UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica

Illuminant map analysis for image splicing detection

Candidato

Lorenzo Cioni

Relatori

Prof. Alessandro Piva

Prof. Carlo Colombo

Correlatori

Dott. Massimo Iuliani

Dott. Marco Fanfani

Anno Accademico 2015/2016

Abstract

With the advent of powerful image editing tools, manipulating images and changing their content is becoming a trivial task. Unfortunately, most of the times, these modifications are aimed at deceiving viewers, changing their opinions or even affecting how people perceive reality. Therefore, the development of efficient and effective forgery detection techniques has become a hot research topic.

From all types of image forgeries, image compositions are specially interesting. This type of forgery uses parts of two or more images to construct a *new reality* from scenes that never happened. Among all different telltales investigated for detecting image compositions, illumination inconsistencies are considered the most promising ones since a perfect light matching in a forged image is still difficult to achieve.

This thesis builds upon the hypothesis that image illumination inconsistencies are strong and powerful evidence of image composition and presents an extension of two approaches in literature, both based on the illuminant maps analysis. The first method is specific for human face forgeries, the second is a blind approach for detecting regional splicing.

Sommario

Con l'avvento di sempre più potenti strumenti di modifica di immagini, la manipolazione di contenuti visuali è diventata un'operazione molto comune. Purtroppo, il più delle volte, queste modifiche sono volte ad ingannare gli spettatori, cambiare le loro opinioni o perfino a modificare il modo in cui percepiscono la realtà. Di conseguenza, lo sviluppo di tecniche sempre più efficienti ed efficaci per la rilevazione di fotomontaggi è diventato un argomento di ricerca molto importante.

Tra tutti i tipi di falsificazione, la composizione di più immagini è particolarmente interessante. Questo tipo di operazione utilizza le parti di due o più immagini al fine di costruire una vera e propria nuova realtà, ricostruendo scene mai accadute realmente. Tra tutte le diverse tecniche di rilevazione di composizione di immagini, quelle basate sulle inconsistenze della luce sono considerate fra le più promettenti, poiché è quasi sempre difficile riuscire a rispettare la coerenza di illuminazione all'interno della stessa immagine quando questa viene modificata.

Questa tesi pone le sue basi proprio sull'ipotesi che l'individuazione di inconsistenze nella luce presenti in un'immagine sia una prova evidente di contraffazione. A partire da questa ipotesi, è presentata un'analisi ed un'estensione di due approcci presenti in letteratura che pongono le loro basi sulle *Illuminant Maps*. Il primo metodo è specifico per la rilevazione di contraffazioni che riguardano volti umani nelle immagini; il secondo invece è un approccio molto più generico che cerca di individuare zone dell'immagine che presentano delle inconsistenze nel colore dell'illuminante.

Ad Agnese e alla mia famiglia

Contents

Introduction

With the rapid development of multimedia and network technology, digital images, as an effective carrier of information, are having a more and more important impact on people's daily lives.

However, the image content can easily be tampered with the increasingly powerful image processing software, which threatens the integrity and authenticity of digital images. Moreover, if the tampered images were used for illegal purposes, they may entail negative consequences both for the individual and for the society.

Images manipulated with the purpose of manipulating and deceiving user opinions are present in almost all communication channels including newspapers, magazines, TV shows and, obviously, on Internet. [39]

Image splicing is a fundamental operation used in digital image tampering. It is a copy-and-paste operation of image regions from one image onto the same or another image without performing post-processing such as smoothing [29]. Even without post-processing, the artifacts introduced by image splicing may be almost imperceptible. Therefore, the detection of image splicing is a preliminary but desirable study for image forensics.

Investigating image's lighting is one of the most common approaches for splicing detection. This kind of approach is particularly robust since it's really hard to preserve the consistency of the lighting environment while creating an image composite (i.e. a splicing forgery).

In this scenario, there are mainly two approaches:

1. based on the object-light geometric arrangement

2. based on the image illuminant colors

We focused our attention on the illuminant-based approach, which assumes that a scene is lit by the same light source. More light sources are admitted but only far enough such as to produce a constant brightness across the image. In this condition, pristine images will show a coherent illuminant representation; on the other hand, inconsistencies among illuminant maps will be exploited for splicing detection.

Illuminant maps locally describe the lighting in a small region of the image. In the computer vision literature many different approaches exist for determining the illuminant of an image. In particular, such techniques are divided into two main groups: statistical-based and physics-based approaches.

Regarding the first group, we start investigating on the *Grey-World algorithm* [3], which is based on the Grey-World assumption, i.e. the average reflectance in a scene is achromatic. In [13], this algorithm proved to be special instances of the Minkowski-norm. Van de Weijer *et al.* [46] than proposed an extension of the Gray-World assumption, called *Gray-Edge hypothesis* [46], which assumes that the average of the reflectance differences in a scene is achromatic.

These differences can be determined by taking derivatives of the image. Therefore, the authors present a framework with which many different algorithms can be constructed. We focus our attention on this framework, called *Generalized Grey-World algorithm* (GGE).

For the latter group, it was investigated the method proposed by Riess *et al.* [37], which extends the *Inverse Intense Chromaticity* (*IIC*) space approach proposed by Tan *et al.* [43] and tries to model the illuminants considering the dichromatic reflection model [44]. In this case, the illuminant map is evaluated dividing the images into blocks, named *superpixels*, of approximately the same object color, then the illuminant color is evaluated for each block solving the

lighting models locally.

Carvalho *et al.* [4] then presented a method that relies on a combination of the two approaches for the detection of manipulations on images containing human faces. In addition to maps, a large set of shape and texture descriptors are used together. Note that, from a theoretical viewpoint, it is advantageous to consider only image regions that consist of approximately the same underlying material: for this reason, in [4] the authors focused their analysis on human faces.

Finally, in [9] Fan *et al.* used inconsistencies of the illuminant color in the object region in order to detect the region splicing forgeries based on local illumination estimation. They proposed to combine five low-level statistics-based algorithms to estimate illuminant of each horizontal and each vertical band.

This thesis propose two different approaches, divided into two modules, starting from the works presented in [4] and [9]. We analyzed illuminant maps entailing the interaction between the light source and the objects contained in a scene in order to detect image forgeries. The first module is aimed at detecting forgeries involving people; the second approach is based on considering a set of different low-level illuminant color estimators and a simple image segmentation for detecting forgeries without any *a priori* knowledge about the image content.

This work is structured as follows. In Chapter 1, image composition is defined and the importance of devising and developing methods able to detect this kind of forgery is discussed. Some recent methods in the literature that consider illuminant information for detection of image splicing are also presented. Chapter 2 describes the main methods grounded on illumination inconsistencies for detecting image composition and the background theories on which this work is based on. Chapter 3 collects the experimental results, putting our research in

perspective and discussing new research opportunities.

Chapter 1

Related work

1.1 Image forgery

When dealing with a digital image, it is quite common to wonder if it is original or it has been counterfeited in some way. Images and videos have become the main information carriers in the digital era and they are used to store real world events, but they are very easy to manipulate because of the availability of powerful editing software and sophisticated digital cameras.

The contexts where doctored pictures could be involved are very disparate; they could be used in a tabloid or in an advertising poster or included in a journalistic report but also in a court of law where digital (sometimes printed) images are presented as crucial evidences for a trial in order to influence the final judgment.

Generally speaking, the objective of the image editing operation could be, for example, to improve its quality or to change its semantic content [34]. In the first case, the processed image will carry the same information as the original one, but in a more usable/pleasant way. Hence, we refer to this kind of editing as *innocent*. In the latter case, the semantic information conveyed by the image is altered, usually by adding or hiding some information. This kind of editing is considered *malicious* [34].

Forgeries can be classified into three categories:

- *Copy-move forgery*
- *Image retouching*
- *Image splicing*

Copy-move forgery is one of the most popular forms of tampering in which some regions are copied from a particular location in an image and thereafter pasted at one or more locations within the same image or a different image of preferably the same scene. This technique is useful when the forger wants either to hide or duplicate something that is already present in the original image.

Image retouching is an image editing method that pertains to a slight change in the image for various aesthetic and commercial purposes, not necessarily conforming to the standards of morality. The retouching is mostly used to enhance or reduce the image features. Usually this type of forgery is realised by changing the colour or texture of the objects, intensify the weather conditions or simply introducing some blur for defusing the objects.

Image splicing, also known as *cut-and-past attack*, is another common form of photographic manipulation. This kind of forgery involves the composition of more than one image that are combined together to generate a tampered image, usually to alter its content and original meaning [36] [34]. When performed carefully, the border between the spliced regions can be visually imperceptible.

This composition process includes all the necessary operations (such as brightness and contrast adjustment, affine transformations, color changes, etc.) to construct realistic images able to deceive viewer. In this process, normally, we refer to the parts coming from other images as *aliens* and to the image receiving the other parts as *host*. One common studied case is image composition involving very popular people, which can be employed with many different aims.

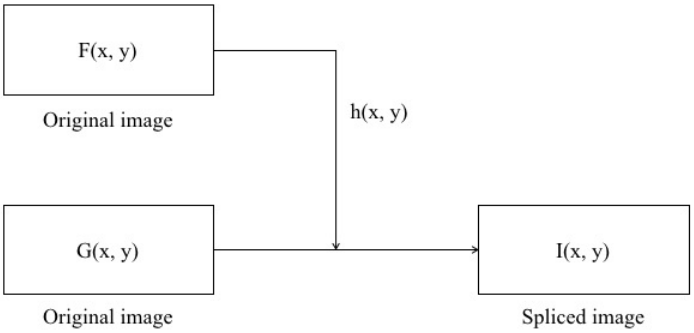

Figure 1.1: Image splicing process

1.1.1 Some famous cases

Photography has lost its innocence since the early days of its birth. Image forgery can be traced back to as early as 1840s when Hippolyte Bayrad created the very first fake image 1.2, in which he was shown committing a suicide, only a few decades after Niépce created the first photo.

Figure 1.2: The first tampered image

O.J. Simpson - June 1994 This altered photo of O.J. Simpson (Fig. 1.3) appeared on the magazine's cover Time Magazine, soon after his arrest for

murder.

Figure 1.3: The Time Magazine cover with O.J. Simpson

In fact, the photograph was altered compared to the original image that appeared on the cover of Newsweek magazine. Time magazine was accused of manipulation of the photography in order to make darker and more menacing the figure of Simpson.

Iraq - April 2003 This composition (Fig. 1.4) of a British soldier in Basra, who keeps pointing toward a civilian Iraqi gesticulating, appeared on the cover of the Los Angeles Times, immediately after the Iraq invasion.

Brian Walski, a staff photographer for the Los Angeles Times was summarily fired from his publisher because he merged two of his shots in order to improve the composition.

George W. Bush - March 2004 This image (Fig. 1.5), taken from the election campaign of George W. Bush, outlined a packed audience of soldiers as a backdrop to a child who was flying the American flag. This image was digitally edited, using a crude copy and paste, removing Bush from the podium.

Cases like these show how present image composition is in our daily lives. Unfortunately, it also decreases our trust on images and highlights the need for developing methods for recovering back such confidence.

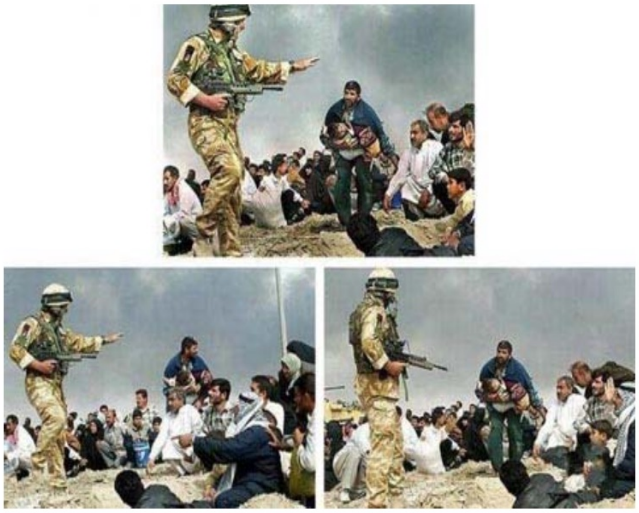

Figure 1.4: An example of image composition

Figure 1.5: An example of image composition

1.2 Image forgery detection techniques

As a consequence of all previous facts, a reliable assessment of image integrity becomes of fundamental importance [52] [10] [34].

Forgery detection intends to verify the authenticity of images. The research community interested in multimedia content security has proposed several approaches aimed at detecting image forgeries that can be classified into *active* and *passive* techniques.

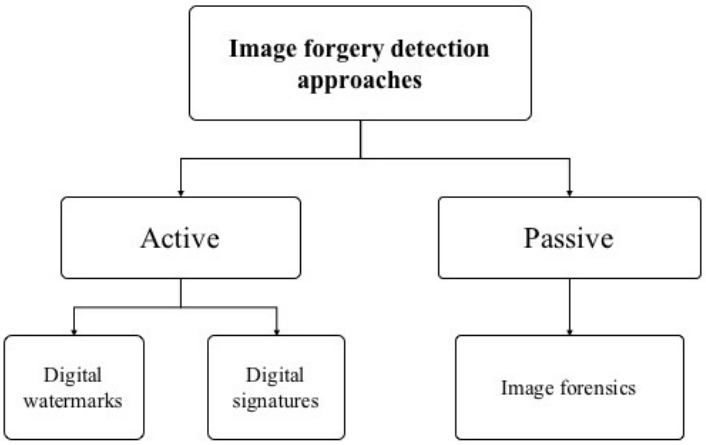

Figure 1.6: Image forgery detection approaches classification

1.2.1 Active approaches

Active approaches basically envolve the data hiding approaches and the digital signature approaches. Active approaches are inspired by the idea of granting authenticity to the images generated by digital cameras. [16] [2].

In this kind of methods, some additional information is embedded into the image at the instant of its acquisition and any later modification can be detected by checking the value of the *digital watermark* [6] [23] [5] or the *digital signature* [38] [35] [30] at the moment of its fruition. Therefore these techniques are also called intrusive techniques [25].

The major drawback of these schemes is that they must be used by the image capturing device itself at the time of recording or later by an authorized person, using a specialized embedding software. Therefore, specially equipped cameras or special hardware and software are needed. Furthermore, this kind of approaches may degrade the quality of the original image [47].

From the implementation point of view, active approaches need specially equipped camera consisting of a watermarking or a digital signature chip and

some private key hard-wired in the camera itself, using which every image captured by the image is authenticated before saving it on its memory card.

1.2.2 Passive approaches

Passive approaches [28] were proposed to overcome the problems encountered in the active approaches. These schemes do not require any image preprocessing and do not alter the image content.

These approaches are based on the fact that the digital image has some consistent inherent patterns (statistical properties) which are acquired from the distinct phases that are part of the image history [34], like the acquisition phase, storage and compression, post processing operations etc.. Each phase leaves a unique trace on the image, which works as a digital fingerprint. These patterns are altered during image forgery operations is performed on the image and the mutated image properties can be detected by applying some statistical image functions.

This technology, defined as *multimedia forensics* [11] [47] [25], deals with such issues by studying and developing technological tools which generally permit determining, by only analyzing a digital photograph, if the considered asset has been manipulated or even which could have been the adopted acquisition device. Moreover, if it has been determined that something has been altered, it could be important to understand in which part of the image itself such modification occurred: for instance, if a person or a specific object has been covered, if an area of the image has been cloned, if something (*i.e.*, a face, an object) has been copied from another different image, or, even more, if a mixture of these processes has been carried out.

Methods for detecting image composition have become a hot research topic in the forensic analysis process. Different types of schemes have been proposed

for detecting image forgeries: methods based on inconsistencies in compatibility metrics, JPEG compression features and perspective constraints are just a few examples of inconsistencies explored to detect forgeries.

1.3 Methods based on light inconsistencies

After studying and analyzing the advantages and drawbacks of different types of methods for detecting image composition, this work herein relies on the research hypothesis that image illumination inconsistencies are strong and powerful evidence of image composition.

This hypothesis has already been used by some researchers in the literature whose work will be detailed in the next sections, and it is specially useful for detecting image composition because, even for expert counterfeiters, a perfect illumination match is extremely hard to achieve. Furthermore, there are some experiments that show how difficult is for humans to perceive image illumination inconsistencies. [31]

We can divide methods that explore illumination inconsistencies into three main groups:

1. methods based on inconsistencies in the **light setting**: this group encloses the approaches that look for inconsistencies in the light position and the models that aim at reconstructing the scene illumination conditions.

2. methods based on inconsistencies in the **shadows**: this group encloses the approaches that look for inconsistencies in the scene illumination using telltales derived from shadows.

3. methods based on inconsistencies in **light color**: this group of methods encloses the approaches that look for inconsistencies in the color of illuminants present in the scene.

1.3.1 Inconsistencies in the light setting

Johnson and Farid [22] proposed an approach based on illumination inconsistencies, looking for chromaticity aberrations as an indicator of image forgery. They analyzed the light source direction on different objects in the same image trying to detect traces of tampering. The authors start by imposing different constraints for the problem:

1. All the analyzed objects have *Lambertian surface*[1] [26].

2. The surface reflectance is constant.

3. The object surface is illuminated by an infinitely distant light source.

Using RGB images, the authors assume that the standard deviation of the chromaticity is constant for all color channels and they create a model, based on image statistical properties, to provide how the ray light should split for each color channel. Given this premise and using the green channel as reference, the authors estimate deviations between the red and green channels and between the blue and green channels for selected parts of the image. Inconsistencies on this split pattern are used as telltales to detect forgeries.

A drawback of this method is that chromaticity deviation depends on the camera lens used to take the picture.

1.3.2 Inconsistencies in the shadows

Another set of methods is based on inconsistencies on the shadows in the image.

[1] A Lambertian surface for reflection is a surface that appears uniformly bright from all directions of view and reflects the entire incident light. Lambertian reflectance is the property exhibited by an ideal matte or diffusely reflecting surface.

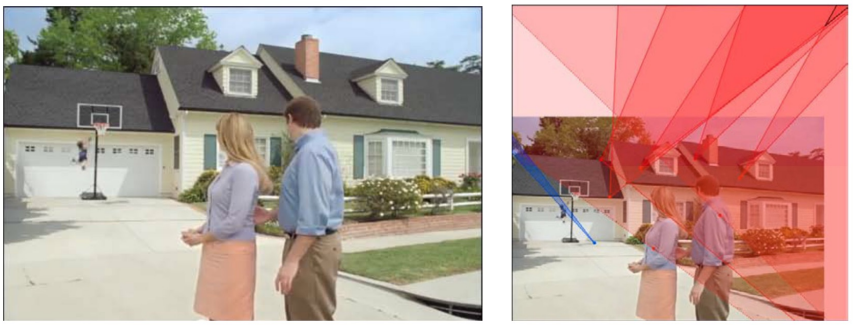

Figure 1.7: Original image (left) and the extracted shadows constraints (right)

Zhang and Wang [51] proposed an approach that utilizes the planar homology [40], which models the relationship of shadows in an image for discovering forgeries.

Based on this model, the authors proposed to construct two geometric constraints. The first one is based on the relationship of connecting lines. A connecting line is a line that connects some object point with its shadow. According to planar homology, all of these connecting lines intersect in a vanishing point.

The second constraint is based on the ratio of these connecting lines. In addition, the authors also proposed to explore the changing ratio along the normal direction of the shadow boundaries.

Geometric and shadow photometric constraints together are used to detect image compositions. However, in spite of being a good initial step in forensic shadow analysis, the major drawback of the method is that it only works with images containing casting shadows, a very restricted scenario.

1.3.3 Inconsistencies in light color

The last group of methods investigate the presence, or not, of composition operations in digital images using color inconsistencies.

Gholap and Bora [18] [14] pioneered this approach using the *illuminant colors*. For that, the authors used a *dichromatic reflection model* proposed by

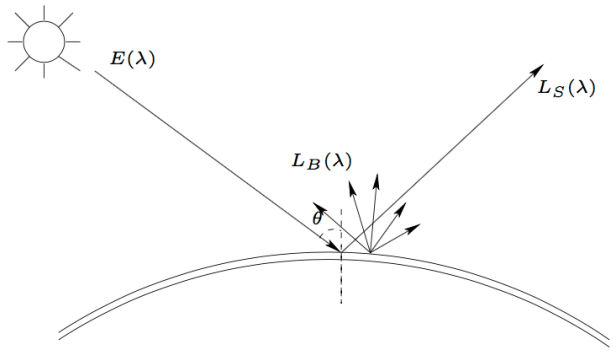

Figure 1.8: The dichromatic reflection model

Tominaga and Wandell [44], which assumes a single light source to estimate illuminant colors from images.

According to this model, the reflection of any non-homogeneous material may be modelled as additive mixture of two components, diffused reflection and surface reflection as shown in Fig. 1.8.

Considering an object point illuminated by a light source, the reflected ray consists of diffuse reflection $L_B(\lambda)$ and surface reflection $L_S(\lambda)$. The reflected light $L(\Theta, \lambda)$ can be written as:

$$L(\Theta, \lambda) = m_S(\Theta)L_S(\lambda) + m_B(\Theta)L_B(\lambda) \tag{1.1}$$

where $m_S(\Theta)$ and $m_B(\Theta)$ are geometrical factors and Θ the angle of the incident light. This equation can be rewritten in terms of RGB sensors in matrix form.

The two vectors $L_B(\lambda)$ and $L_S(\lambda)$ span the two dimensional plane called *dichromatic plane*.

Dichromatic planes can also be estimated using principal component analysis (PCA) from each specular highlight region of an image. By applying a *Singular Value Decomposition (SVD)* on the RGB matrix extracted from high-

lighted regions, the authors extract the eigenvectors associated with the two most significant eigenvalues to construct the *dichromatic plane*. This plane is then mapped onto a straight line, named dichromatic line, in normalized *r-g-chromaticity* space.

For distinct objects illuminated by the same light source, the intersection point produced by their dichromatic line intersection represents the illuminant color. If the image has more than one illuminant, it will present more than one intersection point, which is not expected to happen in pristine (non-forged images). This method represented the first important step toward forgery detection using illuminant colors, but has some limitations such as the need of well defined specular highlight regions for estimating the illuminants.

Following Gholap and Bora's work, Riess and Angelopoulou [37] used an extension of the *Inverse-Intensity Chromaticity Space*, originally proposed by Tan *et al.* [43], to estimate illuminants locally from different parts of an image for detecting forgeries.

In addition, Wu and Fang [50] proposed a new way to detect forgeries using illuminant colors. Their method divides a color image into overlapping blocks estimating the illuminant color for each block. To estimate the illuminant color, the authors proposed to use the algorithms Gray-World, Gray-Shadow and Gray-Edge [46].

These approaches will be better explained in the next section due their importance for this proposed method.

1.4 Illuminant color estimation

The color of an object observed in an image depends both on its intrinsic color and on the color of light source *i.e.*, the illuminant.

It is important to keep in mind that even the same light source can generate different illuminants. The illuminant formed by the sun, for example, varies in its appearance during the day and time of year as well as with the weather. We only capture the same illuminant, measuring the sunlight at the same place at the same time.

We can explore illuminants in forensics to check the consistency of similar objects in a scene. If two objects with very similar color stimuli (e.g., human skin) depict inconsistent appearance (different illuminants), it means that they might have undergone different illumination conditions hinting at a possible image composition. On the other hand, if we consider a photograph with two people and the color appearance on the faces of such people are consistent, it is likely they have undergone similar lighting conditions.

Riess and Angelopoulou [37] pioneered the approach of using a color-based method that investigates illuminant colors to detect forgeries in forensic scenario. In their work, illuminant colors are estimated locally, effectively decomposing the scene in a map of differently illuminated regions. Inconsistencies in such a map suggest possible image tampering.

The method can be divided into four steps:

1. Image segmentation. The image is segmented into regions, called *superpixels*, of approximately the same object color. Each superpixel must be:
   (a) Directly illuminated by the same light source.
   (b) Compliant with the *dichromatic reflectance model* [43].
2. Manual user selection of superpixels under investigation.
3. Local illuminant color estimation for each superpixel (both for segmented superpixels and user selected regions).

Figure 1.9: An example of a generated distance map using Riess and Angelopoulou [37] approach. From left to right: (a) the original image, (b) *illuminant map*, (c) generated distance map

4. Reference illuminant selection (manual) and *distance map* evaluation. The distance map is the base for an expert analysis for forgery detection.

The final decision of this approach is delegated to an expert and not to the algorithm itself.

1.4.1 Illuminant maps

1.4.1.1 Generalized Greyworld estimation

The starting point is the *gray world assumption*, proposed by Buchsbaum [3]. In its simplest version, it is assumed that information in the average of each channel of the image is the representative gray level.

Let $f(x) = (\Gamma_R(x), \Gamma_G(x), \Gamma_B(x))^T$ the RGB color of a pixel at position x and $\Gamma_i(x)$ the intensity of that pixel in the i-th channel. In their paper, Van de Weijer *et al.* assumes that the camera response is linear. It is also assumed that the scene is illuminated by a single light source.

The RGB color $f(x)$ can also be rewritten as

$$f(x) = \int_\omega e(\beta, x)s(\beta, x)c(\beta)d\beta \tag{1.2}$$

where ω is the visible light spectrum, β is the light wavelength, $e(\beta, x)$ is the illuminant spectrum, $s(\beta, x)$ is the surface reflectance of an object and $c(\beta)$ is

the color sensitivity of the camera for each channel.

As an alternative to the gray-world hypothesis, Van de Weijer *et al.* [46] proposed the *gray-edge hypothesis*: the average of the reflectance differences in a scene is achromatic.

This idea led to a framework for low-level based illuminant estimation, called *Generalized Grayworld*.

$$\left(\int \left| \frac{\delta^n f^\sigma(x)}{\delta x^n} \right|^p dx \right)^{\frac{1}{p}} = k e^{n,p,\sigma} \tag{1.3}$$

where k denotes a scaling factor, $|\cdot|$ the norm operand, δ the differential operator and $f^\sigma(x)$ the pixel intensities at position x, smoothed with a Gaussian kernel σ.

The framework defined by (1.3) produces different estimations for the illuminant color based on three parameters:

1. The order n determines if the method is a gray-world or a gray-edge algorithm. The gray-world methods are based on the RGB values, whereas the gray-edge methods are based on the spatial derivative of order n.

2. The Minkowski norm p determines the relative weights of the multiple measurements from which the final illuminant color is estimated.

3. The scale σ of the local measurements. For first or higher order estimation, this local scale is combined with the differentiation operation computed with the Gaussian derivative. For zero-order gray-world methods, this local scale is imposed by a Gaussian smoothing operation.

On varying of these parameter, a different method can be used. An overview of the possible methods that derive from Eq. (1.3) is displayed in Table 1.1.

Table 1.1: Overview of different illuminant estimation methods based on 1.3

Name	Symbol	Equation	Assumption
Gray-World	$e^{0,1,0}$	$(\int f(x)dx) = ke$	The average reflectance in a scene is achromatic
max-RGB	$e^{0,\infty,0}$	$(\int \lvert f(x)\rvert^{\infty}dx)^{\frac{1}{\infty}} = ke$	The maximum reflectance in a scene is achromatic
Shades of Gray	$e^{0,p,0}$	$(\int \lvert f(x)\rvert^{p}dx)^{\frac{1}{p}} = ke$	The p-th Minkowski norm of scene is achromatic
General Gray-World	$e^{0,p,\sigma}$	$(\int \lvert f^{\sigma}(x)\rvert^{p}dx)^{\frac{1}{p}} = ke$	The p-th Minkowski norm of scene is achromatic after smoothing
Gray-Edge	$e^{1,p,\sigma}$	$(\int \lvert f^{\sigma}_{x}(x)\rvert^{p}dx)^{\frac{1}{p}} = ke$	The p-th Minkowski norm of the image derivative is achromatic
Max-Edge	$e^{1,\infty,\sigma}$	$(\int \lvert f^{\sigma}_{x}(x)\rvert^{\infty}dx)^{\frac{1}{\infty}} = ke$	The maximum reflectance difference in a scene is achromatic
2nd order Gray-Edge	$e^{2,p,\sigma}$	$(\int \lvert f^{\sigma}_{xx}(x)\rvert^{p}dx)^{\frac{1}{p}} = ke$	The p-th Minkowski norm of the second order derivative in a scene is achromatic

1.4.1.2 Inverse Intensity-Chromaticity estimation

The other considered illuminant estimation method is based on the idea proposed by Tan *et al.* [43], called *inverse intensity-chromaticity*.

The base for this kind of approaches is the dichromatic reflectance model [18], which states that the amount of light reflected from a point, x, of a dielectric, non-uniform material is a linear combination of diffuse reflection and specular reflection. Further assumptions assume that the color of the specularities approximates the color of the illuminant, and that the camera response is linear.

Considering a trichromatic camera, the sensor response $I_c(x)$, for each color channel $c \in \{R, G, B\}$ is:

$$I_c(x) = m_S(x)L_{S_c}(x) + m_B(x)L_{B_c}(x) \tag{1.4}$$

as described in 1.1.

Let σ_c the image chromaticity, $\Delta_c(x)$ the diffuse chromaticity and $\Gamma_c(x)$ the specular chromaticity defined as follows:

$$\sigma_c(x) = \frac{I_c(x)}{\sum_i I_i(x)} \text{ where } i \in \{R, G, B\} \tag{1.5}$$

$$\Delta_c(x) = \frac{L_{S,c}(x)}{\sum_i L_{S,i}(x)} \text{ where } i \in \{R, G, B\} \tag{1.6}$$

$$\Gamma_c(x) = \frac{L_{B,c}(x)}{\sum_i L_{B,i}(x)} \text{ where } i \in \{R, G, B\} \tag{1.7}$$

Thus the Eq. 1.4 can be rewritten as

$$I_c(x) = m_S(x)\Delta_c(x) + m_B(x)\Gamma_c(x) \tag{1.8}$$

Tan *et al.* [43] derived a linear relationship between diffuse, specular and image chromaticities:

$$\sigma_c(x) = m(x)\frac{1}{\sum_i I_i(x)} + \Gamma_c(x) \tag{1.9}$$

where $i \in \{R, G, B\}$ and $m(x)$ is a geometrical factor (light position, surface orientation, camera position, ...), that can be approximated. In the illuminant estimation, the most important aspect is the y-intercept Γ_c.

The 2D space defined by $\frac{1}{\sum_i I_i(x)}$ as domain and $0 \leq \sigma_c \leq 1$ as range is called *inverse-intensity chromaticity (IIC)* space.

An example of the IIC plots for a single channel of a synthetic image is shown in Fig. 1.10. Pixels from the green and purple balls form two clusters. The clusters have spikes that point towards the same location on the y-axis. Considering only such spikes from each cluster, the illuminant chromaticity is

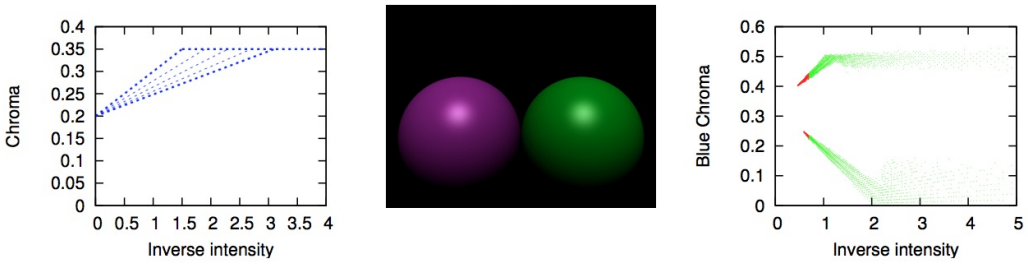

Figure 1.10: Pixel distribution in inverse-intensity chromaticity (IIC) space. Left are as ideal distribution, right on a synthetic image (in the center).

estimated from the joint y-axis intercept of all spikes in IIC space.

Instead of examining the entire pixel distribution, Riess *et al.* [37] perform the analysis over small connected image regions of roughly uniform object color (*superpixels*). Depending on the outcome of our shape analysis, we can either use this local region to obtain an illuminant estimate, or reject it if it does not seem to fulfill the underlying assumptions of the proposed model. Using local regions allows us to incorporate multiple sampling and voting in the estimation of local illuminants.

1.5 Human faces splicing detection

The approach proposed in Chapter 2 is based on two different works, published by Carvalho *et al.* [4] and Fan *et al.* [9].

The method proposed by Carvalho *et al.* [4] aims to detect splicing focusing on human faces, minimizing the user interaction. Faces are previously labeled by a human selecting the box within is contained, than the process will associate a label to each face (*i.e. normal* or *fake*).

The method can be divided into four steps:

1. *Description*: in this step illuminant maps are estimated with the two different approaches, GGE and IIC, and feature vectors are generated.

Each feature vector is associated with a face pair.

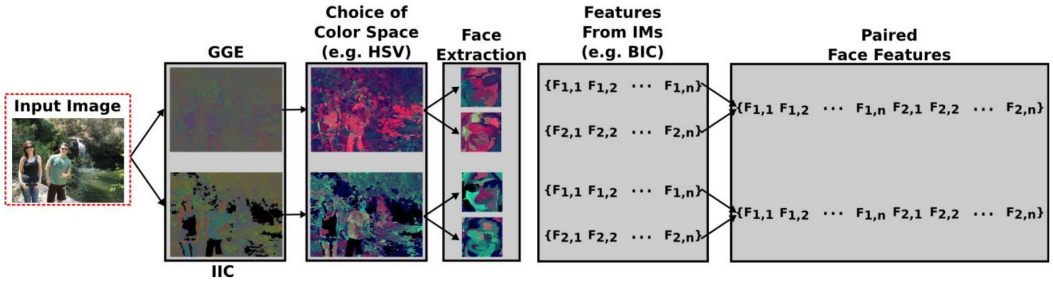

Figure 1.11: Image description pipeline for extracting paired face feature vectors

The Fig. 1.11 shows the image description extraction pipeline. Given an input image, the illuminant maps are estimated and converted to a selected color space (e.g. HSV). So, the faces in the image are extracted (using the defined area). For each face a descriptor is used to generate a feature vector (e.g a color descriptor, texture descriptor or shape descriptors) and finally they are coupled in order to generate paired feature vector simply concatenating the two original vectors.

In this step multiple descriptors and color spaces are used in order to increment the number of final classifiers.

2. *Face Pair Classification*: a set of classification models are trained using the previous step feature vectors. Based on the number of color spaces and descriptors used, a set of KNN classifiers are trained over the paired face feature vectors. The final result is given by a majority voting of all the selected classifiers.

3. *Forgery Classification*: given an image I containing q people (faces), it is characterized by a set $\mathcal{S} = \{\mathcal{P}_1, \ldots, \mathcal{P}_m\}$, where $\mathcal{P}_i$ is the i-th paired feature vector and $m = \frac{q(q-1)}{2}, q \geq 2$. If any $\mathcal{P}_i \in \mathcal{S}$ is classified as fake, the image I is classified as fake. Otherwise, the image is considered as

pristine.

4. *Forgery Detection*: once knowing that an image is fake, in this stage it is identified which one is more likely to be fake in the image. This is done using a specific SVM classifier.

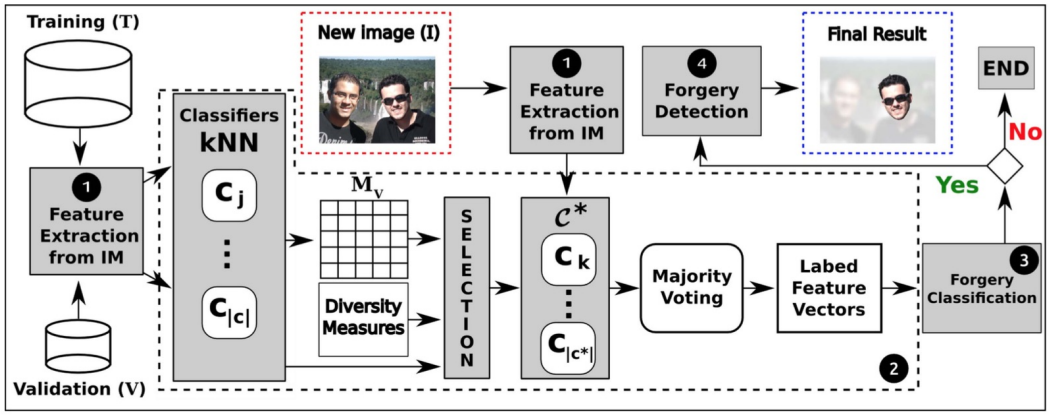

Figure 1.12: Image splicing detection over human faces

The Fig. 1.12 shows the above described method pipeline.

1.5.1 Method drawbacks

The main drawback of this approach relies in the manual selection of faces by a human agent. The definition of a face in the image by the operator is made by identification of the bounding box in which it is fully contained.

This manual operation does not allow a blind approach to the problem. Because the faces are considered in pairs, this method can be applied only in the case when there are at least two faces, so you need to know in advance the content of the images to assess the applicability of this algorithm.

Working with pairs of faces, another problem is the choice of which face has been manipulated among those considered.

In addition, this method is based on the observation that, given two faces (a pristine face and a spliced one), the difference between the two forged faces illuminant maps tend to be higher. This intuition, however, proved to be not applicable in all contexts, so cannot be considered as a starting point for the final evaluation.

1.6 Region splicing detection

The other considered approach is based on the work of Fan *et al.* [9]. This method relies on the Van de Weijer *et al.* [46] Generalized Gray-World framework 1.3 in Eq. 1.3. This algorithm performs better on a scene when it is rich of colors, but in a splicing detection scenario we are interested in a specific region of the image.

This approach splits the image in vertical and horizontal bands (rectangular regions), assuming that each band contains sufficient colors for a correct illuminant estimation. Five different algorithms are used, deduced from Eq. 1.3: Grey-World, Max-RGB, Shades of Grey, first- order Grey-Edge and second-order Grey-Edge, in order to have as many illuminant estimates for each band.

This algorithm can be divided into two steps:

1. Subsampling of the image horizontally and vertically;

2. Illuminant estimation for each band using the 5 different algorithms and spliced region location.

In the first step the image is sampled into two band categories: horizontal and vertical bands. The height and the width of each band is configured *a priori* based on the minimal height and minimal width separately among the objects of interest. An object of interest is encompassed by a virtual rectangle with its height and width decided by the object itself.

The second step can be also divided into five steps:

1. For each direction and for each algorithm, a reference illuminant is estimated for a single band.

2. For each direction and for each algorithm, a reference illuminant for each direction is evaluated as the median of all references of that direction. At this stage there are two reference illuminants (one for the vertical and one for the horizontal direction) for each algorithm.

3. A detection map is created, with the same dimensions of the image.

4. For each algorithm, every band estimate is compared with the reference illuminant of that band direction and algorithm with the Euclidean distance. If the distance exceeds a fixed threshold, the band is considered fake and all the pixel values in the detection map are increased by one unit.

5. The forgery is located using the resulting thresholded detection map.

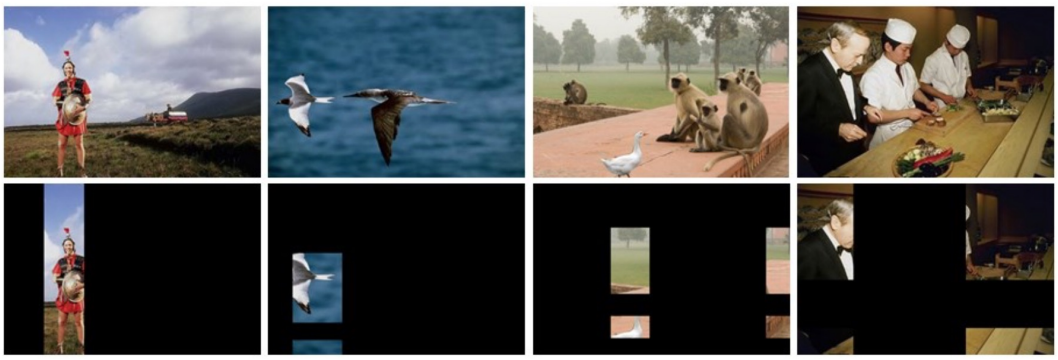

Figure 1.13: Image regions splicing detection

The main advantage of this method is its blind approach: the algorithm is not based on the image content (as it was in the previous case) and no human interaction is needed. The only parameters are the bands minimal dimensions and the threshold. Due these facts it is very simple to implement.

1.6.1 Method drawbacks

Due its simplicity, this approach has some drawbacks. First, because the classification is based on a simple distance between a reference value, a key point is the identification of optimal thresholds.

These thresholds can be calculated experimentally, but their accuracy will depend on the dataset with which they were computed.

For how it's built, moreover, this method tends to have a low detection rate compared to a high false positive rate.

Chapter 2

Proposed approach

In the previous chapter, a review of two splicing detection methods based on illuminant colors analysis has been presented. However, their effectiveness still to be improved for real forensic applications.

The approach proposed in this chapter has been developed to correct some of the two approaches drawbacks and mainly to combine them together in order to realize a more general algorithm for image forgery detection.

2.1 Overview

Most of the times, the splicing detection process relies on the expert's experience and background knowledge. This process usually is time consuming and error prone once that image splicing is more and more sophisticated and an aural (e.g., visual) analysis may not be enough to detect forgeries.

This approach to detecting image splicing is developed aiming at minimizing the user interaction.

The two methods, presented in the previous chapter ([4] and [9]), are now being used in synergy, to analyze the suspicious image and at the same time looking for potential signs of forgery.

Starting from an image we want to analyze, the method will output a set of results.

- A classification **label** indicating whether an image is believed to be original.
- A classification **score** indicating the confidence of the algorithm output.
- A **detection map** highlighting the detected spliced regions.

The proposed approach minimizes the needed human interaction and aims at being a fully automated process. However, face splicing detection module works only if the number of faces in the image is greater than or equal to two.

2.2 Face splicing detection module

The first module consists in the implementation, with few enhancements and simplifications, of the method proposed by Carvalho et al. [4] and presented in Section 1.6.

The algorithm can be used on an image that contains at least two human faces, looking for inconsistencies in the illuminant color in presence of human skin. In particular, in case of a forgery detection, the algorithm tries to guess which faces in the images are doctored.

The algorithm requires an initial training phase, so a dataset with annotated human faces is needed.

This module consists of 5 consecutive stages:

1. **Illuminant maps extraction**: given an input image, two different illuminant maps are evaluated, using the IIC and GGE extraction methods.
2. **Face detection**: after the illuminant maps extraction, the human faces in the image are detected. In the training phase the faces positions are read from the groundtruth file. If the given image contains less than two faces, it is discarded.

3. **Paired face feature extraction**: human faces are considered and classified in pairs. From each extracted face in the previous step, a color descriptor is used in order to extract features.

4. **KNN models training**: fixed a value of k[1], a set of kNN models are trained using previous feature vectors.

5. **Forgery detection and classification**: in this step an image is classified as fake or normal. Given an image classified as fake, the analysis is refined pointing out which face of the image is the result of a composition.

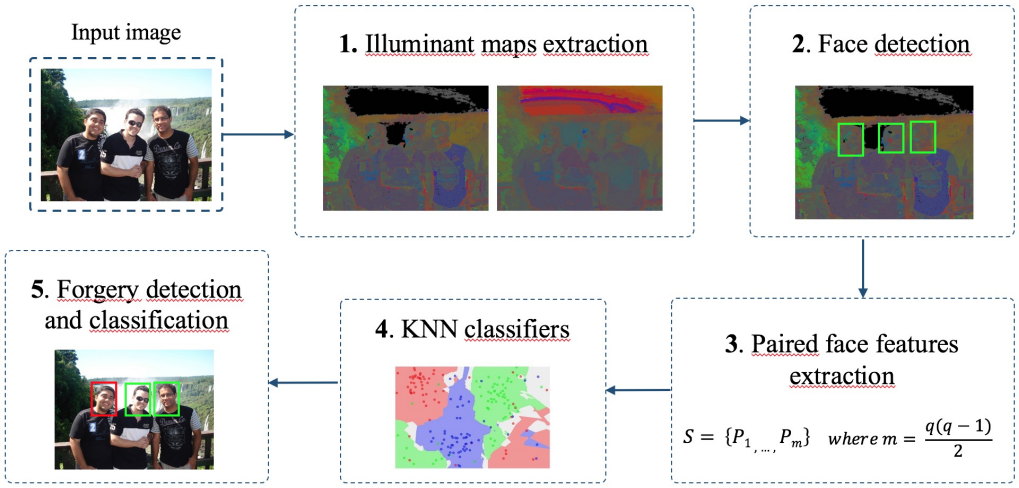

Figure 2.1: Image face splicing detection module pipeline.

Figure 2.1 gives an overview of the proposed module.In the following, these five steps will be described in detail.

[1]In kNN classification, an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.

2.2.1 Illuminant maps extraction

Given a single image I the two different illuminant maps are extracted: both generalized grayworld algorithm (GGE) and inverse-intensity chromaticity estimation (IIC) are used.

Figure 2.2: The original image

Figure 2.3: GGE illuminant map

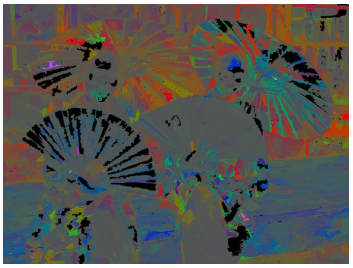

Figure 2.4: IIC illuminant map

Differently from de Carvalho *et al.* [4], who have used the conversion of IMs to the YCbCr color space, the two IMs are here considered as two different color maps theirself.

The illuminant maps extraction algorithm implementation used is the one proposed by Riess *et al.* for [37] and downloadable in the project page[2]. The code provides an implementation of the two illuminant map extraction methods, using a local estimation approach.

2.2.2 Face detection

One of the drawback of the method proposed by Carvalho et al. [4] relies on the user interaction needed for detecting human faces in the image. In this approach a face detection module is used to achieve the same results, aiming at minimizing

[2]Homepage of the project [37] - www5.cs.fau.de/research/areas/computer-vision/image-forensics/scene-illumination-as-an-indicator-of-image-manipulation

the user dependency and making the entire algorithm user independent.

The face detector used is the one initially proposed by Viola and Jones [48] [49] and improved by Lienhart *et al.* [24]. Usually called simply *Viola-Jones*, its original motivation was face detection, but it can be trained to detect different object classes.

This detector combines four key concepts:

- Simple rectangular features, called *Haar features*
- *Integral Image* [7] concept for rapid feature detection
- *AdaBoost* [15] machine-learning method
- A *cascade classifier* to combine all features efficiently

The used rectangle combinations are not true *Haar wavelets* [19]. Instead, they contain better suited rectangle combinations used for visual object detection. The presence of an Haar feature is determined by subtracting the pixel values of the dark region from the pixel values of the light one. If the difference exceeds a threshold value set during the training process, the feature is said to be present.

OpenCV[3] provides an implementation of the Viola-Jones face detector as *cvHaarDetectObjects*.

Figure 2.5 shows the output of the face detector module. Given the output faces bounding boxes, the corresponding regions in the illuminant maps are extracted, Figure 2.6.

[3]OpenCV - Open source computer vision. http://opencv.org

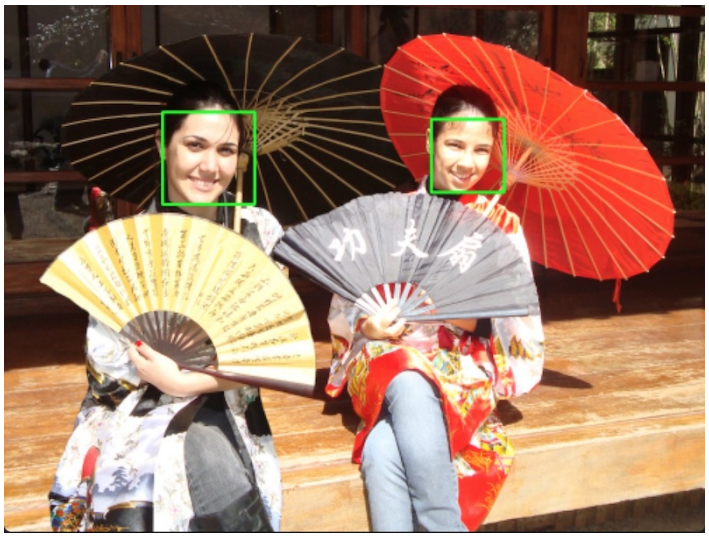

Figure 2.5: The output of the Viola and Jones face detector

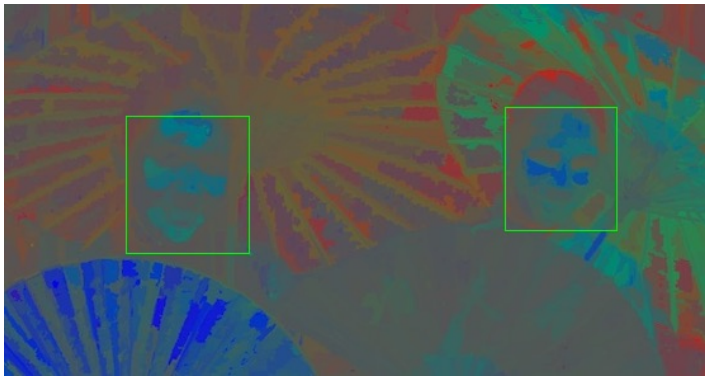

Figure 2.6: The extracted faces from illuminant map

2.2.3 Paired face feature extraction

From each extracted face in the previous step, we need to find telltales that allow identification of spliced images.

According to Riess and Angelopoulou [37], when the output illumination maps are analyzed by an expert for detecting forgeries, the main observed feature is color. Thus, differently from Carvalho *et al.* [4] who have explored a large set of descriptors of different visual properties (e.g., texture, shape, color, among

others) we focused our attention on color descriptors.

The considered color description techniques are ACC [20], BIC [41], CCV [32], and LCH [42].

2.2.3.1 ACC descriptor

Color Autocorrelogram (ACC) [20] maps the spatial information of colors. Let I an image, the *autocorrelogram* (the term *correlogram* is adapted from spatial data analysis [45]) computes the probability of finding two pixels in I with a color c in a distance d from each other.

After the autocorrelogram computation, a set of m probability values for each distance d are considered, where m stands for the number of colors in the quantized space.

The implemented version quantized the RGB color space into 64 bins and considered 4 distance values (1, 3, 5, and 7). The *L1* distance function is used.

2.2.3.2 BIC descriptor

Border/Interior Pixel Classification (BIC) [41] is a region-based color descriptor. Its extraction algorithm classifies the image pixels in border or interior pixels.

The image is first quantized into 64 colors in RGB color space. Then, each pixel is classified as interior if its neighbors (above, below, left and right pixels) have the same color. Otherwise it is classified as border pixel. After the classification, two histograms are generated: one for border pixels and another for interior pixels. These histograms are stored as one single histogram with 128 bins.

2.2.3.3 CCV descriptor

Color Coherence Vector (CCV) [32] is a very popular color descriptor in the literature. Its extraction algorithm classifies the image pixels in coherent or incoherent pixels. This classification considers if the pixel belongs or not to a region with similar colors, called *coherent region*. After the classification, two color histograms are computed: one for coherent pixels and another for incoherent pixels.

Both histograms are concatenated to compose the final feature vector. The RGB color space is quantized into 64 bins and L1 distance function is used.

2.2.3.4 LCH descriptor

Local Color Histogram (LCH) [42] is one of the most popular descriptors that is based on fixed size regions to describe image properties. Its extraction algorithm splits the image into fixed size blocks and computes a color histogram for each region. After that, the histograms of each region are concatenated to compose one single histogram.

The implemented version splits the image in 16 regions (4x4 grid) and quantizes the RGB color space into 64 bins. This generated feature vectors with 1024 values. The L1 distance function is used.

2.2.3.5 Paired features

In order to detect a face forgery, we need to compare the current analyzed image region (*i.e.* the current face) against another one, looking for color inconsistencies. Thus, instead of considering each face separately, faces are encoded in pairs.

Given two face feature vectors of the same type (*i.e.* extracted with the same color descriptor), $\mathcal{D}_1$ and $\mathcal{D}_2$, whose length depends on the color descriptor used,

the final descriptor is computed concatenating them into a single feature vector $\mathcal{P}$. So, in an image I containing q faces, a set S of feature vectors are extracted, with

$$S = \{\mathcal{P}_1, \ldots, \mathcal{P}_m\} \quad \text{where } m = \frac{q(q-1)}{2}$$

if $q \geq 2$.

Each paired face descriptor is computed twice for each considered color descriptor, one based on the GGE map and one based on the IIC map.

2.2.4 KNN models training

As in the Carvalho *et al.* [4], k-Nearest Neighbor (kNN) classifiers [1] are chosen for the classification step.

After the feature vectors are extracted from all faces in the images, a set of 8 *k-nearest neighbors* binary classifiers are trained, 4 models based on the GGE map and 4 based on the IIC map.

Accordingly to the analysis presented in [4], the value of k is set to 5.

Given a set of paired feature vectors $\mathcal{P}$, extracted from a new image I, each classifier C_i $(i = 1, \ldots, 8)$ predicts a label (forgery or real) for these feature vectors, producing k outcomes. All the outcomes are used as input of a fusion technique (in this case, majority voting) that takes the final decision regarding the definition of each paired feature vector $\mathcal{P}$ extracted from I.

$$Score(\mathcal{P}) = \sum_{i=1}^{8} prediction_{C_i}(\mathcal{P}) \quad \text{where } prediction_{C_i}(\mathcal{P}) \in [0, 1] \tag{2.1}$$

Finally, $\mathcal{P}$ is considered as fake if its score is higher than a given threshold.

An alternative approach for the feature score evaluation has been experi-

mented and presented in Chapter 3.

2.2.5 Forgery detection and classification

Differently from Carvalho *et al.* [4], we used another approach for classifying the single face as fake using the results of the paired faces classification.

Let $P_{i,j}$ the paired feature vector computed concatenating the i-th and the j-th face feature. If $P_{i,j}$ is classified as fake, the scores of the considered faces are increased by the prediction output value. At the end of the process, the faces with higher associated scores have been classified as fake multiple times.

The resulting scores are then thresholded for selecting the most probably doctored faces.

Given an input image I containing q faces, the forgery detection algorithm is summarized in Algorithm 1.

Algorithm 1 Face forgery detection

$Score_i = 0 \quad \forall i \in Faces$
for $im \in \{GGE, IIC\}$ **do**
  for $desc \in \{ACC, BIC, CCV, LCH\}$ **do**
    $F = extractFeatures(im, desc)$
    So $F = \{\mathcal{P}_1, \ldots, \mathcal{P}_m\}$ where $m = \frac{q(q-1)}{2}$
    for $f \in F$ **do**
      $prediction = KNN.predict(f)$
      Let i, j the faces considered for f evaluation
      Increment i, j scores by $prediction \in [0, 1]$
    end for
  end for
end for
for $i \in Faces$ **do**
  if $Score_i > Threshold$ **then**
    Mark i as *fake*
  end if
end for

The output of the detector is a resulting *detection mask*, which highlights the spliced part of the image.

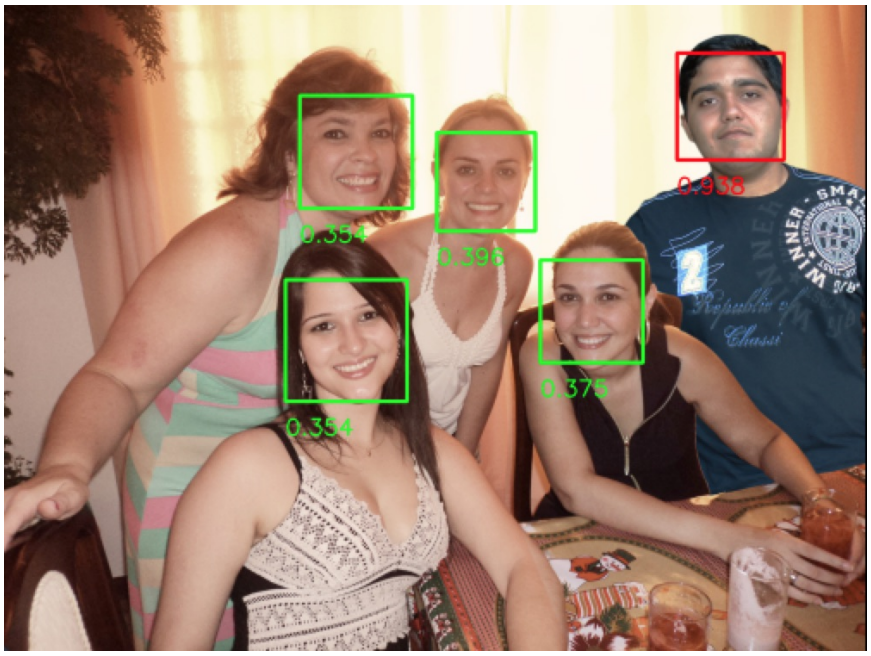

Figure 2.7: An example output of the face splicing detection module. Each face is marked in *green* if it has been classified as *normal*, *red* if considered as *fake*. A classification confidence score is presented below each bounding box.

An example is shown in Figure 2.7. Each face is marked as fake or normal and provided of a classification confidence score. The higher the score, the higher the confidence of the prediction output. In the analyzed image (Figure 2.7), four of the total faces are marked in *green* because they have a low score value. The last face, corresponding to the spliced person, has a very high prediction score and it is therefore marked in *red*.

2.3 Region splicing detection module

The second module is based on the work of Fan *et al.* [9] (presented in Section 1.7) with some improvements and a different approach for splicing detection.

In summary, this module consists of the 6 main steps:

- **Image segmentation**: relies on vertical and horizontal image segmentation. The outputs of this stage are two sets of directional image bands.

- **Band illuminant estimation**: consists in estimating the illuminant color for each segmented band using 5 different GGE algorithms.
- **Reference illuminant estimation**: consists in estimating an illuminant reference value.
- **Feature vector evaluation**: relies on encoding the singular band illuminant information into a feature vector for further classification. The feature vector elements are the differences between the current illuminant color and the illuminant reference.
- **Band classification**: consists in labeling each image band into one of the known classes (real or fake) basing on the previously learned classification model.
- **Output detection map**: using the classification output of the previous step, a detection map for the image is computed. The higher the value of the pixel in the map, the higher the resulting classification score for that pixel.

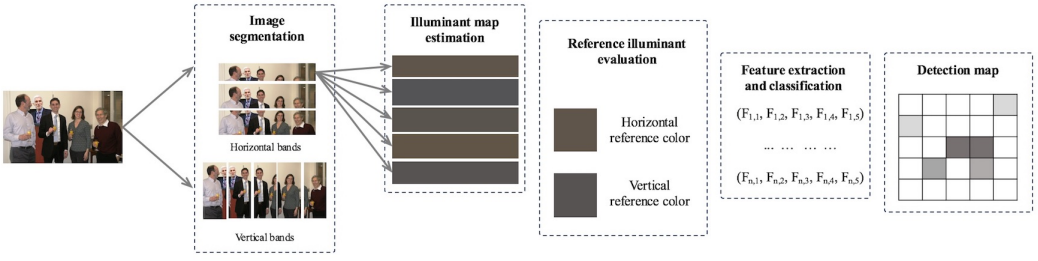

Figure 2.8: Image regional splicing detection module pipeline

The Figure 2.8 summarizes the module pipeline.

2.3.1 Image segmentation

In the first step of the process, the input image is segmented in order to obtain two image bands categories: horizontal and vertical bands. This kind of segmentation method is chosen because of its simplicity.

First, a band width B_w and a band height B_h is set. Due the fact that the segmentation has to produce overlapping bands, we set a delta factor of $\frac{1}{4}$. As a result we get overlapping stripes for a total of 25% of their area.

The choice of the band size is a crucial phase of the algorithm: a band too tight would fail to capture the information necessary to classify our object of interest as falsified, unlike a band too wide would capture too much additional information instead.

The choice of the overlapped area percentage makes possible a more detailed evaluation. In this way it is possible to classify the same region of the image more than once, increasing the expressive power of our final classifier.

In summary, let I be the input image. After the segmentation process we obtain a set B of bands containing all vertical and horizontal bands:

$$B = \{V_1, \ldots, V_n, H_1, \ldots, H_m\}$$

The dimension of B is given by the sum of the number of vertical (n) and horizontal (m) bands.

2.3.2 Band illuminant estimation

The resulting image bands are now processed in order to evaluate their illuminant colors using 5 different techniques.

For this step, the Generalized Grayworld [46] algorithms are used, as presented in Chapter 1. For each band, the illumination estimation is accomplished

by using one of the algorithms composed of Grey-World, Max-RGB, Shades of Grey, first-order Grey-Edge and second-order Grey-Edge. Thus, we get 5 illuminant estimates for each band. Table 1.1 in Chapter 1 summarizes the algorithms parameters.

$$\forall b \in B \qquad R_a(b) = GGE_a(b) \qquad a \in \mathcal{A}$$

where $\mathcal{A}$ is the set of the previously mentioned algorithms and GGE_a is the algorithm implementation.

2.3.3 Reference illuminant color estimation

After estimating the illuminant of each horizontal/vertical band, the reference illuminant color is evaluated. Differently from Fan *et al.* [9], the whole image illuminant color (obtained using the previous algorithms) is considered as reference instead of the median of all the illuminant colors extracted from all the bands of a specific direction.

In the approach proposed in [9] the reference colors were computed as follows.

$$\forall a \in \mathcal{A} \qquad RV_a = median(R_a(b)) \quad \forall\ b \text{ vertical}$$

$$\forall a \in \mathcal{A} \qquad RH_a = median(R_a(b)) \quad \forall\ b \text{ horizontal}$$

where RV_a and RH_a are the reference illuminant colors for the a algorithm for vertical and horizontal direction respectively. As these two values are calculated using the median, they will be identical to one of the value of a band of that same direction.

More simply, in our case the considered reference colors are the estimation results of the 5 different algorithms over the whole image. A comparison between the two estimation approaches is presented in Chapter 3.

2.3.4 Feature vector estimation

Given the two reference colors for each of the two directions, a feature band for each single image band can be built.

Assuming a single light source in the image, all the evaluated illuminants will point at the same color. Based on this assumption, a feature vector aimed at capturing illuminant colors inconsistencies can be computed.

Let $b \in B$ a single band lying on $d \in \{vertical, horizontal\}$ direction. The feature vector for b will be:

$$f_b = \{m_1, m_2, m_3, m_4, m_5\} \tag{2.2}$$

where

$$m_i = dist(R_a(b) - RC)$$

and RC is the chosen reference color and $dist$ is the Euclidean distance function between two RGB values.

2.3.5 Band classification

Given a feature vector, a machine learning approach is used to automatically classify the band.

After training a Support Vector Machine (SVM) [1] classifier with a radial basis function (RBF) kernel, we classify the resulting multidimensional feature vector f_b and collect the positive output probability of this input vector, considering it as the band score.

For the SVM model training two custom datasets were specifically built containing images with single spliced band. More details about training are given in Chapter 3.

$$\forall\, b \in B \quad Score(b) = SVM.predict(f_b)$$

2.3.6 Output detection map

In order to collect all the classification outputs, a detection map is built and updated after each classifier prediction. If the result of the classification is *positive* (*fake*), all the pixel values that belong to the band portion in the image will be increased by the prediction score.

At the end of the process, the splicing region pixels should have greater values than the others. At this stage a color map is displayed to give a visual feedback for locating the splicing image parts.

Figure 2.9: The classification map for an input image. The map is presented in the *jet* color map representation. The map matrix intensities are in the range $[0, 1]$, and the color scheme is presented in Figure 2.10.

Figure 2.10: The *jet* color map scheme. The color intensities ranges from 0 (*cold* colors) to 1 (*hot* colors)

In Figure 2.9 is presented an example of a generated detection map. In this case, the spliced region consists in the woman in the middle of the scene. As the

map underlines, the more interesting region is located precisely in the middle.

Finally, the detection map is thresholded for selecting the output detected spliced regions.

Figure 2.11: The detection output of the regional detection module.

An example of the algorithm output is proposed in Figure 2.11.

Chapter 3

Experiments and results

This chapter describes the experiments we performed to show the effectiveness of the proposed method.

3.1 Evaluation datasets

To quantitatively evaluate the proposed methods, and to compare it to related work, we considered two datasets: the DSO-1 and DSI-1 datasets[1]. Each dataset comes with a face position groundtruth and a splicing region mask. Additionally, the ColorChecker dataset [17] is used as a source of pristine images.

The latter dataset has been mainly used for experimenting on pristine data: due to its characteristics it is very varied and lends itself well to image analysis based on color.

3.1.1 DSO-1

The DSO-1 dataset is composed of 200 indoor and outdoor images with the resolution of 2048 $\times$ 1536 pixels. Out of this set of images, 100 are original, i. e. they have no adjustments, and 100 are forged.

The forgeries were created by adding one or more individuals in a source

[1]Public available for download at https://recodbr.wordpress.com/code-n-data

Figure 3.1: DSO-1 sample original image

Figure 3.2: DSO-1 sample spliced image

image that already contained one or more people. When necessary, some post-processing operations were made (such as color and brightness adjustments) in order to increase photorealism.

3.1.2 DSI-1

The DSI-1 dataset is composed of 50 images with different resolutions (25 original and 25 doctored) downloaded from different websites in the Internet. Original images were downloaded from Flickr and doctored images were collected from different websites such as *Worth 1000*, *Benetton Group 2011*, *Planet Hiltron*, etc.

Figure 3.3: DSI-1 sample original image

Figure 3.4: DSI-1 sample spliced image

3.1.3 ColorChecker

The ColorChecker dataset is a collection of images for evaluating Color Constancy algorithms built as additional material to [17]. It consists in 568 RGB colored images of different scenes, both indoor and outdoor taken under different illuminations. In each scene a *Gretag MacBeth Color Checker Chart*[2] was placed such that it was illuminated by the main scene illuminant and thus its color could be retrieved. The data is available in Canon RAW format free of any correction.

Figure 3.5: Example of an image of the ColorChecker dataset

3.2 Face forgery detection module performance

This section describes the experiments performed to evaluate the face forgery detection module. In these experiments, the DSO-1 and the DSI-1 datasets are

[2]The *ColorChecker* consists in a rectangular card measuring about 11 × 8.25 inches containing a series of six gray patches, plus typical additive (Red-Green-Blue) and subtractive (Cyan-Magenta-Yellow) primaries, plus other *natural* colors such as light and dark skin, sky-blue, foliage, etc. It is usually used for color calibration of digital cameras. The color pigments were selected for optimum color constancy when comparing pictures of the chart with pictures of the natural colors. The ColorChecker was introduced in a 1976 paper by McCamy, Marcus, and Davidson in the *Journal of Applied Photographic Engineering* [27].

used for evaluation.

After having characterized an image with a specific image descriptor, the next step consists of training a classifier (a kNN model in our case).

The proposed method focuses on using complementary information (i.e the color descriptors) to describe the computed *illuminant maps*. Accordingly to Carvalho *et al.* [4], we selected the *k-Nearest Neighbor (kNN)* classifier instead of more powerful and computational intensive ones such as *Support Vector Machines (SVM)* due to the fact that the two types of classifiers (or even fusion of both learning methods) present very similar results. For that reasons, a simpler learning techinque is chosen [4]. As in [4], the value of k for the kNN classifiers is set to 5.

For all considered color descriptors (ACC, BIC, CCV and LCH), we have used the standard configuration proposed by Penatti *et al.* [33]. As described in Chapter 2, eight different kNN models are trained and used for the classification, considering all the possible combinations of the couples composed by a IMs transformed space (GGE e IIC)and a color descriptor (chosen from ACC, BIC, CCV and LCH).

3.2.1 Color descriptors accuracies

Since the final classification output is given by majority voting of all the selected classifiers, the first round of experiments aimed at evaluating the accuracy of each image descriptor.

Table 3.1 shows the results of all tested combinations of training and test set. The classification results show that, generally, LCH color descriptor yielded the higher accuracy, but there is not a descriptor that clearly outperforms the others.

Test case	Train	Test	ACC	BIC	CCV	LCH
Test 1	DSO-1	DSO-1	0.75	0.75	0.72	**0.78**
Test 2	DSI-1	DSI-1	0.78	0.79	0.77	**0.82**
Test 3	DSO-1	DSI-1	0.56	**0.57**	0.53	0.53
Test 4	DSI-1	DSO-1	0.58	0.55	0.51	**0.59**

Table 3.1: Accuracy for kNN technique using a single color descriptor. Experiments are performed using 10-fold cross-validation in test case number 1 and 2.

3.2.2 Test cases

After analyzing each single descriptor accuracy, we proceeded to evaluate the overall module performance using a combination of all of them, as described in Chapter 2. Since not all the color descriptors perform equally, a different weight w is given at each classifier. Thus, the Eq.(2.1) presented in Chapter 2 becomes:

$$Score(\mathcal{P}) = \sum_{i=1}^{8} w_i * prediction_{C_i}(\mathcal{P}) \tag{3.1}$$

where $\mathcal{P}$ is the considered paired face feature, C_i is the i-th kNN classifier, $prediction_{C_i}(\mathcal{P}) \in [0, 1]$ and $w_i \geq 1 \; \forall i = 1, \ldots, 8$.

In the following experiments, the method has been applied for classifying a face pair as fake or real using uniform (Table 3.3) and non-uniform (Table 3.4) weights. Essentially, these experiments evaluate the forgery detection performances of the face detector module.

For each test case in the suite it is reported:

1. The training dataset (***Train***);
2. The evaluation dataset (***Test***);
3. The classification accuracy score (***Accuracy***);
4. The area under the *Receiver Operator Characteristic (ROC)* curve (***AUC***)

5. The accuracy score expressed through the F_1 score (also known as ***F-Score***)

Accordingly with Jeni *et al.* [21], when dealing with AUC score to measure the performance of classifier, one of the major drawbacks relies on the fact that an increasing of AUC doesn't really mean a better classifier, but it could be just the side-effect of too many negative examples used in training.

Therefore, another classification accuracy score is also provided, the *F-Score*.

Let the classification *precision* score given by

$$precision = \frac{TP}{TP + FP}$$

where TP and FP are true positives and false positives respectively.

The classification *recall* value is defined as

$$recall = \frac{TP}{TP + FN}$$

where FN stands for false negatives. The F_β score is defined as the harmonic mean of *precision* and *recall* values:

$$F_\beta = (1 + \beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall} \tag{3.2}$$

where β states for the relative importance given to precision comparing to recall. In our experiments, we considered $\beta = 1$ (*i.e.* the F_1 score), so:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} = \cdots = \frac{2 * TP}{2 * TP + FP + FN} \tag{3.3}$$

In order to proceed with the comparison between using uniform and non-uniform weights for classifiers, we had to determine the most appropriate weight values for each descriptor. These values have been found with an exhaustive

search by evaluating the performance of the currently analyzed weights combination over the DSO-1 dataset with a 10-fold cross-validation protocol.

To reduce the computational cost, we limited the range of values to be explored for each classifier basing on the results reached for each single descriptor (presented in Table 3.1) giving more importance to LCH descriptor. Table 3.2 reports the used classifiers weights.

Descriptor	**Weight**
ACC	1.10
BIC	1.25
CCV	1
LCH	1.56

Table 3.2: The weights used for each classifier in the non-uniform case.

Experimental results are collected in Table 3.3, for the uniform weights case, and in Table 3.4, for non-uniform case.

Test case	**Train**	**Test**	**Accuracy**	**AUC**	**F-Score**
Test 1	DSO-1	DSO-1	0.82	0.88	0.77
Test 2	DSI-1	DSI-1	0.87	0.92	0.87
Test 3	DSO-1	DSI-1	0.58	0.58	0.62
Test 4	DSI-1	DSO-1	0.62	0.59	0.53

Table 3.3: Performance of face forgery detection module over paired faces using uniform weights.

Test case	**Train**	**Test**	**Accuracy**	**AUC**	**F-Score**
Test 1	DSO-1	DSO-1	0.84	0.90	0.78
Test 2	DSI-1	DSI-1	0.89	0.92	0.89
Test 3	DSO-1	DSI-1	0.59	0.58	0.64
Test 4	DSI-1	DSO-1	0.63	0.60	0.54

Table 3.4: Performance of face forgery detection module over paired faces using non-uniform weights.

Resulting accuracy scores are obtained as average over 5 consecutive runs of

the algorithm. For a better visualization, Figure 3.6 depicts a direct comparison between the accuracy of both experimental results as a bar graph.

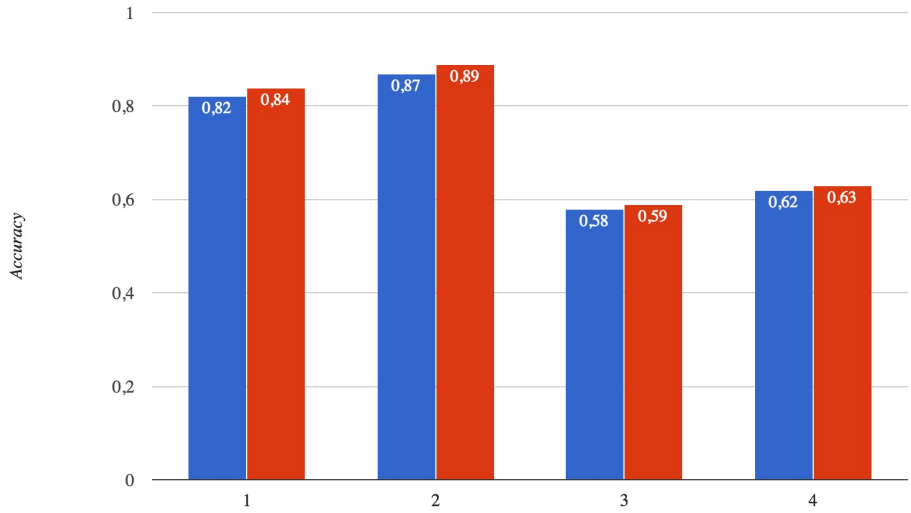

Figure 3.6: Classification accuracies comparison between using uniform and non uniform weights for kNN classifiers over the considered 4 test cases of Table 3.3. The blue bins depict the reached accuracies using uniform weights, reds are for non-uniform case.

These results show that, although we can notice a slight performance improvement in all test cases, the use of non-uniform weights do not affect much the final classification results.

In the following subsections the experimental results for each analyzed test case using non-uniform weights are described. We show results using classical *Receiver Operator Characteristic (ROC)* [12] and *Precision-Recall* curves [8].

3.2.2.1 Performance on DSO-1 dataset

In this experiment, we applied the proposed method for classifying a face pair as fake or real considering only the DSO-1 dataset. We reached an average accuracy of 84% using the 10-fold cross-validation protocol.

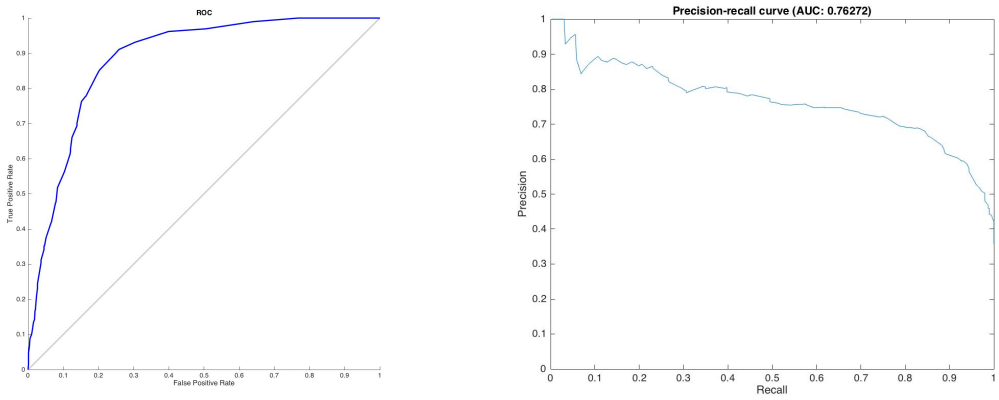

Figure 3.7: ROC curve and Precision-Recall curve for DSO-1 cross-validation classification of paired faces.

3.2.2.2 Performance on DSI-1 dataset

In this experiment we applied the proposed method for classifying a face pair as fake or real considering only the DSI-1 dataset. We reached an average accuracy of 89% using the 10-fold cross-validation protocol.

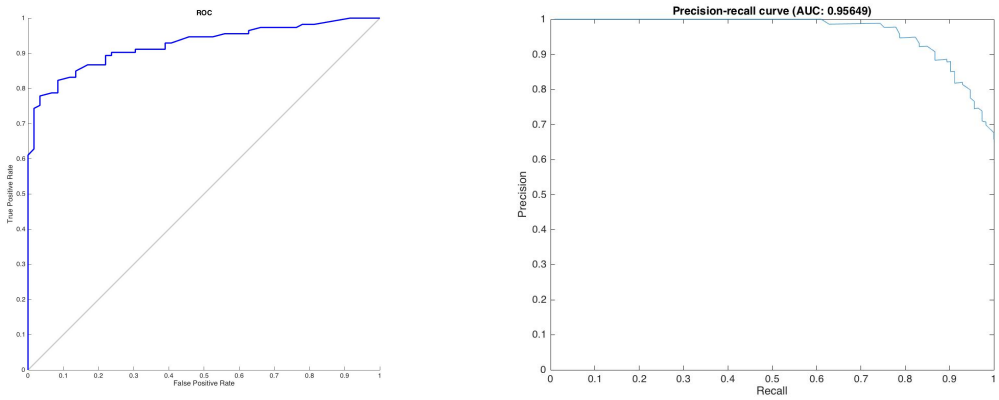

Figure 3.8: ROC curve and Precision-Recall curve for DSI-1 cross-validation classification of paired faces.

3.2.2.3 Cross dataset performance on DSO-1

In this experiment we performed a cross-dataset validation in which we trained our method with images from DSI-1 and test it against images from the DSO-1

dataset. We achieved an average classification accuracy of 63%.

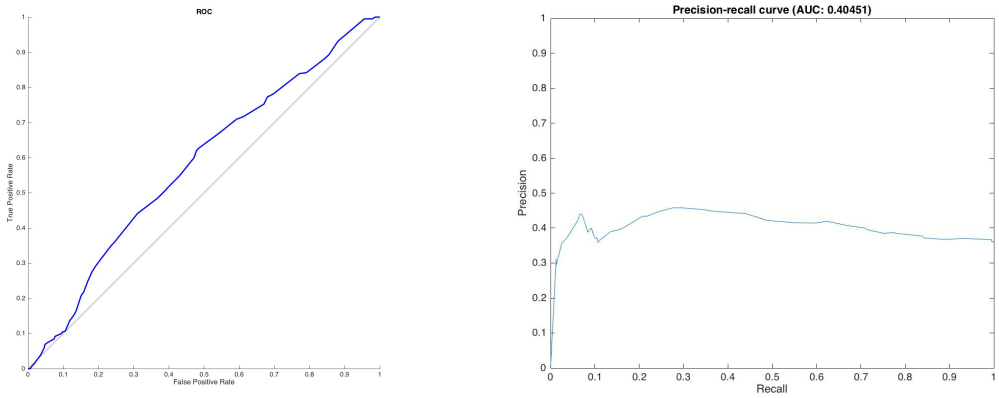

Figure 3.9: ROC curve and Precision-Recall curve for DSO-1 paired faces classification using trained data on DSI-1.

3.2.2.4 Cross dataset performance on DSI-1

In this experiment we performed a cross-dataset validation in which we trained our method with images from DSO-1 and test it against images from the Internet (DSI-1). We achieved an average classification accuracy of 59%.

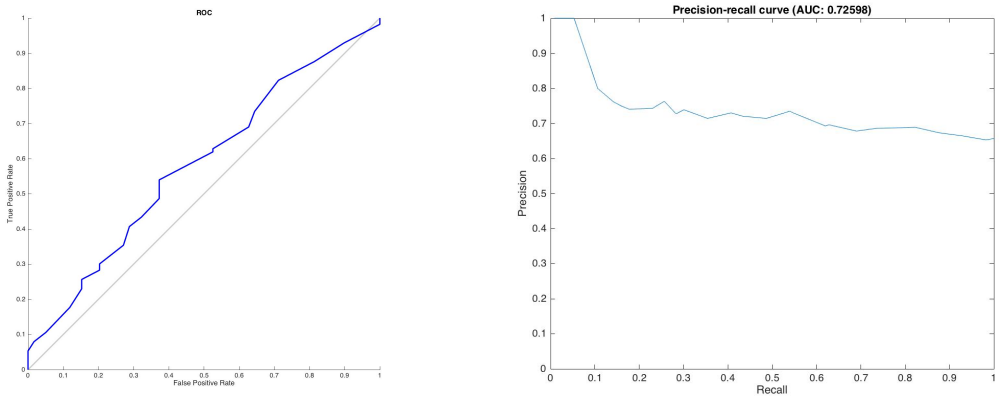

Figure 3.10: ROC curve and Precision-Recall curve for DSI-1 paired faces classification using trained data on DSO-1.

In the two previously considered cases, both of cross dataset classification, the differences between the two dataset compositions greatly influences the final

evaluated performance. Almost all the images in the DSO-1, in fact, present similar lighting conditions. In addition, this dataset contains almost always the same people (and thus the same faces) in every picture. This fact makes the model trained using this dataset less expressive than the one trained using a more heterogeneous set of images, as in the case of DSI-1.

3.2.2.5 Forgery detection performance

In the following experiments, we used the proposed module to detect the face with the highest probability of being the fake face in an image tagged as fake by the classifier.

In this round of experiments, we assume that the input image I has already been classified as fake by the classifier (*i.e.* at least one couple of faces has been classified as fake).

We performed this kind of experiments using both DSO-1 and DSI-1 datasets. For each test in the suite it is reported:

- The training dataset (**Train**)
- The evaluation dataset (**Test**)
- The total number of faces detected in the images (**Faces**)
- The true positive rate (**TPR**)
- The classification accuracy (**Accuracy**)
- The classification precision (**PREC**)
- The classification recall (**REC**)
- The F_1 score (**F-Score**)

Experimental results are collected in Table 3.5.

N.	Train	Test	Faces	PREC	REC	Accuracy	F-Score
1	DSO-1	DSO-1	540	0.58	0.89	0.81	0.64
2	DSI-1	DSI-1	133	0.56	0.95	0.75	0.66
3	DSO-1	DSI-1	540	0.31	0.18	0.63	0.25
4	DSI-1	DSO-1	130	0.34	0.43	0.67	0.37

Table 3.5: Performance of face forgery detection module over single faces using non-uniform weights.

3.3 Regions forgery detection module performance

This section describes the experiments performed to evaluate the regional forgery detection module.

3.3.1 Creating the training set

Since the method is based on the classification of entire bands of images, in order to train the SVM classifiers two different training set with specific requirements were created:

- Each image must have only one spliced band (either horizontal or vertical)
- The spliced region of the image must consist only into the whole band
- The spliced region position in the image must be chosen randomly

Each image in the dataset must be shipped with its spliced band position as groundtruth (for performance evaluation).

For the two new training datasets, the DSO-1 and ColorChecker datasets were used respectively as source. Each generated dataset is split into two subsets containing only vertical or horizontal spliced bands.

The pseudocode used for generating the spliced datasets is proposed in Algorithm 2. That procedure is repeated for both DSO-1 and ColorChecher as

Algorithm 2 Spliced dataset creation algorithm

1: Let $d \in \{vertical, horizontal\}$ a direction
2: Let S be the band size
3: **for** each image $i \in SourceDataset$ **do**
4: Select another image j randomly chosen from the same set with $i \neq j$
5: Resize j to fit i
6: Select a band b of direction d randomly from j
7: Put b in i at the same original position
8: Save i as image
9: **end for**

SourceDataset and for each direction. We called *SplicedDSO* the dataset generated from DSO-1, *SplicedCC* the one from ColorChecker. Both have 200 images vertically and horizontally spliced.

Figure 3.11: Example images from the generated *SplicedCC* dataset. Left with horizontal spliced band, right with vertical.

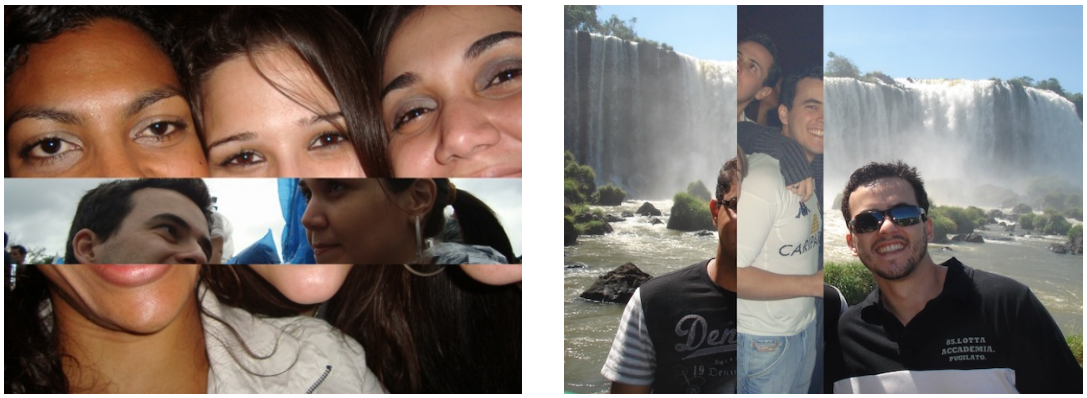

Figure 3.12: Example images from the generated *SplicedDSO* dataset. Left with horizontal spliced band, right with vertical.

Figures 3.11 and 3.12 show two examples of generated images from both datasets.

3.3.2 Evaluating module performance

The proposed method output consists in a *detection map* related to the analyzed image. In order to evaluate the performance of the module, we consider the single pixel classification accuracies. The evaluation dataset for all the experiments is the original DSO-1 dataset (considering only the spliced images).

Given a set of m images as validation test, for each test in the suite:

1. Let I be the i-th analyzed image, a detection map D_i is computed with our module.

2. Let $\mathcal{P}_i$ be the subset of pixels in the image corresponding to the spliced region and let $\mathcal{S}_{\mathcal{P}_i}$ their corresponding values extracted from D. Finally, let N_i be the subset of pixels in the image corresponding to its original part and let $\mathcal{S}_{\mathcal{N}_i}$ be their corresponding values extracted from D_i.

3. Step 1 and 2 are repeated for each image in the validation step, joining the resulting sets of scores:

$$\mathcal{S}_{\mathcal{P}} = \cup_{i=1}^{m} \mathcal{S}_{\mathcal{P}_i}$$

$$\mathcal{S}_{\mathcal{N}} = \cup_{i=1}^{m} \mathcal{S}_{\mathcal{N}_i}$$

where m is the number of images into the validation set.

The two resulting sets of scores, $\mathcal{S}_{\mathcal{P}}$ and $\mathcal{S}_{\mathcal{N}}$ are used for evaluating overall module performances.

The size of the evaluation dataset is limited to 25 images due the required computational time. In the classification stage (in the test cases with a training

phase needed), we used an SVM classifier with an RBF kernel, and classical grid search for adjusting parameters in training samples [1].

In the training process, the images are segmented by non overlapping bands: in this way when dealing with a single image portion (a stripe) all contained pixels are original or modified. Contrary, in the evaluation process, the images are segmented by overlapping stripes. In this way each pixel in the image is classified more than once (depending on the sliding delta size). We considered a delta offset equals to $\frac{1}{4}$ of the band size, so each pixel is evaluated 8 times during all the evaluation process, 4 times as a part of horizontal bands and 4 times as a part of vertical bands.

Experimental results are collected in Table 3.6.

Test case	**Train**	**RC**	**ACC**	**AUC**	**F-Score**
Test 1	-	Median	0.49	0.32	0.25
Test 2	-	Global	0.52	0.40	0.27
Test 3	SplicedCC	Median	0.54	0.53	0.26
Test 4	SplicedCC	Global	0.57	0.57	0.31
Test 5	SplicedDSO	Median	0.53	0.50	0.27
Test 6	SplicedDSO	Global	0.61	0.63	0.33

Table 3.6: Performance of region forgery detection module.

3.3.3 Test cases

The following experiments, collected in Table 3.6, are divided into two categories based on the need of a training process.

3.3.3.1 Performance without training

In the first type of approach, no SVM model is required for classification. Let D be the *detection map* and I be the input image. For each extracted band B:

$$D_{i,j} = D_{i,j} + \sum_{k=1}^{5} d_i$$

for all pixels of D at position (i, j) contained in B, where d_i is the RGB Euclidean distance between the illuminant color of B and the considered reference color.

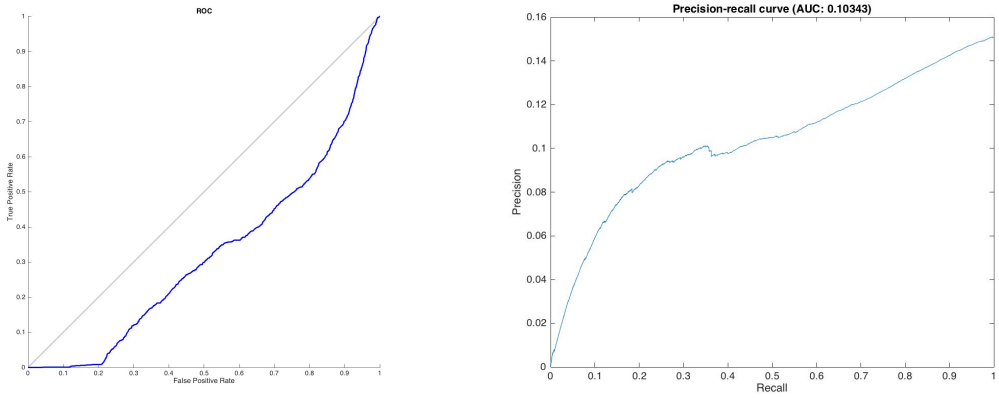

Figure 3.13: ROC curve and Precision-Recall curve for classification without training, using the median reference value.

In these experiments, we applied the proposed method for classifying each pixel in the input image as fake or real considering both median reference color (for each direction), Figure 3.13, and global reference color, Figure 3.14. We reached an average accuracy of 49% in the first case and of 52% in the second case.

3.3.3.2 Performance with SVM training over *SplicedCC*

In the second type of approach, an SVM model is created for classification over a training dataset. For each segmented image band a feature vector is built as described in Eq. (2.2) and classified using the trained model.

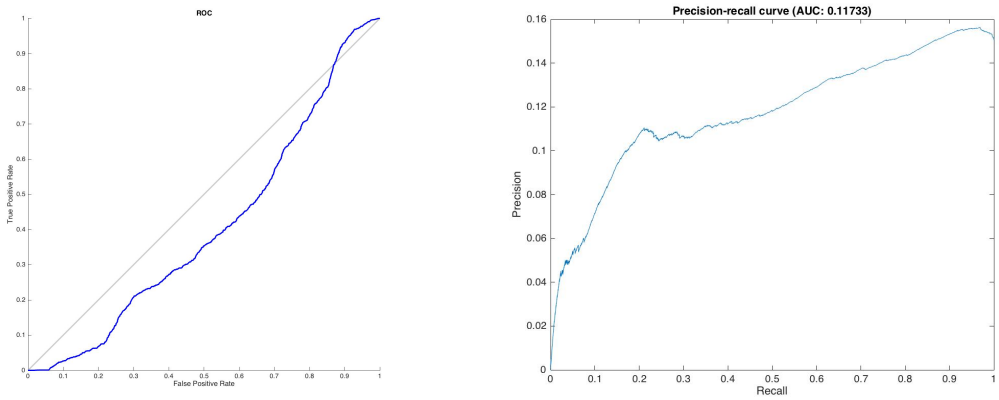

Figure 3.14: ROC curve and Precision-Recall curve for classification without training, using the global reference value.

Let D be the *detection map* and I be the input image. For each extracted band feature vector f_B:

$$D_{i,j} = D_{i,j} + Score(f_B)$$

or all pixels of D at position (i, j) contained in B, where $Score(f_B)$ is the classification output score.

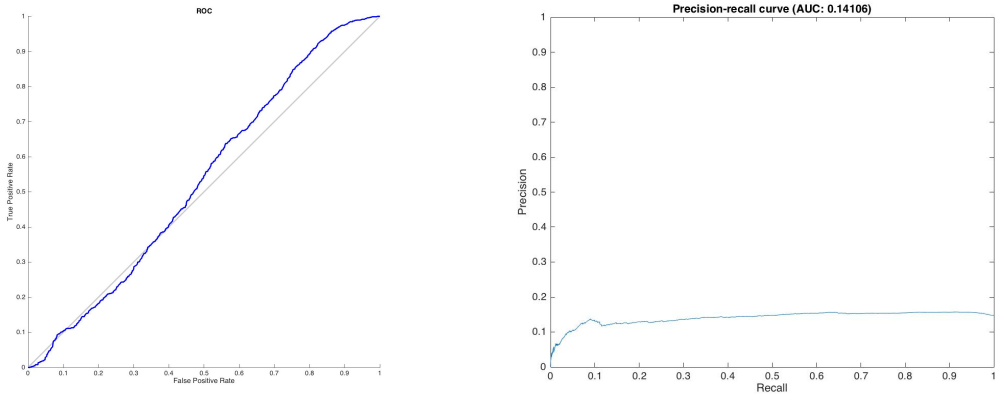

Figure 3.15: ROC curve and Precision-Recall curve for classification with SVM training on *SplicedCC*, using the global reference value.

In these experiments, we applied the proposed method for classifying each pixel in the input image as fake or real considering both median reference color (for each direction), Figure 3.15, and global reference color, Figure 3.16, using

an SVM classifier trained over the *SplicedCC* training dataset. We reached an average accuracy of 54% in the first case and of 57% in the second case.

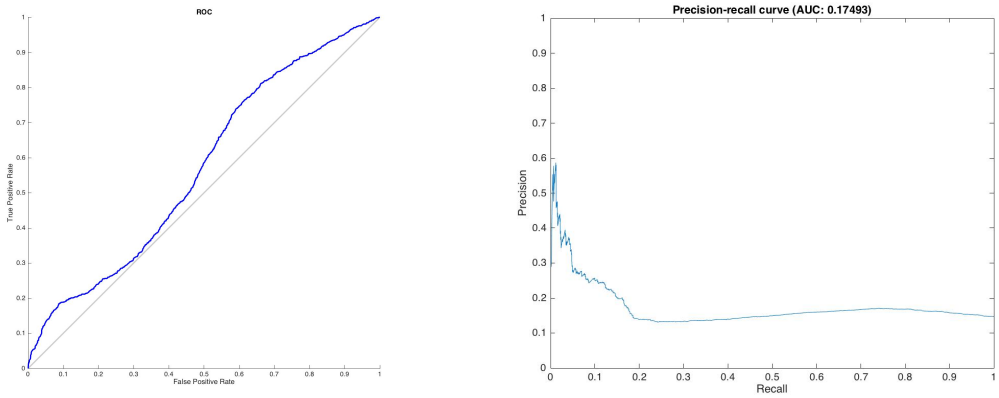

Figure 3.16: ROC curve and Precision-Recall curve for classification with SVM training on *SplicedCC*, using the global reference value.

3.3.3.3 Performance with SVM training over *SplicedDSO*

As in the previous experiment, we used an SVM model trained over a training dataset for classification. Differently from the last case, the classifier is trained over the *SplicedDSO* dataset.

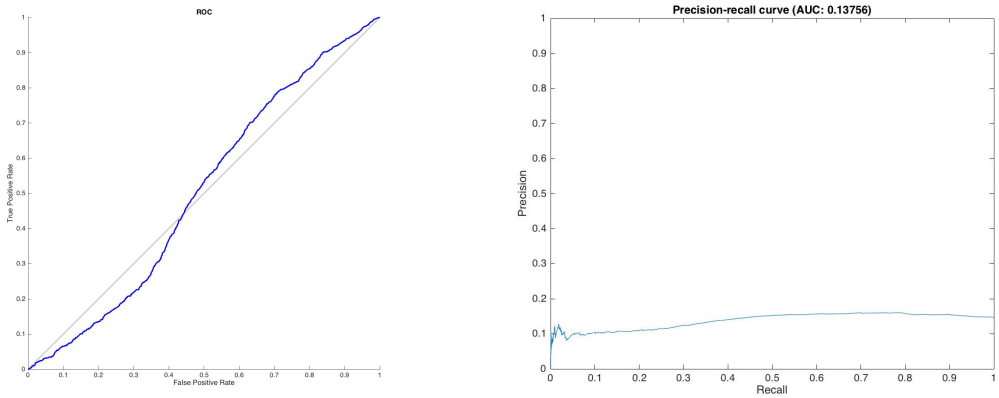

Figure 3.17: ROC curve and Precision-Recall curve for classification with SVM training on *SplicedDSO*, using the global reference value.

Also in this test case, we applied the proposed method for classifying each pixel in the input image as fake or real considering both median reference color

(for each direction), Figure 3.15, and global reference color, Figure 3.16. We reached an average accuracy of 53% in the first case and of 61% in the second case.

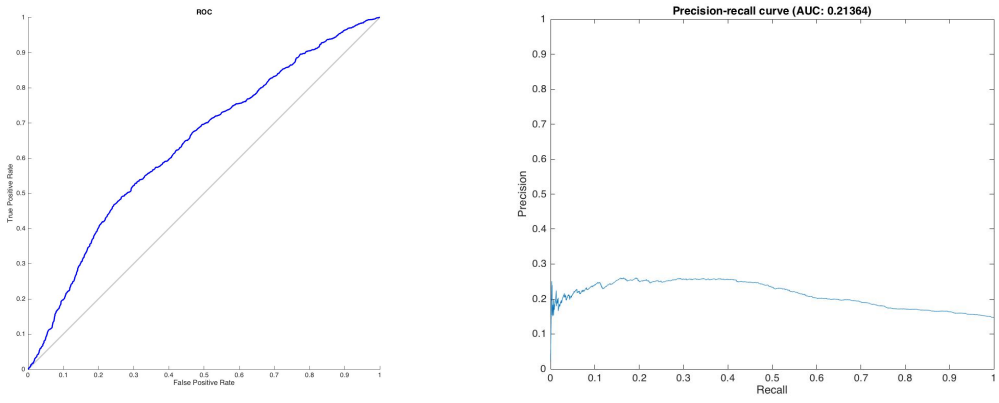

Figure 3.18: ROC curve and Precision-Recall curve for classification with SVM training on *SplicedDSO*, using the global reference value.

3.4 Final remarks

Experimental results presented for the face forgery detection module show that the proposed approach is a good starting point for detecting image composition involving people. Using just simple kNN classifiers and a set of color descriptors over the evaluated image illuminant maps, we achieved an accuracy rate of 81% (with an F_1 score of 64%) in face forgery detection (using cross-validation protocol) and an accuracy rate of 67% (with an F_1 score of 37%) in a cross dataset scenario.

Good results are also achieved over internet images (the DSI-1 dataset), also under cross-database training/testing.

Results shown in Table 3.4 also highlight that the final classification accuracy depends a lot on the kNN models strength: especially in a cross-dataset scenario better results are achieved using a varied training set.

Regarding the second considered module, the more promising results are obtained using a machine learning approach for regional features classification. Table 3.6 shows distinctly that the experiments led using SVM classifiers instead of a simple detection map based on the sum of distances achieved preferable performances. We noted also that using a global image illuminant reference color instead of evaluating the median of the illuminants for each direction is a more suitable approach.

Also in this case, the better the SVM models trained, the higher the classification accuracies reached for the evaluation set. Using the models trained over a set of images closer to those occurring in the test set led to better results, as in the case of the *SplicedDSO* as training set and DSO-1 as test set, which reached the higher accuracy score, 61% of image pixels correctly classified.

In summary, the face forgery detection module reached remarkable results, but it is bound at a specific type of image composition. On the contrary, the regional forgery detection module is a totally blind approach that do not require any assumption on the image content, but we pay this major advantage with a loss of detection accuracy.

Conclusions

Image composition is among the most common types of image manipulation and consists of using parts of two or more images to create a new fake one. In this context, this work has presented a method, composed by two different modules, that relies on illumination inconsistencies for detecting this image forgeries.

In the two different approaches, divided into two modules, proposed in Chapter 2, we analyzed illuminant maps entailing the interaction between the light source and the objects contained in a scene.

The first module, aimed at detecting forgeries involving people, is based on the assumption that similar materials (*e.g.* the human skin) illuminated by a common light source have similar properties in such maps. Features based on color of illuminant maps are now used to describe human face regions and a set of kNN models are trained and used for classification using a fusion technique. However, although image composition involving people is one of the most usual ways of modifying images, other elements can also be inserted into them. To address this issue, we combine the first module with another kind of approach, still based on the illuminant maps analysis, but completely image content independent. This second approach is based on considering a set of different illuminant color estimators (derived from the GGE framework presented in Chapter 1) and a simple image segmentation. Images are divided into two types of stripes, vertical and horizontal and then each stripe is encoded into a 5-dimensional feature vector composed by all the differences between the evaluated illuminant color for the analyzed band and a reference one. Band classification has been made using SVM classifiers.

This face module achieved the most promising results, but it needs some *a priori* knowledge about the content of the image which will be analyzed and, above all, it operates only in presence of human faces. The second approach used has instead brought encouraging results. The method is not reliable in the detection of forgeries due to the large number of false positives obtained in classification, but it can be a starting point for a retrospectively analysis of the image by an expert.

Future developments of this work may include the extension of the first module considering additional and different parts of body (*e.g.*, all skin spots of the human body visible in an image). Given that our method compares skin material, it is feasible to use additional body parts, such as arms and legs, to increase the detection and confidence of the method.

Although promising as forensic evidence, methods that operate on illuminant colors are inherently prone to estimation errors. Thus, we expect that further improvements can be achieved when more advanced illuminant color estimators become available.

Appendix A

Command line tool

In this Appendix some implementation details are presented regarding the source code of the proposed approach. Source code of illuminant maps generation is available online[1].

The source code is divided into three parts:

- The two modules implemented separately (*face-detector* and *region-detector*);
- Illuminant maps estimation using IIC and GGE algorithms;
- Image color descriptor extraction.

All code for the two main modules is written in Python using the open source computer vision library *OpenCV*[2]. OpenCV is an open source computer vision and machine learning software library built to provide a common infrastructure for computer vision applications. The library is written natively in C++, but it has C, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.

Python version used is the *3.4.2* and we used *OpenCV 3.0.0*.

[1]Christian Riess reasearch page for *Scene Illumination as an Indicator of Image Manipulation* - https://www5.cs.fau.de/research/areas/computer-vision/image-forensics/scene-illumination-as-an-indicator-of-image-manipulation

[2]OpenCV, Open Source Computer Vision Library -http://opencv.org

The illuminant maps estimation algorithms implementation and image color descriptor codes are written in C++.

```
ImageSplicingDetection v.0.5.1
usage: main.py
            [-h] [--face-detector] [--region-detector] [--train] [--detect]
            [--evaluate] [--output-mask OUTPUT_MASK] [--display-result]
            [--crossvalidate] [--extract-single-features]
            [--no-extract-features] [--no-extract-maps] [--img IMG]
            [--euclidean-distances] [--heat-map] [--verbose]


optional arguments:
  -h, --help            show this help message and exit
  --face-detector       use face detector
  --region-detector     use region detector
  --train               train the model for further splicing detection
  --detect              detect splice over an image
  --evaluate            evaluate trained models over a set of images
  --output-mask OUTPUT_MASK
                        output mask path
  --display-result      display mask
  --crossvalidate       cross-validate the dataset
  --extract-single-features
                        extract feature vector for a specific image
  --no-extract-features
                        no extract all training images features
  --no-extract-maps     no extract all training images features
  --img IMG             the path of the suspicious image
  --euclidean-distances
                        visualize distances
  --heat-map            display the heat map between GGE and IIC maps
  --verbose             display all messages
```

Bibliography

[1] C. Bishop. Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn. *Springer, New York*, 2007.

[2] P. Blythe and J. Fridrich. Secure digital camera. In *Digital Forensic Research Workshop*, pages 11–13, 2004.

[3] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1 – 26, 1980.

[4] T. Carvalho, F. A. Faria, H. Pedrini, R. d. S. Torres, and A. Rocha. Illuminant-based transformed spaces for image forensics. *IEEE Transactions on Information Forensics and Security*, 11(4):720–733, 2016.

[5] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. *Digital watermarking and steganography*. Morgan Kaufmann, 2007.

[6] I. J. Cox, M. L. Miller, J. A. Bloom, and C. Honsinger. *Digital watermarking*, volume 1558607145. Springer, 2002.

[7] F. C. Crow. Summed-area tables for texture mapping. *ACM SIGGRAPH computer graphics*, 18(3):207–212, 1984.

[8] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM.

[9] Y. Fan, P. Carré, and C. Fernandez-Maloigne. Image splicing detection with local illumination estimation. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2940–2944. IEEE, 2015.

[10] H. Farid. Digital doctoring: can we trust photographs? 2009.

[11] H. Farid. Image forgery detection. *IEEE Signal processing magazine*, 26(2):16–25, 2009.

[12] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[13] G. D. Finlayson and E. Trezzi. Shades of gray and colour constancy. In *Color and Imaging Conference*, volume 2004, pages 37–41. Society for Imaging Science and Technology, 2004.

[14] K. Francis, S. Gholap, and P. Bora. Illuminant colour based image forensics using mismatch in human skin highlights. In *Communications (NCC), 2014 Twentieth National Conference on*, pages 1–6. IEEE, 2014.

[15] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[16] G. L. Friedman. The trustworthy digital camera: Restoring credibility to the photographic image. *IEEE Transactions on consumer electronics*, 39(4):905–910, 1993.

[17] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[18] S. Gholap and P. Bora. Illuminant colour based image forensics. In *TENCON 2008-2008 IEEE Region 10 Conference*, pages 1–5. IEEE, 2008.

[19] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.

[20] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Computer Vision and Pattern Recognition,*

1997. Proceedings., 1997 IEEE Computer Society Conference on, pages 762–768. IEEE, 1997.

[21] L. A. Jeni, J. F. Cohn, and F. De La Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 245–251. IEEE, 2013.

[22] M. K. Johnson and H. Farid. Exposing digital forgeries by detecting inconsistencies in lighting. In *Proceedings of the 7th Workshop on Multimedia and Security*, MM&Sec '05, pages 1–10, New York, NY, USA, 2005. ACM.

[23] S. Katzenbeisser and F. Petitcolas. *Information hiding techniques for steganography and digital watermarking*. Artech house, 2000.

[24] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–I. IEEE, 2002.

[25] B. Mahdian and S. Saic. A bibliography on blind methods for identifying image forgery. *Signal Processing: Image Communication*, 25(6):389–399, 2010.

[26] B. Mazin, J. Delon, and Y. Gousseau. Estimation of illuminants from projections on the planckian locus. *IEEE Transactions on Image Processing*, 24(6):1944–1955, 2015.

[27] C. S. McCamy, H. Marcus, and J. Davidson. A color-rendition chart. *J. App. Photog. Eng*, 2(3):95–99, 1976.

[28] T.-T. Ng, S.-F. Chang, C.-Y. Lin, and Q. Sun. Passive-blind image forensics. *Multimedia Security Technologies for Digital Rights*, 15:383–412, 2006.

[29] T.-T. Ng, S.-F. Chang, and Q. Sun. Blind detection of photomontage using higher order statistics. In *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, volume 5, pages V–V. IEEE, 2004.

[30] N. Nikolaidis and I. Pitas. Copyright protection of images using robust digital signatures. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 4, pages 2168–2171. IEEE, 1996.

[31] Y. Ostrovsky, P. Cavanagh, and P. Sinha. Perceiving illumination inconsistencies in scenes. *Perception*, 34(11):1301–1314, 2005.

[32] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 65–73. ACM, 1997.

[33] O. A. Penatti, E. Valle, and R. d. S. Torres. Comparative study of global color and texture descriptors for web image retrieval. *Journal of Visual Communication and Image Representation*, 23(2):359–380, 2012.

[34] A. Piva. An overview on image forensics. *ISRN Signal Processing*, 2013, 2013.

[35] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13(3):361–396, 2000.

[36] T. Qazi, K. Hayat, S. U. Khan, S. A. Madani, I. A. Khan, J. Kolodziej, H. Li, W. Lin, K. C. Yow, and C.-Z. Xu. Survey on blind image forgery detection. *IET Image Processing*, 7(7):660–670, 2013.

[37] C. Riess and E. Angelopoulou. Scene illumination as an indicator of image manipulation. In *International Workshop on Information Hiding*, pages 66–80. Springer, 2010.

[38] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[39] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein. Vision of the unseen: Current trends and challenges in digital image and video forensics. *ACM Computing Surveys (CSUR)*, 43(4):26, 2011.

[40] C. Springer. Geometry and analysis of projective spacesfreeman. *New York*, 1964.

[41] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 102–109. ACM, 2002.

[42] M. J. Swain and D. H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[43] R. T. Tan, K. Nishino, and K. Ikeuchi. Color constancy through inverse-intensity chromaticity space. *JOSA A*, 21(3):321–334, 2004.

[44] S. Tominaga and B. A. Wandell. Standard surface-reflectance model and illuminant estimation. *JOSA A*, 6(4):576–584, 1989.

[45] G. Upton, B. Fingleton, et al. *Spatial data analysis by example. Volume 1: Point pattern and quantitative data.* John Wiley & Sons Ltd., 1985.

[46] J. Van De Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *IEEE Transactions on image processing*, 16(9):2207–2214, 2007.

[47] T. Van Lanh, K.-S. Chong, S. Emmanuel, and M. S. Kankanhalli. A survey on digital camera image forensic methods. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 16–19. IEEE, 2007.

[48] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[49] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[50] X. Wu and Z. Fang. Image splicing detection using illuminant color inconsistency. In *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on*, pages 600–603. IEEE, 2011.

[51] W. Zhang, X. Cao, J. Zhang, J. Zhu, and P. Wang. Detecting photographic composites using shadows. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1042–1045. IEEE, 2009.

[52] B. B. Zhu, M. D. Swanson, and A. H. Tewfik. When seeing isn't believing [multimedia authentication technologies]. *IEEE Signal Processing Magazine*, 21(2):40–49, 2004.