPEG-compression, the others, thanks to their generality, with the entire our feature set.

3.1.1 Example of JPEG detection

In the article [5] a method to detect JPEG compression in a bitmap image is described and how to estimate quantization table of an eventual compression is shown.

JPEG compression usually creates blocking artefacts in images because of splitting in JPEG algorithm. As many other methods [17], this approach is based on the presence of these blocking artefacts.

In this scheme the block grid is known. They define a simple function that has four pixels as input. The idea is to compare the result of this function where the pixels in the middle of the block are used as input with the value obtained where the pixels in the crossroad of blocks are used.

A	B							
C	D							
				E	F			
				G	H			

Table 3.1: Example of 8x8 pixels block. {A,B,C,D} indicate pixels in the crossroad, {E,F,G,H} indicate pixels in the middle.

The function used is

$$Z'(i,j) = |A - B - C + D|$$

$$Z''(i,j) = |E - F - G + H|$$

For both Z' and Z" normalized histograms are computed and are compared to each other. As a measure for comparison the energy difference between histograms is used. Let H_I be the histogram of Z' and H_{II} the histogram of Z'', the energy is evaluated as

$$K = \sum_n |H_I(n) - H_{II}(n)| \tag{3.1}$$

This method is based on the knowledge of the position of the JPEG grid. For this reason if the dataset includes cropped or pasted images, the algorithm becomes useless. However it is possible to suppose that the difference between pixels would be maximal if the pixels belonged to different blocks. Let $y(m, n)$ be the value of the pixel in position (m, n) and $\{(p, q)|0 \leq p \leq 7, 0 \leq q \leq 7\}$.

$$E_{pq} = \sum_i \sum_j |y(8i+p, 8j+q) - y(8i+p, 8j+q+1) - y(8i+p+1, 8j+q) + y(8i+p+1, 8j+q+1)| \tag{3.2}$$

So, in order to find the alignment of blocking, the grid origin is chosen where (3.2) is maximum.

The described algorithm is tested by authors and the results show that they can detect most compressed images with QF as high as 90 .

From our point of view the weak points of the technique are twofold. Since the 8x8 grid is expected to be detected it would be preferable to use the whole information, without restricting the analysis to selected 2x2 matrix subsets. Moreover the authors use the JPEG grid specification from one side, but from the other they limit the analysis in real space as opposed to what happens in the compression scheme, where quantization is performed in DCT space. We believe the problem of detecting JPEG coding as well as other compression processes can be approached in a completely different way. Since the goal is to detect a loss of information, the loss should be made evident and subsequently the information measured. The first step of this strategy would be for example to process the image in order to bring the data in the same situation right before the information losses. The second step is then to use a statistical method to evaluate the remaining amount of information. Any slight modification in the process of the first step leads to an apparent increase of the information contained in the image, thus creating a sharp minimum easy to identify. The job of the forensic analyst is, as a consequence, the development of a robust algorithm to perform the first step, i.e. transforming the image so as to bring the

loss of information into sight. In the case of JPEG compressed images, this step is the DCT transformation with the correct grid. Clearly it is exactly in that situation that any measurement that attempts to evaluate the remaining information inside the image will show the actual losses. We will discuss our proposed technique in details in next section.

3.1.2 Subtractive pixel adjacency matrix (SPAM)

In [18], the main idea is that dependence between adjacent pixels in an image is violated by steganographic embedding. In fact, stego noise is usually independently distributed and independent of the cover. So the authors provide a model for the difference between adjacent pixels based on first and second order Markov chains.

In literature this set of features is probably the most similar to ours thanks to its characteristic of detecting a lot of properties inside an image. Moreover, their adoption of a large number of features helps catching the properties of a dataset.

Instead of studying the histogram of pairs or larger groups of pixels, they prefer to analyze histogram of difference between adjacent pixels. This is due to the curse of dimensionality that one can encounter because of the high amount of bins in histograms of pairs. Moreover this kind of histograms can be influenced by noise or image content.

In order to avoid these problems, the authors assume that difference between adjacent pixels is not dependent on pixels value, so the joint distribution of two adjacent pixels can be approximated as the product of differences distribution and pixel value distribution. Furthermore they proved that the mutual information between these two variables is quite low, confirming hypothesis of independence.

$$P(I_{i,j+1} = k \wedge I_{i,j} = l) \approx P(I_{i,j+1} - I_{i,j} = r)P(I_{i,j} = l) \tag{3.3}$$

The authors choose to model differences between adjacent pixels as a Markov chain. Based on the histogram of differences they focused themselves only on differences in a small arbitrary range $[-T, T]$.

The features extracted from this model are calculated by averaged sample Markov transition probability matrices. Firstly difference arrays are computed for each direction $\{\leftarrow, \rightarrow, \uparrow, \downarrow, \nearrow, \searrow, \swarrow, \nwarrow\}$ and then the first-order SPAM features are evaluated.

$$D^{\rightarrow}_{i,j} = I_{i,j} - I_{i,j+1} \tag{3.4}$$

$$M^{\rightarrow}_{u,v} = P(D^{\rightarrow}_{i,j+1} = u | D^{\rightarrow}_{i,j} = v) \tag{3.5}$$

$$u, v \in \{-T, ..., T\}$$

In order to decrease further dimensionality of the feature set, the assumption of image symmetry with respect to mirroring and flipping is made. So vertical and horizontal matrices can be averaged as the diagonal ones in order to compute the final features set.

$$\begin{aligned} F_{1,\ldots,k} &= \frac{1}{4}[M^{\rightarrow}, M^{\leftarrow}, M^{\uparrow}, M^{\downarrow}] \\ F_{k+1,\ldots,2k} &= \frac{1}{4}[M^{\nearrow}, M^{\searrow}, M^{\swarrow}, M^{\nwarrow}] \end{aligned} \tag{3.6}$$

where $k = (2T + 1)^2$.

The calculation of the difference array can be interpreted as high-pass filtering with the kernel $[-1\ 1]$. The filter effect is to suppress image content and to show high frequencies where stego noise is stored. Compared with previous methods, errors made by the detector based on SPAM features are lower.

Although SPAM features are originally developed to detect stego images, they can be used for digital forensic applications too. Many forensic authors compare the results obtained with their methods to results achieved with SPAM features in detection of noise due to JPEG-compression, anti-forensic dither [21] or in median filter detection [6].

The strong point of SPAM features is that they try to relate adjacent pixels using differences of pixel values. That is definitely a clever decision considering that many filters and forgeries actually change the relationship between one pixel and the next one. SPAM features therefore appear as a natural and general tool to solve problems that even sounds closely related to behavior of neighboring pixels as median filter detection. It must be pointed out that, after median filter process, all values of the new image were already belonging to the original image. We think this fact must be exploited in order to create an algorithm capable of identifying the application of the median filter that aims to be robust. Therefore we developed a new feature for the median filter which is described in the next section. A non-secondary drawback of the SPAM approach comes from the required dimensionality of feature space. Even

the minimal version of SPAM features set is composed by 162 dimensions. In such big vector space detectors and classifiers requires a huge training set, suggested to be 10 times bigger than the number of features.

3.1.3 Difference domain approach

A method that we closely analyze about median filter detection is described by Chen et al. [4].

In the article the authors explain how they construct two sets of features based on difference domain with the aim of detecting median filtering. The first set, *Global Probability Feature Set* (GPF), is based on empirical distribution function for $k-th$ order difference. The second set, *Local Correlation Feature Set* (LCF), is based on the correlation between adjacent difference pairs in the difference domain.

The authors assume that, with the inherent nature of the median filter, it is expected that pixels in median filtered images are correlated to their neighbours in a specific way. For this reason they choose to work in difference domain. With the following quantities: $(p, q) \in \{-1, 0, 1\}^2$ with $|p| + |q| \neq 0$ indicating the direction where difference is computed, k defining the order of difference, Δ_0 being the original signal and (m, n) the coordinate of element computed; the authors define the $k-th$ order difference as

$$\Delta_k^{(p,q)}(n, m) = \Delta_{k-1}^{(p,q)}(n, m) - \Delta_{k-1}^{(p,q)}(n + p, m + q) \tag{3.7}$$

Invariance of an image with respect to mirroring and flipping is assumed [18]. Furthermore they define the *empirical distribution function* (CDF) for the $k-th$ order difference as the proportion of value of the matrix $\Delta_k^{(p,q)}$ less than of a chosen threshold t.

$$F_k^{(p,q)}(t) = Pr(|\Delta_k^{(p,q)}(n, m)| \leq t)) \tag{3.8}$$

The curve of CDF varying the threshold shows a different shape for original images and filtered images. In fact, in filtered images part of high frequency components are removed.

In order to build the first set of features, CDF of each $|\Delta_k^{(p,q)}|$ is evaluated for different thresholds and stored in a vector $P_k^{(p,q)}$ of length T+1. This procedure is applied for every k order of difference. Finally, with the aim to reduce dimensionality and under the hypothesis of symmetry, these vectors are averaged and the mean is saved in a shorter vector with $2(T + 1)K$ dimension.

$$\mathbf{P}_k^i = \frac{1}{4} \sum_{|p|+|q|=i} \mathbf{P}_k^{(p,q)} \tag{3.9}$$

Thus, feature vector is formed as

$$\mathbf{P}_k = [\mathbf{P}_k^1, \mathbf{P}_k^2] \tag{3.10}$$

The first set of features, *the global probability feature set* (GPF), is built by concatenating all the $\mathbf{P}_k$ $(k = 1, 2, ..., K)$.

To build the second feature set the authors need to introduce the *joint probability* of $\Delta_k^{(0,1)}$. Joint probability is defined as the probability that in $\Delta_k^{(0,1)}$ matrix two particular values (t_x, t_y) in a particular position (n, m) and $(n, m + l)$ concurrently occur.

$$P_{k,l}(t_x, t_y) = Pr(\Delta_k^{(0,1)}(n, m) = t_x, \Delta_k^{(0,1)}(n, m + l) = t_y) \tag{3.11}$$

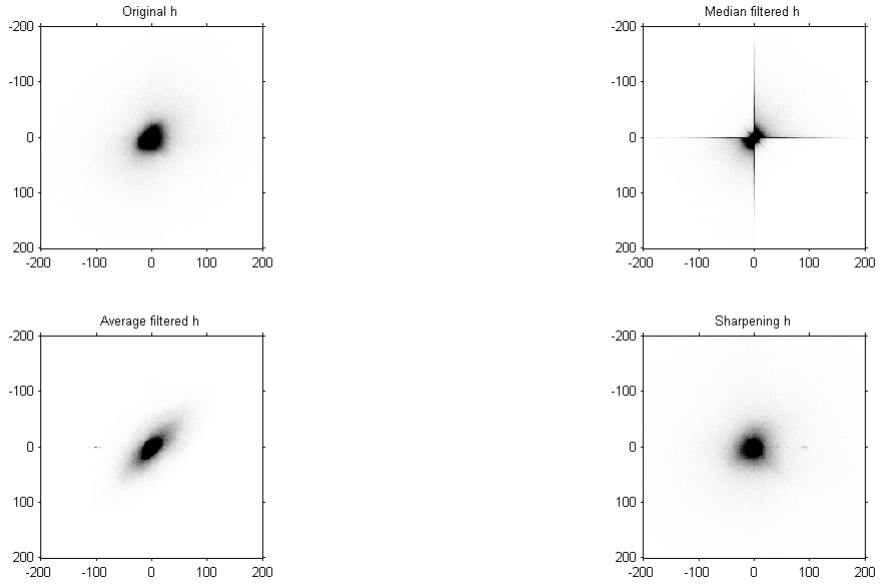


Figure 3.1: Adjacent difference pairs for several processed images. Images subjected to median and average filter present a characteristic shape that can be used to recognize the presence of these manipulations.

In accord with literature [26], median filtered images in difference joint distribution present high probability of 0 value. Moreover, thanks to the median filter good

edge preservation property, adjacent difference pairs have a high probability to share the same sign. Instead, because of the smoothing edge property, the distribution of different adjacent pairs for average filtered images is less scattered than the original images or median filtered images.

Thus a 1xB window is chosen and difference matrix is scanned in (overlapping or not) within this window along the same direction as it is computed. This procedure is used for all K difference matrices where $|p| + |q| = 1$. Afterwards every value in the same position in windows are considered as a sample of distribution of a random variable and therefore to have B random variables.

For each pairs of random variables is computed the normalized cross correlation coefficient (NCC) defined as

$$\gamma(x, y) = \frac{cov(x, y)}{\sqrt{cov(x, x)cov(y, y)}} \tag{3.12}$$

Authors scan (overlapping scan) the 2D difference array $\Delta_k^{(p,q)}$ with a 1D window of width B along the same direction as it is computed and rearrange all B pixels in the column of a matrix $\mathbf{Y}_k^{(p,q)}$. Then, matrices $Y_k^{(p,q)}$ with lags $|p| + |q| = 1$ are combined to obtain

$$\mathbf{Y}_k^1 = [\mathbf{Y}_k^{(0,1)}, \mathbf{Y}_k^{(0,-1)}, \mathbf{Y}_k^{(1,0)}, \mathbf{Y}_k^{(-1,0)}]^T \tag{3.13}$$

All NCC coefficients of the column of $\mathbf{Y}_k^1$ are combined in the vector C_k^1 of $(B^2 - B)/2$ dimensions.

A similar vector is built with lags $|p| + |q| = 2$

Finally, concatenating all C_k $(k = 1, 2, ..., K)$ together leads to a $(B^2 - B)K$ dimensional local correlation feature set (LCF).

The final set is obtained combining all features computed so far. It is a $K[2(T + 1) + (B^2 - B)]$ dimensional vector.

Thus to detect smoothing techniques the idea relies on looking at difference between adjacent pixels. During the smoothing operation high frequencies are typically removed bringing adjacent values close together. As shown in fig (3.1), the computation of median or average filters force neighbouring to pixels share the same sign. Very impressive is the case of median filtering process that shows very high chance for a median to be common for adjacent pixels belonging to the same row. Indeed comparing two different windows for side by side pixels, we would like to emphasize that only one line of the window-matrix is changed. This way most of

the values are common between the two with a consequent high chance to obtain the same median.

This method has the robustness we are seeking, creating a characteristic pattern when equal values appear in the image pixels. However our approach will directly consider also the value of the pixel itself reducing by far the complication of joint distributions. As we will show, in the fifth feature presented in section 3, we can face the problem without looking at the surrounding values of a pixel and will have no need for choosing any privileged directions and distance as in the presented case.

3.2 Proposed features

In this section the five scalar features proposed by this thesis and the reasons for their choice are illustrated. In order to extract the features needed to distinguish the considered modification the guideline adopted in this section is to choose one possible alteration at a time and to look for the main character that marks the modified image. From the analysis of these properties one feature at a time is extracted.

Because the considered alterations have an influence on the entire image, our work is focused on the extraction of global image features, i.e. values evaluated by the elaboration of all pixels in the same way.

Five features are found. The research of first two is inspired by the JPEG-compression algorithm. As described in chapter 2, JPEG-compression reduces information quantity thanks to quantization of discrete cosine transform (DCT) coefficients inducing a decrease of the entropy in their distributions. On the other hand, distribution of DCT coefficients of noisy images has a higher entropy than the original since the uncertainty of the noise courses increasing of entropy.

The third feature is created with the purpose of detecting filters. The idea is that the presence of a filter changes the relation between adjacent pixels [4].

The fourth is based on low-pass and high-pass filters properties. In fact thanks to considerations about filters properties, a useful feature can be extracted studying the Fourier Transform shape.

For the last feature we start from median filter particularly characteristic of eliminating outlier and we apply this particular enhancement to a new image in order to analyzed the difference between images before and after filter application.

The vector defined combining these five features will be the input of classifiers described in chapter 4. Low dimension and generality of features vector are one of the strength of the entire method.

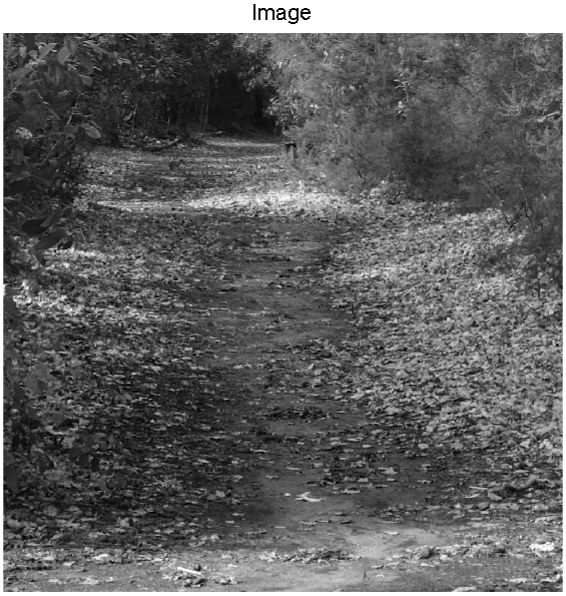
3.2.1 Entropy features

In this section two features are described. They have been created studying the information properties of the DCT coefficients computed as for JPEG-compression algorithm. The analysis gives rise to one feature to recognize JPEG-compression and one to reveal noise addition. In order to explain the procedure, quantities and tools used are illustrated.

Computation of entropy

In the JPEG-compression algorithm the values obtained by DCT are quantizated so as reduce the amount of bytes needed to save the image on a hard drive. We can consider the number of needed bytes per image as the measurement of the amount of information stored into the picture. This quantity reflects the number of typical sequences that the digital signal can assume which are connected in information theory to a quantity called entropy. We report a brief introduction about entropy, typical sequences and information in the appendix A. Here we will just mention that the number of possible typical sequences, the value of the entropy and information for an image are all reduced to the quantization of DCT's coefficients matrix. We refer the reader to chapter 2 where the JPEG-compression algorithm is introduced for an explanation of how the quantization is actually computed.

Since the reason for the loss of information is the quantization done on DCT the effect is made evident when the same transformation is performed. This is the obvious way to show this regularity in the coefficients that would remain otherwise hidden.



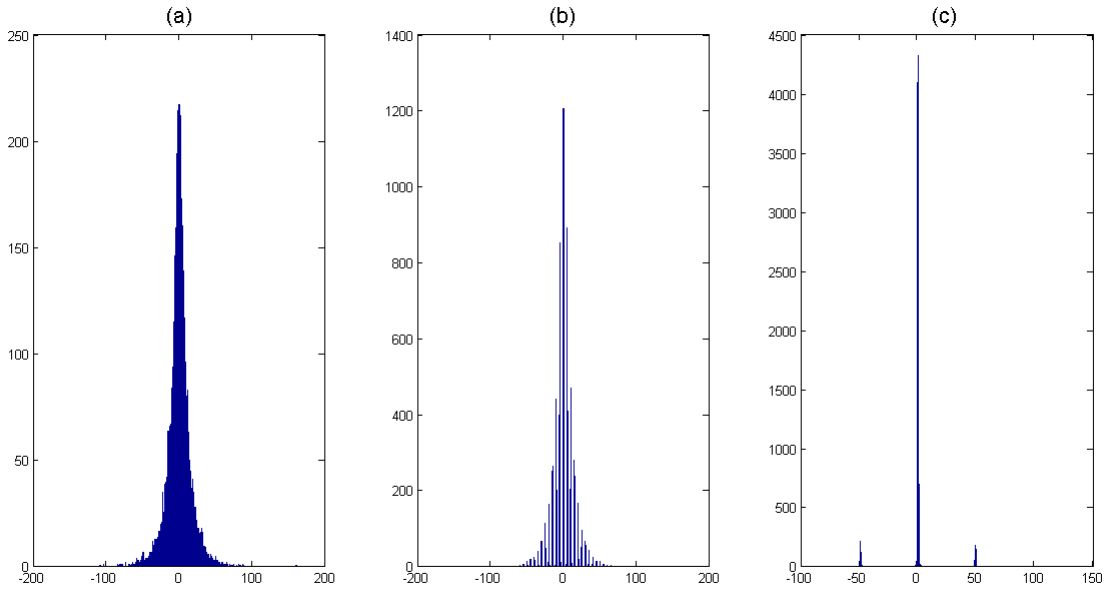


Figure 3.2: Distribution of DCT coefficient in position (1,7) of (a) Original image, (b) JPEG-compressed image QF 95, (c) JPEG-compressed image QF 50

Figure (3.3) shows how the mean entropy of the DCT coefficients varies as a function of the dimension of the square window used for the DCT evaluation has been created. Each point of the graph is relative to a window dimension, that has been used to subdivide the considered image into non overlapping blocks. Over each block the DCT is calculated and an histogram has been created for each coefficients. The histograms are the used to evaluate the coefficient entropies. As a last step the entropies are averaged. It is important to take into account that JPEG uses a windows of dimension 8 pixel for the DCT evaluation.

In order to make comparable average entropies computed on different windows' size, they are normalized between $[0, 1]$ dividing them for their maximum possible values. The configuration where entropy is maximum is that where all histogram bins have the same probability to occur, that is where distributions of DCT coefficients are uniform.

Let X be the random variable that describes the behaviour of one pixel, let N be the number of bins chosen, let x_i with $i \in \mathbb{N} \cap [0, N-1]$ be the possible values that X can take and let $p_i = P(X = x_i) = \frac{1}{N}$ be the probability that X takes the $i-th$ value: to maximize entropy X must have uniform distribution $p_i = p_j$ $\forall i, j \in \mathbb{N} \cap [0, N-1]$. Entropy is evaluated to be

$$H_{max} = -\sum_{i=0}^{N-1} p_i log p_i = -\sum_{i=0}^{N-1} \frac{1}{N} log \frac{1}{N} = -\frac{N}{N} log \frac{1}{N} = log N \tag{3.14}$$

We indicate as $p_i(L)$ the probability that i-th bin has to occur when the image is divided into LxL windows.

$$H_{avg}^{IM}(L) = -\frac{1}{H_{max}} \frac{1}{L^2} \sum_{k=1}^{L} \sum_{j=1}^{L} \sum_{i=0}^{N(L)-1} p_i(L) log p_i(L) \tag{3.15}$$

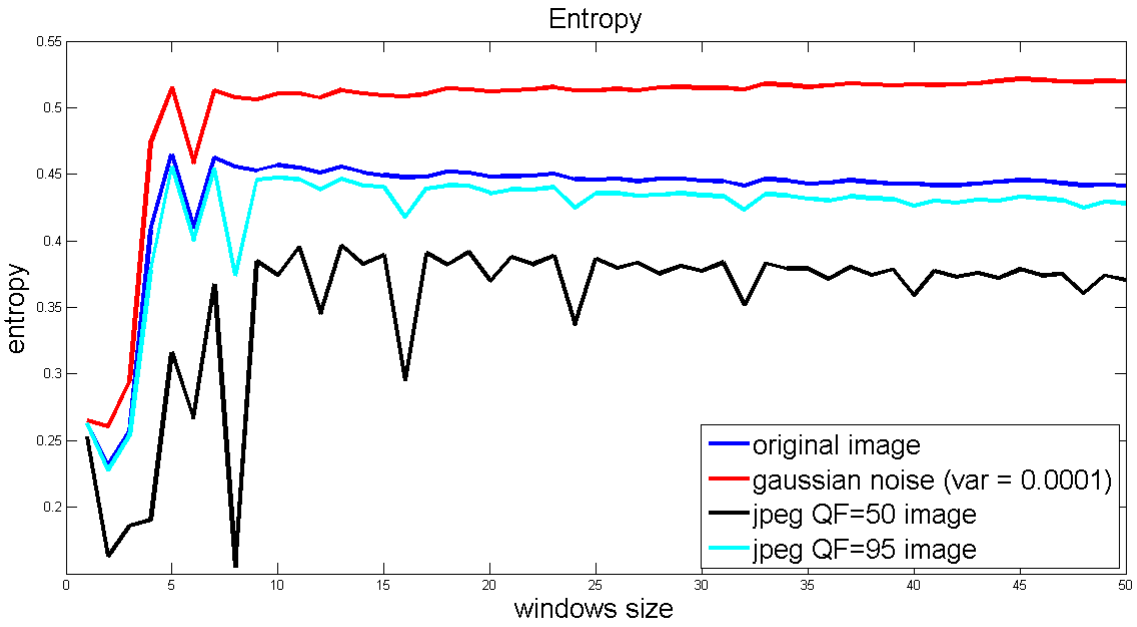


Figure 3.3: Normalized average entropy vs DCT's windows size for a generic image. In JPEG-compressed images a strong decrease of average entropy for 8x8 windows size and multiples is visible. This phenomenon is evident also for compressions with high quality factor.

From figure (3.3) it is possible to gather that JPEG-compression influences entropy of distribution of DCT's coefficients but just if it is computed on an 8x8 pixels window (or multiples), otherwise the average entropy of DCT coefficients has a shape quite smooth and regular. For this reason we found sufficient to study the variation of average entropy for DCT windows for size of 7-8-9 pixels. In particular a good scalar feature could be the difference between the mean value of average entropy for windows with size 7 and 9 pixels and the average entropy for a 8 pixels side windows:

$$F = \left(\frac{(\overline{H}_{avg} - H_{avg}(8))}{\overline{H}_{avg}} \right)^2 \tag{3.16}$$

where

$$\overline{H}_{avg} = \frac{H_{avg}(7) + H_{avg}(9)}{2} \tag{3.17}$$

Angular average entropy

Using the previously defined feature all DCT coefficients are considered with the same weight, but we know that most part of natural image information is normally stored in low frequencies, whereas high frequencies contain knowledge about details and edges present in the scene. Moreover, the quantization table of JPEG-compression method has bigger factor of quantization in correspondence with high frequencies because the loss of information is less visible for the human eye. For this reason entropy at low frequencies are less subjected to quantization than for high frequencies after JPEG-compression has been applied.

We choose to consider images frequencies isotropic, disregarding waves direction and focusing only in period of waves. In practice we assume that all frequencies that have the same Euclidean distance from the left high corner of the matrix have the same properties. As an example we see that amplitude of the frequency (1,2) will be equivalent to the one of frequency (2,1)

Let $f(x, y)$ be a two variables (x, y) function. The *angular average* is defined as the mean of the amplitude values whose frequency ranges between two specific distances from the origin. In this case the central point is set to be the central point in the pixel placed in the left high corner of the DCT coefficients matrix.

The aim of the computation of angular average is to average properties for frequencies equidistant to (0,0) point. Let l be the width of subband that group frequencies, then the $z - th$ subband in continuous case can be described as

$$\mathcal{S}_z = \{(x, y) | (z-1)l \leq \sqrt{x^2 + y^2} < zl, z \in \mathbb{N}^+, (x, y) \in \mathbb{R}^2 \cap \{x \geq 0, y \geq 0\}\} \tag{3.18}$$

Let be $\mathcal{A}rea_z$ the area of z-th subband in a continuous domain

$$\mathcal{A}rea_z = \int_0^{\frac{\pi}{2}} \int_{(z-1)l}^{zl} \rho d\rho d\theta = \frac{\pi}{2} \frac{l^2(2z-1)}{2} \tag{3.19}$$

Continuous angular average of a function $f(x, y)$ in the z-th subband is the result of a polar integral

$$\mathcal{A}_z = \frac{1}{Area} \int_0^{\frac{\pi}{2}} \int_{(z-1)l}^{zl} \rho f(\rho cos\theta, \rho sin\theta) d\theta d\rho \tag{3.20}$$

In the simpler discrete case the function is a matrix where each element identifies a value of this function and row and column determine the (i, j) position.

Let $c_{i,j}$ be the element in $i-th$ row and $j-th$ column of matrix, l the width of subband, than the $z-th$ subband can be define as

$$S_z = \{c_{i,j} | (z-1)l \leq \sqrt{i^2 + j^2} < zl, z \in \mathbb{N}^+\} \tag{3.21}$$

In order to increase precision each pixel is divided into more subpixels and each pixel is assigned to a particular subband weighed by the proportion of subpixels that lies in this subband. Let k_{sp} be the number of subpixels in each pixel and (k, h) the subpixels indexes, z-th subband is defined as

$$S'_z = \{c_{k,h} | (z-1)lk_{sp} \leq \sqrt{k^2 + h^2} < zlk_{sp}, z \in \mathbb{N}^+\} \tag{3.22}$$

and consequentially the angular average is

$$A_z = \frac{1}{\#S'_z} \sum_{c \in S'_z} f(k, h) \tag{3.23}$$

Where with the symbol # we want to indicate the cardinality of a set.

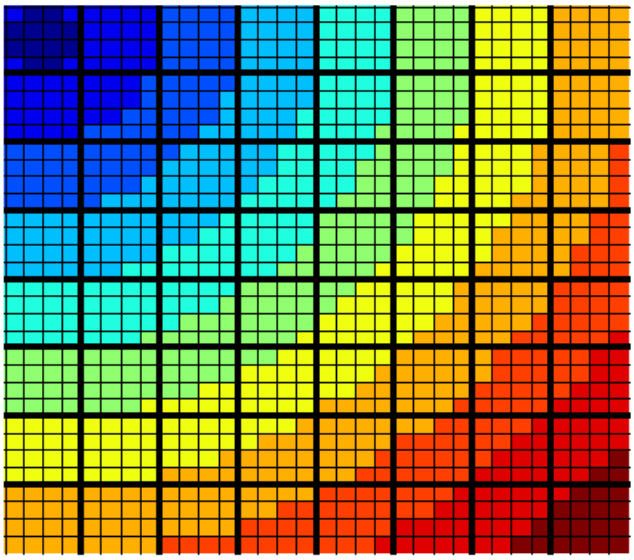


Figure 3.4: Example of subbands for a angular average computation with 4x4 subpixel subdivision.

In each of these subbands the mathematical average of elements A_z is evaluated and inserted in a vector in $z-th$ position. Angular average can be considered as a reduction of dimensionality. In fact, a vector is extrapolated from a 2D representation of data.

In our case the function studied is the matrix where entropies are stored. Position of elements in this matrix is related to frequency domain, so each value of angular average represents the average for a determinate frequencies subband, independently by direction.

We define *angular average entropy* the vector $H_{ang}(z)$

$$H_{ang}(z) = \frac{1}{\#E_z} \sum_{c \in E_z} H(c) \tag{3.24}$$

Width of subbands is set to 1 pixel and the number of subpixels used is 8. This choice entails that varying windows size the length of output vector changes too.

In the graph (3.5), each subband is identify by the spacial frequency that corresponds to the DCT coefficient along the diagonal direction of the DCT matrix at the center of the considered band. Therefore the considered spacial frequencies ranges from 0 to $\frac{1}{\sqrt{2}} pixel^{-1}$. Because of phenomena described in Nyquist's theorem the minimum frequency observed is depending on the number of samples that are present in the window. For this reason the larger is the window, the smaller is the

distance between observable frequencies.

In order to compare average entropy of DCT coefficients evaluated on different windows size a resampling of the average entropy curves is necessary: we choose to use a spline interpolation in order to be able to sample functions at the same point.

Considerations done until now allow to think that the most of change of entropy's value appears between curves built by windows with side equal to seven, eight and nine.

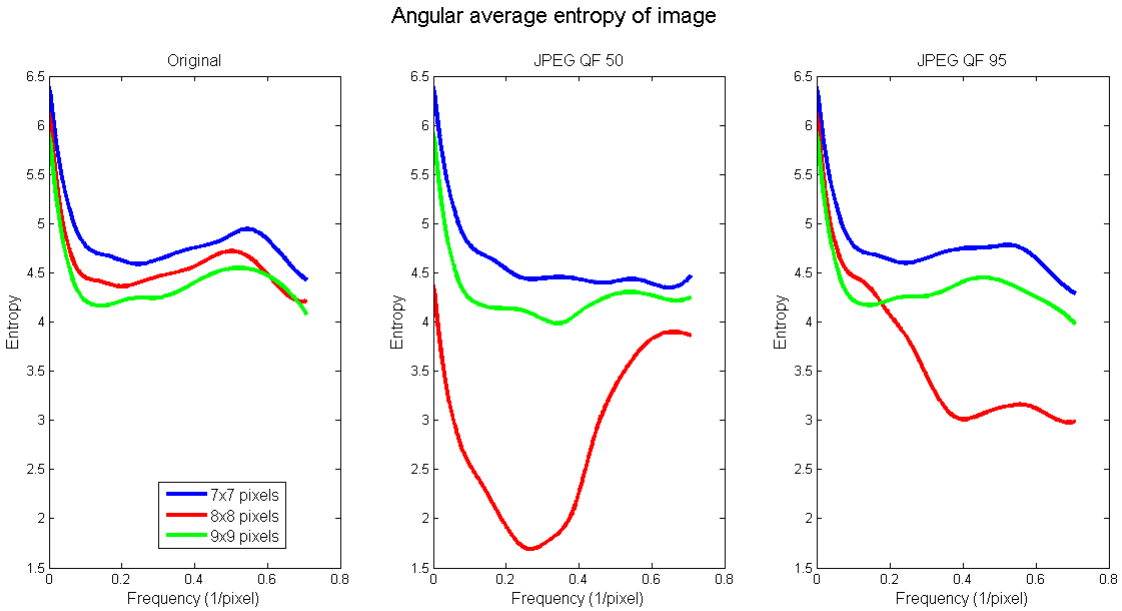


Figure 3.5: Spline interpolation of angular average entropy for 7x7, 8x8 and 9x9 windows for different JPEG-compression. The curve that represents the window with size 8x8 is manifestly lower than the others.

As we expect in a JPEG-compressed image entropy for an 8x8 window is lower than entropy for 9x9 or 7x7 that have almost the same behaviour. The range of frequencies most influenced by quantization depends on quality factor.

Number of bins

The calculation of the entropy of DCT coefficients relies on the knowledge of the probability distributions. Each image has its own probability distribution for the coefficients, but obviously images, that we are interested in, should have some common properties or shape that this distribution typically looks like. In literature the model of this distribution is a discussed issue as still an open question[27]. Whichever this typical distribution should be, we are only interested in determining in which con-

dition we can approximate the distribution by a histogram. Our histograms will be computed using the DCT values coming from each window as the DCT is done. Two big factors come into play in determining the good condition the limits in which our histograms should be computed.

The first is obviously the limitation of the number of windows, that automatically fixes a bound on the number of samples on which histograms are computed. Trying to approximate the probability distribution with a too large number of bins can effectively reduce the precision. In this case the average number of samples per bin could be insufficient and thus inducing enormous fluctuations of their values while destroying the possibility to get to a good approximation of the distribution.

On the other hand the possible values that the DCT coefficients can achieve are not infinite, since we start with 8 bit quantized values for the image in real space. This means that even in the eventuality of having an infinite number of windows where to take DCT coefficients values, a minimum interval step for bins can be defined as the problem is intrinsically discrete and finite in space domain.

We empirically tested different number of bins and we found that the most suitable values can be reached with 1024 bins.

JPEG feature

The first feature is built by improving equation (3.16) with the principle of angular average entropy. The idea is to compare the entropy of an image with the entropy of the same image as if as it was never compressed. Obviously the state of images is not known: the original image is often not available.

The reference measure for the entropy of a natural image can be the angular average of 9x9 or 7x7 DCT coefficients matrix. In order to make this value more stable we decide to consider the mathematical average between the angular average entropy curve of 9x9 and 7x7 DCT coefficients matrix.

The integral of the reference curve and the integral of the angular average entropy curve of 8x8 DCT coefficients matrix are compared and the normalized difference between these two values is considered as the first feature.

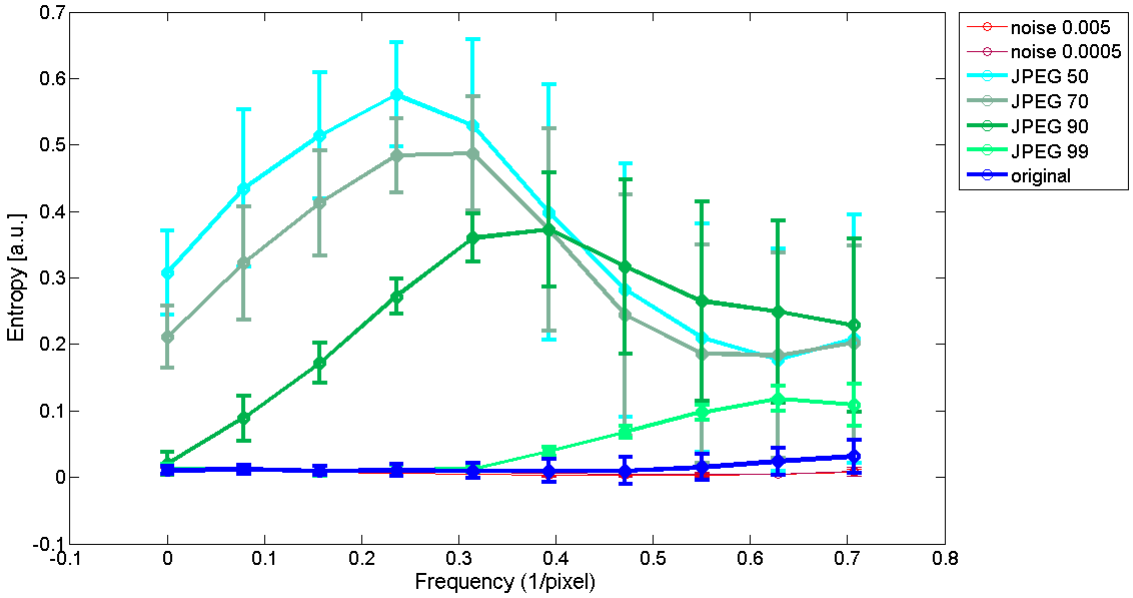


Figure 3.6: Mean value (and standard deviation) of the first feature computed on several non overlapped subbands of frequencies. The mean and standard deviation are computed on 100 images. The feature results effective for each considered quality factor but in different subband. For this reason we find that consider all frequencies can be a solution whenever quality factor of possible compression is unknown.

In order to optimize the solution, the integral extremes are varied. F_1 is computed with 10 different non-overlapped domains of integration for 100 images. The number 10 is chosen because the number of subbands where the angular average entropy is evaluated in a 7x7 window. Mean values and standard deviations are plotted in figure 3.6.

Because of the strong dependence between the quality factor and the influenced frequencies we choose to consider the whole range of frequencies.

Let $c_m(x)$ be the mathematical mean of curve $c_7(x)$ and $c_9(x)$

$$c_m(x) = \frac{c_7(x) + c_9(x)}{2} \tag{3.25}$$

then, the first feature is calculated as

$$F_1 = \int_0^{\frac{1}{\sqrt{2}}} \left(\frac{c_m(x) - c_8(x)}{c_m(x)} \right)^2 dx \tag{3.26}$$

Noise feature

Entropy can also be used to detect the presence of additional random noise. In order to understand the idea let us remember that random noise in spatial domain remains random noise once transformed in DCT domain. When random quantities are added to an image, information on high frequencies of image increases and consequentially entropy increases, too.

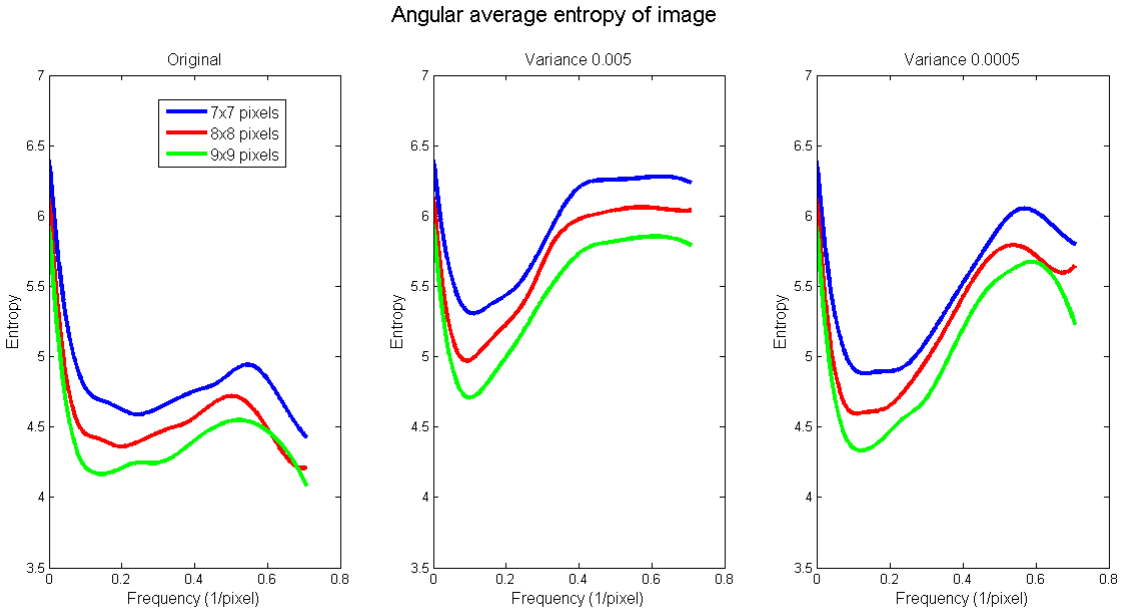


Figure 3.7: Spline interpolation of angular average entropy for 7x7, 8x8 and 9x9 windows for addition of Gaussian random noise with different variance. All curves have the same behaviour for each kind of manipulations but if random noise is added, high frequencies present higher entropy.

The second feature is built specifically to detect noisy images. It is based on the integral of the curve obtained interpolating frequencies for a 9x9 and 7x7 DCT coefficients window. The choice of these particular windows is not mandatory: in fact in not JPEG-compressed images entropy is quite constant varying the length of the side of the windows. On the other hand it is important not to use 8x8 window whereas the image should also be JPEG-compressed and in this case the entropy minimization due to the quantization operation hides the raise of the same value that is proper of a noisy image.

The mathematical average between these two curves is computed in order to make the feature more stable to avoid any possible error. The feature is evaluated as 2-norm of average curve calculated only for frequencies higher than $\frac{1}{2}\frac{1}{pixels}$.

Let $c_7(x)$ be the curve that describes angular average entropy for 7x7 windows and $c_9(x)$ be the same curve for 9x9 windows. The second feature F_2 is defined as

$$F_2 = \int_{0.5}^{\frac{1}{\sqrt{2}}} \left(\frac{c_7(x) + c_9(x)}{2} \right)^2 dx \tag{3.27}$$

The benefits of this feature are evident thanks to a deviation between high frequency values of noisy images and the original.

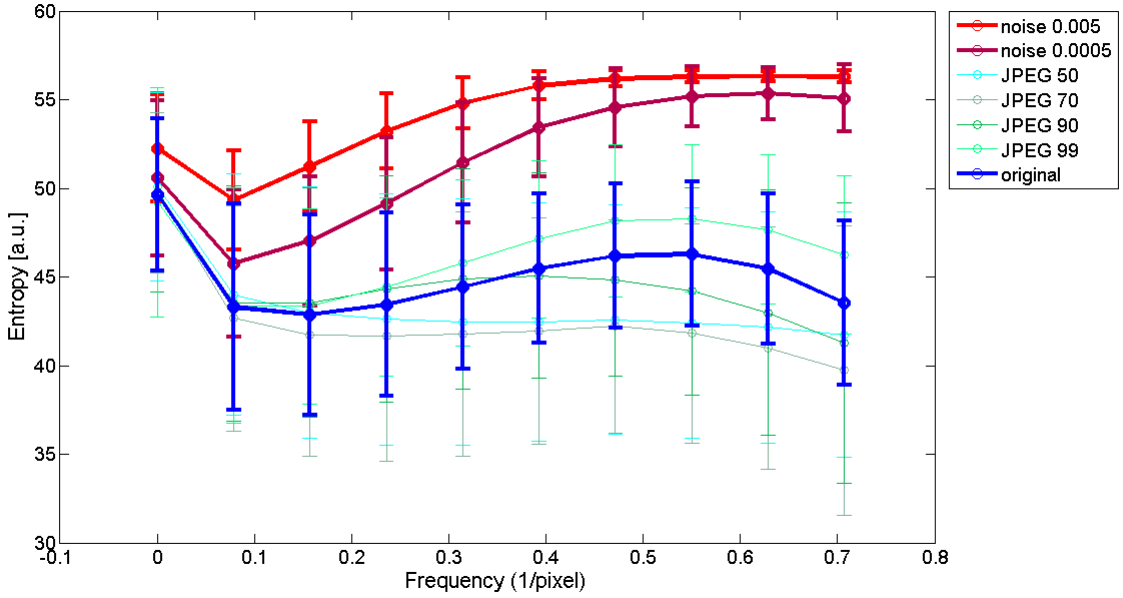


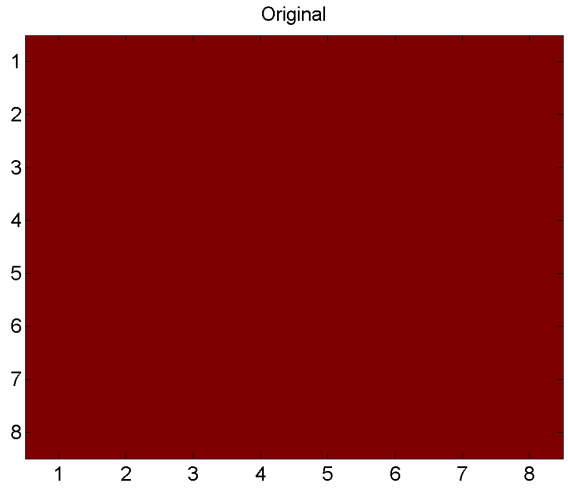
Figure 3.8: Mean value (and standard deviation) of second feature computed on several non overlapped subbands of frequencies. The mean and standard deviation are computed on 100 images. As predictable, the feature results effective for the higher frequencies.

Anti-forensic attack: cropped image

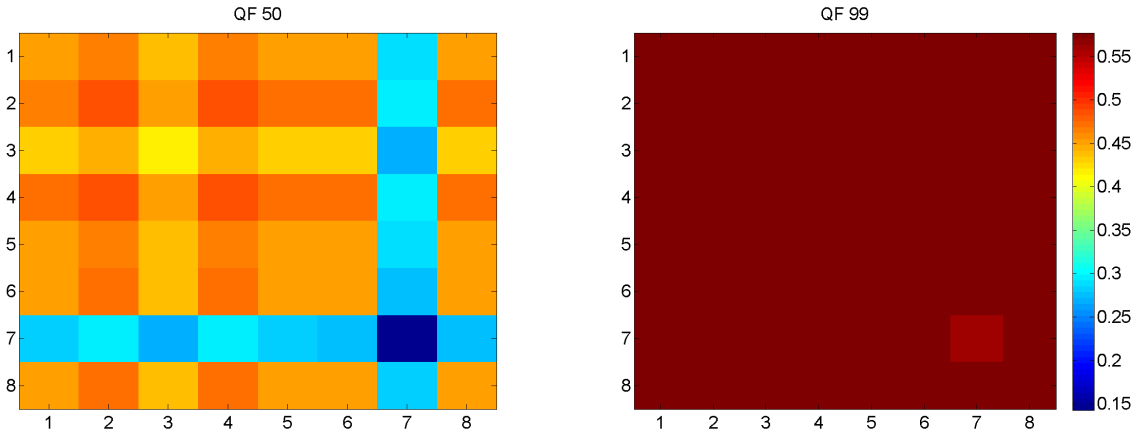
These features, especially the first, can be easily attacked by anti-forensic methods. In fact, as in [5], they are based on the identification of regularity in the structure of JPEG compression. If an image is cropped and the size is reduced the position of windows where DCT was performed is lost, thus defeating the detection of JPEG compression.

Nevertheless we know that, if the image was compressed, a configuration exists where the average entropy evaluated in DCT coefficients for an 8x8 window is lower

than the other average entropy. In the light of that, a possible solution is to compute 64 times average entropy described in equation (3.15) with 8x8 DCT coefficients windows, each time cropping the image in a different way, in order to prove all possible configurations. Average entropy will be lower where it is computed on the same windows where DCT is performed. That allows us to place the grid in the right position and then to proceed with the algorithm described.



(a) *Original*



(b) *QF 50*

(c) *QF 99*

Figure 3.9: An image has been compressed with different QF and the first two rows and columns are cropped. Mean entropy for each starting point is computed. Let us note that also in JPEG-compression with quality factor of 99 the starting point of the grid where was computed the DCT is evident

Although the difference is really low in the case of JPEG-compressed image with quality factor of 99, the pixel corresponding to DC coefficient in JPEG formulation is

still visible (fig 3.9). On the contrary, in the original image the result of the average entropy computed on each possible configuration are really close each other.

3.2.2 Spatial feature

In their work Chen et al. [4] use $K[2(T+1)+(B^2-B)]$ features, where K is the order of difference considered, T is the number of possible thresholds for the cumulative distribution function tested and B is the window size in computation of Local Correlation Features. In the setting of their experiment there are in fact 56 features. All these features are built with the purpose of detecting median and average filtered images. We would like to find a single feature that can describe the dissimilarity between original images and filtered images as best as possible.

In the application of the filter the correlation between adjacent pixels in filtered images is expected to be changed, because of the use of overlapping windows. This is the starting point of many works present in literature [4], [26], [18].

We choose to base the third feature on the characteristics that an image can have in a difference domain. With this aim the joint distribution of adjacent difference pairs is studied. Let $\Delta_h(n,m)$ be the difference between two pixels placed side by side and $\Delta_v(n,m)$ the difference between two pixels placed one under the other

$$\Delta_h(n,m) = X(n,m) - X(n,m+1) \tag{3.28}$$

$$\Delta_v(n,m) = X(n,m) - X(n+1,m) \tag{3.29}$$

As opposed to Chen et al. [4] only the first order difference is considered and the step l is considered equal to 1.

Distribution of different adjacent difference pairs is empirically created

$$P_v(t_x,t_y) = \mathbf{P}(\Delta_v(n,m)=t_x, \Delta_v(n+1,m)=t_y) \tag{3.30}$$

$$P_h(t_x,t_y) = \mathbf{P}(\Delta_h(n,m)=t_x, \Delta_h(n+1,m)=t_y) \tag{3.31}$$

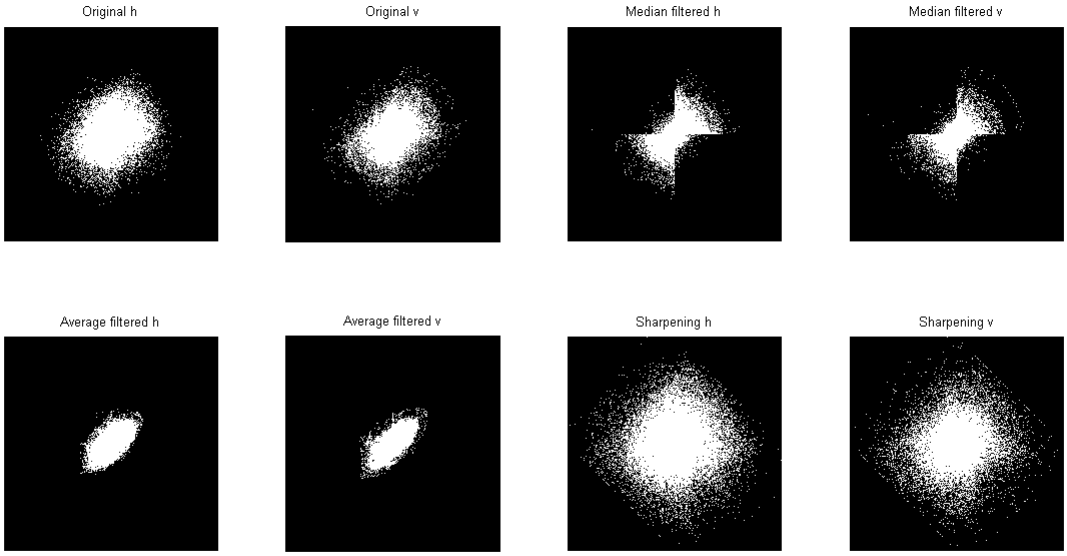


Figure 3.10: Representation or joint distribution of difference P_h and P_v for image subject to different processes. Images subjected to median and average filter have a really distinctive shape.

The original images have a rather a symmetric distribution; the spread of samples is more or less uniform for every direction. On the contrary, median and average filtered images have a distribution condensed in the first and third quadrant.

This particular shape is due to properties of smooth filters. In fact, removing noise and outliers, the probability that adjacent difference share opposite sign is lower. Moreover the median filter maintains the integrity of the edges and for this reason the first and third quadrant are almost unchanged.

On the other hand the capacity of sharpening filters of *sharpening* edges allows the distribution to be more spread out in all directions.

Feature

The third feature described in this work studies the shape of different adjacent difference pairs distribution. Unlike other authors [26] we focus on the ratio of adjacent difference that shares the same sign and we do not require them to be equal.

Because the hypothesis of isotropy differences are computed both in horizontal and in vertical directions. After that, for each direction the probability to be in the odd quadrant is evaluated. These two quantities are mathematically averaged in order to enhance the stability of the feature.

$$F_3 = \frac{P_v(t_x \geq 0, t_y \geq 0) + P_h(t_x \geq 0, t_y \geq 0)}{2} \tag{3.32}$$

3.2.3 Fourier space feature

The fourth feature is extracted from Fourier domain and is focused on detection of average and sharpening filtered images. The property of low and high-pass filers of influencing different part of amplitude of Fourier transformation can reveal useful information to recognize images subjected to average and sharpening filters.

Angular average

Unlike what is done for the entropy features the image is not divided in windows but the Fourier Transform is computed on the entire image to fit the variation of frequencies coefficients better.

The *fft shift* MATLAB function is used in order to arrange coefficients obtained so that the continuous component is in the center of the Fourier transformed image, precisely in position $[\frac{M}{2}+1, \frac{N}{2}+1]$ of coefficients matrix where (N, M) is the size of image.

The assumption of isotropy is still valid because of symmetry of high-pass and low-pass filters, so the angular average can be evaluated. As opposed to DCT angular average, this average is computed for an angle of 2π.

Basically the procedure for evaluating angular average of discrete Fourier transform is the same as that for angular average entropy.

$$Area = \int_0^{2\pi} \int_{(z-1)l}^{zl} \rho d\rho d\theta = 2\pi \frac{l^2(2z-1)}{2} \tag{3.33}$$

$$A_z = \frac{1}{\pi l^2(2z-1)} \int_0^{2\pi} \int_{(z-1)l}^{zl} \rho f(\rho cos\theta, \rho sin\theta) d\theta d\rho \tag{3.34}$$

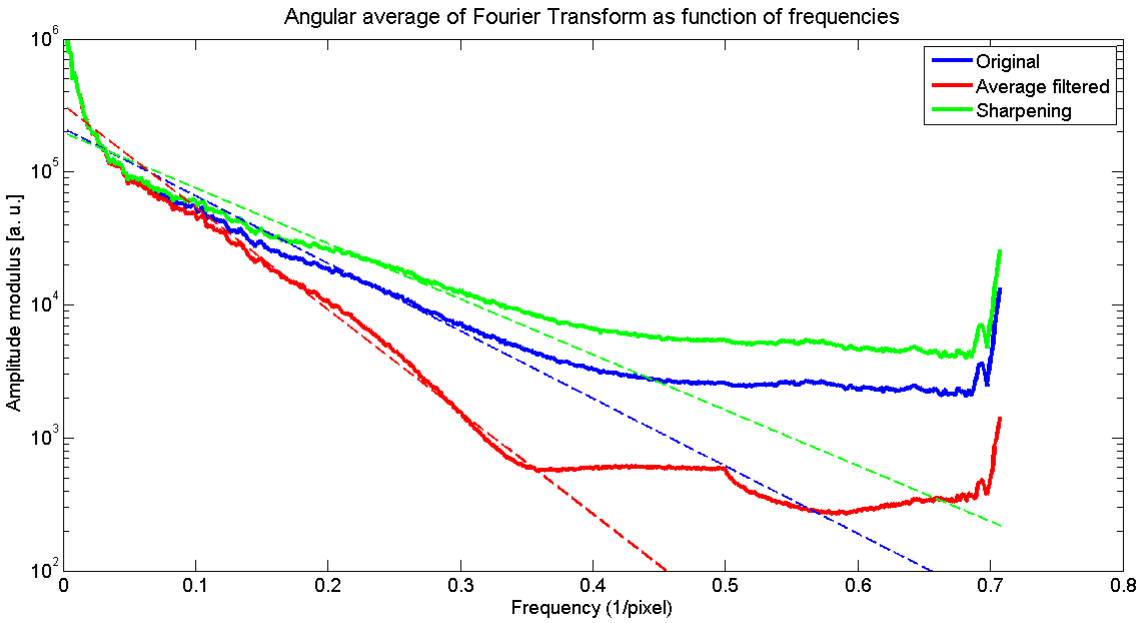


Figure 3.11: Angular average or Fourier Transform of a image. Average filter is computed on [3x3] mask and sharpening filter has a variance of 1.5. The characteristic coefficient of the exponential fit of these curves is typically higher in images subjected to average filtering.

Because of sharpening filters high-pass properties, angular average for images subject to this kind of manipulation decreases more slowly than others. On the contrary, average filters do not influence low frequencies but muffle frequencies higher than its cutoff frequency. For this reason the shape of the angular average curve of Fourier Transform can be useful in order to detect the corresponding manipulations.

The low quantity of information in high frequencies and artifacts due to computation (particularly visible on average filtered image) bias us to focus only on the first half of the frequencies. Angular average amplitude of Fourier Transform is brought in logarithmic scale to give more importance to the general exponential trend at all order of magnitudes without privileging the information at low frequency . Then the curve obtained from logarithmic conversion between 0 and $\frac{1}{2\sqrt{2}}\frac{1}{pixel}$ is interpolated with a polynomial of degree 1. The angular coefficient of this polynomial describes the slope of the curve, equivalent to an exponential decay in the amplitude as a function of the frequency.

Let $ax + b$ be the polynomial that best fits the logarithm of angular average Fourier Transform in the first half frequencies. Than the fourth feature is

$$F_4 = a \tag{3.35}$$

3.2.4 Median filtering feature

The median filter is one of the enhancement most examined in forensic analysis literature [26], [4]. The median filter can be more effective in removing noise in an image than an average filter thanks to the property of edges preserving. For this reason an instrument capable of detecting this kind of manipulations is really useful. On the other hand, the median filter is not computed as convolution with a linear mask as other smoothing filters and this makes it more difficult to identify it. For this reason we studied this particular process with more care.

Recursive application of median filtering

Although the goal of this work is to find features the more general as possible, the fifth feature is based only on median filter properties. We will found that it can detect also average filter thanks to its smoothing property and the presence of random noise.

The median filter has the aim of eliminating outliers and noise. From this consideration it is possible to suppose that a first application of median filter to an image generates a notable change in pixels image and further applications cause fewer and fewer changes. So, if a median filter is applied on an unknown image, counting the number of pixels that change due to a median filtering can be a strong indication in order to point out if the filter was already applied in the past.

Another important characteristic that is shown by this analysis is the presence of noise. In fact, the effect of median filter application is more evident on images where random noise is present. By decreasing discontinuity and eliminating outliers, median filter changes more pixels than in other cases.

Feature

The feature elaborated after the previous consideration is quite simple. A median filter with 3x3 windows is applied to an input image. After that, the number of pixels changed before and after filter application is counted. The fifth feature is the ratio of modified pixels compared to all the pixels of the image. This mimics the calculation of a probability for a pixel to change its value due to median filter application.

Let $p_{i,j}$ be the pixel of an input image in position i,j and $p^f_{i,j}$ the corresponding pixel of image where the median filter is applied

$$F_5 = \frac{\#(p_{i,j} - p^f_{i,j} \neq 0)}{\#p_{i,j}} \tag{3.36}$$

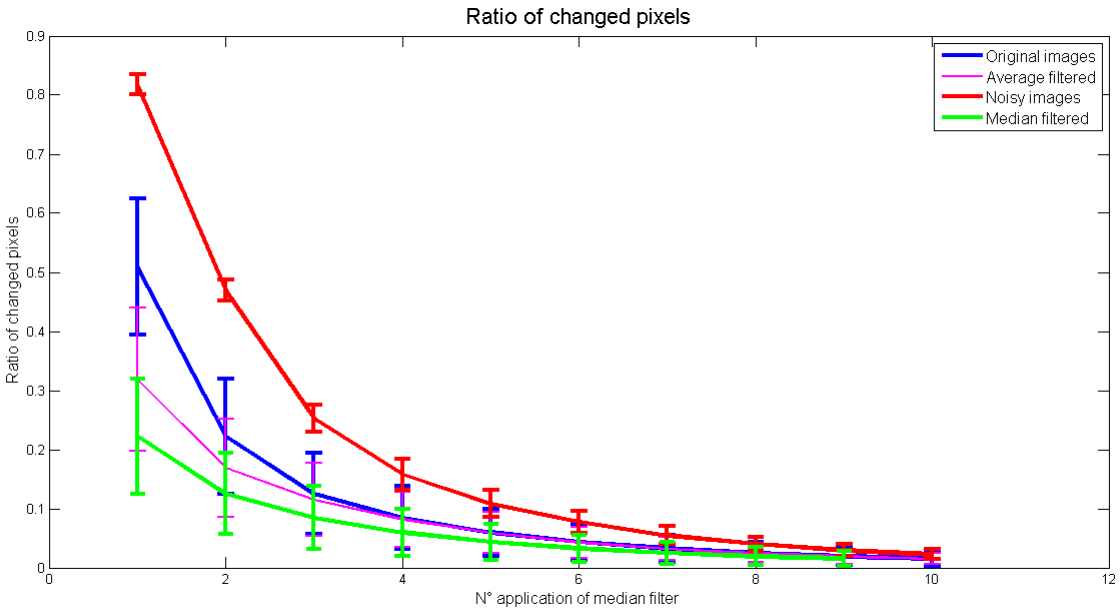


Figure 3.12: Ratio of changed pixles between x-1 and x application of [3x3] median filter. Ratio of changed pixels after one application is clearly higher than the same quantity after two applications. Moreover the amount of changed pixels in a noisy image is really higher respect to that for on original image.

3.2.5 Scatter plot

With the aim of verifying that the five features computed in this chapter are not depended from each others, the correlation is briefly studied with scatter plots. Fifty original images are extracted randomly and represented as point in bidimensional space created by i-th feature versus j-th feature. The closer the shape of points to a cloud, the lower is the correlation between the two features. Correlation between variables indicates that one of the two is in principle not useful in characterizing the problems as one could be expressed as a function of the other.

Figure 3.13 shows that a lightly linear correlation between features is present only between feature 3 and feature 5 with a correlation coefficient around 0.71. This behaviour is comprehensible because of the common purpose of median filter detection. We decide to not take any measures about this correlation because of the low dimension of the feature vector and the difficulty in recognizing median filter presence.

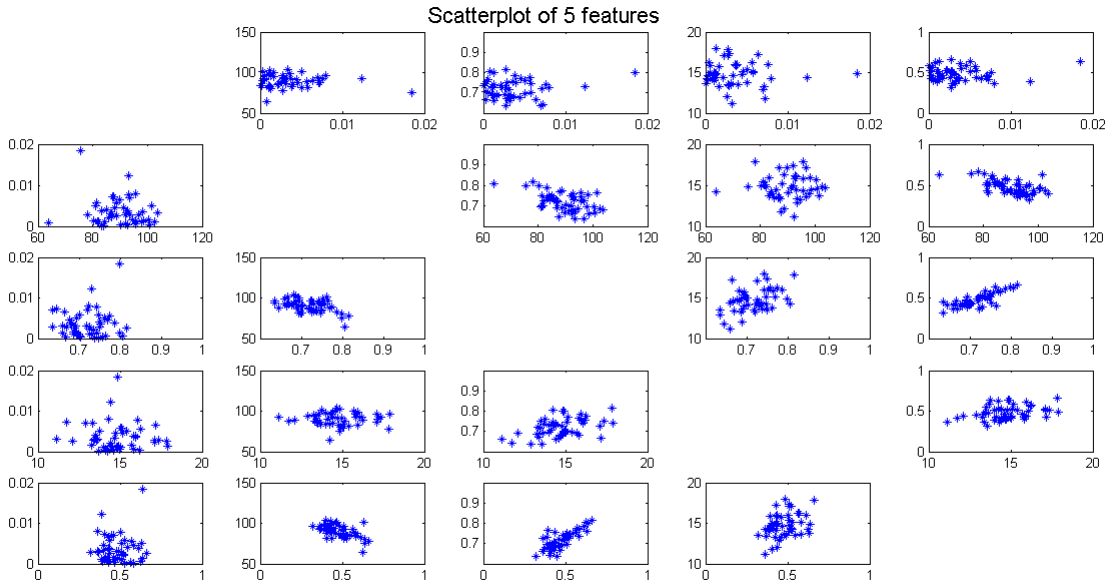


Figure 3.13: Scatter plots of 50 random original images in the space of features described in chapter 3. Feature 3 and feature 5 appear slightly linear correlated (correlation coefficient 0.71). The behaviour is comprehensible because of the common purpose of detecting the presence of median filter.

3.2.6 Summary

In this chapter we presented the features extracted for recognition of the five image modifications we aim to identify. Those features are five as well as the picture manipulations. However only for four there is a one to one correspondence between the feature and the modification that it should recognize. The first feature looks for a particular property own of JPEG-compressed images; the second one is built in order to find images with high Gaussian noise level. The third one tries to recognize filtered images and is based on study of adjacent pixels. Also the fourth wants to distinguish original images from the enhanced ones but is built in Fourier domain. The last feature is created apposite for median filter detection.

These five features are grouped in a vector that will be the input of the classifier.

- JPEG Feature

$$F_1 = \int_0^{\frac{1}{\sqrt{2}}} \left(\frac{c_m(x) - c_8(x)}{c_m(x)} \right)^2 dx \tag{3.37}$$

- Noisy Feature

$$F_2 = \int_{0.5}^{\frac{1}{\sqrt{2}}} \left(\frac{c_7(x) + c_9(x)}{2} \right)^2 dx \tag{3.38}$$

- Difference Feature

$$F_3 = \frac{P_v(t_x \geq 0, t_y \geq 0) + P_h(t_x \geq 0, t_y \geq 0)}{2} \tag{3.39}$$

- Fourier Feature

$$F_4 = a \tag{3.40}$$

- Median Feature

$$F_5 = \frac{\#(p_{i,j} - p^f_{i,j} \neq 0)}{\#p_{i,j}} \tag{3.41}$$

Chapter 4

Classifier

4.1 Introduction

One of the most important problems in machine learning application is the subdivision of data into categories. Categories are made in order to subdivide or recognize common properties belonging to a set of similar objects.

Classification methods are techniques that allow to assign a class to each sample. Usually the starting point of classification is the definition of a *decision function*, a function that associates each element of a dataset to an arbitrary object that represents one class. This object is called *label* and usually it is an integer number.

The purpose of this work is to develop a method capable of discerning pristine images, i.e. never compressed and never subjected to post-process operations, from all others. Given a generic image to decision function we would like to receive as answer whether it is original or any kind of alteration has been applied on the image. This is a two-category problem where the image is recognized to be pristine or not.

Based on the problem's formulation and information included in the dataset, two different approaches can be followed. The first approach is called *clustering* and used when categories are unknown. In order to find an unknown organization of dataset, a distance function is chosen. Data are divided into groups so that the distances within groups is minimized and that between elements belonging to different groups is maximized. In this classification scheme no characteristic or property is required to be known in advance about categories. This is the reason to describe those structures as *unsupervised learning* techniques.

The second approach is used when categories for data are known. The structure of these problems requires well defined categories of interest to be recognized inside the data. Solving this kind of problems requires *supervised learning* techniques, because the properties of the subsets to be recognized are known in advance and must be introduced a priori in order to develop the classifier. In most of the supervised

learning cases a set of objects with its categories association is known and the aim turns to creating a detector for those categories using those objects as samples. The classifier should be able to assign the correct category not only for the available set, but also to any new similar element whose category, maybe, is not yet known.

In supervised learning techniques a definition about the objects to categorize is required. Among these objects it is typically possible to distinguish one object as different from the other with respect to some properties, the so-called features. These properties are then the important things to be defined: a set of quantities or measurements calculated or acquired from the objects on which the differences are investigated.

Our case is an example of supervised classification problem. We start from a bi-dimensional matrix of 8-bit integer numbers that we call image. To look for categories we evaluate the features. We introduced in chapter 3 some features in the hope that they will be able to compose a proper vector space where the division in subsets is made evident. The classifier is then the name of the structure that identifies the correct category for an image as its features are given.

The classifier development will be done in two steps. The first step is to choose the structure of the decision function itself and to optimize its parameters with the purpose of categorization. The second step is a test on the decision function to confirm the optimization process.

To perform both steps the dataset is divided in two. Part of the dataset is used for optimization and is called *training set*, while the second part is called *testing set* and used for the test.

The first part of the dataset, called training set, is used to describe the structure as function of parameters. Because we opt for supervised classification method, labels and classes of training set are known.

Once the decision function is chosen, its parameters must be optimized as the categories definition is made by examples. The examples are the images contained in the training set. However, the selection of a particular training evidently lacks in generality. Any classifier up to the optimization should be challenged with a general problem to void the doubt that it can solve only the particular recognition problem for which it has been optimized. For this reason the test step is required. Here the detector is used over the images belonging to the testing set. The test aims to evaluate the actual efficacy that the classifier can reach when asked to solve the target problem on never before seen images. In fact, testing set classes are also known and confronting classification results with true labels is possible in order to evaluate the performance of this method.

A classifier can be evaluated through its accuracy (percentage of well classified samples) or precision (percentage of true positive, i.e. the percentage of samples

belonging to a particular class and evaluated as being part of the right class). The first evaluating approach is usually followed when there is not a preferred class and the aim is to equally distinguish classes. The latter is used when the goal of the classification is to recognize all samples belonging to a specific class. In this work we prefer to use the true negative rate in order to measure the performance of our classifiers because the approach followed required to set in advance the estimate value of true positive rate.

		Condition	
		Positive	*Negative*
Test	*Positive*	True Positive	False Positive
	Negative	False Negative	True Negative

Table 4.1: Confusion matrix of a binary classifier. In our case the "training class", i.e. the class of original images, is the Positive.

Name	Definition
True Positive Rate	TPR := $\frac{TP}{P}$
True Negative Rate	TNR := $\frac{TN}{N}$
False Positive Rate	FPR := $\frac{FP}{P}$
False Negative Rate	FNR := $\frac{FN}{P}$
Accuracy	ACC := $\frac{TP+TN}{P+N}$

Table 4.2: Statistic based on a binary confusion matrix.

Usually decision fusions performing at the feature level have some drawbacks as high features dimensionality or preparation of training set. In our case the former situation does not subsist and the latter problem is solved with the use of a One-Class Classification technique.

The purpose of this work is to distinguish pristine images from modified images, regardless of the by process that was applied on them. With this aim we choose to compose our training set with never-processed images only. This choice is directed at making or classifier as general as possible and to make it able to detect all considered processes indifferently.

4.1.1 One-Class Classification

To solve our problem in a traditional way a set of original and a set of non-original images is required and a clever distance function must be developed. This way of looking at the problem requires a definition of non-original images. In this work we

study the classification problem considering only five kinds of manipulation but the idea is to create a structure capable of recognizing as much modification as possible, whatever it is. This purpose makes it quite impossible to define a training set that includes all editing, enhancements and compressions. However in the final stage of the work we want to formulate assumptions only about what the original is. The definition for non-original images will emerge naturally later as what we will not classify as original. Starting from this requirement a particular kind of classifier is used.

In this framework we could develop the classifier with original images only, as labeling each as original will automatically enable us to group them together. With this approach we will skip the definition of non-original for the final optimization.

One-Class Classifiers are a category of binary classifiers which training set is composed by one class only. They are used with the purpose of outliers detection or in presence of a particular dataset. The aim of these techniques is to describe as well as possible the "training class", considered the "positive class", and to recognize all new samples that are different from the training class and are considered the "negative class". Actually, compared with *Multi-Class Classifiers*, OC-classifiers are less precise because of less information contained in the training set, but they are essential in some particular situations.

In this chapter two different kinds of binary classifiers are described. The first is based on principal component analysis. The main idea is taking advantage of particular shape of dataset and making the feature representation of training set as spherical as possible. The second technique used is called Support Vector Machine (SVM), a classifier founded on solving an optimization problem and, although it is linear, can be adapted to non linear cases.

4.2 Spherical classifier

The simplest way to build a One-Class classifier is to englobe the training set as well as possible in the feature space with a multidimensional surface. The surface is intended to separate the features space in two regions, the smaller one that includes samples recognized as belonging to the training class, the second, the larger one, in which we find all other samples.

Thanks to the compactness of our dataset we can guess that the best surface is a closed surface that contains the training class: new samples that lie outside the frontier do not belong to this class.

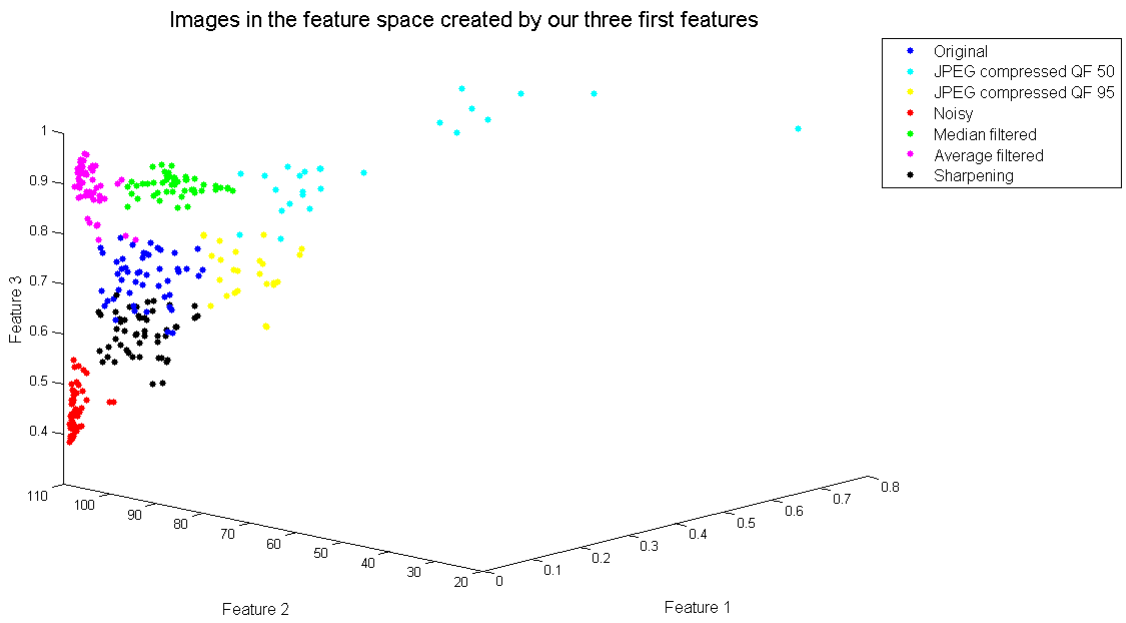


Figure 4.1: Visualization of 50 random images modified with different processing operations in the vector space constituted by the first 3 features.

One of the simplest surfaces that can englobe a compact set of samples is a hypersphere.

In this situation we can define the decision function as

$$f(\mathbf{x}_i) = \begin{cases} 1 & \text{, if } ||\mathbf{x}_i||^2 \leq r \\ -1 & \text{, if } ||\mathbf{x}_i||^2 > r \end{cases} \tag{4.1}$$

Where $\mathbf{x}_i$ is the vector in the feature space representing the i-th sample and r is the radius of the hypersphere. 1,-1 are labels that indicate respectively "original class" and "modified class".

In order to make the values comparable and to enhance numerical stability, we normalise the data so that the average of each feature becomes in 0 and the variance becomes 1. Let $\mu_{x_j}^{tr}$ be the mean value of the $j-th$ feature evaluated only on the training set and let $\sigma_{x_j}^{tr}$ be its standard deviation, the normalization can be expressed as

$$\widehat{x}_{i,j}^{tr} = \frac{x_{i,j}^{tr} - \mu_{x_j}^{tr}}{\sigma_{x_j}^{tr}} \tag{4.2}$$

Normalization allows us to compare features that have values in different ranges without losing relative relationships between samples.

4.2.1 Principal Component Analysis

Although data are normalized their volume in the feature space can be still far from a spherical shape. To enhance the roundness a Principal Component Analysis (PCA) can be applied to remove any ellipsoidal shape in the Gaussian distribution approximation.

As matter of fact, features can be seen as the realization of random variables with a certain distribution and a certain correlation. Information of dataset variance is stored in the *covariance matrix*. The latter is a matrix of size NxN where N is the total number of scalar features of dataset; diagonal values are variances of features and non-diagonal values are the correlations between different features. Thus, if all features are uncorrelated, the covariance matrix is diagonal.

Let $\mathbf{x}_j$ be the vector of values for $j-th$ feature and $\mathbf{x}_k$ be the vector of values for $k-th$ feature. Let $\mu_{\mathbf{x}_j}$ be the average of j-th, and μ_{x_k} the average of k-th feature. Then the elements of covariance matrix are defined as

$$\sigma^2_{j,k} = E[(\mathbf{x}_j - \mu_{\mathbf{x}_j})(\mathbf{x}_k - \mu_{\mathbf{x}_k})] \tag{4.3}$$

Because of its definition, the covariance matrix is a symmetric positive-definite matrix. Thus, all its eigenvalues are positive and its eigenvector are orthogonal to each other according to the Spectral Theorem. Geometrically the figure associated to this linear application is the ellipsoid that best describes the variance of the dataset: a bidimensional example is shown in fig (4.2). Eigenvectors are directions of maximum variance of the dataset and eigenvalues are the length of the square of semi-axis.

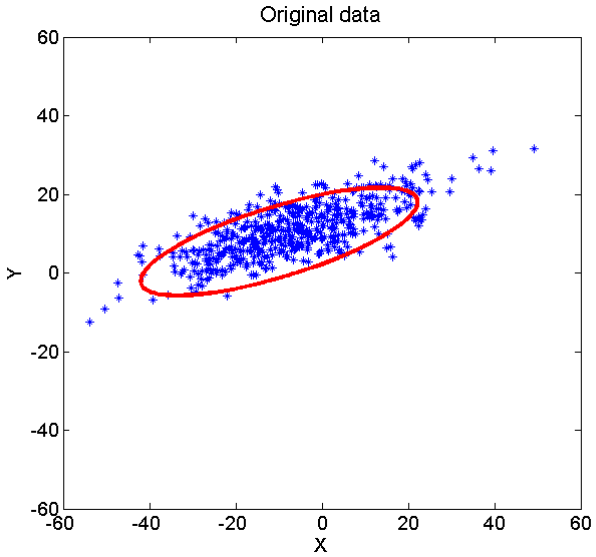


Figure 4.2: Representation of the ellipse described by correlation matrix of distribution.

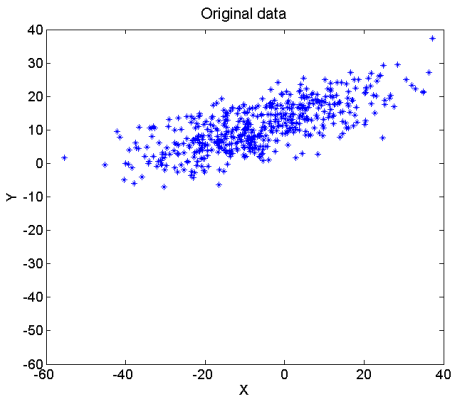
Because in covariance matrix information about the spread of data is stored, this linear application can be used in order to make the training set as spherical as possible in normal distribution approximation. The procedure is to calculate the ellipsoid and transform the feature in the space where the covariance matrix is diagonal. Then each new feature value is divided by the corresponding standard deviation, thus reducing the ellipsoid to a sphere.

Let C be the covariance matrix of the training set, let T be the eigenvectors matrix and Λ the diagonal matrix of eigenvalues. As T is an orthogonal matrix and for this reason $T^{-1} = T^T$. The covariance matrix can be decomposed as

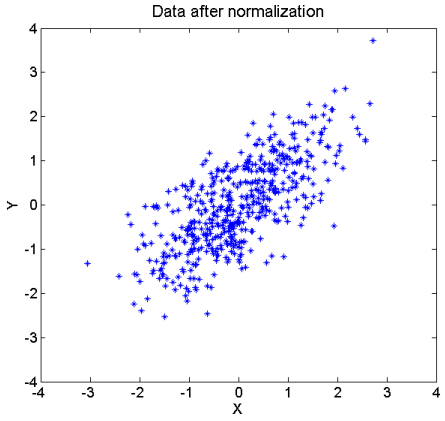
$$C = T\Lambda T^T \tag{4.4}$$

In order to make the spread of training set uniform in all directions a linear transformation is applied to the data. With the multiplication for the inverse of the eigenvectors matrix T, data are rotated so that the cardinal axes of feature space coincide with the ellipsoid axis. The application of inverse of square root of diagonal eigenvalues matrix Λ makes it so that all axis have the same length equal to 1. The whole process can be expressed as the following equation, where $\mathbf{x}$ is the original feature vector and $\mathbf{x}'$ is the transformed vector in the new space.

$$\mathbf{x}' = \Lambda_{tr}^{-\frac{1}{2}} \cdot T_{tr}^T \cdot \mathbf{x} \tag{4.5}$$



(a) *Original data.*



(b) *Data after shift of average and division by standard deviation.*

(c) *Data after normalization and PCA transformation.*

Figure 4.3: Example of the adopted data transformation process to reduce the data volume in a spherical domain. The final distribution has a "circular" shape.

4.2.2 Classifier

Once the representation of training set variance has become "spherical", a hypersphere can be defined around training samples. Hypersphere has as goal to bind the volume enclosing the training set.

With this aim the eigenvectors and eigenvalues of the matrix are evaluated both on training set only. These values are then used to transform all features to the new space.

Thanks to the normalization applied on the training data, the hypersphere can be centered on the origin of the coordinate system.

We fix the radius of the hypersphere so as to have only the 3% of training data misclassified. This is also the estimated error on True Positive. Let z be the random

variable representing the empirical distribution of the training set and r the unknown value of the radius. We set r that satisfies the followed equation:

$$P(z \notin S^r) \leq 0.03 \tag{4.6}$$

The application of this classifier is therefore done in two phases. First the image features are transformed in the new space calculated with the information contained in the covariance matrix of the training set only. Then the decision function f is applied.

For each testing sample the Euclidean distance is evaluated in order to compare it with the hypersphere radius. If the sample lies in the region bounded by hypersphere, then that sample is labeled as belonging to the training class, otherwise it belongs to the other class. The classification algorithm can be mathematically modeled as:

$$f(\mathbf{x}_i) = \begin{cases} 1 & \text{, if } ||\Lambda_{tr}^{-\frac{1}{2}} \cdot T_{tr}^T \cdot \mathbf{x}_i||^2 \leq r \\ -1 & \text{, if } ||\Lambda_{tr}^{-\frac{1}{2}} \cdot T_{tr}^T \cdot \mathbf{x}_i||^2 > r \end{cases} \tag{4.7}$$

4.3 Support Vector Machine

The Support Vector Machines (SVM) are supervised learning methods used in machine learning for classification and regression analysis. They consist in construction of an hyperplane that divides two categories of samples living in a feature vector space as well as possible. Although they are only based on linear methods, they are widespreadly used, as the subdivision in two categories is a common problem. Indeed, in our case we would like to distinguish if an image can be regarded as original or modified, thus constituting a two-class problem. Moreover SVMs offer the possibility to improve their performance from a simple hyperplane classification embedding the vector space in higher dimensional space with a nonlinear function. The transformation can help in splitting the data and in this higher-dimensional space, where a separator hyperplane may be easier to find [13, 28, 28].

We will report the non-linear case only, the linear case is described in detail in appendix B.

4.3.1 Non-linear SVM

SVM can be generalized to cases where no linear hyperplane can split data, as well as the separable formulation gives rise to an amount of errors higher than acceptable. In this case a non-linear formulation is necessary. The *kernel trick* was studied in

1964 by Aizerman [29] and allows to bypass the linearity constrain. This technique is based on the idea that data can be linear separated in higher-dimensional spaces.

Suppose that now we map data from feature space to some higher Hilbert space $\mathcal{H}$, eventually even infinitely dimensional. If the appropriate transformation is used, data can be linearly separated in a new space. There linear SVM formulation can be used to solve the classification problem, after applying the transformation.

$$\Phi : \mathbb{R}^d \to \mathcal{H} \tag{4.8}$$

The increase of dimension for the space, especially if it is infinitely dimensional, would be a problem. However in linear SVM formulation samples appear in dot product combination only.

The scalar product is a function defined in Hilbert spaces that uses two elements returning a real number. The *kernel* function $K(\cdot, \cdot)$ is the dot product function in $\mathcal{H}$ space.

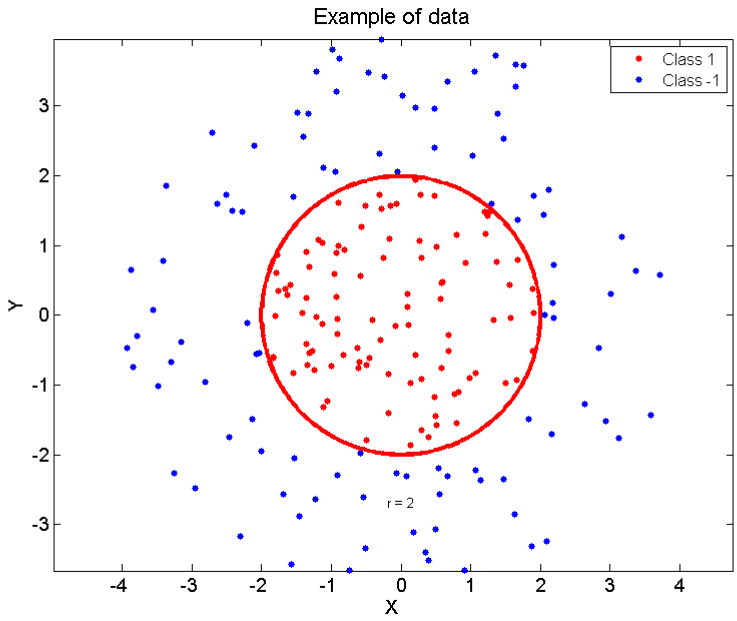
The computation of $\Phi(x)$ can be really difficult and sometimes even impossible. However the function itself is not really necessary as only the kernel, i.e. the scalar product, is involved during the optimization.

The most used kernel functions are

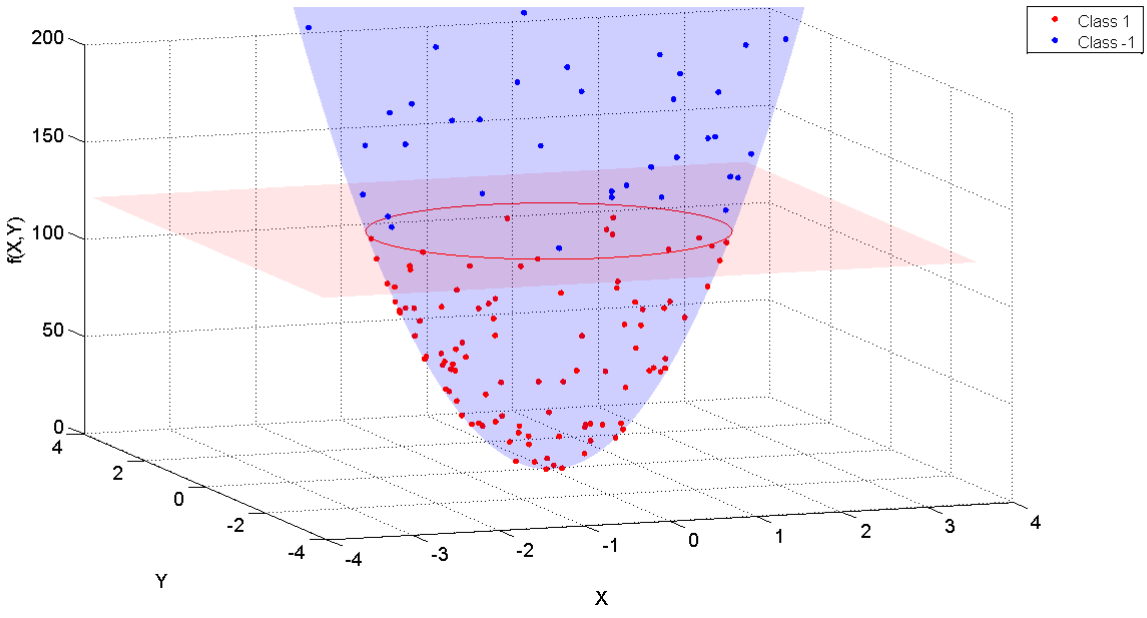
- Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- Polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d \ \gamma > 0$
- Gaussian radial basis function (RBF): $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \ \gamma > 0$
- Sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j + r)$

where $\gamma, r \in \mathbb{R}$ and $d \in \mathbb{N}$.

Figure (4.4) shows an example of bidimensional dataset not linearly separable. Once the starting feature space is mapped in a higher dimensional space a linear surface able to split data is found.



(a) *2D representation*



(b) *3D representation (polynomial kernel)*

Figure 4.4: Example of non-linear SVM with a polynomial kernel. In figure (a) there is no way to divide different classes with a straight line. After transformation in 3D dimension, in fig (b), a plane able to split data exists.

4.3.2 One-Class SVM

As described in [14], SVMs can be used also to solve One-Class classification problems. The idea is to treat the origin as the only member of the second class. Thanks to slack variables, separating one class from the other became possible. Then the standard algorithm is used.

Suppose that a dataset with N elements has a probability distribution P in feature space. Let S be the region of the feature space such that probability that a sample from training set lies outside of it is upper bounded by a parameter value $\nu \in (0, 1)$.

The aim is to define a function that takes value +1 in S and -1 on its complementary.

Thus, the optimization problem that has to be solved is described by the following formulation:

$$\begin{aligned} \min \quad & \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \\ \text{subject to} \quad & (\mathbf{w}^T \cdot \Phi(\mathbf{x}_i)) \geq \rho - \xi_i \\ & \xi_i \geq 0 \qquad\qquad i = 1, ..., N \end{aligned} \tag{4.9}$$

where ρ is the distance from origin and is called *bias term*.

In this case the dual problem is

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T Q \alpha \\ \text{subject to} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu N} \quad i = 1, ..., N \\ & \sum_{i=1}^{N}\alpha_i = 1 \end{aligned} \tag{4.10}$$

where $Q_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$

In One Class non-linear SVMs (OC-SVM) the decision function is define as

$$f(\mathbf{x}_i) = sign(\sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_j) - \rho) \tag{4.11}$$

Chapter 5

Experimental results

5.1 Dataset

In this thesis work we use two datasets. Both are composed of 256-level grayscale never compressed images with different sizes.

All preliminary analysis are done on dataset A so that final tests on dataset B are not biased from individual characteristics own of a single dataset.

5.1.1 Dataset A

Dataset A is composed of two different datasets from *Università di Firenze* and *Universitade de Vigo*.

The former part is created by the team *LESC* (Laboratorio Elaborazione Segnali & Comunicazioni): the dataset is composed of 353 uncompressed TIFF images, representing heterogeneous contents, from three different cameras (Nikon D90, Canon EOS 450D, Canon EOS 5D) [30].

The second image set has been created within the REWIND *REWIND* project, (REVerse engineering of audio-VIsual coNtent Data), and it is composed of 200 uncompressed TIFF digital images by Nikon D60 camera. In this data set all the images have a 3872x2592 pixel dimension and are consisting of indoor and outdoor scenes [31], [32].

We found some duplicates among the images and we removed one image from each pair of identical images. Then each image was cropped into two non-overlapping 1024x1024 pixels matrices and transformed into 256-level grayscale images saved in a lossless format (png). In this way we have 1080 256-level grayscale images with

size 1024x1024 pixels.

5.1.2 Dataset B

Dataset B is the Uncompressed Colour Image Database (UCID). The used version of the database has 1338 uncompressed TIFF images and includes natural scenes and man-made objects, both indoors and outdoors. All images has been taken with a Minolta Dimage 5 digital colour camera with automatic setting [33].

Each image has been transformed in 256-level grayscale and saved in a lossless format (png).

5.2 Experiments setup

In this chapter forensic classifiers based on features previously described are trained and tested. Results are then compared with methods proposed in literature [5], [4], [18].

The procedures for training and testing are described for the classifiers. The analysis of the different approaches is presented to explain the adopted optimization and testing strategies. As a common working baseline for one-class training phase we chose to set the upper bound on the original misclassified images to be equal or less than 3%. The 3% limit was experimentally found to account for the presence of outliers.

One of the main problems in developing a classifier is to show the reproducibility of the results. The adopted strategy to check for stability is based on standard deviations coming from different realizations of classifiers. We choose to use a repeated random sub-sampling validation model where to create a single classifier session, the training set is extracted randomly from the entire set of original images. We can assume that a repeated random sub-sampling validation will limit to the statistical properties of all possible classifiers constructible out of the dataset. So, for each trial, we extract from the whole set of original images, randomly without repetition, the samples to be used to compose the training set and we use the remaining images to create the testing set. Let us note that images used in the training set composition will never be used again in the same trial.

Results for the classifiers are evaluated on a modification basis. The aim is to test each modification class by itself own in opposition to the original class. This method allows us both to understand the behavior of the classifier separately for the different five modifications and to compute the four important rates of binary classification: true positive, true negative, false positive and false negative rate. In

all the tests a balanced set of original-modified images has always been taken into consideration.

A calculation out of the previous five analyses (one for each modification) was performed to evaluate an average behavior, assuming randomly chosen modifications.

Since the development of this work was based on dataset A it was desirable to obtain a second dataset for testing purposes. This way our proposed methods could be tested for validating the generality of the adopted model. Keeping the same structure used in the previous case, new training sets and testing sets are extracted from a dataset B 5.1.2. All procedures described above for dataset A are repeated on dataset B and the two groups results are compared with each other.

5.3 Data organization

The adopted classification methods require the division of the dataset into two subsets: one for the training procedure and one for the testing phase. The amount of images required to compose the training set will be calculated in the next section. All the images that are not included in training set are used for testing the performance of the classifiers and composing the testing set. Furthermore during the testing phase it is of primary importance to determine if and how the classifier can recognize a processed image especially for the selected modification. For this reason for the image belonging to the testing set several modified versions are generated. Every image of the testing set is thus used several times in order to test the classifier under different manipulations. The list of selected modifications for testing is as follows:

- original images;
- JPEG-compressed images
  - with quality factor 50;
  - with quality factor 60;
  - with quality factor 70;
  - with quality factor 80;
  - with quality factor 90;
  - with quality factor 95;
- images with addition of Gaussian noise
  - with mean 0 and variance 0.0001;

  - with mean 0 and variance 0.0002;
  - with mean 0 and variance 0.0005;
  - with mean 0 and variance 0.001;
- median filtered images
  - with a 3x3 window;
  - with a 5x5 window;
- average filtered images
  - with a 3x3 window;
  - with a 5x5 window;
- sharpening filtered images
  - with variance 0.4;
  - with variance 1.0;
  - with variance 1.5;

5.4 Spherical classifier

In the spherical classifier method we transform the feature space in order to enclose the original images inside a volume defined by a sphere. The radius of the sphere is the actual parameter to optimize since everything that falls inside will be considered original, otherwise it will be classified as a processed image. For the computation of the characteristic threshold distance we evaluate an optimum cardinality for the training set to ensure good repeatability in obtained values.

Here, as in the other cases, we applied a 3% limit on the misclassification of positive class result as the training criterion for the optimization process. In other words the radius is chosen in order to include 97% of the training original images.

Cardinality of training set

Training set cardinality is an important value to be determined in order to obtain good characterization of the problem, without incurring over fitting. As the latter problem is probably of secondary relevance for this classifier the first is definitely to be considered. In fact, the fewer the training elements, the harder it is for the classifier to correctly determine the characteristic sphere.

A measure for appropriate behaviour is obtained looking for stability in the reproducibility of the radius or, in other words, for stability and performance in original images classification efficiency. It is expected of both quantities to be substantially unaffected by changing training set as they should characterize the abstract original-images class and not the particular properties of the training set. However stability cannot be measured with a single classifier, but must be evaluated out of many different realizations. For different cardinalities we tested 100 different classifiers using the random sub-sampling validation model to get either average value or variance for the efficiency. This procedure allows us to estimate the optimal cardinality for the training set. Our data reported in figure (5.1) show how the increase of the number of training images affects the mean value and standard deviation of true positive rate.

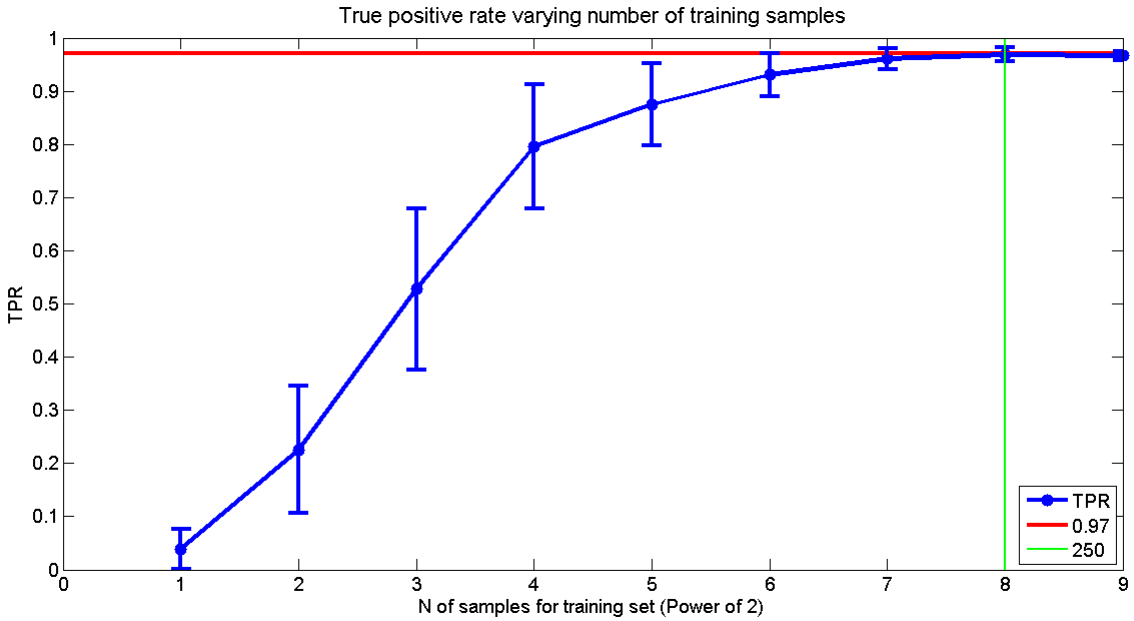


Figure 5.1: Rate of original images classified as original evaluated with a different number of training samples for the spherical classifier. (Mean and standard deviation evaluated on 100 trials). The stability at the required true positive ratio is reached by a training set with a cardinality of 250 samples.

We choose as training set cardinality the number 250.

Results

Five different testing sets are built, each one containing the original images that are not used in the training set and a modified version of these images subject to one

particular processing operation so that we have a testing set composed of original images and, for instance, JPEG-compressed images.

A spherical classifier is built starting from the training set: all data are normalized and are transformed in space where training samples form a sphere-like shape as described in chapter 4.

In order to illustrate the performance of the spherical classifier, the receiver operating characteristic (ROC) curve is plotted (fig (5.2)). A hypersphere centered in the middle of the training samples with a variable radius is created. For each value of the radius the position of testing samples is studied and the number of testing images assigned at the right class is counted. One point in the ROC space represents a spherical classifier with a specific radius.

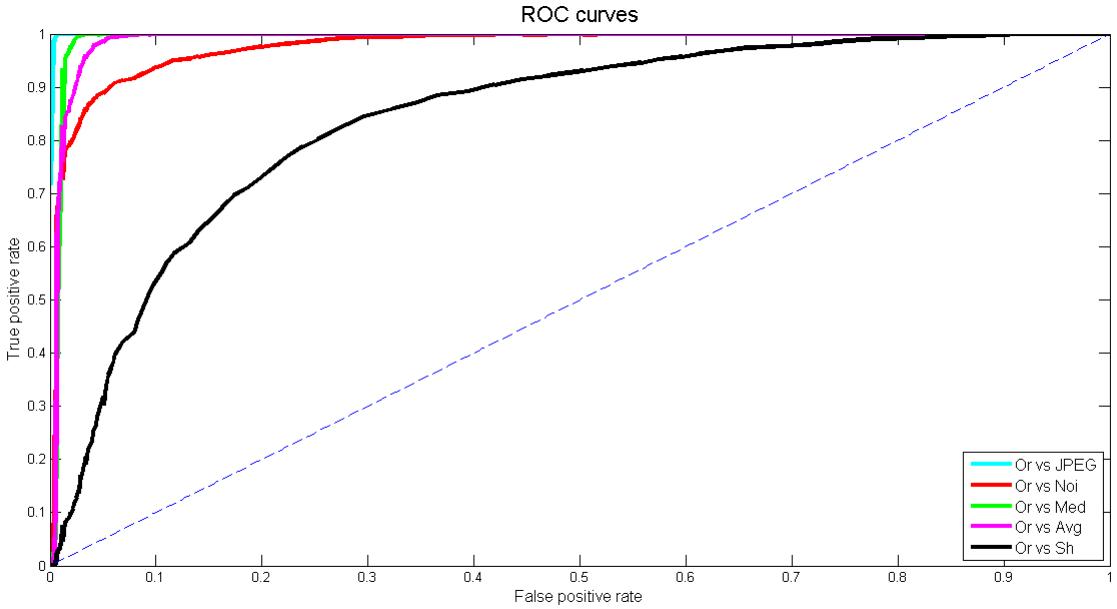


Figure 5.2: ROC curves of spherical classifiers trained with the same training set on dataset A. JPEG-compressed images are classified almost perfectly; the images that are less recognized are those subject to sharpening filtering.

The spherical classifier is then tested on a different dataset to fairly evaluate the effectiveness of the adopted classification framework. Keeping the same structure used in the previous case, new training sets and testing sets are extracted from a dataset B [33]. All procedure described above for dataset A is repeated on dataset B with the following results.

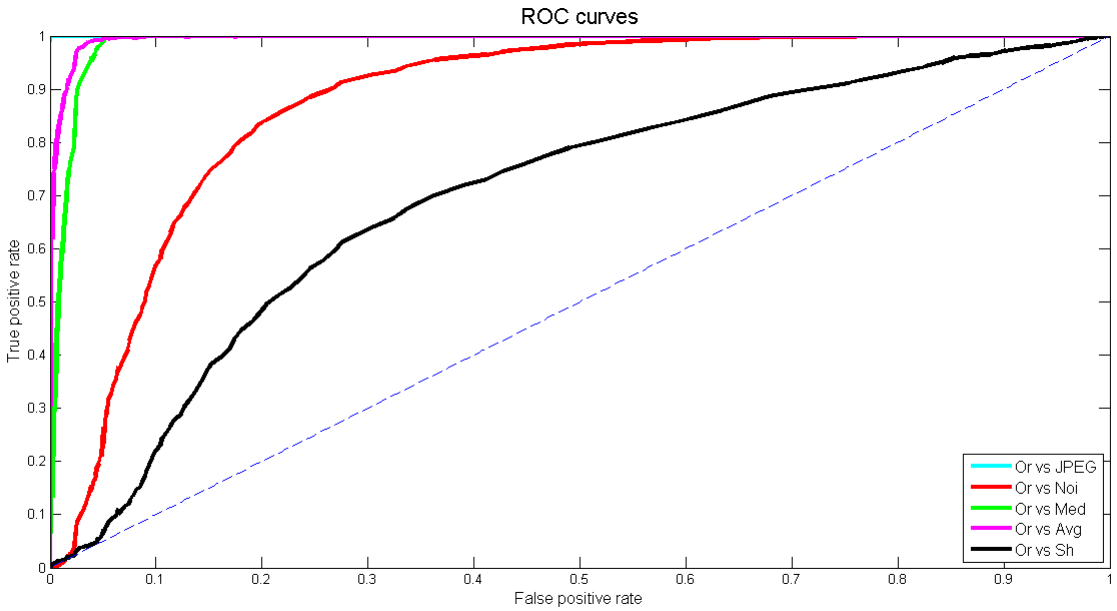


Figure 5.3: ROC curves of spherical classifiers trained with the same training set on dataset B. JPEG-compressed images are classified almost perfectly and the images that are less recognized are those subject to sharpening filtering. The ability of the classifier to recognize noisy images in dataset B is clearly lower than that in dataset A.

Different performances of spherical classifier on dataset A and dataset B can be explained with two main reasons. The first is related to the fact that features are designed to fit on dataset A. Dataset B has never been used to test or verify the effectiveness of features. The second reason is that in dataset B the range of images quality is larger than in dataset A. The detectable noise threshold depends on the amount of noise already present in the images composing the training set. In fact is the training set deciding what should be original, so if noise images are allowed for training the detectable added noise is consequentially stronger.

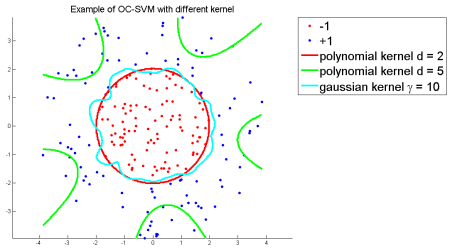
5.5 One-class support vector machine

To go beyond the results of the spherical classifier we build a One-Class Non Linear Support Vector Machine. The SVM should be more flexible in the definition of the original images volume and therefore we expect a better characterization of the class.

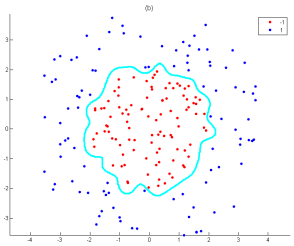
To perform SVM classification we used LIBSVM [34], a free integrated software for support vector classification, regression and distribution. The library provides

an interface with software packages or programming languages like MATLAB, C++, R, RapidMiner and others.

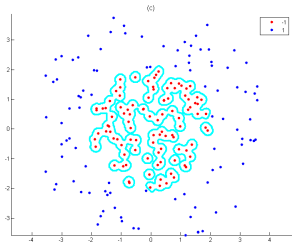
The chosen SVN kernel is the radial basis function (RBF), because of its ability to fit data with structure similar to ours.



(a) *RBF kernel (γ = 1).*



(b) *RBF kernel (γ = 10).*



(c) *RBF kernel (γ = 100).*

Figure 5.4: Example of OC-SVM with different kernels. The higher the parameter γ, the higher the risk of incurring in overfitting.

Parameter γ is set as described in [20]

$$\gamma = \frac{1}{\eta^2} \tag{5.1}$$

Where η is the median value of Euclidean distances between samples in the feature space.

Cardinality of training set

As for the spherical classifier, the number of images included in the training set is estimated for the OC-SVM classifier. As value for ν, similar to the spherical classifier, we choose 0.03. Let us remember that ν represents the upper bound of error in the training set, it does not indicate the error itself.

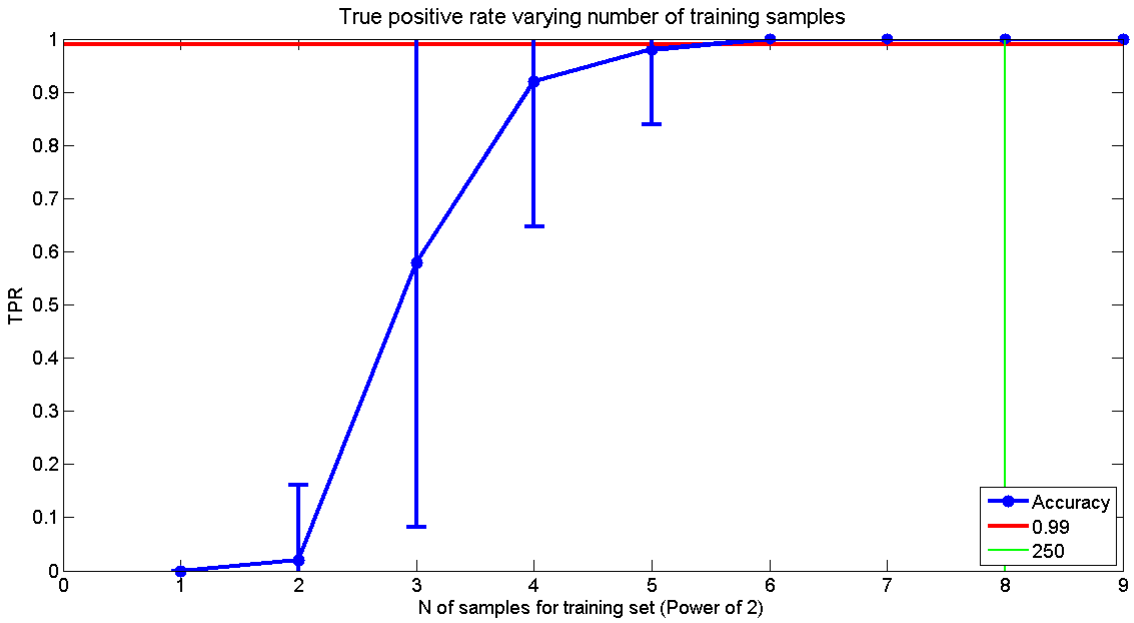


Figure 5.5: Rate of original images classified as original evaluated with different number of training samples for spherical classifier. (Mean and standard deviation evaluated on 100 trials). The stability is reached by a training set with a cardinality of 250 samples.

Figure 5.5 allows us to keep the parallelism with the spherical classifier and to fix the number of images needed to build the training set as 250.

5.6 One-class support vector machine with principal component analysis dimensionality reduction

Although the number of considered features is really low compared with methods present in literature [4], [18], we investigate on a possible dimensionality reduction with a Principal Components Analysis (PCA). The analysis shows our last effort for creating a strategy capable of keeping the dimensionality of the problem contained. Thanks to the achieved results we will also be able to present our result in the feature space, realizable only with maximum of three variables.

Principal Component Analysis

For this analysis original images are clearly not sufficient to describe the desired variance caused by the modification as we need to investigate the distribution induced by those modifications. For this purpose images are randomly chosen so as to have:

- 20 original images;
- 20 JPEG-compressed images with quality factor randomly chosen in {50,60,70,80,90,95};
- 20 noisy images with variance randomly chosen in {0.0001, 0.0002, 0.0005, 0.001};
- 20 median filtered images with mask of 3x3 or 5x5;
- 20 average filtered images with mask of 3x3 or 5x5;
- 20 Gaussian sharpening images with random variance chosen in {0.4,1,1.5};

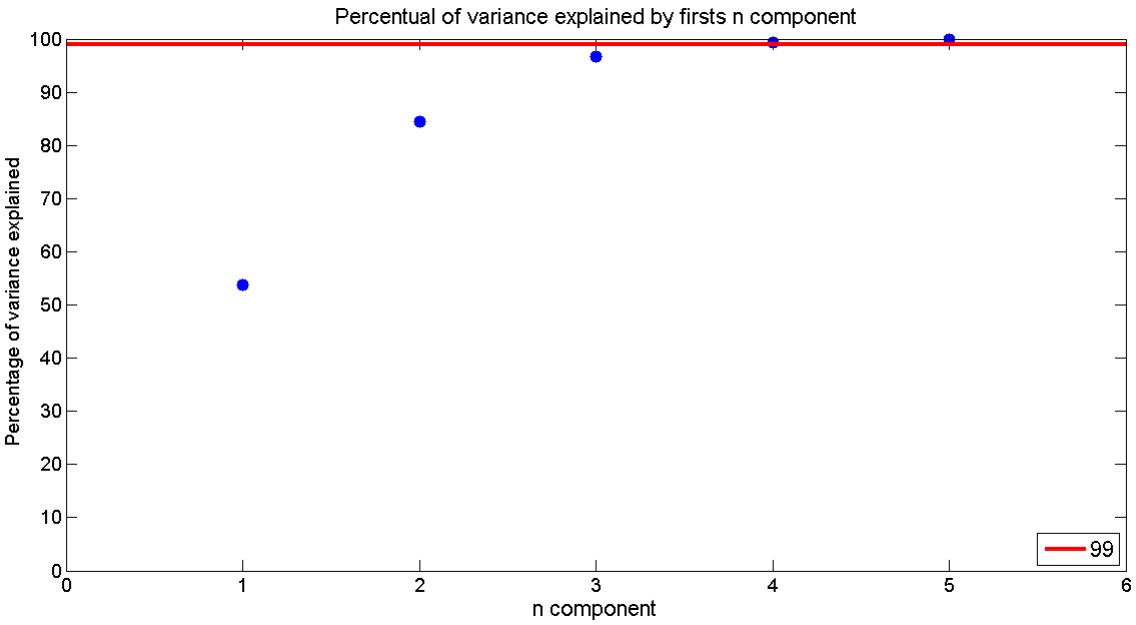


Figure 5.6: Percentage of variance explained by first n component of PCA. The first 3 components can explain more than 99% of variance

The eigenvectors extracted from the diagonalization of the covariance matrix constitute the orthonormal base of the best ellipsoid approximation of the volume

data. We do expect our data to be divided into subregions where original images are separated from the modified one, thus not really of cloud shape. However directions of maximum variance will detect where the cluster of modified and original images better separates one from the others. This way, looking at the percentage of variance measurable in each direction, we can identify the privileged direction in feature space where the separation is made more evident. Keeping the most important directions and disregarding the others is a way to simplify the classification space, while preserving the information coming from a richer dimensional problem. Therefore the analysis of eigenvalues can show how much a direction is important in data representation. In our case the cumulative sum of eigenvalue shows that the first three principal components are already sufficient to explain the 99% of data variance.

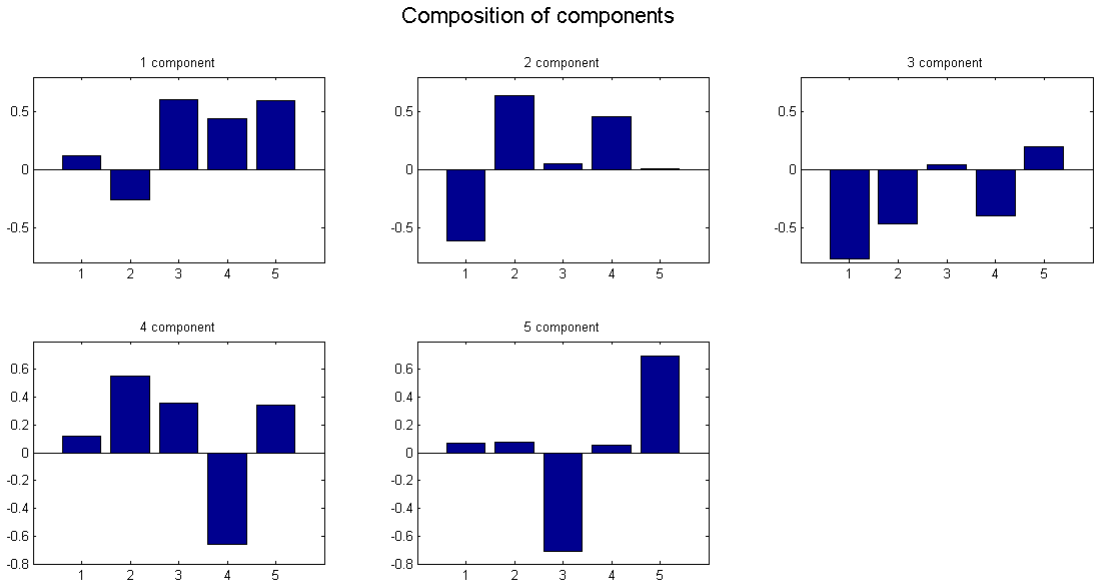


Figure 5.7: A qualitative explanation can be traced back from the basic principle that we developed in chapter 3. With no aim to make here rigorous statements we suggest that the first component, composed mainly of F_3,F_4 and F_5, explains the variance due to filters. It is possible that the second one compares compressed images with others. The third could catch the difference between median filtering and other manipulations. We suppose that the fourth is concentrated on variance due to average filtering and the contribution of F_4; the last one shows the difference between F_3 and F_5.

A reduction of dimensionality through the PCA method will have some advantage. To explain 99% of data variance using three variables only is really convenient, especially because the 3D plot of the solution can be easily illustrated. On the other hand, PCA computation needs a set of images that includes all kinds of modifications and not only originals. This can be implemented with the introduction of validation set used for PCA computation. Analysis shows that, eliminating the last two components, our dataset does not lose too many information.

Principal Component One-Class Support Vector Machine

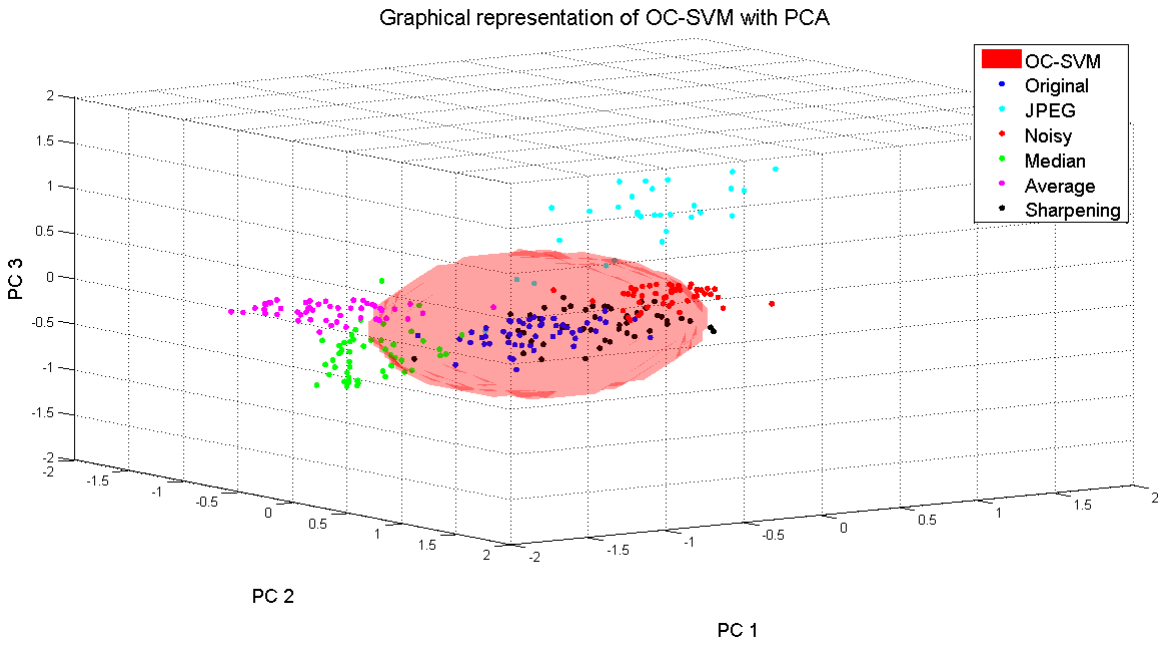


Figure 5.8: 3-D representation of region in PCA space where images are considered original by OC-SVM with PCA reduction.

Although there are only five features extracted, principal component analysis showed that a reduction of dimensionality can be performed without losing too much information. In fact the last two principal components found in section 5.6 explain less than 1% of the system variance.

On the other hand, PCA computation has to be done on a subset that includes all kinds of considered manipulations and not only on original images. This is voids partially the claimed generality of the approach for dimensionality reduction. However we will persist in using one-class trained classifier.

We divide the dataset in 3 parts. The first is composed of 120 images, each one used only once and subjected to a specific process (as described in PCA dataset preparation, section 5.6). These images and their modified version will never be

considered again in this classification, neither in training set nor in testing set. The second is the training set and includes 250 pristine images. The remaining images form the testing set and each testing image is considered with all kinds of manipulations.

5.7 Classifiers comparison

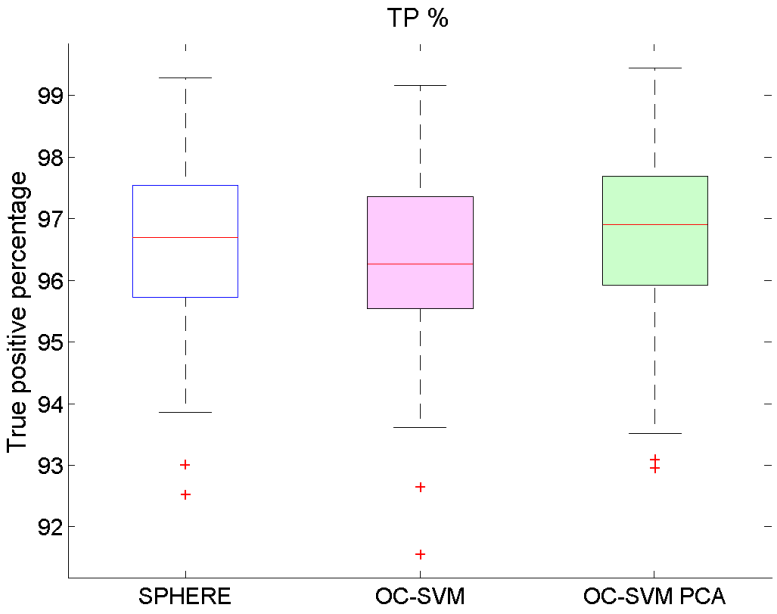
In this section the performances of the three classifiers are compared. Classifiers are trained on original images only and tested for the five image modifications separately. Then we consider the total accuracy as result of the classifiers on a testing set composed 50% of original images and 50% of images with a random modification in those considered.

Dataset	Sph	OC-SVM	PCA OC-SVM
A	88.60 (5.16)	87.76 (1.49)	86.83 (2.02)
B	78.87 (2.17)	84.09 (1.37)	81.98 (1.55)

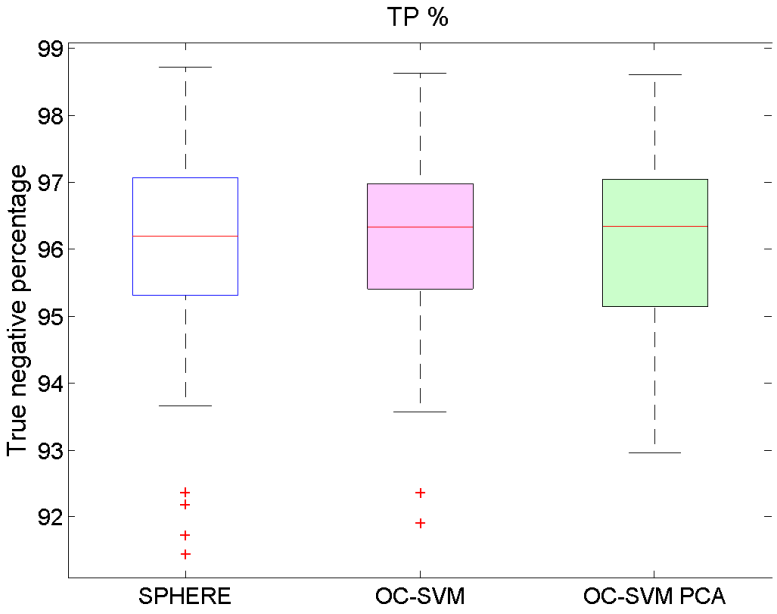
Table 5.1: Mean total accuracy (and standard deviation) of classifiers. For each classifier repeated random sub-sampling validation is used for 100 times.

All classifiers have a total accuracy higher than 78% with higher performances for OC-SVM (tab 5.1). While the spherical classifier has the highest accuracy on dataset A, it behaves worse on dataset B. As expected, the OC-SVM evaluated on reduced feature space has an accuracy slightly lower than the classical OC-SVM.

Here in figure (5.9,5.10, 5.11, 5.12, 5.13, 5.14) we report boxplots showing distribution of TPR and TNR of 100 different training (and testing) set for each kind of considered manipulation.

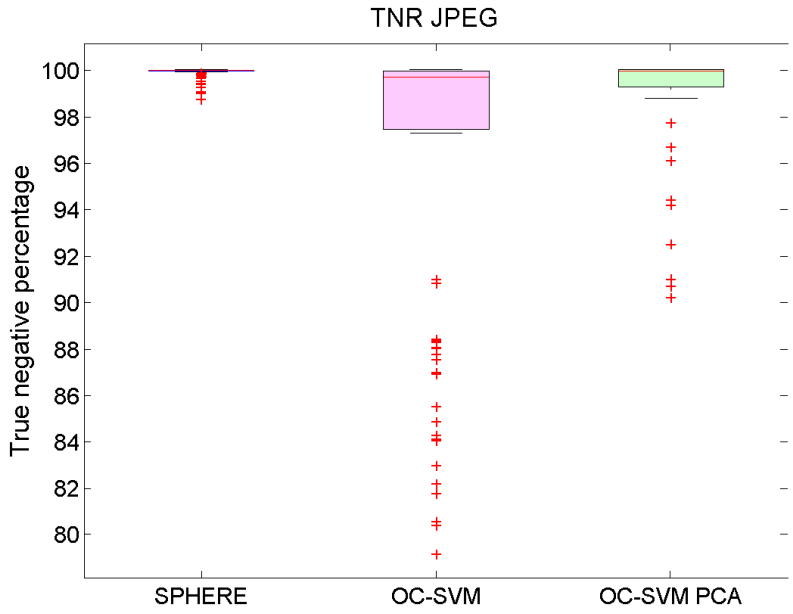


(a) *Dataset A*.

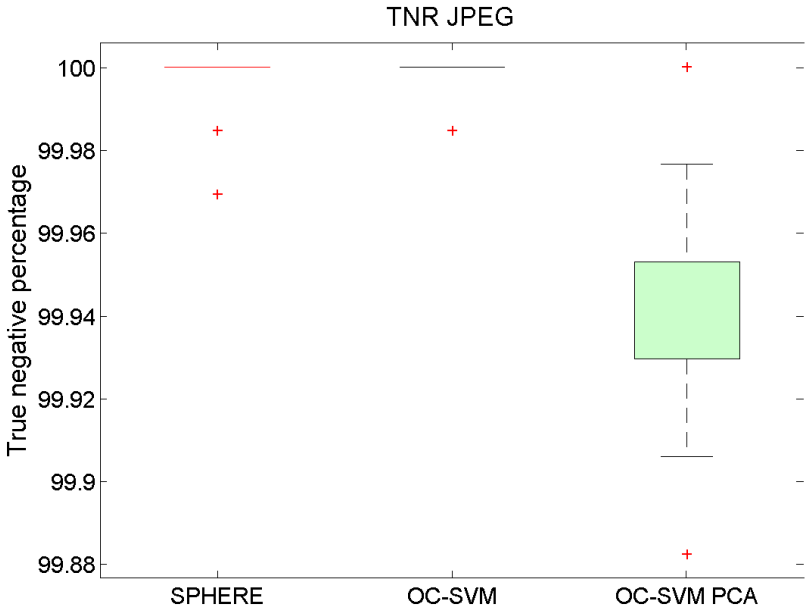


(b) *Dataset B*

Figure 5.9: Percentage of true positive evaluated with different classifiers. Error on true positive is close to the value set by parameter ν (0.03).

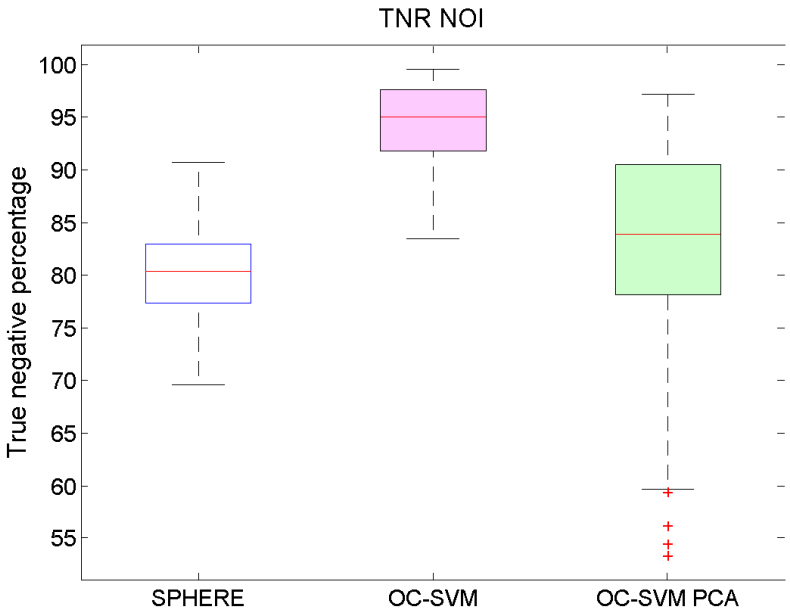


(a) *Dataset A*.

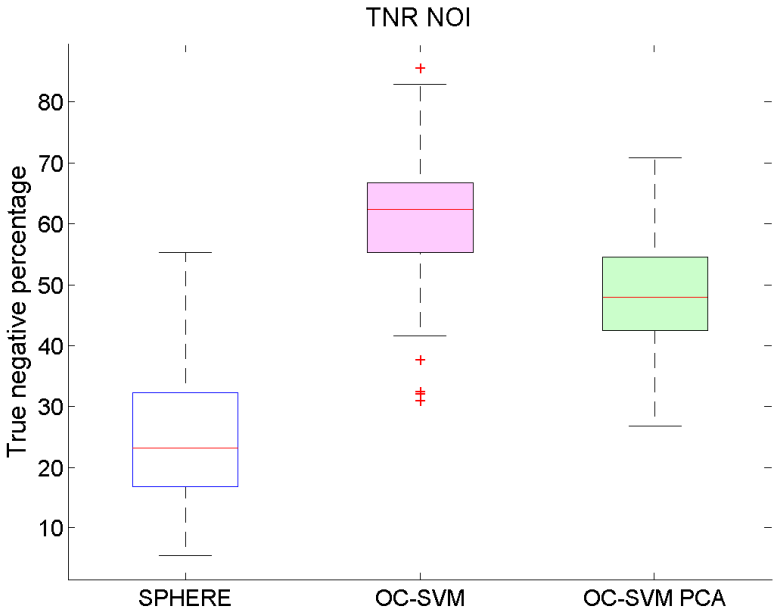


(b) *Dataset B*

Figure 5.10: Percentage of true negative evaluated on JPEG-compressed images. The results obtained with OC-SVM have larger variance than others but the mean is always higher than 98% on dataset A. On dataset B the percentage of compressed images not recognized is inferior than 0.02% if computed with OC-SVM or spherical classifier.

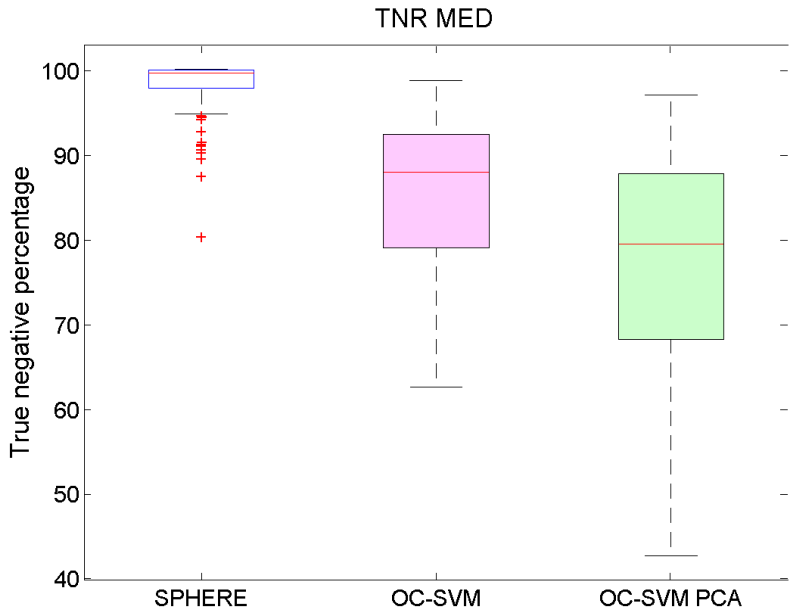


(a) *Dataset A*.

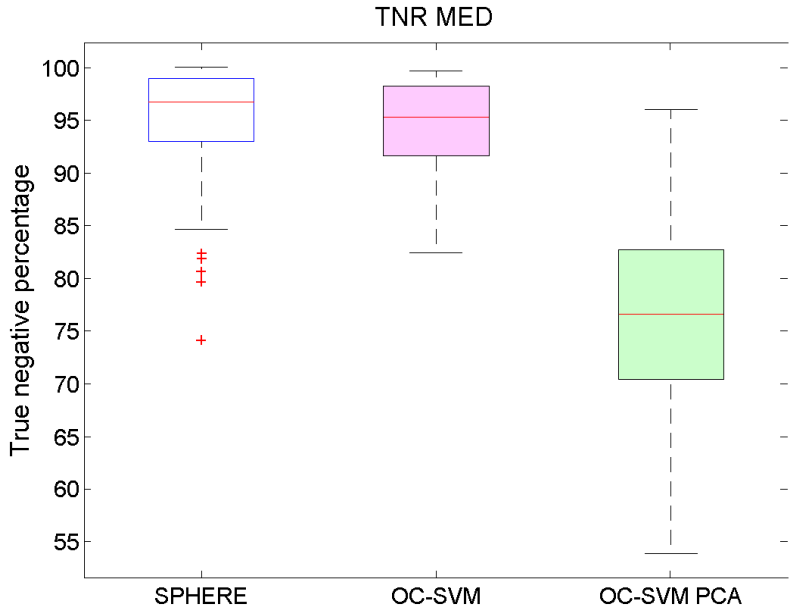


(b) *Dataset B*

Figure 5.11: True negative rate for noisy images. The relationships between classifiers result the same on dataset A and B, but in the second case noisy images are recognized less than 60% of times. We suppose that this behaviour is due to low quality of images belonging to dataset B.

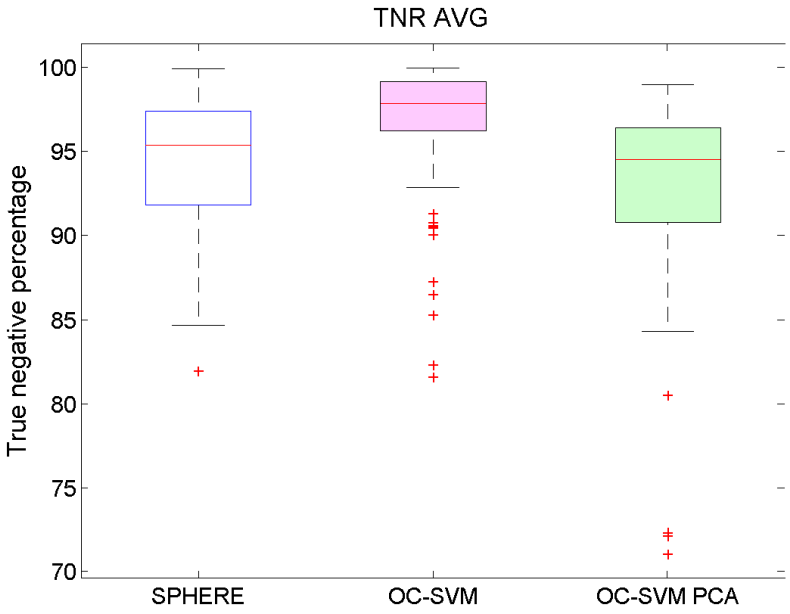


(a) *Dataset A*.

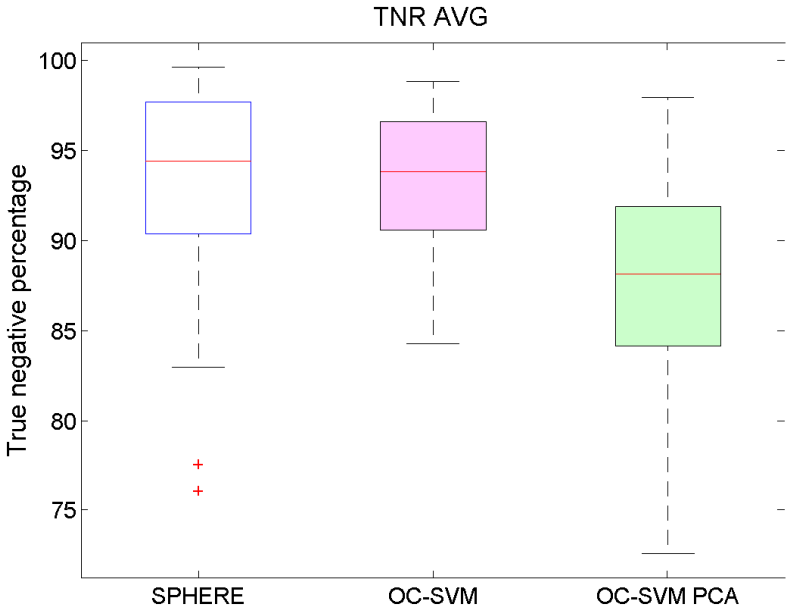


(b) *Dataset B*

Figure 5.12: Percentage of true negative evaluated on median filtered images. Unlike the case of noisy images, detecting median filtered images is easier in dataset B than in dataset A. As predictable, performance of OC-SVM with PCA are slightly lower than that with OC-SVM. In this case the spherical classifier works better than others.

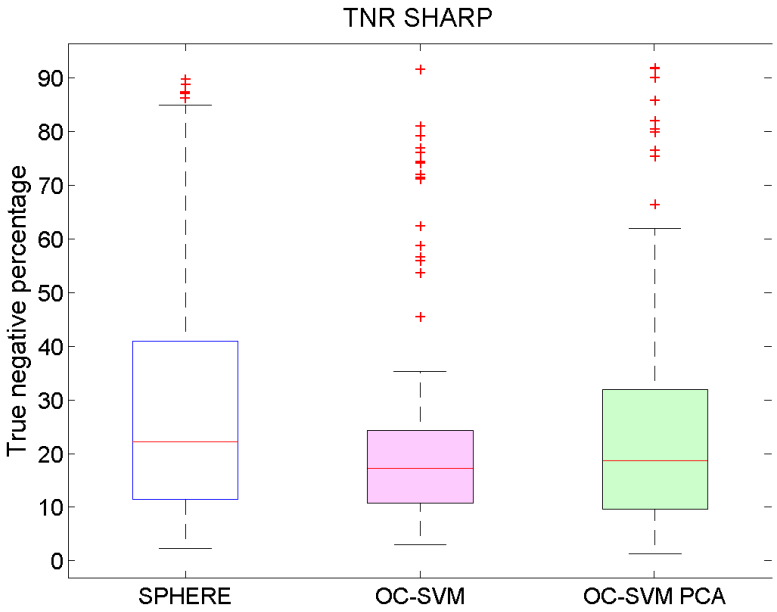


(a) *Dataset A*.

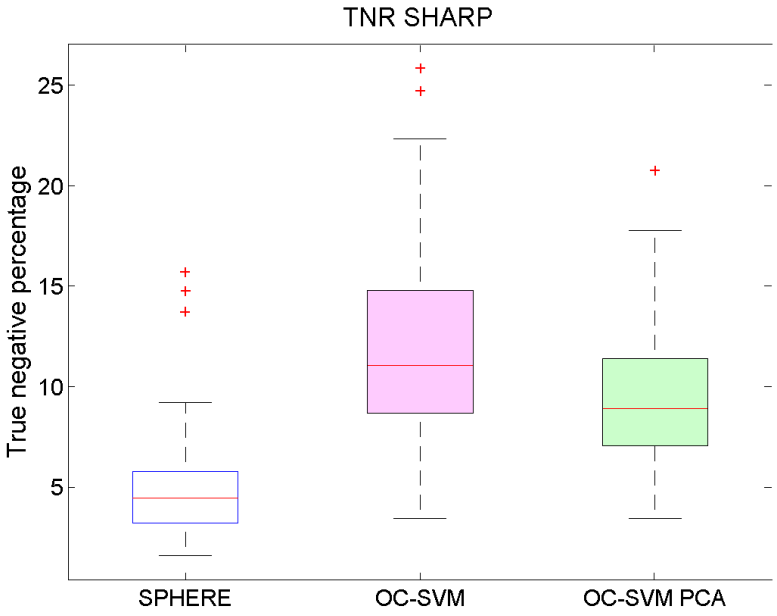


(b) *Dataset B*

Figure 5.13: The percentage of average filtered images recognized is higher than 90 for all classifiers.



(a) *Dataset A*.



(b) *Dataset B*

Figure 5.14: None of the classifiers is capable of recognizing sharpening filtered images.

5.8 Comparison with literature

In this section a comparison of our best performing classifier (OC-SVM) with other methods presented in literature will be described. As we did for the classifier of the previous section, the comparison will be carried out modification by modification. One-Class Support Vector Machine will be trained on original images only, as usual.

The first two features set considered for comparison are based on SPAM [18] and GLF [4] techniques. We consider the low-dimensional version of SPAM features that is 162-dimensional. For GLF the feature dimension is 56.

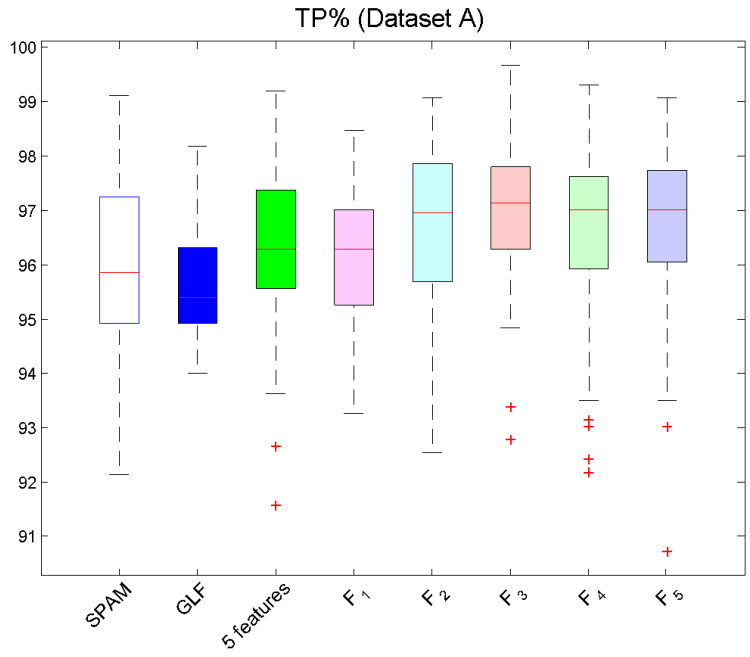
To handle the problem arising from the higher feature dimensionality space a different validation model was adopted. The model validation techniques for these classifiers are the 5 folder cross-validation. One step of cross-validation cycle involves partitioning original images into five complementary subsets. Four folders are used to compose training set, meanwhile images in the fifth folder compose the testing set. To reduce variability, five rounds of cross-validation are performed using different folders. This procedure is repeated 10 times with different random partitions. Let us remember that also in this case we use a OC-SVM: the training set is always composed of original images only, meanwhile the testing sets for single modifications are balanced and include an equal number of original and modified images. Parameter γ is obtained with an optimization procedure and is set to 0.125 for GLF and to 0.025 for SPAM.

Since the 5 features were developed for detecting different image modifications it is an interesting question whether or not the feature itself is adequate for the planned purpose. In addition it would be interesting to show that, as planned, the classifier can gain in performance considering the problem in the complete 5 dimensional space instead of detecting each process separately.

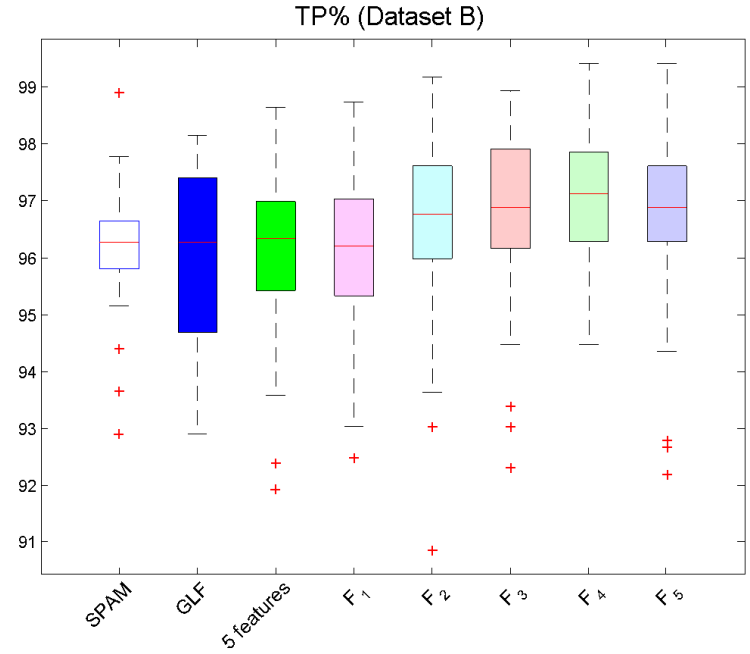
The results are presented in figures (5.15,5.16,5.17,5.18,5.19,5.20,5.21). The comparisons of the different classifiers are in agreement with our prediction and proposed targets.

Dataset	5 features	SPAM	GLF
A	87.76 *(1.49)*	76.64 *(0.57)*	70.06 *(0.62)*
B	84.09 *(1.37)*	78.76 *(1.2)*	62.50 *(0.98)*

Table 5.2: Accuracy evaluated with different feature vectors in a balanced problem where the testing set includes 50% of original images and 50% of images subjected to a random modification

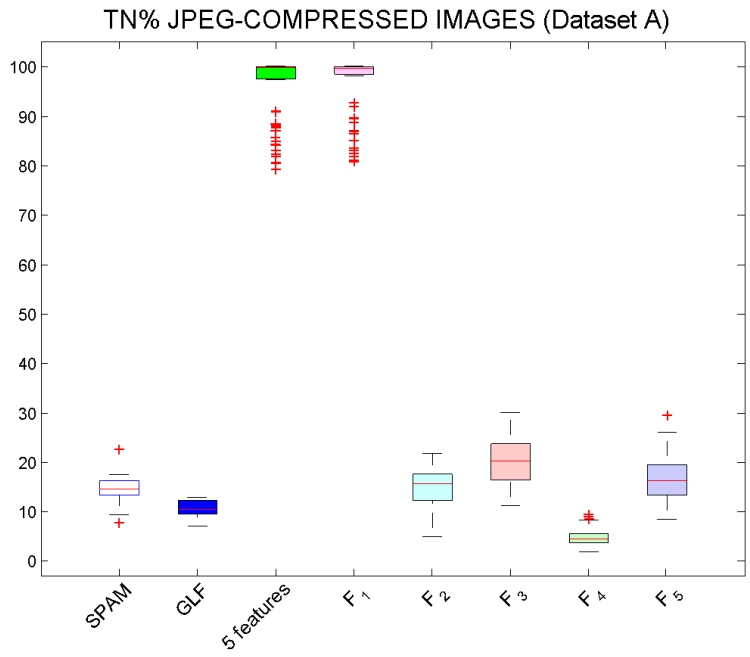


(a) *Dataset A*.

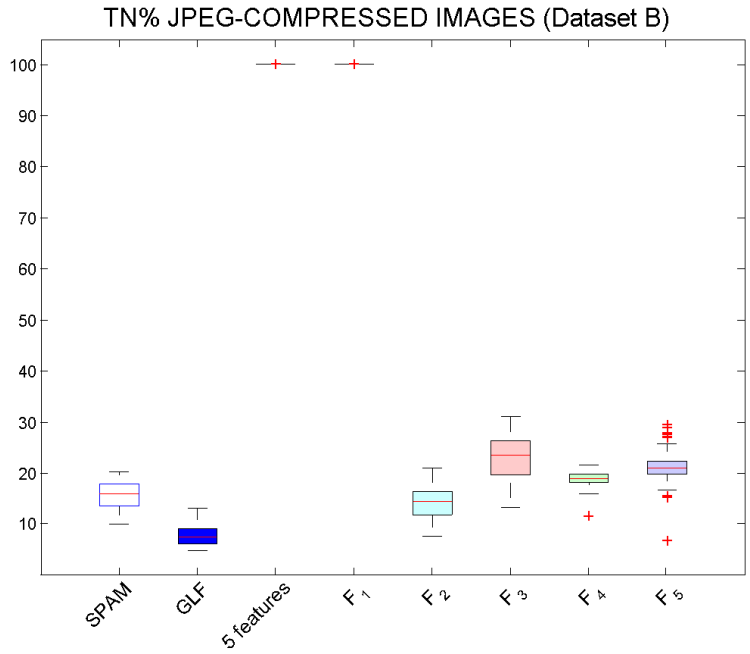


(b) *Dataset B*

Figure 5.15: Percentage of true positive evaluated with OC-SVM using different features. Error on TPR for all classifiers reaches the value set by parameter ν (0.03).

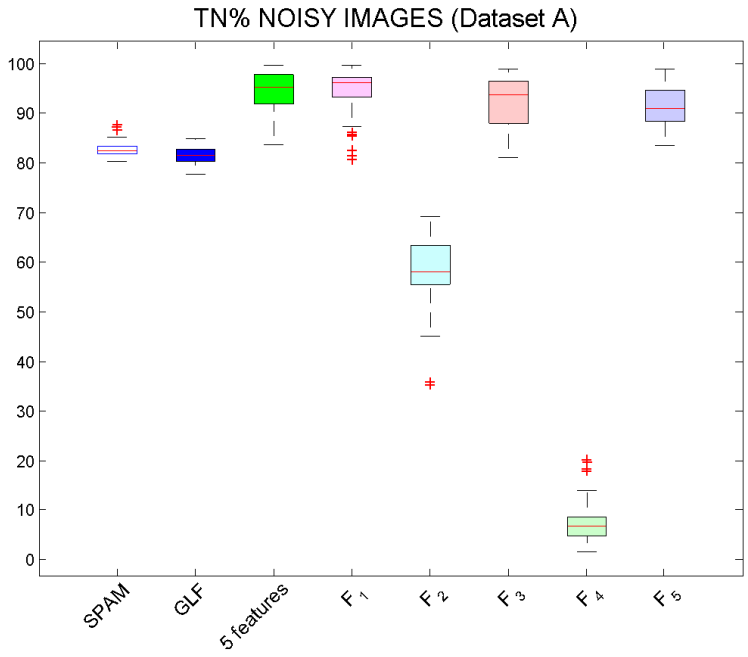


(a) *Dataset A*.

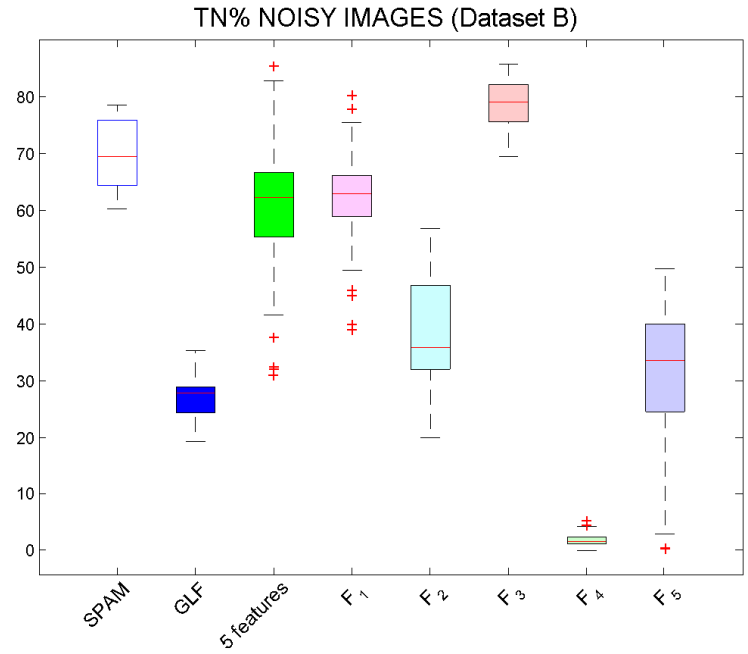


(b) *Dataset B*

Figure 5.16: JPEG-compression is identified very well with all *5 features* and, as predictable, with the feature related to entropy. To this discussion we dedicate the last section of this work. See section 5.8.1 for further details.

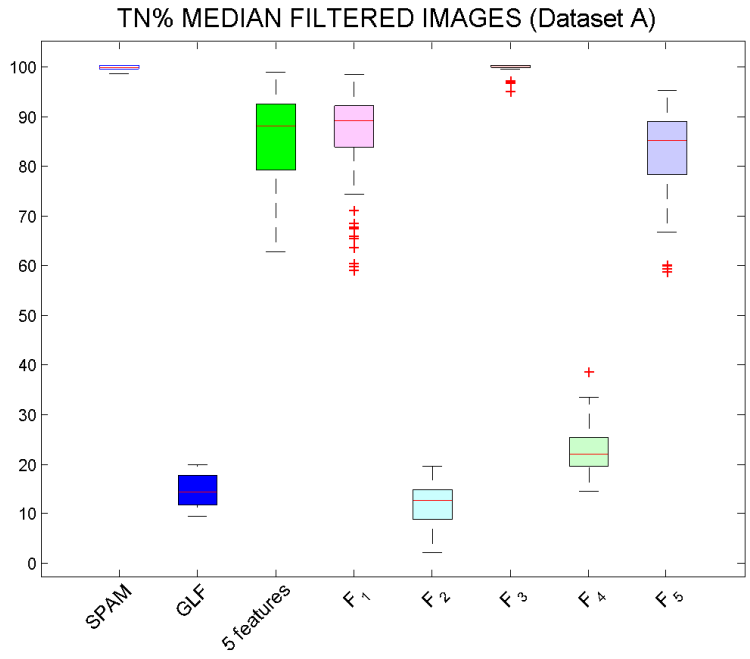


(a) *Dataset A*.

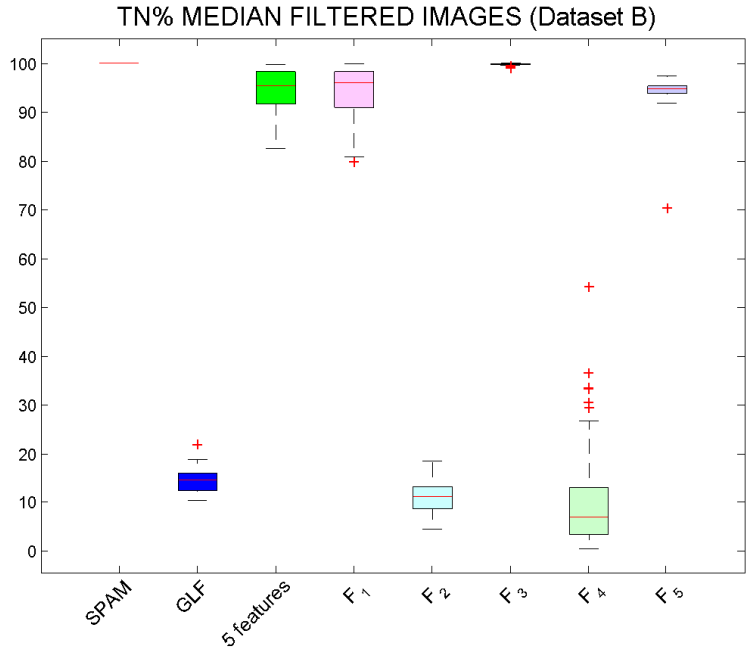


(b) *Dataset B*.

Figure 5.17: SPAM features have good performance in noisy images recognition. In fact the starting idea of SPAM is the detection of embedded messages thanks to a study of noise. *5 features* classifier is competitive (or superior on dataset A) with SPAM and this is due to the contribution of features F_1, F_3 and F_5. Surprisingly F_2, created specifically to recognize noisy images has a percentage of true negative less than 60%.

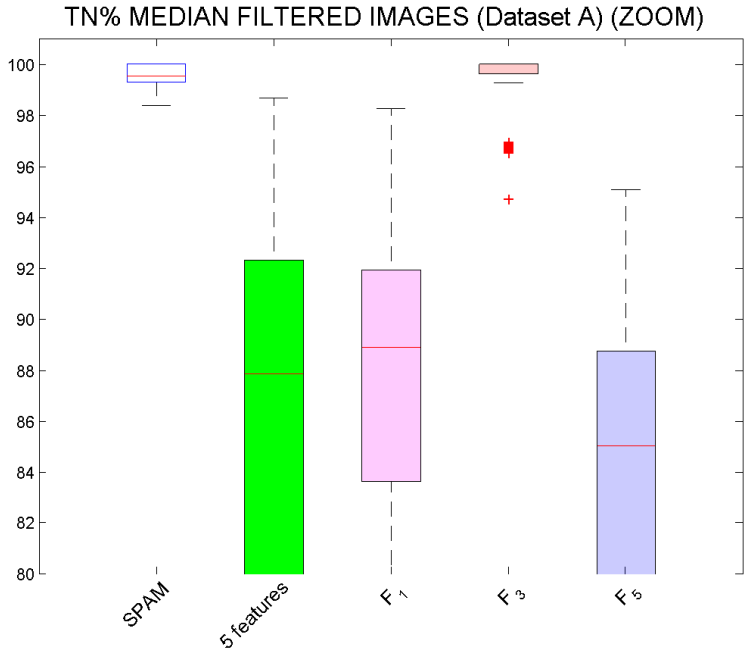


(a) *Dataset A*.

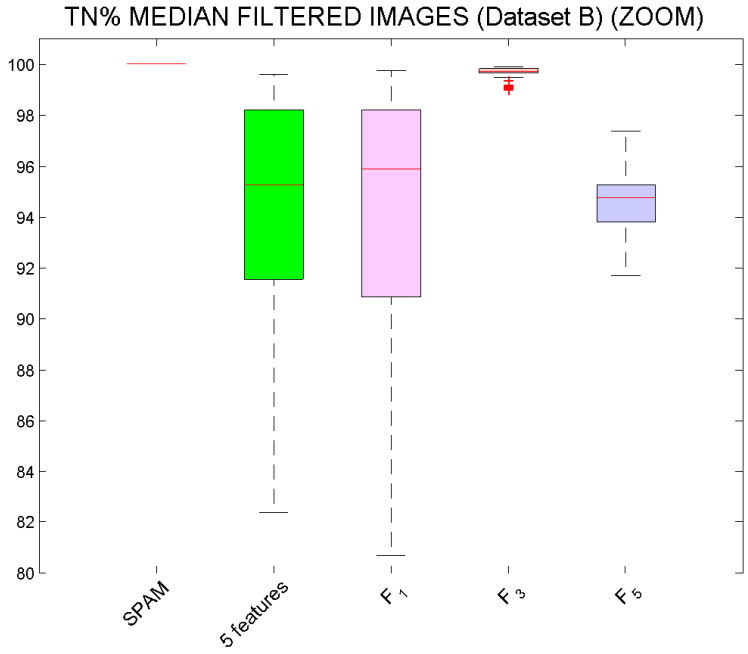


(b) *Dataset B*.

Figure 5.18: SPAM features have definitely the best performance in the recognition of median filtering. *5 features* has a good performance thanks to the contribution of F_1, F_5 and above all F_3 but it is not competitive with SPAM. Unexpectedly GLF is not capable of recognizing images subjected to median filter.

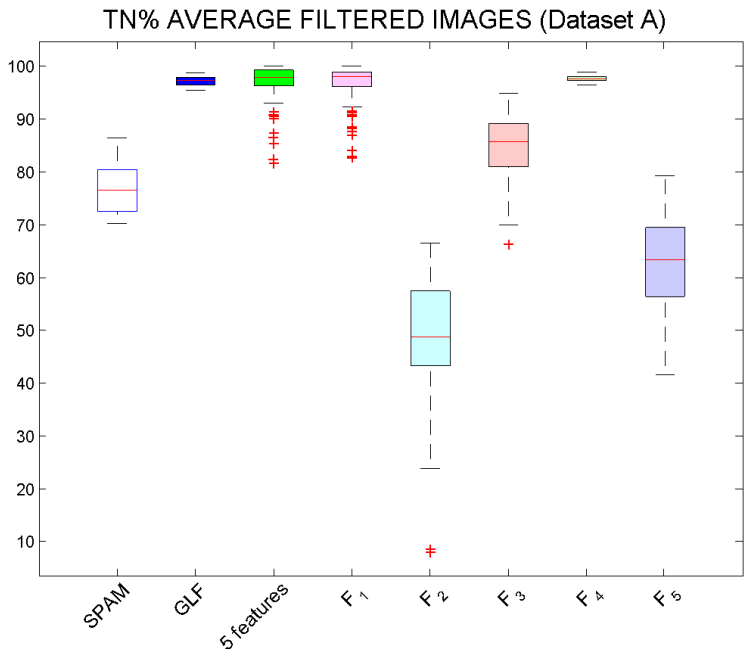


(a) *Dataset A*.

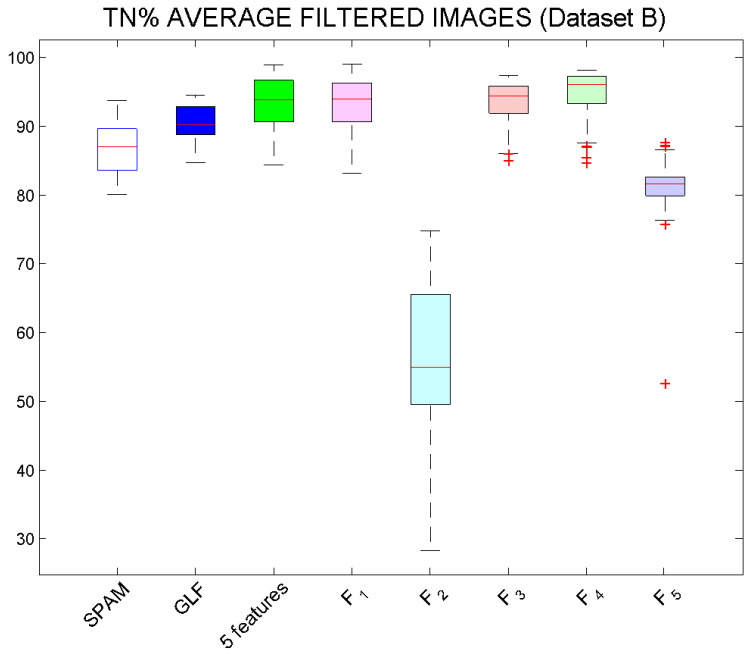


(b) *Dataset B*

Figure 5.19: Particular of figure (5.18) to better visualize the true negative region rates between 80 and 100%. From the plot it is possible to appreciate the superior performance of SPAM features

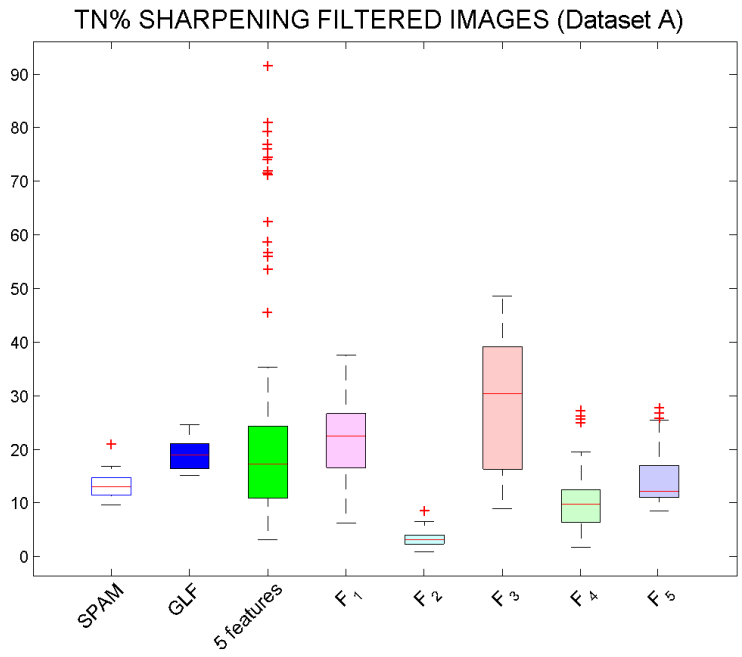


(a) *Dataset A*.

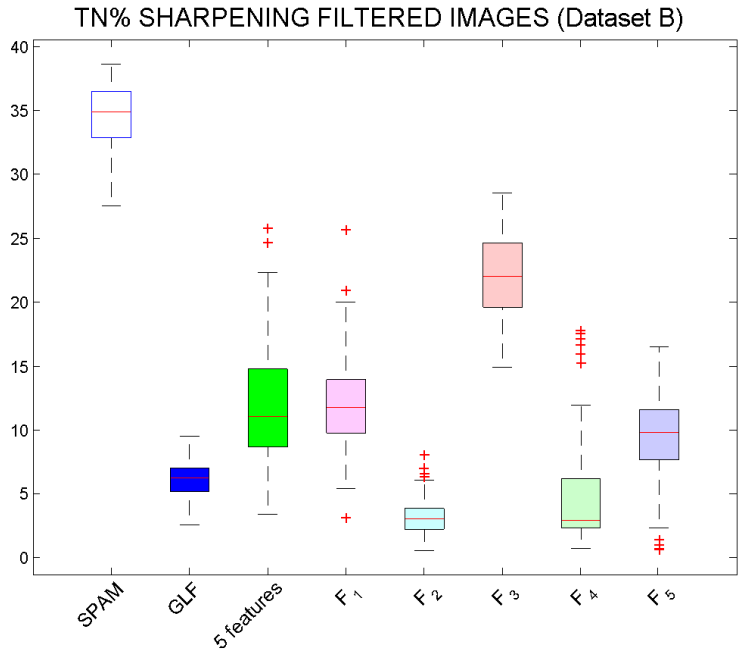


(b) *Dataset B*

Figure 5.20: SPAM features and GLF are both able to split original images from those subject to average filtering. Also our *5 features* classifier has a true negative rate high (around 97%). In this case, in addition to expected features (F_3 and F_4), feature F_1 considered alone is really effective.



(a) *Dataset A*.



(b) *Dataset B*

Figure 5.21: Sharpening filtering is the most difficult editing to reveal. On dataset A the distribution of *5 features* performance has many outliers that can reach value higher than 70%. Nevertheless, we cannot say that any classifiers is effective in splitting sharpening filtered images from originals.

JPEG-compression is identified very well with *5 features* and, as predictable, with the feature related to entropy. To this discussion we dedicate the last section of this work 5.8.1.

We find that the classifier using *5 features* if evaluated on classification of all manipulations together is better performing than the classifier built with SPAM features (results reported on table (5.2)). SPAM features are more suited to recognize noisy and sharpening images. Particularly effective is in the median filter detection: in fact the classifier using SPAM features recognizes images subjected to median filtering 100% of the times. Nevertheless it is completely unable to split JPEG-compressed images from originals.

Performance obtained by GLF features is the worst compared to the proposed method and the SPAM classifier. Although GLF features are designed with the purpose of recognizing median filtering, we find that this manipulation is not split from the original images with the precision that we expected. We assume that this phenomenon is due to the one-class approach used: in fact, the information supplied to a multi-class classifier by the part of the training set composed of negative samples is here lost in one-class approach.

The first single feature is already able to recognize the selected modifications even better than the results in literature. This is of course a result in itself and we will dedicate the next and last session of the thesis to comment the results about JPEG compression detection 5.8.1. Categorization performance better than state of art technique is not always achievable with a single feature approach. In those cases our fusion approach is required to still achieve the best results. We can explain this behavior observing that our features developed for recognizing median filtering (F_3,F_5) are additionally useful in separating noisy and average filtered images from originals. This shows how the proposed features could be useful for treating more problems than the planned ones and how their information can be combined to improve the recognition.

We believe it is worth pointing out that both SPAM and GLF feature dimensionalities are orders of magnitude greater than that of our restricted feature set, respectively 162 and 56 against 5. In our opinion this is supporting our strategies either in feature selection or in fusion decision. The method proposed in the thesis is based on 5 scalar features only, but already delivers better results in most of the cases under investigation.

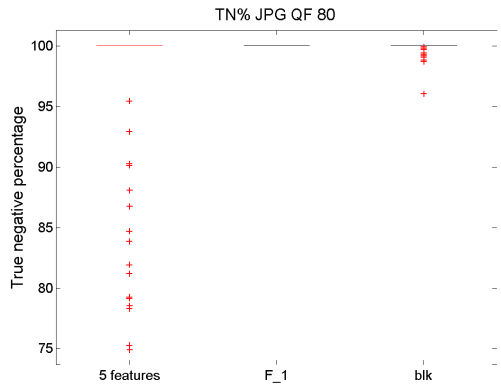
5.8.1 JPEG detection

Here the results of our model for the problem restricted to JPEG-compression identification are presented. The ability of the feature related to entropy to solve this

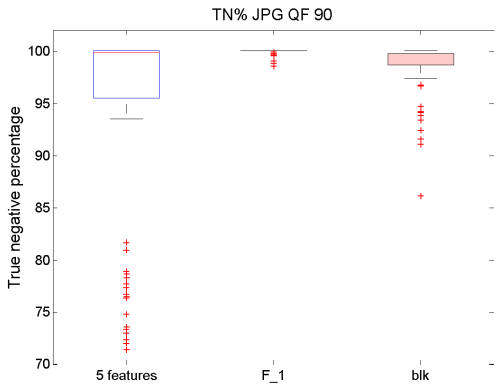
classification problem is verified and compared with the strategy described in [5]. The obtained results are found superior than the referenced technique.

We evaluate 3 OC-SVM that have as goal to characterize the volume of the training set in order to split the original class from the JPEG class. The first is the *5 feature* classifier as proposed in this work, the second an OC-SVM based on the first entropy feature only (F_1), while the last classifier uses as feature the output of algorithm described in [5] (*blk*).

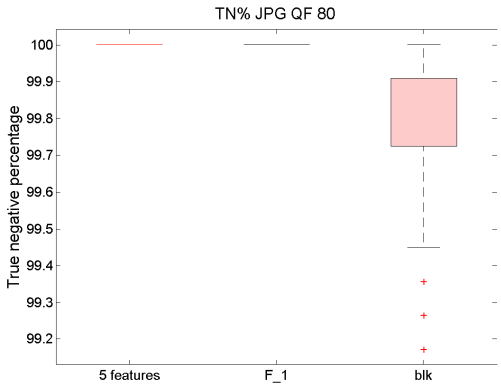
Results are shown in figure (5.22, 5.23, 5.24) for high quality factors only. All classifiers are capable to perfectly distinguishing original images from the JPEG-compressed ones with quality factors lower than 80.



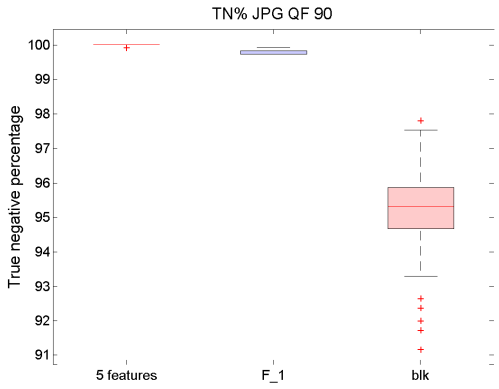
(a) *JPEG QF 80 (Dataset A).*



(b) *JPEG QF 90 (Dataset A)*

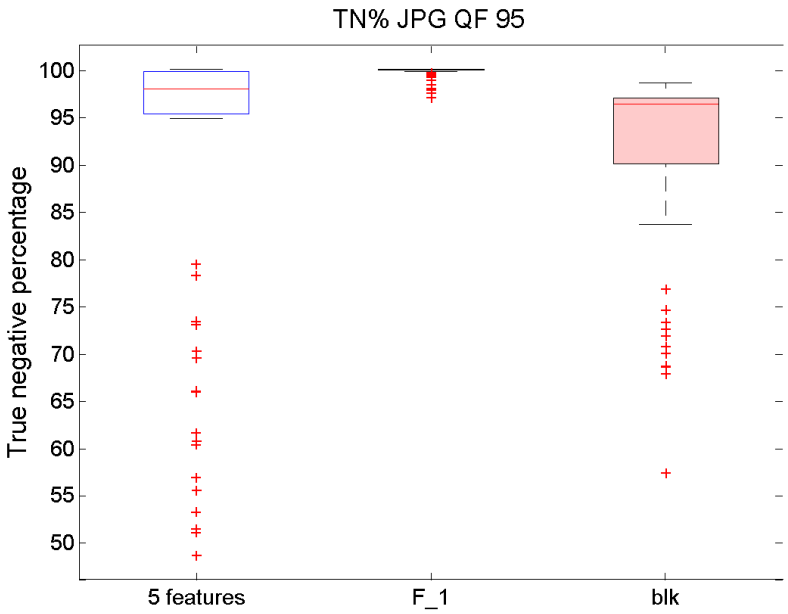


(c) *JPEG QF 80 (Dataset B).*

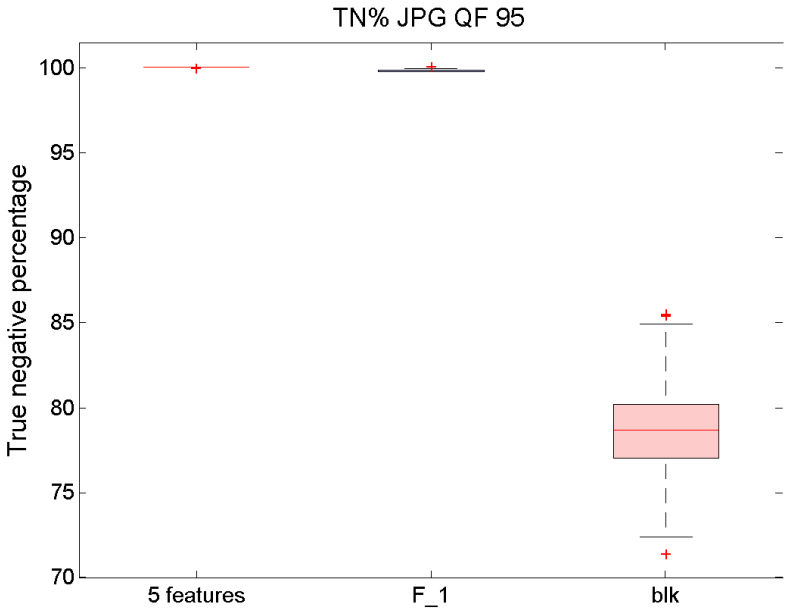


(d) *JPEG QF 90 (Dataset B)*

Figure 5.22: TN percentage for compressions with quality factor of 80, on the left (a)(c), and 90, on the right (b)(d). Although distributions of classification results on dataset A present many outliers, especially for *5 features*, average performance of all feature sets are higher than 97%.

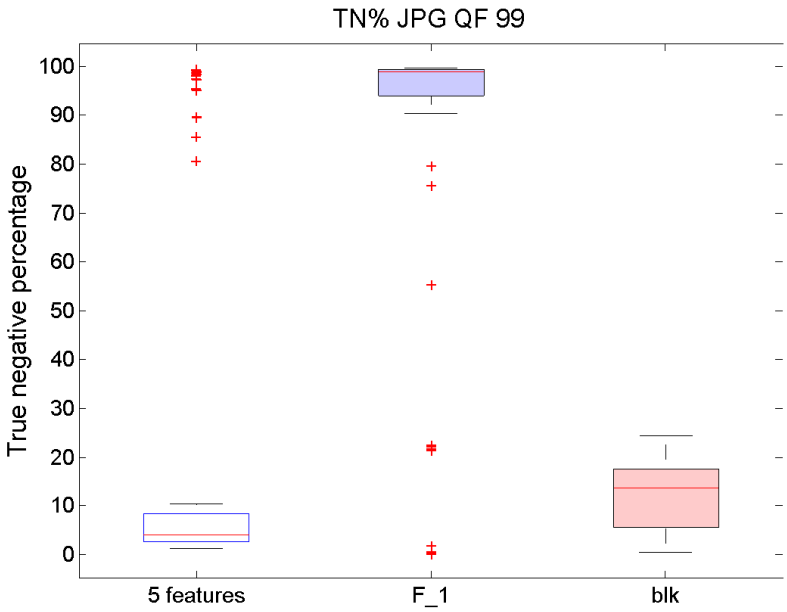


(a) *Dataset A*.

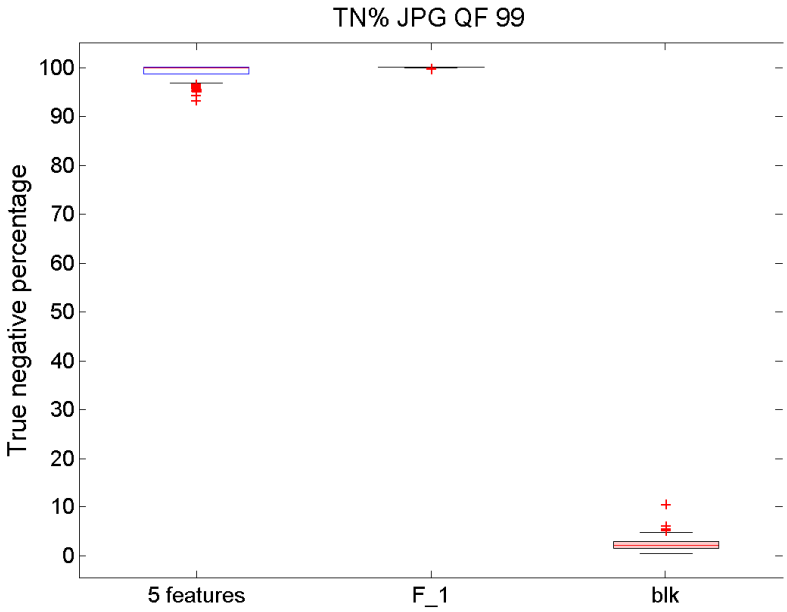


(b) *Dataset B*

Figure 5.23: On dataset A the true negative percentage is (in average) higher than 95% for all feature sets. In dataset B a different behaviour is shown by *blk* and our features. In fact, meanwhile the first has a TNR inferior than 80%, for other feature sets that value reaches almost 100%.



(a) *Dataset A*.



(b) *Dataset B*

Figure 5.24: The true negative rate for compression with quality factor of 99 is higher than 95% if evaluated on dataset A with F_1 only. If we consider all *5 features* or *blk*, this percentage decrease until less than 20%. In dataset B, both F_1 and *5 features* are capable of recognizing the compression nearly 100% of the time, contrary to *blk*, whose TNR is lower than 10%

In dataset B *5 feature* the classifier is even capable of recognizing images compressed with a quality factor of 99 with a true negative rate higher than 0.95. On dataset A *5 features* classifier has a really low true negative rate if tested on images compressed with a very high quality factor (more than 95), but, when we consider the first feature only, the performance improves. That is probably because of the influence of other features that do not recognize the difference between considered JPEG-compressed images and original ones. Instead, although the measure *blk* is revealed really effective for quality factors lower than 95, in the other cases the results are not comparable with ours.

Chapter 6

Conclusion and future work

In this work we tackled an image forensic classification problem focused on the identification of "original" images, i.e. never compressed and never processed images. In particular we investigated the difference from images subjected to five possible modifications: the lossy compression operation JPEG, the addition of Gaussian random noise and three filtering processes: average, median and sharpening filters.

With the aim of developing a technique that is as general as possible, we decided to follow an approach based on feature level decision fusion. Usually the drawbacks of this procedure are due to a large amount of features and the difficulty in training set construction. These problems were both solved, the former with the extraction of five scalar operating features and the latter with the implementation of a particular classifier typology trained by one class only. The method allows to recognize never-compressed and never-processed images as original with an average accuracy higher than 84% over the selected modifications. However we should point out the large fluctuation in detecting the different considered processes. This 84% result is in fact largely biased by the discrimination capabilities of sharpening filter where more than 70% of modified images are systematically misclassified.

Testing only one modification at a time we achieve the following accuracies: JPEG-compression 98%, Gaussian random noise 80%, median filter 95%,average filter 95% and sharpening filter 54%.

We found no equivalent techniques in the literature for modifications identification approaching a similar type of procedure. Therefore we compared our classifiers with state of the art features described in [5], [18] and [4]. In particular, we prepared analogous classifiers to the one proposed in this thesis, but based on these different features.

The analysis of the comparison results shows that the average accuracy over the selected manipulations is higher than that evaluated by SPAM features and GLF features. While the performance of the latter are really low, unexpectedly, even

on median filter detection, SPAM features are very effective in the recognition of images subjected to Gaussian random noise and especially to median filter process.

Not only we compared classifiers using the five features together, but also we measured the ability of the single features. In particular we focused on performance of the first feature. In fact, in addition to be useful in recognition of almost all manipulations, the improvement given by the study of entropy in JPEG-compression detection is undeniable. While the measure evaluated in [5] is absolutely not able to distinguish an image compressed with a quality factor of 99, our feature can recognize this modification with an accuracy of 97%.

At least the validity of our approach by comparison on another dataset [33] has been proved. Once all techniques were prepared we repeated the analysis starting from a second dataset. None of the images belonging to this new set was considered during the development phase. Outcoming results well reproduce the behviour obtained on the first dataset, thus proving the good generality of the proposed approach. The only exception was for the identification of noisy images. We believe that this phenomenon is due in large extent to the poor image quality in the dataset [33]. In fact recognizing an image where noise was added to a low quality original image is hard.

We found particularly promising a deeper study of the feature related to entropy. In fact, images compressed with different quality factors present a variation of entropy in different frequencies. An accurate analysis of this phenomenon can contribute in quality factor estimation. Moreover we are confident that a double compression, both aligned and nonaligned, leaves some traces in DCT coefficient distribution and that these traces can be revealed with the help of entropy.

In order to recognize sharpening filters we suggest giving more attention to the tail of Fourier transformation angular average used in computation of the fourth feature.

In this work only one editing at a time was applied on testing images. All features were studied with the hypothesis that combinations of modifications were not allowed. A possible development of this analysis can be extended to a scenario where more manipulations (or combination of manipulations) are considered.

Three possible models of One-Class classifier were tested. The obtained results biased us to prefer the use of OC-SVM. In fact, we believed from one side that a reduction of dimensionality is not necessary and the sphere classifier approach would eventually become too simple for further improvement of the technique.

A different employment for the features described in this work could be in blind image quality assessment [35]. The problem tackled in this field is to find a way to automatically evaluate and control the perceptual quality of the visual content

as a function of multiple parameters. Objective image quality assessment refers to predicting the quality of distorted images as would be perceived by an average human being. The idea of the spherical classifier is, as it has been for this work, a valid starting point that could be extended to this new or other application. The feature space manipulation adopted in the spherical classifier can be a good playground for image quality assessment: in fact, as measure of quality the distance (that should to be defined) is conceivable between origin and image in a new feature space.

Appendices

Appendix A

Entropy

In 1948 C.E. Shannon [36] introduced concept of entropy in information theory field. He chose this name in accord to similarity with thermodynamic entropy in statistical mechanics, a quantity that describes the minimum number of yes-no questions needed to be answered in order to fully specify the microstate, given that macrostate is known.

Shannon entropy citeentr of X, where X is a random variable, is a value that describes the quantity of information that is gained, on average, when this value is learn and is, also, the measure of the amount of uncertainty about X before its value is learnt. So entropy can be view either as a measure of uncertainty *before* the random variable is learn, or as a measure of how much information is gained *after* learning the value of X. Labels attached to possible value of a random variable do not influence information content. For this reason, entropy of a random variable is defined to be *"a function of the probabilities of the different possible values the random variable takes, and is not influenced by the labels used for those values"* [37]. Entropy is often written as a function of probability distribution, $p_1, ..., p_n$. The *Shannon entropy* associated with this probability distribution is defined by

$$H(X) \equiv H(p_1, ..., p_n) \equiv -\sum_x p_x log_2 p_x \tag{A.1}$$

Obviously, when p_x is equal to 0, $log_2 0$ is undefined: an event which can never occur should not contribute to entropy, so by convention $0 log_2 0 \equiv 0$. Furthermore, mathematically speaking $\lim_{x \to 0} x log x = 0$.

Entropy is also used to quantify the resources needed to store information.

It is possible to assume that a digital signal can be represent as a sequence of independent and identically distributed random variables $X_1, X_2, ...$ representing the bit sequence characterizing the digital signal.

For instance, suppose that an information source is producing bits $X_1, X_2, ...,$

each bit will be equal to 0 with probability p and equal to 1 with probability $1-p$. Each sequences of values $x_1, x_2, ...$ of the signal can be classify as *typical sequence* or *atypical sequence*: the first are sequences which are highly likely to occur, the second are sequences that occur rarely.

The bigger is the signal higher is the expectation that a fraction of the values 0 over the total number of random variables will be close to p complementary to the fraction of 1 which will be close to $1-p$. In this case the sequence is called *typical sequence*.

In this case the probability that random variables take those particular values for the typical sequences is

$$p(x_1, ..., x_n) = p(x_1)p(x_2)...p(x_n) \approx p^{np}(1-p)^{(1-p)n} \tag{A.2}$$

taking logarithms on both sides

$$-log_2 p(x_1, ..., x_n) \approx -nplog_2 p - n(1-p)log_2(1-p) = nH(X) \tag{A.3}$$

Where $H(X)$ is the entropy of the source distribution. There can be at most $2^{nH(X)}$ typical sequences, since the total probability of all typical sequences cannot be greater than one.

Because of Low of Large Numbers, the bigger is the number of signal's component the higher is the probability that the sequence is a typical sequence: since there are at most $2^{nH(X)}$ typical sequence it only requires $nH(X)$ bits to uniquely identify a particular typical sequence.

Studying the curve of entropy in case of a binomial random variable it is possible to understand easily that the maximum value for the entropy is in correspondence of equal probability for each possible value. This kind of configuration turns outresults when the uncertain is maximum.

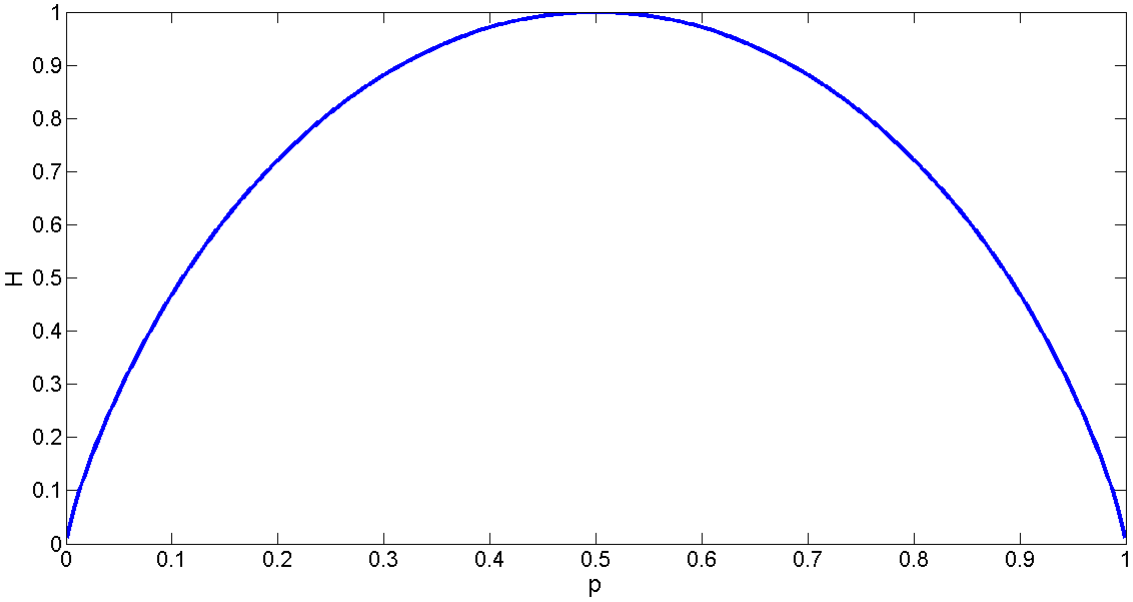


Figure A.1: Entropy as function of probability of one variable in binomial case

Appendix B

Linear SVM

Each image is represented by a vector of features and can be identified as a point in the feature space. When the aim of analysis is to split dataset into two different classes, two cases can verify: training data can be linearly divided in feature space or not.

In the first case SVM works figuring out the best hyperplane that can split the training set into the right classes. In the second case an hyperplane capable of separating the two kinds of samples does not exist in feature space. Thus resorting to compromises is necessary.

Here we will describe the working principle of the SVM, supposing that a plane could be used to divide the set under investigation into two separate classes. As all classifiers, SVMs are functions that have as input a sample and as output a value corresponding to a class. In SVM the decision function is defined as

$$f(\mathbf{x}_i) = sign(\mathbf{w}^T \cdot \mathbf{x}_i + b) \tag{B.1}$$

Where $\mathbf{x}_i$ is the point in feature space representing i-th sample and $\mathbf{w}$ and b are the parameters characterizing the SVM.

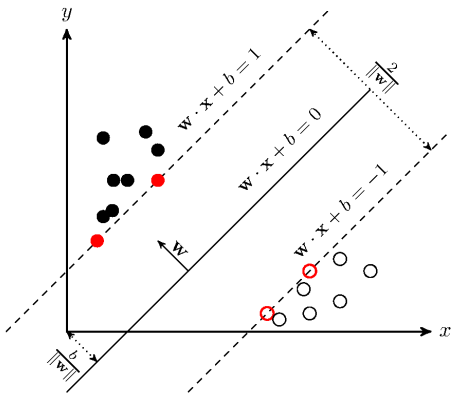
Let $\mathbf{x}_i$ be the i-th sample of the training set and N its cardinality. Let $y_i \in \{1, -1\}$ be the label corresponding to the real class of i-th sample. Error function is defined as

$$R(\mathbf{x}_i) = \sum_{i=1}^{N} (f(\mathbf{x}_i) - y_i) \tag{B.2}$$

The main idea is to minimize the error function changing the values of $\mathbf{w}$ and b. When a vector $\mathbf{x}_i$ is correctly identified its contribution to the error is null and -2 or +2 otherwise.

B.1 Separable case

In the linear separable case, training samples belonging to different classes can be separated by a hyperplane. In order to determine this division, SVM looks for the region between the two classes, called *margin*, and tries to maximize it.



(a)

Figure B.1: Bidimensional example of separable linear SVM. The red samples are the Support Vector.

The separator hyperplane is placed in the middle of the margin, so it is equidistant from both classes. Samples that lie on the bound of margin are called *support vectors* and their position is the only information used by SVM.

In fact, one of the strength of SVM is the possibility to repeat the structure starting from SV only. All other training samples are not necessary to define the separator hyperplane. As a matter of facts their contribution to the error function is null and invariant, as they are already correctly categorized. Therefore they are not required in the evaluation of the separator plane.

Let $\mathbf{x}_i \in \mathbb{R}^n, i = 1, ..., l$ be the l vectors of features of the training set and $y_i \in \{1, -1\}$ the class of $i - th$ vector.

The equation of the hyperplane that divides points having $y_i = 1$ from those having $y_i = -1$ can be written as

$$\mathbf{w}^T \cdot \mathbf{x} - b = 0 \tag{B.3}$$

In order to mathematically set the parameters that define margins, the equations that training samples have to respect are determined.

$$\begin{cases} \mathbf{w}^T \cdot \mathbf{x} - b \geq 1 & \text{for } \mathbf{x}_i \text{ of the first class} \\ \mathbf{w}^T \cdot \mathbf{x} - b \leq -1 & \text{for } \mathbf{x}_i \text{ of the second class} \end{cases} \tag{B.4}$$

Parameters investigated are found solving a linear optimization problem

$$\begin{aligned} &\underset{(\mathbf{w},b)}{\text{argmin}} && ||\mathbf{w}||_2 \\ &\text{subject to} && y_i(\mathbf{w}^T \cdot \mathbf{x}_i - b) \geq 1 \end{aligned} \tag{B.5}$$

where the constrain of problem (B.5) is a reformulation of equation (B.4).

The problem (B.5) is not easily solvable because of the presence of square root in norm computation. For this reason usually a quadratic programming optimization problem with the same minimum as solution (B.5) is addressed.

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{argmin}} && \frac{1}{2}||\mathbf{w}||^2 \\ &\text{subject to} && y_i(\mathbf{w}^T \cdot \mathbf{x}_i - b) \geq 1 \end{aligned} \tag{B.6}$$

This problem is easily solvable with the method of Lagrange multipliers. With the introduction of Lagrange multipliers α the objective function of the above optimization problem can be expressed in a new way as:

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) + \sum_{i=1}^{l} \alpha_i \tag{B.7}$$

The primal expression of the problem requires that L_P is minimized with respect to $\mathbf{w}$ and b. In this way the derivative of L_P with respect to all α_i vanishes, subject to the constraints $\alpha_i \geq 0$.

Thanks to its convex form, the solution for the dual problem is the same as that of the primal one. For this reason solving the dual formulation offers the desired solution.

Thus, L_P is maximized with the constraints imposed by the problem. The gradient of Lagrangian function with respect to $\mathbf{w}$ and b is posed equal to 0.

$$\begin{aligned} &\min && L_P \\ &\text{subject to} && \frac{\partial L_P}{\partial \alpha_i} = 0 \\ & && \alpha_i \geq 0 \end{aligned} \tag{B.8}$$

$$\begin{aligned} \max \quad & L_P \\ \text{subject to} \quad & \frac{\partial L_P}{\partial \mathbf{w}} = 0 \\ & \frac{\partial L_P}{\partial b} = 0 \\ & \alpha_i \geq 0 \end{aligned} \tag{B.9}$$

If the gradient of L_P with respect to $\mathbf{w}$ and b vanishes, conditions on $\mathbf{w}$ and $\sum \alpha_i y_i$ are obtained.

$$\begin{aligned} \mathbf{w} &= \sum \alpha_i y_i x_i \\ \sum \alpha_i y_i &= 0 \end{aligned} \tag{B.10}$$

Substituting this condition in Eq (B.7) a new formulation for Lagrangian equation is obtained. This new equation is the objective function that will be maximized in dual problem, while L_P is the objective function of the primal problem.

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{B.11}$$

B.2 Non separable case

Not all datasets are linearly separable, so sometimes the problem described above does not have a feasible solution, that is, a plane that completely divides the two datasets does not exist. In order to extend this method to non-separable data, some constraints must be relaxed. These relaxations are done in order to allow the presence of a few mislabeled training samples. If there exists no hyperplane that can split the two classes the SVM chooses the hyperplane that divides the samples as cleanly as possible.

With this aim, positive slack variables ξ_i are introduced. These variables measure the degree of misclassification of $i-th$ sample and change the constrains of the problem (B.6).

$$\begin{aligned} & \mathbf{w}^T \cdot \mathbf{x}_i + b \geq +1 - \xi_i \quad \text{for } y_i = +1 \\ & \mathbf{w}^T \cdot \mathbf{x}_i + b \leq -1 + \xi_i \quad \text{for } y_i = -1 \\ & \xi_i \geq 0 \qquad\qquad\qquad\qquad \forall i \end{aligned} \tag{B.12}$$

The objective function is also changed. To the objective function in problob(B.6) an addition term is added in order to penalize non-zero ξ_i:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_i \xi_i) \tag{B.13}$$

In (B.13) C is a parameter that controls the maximum error on the training set that division can do. The larger C is, the higher are the penalties associated to errors. Thus, the solution of (B.13) subject to (B.12) becomes a trade-off between a large margin and a small error penalty.

One big advantage is that in dual problem formulation slack variables do not appear. The new formulation of the problem is only dependent on the parameter C, decided by the user.

$$\begin{aligned} \text{max} \qquad & L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ \text{subject to} \qquad & 0 \geq \alpha_i \geq C \\ & \sum_i \alpha_i y_i = 0 \end{aligned} \tag{B.14}$$

Bibliography

[1] P. D. Pandit and M. Rajput, "Survay on anti-forensics operations in image forensics," *International Journal of Computer Science and Information Technologies*, vol. 5.

[2] A. Piva, "An overview on image forensics," *Hindawi Publishing Corporation ISRN Signal Processing*, vol. 3013.

[3] M. Stamm, S. Tjoa, W. S. Lin, and K. J. R. Liu, "Anti-forensics of jpeg compression," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process*, 2010.

[4] C. Chen, J. Ni, and J. Huang, "Blind detection of median filtering in digital images: A difference domain based approach," vol. 22, December 2013.

[5] Z. Fan and R. L. D. Queiroz, "Identification of bitmap compression history: Jpeg detection and quantizer estimation," *IEEE Trans. Image Process.*, vol. 12, no. 2, 2003.

[6] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu, "Robust median filtering forensics using an autoregressive model," vol. 8, no. 9, september, 2013.

[7] D. Zoran and Y. Waiss, "Scale invariance and noise in natural images," *IEEE 12th International Conference on Computer Vision*, 2009.

[8] G. Cao, Y. Zhao, R. Ni, L. Yu, and H. Tian, "Forensic detection of median filtering in digital images," *IEEE International Conference on Multimedia and Expo (ICME)*, 2010.

[9] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," *Proc. SPIE 7541, Media Forensics and Security II*, 2010.

[10] G. Valenzise, V. Nobile, M. Tagliasacchi, and S. Tubaro, "Countering jpeg anti-forensics," *Proc. IEEE Int. Conf. Image Process.*, 2011.

[11] S. Lai and R. Bohme, "Countering counter-forensics: The case of jpeg compression," *13th International Conference, IH 2011, Prague, Czech Republic, May 18-20, 2011, Revised Selected Papers*.

[12] W. Luo, J. Huang, and G. Qiu, "Jpeg error analysis and its applications to digital image forensics," *IEEE Transaction on Information Forensics and Security*, vol. 5, no. 3, September 2010.

[13]

[14] L. M. Manevitz and M. Yousef, "One-class svms for document classification," *Journal of Machine Learning Research 2*, 2001.

[15] M. C. Stamm and K. J. R. Liu, "Forensic detection of image tampering using intrinsic statistical fingerprints in histograms," *IEEE Transactions on Information Forensics and Security*.

[16] L. Gaborini, "Image tampering detection and localization," 2013-2014.

[17] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "A variational approach to jpeg anti-forensics," *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference*.

[18] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *Information Forensics and Security, IEEE Transactions*, 2010.

[19] S. Lyu and H. Farid, "Steganalysis using higher-order image statistics," *IEEE Transactions on Information Forensics and Security*, 2010.

[20] T. Pevny and J. Fridrich, "Novelty detection in blind steganalysis," *Proceedings of the 10th ACM workshop on Multimedia and security*.

[21] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Revealing the traces of jpeg compression anti-forensics," *IEEE Transactions on information forensics and security*, vol. 8, no. 2, February 2013.

[22] M. Fontani, T. Bianchi, A. D. Rosa, A. Piva, and M. Barni, "A framework for decision fusion in image forensics based on dempster-shafer theory of evidence," *Journal of Latex Class Files*, vol. 6, no. 1, January, 2007.

[23] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image and Vision Computing*, vol. 27, 2009.

[24] H. Farid, "Exposing digital forgeries from jpeg ghosts," *IEEE Transactions on Information Forensics and Security*, 2009.

[25] H. Farid, "Digital image ballistics from jpeg quantization," *Dept. Comput. Sci. Dartmouth College*, 2006.

[26] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," *Proc. SPIE 7541, Media Forensics and Security II*.

[27] E. Y. Lam and J. W. Goodman, "A mathematical analysis of dct coefficient distributions for images," *IEEE Transactions on image processing*, vol. 9, no. 10, October, 2000.

[28] H.-C. Kim, S. Pang, H.-M. Je, D. Kim, and S.-Y. Bang, *Support Vector Machine Ensemble with Bagging*. 2002.

[29] M. A. Aizerman, E. M. Braverman, and L. I. Rozoner, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25.

[30] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of jpeg artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 1003–1017, June 2012.

[31] D. Vazquez-Padin and F. Perez-Gonzalez, "Prefilter design for forensic resampling estimation," *IEEE Interational Workshop on Information Forensics and Security (WIFS)*, Nov, 2011.

[32] D. Vazquez-Padin and P. Comesana, "Ml estimator of the resampling factor," *IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 205–210, Dec. 2012.

[33] G. Schaefer and M. Stich, "Ucid - an uncompressed colour image database," *Storage and Retrieval Methods and Applications for Multimedia 2004*.

[34] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[35] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, March 2013.

[36] C.E.Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, july, October 1948.

[37] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*.

Ringrazio il mio relatore, il professor Stefano Tubaro, per la disponibilità dimostrata e i miei supervisors, maître de conférences Grenoble-INP François Cayre e chargé de recherche Kai Wang, per aver seguito il mio lavoro durante lo stage a Grenoble e il GipsaLab per averlo reso possibile, sia burocraticamente che finanziariamente.

Ringrazio la mia famiglia (animali compresi, ovviamente) per la pazienza che ha avuto in questi anni e il supporto dato.

Un enorme grazie a Giovanni che, nonostante 580 km di distanza, mi è sempre vicino.

Ringrazio tutte le persone, fisici e ingegneri, che mi hanno accompagnato in questo percorso universitario, in particolare Manu, Ele, Silvia, Kens, Jack, Nick, Andre, Pepe, Boes e Ruggero. Ringrazio allo stesso modo gli INNOI amici di sempre su cui si può sempre contare.

Ringrazio Francesca, Josselin e Artù, la famiglia Clavier, Cecilia e Cinzia per avermi fatto sentire a casa.

E infine un ringraziamento di riguardo a tutti coloro che ho incontrato nell'ultimo anno e che hanno fatto sì che il mio soggiorno in Francia sia stato bellissimo: Ayan, Mahdi e Andrea le prime persone che ho conosciuto, Giusi, Elena e tutti gli altri medici italiani e rumeni a Grenoble, Kristina e i suoi amici, Viktorija e Ivan e gli altri ragazzi della residenza, Gulay, Katia, Emmanuelle, Hélène, Jeremy, Romain, Antoine, Luis e Michael, Alessandro e Matteo.