Authentication of Digital Images and Videos

Gaël Mahfoudi

Thèse
de doctorat
de l'UTT

Gaël MAHFOUDI

Authentication of Digital Images and Videos

Champ disciplinaire :
Sciences pour l'Ingénieur

2021TROY0043 **Année 2021**

utt
UNIVERSITÉ DE TECHNOLOGIE
TROYES

THESE

pour l'obtention du grade de

DOCTEUR
de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

en SCIENCES POUR L'INGENIEUR

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Gaël MAHFOUDI

le 14 décembre 2021

Authentication of Digital Images and Videos

JURY

Mme C. FERNANDEZ-MALOIGNE	PROFESSEURE DES UNIVERSITES	Présidente
M. P. BAS	DIRECTEUR DE RECHERCHE CNRS	Rapporteur
M. C. CHARRIER	MAITRE DE CONFERENCES - HDR	Rapporteur
M. B. EL HASSAN	PROFESSEUR	Examinateur
M. M. PIC	DOCTEUR	Examinateur
M. F. MORAIN-NICOLIER	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. F. RETRAINT	PROFESSEUR DES UNIVERSITES	Directeur de thèse

À ma grand-mère, mes parents, mon frère et ma soeur,

À Marie Marty, ma compagne,

Pour leur amour et leur soutien,

À la mémoire de Jean-Pierre Rigaud, mon grand-père,

Acknowlegment

This thesis has been carried out in partnership with SURYS and the Computer Science and Digital Society (LIST3N) laboratory at the University of Technology of Troyes.

This work has been accomplished under the supervision of M. Florent RETRAINT, M. Frédéric MORAIN-NICOLIER and M. Marc Michel PIC. I would like to express my gratitude to them for their precious advices, support but also their kindness throughout my doctoral project. M. Marc Michel PIC and M. Florent RETRAINT followed me since my master's internship and encourage me to pursue this thesis. I am grateful to M. Marc Michel PIC for the high confidence he has given to me professionally. I would also like to thank M. Frédéric MORAIN-NICOLIER and M. Florent RETRAINT who made me discover and enjoy the field of image processing during my engineer's degree.

I would like to thank M. Jean-Luc DUGELAY with whom I had the pleasure to work.

I would like to thank my coworkers at SURYS for the friendly environment which they have offered me. I would like to express a particular thanks to Amine OUDDAN for his precious support and advice during those three years.

I would like to express my deepest gratitude to M. Patrick BAS and M. Christophe CHARRIER who agreed to review my thesis. I would also like to thank M. Bachar EL HASSAN and Mme. Christine FERNANDEZ-MALOIGNE who accepted to examine this thesis.

Abstract

Digital media are parts of our day-to-day lives. With years of photojournalism, we have been used to consider them as an objective testimony of the truth. But images and video retouching software are becoming increasingly more powerful and easy to use and allow counterfeiters to produce highly realistic image forgery. Consequently, digital media authenticity should not be taken for granted any more. Recent Anti-Money Laundering (AML) regulation introduced the notion of Know Your Customer (KYC) which enforced financial institutions to verify their customer identity. Many institutions prefer to perform this verification remotely relying on a Remote Identity Verification (RIV) system. Such a system relies heavily on both digital images and videos. The authentication of those media is then essential. This thesis focuses on the authentication of images and videos in the context of a RIV system. After formally defining a RIV system, we studied the various attacks that a counterfeiter may perform against it. We attempt to understand the challenges of each of those threats to propose relevant solutions. Our approaches are based on both image processing methods and statistical tests. We also proposed new datasets to encourage research on challenges that are not yet well studied.

Keywords : digital forensic, image processing, statistical tests, image forgery

Résumé

Les médias digitaux font partie de notre vie de tous les jours. Après des années de photojournalisme, nous nous sommes habitués à considérer ces médias comme des témoignages objectifs de la réalité. Cependant les logiciels de retouches d'images et de vidéos deviennent de plus en plus puissants et de plus en plus simples à utiliser, ce qui permet aux contrefacteurs de produire des images falsifiées d'une grande qualité. L'authenticité de ces médias ne peut donc plus être prise pour acquise. Récemment, de nouvelles régulations visant à lutter contre le blanchiment d'argent ont vu le jour. Ces régulations imposent notamment aux institutions financières de vérifier l'identité de leurs clients. Cette vérification est souvent effectuée de manière distantielle au travers d'un Système de Vérification d'Identité à Distance (SVID). Les médias digitaux sont centraux dans de tels systèmes, il est donc essentiel de pouvoir vérifier leurs authenticités. Cette thèse se concentre sur l'authentification des images et vidéos au sein d'un SVID. Suite à la définition formelle d'un tel système, les attaques probables à l'encontre de ceux-ci ont été identifiées. Nous nous sommes efforcés de comprendre les enjeux de ces différentes menaces afin de proposer des solutions adaptées. Nos approches sont basées sur des méthodes de traitement de l'image ou sur des modèles paramétriques. Nous avons aussi proposé de nouvelles bases de données afin d'encourager la recherche sur certains défis spécifiques encore peu étudiés.

Mots clés : criminalistique, traitement d'images, test statistique, falsifications d'images

Contents

List of Figures

LIST OF FIGURES

List of Tables

CHAPTER 1

General Introduction

1.1 Context

Images and videos are parts of our day-to-day lives. Whether we watch a documentary, read a journal or share a memory on social networks we use images or videos to share information. The use of photographs as an information media is not new, in fact, photojournalism can be traced as far back as 1848 with a photograph of the barricades of Paris during the June Days uprising in the French journal "L'illustration". With years of photojournalism, we have been used to seeing photographs as an objective testimony of the truth. As we say, a picture is worth a thousand words.

With the transition from analogue film photography to fully digital photography, those media gain even more attention as they became much easier to share. Most newspapers offer their articles digitally which are easily shared on social networks. At the same time, digital image retouching technologies has come a long way. Nowadays retouching software such as Photoshop or Affinity Photo allows one user to produce extremely realistic image forgery with great ease. The massive use of digital images, the advance of retouching software and the excessive confidence over media's integrity inevitably led to an increased awareness regarding the risk of digital tampering and the spread of fake news.

More recently, new Anti-Money Laundering (AML) regulations have emerged and introduced the notion of Know Your Customer (KYC). The idea behind KYC is to enforce banks to verify their client's identity to prevent money laundering. When the client is a customer and not a company, KYC consist in the verification of an identity document. Like many news media, banks and financial institutions are also increasing their use of digital technologies. Electronic-KYC (eKYC) soon emerged and allows one customer to prove his identity remotely. The idea of eKYC is to let the user send pictures of his document remotely rather than having to present those in person. This process is also known as remote onboarding.

If digital image tampering can help the spread of fake news, it can also have serious implication in remote onboarding systems. One could perform identity theft, create a bank account with fake IDs, etc. It is thus essential to verify the integrity of all media used in a typical remote onboarding scenario.

1.2 Outline

In this thesis we explore various threats against remote onboarding systems and try to propose countermeasures to fight against those.

In chapter 1, we introduce the general context and the motivation behind this thesis. We also present the outline of the thesis and give a summary of all the scientific contributions made during the thesis.

In chapter 2, we go into deeper detail of the current state of the art of digital forensic. We first introduce the complete image formation process and then present the current-art forensic methods. Then we give a detail presentation of a typical remote onboarding system and how a user is usually asked to acquire is ID document. We then present the classical structure of an ID document and the typical frauds we expect. From this, we explain how those frauds relate to the remote onboarding system and what forensic analysis can be performed and at which point.

In chapter 3, we introduce the DEFACTO dataset. We present the context which led to the creation of this dataset and its various objectives. Finally, we present the automatic tampering process developed which allowed us to create the DEFACTO dataset.

In chapter 4, we present our work on Copy-Move detection. We first give a brief introduction to the challenge for copy-move forgery on ID documents. Then we present our method and the results obtained on state of the arts datasets. After, we explain why and how we created a new dataset for copy-move forgery on ID documents. We evaluate several methods on this dataset and show how challenging copy-move forgery detection is on ID documents. We then conclude with a few perspectives on copy-move forgery.

In chapter 5, we follow our work on copy-move forgery by addressing object-removal forgery. We explain briefly how copy-move and object-removal relate and what motivates the need for a method for object-removal. We then present the method we developed and some qualitative results. Finally, we give perspectives on future work to enhance this method.

In chapter 6, we introduce the face morphing attack. We will explain the two main detection strategies. We then present a new method to detect such a forgery

and present various results obtain on two datasets. From those results, we question the applicability of one detection strategy in some contexts.

In chapter 7, we extend our work to videos. We give our motivation for the study of videos rather than images. Then we present the video compression standard H.264 which is widely used. After we present a novel method for double H.264 compression detection and present various theoretical and experimental results.

Finally, in chapter 8, we conclude this thesis and present perspectives and opened challenges.

1.3 Communication

Conference papers

1. **G. Mahfoudi** et al. "DEFACTO: Image and Face Manipulation Dataset". In: *2019 27th European Signal Processing Conference (EUSIPCO)*. 2019, pp. 1–5

2. **G. Mahfoudi** et al. "Copy and Move Forgery Detection Using SIFT and Local Color Dissimilarity Maps". In: *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2019, pp. 1–5

3. M. Pic, **G. Mahfoudi**, and A. Trabelsi. "Remote KYC: Attacks and Counter-Measures". In: *2019 European Intelligence and Security Informatics Conference (EISIC)*. IEEE. 2019, pp. 126–129

4. **G. Mahfoudi** et al. "Object-Removal Forgery Detection through Reflectance Analysis". In: *2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE. 2020, pp. 1–6

5. **G. Mahfoudi** et al. "CMID: A New Dataset for Copy-Move Forgeries on ID Documents". In: *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2021, pp. 3028–3032

Book chapter

1. M. Pic, **G. Mahfoudi**, T. Anis, et al. “Face Manipulation Detection in Remote Operational Systems”. In: *Handbook of Digital Face Manipulation and Detection*. Ed. by R. Christian, T. Ruben, B. Christoph, et al. Springer International Publishing, 2022,

Journal article

1. **G. Mahfoudi** et al. “Statistical H.264 Double Compression Detection Method Based on DCT Coefficients”. In: *IEEE Access* 10 (2022), pp. 4271–4283. DOI: 10.1109/ACCESS.2022.3140588

Patents

1. **G. Mahfoudi** et al. *Method for processing a candidate image*. French Patent FR2000395. Jan. 2020

2. M. M. Pic, A. Ouddan, **G. Mahfoudi**, et al. *Image processing method for an identity document*. French Patent FR19004228. Feb. 2019,

3. **G. Mahfoudi**. *Procédé de vérification d’une image numérique*. French Patent FR2012868. Oct. 2020

4. M. M. Pic, **G. Mahfoudi**, et al. *Procédé d’authentification d’un élément optiquement variable*. French Patent FR2006419. Feb. 2020,

5. **G. Mahfoudi**, **M. M. Pic**, and A. Trabelsi. *Method for automatically detecting facial impersonation*. French Patent FR3088457, World Patent WO2020099400. May 2020

6. **G. Mahfoudi** and **A. Ouddan**. *Digital image processing method*. French Patent FR3091610B1. May 2020

CHAPTER 2

Overview on Image forgery detection

2.1 Introduction

As mentioned in chapter 1, this thesis focuses on the detection of image forgery. In particular, we are interested in the detection of forgeries in distant identity verification processes also called Remote Identity Verification (RIV) systems.

Because those systems rely heavily on digital media, it is essential to be able to authenticate those. In fact, RIV systems are usually just a small part of a much bigger system. Often they allow the user to create a digital identity which he will use to authenticate himself and log into different services. Subsequent service security thus relies only on the robustness of the RIV systems.

We will start by introducing the basic architecture of a typical RIV system. From there, we will introduce two common scenarios for the acquisition of the user ID documents and present how the recent French PVID regulation [14] will affect the overall architecture of a RIV system.

After, we will take a closer look at the typical structure of an ID document. We will briefly introduce the classic physical tampering methods of such documents before explaining how we expect digital tampering may be preferred over those methods. Then we will present the part of a document that will most likely be tampered.

We will then give an overview of digital forgery in general. We will go through a formal definition of what we consider to be a forgery (as opposed to the enhancement) and present the main categories of forgeries. Then we will explain how each of those forgeries relates to RIV systems and ID documents.

Then, we will give a brief overview of digital image forensic. Because each chapter of this thesis address very different topics, we will give a more detailed state of the art on a per chapter basis. This overview will introduce key concepts and give a few examples for each. We will first present the image acquisition pipeline. From there we will differentiate passive and active image forgery detection. Then we will introduce the common passive image forgery detection approaches.

Finally, we will conclude by explaining the main objectives of this thesis and the motivations behind those.

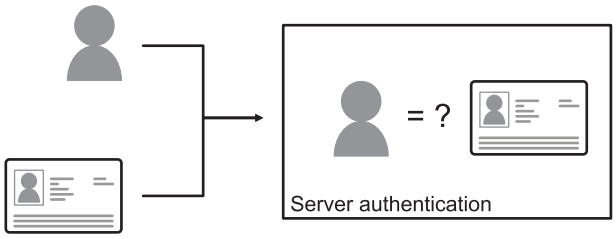


Figure 2.1 : Basic Remote Identity Verification System

2.2 Remote Identity Verification systems

In the next sections, we will describe the typical architecture of RIV systems and describe the two main scenarios regarding the acquisition device. We will then briefly present the new PVID regulation and how it will affect the architecture of RIV systems.

2.2.1 General Overview

In its simplest form, a Remote Identity Verification system consists of two input streams and one verification system as seen in Fig. 2.1.

One of the inputs is an acquisition of the user ID documents. In most cases, RIV systems ask the user to present an official ID documents because they have been issued by a trusted entity and are equipped with known security features which can be used to perform the authentication.

But ID documents on their own cannot be used to enrol a person. In fact, we have no guarantee that the person who acquired the image of the document is the rightful owner of this document. This is where the second input stream comes into play. The RIV system needs a proof that the distant user is the rightful owner of the document. To do so, the system will typically ask the user to send a biometric proof that will link the user to his document. The simplest solution that comes to mind is a picture or a video of the user face as this can be directly compared to the portrait photo on the document. But it is worth noticing that fingerprints or iris could also be used as they are present in biometric passports. Unfortunately, that

biometrics are stored in the passport chip which requires the user to have an NFC capable smartphone and the RIV provider to have an Extended Access Control (EAC) [15]. Also, fingerprint and iris sensors are not widely used by smartphone manufacturers and when they are present software editing companies often have almost no control on those. A RIV system based on that biometrics would limit their client base to people having NFC capable sensors and a biometric passport. For these reasons, the face is often preferred over other biometry as it requires a simple RGB camera and any ID documents. In the rest of the thesis, we will thus assume that face biometry is used.

Once those two streams are acquired, they are typically sent to a distant server who will perform a few verification. This server will mainly try to answer two questions. Is the document authentic ? And is the person the rightful owner of the document ? To answer the first question, the verification system will first need to determine what document has been presented. Is it a French ID card ? Is it a German passport ? The system will only then check various known security features for this specific document such as variables inks, holograms, microprinting, etc. Once the document is authenticated, the information (portrait, last name, first name, date of birth ...) can be extracted and considered safe.

To answer the second question, the system must verify mainly two things. Is the person really behind the camera ? And does the person match the portrait of the document ? The first interrogation is often referred to as Liveness detection. The objective is to verify if the user acquired a living person rather than a printed photograph or screen for instance. Once the liveness detection is performed, a face-recognition software is used to compare the extracted portrait and the acquired user face. If they are matched, the RIV is successful and the person is enrolled in the system. One of the pictures will then be stored to allow re-authentication later on.

2.2.2 Controlled Acquisition Device

As we explain, the RIV system acquires two streams. The person's face acquisition is always performed live as some liveness detection methods will be used. Because this requires to develop some user interface which performs this acquisition, RIV

providers often include the document acquisition in that interface. We will refer to this use cases as the controlled acquisition device scenario.

In that case the RIV provider has some control over the acquisition device. This is convenient for many reasons. One of which is the ability of the RIV provider to give precise instruction to the user. This allows the RIV provider to avoid receiving poor quality pictures. Initially, RIV providers implemented those users guided acquisition processes to reduce the rejection rates of the documents. The user is typically asked to avoid blurry images, glares on the documents and so on. Having some control over the acquisition device also allows the RIV provider to inject some knowledge in the digital media. He can decide to apply a watermark, compute a cryptographic signature, choose a specific file format, etc.

We see that the control of the acquisition device can greatly help to authenticate the document in later stages. Even though the acquisition is controlled, we must recall that this control can be limited. The device could be provided by the RIV provider himself in which case he has a full control over it and can have a high confidence in the acquired media. But the RIV provider may perform the acquisition through the user's terminal. In that case, the RIV provider would require the user to install an application on his device which would be responsible for performing the acquisition. Even if we assume this application to be perfectly secure, we understand that it may not have complete control over the device sensors. Moreover, the device itself could be compromised in the first place.

With those facts in mind, we understand that we should not assume the media coming from the device cannot be considered authentic just because the said device is controlled. But this may allow the RIV provider to embed enough information in the media to prevent Man in the Middle attacks between the device and the verification server.

2.2.3 Uncontrolled acquisition device

Even though it is preferable to have control over the acquisition media, some RIV providers used to allow the user to send a previously acquired media.

Earlier Identity Verification systems required the user to physically come to an administration where an employee would perform the acquisition (generally a scan)

for him. The authenticity of the digital acquisition was then implicitly assumed and the Identity Verification system would only look for physical tampering of the document.

Those Identity Checks were at first directly used remotely. At first, they would require the user to send a scanned version of his document and then allowed photographs of the documents. A counterfeiter would then have all the time he needs to perform a digital forgery. This makes the authentication process much harder.

This solution was acceptable at first as it allows the RIV provider to check for physical tampering if the quality of the picture is good enough. Unfortunately, it leaves too many opportunities for the counterfeiter to perform a good quality forgery.

2.2.4 Remote Identity Verification Provider regulation (PVID)

With the COVID-19 pandemic, we saw an increasing use of RIV systems. Many companies started developing RIV services. Some reused older identity verification system and implemented uncontrolled acquisition device scenarios whereas others started implementing controlled acquisition device scenarios.

With the increasing number of RIV providers, all with very different architectures and procedures, the French National Cyber Security Agency (ANSSI) quickly saw the need to define a set of security requirements to limit frauds against RIV systems. This led to the writing of the PVID regulation [14] in late 2020. The first applicable version was released on the 1st of March of 2021.

The PVID regulation first give formal definition of a RIV system which is mostly similar to the definition given in the section 2.2.1.

The most notable element of this regulation is that it acknowledges the risk of digital tampering of the document. A direct consequence of that is that it imposes RIV systems to operate in the controlled acquisition device scenario. This in fact seems necessary to fight against digital frauds.

The PVID regulation also imposes both the acquisition of the ID document and

the face to be videos and not images. Finally, the RIV system must authenticate the ID document, perform the liveness detection and compare the document and with the person. It is worth noticing that the first version of the PVID regulation requires the RIV system to perform both a human and a machine verification.

2.3 ID Documents Tampering

Here we will first introduce the overall structure of ID documents. Then we will present the common fraud categories on such documents. We will then give a brief overview the common physical forgeries and then move on to digital tampering on which we will focus our attention in the rest of the thesis.

2.3.1 ID Documents Structure

An identity document is any document that is used to prove someone's identity. There exist multiple types of identity documents such as passports, national identity cards, driving licences. Those documents are issued by the states.

To facilitate the use of one ID documents across different states, the ISO/IEC 7810 [16] and the ICAO Doc 9303 [17] formalise various aspects of ID documents.

In particular, we can distinguish three major zones present in every ID documents which may be targeted by a counterfeiter. The first one is the identity picture. This is a portrait of the document owner which is used to identify him. The portrait must follow a set of rules described by the International Civil Aviation Organization (ICAO) such as the inter-eye distance, face angle, exposure. The second zone of interest is called the Variable Information Zone (VIZ). It contains all the information about the owner such as the first name, last name. Finally, we have the Machine Readable Zone (MRZ). The MRZ is a summary of the information contained in the VIZ and is meant to be automatically read by a machine. It also contains a checksum which allows confirming that it was read properly.

Various size or templates can be used for the documents. For example, the French ID card used to have the ID2 format with the photo, the VIZ and the MRZ on the front. But since the 15^{th} of March 2021, it changed to the ID1 format with the photo and the VIZ on the front but with the MRZ on the back.

Figure 2.2 : Microprintings on the French driver licence

2.3.2 Security Features

At first, the document contains no information. We call it a blank document. A blank document usually already contains a lot of security features to prevent frauds.

Common security features on the blank documents are microprintings which are very detailed elements printed on the documents. Microprintings are often used in the background of the document as seen in Fig. 2.2.

Another common security feature is the presence of variable inks. Those inks change colours depending on the viewing angle.

Once the person information is acquired, they are printed onto the blank document. We call this the customisation of the document. Other security features are often added after the customisation.

One common example is the addition of a holographic laminate on top of the document. Much like the variable ink, a holographic laminate will look different depending on the viewing angle. An example of the varying nature of a hologram is given in Fig. 2.3.

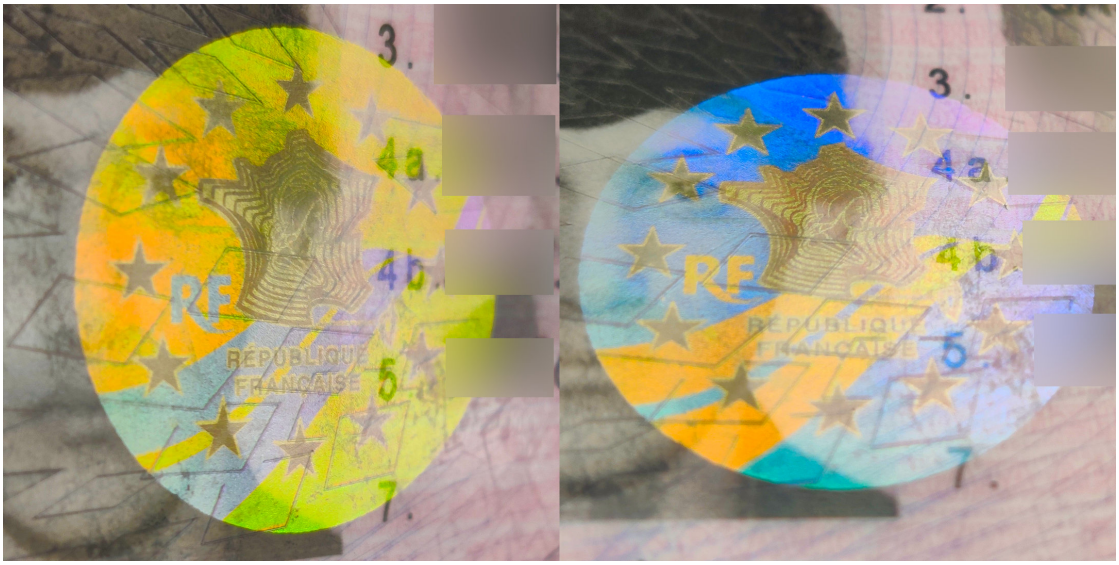


Figure 2.3 : Hologram on the French driving licence from two different viewpoints

2.3.3 Fraud Categories

The basic issuing process of an ID document can be summarised as follows; An individual first request a new ID document. A blank document is thus customised with personal data and then secured with additional security features. Once this process is done, the finished document is issued to the individual.

This process allows mainly three categories of document fraud [18]. The first category is often called blank stolen documents. In that case, the counterfeiter was able to steal blank documents prior to the customisation. Assuming that he also knows the exact customisation process and that he is also able to add the final security features, he would theoretically be able to produce a completely legitimate fraudulent ID that would be undetectable. Nowadays, this kind of fraud can reasonably be considered impossible.

Another approach is to try to completely recreate the ID document. Which is called counterfeiting. Similarly to the blank stolen document fraud, the counterfeiter would need to have both the knowledge and the machinery to produce a convincing fraudulent ID. This makes the production of a convincing counterfeit extremely hard. In general, counterfeiters will produce bad counterfeit and sold

them to naive peoples.

In this thesis we will be mostly interested in the last category of fraud, document forgeries. A document forgery is the alteration of one or many parts of an ID document. In that case the counterfeiter starts from an issued ID documents which contain every security elements and try to change one or many fields without degrading the security features. In that case, less knowledge about the production of the document is needed which make this kind of fraud more accessible.

2.3.4 Physical and digital tampering

In the case of a RIV system, the document tampering can be performed in two different ways.

The first approach is to tamper the document physically. In that case, the counterfeiter will directly alter the document. Typically, he will try to alter the photo or some information in the VIZ and the MRZ. For example, to perform an identity theft, the counterfeiter would replace the photo of a stolen document but leave the information in the VIZ intact.

Such physical tampering can be extremely difficult to make and requires the counterfeiter to be highly skilled. He must have excellent knowledge about the various security features present in the document and also be aware of specific production details such as the printing techniques used for the text, the photo. A well-executed forgery can be extremely difficult to detect and specific equipment may be required to perform the detection. One would look for inconsistent use of printing techniques, visible degradation due to the detachment of the hologram laminate, etc.

Because such tampering requires the help of a skilled counterfeiter, it usually cost a lot. It thus makes more sense if the tampered document is intended to be used to cross the border or to be presented during a police control for instance.

To attack a RIV system, a simple and more accessible digital tampering might be preferred. Much like a physical tampering, a digitally altered document might be used to perform an identity theft or just to modify a few information (date of birth, end of validity, etc.).

Unlike the physical tampering, the counterfeiter does not need to be particu-

larly skilled. Because of some possible acquisition pipeline (smartphone cameras, webcams) and the various compression applied to the media, it is unlikely that the verification system will be able to verify fine scaled details such as the printing techniques small degradation.

One concern regarding digital tampering is the possibility of one counterfeiter to develop an automatic tampering software similarly to what happened with the deepfakes. This could allow anyone to produce extremely realistic forgeries.

In this thesis we will focus only on the digital tampering of the ID documents which we think are one of the main threats against RIV systems.

2.4 Digital forgeries

We gave a brief overview on RIV systems. We saw that a RIV system must be able to authenticate digital media to ensure its reliability. For a RIV system, digital image and video forgeries are a dangerous threat.

In this section, we will give an overview of the various images and video forgery categories. Then we will show how each of those relates to RIV systems.

2.4.1 Image and Video Forgeries

First we will define an image forgery as any local alteration that changes the semantic of a given image. In opposition, we will call a global modification that does not change the semantic an enhancement. For instance, a global adjustment of the contrasts would be called an enhancement whereas a local colour change will be considered as a forgery.

A video can be seen as a three-dimensional image where the third dimension is the time. A video forgery consists of a local alteration that changes the semantic of the video either spatially (x and y dimension) or temporally (time dimension). We will thus talk about Spatial and Temporal video forgeries.

Spatial video forgeries can be seen as simple image forgeries applied to multiple frames. We will consider four main categories [19, 20, 21] of image and spatial video forgeries which we will describe in the next sections.

Figure 2.4 : From left to right : Original image, splicing without colour adjustment, splicing with colour adjustment

In terms of detection, any temporal forgery can be grouped in two categories [21]: frame deletion and frame insertion, which we will briefly describe in afterward.

Splicing forgeries

The first and best known forgery is called a Splicing. A splicing forgery involves at least two images which we will call the source images and the target image.

The splicing consists of the insertion of one or many elements from the source images into the target image. The inserted elements may then be further modified to fit properly. For example, the counterfeiter may want to adjust the luminosity, the sharpness and so on.

This is a very common forgery that may be used in a wide range of applications. It is worth noticing that it is by far the most elaborate forgery that one can make. In fact, to produce a convincing splicing, the counterfeiter must be able to match the lighting, the viewing angle of the inserted element, it's sharpness, etc. Even the slightest mismatch can make the splicing looks completely fake. An example is given in Fig. 2.4 where we show the same splicing of a person in a scene before and after adjusting the luminosity and colours.

Figure 2.5 : From left to right : Original image, various tampering using Copy-Move

Because a splicing forgery requires many adjustments in order to look realistic, it usually leaves more traces which arguably makes it easier to detect than other forgery types.

Copy-Move forgeries

As opposed to splicing forgeries, copy-move only involve one image.

In a copy-move forgery, one region (the source) is copied and paste onto another region (the target). The duplicated element can be translated, rotated, scaled or transformed. As for splicing, the duplicated element may be modified to fit better in the scene.

A copy-move forgery is usually much easier to create than a splicing forgery. Even though it seems quite limited, it is actually quite versatile. One can use a copy-move to fool on a quantity, to hide an element, or to alter a text field. Such forgeries can be seen in Fig. 2.5.

Because the duplicated element comes from the same image, it generally requires less modification to create a convincing forgery. Copy-move will thus usually produce fewer detectable artefact than splicing.

Object-Removal Forgeries

As for copy-move, object-removal only involves one image. Object-removal consists of the suppression of one element from an image.

Such forgery can be performed using a copy-move but there exists other methods to create this forgery. In particular, an object removal is often performed using an inpainting algorithm. There are many types of inpainting techniques [22] such as exemplar-based inpainting, diffusion-based, deep learning based. All those algorithms are used to fill a given area in an image using only information present in that image. Exemplar-based method for example progressively fill the area by copying and pasting extremely small patches. Diffusion-based, on the other hand, propagate the information at the boundary of the area to fill.

An object-removal forgery is extremely simple to create. In particular with modern photo retouching software such as Photoshop or Affinity photo. An example of object removal is given in Fig. 2.6 which required only click in affinity photo.

When the object to remove is on a relatively simple background, the results will most likely be impressively realistic. As for copy-move, object removal often leaves fewer artefact than splicing forgeries.

Face Manipulations

A last forgery category that we will consider is face manipulations. It could be argued that those are just splicing forgeries applied to facial images. And in fact, some face manipulations are effectively simply splicing.

But because those are really specific to facial images, they allow the counterfeiter to use techniques that goes beyond a simple splicing so we argue that they are a kind a forgery on their own.

The simplest and best known facial manipulation is called a face swap. A face swap is a simple splicing of one face onto another. Earlier face swap would only work for static images. With the advance of deep neural networks, it is now possible to produce very realistic face swapping even on videos effortlessly. Those dynamics face swapping methods are often called DeepFakes. They generally perform the swapping by using a deep neural network to synthesise a fake face in place of the

Figure 2.6 : From left to right : Original image, Object-removal forgery

genuine one hence the name DeepFakes.

More recently another approach to perform a face swapping as gain a lot of attention. Instead of replacing the face of a person A by the face of a person B in a video, it was proposed to use a video of the person B and modify it according to a video of the person A. Such forgery is called a face reenactment as the acting of the person A is applied to the video of B rather than superimposing the face of B onto A.

Another facial forgery is called de-identification. The objective is to remove every element of a facial image that would allow a Face Recognition Software (FRS) to operate properly.

A last common facial manipulation is called face morphing. Face morphing can be seen as a generalisation of a simple face swapping. Having a person A and a person B, a Face Morphing consists in the mix of the two faces. This results in a synthetic face which shares the biometry of both A and B. In chapter 6 we will describe more deeply this kind of forgery.

Frame Deletion

Frame deletion is specific to video forgeries. As the name implies, it consists of the suppression of one or many frames. The deleted part can be anywhere in the video.

It is typically use hide parts of a video. For example, a counterfeiter may want to tamper a surveillance video in which he appears. He would thus remove every frame in which he is visible.

Frame Insertion

Frame insertion, on the other hand, consists of the addition of new frames inside a video. Those new frame can come from the same video or another.

The primary use for such a forgery is to create a video composite of multiple sequence. It could also be used alongside a frame deletion forgery to fill the removed portion of the video.

2.4.2 Application in RIV system

We will now discuss how those attacks relates to a typical RIV system.

To begin with, a counterfeiter could have three main objectives while attacking a RIV. A first goal would be to perform an identity theft. When doing an identity theft, the counterfeiter wants to use someone's else personal information. Another objective maybe to access some service anonymously. In such case, he may provide a completely fake identity. Finally, he might simply want to alter a few information on his ID.

Depending on his objectives, the counterfeiter will tamper specific parts of the document or target a specific part of the RIV process. Consequently, he will also use specific forgery categories.

During the RIV process, we except the forgeries to targets three specific things. It is likely that either the photo on the ID document or the live acquisition of the face will be tampered. And we also expect the information on the document to be targeted. In the next sections, we will explain how specific forgery categories would be applied to tamper those various elements.

Figure 2.7 : From left to right : Original image, Complete replacement of the photo, Face swapping

Alteration of the ID Documents Photo

As we mentioned, the photo present on the ID document is a probable target for the counterfeiter as it bears a lot of information about the user. And how we explain, face biometry plays a central role in a RIV system.

Consequently, the photo on the ID document is a vector for many attacks on the system. For example, to perform an identity theft, the counterfeiter will most likely steal an ID document. Because the RIV system makes a comparison of the photo and the user, the counterfeiter must alter the ID document to pass this verification. To perform such an attack, the easiest way would be to perform a simple splicing by replacing the complete photograph. Such forgeries can be seen on Fig. 2.7. A more elaborate forgery would consist of a face swap. In that case, only the inner part of the face is tampered which leaves fewer traces of forgery as in Fig. 2.7.

The photo may be tampered similarly to perform a de-identification [23, 24, 25]. The counterfeiter may in fact share his real information but may not be willing to share his biometry. In such case, the live acquisition would also have to be tampered.

Another attack that will be explored more deeply in chapter 6 is called Face Morphing. When performing a Face Morphing, the counterfeiter will replace the ID document photo by a synthetic face. His goal will be to create a document that could be shared by two people. This would allow two people to access services using the same digital identity.

Figure 2.8 : From left to right : Original document, Document with first and last name tampered

Alteration of the ID documents variable fields

The counterfeiter may not necessarily try to fake his biometry, in fact, he may use his own document on which he would like to change only a few information.

The motivation for such forgery could be multiple such as hiding his name, faking his date of birth or use an expired document.

In those cases, he would not need to alter either the photo on the document nor the acquisition of his face but would rather have to tamper the text fields (or variable fields) of the document.

The most straightforward approach to perform such a forgery is to perform a copy-move. This is in fact the easiest and most realistic way to proceed. Because ID documents use very specific fonts, it is usually much easier to reuse letters already present in the document to alter some fields. An example is given in Fig. 2.8. It can be seen that those forgeries are barely visible if not invisible without telling where they are located.

Alongside a copy-move, an object-removal might be needed to erase previous text fields.

While Copy-Move is the preferred choice, the counterfeiter may not find the needed letters. In such case, he would splice the new text field instead.

Alteration of the Face Acquisition

Lastly, the face acquisition may be the target of several attacks. The objectives would then be very similar to the ID documents photo scenario but the approach would be very different.

Because the acquisition of the face will undergo a variety of liveness detection test and challenges, a simple static face swap will not be enough. A more dynamic approach is needed and the counterfeiter might thus use more advance forgery methods.

For such forgeries, realtime face reenactment methods are the most appropriate. Those forgeries seemed hypothetical at first, because of the complexity of older face reenactment algorithms and also because of the resource needed to run such algorithms. But recent advances of deep learning based methods made it possible for anyone to create a convincing realtime face reenactment forgery. Freely available tools are now accessible to anyone to create such forgeries even on a modest laptop. While complete solutions are not yet available on smartphones, there already exists tools that allow 3D face landmarks detection in a web browser that works in realtime even on a smartphone [26, 27]. We thus can reasonably expect to see convincing 3D face swapping tools to be available on smartphones in the next future.

2.5 Overview on Image Forgery Detection

In this section we will give a general overview of digital image forgery detection. A more thorough state of the art will be presented on a per chapter basis as each chapter addresses very different topics.

We will start by introducing the image formation pipeline by dividing it into three main steps. We will first briefly introduce the hardware part of an acquisition device and how it impacts the raw image formation. Then we will present the various post-processes which are included in the software of the acquisition device. Finally, we will talk about the compression stage which is necessary to reduce the information needed to store the digital media.

After, we introduce the common classification of image forgery detection algorithms. From this we will derive three categories which differ based on the studied artefacts.

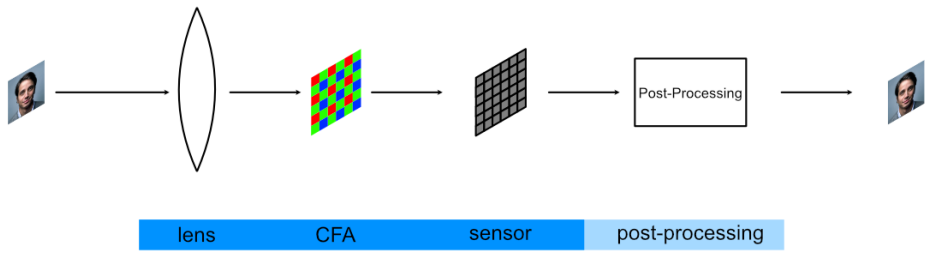


Figure 2.9 : Image Formation Pipeline

2.5.1 Image Formation Pipeline

This section will introduce the image formation pipeline. This pipeline can be seen in its simplest form in Fig. 2.9.

Such an overview is necessary to understand the underlying assumption of the various image forgery detection algorithms. In the later section, we will also see how some assumption about this pipeline may or may not hold in the case of a RIV system.

Acquisition Device

We will start by giving an overview of the hardware of a modern camera. The hardware consists of three main parts [28] (the lens, the Colour Filter Array and the sensor) which can be seen in Fig. 2.9.

A digital camera acquired the light coming from the scene. Those rays of light first go through the lens. The lens redirect the beams coming from the scene to the sensor. The lens is thus responsible for the field of view of the camera but also partially responsible for the amount of light that can reach the sensor. The amount of light reaching the sensor is also controlled by the shutter speed. When taking a picture, the shutter open and let the light go through the lens and to the sensor during a specific amount of time and then closes back to block the light.

Once the light has gone through the lens, it is filtered by the Colour Filter Array (CFA). The CFA physically filter the red, green and blue light before being finally captured by the sensor.

The light received light by the sensor is subject to two perturbations. The first one is inherent to the physic of light and is called the shot noise. The shot noise is due to the fluctuation of the number of photons reaching each individual pixel of the sensor. Another perturbation is due to the CFA. Because the manufacturing of this filter cannot be perfect, small differences in pixel intensity may be observed for nearby pixels. Those imperfections produce a fixed noise-like pattern called the Photo Response Non Uniformity (PRNU) which is unique for each camera. This pattern will be present for every image taken by the same device.

The sensor counts the number of photos reaching each individual pixel. This information is then electrically amplified to map a given photon count to a given pixel intensity. The mapping is usually called the Camera Response Function (CRF) [29]. The CRF varies from one camera manufacturer to the other. During this amplification two perturbations occur. The first one is called the dark current noise. Dark currents are small current fluctuation presents in every electronic device. In the case of a camera sensor, the dark current gets amplified with the remaining of the information. The other source of noise is called the readout noise and is due to random errors while reading the pixel information from the sensor.

At the end of the acquisition of an image, the RAW pixel data is thus affected by four different sources of noise. We will see later that those can be used to authenticate a digital image.

Post-processes

Once the RAW image data has been acquired, it must go through a few post-processes in order to be shown as a regular RGB image on a screen.

First of all, the image must first be demosaiced. As we mentioned, the light goes through the CFA before reaching the sensor. The CFA is used to separate the light into three distinct colours i.e. red, green and blue.

The CFA filter the light following a specific pattern (e.g. the Bayer pattern). The RAW image is thus a single image of size $N \times M$ where each pixel record the

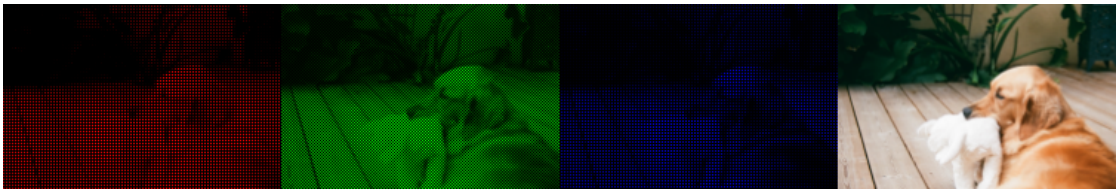


Figure 2.10 : From left to right : Mosaiced red channel, Mosaiced green channel, Mosaiced blue channel, Demosaiced image

intensity of one of the three colours.

A demosaicing algorithm is used to infer three colour channels of size $N \times M$ from this single RAW image. Basically, the demosaicing algorithm first extract each individual colour information from the single-channel RAW image and then interpolate the missing pixels to produce three $N \times M$ images. On Fig. 2.10 we can see the three channels extracted from the single-channel raw image. Those channels are interpolated and combined to produce the RGB image.

Once the image is demosaiced, it is usually white balanced. Depending on the lighting condition, an unwanted colour-cast may be present in the image. White balancing is used to remove such colour casts. Typically, a white balancing algorithm finds a set of pixels for which it assumes that their true colour is neutral grey. Then it computes a scalar for each channel and correct those pixels to make them neutral grey.

A last-processing step is called the gamma correction. The gamma correction is a non-linear operation applied to all pixels. It is typically a simple power law :

$$I_{out} = I_{in}^{\gamma}. \tag{2.1}$$

The human visual system is more sensitive to differences of darker shades than brighter shades. The gamma correction thus takes advantages of that to compress the highlights in order to use more space to encode darker shades. The impact of gamma correction can be seen on Fig. 2.11.

Image Compression

The resulting post-processed RAW images from the two previous steps usually results in a relatively large file. In fact, at this point, a lot of unnecessary infor-

Figure 2.11 : From left to right : Raw image with linear intensity, Gamma-corrected image

mation is encoded in the file. For instance, the RAW image has been corrupted by some perturbation noises that are barely visible to the human eye.

In order to reduce file sizes, image compression algorithms are often used. In fact, it is very unlikely that a digital media will never be compressed during its life cycle. We thus consider the compression as part of the acquisition process.

On famous image compression algorithm is the JPEG compression. The JPEG compression first convert the image into the YCbCr colour space. In this space, the image is separated into one luminance channel (Y) and two chroma channels (Chroma blue and Chroma red resp. Cb and Cr). Because the human eye is more sensitive to luminance variation, JPEG first highly reduce the size of the chroma channels which is sometimes called chroma subsampling.

Once the chroma subsampling is performed, the JPEG compression also reduces the file size by removing high frequencies which are imperceptible.

To do so, the image is first sliced into non-overlapping blocks of size 8×8. Each block is then transformed using a Discrete Cosine Transform (DCT). Those blocks are then quantified to remove high frequencies. At the end of this process, the image contains many null pixels. This can be further compressed using lossless entropy coding such as the Huffman coding.

The JPEG compression algorithm is a fairly old algorithm, but it is still heavily used. Nowadays many algorithms use principle introduced by the JPEG algorithm.

In this thesis we will study the H.264 video compression algorithm in chapter 7 which has many similarities with JPEG. As for JPEG, H.264 also uses chroma

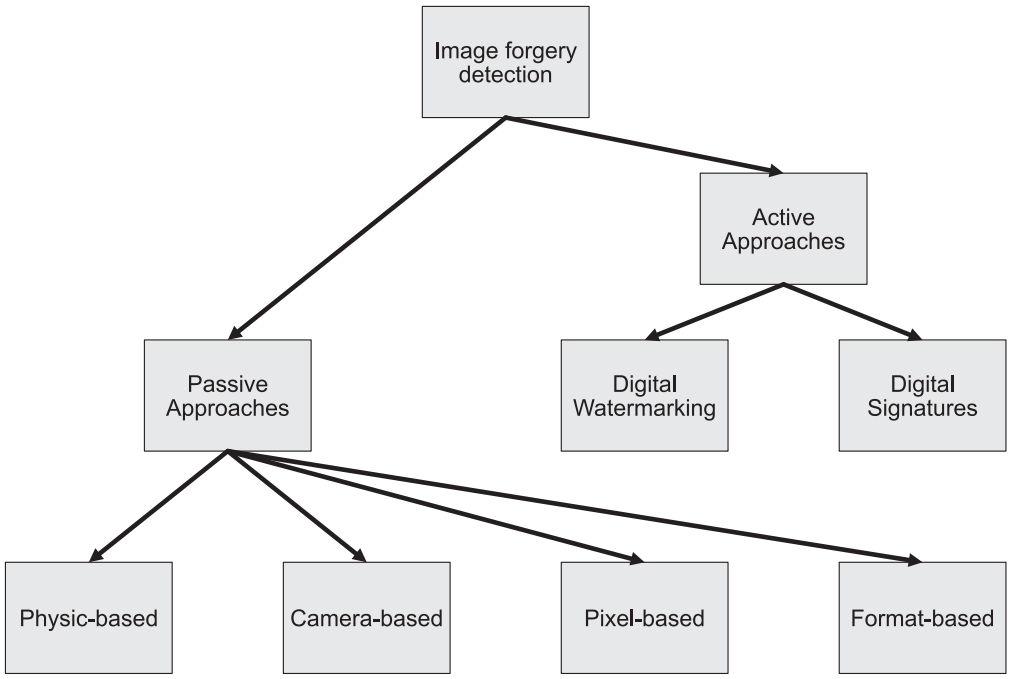


Figure 2.12 : All categories of image forgery detection techniques

subsampling and also compress the luminance using the DCT on non-overlapping blocks.

2.5.2 Forgery Detection Categories

There exist many different categories of image forgery detection methods. An overview of all those categories can be seen in Fig. 2.12.

We can see two main image forgery detection families. The passive approach and the Active approach.

The active approach can be categorised into two main families. The approach based and watermarking and the methods using cryptography. The idea of both families is to add an element to the media that cannot be tampered. This requires to have access to the media and to be certain that the media is authentic at that time. If the idea is similar, the methods used are not. Digital watermarks alter

the pixel information to insert what is called a watermark. A watermark can be visible such as clear text on top of the image or invisible. A watermark can be further classified based on its robustness. For digital tampering detection, fragile watermarks are often used. Such watermarks are destroyed even by the slightest alteration of the media. Cryptography based approaches, on the other hand, do not alter the pixel information. Instead, the media is either encrypted using a cipher algorithm so it cannot be altered without knowing the encryption key. Or it can be digitally signed, so any alteration would be detected by a mismatch between the tampered media and the signature.

On the other hand, passive approaches are based on knowledge about the image formation or assumption on the tampering process. We can differentiate four passive detection categories which we will describe in the next sections.

Physic Based Methods

The first category is the physic-based methods. Those methods rely on physical principles to detect forgeries. With some knowledge on what kind of scene has been acquired, it is in fact possible to model some physical property that elements in the scene should respect.

Many methods rely on the analysis of the lighting environment [30, 31, 32, 33, 34, 35, 36].For example, for an outdoor image during a sunny day we can expect the light direction to be consistent across the image because we can assume the light from the sun to originate from an infinitely far point. Similarly, for an indoor image with people on it, we could look for inconsistencies in the reflection in the eyes to see if a person has been added to the scene . For the detection of deepfakes, authors of [37] proposed to study the inconsistent face pose due to the imprecision of the generation process.

Those methods are extremely specific as they model physical property for well-defined scenarios. One advantage of those approaches is that they are completely independent of the camera, the media type (image or video) and even the compression used. They rely on mistakes made by the counterfeiter that can be extremely hard to avoid (e.g. inconsistent lighting in a splicing).

Camera-Based Methods

Camera-based methods focus on the acquisition device. They will either try to model the image formation pipeline as a whole or instead focus on very specific artefacts. They would model how an image would normally be formed and look for abnormality regarding this model.

For example, some methods [38, 39] modelled how the chromatic aberration due to the lens should be oriented and exposed the forgery by looking for inconsistent orientation. Some approaches [40, 41, 42, 43] assumed that the estimated CRF must be unique for every sub portion of an image and try to detect if there exist a significantly different estimated CRF to expose traces of forgeries. Many methods [44, 45, 46, 47, 48, 49] try to infer the CFA grid and try to expose misaligned or mismatch area as traces of forgeries. Other uses multiple photos from one device to estimates the PRNU [50, 51, 52, 53, 54, 55] which allow them to authenticate later images from the same device or to detect forgeries. Finally, many methods [56, 57, 55, 58, 59, 60] try to expose inconsistencies in the noise as evidence.

The performances of such methods varies greatly depending on the studied artefact. In general, those methods work best when the image quality is sufficient enough. The advantage of those methods is that they provide great evidence of forgeries which are easily interpretable.

Pixel-Based Methods

Unlike the previous two categories, pixel-based methods makes no assumption on the acquisition media nor the physical properties of the acquired object. Rather it makes assumptions about the tampering process that the counterfeiter might use.

For example, Copy-Move detection methods assume that the duplicate part won't be altered much after being pasted. They thus look for abnormally similar elements in the image [61]. Some methods [62, 42] assume that the counterfeiter may blur the boundary of a splice object or even the whole object to seamlessly blend it. They will thus look for traces of artificial blur. Similarly, some methods [63, 64] will look for inconsistent white balance assuming that the counterfeiter will try to match the spliced object's colours.

By definition those methods are generally specific to a certain forgery. Like

physic-based methods, they make no assumption on the device nor the compression.

Format-Based Methods

Finally, the format-based methods focus on the compression applied to the media. They thus must assume that the media was authentic prior to the first compression.

Similarly to camera-based approach, they model either the whole compression process or only a specific part. From there, they will look for inconsistencies in some JPEG artefacts [65], look for traces of double compression [66] or detect previous traces of compression in uncompressed images [67].

In general, those methods are extremely effective. One major disadvantage of such method is that they cannot detect forgeries which occurred prior to the first compression. Although, this can be convenient as those methods can be used in a more active manner by constraining the compression parameters to specific values and authenticate the media based on this knowledge.

2.6 Image and Video Forgery Detection in RIV System

In the previous sections, we presented the standard architecture of a RIV system. We showed that two scenarios could be considered. The acquisition device could be either control or uncontrolled. As explained, having no control over the acquisition device makes the authentication process much harder. This scenario must then be avoided if possible. We will thus only consider RIV system with controlled acquisition device.

We then gave an overview of digital forgery and how those forgeries could be used against a RIV system. From there, we presented the image formation process as a whole and presented the various approaches to detect image forgeries.

We already have an intuition of how and what the counterfeiter might tamper and we introduced the various methods for forgery detection. To select the most appropriate detection approach, we have yet to consider at which point the

counterfeiter will attack the RIV system.

We will first look at where a RIV system using a controlled acquisition device could be compromised. After, we will see how this influence the choice of one forgery detection category to secure a RIV system.

Based on those observations, we will finally present the motivation behind each chapter of this thesis.

2.6.1 Attacks against RIV systems

As already mentioned we only consider RIV systems with controlled acquisition device as we believe that uncontrolled device cannot allow the creation of a secure process.

As we mentioned, in a controlled acquisition device scenario the RIV provider may use the user's terminal as the acquisition device. In such case, an application would be installed on the user's terminal. This application would be responsible for controlling the acquisition device. In this context, it is unwise to ignore the possible for a counterfeiter to compromise the device itself. A more detail pipeline of the acquisition process in such a scenario is given in Fig.

We can see that the image is produced by the acquisition device. The RIV provider's app can control the acquisition device but as no control over the user's terminal as a whole. Once the acquisition is performed, the media is handled by the application which later sends it to a distant authentication server. Once again, the application can control the sending process still can't control the whole terminal. Consequently we can expect the counterfeiter to attack the system at three different points assuming that the distant server cannot be compromised.

The earliest attack can occur if the counterfeiter compromise the terminal itself. In such case, the RIV provider's application cannot even trust the incoming media stream. Similarly, the counterfeiter may compromise communication channel between the user's terminal and the distant server by performing a man in the middle attack. In that case, the stream could be intercepted directly on the user's terminal or through an independent system in between the user's terminal and the distant server. A last approach would consist of the alteration of the application itself. The application would thus directly send tampered media.

When the media finally reaches the authentication server, we do not know which of those scenarios may have occurred. In the best-case scenario, the device is not compromised prior to the application and the application itself is not compromised either. In that case only a man in the middle attack is possible to tamper the digital media. We can thus use any type of forgery detection category. In a less optimistic scenario, the acquisition device is compromised. In that case, not all forgery detection methods may be appropriated. Camera-based methods and format-based methods become less relevant because we cannot trust the stream in the first place. In fact, the counterfeiter may have been able to apply some counterforensic methods to hide traces of forgery. We will see in chapter 6 how such counterforensics can completely fool camera-based approach. Camera-based methods can still be somewhat relevant as the counterfeiter may not be able to completely hide is traces. It is still interesting to use format-based method as we will never know for sure if the tampering is applied after or before the application but we must keep in mind that in the latter case such forgery detection approach won't do much.

In any case, physic-based and pixel-based method stay relevant as no assumption on the acquisition device is made. Because the acquisition device is controlled, the user can be asked to perform specific action that would facilitate physic-based or pixel-based detection later on.

2.6.2 Securing RIV Systems

As we saw, when considering a RIV system with a controlled acquisition device, pixel-based forgery detection methods seems more appropriate. Furthermore, we will be more interested in methods that can be used to detect forgeries in a text or facial manipulations. Even though one of the main focuses of this thesis is to secure RIV systems, we also took part in the ANR-16-DEFA-0002 project called DEFALS which focuses on general image forensic. For those reasons we explored various forgery detection methods during this thesis often with a strong focus on ID documents forgeries but also more general approaches.

The security and reliability of a RIV system rely on effective image tampering detection methods. To assess the effectiveness of such methods, it is necessary to

be able to evaluate those methods on representative data. As part of our work in the project DEFALS, we realised that very few datasets were available at the time. The first work of this thesis was thus to propose a dataset for general image forensics. Because we are interested in RIV security, we also include Facial Manipulation in this dataset. This dataset will be presented in chapter 3.

As we explain, we were then interested in forgery detection methods that focus on text and facial forgeries. When tampering a text, we showed that the easiest and yet most convincing approach was to use a Copy-Move forgery. In the context of a RIV system, the copy-move would be applied to an ID document . In such images, many elements look really similar and are called Similar but Genuine Objects (SGO). Copy-Move detection methods were shown to be less effective in the presence of SGO. We thus decided to address the challenge of SGO for copy-move detection in chapter 4. We also proposed a new dataset of forged ID documents with Copy-Move to promote research in that field.

Before tampering a text, it is likely that the counterfeiter would first have to erase what was already written. He would then use an object-removal forgery. We study the possibility to expose object-removal forgery through the analysis of the reflectance of an image in chapter 5.

One well-known attack against ID document portrait is the Face Morphing attack. In Chapter 6, we study this attack by proposing a novel pixel-based approach. One specificity of the Face Morphing forgery is that it might have been used during the creation of the ID document. In such cases, the attack is not digital. We studied how our method and other state-of-the-art approach react to digital counterforensic techniques and then questioned the applicability of those methods for general Face Morphing detection.

Finally, we wanted to evaluate how we could use a sequence of images to enhance the detection of forgeries. The acquisition of a sequence of images rather than a single image seems indeed intuitive in a RIV system as ID documents and living peoples are not static elements. In fact, an ID document contains dynamic elements such as holograms and variable inks. With the publication of the first PVID regulation which imposed the use of video for RIV system, we finally proposed a new method to detect a double H.264 video compression in chapter 7. Such a format-based method might not be ideal in the case of a RIV system but

is nonetheless interesting as a first approach to video authentication.

2.7 Conclusion

The main focus of this thesis is the use of image forgery detection in Remote Identity Verificaton system. We first gave a formal definition of a RIV system and then presented the common architecture of such a system. We saw that a RIV system's main goal was to authenticate a user against his ID document. To do so, two streams are acquired. One stream containing the ID document and one stream containing himself. The RIV system starts by verifying he ID document and perform some liveness verification on the person. Only then is the person compared against the document. In general the comparison is performed using face biometry. We thus except the document or the person to be the target of any counterfeiter.

After, we presented the structure of an ID document. We gave a brief overview of the various security features and explains the different types of document frauds. We explained how the security features of the document makes a physical tampering difficult and how, on the other hand, a digital alteration was much simpler and accessible. This motivates the study of digital forgeries in the case of a RIV system.

We thus gave an overview of the common image forgery methods and how each of these relates to the case of a document tampering. We then introduced the complete image formation pipeline and presented the main image forgery detection categories.

We then presented how a counterfeiter could attack a RIV system when the acquisition device is controlled. From there, we explain that some forgery detection categories may not be the most appropriate in the case of a RIV system. We finally presented the motivation behind each chapter of this thesis. We explained that a lack of image forgery datasets motivated us to develop the DEFACTO dataset in chapter 3. Then to detect text forgeries on the ID documents we proposed to study copy-move forgeries in chapter 4 but also object-removal forgeries in chapter 5. The other commonly attacked element of ID documents is the portrait. We

thus decided to study Face Morphing forgeries in chapter 6. Finally, in chapter 7, we decided to study the double compression of H.264 video as a first step toward the transition to video imposed by the PVID regulation.

CHAPTER 3

DEFACTO Dataset

3.1 Introduction

This thesis focus on the detection of image forgery in the context of a RIV system. Even though the content of the studied image is specific, the detection method must stay general. In fact, in some RIV systems where the acquisition device is controlled camera-based or format-based method will arguably be more appropriate than method based on some knowledge about the image content.

For this reason, we actively participate in the project ANR-16-DEFA-000 called DEFALS which stands for "Détection de FALSification dans des images"[1]. The goal of this project was to propose image forgery detection methods. We took part in this project as members of the DEFACTO team. The DEFACTO team was a partnership between the University of Technology of Troyes, EURECOM and SURYS. The University of Technology of Troyes would propose classical forgery detection methods whereas EURECOM would perform research deep learning based approach.

Deep learning methods requires a large number of representative data to be effectively trained. At the time very few datasets for image forgery detection were available and those would not represent well the images of the DEFALS challenge. We thus decided to construct a novel public dataset.

The objectives for this dataset were multiples. Firstly, it had to be large and properly annotated in order to be able to train a deep learning method on it. Secondly, it had to be representative of the DEFALS challenge. This implies that a wide range of forgeries had to be proposed but also that those forgeries had to be convincing.

The creation of a convincing and properly annotated forged image is extremely time consuming. Even the simplest copy-move of a letter on an ID document could take about five minutes. Previous datasets were created by hand but only consisted of a maximum of about 2000 images. Because the DEFACTO dataset was meant to train deep learning methods, we targeted a minimum of 200000 forged images. By assuming the creation time of a single forgery to be five minutes, it would then take more than 16000 hours for a single person to produce a total of 200000 forged

[1]English translation : Detection of forged images

images. We thus quickly decided to develop an automatic tampering process.

We will first give a brief overview of the dataset (forgery, categories, annotations, etc.). Then we will describe in greater detail how we developed our automatic tampering algorithm in order to produce meaningful forgeries in each category. Finally, we will conclude on some perspectives on how to further develop this work.

3.2 Related Work

Several publicly available datasets exist, the first one was the Columbia Gray Dataset [68]. It contains only splicing forgeries, no ground truth and images were only in grayscale. Two years later in 2006, they released a new version called Columbia Colour Dataset [41] to extend the first one.

CASIA V1.0 and V2.0 [69] were introduced in 2009 to propose a larger (about 6000 tampered images) datasets with more realistic tampering. It contains splicing and copy-move forgeries with post-processing (blurring along edges or in other regions). They are still widely used as they contain a large number of forged images and their associated authentic images. Many datasets have been released later which only propose copy-move forgeries (MICC [70], IMD [71], CoMoFoD [72], COVERAGE [73], GRIP [74], FAU [71]). The National Institute of Standards and Technology (NIST) also released multiple datasets in the context of the Media Forensic Challenge 2019 [75]. It includes both image and video manipulations. While some parts are accessible upon a simple request, one needs to participate in the challenge to get access to the most recent version of the data.

Previously cited datasets were all made manually by the authors. The Wild Web dataset [76] released in 2015 was the first to introduce the real world tampered images. This dataset contains about 10000 images for which they manually created the ground truth binary masks to locate the forgery. More recently, the PS-Battle Dataset [77] was released. In this work, the author collected images from the active Reddit thread Photoshop Battles. In this thread, people try to produce the best photo manipulation from a given image. They gathered more than 10000 original images and 90000 tampered images. This dataset is meant to provide a long-lasting benchmark dataset and will keep growing with the Reddit community.

In this chapter, we propose a novel dataset called DEFACTO meant for the study and training of image manipulation detection algorithms. We tried to produce a large amount of semantically meaningful forgeries for each category we defined in 3.3.1.

3.3 Dataset Overview

3.3.1 Forgery categories

In this dataset, we wanted to cover most of the common methods that one could use when creating a forgery. Hence, four major categories of forgeries have been considered : copy-move, splicing, object-removal and morphing.

Copy-move forgeries consist in the duplication of an element within the image. For splicing, one portion of an image is copied and pasted onto another image. In object-removal, an object is removed from the image by the use of inpainting algorithms. Finally, morphing consists in warping and blending two images together. For each forgery, post-processing may be applied (rotation, scaling, contrast ...). Those four categories can be seen as elemental forgery operations. An image composite would most likely be a composition of those basic operations. As the methods to detect those categories can be quite different, we decided to first construct a dataset where each image as only been forged using one of those categories only. The whole dataset content is detailed in Table. 3.1. Generating those forgeries in a random manner is simple. This would allow us to produce a large number of forged images but they would be semantically meaningless and easy to detect by the human eyes. We wanted to create a dataset with meaningful forgeries and challenging for the human eye by removing most of the traces that an automatically generated forgery could contain. This goes from generating a proper segmentation of the object to select where to paste it in the final composite image. We believe that some mistakes (strong edges of the forged element) could introduce a bias in learning algorithms and wanted to address this issue.

3.3.2 Annotations

One advantage of automatically generated forgeries is that we can provide precise annotations for each image. In our dataset, each generated forged image is accompanied by diverse annotations to give further information on the tampering process.

General information

for each image, a detailed JSON file is provided. In this file, every operation made on the ground truth images are listed. Parameters used by each operation are detailed.

Localisation

every image is also accompanied by one or more ground truth binary masks. One binary mask serves to localise the forgery under the *probe_mask* directory. For splicing, copy-move, face morphing and swapping, a binary mask under the *donor_mask* directory gives the localisation of the source. Object-removal has a binary mask under the *inpaint_mask* directory which localise what has been filled by the inpainting algorithm.

3.4 Automating forgery creation

3.4.1 Segmenting meaningful objects

To produce meaningful forgeries, we took advantages of MSCOCO dataset [78]. They collected more than 300,000 non-iconic images from Flickr. Afterwards, they defined 91 object categories of objects and annotated all the images. Those annotations include the segmentation of the objects that we use as a base to produce our forgeries. The raw segmentation annotation cannot be used directly to generate a forgery as they are not precise enough (Fig. 3.1a). They need to be processed to obtain more suitable segmentation.

Table 3.1 : Number of images per category in DEFACTO

Forgeries	Copy-Move	Inpainting	Splicing	Morphing
Images	19000	25000	105000	80000

3.4.2 Refining segmentation

MSCOCO has over 2,000,000 object instances. Refining all annotations or even a small part by hand was not possible. We employed an alpha matting technique to refine the masks. Alpha matting consists in finding the foreground of an image

$$I = F\alpha + B(1 - \alpha) \tag{3.1}$$

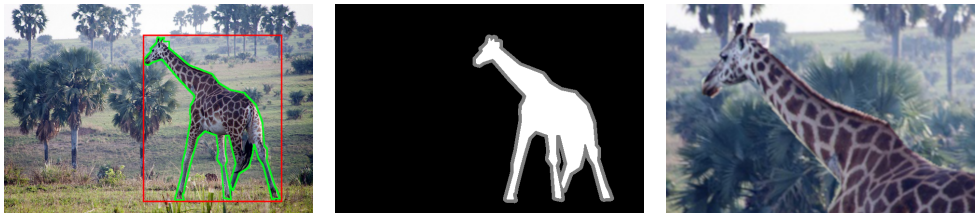
where I is the image, F the foreground, B the background and α the *alpha matte*.
This equation cannot be solved without any prior information. Thus, alpha matting techniques rely on a user input to define areas that are known to be part of the foreground F and areas that are known to be part of the background B. Those inputs can go from simple lines to what is called a trimap.

Trimap defines three areas: the foreground, the background and an unknown area (Fig. 3.1b).

Based on the given foreground and background areas, the alpha matting algorithm automatically affects a value to every unknown pixel to produce the final *alpha matte*.

We used MSCOCO raw segmentation to construct the trimaps. First the foreground region is obtained by applying a morphological erosion to the raw MSCOCO mask to make sure that no background pixel is added to the foreground region. The unknown region is obtained by applying a morphological dilation to the raw mask (Fig. 3.1b). We then use a modified version of [79] to produce the *alpha matte*. This *alpha matte* is finally used to produce a much more convincing segmentation of the objects. This allows us to produce forgeries that are more pleasant (Fig. 3.1d).

Having a good segmentation of the objects is a first step toward the automatic generation of meaningful forgeries. Though it is not enough, removing or copying



(a) Raw MSCOCO annotations

(b) Generated trimap

(c) Copy-move without alpha matting

(d) Copy-move with alpha matting

Figure 3.1 : MSCOCO mask refinement

those objects in a random manner would most certainly produce bad results. For each category of forgery, a set of rules had to be applied to maximise the chances of producing good results.

3.4.3 Object Removal

There are six main categories of image inpainting [80][81][82][83] techniques. Texture synthesis based inpainting, exemplar-based inpainting, Partial Differential Equation (PDE) based inpainting, hybrid-inpainting semi-automatic and fast inpainting and deep learning based inpainting.

Semi-automatic and fast inpainting methods were not usable in our case as they rely on a strong user interaction. PDE based inpainting are not able to fill a large missing region. Texture synthesis based methods are meant to fill the region with homogeneous textures which is something we cannot ensure in our case. Hybrid inpainting methods are a combination of PDE and texture synthesis and thus does not suit our needs. Deep learning based methods are efficient but depend on the performances vary depending on the dataset used for training and the targeted

images to inpaint.

We thus used an exemplar-based inpainting method [84] as they have proved to be effective in various conditions and are still in common retouching software.

Inpainting methods are more efficient if the subject to remove is on a relatively simple background (Fig. 3.2). To produce convincing forgeries, we excluded objects for which the background was too complex. A border region is extracted by dilating the raw MSCOCO mask, and the standard deviation is computed within this region. Objects for which the standard deviation was below a fixed threshold were kept.

3.4.4 Copy-move

For copy-move, the raw MSCOCO annotations were first used to produce the forgeries. Resulting images were almost systematically unpleasing due to strong visual artefacts on the object's borders (Fig. 3.1c). For this reason we had to use alpha matting as described in 3.4.2 to obtain much better results in most cases (Fig. 3.1d). At first, objects would be copied and paste randomly within the image. This would cause the creation of images that were often semantically incorrect (*e.g.* people walking in the sky ...). To reduce chances of producing those kinds of images, we constrained the location of the forgery to be on the same axis as the source object (Fig. 3.3a and 3.3b). Decision to stay on the x or y axis is based on the *width* and *height* of the object. If $width > height$ then the object is duplicated on the x axis otherwise it is duplicated on the y axis. As for the object removal, we only kept objects on a fairly simple surroundings. This is to prevent to copy-move an object that is too tightly coupled with its context. For instance, in the MSCOCO dataset, a person and his backpack would be annotated separately, thus copying the person would not produce a good forgery as the backpack would be missing. Those rules allowed us to produce convincing copy-move forgeries (Fig. 3.3).

(a) Ground truth image (b) Inpainted image

Figure 3.2 : Example of inpainting

3.4.5 Splicing

Splicing is arguably the most complex forgery to generate automatically. When creating a splicing, an object from an image is pasted onto a new one. When manually creating the splicing, we can make sure that the target image has a similar point of view, lighting condition and so on. Those are essential to make a realistic splicing. Unfortunately, it is extremely difficult to assess all those elements automatically, thus it is extremely hard to produce good splicing forgeries. To address this issue, we decided to limit the MSCOCO categories used to produce splicing forgeries. We kept categories for which the objects appearance does not vary or depend too much on the point of view and the context. For example, a person could be standing, sitting, running ... and for each case, the appearance would vary a lot depending on the point of view to make it almost impossible to generate a convincing forged image. But if we take a sports ball, which is spherical, it will always look about the same and will be more easily spliced onto another image. Or if we take a bird which is either standing or flying, cutting and pasting it onto an image already containing birds has reasonable chances of producing an acceptable result. Objects are either pasted on an object of the same category or randomly pasted in a relatively smooth area to avoid pasting an object onto another one (Fig. 3.4b).

3.4.6 Face Morphing

Image morphing was originally used to produce smooth transitions between many images. One of the first notable appearances was in the Michael Jackson music

(a) Copy-move of the kite on the y axis

(b) Copy-move of the bird on the x axis

Figure 3.3 : Example of copy-move

(a) Ground truths

(b) Tampered

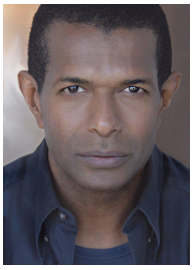
Figure 3.4 : Example of splicing

video "Black or White" where the method described in [85] was used to morph peoples together. Nowadays, face morphing as received particular attention among the forensic community [86]. As shown in [87], Automatic Border Control systems are very vulnerable to face morphing attack. Thus we decided to include such forgeries into our dataset.

We gathered public figure portraits on IMDB website and selected 200 front facing actors with a relatively neutral expression as a base to generate our face morphing forgeries.

The complete face morphing process can be seen in Fig. 3.5. Given two faces A and B, we used Dlib [88] to extract a set of facial landmarks. Thanks to those landmarks, the two are first roughly align with respect to their eyes (Fig. 3.5c). A

(a) Face A (b) Face B (c) Rough alignment of faces A and B (d) Precise alignment of faces A and B (e) Without color balance



(f) With color balance

Figure 3.5 : Automatic Face Morphing creation

weighted average by a factor $\alpha \in [0, 1]$ of the two sets of landmarks is computed, and the two faces are warped [89] to this weighted average to precisely align them (Fig. 3.5d). Finally, the two faces are alpha blended using the same factor α. A local RGB scaling is performed to better take the skin tone into account. The factor α allows us to decide which face's biometric traits are more visible. A α value of one produces what is commonly called a face swapping.

3.5 Conclusion

Earlier image forgery detection methods were mostly based on knowledge of the image format, the camera properties, etc. Many researchers thus published their methods alongside with a handmade dataset representing their use case. As a consequence, there were no large datasets that were publicly available.

To our knowledge, the first initiative to propose such a dataset was the Photo-

shopBattle dataset. They gathered real-world tampered images from the internet and managed to create a dataset of over 80000 images which is perfect to evaluate forensic algorithms.

More recently, researchers have started to develop image forgery detection method based on deep learning approach. For those, a large dataset of representative example is needed. Apart from the PhotoshopBattle dataset, no other large datasets existed. Unfortunately, even though the PhotoshopBattle dataset is excellent for evaluation purposes, it does not provide enough annotation needed to train a deep neural network.

As part of the project ANR-16-DEFA-000 we thus developed a dataset of over 200000 images. We developed an automatic tampering algorithm which could generate convincing forgeries with exact annotations. Since we released the DEFACTO dataset, another large dataset (IMD2020) has been released. We think that the DEFACTO dataset along with the IMD2020 and the PhotoshopBattle dataset now allows for proper training and evaluation of deep-learning-based detection algorithms.

But future works are still needed. In particular, more work on the image provenance is needed. At the moment, both the IMD2020 and the DEFACTO dataset gathered the ground truth images on the web. What we thus consider as a ground truth is in fact already advance in the image acquisition pipeline. The demosaicing has been performed, some global adjustment may have been performed and a first compression may have been applied. Ideally, all forged images should be created from raw input.

Initially we wanted to develop the DEFACTO dataset using the raw RAISE image dataset but without annotation we could only generate completely random forgeries. One of the objectives was to develop an automatic algorithm that would start from the raw image and apply post-processing such as resampling, global enhancements, compression or local forgeries successively. This would allow us to generate a precise history of the image or even playback the tampering process until a certain point and so on. Interestingly, the JSON files in the DEFACTO dataset are actually traces of prior works toward that direction.

Future work may focus on the development of such a tool rather than developing more dataset. A flexible enough tool may allow researchers to design tampering

scenarios from start to finish and apply those scenarios on already existing raw image dataset. This would be extremely beneficial for future research.

CHAPTER 4

Copy-Move detection on ID Documents

4.1 Introduction

Copy-Move is one of the main image forgery categories. Even though this technique seems rather simple and quite limited, it is actually one of the more versatile and most used tampering techniques. In its simplest form, copy-move is used to duplicate some of the image content. To change some quantities (e.g. turn a few people into a crowd) or to alter some text for which we do not have access to the original fonts. In more advanced composites, copy-move is typically used as a way of removing something from an image by progressively hiding it behind small copied and pasted elements. For example, the widely used clone stamp tool in software such as Photoshop, Affinity Photo or GIMP allows the user to simply create a copy-move forgery and is often used to correct small elements such as freckles on a face, electrical wires in landscapes. Finally, many inpainting algorithms use some form of copy-move.

We thus understand how beneficial a copy-move detection method is for general image forensic analysis. It is even more interesting in the case of document fraud detection as copy-move is the way to go to tamper a document digitally. In fact, when tampering a document it is very likely that one will try to alter some of the variable field. In such case, copy-move is both the simplest way to proceed but will also produce the most realistic results. In our case it is thus essential to be able to detect such forgeries.

The task of Copy-Move Forgery Detection (CMFD) mainly consist in the detection of abnormally similar elements in an image. The main challenge of CMFD is thus defined in its own definition. What is considered to be abnormally similar ? Even though this seems like an obvious statement, CMFD has been studied for years without really considering this question. To the best of our knowledge, the first CMFD algorithm can be dated as far back as 2003 [90], it was then quickly followed by [91] and later by many more methods [92, 93, 94, 95]. But it was not until 2016 that the main challenge of CMFD was first exposed in [73]. The authors of [73] in fact realised that CMFD aims at discovering abnormally similar objects but none of the available datasets at that time proposed images containing naturally similar objects. They introduced the term Similar but Genuine Objects (SGO) to refer to those. They thus proposed a dataset with many SGO and showed

that many algorithms performed poorly on this dataset.

Our main goal is to perform a forensic analysis of an ID document. We will thus focus on images which contains a lot of SGO (i.e. the letters). The poor performance of CMFD algorithms in the presence of SGO is then a strong limitation that makes state-of-the-art methods ill suited to our use case. It is then necessary to find a way to reduce as much as possible CMFD false positives while maintaining good detection performances.

In this chapter we will address the problematic of SGO for CMFD algorithms in an attempt to propose a method well suited in the case of ID documents forensic. We will evaluate the impact of adding a block-based filtering steps based on the local dissimilarity map on top of a typical keypoint-based technique. We will show that not only we can achieve better results than state-of-the-art CMFD algorithms, but we do so with a much less elaborated keypoint-based scheme that current methods. This will show one inherent flaws of keypoint-based algorithms that must be addressed in order to properly handle the challenge of SGO.

Then we will introduce a new dataset containing only forged ID documents. Even though the COVERAGE dataset [73] focuses on SGO, the dataset is rather small (only 100 images) and the images are also quite small. In addition, we believe that images from COVERAGE do not represent practical examples as the forgeries are large, which may simplify the detection of the forgery even in presence of SGO. We will first give a brief overview of our dataset and its advantages over COVERAGE and then evaluate our method and other state-of-the-art algorithms on this newly proposed dataset.

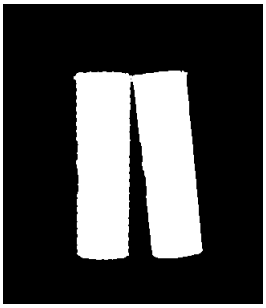
4.2 Related work

Copy-move forgery detection algorithms can be divided in four major categories. Block-based detection, Keypoint-based detection, hybrid detection and deep learning methods.

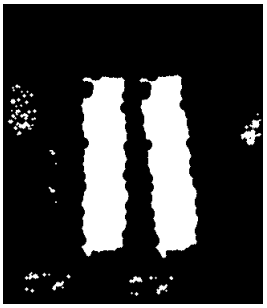
Block-based detection algorithms [96, 97, 94, 92] divide the image into regular or non-regular blocks which are then clustered according to some defined similarity measure. Keypoint-based detection algorithms [93, 98, 99, 100, 101, 102, 103, 104]

(a) Original image

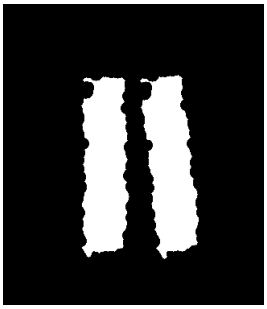
(b) Tampered image



(c) Ground truth mask



(d) Detection without LDM^{XYZ}



(e) Detection with LDM^{XYZ}

Figure 4.1 : Images from COVERAGE dataset, LDM^{XYZ} and examples of detections

can be divided into two steps. First a set of keypoints is extracted from the image and one or many descriptors are computed for each keypoint. Then those keypoints are matched and clustered. The clustering process aims at removing wrongly matched keypoints by discarding clusters which do not have correct properties (size, structure ...). Hybrid-based detection algorithms [105, 106] try to combine the advantages of both the sides. Lastly, many deep learning based approaches have emerged [107, 108, 109, 110, 111, 112]. Such methods train deep learning networks to detect and locate copy-move forgeries.

Whatever the method used, copy-move detection algorithms tend to produce high false positive rates on images that have repeated or similar structures. As mentioned earlier, authors of [73] proposed a dataset called COVERAGE meant for studying how detection algorithms behave in the presence of highly similar but

genuine objects in images. Examples from the COVERAGE dataset can be seen in Fig. 4.1a and Fig. 4.1b. The ground truth binary mask (Fig. 4.1c) indicates that rightmost box of potato chips has been copied and moved onto the leftmost one. In [99] many state-of-the-art algorithms have been evaluated onto the COVERAGE dataset and have generated high false positive rates [99, Table II].

In this chapter, we propose a novel hybrid method. We evaluate the use of a colour local dissimilarity map as a way of reducing false positive rates of keypoint-based detection algorithms.

4.3 Overview of the method

As presented in section 4.2, four main kinds of CMFD exist. The Keypoint-based methods, the block-based methods and the deep-learning-based methods. A last category contains the so-called hybrid methods which combine multiple techniques to take full advantage of each. Our target application is the detection of copy-move forgeries in a remote onboarding system. In this application, processing time is of crucial importance as a lot of images must be processed in very little time. For this reason, we decided to focus our attention on keypoint-based methods which are much more efficient. Unfortunately, they are also extremely sensitive to SGO. Keypoint-based methods are typically based on state of the arts interest point detection methods such as SIFT [113], SURF [114], ORB [115] . Such methods are in general designed to be invariant to rotation, translation, scaling, etc. This allows those methods to detect one object even with slight variations of points of view, lighting condition, etc. This is perfect for object detection but will inevitably cause troubles for CMFD in presence of SGO. To compensate for this, Keypoint-based CMFD rely on the fact that the matched keypoints will have a remarkable structure. Various clustering methods are thus applied to try and filter false positives. If this is in fact enough to filter obvious false positives recent studies on the COVERAGE datasets [73, 99] indicates that this might not be enough to tackle the challenge of SGO.

We believe that clustering is not enough to differentiate a copy-move forgery from a SGO. We thus decided to employ a hybrid approach. In general, hybrid

approaches are mostly used to perform a more precise pixel wise detection. In our case, we decided to use another block-based filtering steps to remove falsely matched keypoints instead of improving pixel wise detection. If this helps with in presence of SGO, we will see that it has the drawbacks of reducing the localisation performances.

Our method will thus follow a typical Keypoint-based CMFD pipeline. First keypoints are extracted and matched, then a clustering step aims at removing obvious false positives. We then proposed to add a new filtering step based on the local dissimilarity map to further verify the remaining matched keypoints. In the next section, we will present each step and then we will present some results obtain on three common datasets including COVERAGE.

4.4 SIFT Detection

4.4.1 Keypoint extraction

Keypoint extraction is done using SIFT. In order to detect all kinds of copy-move forgeries, it is important to extract as much keypoints as possible. If the duplicated portions of the image are not sufficiently covered with keypoints, it would be almost impossible to detect the forgery. To address this issue, keypoints are extracted using non-overlapping sliding windows. For each window the contrast threshold is adjusted, so keypoints in portions of the image with lower gradients are kept.

4.4.2 Matching

Authors of [113] describe a method to match the SIFT descriptors using the second nearest neighbour ($2NN$).

For a given descriptor, the two nearest neighbour with distances d_1 and d_2 are found. To decide whether a match is a false positive, the following condition is tested :

$$\frac{d_1}{d_2} < \delta, \delta \in [0, 1]. \tag{4.1}$$

The match is considered to be a false positive if (4.1) is false. The recommended value by [113] for δ is 0.6 which we used in all our experiments.

When using the $2NN$ test, only one match is considered per keypoint. The $2NN$ test was designed to search for a single occurrence of a given object inside an image. In the case of copy-move forgery, there may be many instances of the object in the image thus making the $2NN$ test not adapted for multiple copy-move detection.

For this reason, the authors of [93] propose a generalised version $g2NN$ that can handle multiple copy-move. In the $g2NN$ test, the k nearest neighbours are found with distances $d_i, i \in [1, k]$. For each pair (d_i, d_{i+1}), if $\frac{d_i}{d_{i+1}} < \delta$ then the i^{th} nearest neighbour is kept as a true positive match.

We used this generalised test which allows the detection of multiple copy-move forgery.

4.4.3 Clustering

During the matching step, the threshold δ cannot be too low in order to detect copy-move with post processing applied. A tradeoff has to be made in order to maintain a low false positive rate and a high true positive rate. The matching step is not sufficient enough to discard all false positives so a second filtering is needed.

One common way of filtering the remaining false positives is to take advantage of the properties that clusters of true positives match should have. In particular, when an object is duplicated with a simple transformation (i.e. an affine transformation), we should observe large clusters of matched keypoints having a similar norm and orientation (see Fig. 4.2).

We achieve the clustering using the method described in [116] which do the clustering according to a set of predefined rules.

Given an object O and O_D its duplicate (Fig. 4.2). Given A and C two keypoints inside O and the keypoints B and D inside O_D. Given the match M_{AB} between A and B forming $\overrightarrow{AB}$. Given the match M_{CD} between C and D forming $\overrightarrow{CD}$.

The two matches M_{AB} and M_{CD} are considered equivalent if :

$$\|\overrightarrow{AB} - \overrightarrow{CD}\| < \delta_1 \tag{4.2}$$

$$\|AC\| < \delta_2 \text{ and } \|BD\| < \delta_2, \tag{4.3}$$

$$\|AB\| > \delta_3 \text{ and } \|CD\| > \delta_3. \tag{4.4}$$

The threshold δ_1 constrains the difference in orientation between the two vectors $\overrightarrow{AB}$ and $\overrightarrow{CD}$. If δ_1 is set too low, the number of generated clusters will tend to grow. A duplicated object that has been slightly transformed will be cut into smaller clusters. If δ_1 is too high, the number of generated clusters will tend to decrease and false positives match might be added into true positives clusters. This threshold does not depend on the image size. In our experimentation, δ_1 has been set to 10.

The maximum distance between A and C is bounded by the size of the object O. If δ_2 is too high, the number of generated clusters will tend to decrease and false positives match might be added into true positives clusters. But δ_2 must not be too small because of some regions where the density of the extracted keypoints might be lower.

The distance between A and B cannot be lower than a given threshold δ_3 and so does the distance between C and D. δ_2 and δ_3 thresholds mostly depends on the image size and the SIFT keypoints density. δ_2 and δ_3 were experimentally set to 50.

The choice of values for δ_1, δ_2 and δ_3 is not critical if the contrast threshold in 4.4.1 is chosen arbitrarily small.

At the end of the clustering step, every cluster containing fewer than three matches are discarded as we expect large clusters on duplicated objects.

4.5 Filtering with a local dissimilarity map

The SIFT descriptor is the histogram of oriented gradients around the keypoint. This descriptor does not contain enough information about the local structure around the keypoint. In the case of copy-move forgery detection, many false

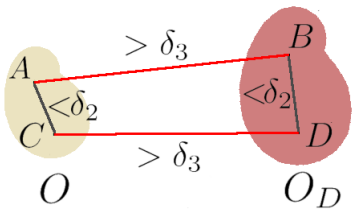


Figure 4.2 : Duplication of O and equivalence rules

positives can be produced when the image contains highly repetitive or similar structure. The use of this information about the structure and colour around the keypoint could help to discard obvious false positives. We propose to use the Local Dissimilarity Map described in [117] to filter those false positives.

4.5.1 Local Dissimilarity Map

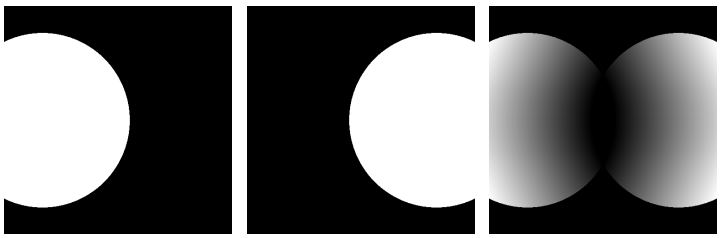
The Local Dissimilarity Map (LDM) introduced in [117] allows the quantification of local dissimilarities between binary images. For this, a modified version of the Hausdroff distance is proposed. For two binary images A and B, the LDM is defined from $\mathbb{R}^2 \times \mathbb{R}^2$ to $\mathbb{R}^2$ by

$$\mathrm{LDM}_{\mathrm{bin}}(A, B)(p) = \mid A(p) - B(p) \mid \max(d_A(p), d_B(p)) \tag{4.5}$$

with $p = (x, y)$ and $d_X(p)$ the distance transform of X at p. In the distance transform $d_X(p)$, every pixel p takes for value its distance to the closest zero-pixel in the image. An example of binary LDM is given on Fig. 4.3, darker pixels mean higher similarity.

An extension of the LDM for the greyscale images is used [118]. Images are first cut into two sets of binary images. The greyscale LDM is then computed as the sum of the LDM of each binary image pair. Given A and B two greyscale images, the LDM is thus defined from $\mathbb{R}^2 \times \mathbb{R}^2$ to $\mathbb{R}^2$ by

$$\mathrm{LDM}(A, B)(p) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{LDM}_{\mathrm{bin}}(\mathrm{A_i}, \mathrm{B_i})(\mathrm{p}) \tag{4.6}$$



(a) Image A (b) Image B (c) LDM between A and B

Figure 4.3 : Binary Local Dissimilarity Map

where N is the number of cuts, A_i (respectively B_i) is a binary version of A (respectively B), obtained by a global thresholding of $A > s_i$. The thresholds s_i are regularly spaced between 0 and the maximum m of each image. For example, for A : $s_i = \frac{i}{N} m_A, i \in [1..N]$.

4.5.2 LDM for images with C channels

For the copy-move, we propose a direct extension of the greyscale LDM for images with C channels. In our case, images are converted into the CIE XYZ colour space which provides a colour representation that is closer to the human visual system and was less sensitive to slight illumination and colour change. Given A^k the k^{th} channel of image A in CIE XYZ colour space. The LDM^{XYZ} is defined as

$$\text{LDM}^{\text{XYZ}}(A, B)(p) = \frac{1}{3} \sum_k \text{LDM}(A^k, B^k)(p). \tag{4.7}$$

Table 4.1 : TPR, FPR, F_1 (image level) on FAU [71] GRIP [94] and COVERAGE [73] and F_P (pixel level) on GRIP

Method	COVERAGE [73]			FAU [71]			GRIP [94]			
	TPR	FPR	F_1	TPR	FPR	F_1	TPR	FPR	F_1	F_P
Amerini [93]	85.71	54.95	71.23	66.67	10.42	75.29	70	20	73.68	-
Cozzolino [94]	59.34	21.98	65.45	97.92	8.33	94.95	98.75	8.75	95.18	92.99
J. Li [96]	87.91	63.74	69.87	72.92	22.92	74.47	83.75	35	76.57	27.24
Y. Li [99]	80.22	41.76	72.28	100	2.08	98.97	100	0	100	94.66
Proposed	70.02	7.89	78.73	100	0	100	100	0	100	82.40

4.5.3 LDM Filtering

At the end of the clustering step 4.4.3, most of the false positives from 4.4.2 are removed.

On Fig. 4.1d, we can see the detection result without filtering with the LDM. Elements are still wrongly detected as copy-move. To remove those last false positives, every remaining matching after 4.4.3 will be confirmed using the LDM^{XYZ}.

For each pair of matched keypoints in a cluster, two windows W_1 and W_2 are extracted. The windows are centred on the keypoints. The sizes of the windows are fixed by the sizes associated with the keypoints and the windows are rotated according to the keypoints angles. The match is finally considered as a false positive if

$$\|\mathrm{LDM}^{\mathrm{XYZ}}(W_1, W_2)\|_2 > \delta_{\mathrm{LDM}}. \tag{4.8}$$

As in 4.4.3, all the clusters containing fewer than three elements are considered as false positives and thus removed. We can see the detection result after the LDM filtering step on Fig. 4.1e.

4.6 Experimentation

Experiments were conducted on the FAU [71], GRIP [94] and the COVERAGE [73] dataset.

On FAU and GRIP datasets the threshold δ_{LDM} has been fixed to 7 while δ_1, δ_2 and δ_3 were chosen to maximise the true positive rates. The true positive rate (TPR), false positive rate (FPR) and the F_1 score is computed at image level on each dataset. For the GRIP dataset, the F_1 score has also been computed at pixel level and is reported in Tab. 4.1 as F_P. We can see in Tab. 4.1 that our detector performs without error at image level on the FAU and GRIP dataset. We further evaluate the performance at pixel level on GRIP. Even though the F_1 score at pixel level is lower than state-of-the-art methods, it is worth noticing that the true positive rate at pixel level is 74.92% and the false positive rate is only 00.16%.

For the COVERAGE δ_1, δ_2 and δ_3 are fixed to maximise the true positive rate without LDM filtering and δ_{LDM} vary.

The detection algorithm is applied to each image. The TPR and FPR at image level is evaluated with and without LDM filtering. On Fig. 4.4 the TPR, FPR and F_1 scores are reported with the varying threshold δ_{LDM}.

Without filtering, the TPR is maximum but the FPR is extremely high (i.e. 93%). Those initial performances are reported as horizontal dotted lines on Fig. 4.4. We can see that the detector (without filtering) performs really poorly on the COVERAGE images containing highly similar objects.

The benefit of the LDM filtering is evaluated by varying the threshold δ_{LDM}. For a strict threshold (i.e. 2), the LDM allow obtaining an FPR of 0%, but the TPR drops from 100% to only 30% and an F_1 score of only 45%. With δ_{LDM} increasing, the detector reach is best performance at $\delta_{\text{LDM}} = 7$, with a TPR of 70%, FPR of 7% and an F_1 score of 78%. The TPR, FPR and F_1 score then tend to the initial performance without filtering.

The impact on the processing time depends on the remaining clusters that need to be validated with the LDM at the end of 4.4.3 as well as the window size of every compared keypoints pair. On the COVERAGE dataset, the global processing time is about 400 seconds. With the LDM filtering it reaches about 815 seconds which is an average of 1 second of added processing time per image.

4.7 CMID dataset

As we saw in Table. 4.1 images containing SGO is a real challenge for CMFD algorithms. When not designed with this problematic in mind, we will observe that they tend to produce many false positives making them impractical. When design with this goal, we will see that it is possible to reduce the false positives but at the cost of a lower detection rate. In this section we propose a new publicly available dataset called CMID[1] to address a few novel challenges. The objective of this dataset is to provide challenging copy-move forgeries that also correspond to a real life problematic. We thus decided to produce digital ID document forgeries.

This specific type of content comes with many challenges for CMFD algorithms. The first one being the large number of SGO. Because the font in an ID is most

[1] https://cmiddataset.github.io/

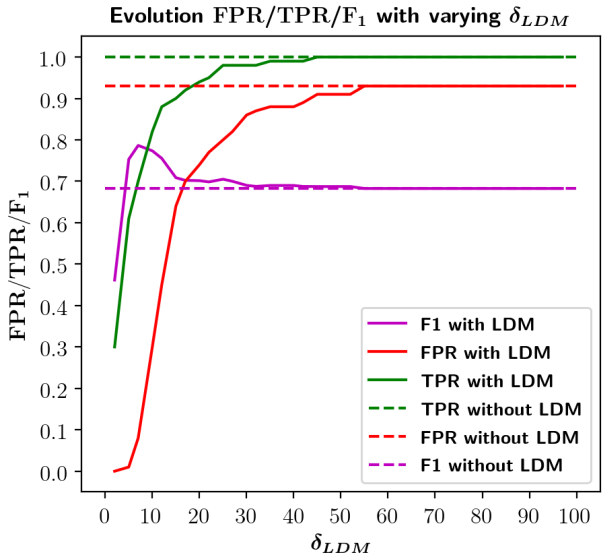


Figure 4.4 : Evolution of the false positives rate (FPR), true positives rate (TPR) and the F_1 score with respect to δ_{LDM} on the COVERAGE dataset

of the time the same across the whole document, we will most likely observe many occurrences of the same letter at the same scale. To complicate the task, standard ID documents fonts are typically quite simple making it more difficult to distinguish two distinct letters in the first place (see Fig. 4.5).

Another challenge that arise from those kinds of forgeries is the size of the tampering. When producing a digital forgery of an ID document, the counterfeiter will only duplicate a small surrounding of the letter and might apply a smooth transition to seamlessly blend the tampering. Doing so we might be left with only a few tampered pixels.

In this context the challenges for CMFD are multiples. First, they must be able to detect the tampering at this small scale. We will see that this is not a trivial task even in the absence of post-processing. Having detected the tampering, the CMFD must then be able to tell the difference between a SGO and an actual fake. This notion of scale is an addition in comparison with the COVERAGE dataset where the forgeries are actually quite large.

4.8 Dataset overview

In this section we will give a brief overview of the dataset goals, content and the creation process.

4.8.1 Dataset content

The dataset only contains copy-move forgeries. Genuine images were acquired on a smartphone, cropped to mostly contain the ID document and saved to JPEG with a quality factor of 95. All images from the dataset have the size 1342 by 943.

The dataset contains three folders named ref, gt, $tampered$. The ref folder contains 304 images named :
NAME_Y
with NAME the identifier of the document (e.g. Belgian passport) and Y $\in 0, 1, ..., 18$. The dataset contains a total of 16 different documents for which 19 images are provided.

The *tampered* folder contains all the tampered images, we generated three tampered images for each genuine image. For tampered images, the following naming convention is employed :
X_NAME_Y
With X $\in 0, 1, 2$, NAME the ID document name (matches the reference identifier in ref) and Y $\in 0, 1, ...18$ the reference image number.

The total numbers of pristine and tampered images can be seen in Table 4.2.

Finally, the gt folder contains all the ground truths. Ground truth images will have the same name as the tampered ones. Ground truth images will be presented as coloured images to differentiate the source and the target of the tampering. Pure green pixels are the sources i.e. the region that is duplicated and pure red pixels are the target i.e. the region that has been duplicated.

For convenience, a program is provided with the dataset to compute the image-level scores and pixel-level scores. In particular the program will compute the True Positives (TP), the False Positives (FP), the True Negatives (TN) and the False Negatives (FN) which can later be used to compute other desired metrics. More information on this program is available on the dataset page (link available in 4.7).

Document types	Pristine	Tampered	Image size
16	304	893	1342×943

Table 4.2 : Dataset content

Figure 4.5 : From left to right : Genuine image, Binarisation of the letters, Bounding box of the letters, chosen letter pair

4.8.2 Automatic tampering process

To generate the dataset, we developed an algorithm that automatically produce the document forgeries. The method can be decomposed into three basic steps. The letter binarisation, the bounding box detection and filtering and the pair selection.

In the first step, we try to remove most of the image content to isolate the letters. To do so, we first assume that a letter will always be thin strokes of a dark shade. Following this assumption, we apply a black top-hat with an arbitrary rectangle kernel of size 7×7 to each of the RGB channels to highlight small dark regions in each channel. We then take the maximum value of the resulting filtered channels and then binarise the result following Otsu's method [119]. The result can be seen on Fig. 4.5. We then found the bounding boxes of every connected component of the binarised image. Those boxes are then filtered according to their size. Again the size is fixed experimentally to isolate letters (result visible on Fig. 4.5). Finally, a pair of bounding boxes with approximately the same size is chosen (Fig. 4.5). One of these letters is then copy and paste onto the other one.

4.8.3 Tampering size

As stated in section 4.7, one challenge introduced in our dataset is the size of the tampering. Current public datasets propose copy-move forgeries with various post-processing such as rotation, scaling, JPEG compression and so on. Overall, state-of-the-art methods seem to manage such processing with ease except in some extreme cases. Those properties are indeed wanted. One challenge that has been overlooked is the ability to detect forgeries when both the source and the target of the copy-move are relatively small. Many problems arise in such situation whatever the method type (i.e. block-based, keypoint-based, hybrid ...).

The first and obvious issue is the ability of the method to detect the tampering. For block-based method, the problem will have to do with computation time. Because of the size, if the blocks are too big, the tampering will most likely be missed as the blocks would mostly consist of pristine pixels. By decreasing the block size, the computation time will quickly rise as more and more comparison will have to be performed. For Keypoint-based method, the problem slightly differ. As for block-based method, the computation will also increase but not as notably. The challenge mostly reside in the huge addition of probable false positives. To detect smaller tampering, keypoints must be extracted from more scales and by being less strict on the stability (e.g. lower contrast thresholds for SIFT/SURF). This compromise has to be done in order to have a sufficient number of keypoints in both the source and target location. This increases the chances of falsely match pristine areas and current methods often sacrifice detection performances at smaller scale for better precision.

When the tampering is detected, another challenge arises because of the smaller forgery size. Whatever the method used, some kind of similarity measure is applied to decide if the detection is a tampering. With less information due to the reduced forgery size, it becomes much harder to differentiate a SGO from a tampered comparison. Because our dataset contains only ID documents images, those new challenges arise naturally as we duplicated small letters. In Table 4.3, a few metrics are given as a comparison between our dataset, COVERAGE [73], GRIP [94] and the DEFACTO dataset [1]. All values represent the percentage of tampered pixels (including both source and target) in the whole image. For COVERAGE, we can

Dataset	Minimum	Maximum	Average
COVERAGE [73]	0.02	0.56	0.23
GRIP [94]	0.0131	0.1371	0.047
DEFACTO [1]	0.0006	1	0.0632
Our	0.0015	0.0154	0.0021

Table 4.3 : Datasets tampering size information

see how the forgeries are relatively big with an average of 23% percents of the pixel tampered. Even though the COVERAGE dataset contains many SGO, the size of the tampered area allows CMFD algorithms to be more confident about their detection. For the GRIP dataset, the forgeries are smaller but images do not contain SGO, so CMFD algorithms are less likely to produce false positives. As for GRIP, the DEFACTO dataset contains smaller forgeries but does not contain SGO. Our dataset contains both small forgeries (average of only 0.21%) and also many SGO. We will see in the next section how current state-of-the-art methods handle this new challenge.

4.9 Baseline results

4.9.1 Algorithms

To assess the challenging aspect of our dataset, we evaluated 5 common state-of-the-art methods. We evaluate two simple keypoint-based algorithms from [71], we used the implementation provided by the authors with the default configuration. We also evaluated three modern methods. BusterNet [109] which is a Deep Learning based method, FE-CMFD [99, 120] which is a keypoint-based method and SIFT-LDM [2] which is a hybrid method. For BusterNet, we used the implementation and the pretrained network provided by the authors. We did not use the source/target distinction of BusterNet but instead applied the method they used to evaluate on the CoMoFod dataset (i.e. every pixel below 0.5 in the blue channel is considered as pristine). For FE-CMFD we used the implementation given by the author as is. Finally, for SIFT-LDM, we used the default parameters

Figure 4.6 : From left to right : Tampered images, Ground truths, SIFT [71], SURF [71], BusterNet [109], FE-CMFD [99], SIFT-LDM [2]

described in [2]. We cannot share the implementation of SIFT-LDM but we will provide an executable upon request if needed to reproduce the results present in this chapter.

4.9.2 Metric

For each algorithm, we studied the performances at two levels. At the image level, we evaluate how many images are considered as tampered. An image is marked as tampered if at least one pixel in the detection map is labelled as tampered. In a practical situation, we would like to minimise as much as possible false detection at an image level if the algorithm is meant to be automatic because even a few false positives pixels would then possibly imply a manual verification. We also evaluate algorithms at a pixel level to assess the localisation performances.

Image level

For image-level evaluation, we used the True Positive Rate (TPR), the False Positive Rate (FPR) and the Matthew Correlation Coefficient (MCC). Ideally, all detectors should be able to maintain a high TPR and MCC while keeping the FPR low.

Pixel level

For pixel-level performances, we only computed the scores on tampered images because we are only interested in localisation performances when an image is tampered.

Because the tampered area is small in comparison to the size of the image, the number of true negatives is much higher than the true positives. We thus decided

Method	TPR	FPR	MCC
SURF [71]	0.7919	0.7697	0.0236
SIFT [71]	**0.9676**	0.9145	0.1104
BusterNet [109]	0.1601	0.1607	-0.0006
FE-CMFD [99]	0.0246	**0.0066**	0.0561
SIFT-LDM [2]	0.7917	0.0197	**0.6847**

Table 4.4 : Image-level scores

Method	TPR	FDR	F_1	MCC
SURF [71]	0.2155	0.9792	0.0378	0.0606
SIFT [71]	**0.6004**	0.9610	0.0731	0.1471
BusterNet [109]	0.0016	0.9979	0.0018	0
FE-CMFD [99]	0.0341	0.3114	0.0650	0.1530
SIFT-LDM [2]	0.2555	**0.0541**	**0.4024**	**0.4912**

Table 4.5 : Pixel-level scores

to evaluate the pixel-level performance in terms of True Positive Rate (TPR), False Discovery Rate (FDR), F_1 score and the Matthews Correlation Coefficient (MCC).

4.9.3 Results

Image-Level

Table 4.4 gives the results at image level. Of all the method, we can see that SIFT [71] has the highest TPR (at 0.96) which indicates the best detection performances. SURF [71] and SIFT-LDM [2] follow with an already much lower result (about 0.16 below) meaning that both methods were either unable to detect many of the tampering or decided to reject them. Far behind, we can see that BusterNet [109] and FE-CMFD [99] seems to barely detect the tampered images or are stricter than previous method.

The answer lies in the FPR results for every method. We can first notice that for SIFT, SURF and BusterNet the FPR is almost equal to the TPR. For SIFT and SURF we thus understand that they both struggle with SGO and tend to classify every image as tampered, which yield a high TPR and FPR. For BusterNet the

result is different as the TPR and FPR are both quite low (about 0.16). This probably indicates that BusterNet is unable to detect the forgeries at this small scale, the detection is then more or less random. Even though BusterNet is unable to detect the forgeries, it manages not to trigger too many false alarms which is a good property. The result of BusterNet is overall not surprising as it has never been trained with really small forgeries. FE-CMFD [99] maintain the lowest FPR of all methods but was not able to detect many forgeries. As for BusterNet, the issue most likely comes from the size of the tampering. Though it only detected a few tampered images, it did almost no mistake which could make it practical if a lower FPR is more important than detection performances. For example, in a fully automatic pipeline where no human intervention is possible, one might prefer to miss many tampered images to provide a better user experience. Finally, SIFT-LDM [2] achieve the best overall performances with a TPR of 0.79 and an FPR of 0.02. Even though this result is far from perfect, it is encouraging as SIFT-LDM was designed to tackle both challenges mentioned in 4.7 and seems able to achieve satisfactory results. With time we can expect to achieve even better results.

Pixel Level

For the pixel level evaluation, algorithms have been applied to all tampered images. Then the total number of true positives, false positives, true negatives and false negatives pixels are computed. We evaluate the pixel-level performance across all images as some scores might be undefined in certain conditions (e.g. FDR is undefined for $TP + FP = 0$). Because the images of our dataset contains many SGO and the tampered area are relatively small, we are interested in a particular result. First we would like the true positive rate to be as high as possible while maintaining the lowest possible false discovery rate.

We can first see how both SIFT and SURF method seems able to locate most of the forgeries (see TPR in Table 4.5) but generate a large amount of false positive resulting in an extremely high FDR making both detectors completely impractical. This confirms the tendency observed in Table 4.4.

For BusterNet, we can see that the True Positive rate is almost 0 while the FDR approaches 1. This corresponds to the observed result in 4.4. Because of the

small size of the tampered areas, BusterNet seems unable to detect the forgery in the first place thus the TPR is extremely low. With only a few false detections, the FDR, on the other hand, quickly rises to 1. Those false detections could probably be discarded with a stricter decision threshold (i.e. greater than 0.5).

For FE-CMFD, we observe a similar behaviour as BusterNet with a low true positive rate. This might be due to the small size of the tampering. But unlike BusterNet, FE-CMFD still manage to maintain a satisfactory FDR. In practice, this would mean that FE-CMFD does not rise to many false alarms which is a desired property. The F_1 score obtained (0.0653) and the MCC (0.1571) still remain quite low because of this strong tendency to reject true positives.

Finally, SIFT-LDM achieve the best performance of all methods with an F_1 score of 0.4024 and an MCC of 0.4912. The better results are explained by the ability to maintain a much lower FDR than all other (5 times lower than FE-CMFD) while discarding less true positives (7 time less than FE-CMFD).

For both FE-CMFD and SIFT-LDM, we see that the FRD is quite low, this indicates that both methods highlight the true tampered area. This is important to notice as the results in Table 4.4 could have been misleading. For instance, in Fig 4.6 we can see that the detection does not necessarily correspond to the true tampered area. Which means that an algorithm could correctly labelled an image as tampered even though it did not detect the forgery. The slightly higher FDR of FE-CMFD is mostly due to the final post-processing on the detection maps which tends to produce a region that is always slightly bigger than the ground truth.

4.10 Conclusion

Copy-move has been well studied over the last decade. Thanks to that, many methods and datasets have been publicly released. Unfortunately, we believed that the challenge of similar but genuine objects has been and still is widely under considered. Consequently, many methods have been shown to be ineffective in the presence of SGO.

In this chapter we wanted to develop a copy-move detection method applicable to ID document forgery. We faced the challenge of SGO and confirmed the dif-

ficulty for many state-of-the -art methods to handle such images. While current methods are able to detect large Copy-move with various transformations (rotation, scaling, lightning changes ...) with ease, they struggle to detect even the simplest copy-move (no rotation, no scaling) when it is small and in the presence of SGO.

We decided to evaluate the benefits of adding an extra filtering step based on the local dissimilarity transform to the typical keypoint-based detection pipeline. We showed that we were able to achieve perfect results in common datasets with a rather trivial keypoint-based detection scheme and that we were able to drastically reduce the false positive rate on the COVERAGE dataset which contains many SGO. This first results were really encouraging as they showed that it was possible to handle SGO by redesigning common CMFD algorithms with those in mind.

To contribute to the research on copy-move detection in the presence of SGO, we decided to propose a new dataset. We focused on ID documents for two main reasons. Firstly, the thesis is oriented towards the detection of forgery on ID documents. Secondly, we think that ID documents are both a realistic use case of Copy-Move forgery but also an extremely challenging one. In fact, we saw that the studied state-of-the-art methods were either producing a lot of false positives or completely unable to detect the forgeries. We also evaluated our method on this dataset. We saw that it performed significantly better than other state-of-the-art methods but there is still a great room for improvement.

It is important for future research to acknowledge the challenge of SGO and small copy-move. Nowadays researches are still often evaluated on datasets that are not representative of a typical copy-move forgery. Most of those datasets contains extremely large forgeries in images with no SGO. In practice, Copy-Move is rarely used in such an obvious way. Most of the time it will be used as a way of performing an Object-Removal forgery or to tamper some text. In both scenarios the forgery will most likely be rather small in images with potentially many SGO. Having more datasets containing SGO is also becoming crucial as many current methods achieve perfect results on many of the public datasets.

CHAPTER 5

Object-removal forgery detection

5.1 Introduction

We began by studying Copy-Move forgery detection. As we stated, copy-move is a simple and widely use tampering technique and is particularly appropriate for tampering an ID document. As we briefly mentioned, copy-move is sometimes used to remove objects from an image. Those kinds of forgeries are called Object-Removal.

Object-Removal forgery is arguably one of the most used tampering techniques. In fact, it is often used alongside with other techniques. For example, you often need to remove a few elements before inserting a new one.

There exist multiple methods to perform an object-removal forgery. As already mentioned, one can perform multiple copy-move forgeries to hide the unwanted elements. On a professional software such as Photoshop or Affinity Photo, one would use the clone stamp tool to perform such a forgery. Another common approach is to use an inpainting algorithm to fill the area we want to hide. One would use tools such as the healing brush tool or the content aware fill in Photoshop. Such tools are generally based on some examplar-based inpainting algorithms [121, 122]. Those work much like the clone stamp tool but the selection of the patch to be duplicated is fully automated. For this reason, they generally don't handle very well scenes with complex backgrounds. Recently, deep learning based approaches to perform object-removal have been more studied to tackle more complex backgrounds, but at the moment those have not yet been implemented in professional software. The counterfeiter will in practice choose the appropriate tool based on the circumstances. For example, the inpainting-based algorithm will work extremely well for objects on simple backgrounds whereas for more complex background a more manual approach using the clone stamp tool will be preferred.

Apart from deep learning based inpainting method, object-removal is essentially a kind of copy-move forgery. While it is true and some CMFD can detect object-removals, there exist cases in which CMFD are inappropriate. In this chapter we are interested in the detection of object-removal forgeries. We would like to be able to detect such forgery regardless of the technique used to perform it. When performing an object-removal forgery, the counterfeiter is recreating some background texture. To seamlessly blend it with the rest of the image, it is common to

use a smooth transition between the crafted texture and the original background. This blending acts like a smoothing operation around the forged area which may serve as a clue for object-removal detection. In this chapter we will thus explore the possibility of exposing those smooth areas to reveal possible traces of forgeries. We will show that other operation such as splicing or resampling can sometimes produce similar artefacts and can be exposed using this method.

We will first briefly cover the various methods to perform an object-removal forgery and the implication of those. Then we will propose a novel method to reveal traces of object-removal forgery by exposing those smooth transitions. We will also show that in some cases our method can expose splicing forgeries by revealing abnormally smooth or sharp region in an image and can also help to perform source-target identification for copy-move forgeries.

5.2 Object-removal forgery

We will first give a few examples of the typical techniques used to perform an object removal forgery.

As we mentioned, there exist two commonly used tools to perform an object removal forgery. The first well-known tool is the clone stamp which is present in almost every image retouching software. In Fig. an example of the use of the clone stamp is given. When using the clone stamp, the digital artist first choose a reference point. He can then duplicate the pixel around this reference point using what is called a brush in most retouching software. A brush has two main properties, its size and its softness. In Fig. 5.1 the brush size is represented as the red circle and the softness can be seen in the middle as the smooth transition around the duplicated region. The tampering process is quite similar when using the healing brush or the content aware fill. The main difference is that the artist does not have to choose the reference point. Rather, the user uses the brush to draw the area he wants to remove and the inpainting algorithm automatically fills that area.

In both cases, the artist often use a so-called brush during the tampering process. The softness of the brush creates a smooth transition around the forged

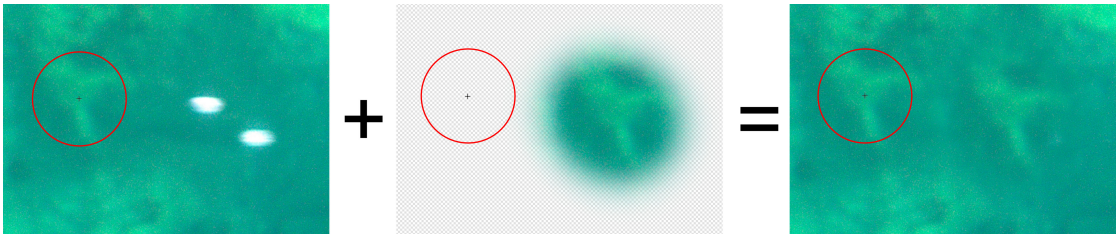


Figure 5.1 : Clone Stamp Tool Usage

area. On the boundary of the forgery, the tampered pixels and the original pixels are thus blended together. We expect this to reduce the sharpness of the texture locally and we will try in the next section to reveal this effect.

5.3 Related Works

Object-removal has been studied through different approaches. As stated in section 5.2, this kind of forgery relies on three common tools that can be found in most retouching software. In a sense, all those tools perform a kind of copy-move forgery at a smaller scale.

Copy-Move has been widely studied over the past years. Two main methods can be distinguished: the block-based and the keypoint-based approach. Block-based [96, 97, 94] method cut the images into smaller regular or non-regular blocks. Then a measure of similarity is performed between each block which allows revealing abnormally similar blocks. Keypoint-based methods [2, 99, 71] work on a similar principle except that keypoints are match instead of blocks. Fewer articles focus specifically on the detection of exemplar-based inpainting [123, 124], which is used by the healing brush tool and the content aware fill.

In this chapter we propose a novel approach that can be used in conjunction with previous method to detect object removal forgery. As described in section 5.2, digital artists often use a soft brush or a smooth transition to produce a more convincing tampering. In this chapter we try to expose this artefact as a trace of object removal instead of finding the duplicated regions. Used in conjunction with a traditional copy-move forgery detection (CMFD) algorithm, this allows performing source/target identification and can also improve interpretability of

CMFD results when duplicated regions are small and scattered. We will also show how this method sometimes applied to other forgeries.

5.4 Proposed feature extraction

In the following section, we will describe the method used to measure the local sharpness of a texture relative to the rest of the image.

5.4.1 Image reflectance estimates

A simple model of image formation is the illumination-reflectance model. In this model, the image is modelled as the product of illumination and reflectance.

$$I(x, y) = L(x, y) * R(x, y) \tag{5.1}$$

where L is the illumination, R the reflectance, I the image and (x, y) the pixel coordinates. In this simplified model, the illumination L is the amount of light coming to the surface visible at (x, y). The reflectance R is the capacity of the surface (x, y) to reflect the incoming light L.

In most contexts, L varies relatively slowly with respect to R hence it has been proposed [125] to extract the reflectance R using a homomorphic filtering. By taking the logarithm of (5.1), we separate L and R with R lying in the high frequencies.

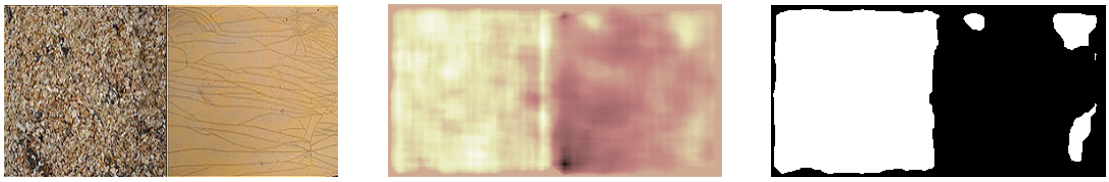
$$\ln(I(x, y)) = \ln(L(x, y)) + \ln(R(x, y)). \tag{5.2}$$

Applying a high-pass filter in the Fourier domain thus isolate R from L.

$$r(u, v) = \mathcal{F}(\ln(I(x, y)))H(u, v) \tag{5.3}$$

with H a high-pass filter in the Fourier domain.

We get our final reflectance estimate $\hat{R}$ by taking the exponential of the invert



(a) Texture image (b) Feature map S (c) Obtained segmentation

Figure 5.2 : From top to bottom : An image with two textures, the feature map S, segmentation by applying Otsu's method on S

Fourier transform :

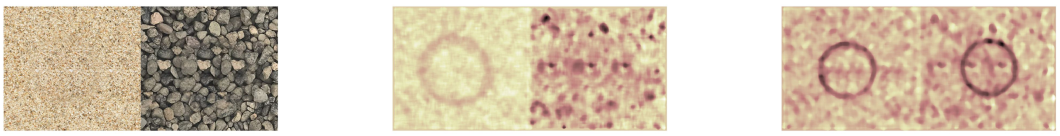
$$\hat{R}(x, y) = \exp(\mathcal{F}^{-1}(r(u, v))). \tag{5.4}$$

5.4.2 Local reflectance variability measure

The extracted reflectance $\hat{R}$ gives us an idea of the surface roughness (x, y). The more $\hat{R}$ varies around a neighbour of (x, y) the rougher the texture is at this location. As a sharpness measure, we use the local standard deviation with a neighbour size n of the reflectance estimate. In all our experiments, n is fixed to 11. From this we obtain an image $\boldsymbol{\sigma}_{\hat{R}}$ with the same size as I.

Once the standard deviation is computed. The image is cut into K non-overlapping segments s_k depending on the dynamic range. For each pixel p_k in the segment s_k, we compute the median absolute deviation of its local reflectance standard deviation $\boldsymbol{\sigma}_{\hat{R}}$ within its segment s_k to obtain our final local reflectance variability measure S. The final result tells us how sharp a pixel is relative to the rest of the image. Large positive values indicate the most variable pixels in the image and large negative values indicate the less variable pixels in the image.

An example of application is given in Fig. 5.2. Given two textures on Fig. 5.2a, the feature map S is computed and a median filter is applied to reduce noise (Fig. 5.2b). The feature map is the binarised following Otsu's method [119] and a morphological closing is applied (Fig. 5.2c). We can see that the left texture is clearly marked as sharper. Also three areas of the right texture are marked as



(a) Tampered image (b) Feature map S (c) Feature map S with noise added

Figure 5.3 : From top to bottom : Tampered image, Feature map of the tampered image, Feature map S with synthetic noise added before tampering

sharp which indeed correspond to sharper regions of the right texture.

In case of a forgery, we expect to observe a notable drop in a relatively uniformly textured area. An example is given in Fig. 5.3b, where we can observe a less sharp ring that does not correspond to anything visible on Fig. 5.3a.

5.5 Remarks

5.5.1 Impact of texture scale

The proposed feature measure the relative sharpness of one pixel relative to the rest of the image. We expect to observe less sharp pixels at the boundary of the forged region if a smooth transition is applied as described in section 5.2. It is important to remark that this phenomenon can only be seen if the transition area is sufficiently greater than the scale of the surface details. An example is given in Fig. 5.3a where the healing brush tool has been used on two textures (sand and gravel). We can see how the forged area is visible in the sand texture (left image), because the transition is much larger than the texture size. Whereas nothing is visible for the gravel texture. Increasing the neighbour size n when computing the local standard deviation of $\hat{R}$ can slightly increase the visibility of tampered region for large-scale texture. But this also largely degrades the detection performance for lower scale texture which is overall not worth it.

5.5.2 Impact of sensor's noise

While extracting the reflectance estimate $\hat{R}$ we do include the image sensor noise. This is important to keep in mind as this noise can be seen as kind of a dominant texture across the image. In the presence of a strong sensor noise, S will mostly measure this noise leading in a more uniform appearance whereas a week sensor noise will lead to a more variable S. This effect is to our advantage as the sensor noise is a small-scale texture and, as previously mentioned, tampered region will be more visible in the area with a fine-grained texture. This positive effect can be seen in Fig. 5.3c where white Gaussian noise is added on top of the textures prior to tampering. We can now clearly see the tampering even in the gravel textures as the noise becomes dominant.

To illustrate this effect, three photos of a pen where took (Fig. 5.4) with a smartphone with three different ISO settings. From left to right the ISO are 3200, 800 and 100. We can see how the tampering is more visible with a higher ISO setting. Also note how S becomes more uniform with higher ISO this is because the sensor noise become the predominant texture hence the sharpness measure is mostly defined by the sensor noise.

5.5.3 Impact of post-processing

We tested the impact of JPEG recompression and resampling on our method which are two extremely simples and common post-processing. In both cases, the tampering is performed on the raw image and first saved to JPEG with a quality factor of 90.

In Fig. 5.5, we recompressed the tampered image using different quality factors (i.e. 80, 70, 50 and 20). We see that the tampering progressively disappear until it becomes barely noticeable for a quality factor of 20. This behaviour is expected as JPEG will tend to suppress more and more detail which leads to the sharpness measure being almost null on the whole image. It is interesting to notice that if the detection was possible at the creation of the tampering, it will stay visible even with extremely low quality factors (i.e. 50) by nowadays standard.

We then tested the detection against resampling. In this case, the tampered

image in Fig. 5.6 is progressively downsampled to a factor of 0.1. Once again we can see that the detection becomes less and less visible. As for JPEG recompression, the detection still remains possible for low resampling factors (i.e. 0.5).

Those experiments show that the detection is quite resilient to common post-processing. In the next section, we tested our method of real-world examples to assess its applicability.

5.6 Qualitative results

As the method is quite specialised and mostly perceptual, we present here some qualitative results. We made the first example from a JPEG image that directly comes from the camera. To assess the applicability of our method of real-world images, we also tested it on three images from the PhotoshopBattles reddit forum on which participants are asked to retouch a given image.

Because we compute the median absolute deviation of each pixel p_k to produce the feature map S, we can observe strong positive deviation at edge location. As we are only interested in finding smoother regions of the image (i.e. large negative values in S), we can set every positive value to 0 in our feature map S. This will significantly reduce the noisy appearance of S and further enhance the tampered region. This step is applied to all the following examples (Fig. 5.8).

The first image has been taken in low light condition, as stated in 5.5.2 this causes our feature map S to mostly characterise the sensor noise leading to a more uniform result and will make the tampering more visible even if the transition size is small. On Fig. 5.8d we can see a large darker region where the woman has been removed. This matches the ground truth on Fig. 5.8c. This example was made using the clone-stamp tool in Affinity Photo and shows how the recreated background appears less sharp in the forged area.

The other examples have been found on the PhotoshopBattles reddit. For each example we found the original image and recreate an approximate ground truth.

On the first example from reddit (Fig.5.8e-5.8h) the artist[1] had to remove the character from the original image to insert the kid. The feature map Fig. 5.8h

[1]Artist : kann_i, Link : `https://www.reddit.com/r/photoshopbattles/comments/hb01hh/psbattle_a_girl_staring_at_her_neighbor_mowing/`

reveal those two steps as we can see a region that is considerably smoother than the rest of the image just around the kid. Also, because the inserted kid comes from an image with a much lower quality, we can see how he also appears much smoother than the rest of the image.

In the second example, the artist[2] removed the legs and added a hand. Once again large smooth region appears on Fig. 5.8l where the snow texture has been reconstructed. One might notice some circular patterns appearing on the feature map. Those are likely due to the correction of a lens distortion. In fact, looking at the buildings on the original image (Fig. 5.8i), their curvy appearance indicates that the picture was taken with a wide-angle lens which causes what it is called a barrel distortion. While correcting this distortion, a resampling is performed on some pixels which decrease the variance of the sensor noise and might be the source of those circular pattern. To confirm this thought we apply this correction on the photo of the pen with the highest ISO and observed similar patterns S (see Fig. 5.7).

The last example is here to illustrate how this method can sometimes be used to detect a splicing forgery. In this example, the artist [3] inserted a woman and the David head onto the well-known painting by Michelangelo. As for the first example, notice how the two inserted elements appear much smoother than the rest of the image. Even though our method is not meant to perform splicing detection, we can see that in some cases the computed feature map S can give a strong intuition on a probable tampering because the quality of the inserted element might not match the quality of the target image, we can either observe an abnormally smooth element like in this example or an abnormally sharp element in the image.

[2]Artist : KrombopulosJeff, Link : `https://www.reddit.com/r/photoshopbattles/comments/gz9lsl/psbattle_man_sat_on_snowy_roof_overlooking/`

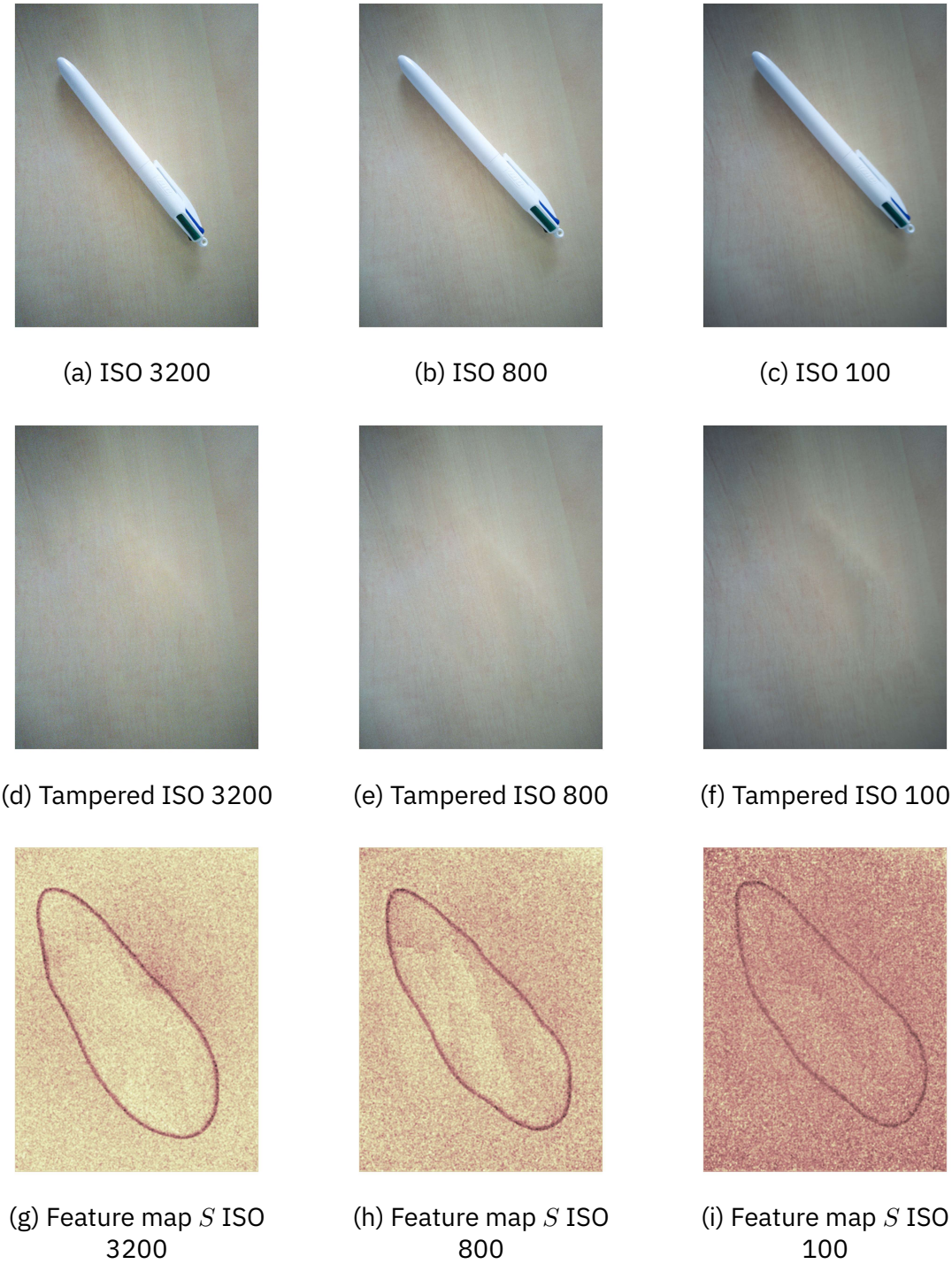
[3]Artist : V_LochNessLobster_V, Link : `https://www.reddit.com/r/photoshopbattles/comments/heqrwr/psbattle_this_woman_with_david/`

5.7 Conclusion

When creating a forgery, a digital artist might have to remove a subject from the scene. To do so, he will sometimes need to recreate parts of the background that were hidden by the subject. Artists will often use tools which are available in most professional software such as the clone-stamp tool, the healing brush tool or the content aware fill. It is common to use a soft brush or apply a smooth transition to enhance the resulting forgery. In this chapter, we suggest that this operation will lead to the apparition of less sharp areas in the image that might give an intuition for the presence of a tampering. To do so, we proposed a relative sharpness measure for a pixel within an image based on the reflectance of that image. We show that it is possible to expose forgeries using this feature under certain circumstances. The image must have sufficient detail for our sharpness measure to work. And the size of the transition area (for the soft brush or the soft transition) must be greater than the underlying texture scale. We show that for modern cameras with reasonable ISO settings those criteria are easily met. Then we observed that the detection is quite resilient to simple post-processing such as JPEG recompression and rescaling which is really important as images are often recompressed or resampled when uploaded on social media. Finally, we assessed the applicability of our method on real examples took on the PhotoshopBattle reddit. While not automated, our method can allow a human operator to get a strong intuition on the possible traces of a forgery. It also allows the operator to locate and interpret what the tampering process was.

We showed that our method allows us to detect object removal forgeries when a soft transition is applied but can also be used in splicing detection. For splicings, our method can sometimes amplify a difference in sharpness (between the inserted element and the rest of the image) or exposed a synthetic noise added to blend more seamlessly an element.

Further works need to be done to enhance the results of our method in order for it to work in an automated manner. As is, it is a quick yet efficient tool for multimedia forensic, in particular if the image provenance if somewhat controlled (e.g. constraints on a minimum required quality).

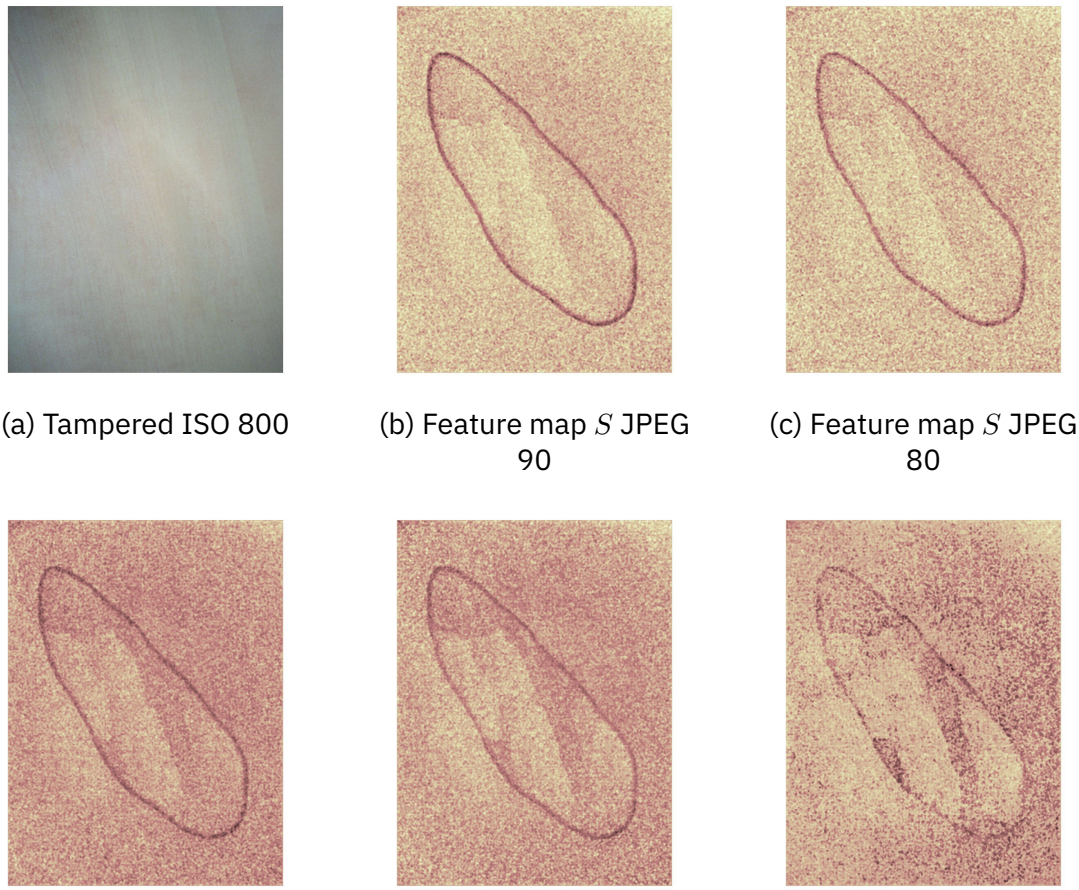


(a) ISO 3200 (b) ISO 800 (c) ISO 100

(d) Tampered ISO 3200 (e) Tampered ISO 800 (f) Tampered ISO 100

(g) Feature map S ISO 3200 (h) Feature map S ISO 800 (i) Feature map S ISO 100

Figure 5.4 : Impact of the sensor noise on the detection



(a) Tampered ISO 800

(b) Feature map S JPEG 90

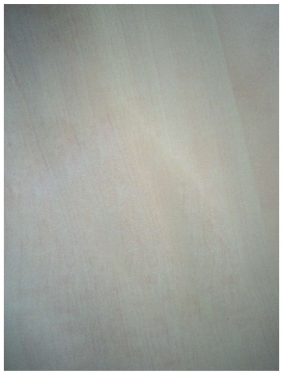
(c) Feature map S JPEG 80

(d) Feature map S JPEG 70

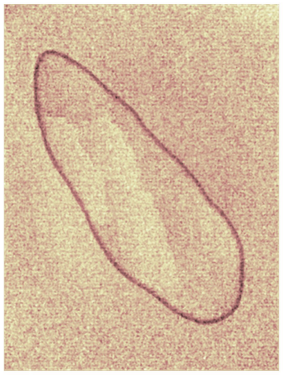
(e) Feature map S JPEG 50

(f) Feature map S JPEG 20

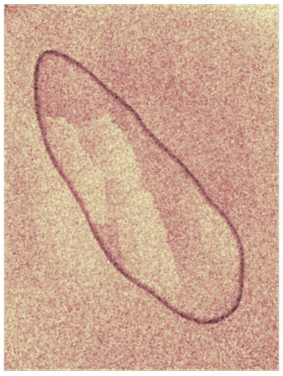
Figure 5.5 : Impact of JPEG compression on the detection



(a) Tampered ISO 800

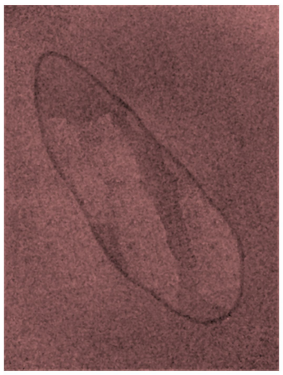


(b) S after resampling by a factor 0.9



(c) S after resampling by a factor 0.7

(d) S after resampling by a factor 0.5

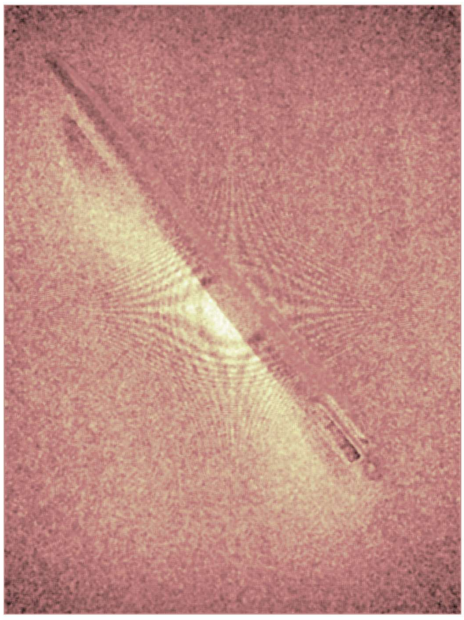


(e) S after resampling by a factor 0.3

(f) S after resampling by a factor 0.1

Figure 5.6 : Impact of resampling on the detection

(a) Lens correction applied

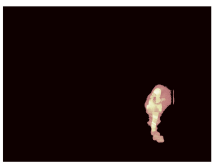


(b) Circular patterns due to the lens correction

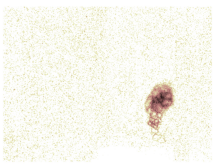
Figure 5.7 : Visible resampling in the feature map S

(a) Genuine image

(b) Tampered image



(c) Ground truth



(d) Feature map S

(e) Genuine image

(f) Tampered image

(g) Ground truth

(h) Feature map S

(i) Genuine image

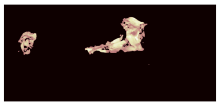
(j) Tampered image

(k) Ground truth

(l) Feature map S

(m) Genuine image

(n) Tampered image



(o) Ground truth

(p) Feature map S

Figure 5.8 : From left column to right column : Original images, Tampered images, Ground truths, Feature map S

CHAPTER 6

Face morphing detection

6.1 Introduction

Face morphing has been proved to be a threat to face recognition software. In [126] the authors showed that two peoples could share a single ID document by submitting a morphed face images during an official ID onboarding procedure. This attack is plausible as [127] shown that humans are not good at detecting face morphing thus making it possible to obtain a fraudulent ID (FID). Once the FID has been acquired, two peoples can use it and pass automated face recognition system tests. The most obvious threat presented by [126] is the ability for two peoples to pass automated border control (ABC) systems with a shared passport. With face biometry being used in many systems and with the rise of digital identity, the ability to detect face morphing is becoming crucial.

Two kinds of Face Morphing Detectors (FMD) can be differentiated. The no-reference FMD and the differential FMD.

For the differential FMD, the task is similar to Face Recognition (FR). Having a Bona Fide image of a person A, we want to decide if a given photo of the person A is a Face Morph (FM). Unlike FR, differential FMD does not have to be able to recognise A in any complex scene (e.g. complex lightning condition). It can be assumed that both the Bona Fide photo and the suspect photo A will have a sufficient quality and may be taken similarly (e.g. frontal with a neutral expression). In [128], authors propose to demorph the picture on the FID then check if the face recognition system still match the two pictures. Authors of [129] propose to study the difference in the landmarks between the two photos and to use an SVM to decide if the difference is due to a pose and expression variation or due to a face morphing attack. Other approaches [130, 131, 132, 133, 134] uses deep learning methods to detect the face morphing attack.

For the no-reference FMD, no Bona Fide is available. The detection is blind in a single image. The task becomes much harder as no history of the image can be assumed, the photo could have been printed and scanned for instance. No-reference FMD would be used when submitting a photo to obtain an ID document to validate the integrity of that photo before creating a document. Many methods exist [135, 136, 137, 138, 139] and addresses very different artefacts. In [140] authors propose to train an SVM on texture descriptors to detect the Face Morphing attack. In

[141] authors propose to use the sensor pattern noise (SPN) to detect the face morphing attack, they use a SVM trained on features extracted from the Fourier spectrum of the SPN to decide if the image is a morph. Authors of [142] proposed to use the photo response non-uniformity (PRNU) and analyse the difference of variance of the PRNU across the image to detect the face morphing attack.

In this chapter we present the Face Morphing creation process and how this affects the overall image. We show how this can be used to create a simple yet efficient no-reference FMD for common JPEG quality factor. We then analyse how this FMD can be fooled and show that many no-reference FMD suffer from similar flaws. Finally, we propose to analyse FMD in various scenarios, and show that performance can vary with Bona Fide image provenance and quality raising the question of the applicability of no-reference FMD for Blind Face Morphing Detection.

6.2 Problem Statement

Two main FMD exists. The no-reference and the differential detectors. For both kinds of detectors, the same attack scenario is considered. In [143], authors proposed a formal document lifecyle modelling. They decomposed the life of a document into three steps. The image acquisition, the document issuing process and the document usage. As they explained, Face Morphing attacks could occur during two steps. The attacker could inject the morph at the document issuing process or could manipulate the document obtained with a genuine photo to produce a forged id. In those two cases, no-reference FMD cannot make precise assumptions regarding the photo history. Whether the verification is performed before the document generation or at the usage. In this chapter we will consider this task as Blind Face Morphing Detection (BFMD) as opposed to Controlled Face Morphing Detection (CFMD) where the acquisition pipeline is controlled. For both BFMD and CFMD, cross-dataset performance and performance against simple counter forensic has been identified as an issue [144][143]. In this chapter, we will formally present the fully digital Face Morphing attack and introduce a simple method for CFMD that requires no training and performs well against various simple post-

(a) Face A (b) Face B (c) Rough alignment of faces A and B (d) Precise alignment of faces A and B (e) Without colour balance (f) With colour balance

Figure 6.1 : Automatic Face Morphing creation

processing. We will show that, as other FMD algorithms, our method suffers from targeted counter-forensic method. We will then consider another potential pitfall for FMD by looking at performance variation against varying Bona Fide provenance and quality. We will thus discuss on the applicability of no-reference FMD for the BFMD task.

6.3 Face Morphing Attack

We will begin by introducing the basic steps to create a Face Morph. We will consider two face images I^A and I^B where I^A is inserted into I^B.

6.3.1 Automatic Face Morph creation

Landmarks extraction and face alignment

The first step is to extract a set of facial landmarks $L_A = \{l_{a,0}, l_{a,1}, ..., l_{a,n}\}$ (respectively L_B) for the two faces we want to morph. To extract the landmarks we use the implementation of [145] in the Dlib [146] library. Once the two sets of landmarks are extracted, we can proceed with the alignment of the two faces. The two faces must first be roughly aligned. To do so, one of the faces is aligned with respect to the eyes of the other with a similarity transformation to keep the facial geometry untouched. When the two faces are aligned a weighted average of the

landmarks of L_A and L_B is computed to produce a set L_W with

$$l_{w,i} = \boldsymbol{\alpha_l}_{,i} l_{a,i} + (1 - \boldsymbol{\alpha_l}_{,i}) l_{b,i}, \boldsymbol{\alpha_l}_{,i} \in [0, 1] \tag{6.1}$$

$\boldsymbol{\alpha_l}$ being the shape vector determining how much of each faces biometry is kept and would be set by the attackers to produce a Face Morph convincing enough to fool a human inspector. In the rest of the chapter, $\boldsymbol{\alpha_l}$ is considered to be a constant α_l for all the detected landmarks.

The two faces are then warped to L_W before performing any kind of blending to avoid ghost artefacts.

During that first step of the Face Morphing creation process, the two images I^A or I^B will be interpolated many times. We can thus except that they will have a lower noise variance due to the multiple interpolations. From now on I^A and I^B denotes the two face images after being warped.

Skin tone matching

Prior to the blending, a skin-tone matching step can be applied which allows to produce much better morph in more cases.

We perform a local RGB scaling on I^A to match I^B colours. Rough estimates of I^A and I^B expectation are first computed by applying a Gaussian blur, this results in the two images approximations A^A and A^B. Each pixel I^A is scaled to obtain the skin tone matched image B^A

$$B^A = I^A \circ (A^B \oslash A^A) \tag{6.2}$$

where $\circ$ is the Hadamard product and $\oslash$ the Hadamard division.

Blending

Once the skin tone is matched, the two images can be blended. A simple alpha blending is performed.

Given an alpha matte $\boldsymbol{\alpha_b}$ with $\boldsymbol{\alpha_b}_{i,j} \in [0, 1], \forall (i, j)$. The final morph image M^{AB} is given by

$$M^{AB} = \boldsymbol{\alpha_b} \circ B^A + (1 - \boldsymbol{\alpha_b}) \circ I^B. \tag{6.3}$$

In the rest of the chapter, we assume that $\boldsymbol{\alpha_b}$ is a constant α_b inside the facial area. In practice, the exact alpha matte $\boldsymbol{\alpha_b}$ cannot be known. One could decide to only blend important features such as the nose, eyes. It is worth noticing that this definition of Face Morphing contains many other possible attacks. For instance, a simple face swap is performed by setting α_l to preserve I^A biometry and α_b to only keep I^A texture. One might want to keep I^B biometry but with I^A texture. In [128], authors showed that α_b have more impact on FR systems which suggest that, as for us, FR systems tend to be more sensitive to the textural information. Further study has yet to be done when considering an alpha matte $\boldsymbol{\alpha_b}$ and a shape vector $\boldsymbol{\alpha_l}$ instead of constants. In fact, the attacker might not use automatic tools to produce the face morph and would instead try to replicate those steps manually using photo retouching software. In such cases, α_l and α_b would not be constant. Those cases are later called handmade morphs.

6.3.2 Dataset construction

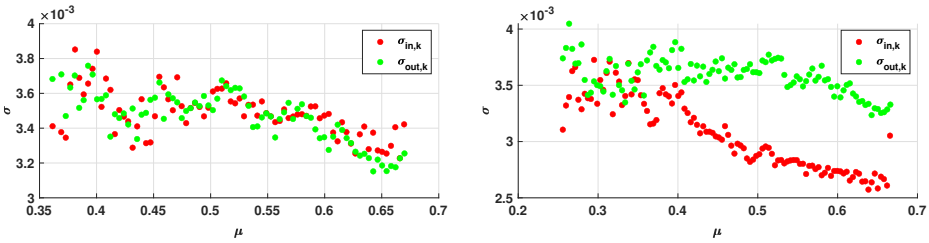
For the rest of the chapter, we constructed two datasets of face morphing attacks according to the process described previously. All the morphs are generated with $\alpha_l = 0.5$ and $\alpha_b = 0.5$.

PUT Morph Dataset

The PUT dataset [147] is composed of 100 subjects. Images of each subject has been taken under various angles but with consistent illumination conditions. For each subject we selected the more frontal and neutral image as reference images. From those references 171 morphs have been generated. Each morph image is then compressed at different JPEG quality factors (i.e. 100, 95, 90, 85, 80, 75).

FERET Morph Dataset

The FERET dataset is a well-known face recognition set composed of 14126 photos of 1199 individuals taken in 15 sessions between 1993 and 1996. At the time, photos were taken on a film camera and then digitalised. We took a subset of those photos to construct two face morphing sets (as in 6.3.2) for training and



(a) $\sigma_{in,k}$ and $\sigma_{out,k}$ estimates on a genuine face

(b) $\sigma_{in,k}$ and $\sigma_{out,k}$ estimates on a morph face

Figure 6.2 : $\sigma_{in,k}$ and $\sigma_{out,k}$ estimates

validation. Original and Morph images are compressed with various JPEG quality factors (i.e. 100, 95, 90, 85, 80, 75). The training set consist of 919 Bona Fide images and 900 morphs. The validation set consist of 1361 Bona Fide images and 992 morphs. The training set and the validation set contains no overlapping identity as recommended in [148].

6.4 Noise based Face Morphing Detector

To detect the Face Morphing attack, we make the assumption that the process described in 6.3.1 alters the noise residual of the resulting image. In this section we will briefly describe how the noise residual is extracted, introduce some notations used in the rest of the chapter, show the effect of 6.3.1 on the noise residual and present a simple yet reliable method to detect such an effect.

6.4.1 Homogeneous block detection

Given an image I, the image approximation A is estimated using a wavelet denoising method. The image residual is then computed as $R = I - A$. Having I, R and A, it is proposed to use [149] to detect homogeneous blocks that can be used to get a correct estimate of the noise variance.

6.4.2 Level-set variance estimation

Once homogeneous blocks are detected, the image is partitioned into K non-overlapping segments S_k by dividing the dynamic range. For each segment S_k, pixels belonging to segment S_k in the image I will be denoted as $I_{k,i}$ with $k \in 1, ..., K$ the k^{th} segment and $i \in 1, ..., n_k$ the i^{th} pixel of the segment k with n_k the cardinality of S_k.

For a segment S_k, the expectation μ_k is estimated from A_k and the variance σ_k from R_k.

With a sufficiently large value of K, $R_{k,i}$ become more probably identically distributed thus for n_k large enough $R_{k,i}$ follows a normal distribution.

We further separate pixels from inside and outside the facial area to get two variance estimates for each segment S_k. In the rest of the chapter, the pixels of a segment S_k of image I inside the facial area will be denoted as $I_{in,k,i}$ and those outside the facial area $I_{out,k,i}$. Similarly the inside variance estimate of segment S_k will be denoted as $\sigma^2_{in,k}$ and the outside variance estimate $\sigma^2_{out,k}$.

6.4.3 Effect of Face Morphing on variance estimates

The Face Morphing process described in 6.3.1 can be seen as a kind of splicing forgery.

Unlike a typical splicing forgery, some knowledge can be use to our advantage. The first one being the location of the forgery which we exploit by comparing the inside from the outside area. The second being that both the inserted element and the insertion location are interpolated prior to a final alpha-blending step.

From (6.3) we have that

$$M^{AB}_{in,k} = \alpha_b B^A_{in,k} + (1 - \alpha_b) I^B_{in,k}. \tag{6.4}$$

Meaning that

$$R_{in,k} = \alpha_b R'_{in,k} + (1 - \alpha_b) R_{out,k} \tag{6.5}$$

Table 6.1 : **EER** on the PUT morph set with varying morphed image JPEG quality factors and counter-forensic (CF) applied

EER at JPEG	100	95	90	85	80	75	CF
LBP	11.20	37.98	59.98	77.82	85.67	73.26	**22.86**
BSIF	26.86	5.86	**1.19**	5.86	11.21	**13.87**	41.63
LPQ	4.26	54.01	33.07	37.05	58.17	53.06	27.43
Proposed method	**0.59**	**0.37**	2.35	**4.73**	**8.48**	21.08	35.86

with R' being the residual of B^A.

As stated in 6.4.2 $R_{k,i}$ (and $R'_{k,i}$) follows a normal distribution, this leads to

$$\sigma^2_{in,k} = (\alpha_b \sigma'_{in,k})^2 + ((1-\alpha_b)\sigma_{out,k})^2. \tag{6.6}$$

With $\alpha_b \in [0,1]$ we have $((1-\alpha_b)\sigma_{out,k})^2 < \sigma^2_{out,k}$. Thus $\sigma^2_{in,k}$ can only be greater than $\sigma^2_{out,k}$ if we have $(\alpha_b \sigma'_{in,k})^2 > \sigma^2_{out,k} - ((1-\alpha_b)\sigma_{out,k})^2$. While this is possible, it is unlikely as B^A_{in} and I^B_{in} are interpolated pixels to begin with.

On Fig. 6.2a we can see that both $\sigma_{in,k}$ and $\sigma_{out,k}$ estimates are similar. While in the presence of a Face Morphing attack $\sigma_{in,k}$ are significantly lower than $\sigma_{out,k}$. As a result, the mean of $\sigma_{in,k}$ and $\sigma_{out,k}$ estimates will vary greatly. We can thus use a two-sample t-test to get the p-value of observing $\sigma_{in,k}$ and $\sigma_{out,k}$.

6.4.4 Remarks

A few remarks must be made regarding this detection method. The first assumption made is that the noise variance should drop in the facial area during the warping steps 6.3.1. This should be generally true, but the extent to which the variance will drop cannot be known as it depends on the interpolation method used and also the warping technic used. In particular, handmade morph might not be as warped as automatic ones as a digital artist can apply precise deformation only to the critical areas (e.g. nose, mouth, eyes). The second assumption we make is on the alpha matte $\boldsymbol{\alpha_b}$. We assume it to be constant in the facial area. In practice it is not hard to vary $\boldsymbol{\alpha_b}$ to preserve the skin texture of one of the two people for example and only blend facial features like the nose, eyes or the mouth. Furthermore, for this method to work, a sufficient number of pixels per segment

S_k must be available to compute a correct estimate of $\sigma_{in,k}$ and $\sigma_{out,k}$. In the rest of the chapter, we only compute $\sigma_{in,k}$ and $\sigma_{out,k}$ if we have at least 100 samples.

Finally, the assumption is made that images are taken with modern camera hardware which follows the generalised noise model in [149]. While this is a reasonable assumption, we will see that detection performances are affected when this assumption does not hold (see 6.5.2).

6.4.5 Noise based FMD Detection performance

To evaluate our approach, we constructed a dataset of Face Morphing attacks as described in 6.3 and evaluated the performance against JPEG compression for various quality factors (100, 95, 90, 85, 80, 75).

As recommended in [148], we evaluate the detection performance in terms of Attack Presentation Classification Error Rate (**APCER**) and Bona Fide Presentation Classification Error Rate (**BPCER**).

APCER being the proportion of attack presentations incorrectly classified as bona fide presentations in a specific scenario and **BPCER** being the proportion of bona fide presentations incorrectly classified as presentation attacks in a specific scenario.

The Equal Error Rate (**EER**) where **APCER**=**BPCER** will be used in the rest of the chapter as a performance metric.

Results for the Noise based FMD can be seen in Table. 6.1.

6.4.6 Iterative residual correction

In 6.4, we showed that a Face Morphing attack can be reliably detected by comparing the noise residual inside and outside the facial area.

This method suffers from a direct flaw. One could correct the noise discrepancy to hide the traces of a Face Morphing. In this section we will introduce a quick counter-forensic method that greatly reduces our FMD performances.

As shown in [149], the noise variance is a non-linear function of the image expectation. To fool the detector proposed in 6.4, the noise variance must then be corrected per level of intensity.

Algorithm 1 Iterative residual correction

Input: I
Output: C
$C \leftarrow I$
Get $\sigma_{out,k}$, $\sigma_{in,k}$ from $residual(C)$
$p_n \leftarrow ttest2(\sigma_{out,k}, \sigma_{in,k})$
$p_{n-1} \leftarrow p_n$
while $p_n \geq p_{n-1}$ **do**
  $R \leftarrow residual(C)$
  for all segment S_k **do**
    Get $\sigma_{out,k}$, $\sigma_{in,k}$ from R
    Compute m_k using (6.7)
    Update R according to (6.8)
  end for
  Get $\sigma_{out,k}$, $\sigma_{in,k}$ from $residual(C + R)$
  if $kstest2(\sigma_{out,k}, \sigma_{in,k}) \geq p_n$ **then**
    $C \leftarrow C + R$
    $p_{n-1} \leftarrow p_n$
    $p_n \leftarrow ttest2(\sigma_{out,k}, \sigma_{in,k})$
  end if
end while
return C

The image residual is first extracted as in 6.4. Then for each segment S_k we want to add enough noise to compensate for the morphing operation.

To do so, for a segment S_k, we compute m_k as

$$m_k = \frac{|\sigma_{out,k} - \sigma_{in,k}|}{\sigma_{in,k}}. \tag{6.7}$$

And update the noise residual as

$$R_{in,k,i} \leftarrow m_k R_{in,k,i}, i \in \{1, ..., n_k\}. \tag{6.8}$$

Those steps are repeated until the p-value obtain from 6.4.5 stops increasing. In practice the algorithm 1 converges from three to four iteration.

6.4.7 Detection Performance after Counter Forensic

To evaluate the impact of the algorithm 1, we applied it on every image with a quality factor of 100 of the dataset 6.3.2 and to produce the results in Table. 6.1.

We can see in Table 6.1 that the Algorithm 1 greatly reduces the proposed detection method performances.

Table 6.2 : **EER** on the FERET morph set with varying morphed image JPEG quality factors, counter-forensic (CF) and sharpening (SH) applied

EER at JPEG	Raw	100	95	90	85	80	75	CF	SH
LBP	9.74	10	14.90	14.57	13.97	13.14	13.22	41.49	57.97
BSIF	5.32	5.78	6.20	**6.09**	**6.65**	**7.29**	**7.96**	13.20	26.2
LPQ	**5.30**	**5.24**	**6.05**	8.37	9.80	11.79	14.71	**9.30**	16.94
Proposed method	6.66	7.12	8.46	13.57	19.74	25.46	31.41	35.10	**9.66**

6.5 Evaluation of no-reference Face Morphing Detectors

In this section, we have implemented three state-of-the-art methods and evaluated them against our simple FMD. We will first study the performances of all the algorithms for various Face Morphed image quality. We will then vary the Bona Fide image quality to assess the impact on detection performances. Finally, we will evaluate all algorithms in a mixed dataset scenario.

6.5.1 Implemented algorithms

For comparison, we implemented three common state of the arts algorithms. Two methods from [144] and one method from [140]. which we will briefly introduce. For each method, the face is detected and cropped to only keep the facial area. Descriptors are then computed on the cropped face and the histogram of those descriptors are then used to train a cubic SVM.

LBP-Based detection

For the LBP-based FMD we used the non-rotational invariant descriptors in an eight-pixel neighbourhood [150].

BSIF-based detection

For the BSIF-based FMD we used learnt filters from [151].

LPQ-based detection

For the LPQ-based FMD, we used a 3×3 window size with decorellation and a short-term Fourier transform with uniform window for local frequency estimation, see [152] for more information .

6.5.2 Baseline results

The first evaluation is the typical training/validation performance test. Algorithms are trained on the FERET morphs training set and test on a disjoint test with no overlapping identities as recommended by [148]. Results are summarised in Table 6.2.

Without any counter forensic applied, it can be seen that all method reasonably classifies the Morphing attack with our method being slightly behind.

As stated in 6.4.4, we assume that photos are taken with modern cameras and follow the noise model in [149] which is not true for the FERET images. Photos were taken with a film camera and digitalise with an unknown process. Our fairly good results thus support the assumption made in 6.4.3 as the variance drop is significant enough to be distinguished even with unsure Bona Fide images. As for the test on the PUT set, performances decrease with lower JPEG qualities as it tends to suppress the noise residual.

For the three state-of-the-art algorithm, performances also decrease with lower JPEG quality with LBP being the most impacted.

When the iterative residual correction is applied, our method performance drops significantly which is not surprising as it is a targeted attack. LBP Performances also drop significantly to about 4 times higher **EER**. The impact on both the BSIF and the LPQ is less visible but it still drops the performances by a factor of two. For a naive sharpening (applied to the whole image). All performances drop with the less impacted method being ours.

Those baselines results indicate that while no-reference FMD seems feasible, it can suffer a lot from quick counter-forensic methods. It is worth noticing that in a real case scenario, sharpening would probably be systematically applied by the attacker. In such cases, none of the method could be used with satisfactory results.

Table 6.3 : **EER** on the FERET morph set with varying Bona Fide JPEG quality, external Bona Fide and worst-case scenario

EER for JPEG	Raw	100	95	90	85	80	75	External	Worst case
LBP	3.93	3.42	4.87	8.07	11.71	16.19	18.42	9.43	62.83
BSIF	**1.63**	**1.75**	**2.23**	**3.87**	6.24	8.75	11.90	44.39	63.04
LPQ	2.64	2.78	2.72	4.80	5.82	8.06	8.86	11.79	**30.25**
Proposed method	7.59	7.79	7.24	6.59	**6.02**	**5.45**	**5.20**	**2.82**	37.10

6.5.3 In-Database performance variation

In 6.5.2 we showed that some simple counter forensic methods can have a significant impact on the detection performances of no-reference FMD. Even without considering cross-dataset performances yet. Systematic evaluations of such counter-forensic methods should be done in order to perceive FMD weakness.

Now we propose to look at the detection performances from another point of view. Instead of varying the morphed image quality, we will vary the Bona Fide image quality to assess the impact on the detection performance. Results are summarised on Table. 6.3.

For our method we can see that performance increase while JPEG quality factors of the Bona Fide decreases. Once again, this is due to the fact that lower JPEG quality tends to remove the noise residual, which leads to our detector classifying everything as Bona Fide.

For the three state-of-the-art approach, we can see that the detection performance decreases while the Bona Fide quality decrease. What should be noted is the amplitude of this variation, from about 6% for the LPQ to as high as 14% for the LBP detector. Our detector being more stable with an amplitude of about 2%.

This show that performance detection can be dependent on the Bona Fide images. We believe that this is a strong issue as no prior knowledge on the image can be done in the no-reference BFMD. The most naive Face Morphing attack could then be confused with a whole range of Bona Fide images.

6.5.4 Mixed database performances

In a real case scenario, FMD algorithms would be trained and tested on some datasets then applied to unseen images. In particular, we cannot make any assumption regarding the quality of the Bona Fide images nor their provenance. We saw in 6.5.3 that the Bona Fide quality can noticeably affect the performance of the FMD and this while staying in the same controlled environment as the training step.

We now propose to study the Bona Fide provenance impact on FMD perfor-

mances. On Table. 6.3, the eighth column corresponds to the detection performance for morphs coming from the FERET morphs test set and Bona Fide coming from the PUT set.

In practice, we want to be able to set a threshold between morph images and Bona Fide that holds whatever the dataset used. In other words, morph images from any dataset should have lower scores than Bona Fide from any datasets.

For our detector, we can see that the performance increases. This is due to the fact that PUT Bona Fide images do come from modern hardware (see 6.4.4).

For other methods, performances drop significantly. Meaning that for those no-reference FMD systematic dataset adjustment is required and cross/mixed dataset performances cannot be guaranteed.

6.6 Conclusion

In most current research, no-reference FMD is applied to the Blind Face Morphing detection (BFMD) task. As presented in [143], Face Morphing attacks constrain us to consider BFMD.

With this scenario in mind, two problems arise. The unknown origins of the Face Morphed images and the unknown origins of the Bona Fide images.

When considering the unknown origins of the Face Morphed images, we can imagine that no-reference FMD will have to deal with counter-forensic methods. But as shown in 6.5.2 current FMD might be easily fooled by some trivial post-processing. Some results in Table. 6.2 might not look alarming but one needs to remember that those results are presented for an in-database scenario where the confidence level of Bona Fide images is extremely high. Which means that those simple counter-forensic methods lead to detectors being extremely confident that morphs images are indeed Bona Fide.

The second issue arise when considering the unknown origins of Bona Fide images. We showed in 6.5.3 that even for an in-database scenario we can see noticeable performance variation when varying Bona Fide qualities. In our case, we simply vary the JPEG quality factors of the Bona Fide images. Even more extreme performance drops are seen when testing in a mixed-dataset scenario as

shown in 6.5.4 when confidence in Bona Fide images becomes lower. When dealing with both problems simultaneously performances collapsed for all FMD (see last row Table. 6.3).

When considering the results from 6.5.3 and 6.5.4 we believe that no-reference FMD is ill suited for BFMD as the combination of unknown Bona Fide sources and unknown Face Morphed images sources is an extremely difficult problem to tackle. On top of those issues, something that it is not studied in this chapter is the impact of non-constant α_b and α_l. For all the results, those parameters are set to 0.5, some combination might lead to less visible impact on the final result. Having α_b set to 1 invalidate equation (6.6) leaving only interpolation traces. Recently NIST organised a Face Morphing detection competition[1]. Current results tend to confirm the difficulty of the task of no-reference FMD. It is worth noticing that in the NIST challenge, the worst performances are observed for handmade morph which indeed suggest that nonconstant α_l and α_b have an impact and should be further studied.

For BFMD, we believe that differential FMD are the way to go as they do not have to deal with previously mentioned issues and would not require more infrastructure changes than no-reference FMD.

While no-reference FMD might not be a good option for BFMD. We still believe that they could be reliable for Controlled Face Morphing detection (CFMD) such as Know Your Customer (KYC) remote onboarding where a priori knowledge can be used (e.g. image format, sensor properties). For those cases, proper evaluation of FMD should be done to ensure robustness against varying Bona Fide provenance and quality.

[1] https://pages.nist.gov/frvt/html/frvt_morph.html

CHAPTER 7

H.264 Double Compression Detection

7.1 Introduction

With the 2019 Coronavirus pandemic, we have seen an increasing use of remote technologies such has remote identity verification. In a remote identity verification system, a video acquisition of both the Identity Document and the person seems like an obvious choice.

In fact, the person and the ID are not static by nature and thus require many frames to be authenticated. Video has been commonly used for some time to perform liveness verification of an individual and is being used more and more to authenticate security elements such as holograms or variable ink on identity documents. Another great advantage of video stream against simple images is the added complexity for a counterfeiter to tamper such a stream.

In fact, with a video stream the counterfeit needs to develop complex tampering algorithms that work in real time. To tamper a text field, a simple copy-move would be enough for an image. For video, the counterfeit would have to detect and precisely track the identity document by using methodology such as shape-from-template. We understand intuitively how challenging the tampering process become in comparison to a simple image tampering. Recently, those arguments were acknowledged and lead to new regulations such as the French requirement rule set for remote identity verification service providers [14] enforcing the use of video in the context of remote identity verification.

The challenging aspect of video tampering must not induce a blind confidence in such media. Remote identity verification is heavily based on face biometry we thus expect attacks on either the live person acquisition or on the identity document picture. If the detection and tracking of the full document are not particularly well study. Face detection and tracking, on the other hand, has been extensively studied for quite some time now. The research in this field is in fact so advanced that it is even possible to detect and track as much as 468 3D face landmarks in realtime in a web browser using open-source frameworks [26, 27]. Assuming that the counterfeit will not be able to tamper the video stream in realtime or inject a prepared video is thus unreasonable.

We see that before any biometric matching between a person and the identity document, it is necessary to first authenticate the video media. While liveness

detection methods are well studied and allow to reasonably reject the hypothesis of an injected stream when combined with random challenges such as eye blinking, smiling, etc. Those are not enough to authenticate the video, as it could be tampered in realtime.

In this chapter, we suppose that a counterfeit will tamper a video in realtime. We assume that the acquisition device is controlled and safe, and that the counterfeit will intercept the stream before being sent to the server. In order to tamper the video, the counterfeit must first decompress the stream then perform the tampering and finally recompress it before sending it back to the server. Detecting the double compression of the video is thus a first step toward authenticating the media. We will focus on the H.264 compression which is, along with VP8, the only codec imposed by the WebRTC RFC [153].

7.1.1 Related works

The first H.264 encoder has been officially approved in 2003. It was proposed to have an extension to the previous encoder i.e. H.263 and aimed at providing a good visual quality while lowering the bitrate as much as possible. This led to a few major differences from previous encoders. Even though H.264 has been around since 2003, many research [154, 155, 156, 157] kept focusing on older versions. This made sense as older encoders were still extensively used at that time and H.264 was still rapidly evolving. Nowadays, H.264 has become one of the most used video encoders in particular for video content on the internet as it is one of the two mandatory video codecs used in the WebRTC protocol.

This extensive use soon encouraged researchers to move their attention to H.264 instead of older encoders. In its core principles, H.264 is similar to the older standards. In particular, it is mainly composed of two stages. A first prediction stage aiming at reducing the amount of information and a second stage which further compress that information using a DCT transformation and quantification. Unlike previous standards, H.264 introduced a new integer approximation of the DCT transform and also introduced a variable size prediction algorithm.

As most video encoding algorithms, H.264 takes advantage of the temporal redundancy in video to reduce the information needed to encode multiple frames.

H.264 groups many frames into a Group Of Pictures (GOP) where an I-frame usually serves as a reference and the next frames (P or B-frames) are predicted based on this I-frame and other B or P-frame of the same GOP. When a video is compressed twice, some I-frame might be recompressed as P or B-frame and vice versa. This is often called frame relocation. Many research focuses on frame relocation to detect double video compression. In [158, 159, 160], authors trained deep neural networks on the frame residual to detect relocated frames. In [161], authors trained a One-Class classifier on the reconstructed frame residual to detect the double compression. In [162], the authors directly study the bit size of each encoded frame. They showed that relocated I-frame requires more bits than typical P or B-frame and can thus be detected. This allows them to estimate the primary GOP size in case of a double compression. Similarly [163, 164, 165, 166, 167, 168, 169] also try to estimate the primary GOP size as an evidence of double H.264 compression. One advantage of those methods is that they are applicable to other video encoder as the principle of GOP is present in many video compression algorithms.

Other approaches such as [170, 171, 172] focus on recompression using the same quantification parameters. They showed that for H.264 the frames converge to a particular state when compressed multiple times using the same quantification parameters. This property can be exposed through an analysis of the DCT coefficient or using the frame noise residual.

Finally, some methods [173, 174] try to expose the double H.264 by studying the DCT coefficient distribution. They trained different classifier on the DCT coefficient to detect if a video is compressed twice.

7.1.2 Organisation of the Chapter

The chapter will be organised as follows. A brief overview of the main step of the H.264 compression will first be introduced. After, the motivation behind the choice of the analysis of the DCT coefficient to expose a double compression will be explained. Then we will present how those coefficients are sampled and modelled prior to the analysis.

We will then derive two hypothesis tests to detect a double video compression.

First, a simple ratio test will be presented when all parameters are known in advance. Then a generalised likelihood ratio test will be introduced to take into account the lack of knowledge regarding some parameters.

Then, a few numerical experimentation will be performed. We will first validate the theoretical model and evaluate the performances on a set of simulated frames. Then the method will be evaluated on a set of real video.

Finally, we will conclude with a few remarks and perspectives regarding the presented method.

7.2 H.264 intra-frame compression

In this section, we will give a brief overview of the main steps of the H.264 compression. We will skip through many aspects of the compression as they are not relevant in our analysis. We encourage the reader to read [175] to get a more in-depth presentation of the complete H.264 encoding process.

We will only focus on the intra-frame compression and on the luma component in the rest of the chapter. Intra-frames, and the luma component, of H.264 stream contains the most of the information.

For those frame, the compression is mostly divided into two major steps. The prediction step and the transformation and quantification step. We will first briefly explain the objective of the prediction step and then explain the transformation and quantification process. Finally, we will briefly introduce the mechanism of the rate control which is a relevant part of the encoding process for our method.

7.2.1 Prediction

At the prediction stage the H.264 aim at producing an estimate of the frame using the least amount of information as possible. To do so, the frame is first split into Macroblocks (MB) of size 16×16. Each MB is then predicted only by extrapolating information from neighbouring MBs. For intra-frames the MB can be predicted at three different sizes i.e. 16×16, 8×8 and 4×4. In each case, the MB is subdivided into smaller sub-blocks that are predicted using information from already decoded sub-blocks or neighbouring MB. For each sub-block, the encoder

find the best approximation (in terms of sum of absolute error) by choosing one of the available prediction modes for a given sub-block size. In the rest of the chapter, we will use the notation PredX with X the size of the prediction used to refer to a MB subdivided into sub-blocks of size X. The prediction PredX dictates which transformation will be used in the following stage, so we will always treat MB with different prediction mode separately.

Once the prediction is made, it is subtracted to the current frame to obtain a residual. This residual is mostly null and can thus be compressed efficiently.

7.2.2 Transformation and Quantification

The residual is compressed using a process similar to JPEG. It first transformed into the frequency domain using a DCT transform and then compressed by removing higher frequencies.

The DCT transformation is an approximation of the integer DCT. In H.264 there exist two main transformations. A 4×4 DCT transformation for MB predicted with Pred4 and Pred16. And an 8×8 transformation for Pred8. It is worth noticing that the 8×8 prediction and transformation are only available in the High compression profile of H.264. In theory, this profile is not mandatory in the WebRTC RFC [153]. In practice, this profile has been included in H.264 version 3 in 2005 and is nowadays the most commonly used profile. Both transformation follows the same principle. First the residual is transformed, then it is scaled and quantised:

$$\mathbf{C} = \lfloor (DCT(\mathbf{R}) \circ \mathbf{Q}) \cdot s \rfloor \tag{7.1}$$

with $\circ$ the Hadamard product, $\mathbf{R}$ the residual sub-block, $\mathbf{Q}$ the quantification matrix and s a scaling scalar.

The quantification matrix $\mathbf{Q}$ and the scaling scalar s depends on the quantisation parameter QP. This quantisation parameter can vary between MBs. In H.264 QP can vary from 0 to 51 with 0 being almost lossless, 23 considered as visually lossless and 51 the strongest compression.

When Pred16 is used, an additional transformation, called the DC transform,

can be applied. This transformation is applied to every DC component just before quantification. We decided to ignore MBs predicted with Pred16 for simplicity. For the rest of the chapter, we will only consider MBs predicted either with $Pred8$ or $Pred4$.

7.2.3 Rate Control

As we mentioned the quantisation parameter QP can vary for each Macroblocks within the same frame. This depends on the rate control used by the H.264 encoder. There exists multiple modes that can be chosen for the rate control. There are mainly two objectives that one might want to achieve when compressing with H.264. He will either want to archive the file or stream the file. For archiving, the typical rate controls used are the Constant QP which maintain a fixed QP for each frame or the Constant Rate Factor (CRF) which will try to maintain a constant visual quality given a target QP. When streaming, rate controls that try to maintain a given bitrate is usually preferred such as the Average Bitrate mode or the Constant Bitrate mode.

Apart from the constant QP rate control, every mode allows the encoder to vary the QP per Macroblock. This implies that the choice of QP for each Macroblock cannot be controlled exactly unless one chooses the constant QP mode. While it is possible to implement a H.264 encoder for which we can control the QP at the Macroblock level, we argue that it is not trivial and we will consider that the counterfeit will use a standard encoder a will thus not have full control over the QP.

7.2.4 Impact of a Double H.264 Compression

We briefly introduced the I-Frame compression in the earlier section. We showed that a frame is first segmented into many Macroblocks of size 16×16. Every Macroblock is then predicted in order to extract a residual. That residual is finally transformed using an integer approximation of the DCT and quantised. One particularity of an H.264 encoder is that it can change the algorithm used to perform the prediction, the type of DCT and the quantisation parameter at the

Macroblock level. All that information can be retrieved for each Macroblock while decoding the H.264 stream. But when compressing a video using a standard H.264 encoder, those parameters cannot be predicted in advance. As a result, when for a Macroblock predicted using PredX and a quantisation parameter QP_1 we expect to observe things in case of a double compression :

1. The MB will be predicted by PredY with $Y \neq X$
2. The MB will be quantised using $\mathrm{QP}_2 \neq \mathrm{QP}_1$

Of course we could have $Y = X, \mathrm{QP}_2 = \mathrm{QP}_1$ in which case the recompression will have no impact on the MB. Nevertheless, it is reasonable to assume that a non-negligible number of MB will be recompressed with either $Y \neq X$ or $\mathrm{QP}_2 \neq \mathrm{QP}_1$ or both.

We thus propose to study the distribution of the DCT coefficient to detect a double compression. In particular, we will see that the coefficients of MB predicted using PredX and a quantisation parameter QP_1 have a characteristic distribution and that the recompression have an impact on that distribution.

7.2.5 Sampling by Quantisation Parameter and Prediction Mode

As previously exposed, the prediction and compression are performed at the level of Macroblocks. While processing a video, it is thus proposed to first partitioned all Macroblocks according to their prediction mode i.e. $Pred4$ and $Pred8$. This partitioning is necessary as the prediction mode also dictates which transformation is applied before the quantification. Then the Macroblocks are further partitioned according to the quality factor QP used. With $\mathbf{B}^{x,q}$ denoting all the sub-blocks predicted with $Predx$ and quantified at QP, we thus have a set of vectors denoted $\mathbf{C}_{i,j}^{x,q}$ containing all coefficients at the location (i, j) of each sub-block $\mathbf{B}^{x,q}$.

7.2.6 Modelling of the Coefficient

In this chapter, we propose to study the DCT coefficient. In particular, we propose to study if the DCT coefficient at a specific quantification level can be char-

acterised. The distribution of DCT coefficients for images has been extensively studied. Firstly, supposed to be normally distributed [176]. It was, then showed that the Laplacian distribution [177] was a better modelling for AC coefficients. Since then, the Laplacian modelling has been a predominant choice because of its simplicity and good overall accuracy. Another model has been proposed such has Cauchy [178], Gaussian mixture [179] etc. More recently the authors of [180] proposed a doubly stochastic model of AC coefficients and showed that it was more accurate than other models. For H.264, the Laplacian and Cauchy distribution remain the preferred choice [181].

We will consider that the DCT coefficients $\mathbf{C}_{i,j}^{x,q}$ follow a Laplacian distribution :

$$\mathbf{C}_{i,j}^{x,q} \sim \text{Laplace}(0, b_{i,j}^{x,q}). \tag{7.2}$$

In Fig. 7.1 it can be seen that the Laplacian distribution is indeed a good approximation.

In Fig. 7.2 it can be seen that for a given QP the parameter $b_{i,j}^{x,q}$ seems stable across multiple videos. To the best of our knowledge, this stability was first pointed in [181]. We will thus consider a single scale parameter b for each tuple (x, q, i, j).

The probability density function for the coefficients $\mathbf{C}_{i,j}^{x,q}$ is given by

$$f(x|b) = \frac{1}{2b} \exp(\frac{-|x|}{b}). \tag{7.3}$$

As shown in [182], it is not possible to assume the coefficient of a DCT transformation independent and identically distributed (i.i.d) when directly applied to the image content. In H.264, the prediction tries to approximate each pixel value. This prediction can be seen as an estimator for each pixel mean. The DCT transformation is finally applied on the residual of the initial frame to which the prediction is subtracted. This allows us to consider $\mathbf{C}_{i,j}^{x,q}$ i.i.d.

In the following section, we will omit the tuple (x, q, i, j) to improve readability. The coefficients $\mathbf{C}_{i,j}^{x,q}$ for a given tuple (x, q, i, j) will simply be denoted as $\mathbf{C} = \{c_1, c_2, ..., c_N\}$ with N the number of coefficients. In the same manner, $b_{i,j}^{x,q}$ will be denoted as b. Finally, all the coefficients $c_i, i \in [1; N]$ will be considered i.i.d.

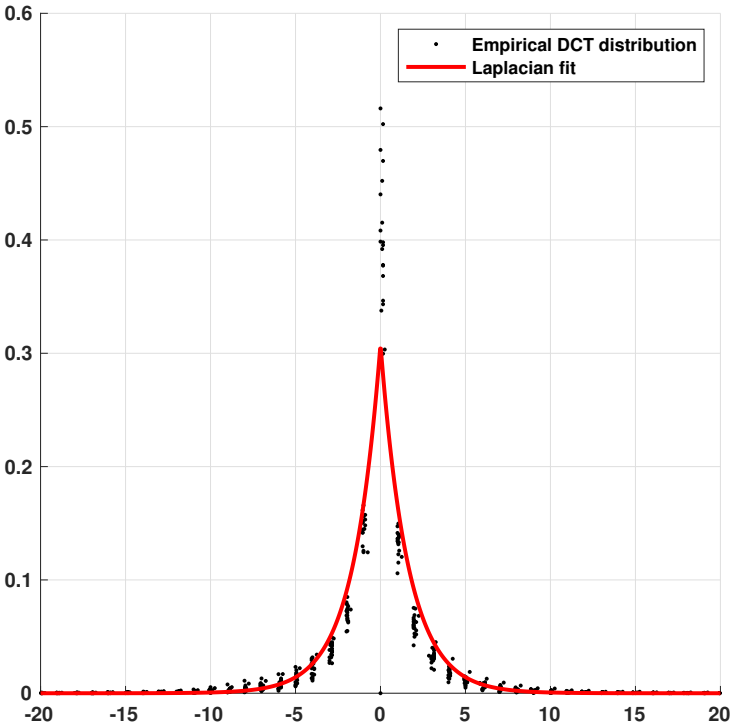


Figure 7.1 : Empirical distribution of the DC coefficient for 20 videos fitted with a Laplacian distribution.

7.3 Statistical Test Design

We consider that $\mathbf{C}$ follows Laplacian distribution with zero mean and with scale b. We expect b to be affected by the double compression process. In the following section, we will first introduce the first statistical test when every parameter is known (i.e. the value of b for the first and second compression). Then we will derive a more practical test where only the first compression parameter is known.

7.3.1 Likelihood ratio test for two simple hypotheses

We saw that for a given tuple (x, q, i, j), the scale parameter b seems to approach a fixed value. We will thus assume in the rest of the chapter that for a video

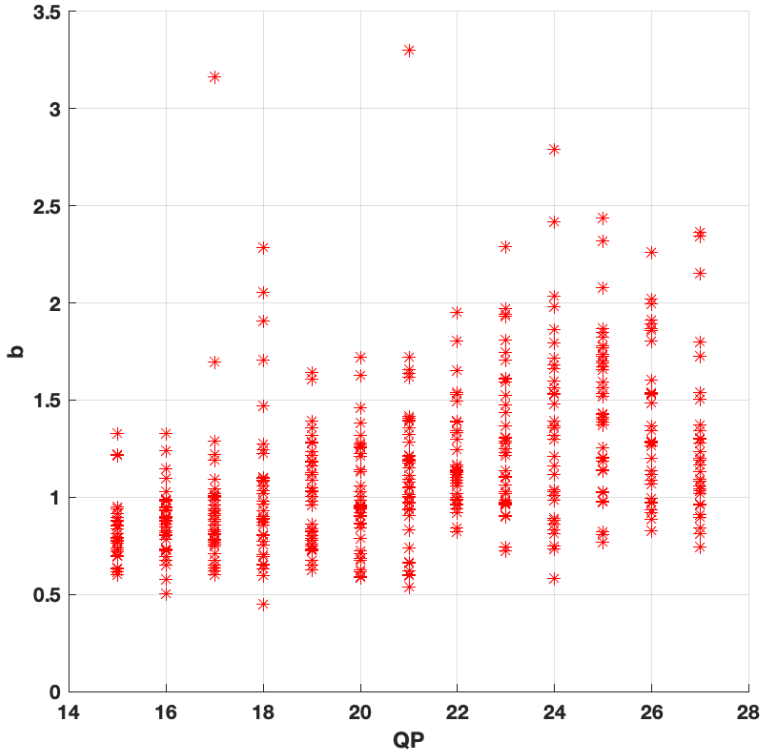


Figure 7.2 : Distribution of $b_{1,1}^{4,q}$ for various QP for 40 video.

compressed with H.264 once. The coefficients $\mathbf{C}$ follows a zero mean Laplacian distribution of scale b_0.

To verify if a video has been compressed twice we then propose to define the following hypothesis test.

$$\begin{cases} \mathcal{H}_0 : \mathbf{C} \sim \text{Laplace}(0, b_0) \\ \mathcal{H}_1 : \mathbf{C} \sim \text{Laplace}(0, b_1), b_1 \neq b_0. \end{cases} \tag{7.4}$$

If the video has gone through a single compression then it should follow a Laplacian distribution of scale b_0. Else, it will follow a Laplacian distribution of scale b_1.

We can define the likelihood ratio as

$$\Lambda(\mathbf{C}) = \frac{\mathcal{L}_1(\mathbf{C})}{\mathcal{L}_0(\mathbf{C})}. \tag{7.5}$$

Because the coefficients $c_i, i = \{1, 2, ..., N\}$ are i.i.d, we can rewrite the likelihood ratio as

$$\Lambda(\mathbf{C}) = \prod_{i=0}^{N} \Lambda(c_i). \tag{7.6}$$

The log-likelihood ratio is then obtained by combining (B.3) and (7.6)

$$\begin{aligned} \Lambda(\mathbf{C}) &= \log \prod_{i=0}^{N} \Lambda(c_i) \\ &= \sum_{i=0}^{N} \log \Lambda(c_i) \\ &= N \log \frac{b_0}{b_1} + \frac{b_1 - b_0}{b_0 b_1} \sum_{i=0}^{N} |c_i|. \end{aligned} \tag{7.7}$$

With $N \to \infty$ the Central Limit Theorem (CLT) gives us

$$\frac{1}{N} \sum_{i=0}^{N} |c_i| \sim \mathcal{N}(\mu, \frac{\sigma}{\sqrt{N}}), \mu = \mathbf{E}|\mathbf{C}|, \sigma = \mathbf{Var}|\mathbf{C}|. \tag{7.8}$$

Under the hypotheses $\mathcal{H}_h, h \in \{0, 1\}$ we have that $|\mathbf{C}| \sim$ Exponentiel$(\mathrm{b_h^{-1}})$ which lead to

$$\sum_{i=0}^{N} |c_i| \sim \mathcal{N}(N b_h, |N| \frac{b_h}{\sqrt{N}}). \tag{7.9}$$

By combining (7.7), (7.9) we have that under $\mathcal{H}_h, h \in \{0, 1\}$:

$$\Lambda_h(\mathbf{C}) \sim \mathcal{N}(\mu_h, \sigma_h), \tag{7.10}$$

with

$$\mu_h = N \log \frac{b_0}{b_1} + N b_h \frac{b_1 - b_0}{b_0 b_1} \tag{7.11}$$

$$\sigma_h = |N\frac{b_1 - b_0}{b_0 b_1}|\frac{b_h}{\sqrt{N}}. \tag{7.12}$$

Let define

$$\Lambda^\star(\mathbf{C}) = \frac{\Lambda_h(\mathbf{C}) - \mu_0}{\sigma_0} \sim \mathcal{N}(\frac{\mu_h - \mu_0}{\sigma_0}, \frac{\sigma_h}{\sigma_0}). \tag{7.13}$$

The statistic $\Lambda^\star(\mathbf{C})$ thus follows a standard normal distribution under $\mathcal{H}_0$.

In virtue of the Neyman-Pearson lemma, the most powerful test δ for the problem (7.4) is the likelihood ratio test :

$$\delta(\mathbf{C}) = \begin{cases} \mathcal{H}_0 \text{ si } \frac{\mathcal{L}_1(\mathbf{C})}{\mathcal{L}_0(\mathbf{C})} < \tau \\ \mathcal{H}_1 \text{ si } \frac{\mathcal{L}_1(\mathbf{C})}{\mathcal{L}_0(\mathbf{C})} \geq \tau \end{cases}. \tag{7.14}$$

We can define the test $\delta^\star$

$$\delta^\star(\mathbf{C}) = \begin{cases} \mathcal{H}_0 \text{ si } \Lambda^\star(\mathbf{C}) < \tau^\star \\ \mathcal{H}_1 \text{ si } \Lambda^\star(\mathbf{C}) \geq \tau^\star \end{cases}. \tag{7.15}$$

Which is equivalent as the logarithm is monotonic and the transformation (7.13) is linear.

One advantage of hypothesis testing is to allow us to guaranty a prescribed false alarm rate α_0. It is also possible to define the theoretical power of the test as a function of the false alarm rate.

The power β of a test δ is given by the probability α of rejecting the null hypothesis $\mathcal{H}_0$ under $\mathcal{H}_1$:

$$\beta(\delta) \triangleq \mathbb{P}_{\mathcal{H}_1}[\delta(\mathbf{C}) = \mathcal{H}_1] = 1 - \alpha. \tag{7.16}$$

For our test $\delta^\star$ the threshold $\tau^\star$ with respect to the false alarm rate α_0 can be deduced by solving

$$\mathbb{P}_{\mathcal{H}_0}[\Lambda^\star(\mathbf{C}) \geq \tau^\star] = \alpha_0, \tag{7.17}$$

then, the power of the test is simply given by

$$\beta(\delta^\star) = \mathbb{P}_{\mathcal{H}_1}[\Lambda^\star(\mathbf{C}) \geq \tau^\star]. \tag{7.18}$$

7.3.2 Generalised likelihood ratio test

For the test $\delta^\star$ define in (7.15), both the parameter b_0 and b_1 are supposed to be known in advance.

If we assume b to mostly depend on the quantisation parameter QP, then b_1 cannot be known in advance. In fact, even though all the coefficients of $\mathbf{C}$ come from macroblocks quantised using the same known quantisation parameter QP_2. The value of the previous quality factor QP_1 is unknown and may even vary for each coefficient.

In practice, in case of a double compression the coefficients $\mathbf{C}$ will not exactly follow a Laplacian distribution as shown by the authors of [173]. We can thus except b_1 to differ from the expected value of a quantisation parameter QP_2.

In case of a simple compression, we expect $\mathbf{C}$ to follow a Laplacian distribution of scale b_0. So to verify if a frame is double compressed, we propose to test if the coefficient $\mathbf{C}$ does follow a Laplacian distribution of scale b_0 which depend on the quantisation parameter QP_2 or if it follows a Laplacian distribution of scale $b_1 \neq b_0$ and with b_1 unknown.

This is equivalent to the test proposed in 7.4 but with the parameter b_1 replaced by the maximum likelihood estimate (B.6).

We thus have the log-likelihood ratio given by

$$\begin{aligned}
\Lambda(\mathbf{C}) &= N\log(\frac{b_0}{\hat{b}}) + \frac{\hat{b}-b_0}{b_0\hat{b}}\sum_{i=0}^{N}|c_i| \\
&= \frac{1}{b_0}\sum_{i=0}^{N}|c_i| - N\log(\frac{1}{N}\sum_{i=0}^{N}|c_i|) + N(\log(b_0) - b_0) \\
&= \frac{N\overline{\mathbf{C}}}{b_0} - N\log(\overline{\mathbf{C}}) + N(\log(b_0) - b_0)
\end{aligned} \tag{7.19}$$

with

$$\overline{\mathbf{C}} = \frac{1}{N}\sum_{i=0}^{N}|c_i|. \tag{7.20}$$

Under $\mathcal{H}_h$ we have that

$$\overline{\mathbf{C}} \sim \mathcal{N}(b_h, \frac{b_h}{\sqrt{N}}). \tag{7.21}$$

Let

$$\mathbf{C}^\star = \frac{\overline{\mathbf{C}} - b_h}{b_h}\sqrt{N} \sim \mathcal{N}(0,1). \tag{7.22}$$

We then have

$$\begin{aligned}\Lambda(\mathbf{C}) = \frac{N}{b_0}(&\frac{b_h}{\sqrt{N}}\mathbf{C}^\star + b_h \\ &- b_0\log(b_h) - b_0\log(\frac{1}{\sqrt{N}}\mathbf{C}^\star + 1)) \\ &+ N(\log(b_0) - b_0).\end{aligned} \tag{7.23}$$

The Taylor expansion gives us that

$$\log(\frac{1}{\sqrt{N}}\mathbf{C}^\star + 1) \simeq \frac{1}{\sqrt{N}}\mathbf{C}^\star - \frac{1}{2N}(\mathbf{C}^\star)^2. \tag{7.24}$$

Finally, by combining (7.23) and (7.24) we have that

$$\begin{aligned}\Lambda(\mathbf{C}) &= \frac{(\mathbf{C}^\star)^2}{2} + \sqrt{N}\frac{b_h - b_0}{b_0}\mathbf{C}^\star + N\frac{b_h - b_0\log(b_h)}{b_0} \\ &+ N(\log(b_0) - b_0) \\ &= \frac{1}{2}(\mathbf{C}^\star + d_h)^2 + a_h\end{aligned} \tag{7.25}$$

with

$$d_h = \sqrt{N}\frac{b_h - b_0}{b_0} \tag{7.26}$$

$$a_h = N(\log(b_0) - b_0) + N\frac{b_h - b_0\log(b_h)}{b_0} - \frac{1}{2}d_h^2. \tag{7.27}$$

In particular, under $\mathcal{H}_0$ we will have $d_0 = 0$ and

$$a_0 = N(1 - b_0). \tag{7.28}$$

Finally

$$\hat{\Lambda}(\mathbf{C}) = 2(\Lambda(\mathbf{C}) - a_0) \sim \chi^2(1). \tag{7.29}$$

In virtue of the Neyman-Pearson lemma, the most powerful test is the generalised likelihood ratio

$$\hat{\delta}(\mathbf{C}) = \begin{cases} \mathcal{H}_0 \text{ si } \hat{\Lambda}(\mathbf{C}) < \hat{\tau} \\ \mathcal{H}_1 \text{ si } \hat{\Lambda}(\mathbf{C}) \geq \hat{\tau} \end{cases}. \tag{7.30}$$

As for the test δ, the threshold $\hat{\tau}$ can be deduced by solving

$$\mathbb{P}_{\mathcal{H}_0}[\hat{\Lambda}(\mathbf{C}) \geq \hat{\tau}] = \alpha_0. \tag{7.31}$$

Finally, the power $\beta(\hat{\delta})$ is given by

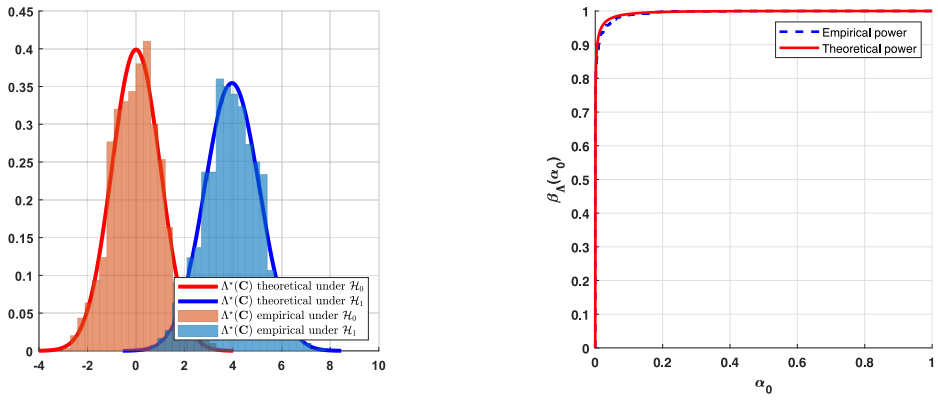
$$\beta(\hat{\delta}) = \mathbb{P}_{\mathcal{H}_1}[\hat{\Lambda}(\mathbf{C}) \geq \hat{\tau}]. \tag{7.32}$$

7.4 Numerical experimentation

7.4.1 Model validation

To verify the validity of the proposed test (7.15), we performed a Monte Carlo simulation. We generated 2000 random vectors $\mathbf{C}$ of 1000 elements c, those 2000 vectors were split in half with 1000 vectors following the hypotheses $\mathcal{H}_0$ and 1000 vectors following the hypotheses $\mathcal{H}_1$. We fixed the value of the parameters to $b_0 = 0.8$ and $b_1 = 0.9$.

In Fig. 7.3, a comparison between the theoretical and the empirical distribution



(a) (b)

Figure 7.3 : From left to right : Theorical and empirical distribution under $\mathcal{H}_0$ and $\mathcal{H}_1$, theoretical and empirical power under $\mathcal{H}_0$ and $\mathcal{H}_1$

is given under $\mathcal{H}_0$ and $\mathcal{H}_1$. One can see how the empirical distributions match the theoretical model given in (7.13).

In Fig. 7.3, the theoretical and the empirical power $\beta(\hat{\delta})$ of the test are shown. Once again the empirical simulation matches the theoretical model.

We performed the same simulation for the test (7.29). On Fig. 7.4 one can see that the empirical distribution once again match with the theoretical model. This is also true for the theoretical and empirical power as one can see in Fig. 7.4.

The power of the two tests mostly depends on the difference between b_0 and b_1 i.e. $|b_0 - b_1|$. On Fig. 7.5 we evaluate the theoretical power of the test $\hat{\delta}(\mathbf{C})$ for a fixed false alarm rate $\alpha_0 = 0.05$ and with varying b_0 and b_1.

We can observe that when we increase $|b_0 - b_1|$, the power increase. This is not surprising as we show in (7.8) that the maximum likelihood estimation of b becomes normally distributed for a sufficient number of samples. Then naturally if $|b_0 - b_1|$ is much greater than the variance of the maximum likelihood estimators then (7.19) tends to become perfectly separable between $\mathcal{H}_0$ and $\mathcal{H}_1$.

It is also important to note that as b_0 and b_1 increase, the difference $|b_0 - b_1|$ must increase to maintain the test power. This is also explained by the distribution given in (7.8). As b increase the variance of the maximum likelihood estimator increase and thus the distance $|b_0 - b_1|$ must also increase to overcome this loss

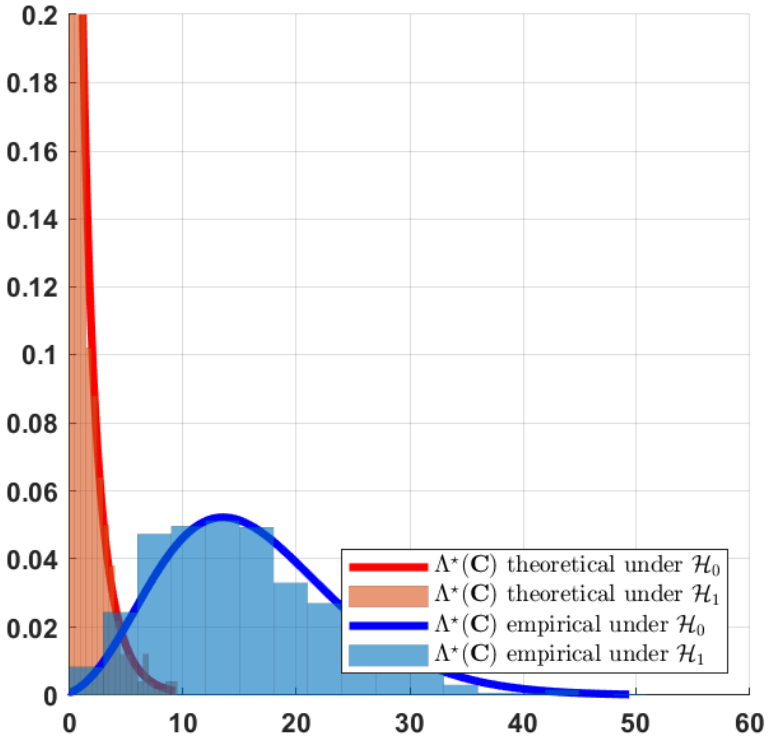


Figure 7.4 : Theoretical and empirical distribution under $\mathcal{H}_0$ and $\mathcal{H}_1$.

of precision. We will see that this phenomenon affects the performances when the quantisation parameter QP is high.

7.4.2 Performances on simulated frames

The test (7.30) is first evaluated on simulated H.264 frame. This allows us to precisely control both the prediction mode and the quality factor use for each macroblock.

To do so, we randomly selected 500 images from the RAISE [183] dataset. For each image, only a central portion of size 504×504 is kept. Those images have then been converted to grayscale, before being compressed. We reimplemented the H.264 compression as described in [184].

We first compressed every image with a prediction and transformation of size

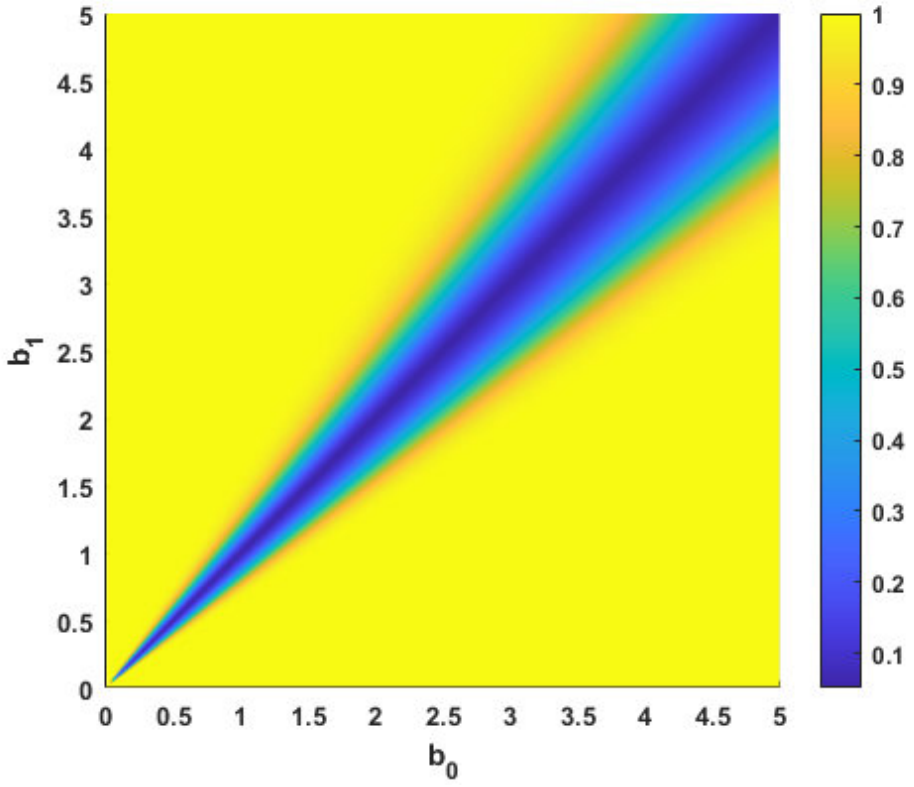


Figure 7.5 : Theoretical power with $\alpha_0 = 0.05$ for varying b_0 and b_1.

4 at various QP_1. We then repeat this process for a prediction and transformation of size 8.

Then each of these compressed images is recompressed with both prediction mode and various QP_2. For a given image predicted with $PredX$ and compressed with a quality factor QP_1 we thus have to scenarios of interest after the recompression

1. The frame is predicted with $PredY$ and $Y \neq X$
2. The frame is compressed at $\mathrm{QP}_2 \neq \mathrm{QP}_1$

We will first focus on the case where $PredY = PredX$ to evaluate the impact of the quantisation parameter on the detection performances.

Then we will study the case where $PredY \neq PredX$ and various QP to evaluate the impact of the prediction mode on the prediction.

In each case, the parameter b_0 is estimated as the median of all the maximum likelihood estimations $\hat{b}$ observed for images simply compressed by a quantisation parameter QP_2.

Recompression with the same prediction mode

The generalised log-likelihood ratio given in (7.29) is calculated for each image compressed at QP_2 and images first compressed at QP_1 and then recompressed at QP_2.

The empirical Area Under the Curve (AUC) was computed in order to obtain an overview of the detection performance for various QP_1 and QP_2. The results are also given for different coefficient i.e. $\mathbf{C}_{1,1}, \mathbf{C}_{1,4}$ and $\mathbf{C}_{4,4}$ In Fig. 7.6 the first and second predictions were made using Pred4.

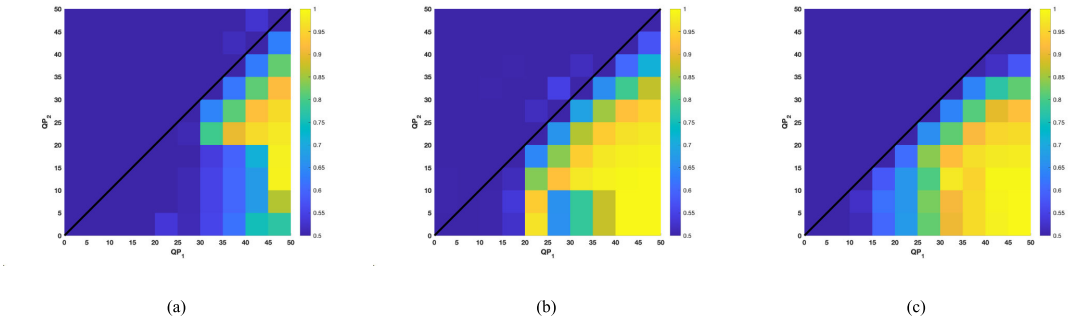
Whatever the coefficient used, the first observation to be made is that for $\mathrm{QP}_1 = \mathrm{QP}_2$ the detection is completely random (i.e. an AUC of 0.5). This is expected as a recompression at the same quantisation parameter in H.264 has no impact.

It is also important to remark that the detection is not possible for $\mathrm{QP}_2 > \mathrm{QP}_1$. In Fig. 7.6, this corresponds to the upper-left part. With $\mathrm{QP}_2 > \mathrm{QP}_1$ the second compression is stronger than the first compression and thus erase any traces of the first compression.

The detection is possible only for $\mathrm{QP}_2 < \mathrm{QP}_1$. In particular, the performance increase with $|\mathrm{QP}_2 - \mathrm{QP}_1|$. We also notice that for every coefficient the detection performances are satisfactory for $|\mathrm{QP}_2 - \mathrm{QP}_1| > 10$.

Finally, the choice of the coefficient has a strong influence on the detection performance. We can see how the performances for lower values of QP_2 are worst for the DC coefficient $\mathbf{C}_{1,1}$ than for the other two. The performances also increase between $\mathbf{C}_{1,4}$ and $\mathbf{C}_{4,4}$.

To understand this phenomenon, it is important to recall two things. First, the value of b_0 depends mostly on the quantisation parameter. And secondly, the compression becomes increasingly stronger for coefficients further away from the DC coefficient. This implies that b_0 decrease as QP_2 increase. But also that for a fixed value of QP_2, b_0 also decrease as the studied coefficient gets farther from the



(a) (b) (c)

Figure 7.6 : Empirical AUC for the coefficient $\mathbf{C}_{1,1}$, $\mathbf{C}_{1,4}$ and $\mathbf{C}_{4,4}$ predicted with Pred4 and recompressed with Pred4 with respect to $|QP_2 - QP_1|$.

DC coefficient. As shown in Fig. 7.5, the performances increase when b_0 and b_1 are lower.

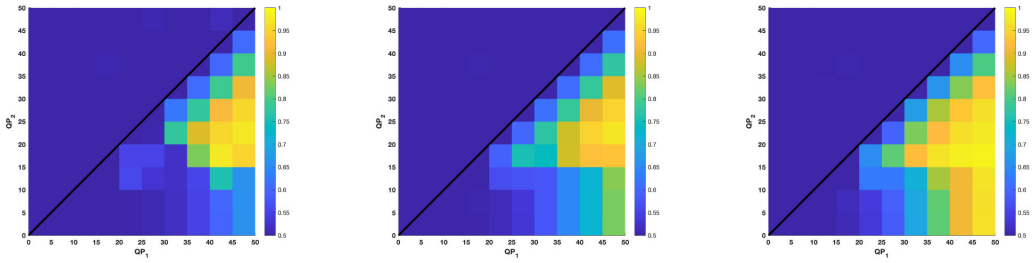
For lower value of QP_2, it is then natural to observe better performance for coefficients farther away from the DC coefficient. But this is only true as long as there exists a sufficiently large number of non-zero coefficients. In fact, one can notice that for $\mathrm{QP}_2 > 35$ the detection becomes random for the coefficient $\mathbf{C}_{4,4}$ whereas for the DC coefficient we still observe an AUC of about 0.8.

On Fig. 7.7, the same simulation has been performed but with a first and second prediction using Pred8. It can be seen that the results are mostly similar. For the 8×8 transform, the results are slightly worse than the 4×4 transform when both QP_1 and QP_2 are lower.

Recompression with a different prediction mode

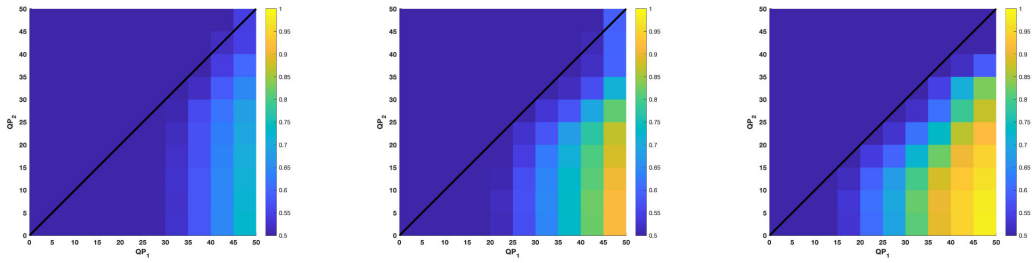
In the previous section, we evaluated the performances in the case where the first and second predictions were the same. As we mentioned, it is also possible to observe Macroblocks for which the first and second prediction will not be the same.

On Fig. 7.8, we can see the result of a first prediction with Pred8 and a second prediction with Pred4. In this case, b_0 is estimated from simply compressed images



(a) (b) (c)

Figure 7.7 : Empirical AUC for the coefficient $\mathbf{C}_{1,1}$, $\mathbf{C}_{1,4}$ and $\mathbf{C}_{4,4}$ predicted with $\mathrm{Pred8}$ and recompressed with $\mathrm{Pred8}$ with respect to $|QP_2 - QP_1|$.



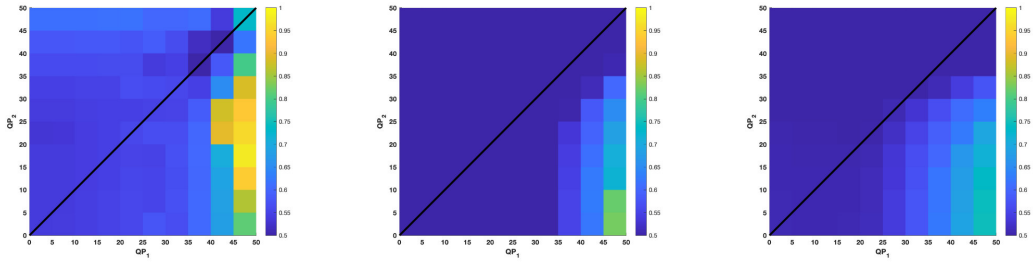
(a) (b) (c)

Figure 7.8 : Empirical AUC for the coefficient $\mathbf{C}_{1,1}$, $\mathbf{C}_{1,4}$ and $\mathbf{C}_{4,4}$ predicted with $\mathrm{Pred8}$ and recompressed with $\mathrm{Pred4}$ with respect to $|QP_2 - QP_1|$.

with Pred4 and QP_2. We can see that the performances are lower but overall similar. The double compression can only be detected for $\mathrm{QP}_1 > \mathrm{QP}_2$.

On Fig. 7.9, we observe similar result when the first prediction is Pred4 followed by Pred8. Interestingly, we can see that the detection is somewhat possible with $\mathrm{QP}_1 \ll \mathrm{QP}_2$ for the coefficient $\mathbf{C}_{1,1}$ but the performances are really low.

We can observe that the performance drop is more important in the case of



(a) (b) (c)

Figure 7.9 : Empirical AUC for the coefficient $\mathbf{C}_{1,1}$, $\mathbf{C}_{1,4}$ and $\mathbf{C}_{4,4}$ predicted with Pred4 and recompressed with Pred8 with respect to $|QP_2 - QP_1|$.

Pred4 followed by Pred8. This can be explained by the fact that Pred8 is less accurate than Pred4, we will thus have a residual that might not be affected by the first compression. In fact, we can see that unless the QP_1 was extremely high (i.e. really strong compression), the detection is pretty much impossible.

For Pred8 followed by Pred4 the performances are slightly better. This time the second prediction is more accurate than the first one. One block of size 8×8 is now predicted using 4 blocks of size 4×4. Because of the first compression, every lower right 4×4 block will appear as if it was more compressed than every upper left 4×4 block. This will create a discrepancy between the Pred4 block which affects the estimation of $\hat{b}$.

Overall, the performances decrease in this scenario. As we explained, H.264 apply the transformation to the residual. When the first and second prediction match, it is likely that the H.264 will choose the same prediction mode. This leads to the same residual data compressed twice. When the prediction size mismatch, this does not hold. The block will be predicted on a different scale and thus the residual will not be the same. The performances are better when the second prediction is more accurate than the first one.

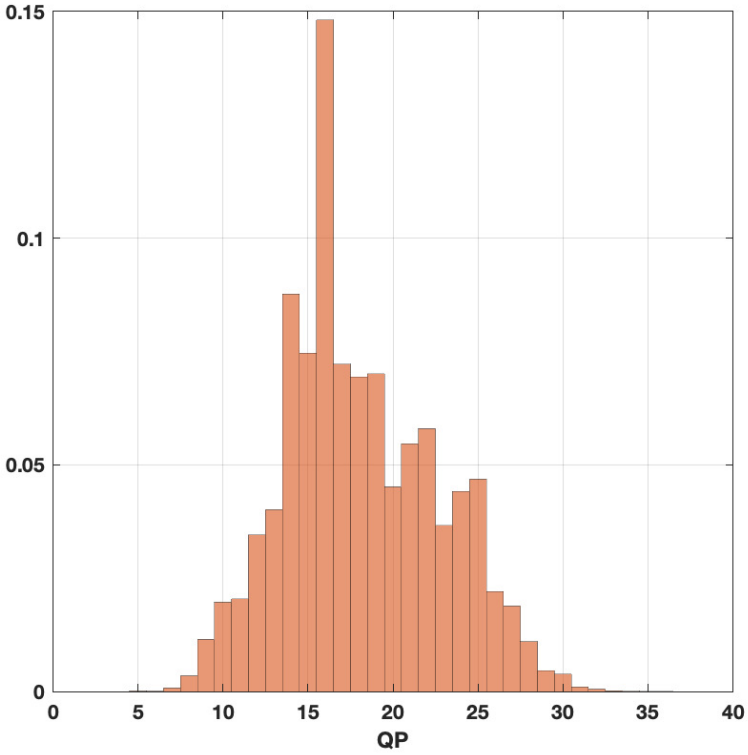


Figure 7.10 : Distribution of QP across the 45 videos.

7.4.3 Performances on Smartphone Videos

In this section we evaluate the performances on a dataset of real videos. The dataset contains 45 videos taken with 4 different smartphones. All videos are in full HD i.e. 1920×1080 pixels. All videos were compressed by the various smartphones H.264 encoders using the high profile. Each video thus contains both 4×4 and 8×8 macroblocks. The videos are then recompressed using the x264 encoder. We recompressed the video using the CRF rate control with different quality factors. On Fig. 7.10, the distribution of the original quantisation parameters for every video is given. The average QP across all videos is around 20. To recall, a quantisation parameter of 23 is considered as visually lossless. We can reasonably consider that the videos were originally compressed with a rate control aiming at maintaining the QP around 23.

We will evaluate three different scenarios. In the first scenario we will set $QP = 15$, so that macroblocks will tend to be recompressed at lower quality factor than the original. In the second scenario, QP is set two 20 so that the second compression is close to the first one. Finally, we evaluate the performances for $QP = 25$ for which macroblocks will tend to be recompressed at a higher quantisation parameter.

Unlike the previous evaluation on simulated frames, we cannot predict the primary prediction mode nor the primary quantisation parameter. We expect the performance to be worst when the second compression is set to $QP = 20$ and $QP = 25$ as it is then less likely that a macroblock will be recompressed at a lower quantisation parameter. In every scenario, b_0 is estimated as the median of the observed $\hat{b}$ for the original video. For the theoretical results b_1 is also estimated as the median of the observed $\hat{b}$ for the recompressed videos.

On Fig. 7.11, the results are given for the coefficients $\mathbf{C}_{1,1}^{4,20}$. Both the empirical power and the theoretical power are given. Firstly, we can see that the recompression does affect the value $\hat{b}$. The difference between b_0 and b_1 is big enough so that the theoretical power is almost perfect. In practice we can observe a significant loss in power. As observed in Fig. 7.2, even though the values $\hat{b}$ seem to vary around some b_0. It is obvious that the assumption that $\mathbf{C} \sim \text{Laplace}(0, b_0)$ does not fully reflect the real world and that b_0 is not only defined by the quantisation parameter and the coefficient position. This variance around the hypothetical value b_0 translates into a loss of power in practice. Nonetheless, we observe good detection performances in that scenario which validate the approach to real-world examples.

On Fig. 7.12, the results are given for $\mathbf{C}_{1,1}^{4,20}$. In that scenario, the second compression approximately matches the first compression. As a result, it is more likely that a macroblock will be recompressed at the same quantisation parameter or above as the distribution of QP overlaps. We indeed observe both lower theoretical and empirical performances as b_0 and b_1 are closer. Once again we observe a loss in power between the theoretical model and the empirical evaluation.

Finally, on Fig. 7.13 the results are given for $\mathbf{C}_{1,1}^{4,23}$. This time the second compression is set to $QP = 25$. In this scenario, it is more likely that a macroblock will be recompressed at a higher quantisation parameter so we expect the perfor-

Table 7.1 : AUC obtained on the smartphone dataset using the naive score fusion for various QP_2

QP_2	15	20	25	30
AUC	0.9995	0.9274	0.9723	0.9990

mances to be lower. We can see in Fig. 7.13 that the performances are indeed slightly lower than for the first scenario (i.e. 7.11) but are still reasonably good.

Those results are really encouraging as they show that even though it is not possible to detect a double compression at the same or higher quantisation parameter. The mechanism of rate control in H.264 introduce enough perturbation to obtain good detection performances. It is important to recall that in Fig. 7.11, 7.12 and 7.13 only a single QP and a single DCT sub-band are used to perform the detection. In practice the test (7.30) can be performed for each QP and each sub-band of a given video. We expect that lower values of QP will yield the better performances as they have more chances of being recompressed at a lower quantisation parameter. In Table. 7.1, we performed a naive combination of the subbands $\mathbf{C}^4_{1,1}$ and $\mathbf{C}^8_{1,1}$ by taking the average value of the test (7.30) for each QP present in the video. We can see how this simple fusion greatly improves the performances.

7.5 Comparaison to state-of-the-art methods

Finally, we evaluate our method against two state-of-the-art methods. For the first method, we implemented the algorithm described in [173] which is based on the DCT coefficients like our approach. They propose to extract non zero coefficients of every I-frame. They then extract all the coefficients in the range $[-10; 10]$ excluding 0. Finally, they compute the empirical probability of a coefficient being equal to $-10, -9, ..., 9, 10$ to create a feature vector of dimension 20. A SVM is then used to perform the classification. For the second method, we used the available implementation of [165]. They study the distribution of macroblocks types to both estimate the GOP size of the first compression and to detect a possible double compression.

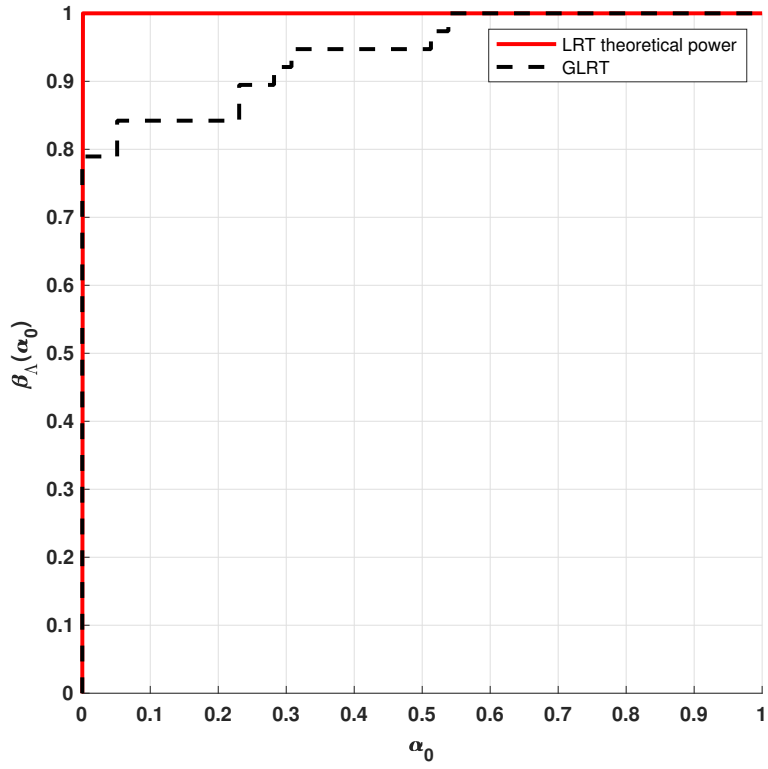


Figure 7.11 : Empirical and theoretical power for the coefficients $\mathbf{C}_{1,1}^{4,20}$ with $QP_2 = 15$

Because the method [173] requires a training dataset and the method [165] requires the first compression GOP size to be fixed, we constructed two datasets. A first dataset of 11 HD videos from [185] which we used to train the method [173] and also to get an estimate of the parameter b_0 for each QP for our method. And a second dataset of 31 CIF videos from [185].

For both dataset we compressed the video using ffmpeg and the x264 encoder with the following compression parameters. We fixed the GOP size to 9 for the first compression and a GOP size of 25 for the second compression. We use the CRF for the rate control mechanism with $QP \in \{18, 20, 23, 25, 30\}$ for both compressions. Finally, we did not specify any parameters regarding the use of B-Frames.

In the previous section, we used a single DCT subband and a single QP to perform the detection. Here we perform a naive combination of the subbands $\mathbf{C}_{1,1}^{4}$

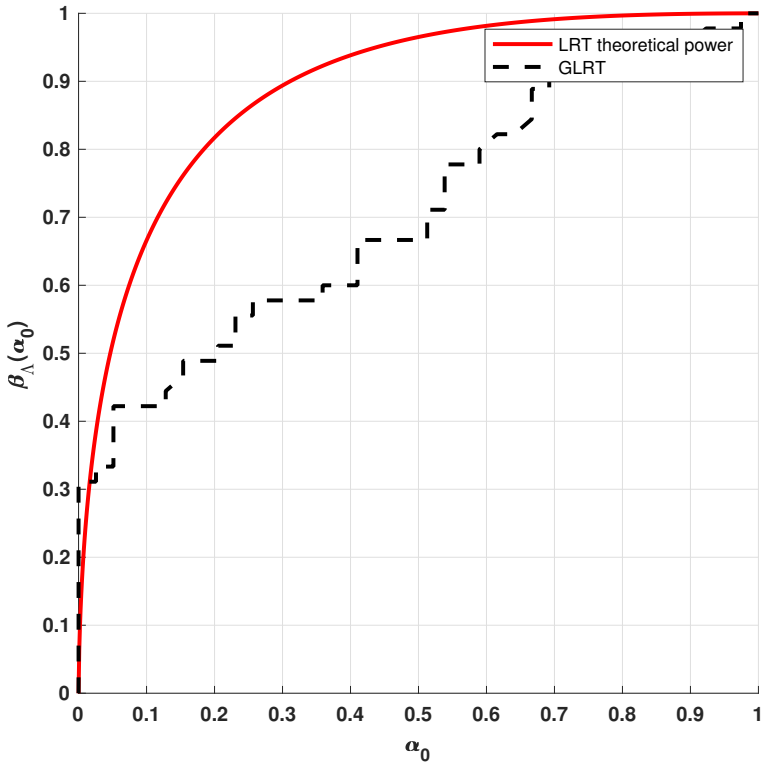


Figure 7.12 : Empirical and theoretical power for the coefficients $\mathbf{C}_{1,1}^{4,20}$ with $QP_2 = 20$

and $\mathbf{C}_{1,1}^{8}$ by taking the average value of the test (7.30) for each QP present in the video.

The detection results are given in terms of Area Under the Curve (AUC) in Table 7.2 for various QP_1 and QP_2. In the first part of the Table, we can see the results for $\mathrm{QP}_2 < \mathrm{QP}_1$. In such case, we see that our method outperform the state-of-the-art algorithms. In the second part of the Table, we show two examples where $\mathrm{QP}_2 > \mathrm{QP}_1$. We know that for a fixed QP the detection is theoretically not possible for our method based on the DCT coefficients. But for our dataset on smartphone video we saw that the rate control introduced enough perturbation to perform the detection. Here we see that the perturbation does not overcome this limitation which could be explained by the implementation of the H.264 encoder. If the variance around the targeted QP value is lower, then it is more likely that

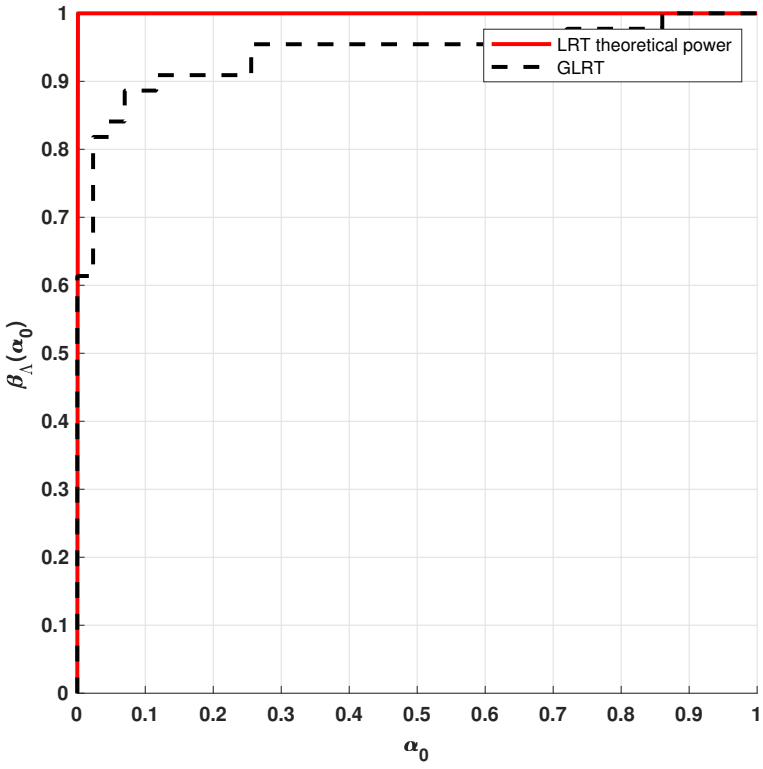


Figure 7.13 : Empirical and theoretical power for the coefficients $\mathbf{C}_{1,1}^{4,23}$ with $QP_2 = 25$

we will have $\mathrm{QP}_2 > \mathrm{QP}_1$ for an individual macroblock. Similarly the method [173] fail in that scenario as it is based on the DCT coefficients. In contrary, G-VPF [165] suffer less in that scenario. The authors of [165] also notice that the performances eventually collapse when $\mathrm{QP}_1 \ll \mathrm{QP}_2$. Here we see that for $\mathrm{QP}_1 = 25$ and $\mathrm{QP}_2 = 30$, the AUC of G-VPF drops to 0.7432.

7.6 Conclusion

In this chapter we proposed a method to detect a double H.264 video compression detection algorithm based on an analysis of the DCT coefficient. We showed that the DCT coefficients can be roughly approximated by a zero mean Laplacian distribution and that the scale parameter is dependent on the quantisation param-

Table 7.2 : Comparison to state-of-the-art methods

Quantization parameter		Method		
QP_1	QP_2	Proposed	SVM-DCT [173]	G-VPF [165]
25	18	**1**	0.9781	0.9580
25	20	**1**	0.7503	0.9170
25	23	**0.9512**	0.6629	0.8926
23	18	**0.9990**	0.8241	0.9561
23	20	**0.9863**	0.7575	0.9111
20	18	**0.9570**	0.5078	0.9268
23	25	0.4453	0.4724	**0.8936**
25	30	0.3028	0.3632	**0.7432**

eter. We thus proposed a statistical test to determine whether or not the observed coefficients follow a Laplacian distribution with a scale parameter b_0 based on the observed QP.

We showed that the detection was only possible when the second quantisation parameter was lower than the first one. Even though this seems like a strong limitation, we showed on real example that in practice this might not be as problematic thanks to the rate control mechanism of H.264. Indeed, in H.264 a single frame can be encoded using many different quantisation parameters. Our experimental evaluation showed that this behaviour introduces enough variation in the difference between the first and second quantisation parameters to make the detection possible.

In future works, many points could be addressed to improve the results of the proposed method. In [182], it was shown that the DCT coefficients for JPEG images could only be assumed i.i.d after suppressing the image content (i.e. the image expectation). Unlike JPEG images, H.264 compression includes a prediction stage prior to the DCT transformation and quantification. In this chapter, we considered this prediction as a rough estimation of the image expectation and thus considered the DCT coefficients to be i.i.d and following a Laplacian distribution. But we can see in Fig. 7.2 that the estimated scale b has a non-negligible variance and on Fig. 7.1 is not perfectly accurate in particular around zero. This suggests that the H.264 prediction may not be considered as a good approximation of the image prediction. In fact, it is not designed to estimate the expectation but rather

to estimate the exact pixel values (noise included).

A first perspective to improve the results of the proposed method would then be to proceed as in [182] by first decoding the H.264 stream in order to compute the expectation and remove it prior to the estimation of the scale parameter.

Another perspective would be to propose a more elaborate model of the DCT coefficients as in [180] by adding the impact of the prediction stage prior to the transformation and quantification.

In this chapter, we proposed a statistical test for a single DCT coefficient at certain quantisation parameters. In practice, it would be interesting to design a method using every coefficient at every QP to maximise the detection performance. A last perspective is to study the application of our method to other video compression algorithms. Here we focused on H.264 compression only but video compression algorithms are often quite similar. For instance, the successor of H.264 (namely H.265) mainly follows the same compression scheme. Similarly VP9 and its successor AV1 also uses a DCT transformation on residual blocks. Moreover, the two latest encoders (i.e. AV1 and H.265) can both be used to perform image compression. This convergence of technologies is a great opportunity to develop forensic algorithms for both images and videos.

CHAPTER 8

Conclusions and Perspectives

8.1 Conclusions

Digital images and videos are parts of our day-to-day lives. Because of their early adoption by news media, we have been used to trust those digital contents and rarely question their authenticity. Unfortunately, the increasing use of such media also came with great advances in photo retouching software making it much easier to produce forged images.

Most recently, Remote Identity Verification systems have become more popular partially due to the COVID-19 pandemic. Such systems rely heavily on digital images and videos to identify and authenticate an individual. Earlier RIV systems would hardly question the authenticity of the digital media itself but rather question the authenticity of the physical document being acquired. Consequently, they would mostly perform verification of the various security elements and their physical properties. While such counterfeiting techniques exist, only a highly skilled counterfeiter may be able to produce convincing results. On the other hand, photo retouching software makes it easier to produce highly realistic tampering even by a non-expert.

A counterfeiter will most likely tamper the photo of the ID document, the text fields or both. We thus focused our research on forgeries targeting such elements. To evaluate image forensic algorithms, representative dataset is needed. As part of the ANR project DEFALS, we decided to propose a novel dataset as no large image forensic datasets were available at the time. We developed an automatic tampering algorithm which allowed us to generate over 200000 tampered images. In this dataset we included face manipulation forgery to encourage research in that field.

In chapter 4, we considered the attacks against ID documents text fields. A counterfeiter will most likely use a Copy-Move forgery to alter the texts present in the ID document. One of the biggest challenges for copy-move forgery detection is the presence of Similar but Genuine Object (SGO). With SGO, copy-move forgery detection algorithms were shown to generate high False Positive Rates (FPR). We proposed a novel hybrid method to reduce the FPR of Keypoint-based detections in the presence of SGO. Typical descriptors used by keypoint-based methods such as SIFT or SURF are designed to be robust to many transformations. While this is a

desired property for object detection in general. It tends to generate a high number of False Positive in the presence of SGO. We thus added a novel filtering stage based on the Local Dissimilarity Map (LDM). Each matched keypoints are filtered by computing the LDM between the two neighbourhoods of those keypoints. We also proposed a novel dataset containing forged ID documents. We showed that our method outperform state of the arts methods on this novel dataset and also common datasets.

A counterfeiter might first need to remove some text before using copy-move to tamper it. In such case he would use an object removal forgery. In chapter 5, we proposed to detect such forgeries. When performing an object removal, counterfeiters will often apply a smooth transition to seamlessly blend the recreated texture. This produces a softer texture around the tampered area. We proposed a novel local sharpness measure based on the reflectance of the image to expose those artefacts. We showed that this method provides an easily interpretable feature map which can reveal object removal forgeries. We also showed that in some cases this method can expose splicing forgeries.

In chapter 6 we studied the face morphing attack, a specific attack against the ID document picture. This forgery represents a serious threat against RIV systems as it can allow two peoples to share an ID document. Because a Face Morphing attack may be performed during the issuing process, many detection methods try to detect digital attacks but also in a print and scan scenario. We proposed a fully digital detection method based on the analysis of the noise. We showed that a Face Morphing attack significantly decreases the variance of the noise in the face area due to the warping and blending of the two faces. We showed that our method can be fooled by a well-designed counterforensic. We also showed that state-of-the-art methods can also be fooled similarly. We showed that the performances of many state-of-the-art methods also drop significantly as the quality of the authentic image decrease. From there we questioned the applicability of no-reference face morphing detection methods for completely blind detection (i.e. possibly print and scan). In RIV system, no-reference face morphing detection methods are still relevant as it is likely that the face morphing attack will be performed digitally.

Finally, in chapter 7, we studied the detection of doubly compressed video using H.264. A RIV system acquires one stream of the person faces and one stream

of an ID documents. Both the person and the ID documents are dynamic elements it thus makes sense to acquire a video stream rather than simple images to authenticate those. In fact, the new PVID regulation recently enforced RIV providers to perform a video acquisition of those elements. We focused our attention on the H.264 compression algorithm as it is the most widely used and also mandatory in the WebRTC protocol used to stream video content. To tamper an H.264 stream, the counterfeiter will first need to decode it. Then he will perform his tampering and finally will recompress it using H.264 or another algorithm. The ability to detect double H.264 compression is thus a first step toward the detection of forged video content. We studied the distribution of the DCT coefficients of an H.264 video. We showed that they follow a zero mean Laplacian distribution. The parameter b depends on the quality factor and also the transformation size (e.g. 4×4 or 8×8). We proposed a Generalised Likelihood Ratio Test (GLRT) to detect the double compression of H.264 videos. We showed that our method outperforms current state-of-the-art approaches.

8.2 Perspectives

In this thesis we addressed the various attacks against a RIV system. Although the proposed works are effective and have been applied to some real-world example successfully, many challenges have yet to be tackled.

In chapter 4, we proposed an approach to reduce FPR of keypoint-based copy-move detection methods.Even though it outperforms current state-of-the-art methods on tampered ID documents, we saw that the results were not yet good enough. As is, the block-based filtering is applied on a per keypoint basis. Only after the clusters containing less than a fixed number of keypoints are considered as copy-move. This is a strong limitation for the detection of small forged region. In fact, small region will inevitably have fewer keypoints and thus have higher chances of being discarded. One approach that could solve this problem would be to perform the filtering on a per cluster basis. For example, the objects corresponding to the cluster of keypoints could be segmented and compared to each other. Another approach could be to compute an overall similarity score by combining the scores

obtain per keypoint.

In chapter 5, the proposed method still requires a visual inspection. In RIV systems this may not be an issue as a human supervisor could double check those results but for a more general forensic application, the decision may need to be automated. The method proposed in 5 reveal the boundary of the forged area. On approach to automate the detection would be to compare this feature map with the image and look for weak edges in the feature map that are absent from the original image. The local dissimilarity map could be used to perform such a task. Such an approach could be used to automate the detection of object-removal forgery but would not allow the automatic detection of splicing.

In chapter6, we proposed a new method for face morphing detection. For the face morphing attack, we fixed the parameters α_b and α_l to 0.5. In reality, it is very unlikely that those parameters will be constant. More studies on the impact of those parameters have yet to be done. Ideally, future work should focus on differential face morphing detection which is applicable in most contexts. Also, because the face morphing attack could occur during the issuing process physical seal or watermarks should be considered to secure the ID photo.

In chapter 7, we proposed an approach for the detection of doubly compressed video. The current modelling of the DCT coefficients does not take the block content into account which is not completely accurate. A more accurate model should be evaluated. Currently, no elaborate methods have been proposed to combine the scores obtained for various DCT subbands. Further works are needed to increase the detection accuracy. At the moment, the method is not applicable when the second compression is stronger. Other approaches must be studied to cover this scenario. As many video compression algorithms use the DCT transformation internally, it would be interesting to evaluate the performance of our method for other compression algorithms.

Recently, the PVID regulation enforced the use of video to acquire both the face and the ID documents. Many methods in this thesis were designed to be applied to images. Future works could also extend those methods to be applicable to videos. The use of videos would also allow us to use liveness verification and take advantage of the dynamic nature of certain security elements such as the holograms or variable inks. A carefully designed user experience could be used to

design robust physical-based forgery detection on those security features. Newer approaches could take advantage of possible temporal inconsistencies due to the added difficulty for a counterfeiter to tamper a video. Finally, the latest video compression algorithms can all be used for image compression. This may allow researchers to develop newer method applicable to both images and video. But also to combine both images and video forgery detection. For example, the same encoder could provide high quality images alongside a lower-quality video. This would allow the use of precise camera-based approach on the images while still exposing the temporal or physical inconsistencies in the video.

APPENDIX A

French Summary

A.1 Introduction

Les images et les vidéos font partie intégrante de notre quotidien. Que ce soit lorsque l'on visionne un documentaire, qu'on lise un journal ou encore que l'on partage des souvenirs sur des réseaux sociaux. Nous utilisons des images et des vidéos pour illustrer nos idées et partager une information.

L'usage de la photographie dans le journalisme n'est pas nouveau. En réalité, on peut retracer l'histoire du photojournalisme jusqu'en 1848 ou le journal français «L'illustration» utilisa une photographie pour illustrer son article sur les barricades de juin 1848 à Paris. Avec plus d'un siècle de photojournalisme, nous avons été accoutumés à l'utilisation de la photographie comme témoignage d'une réalité. Le proverbe «une image vaut mille mots» n'a jamais été plus juste qu'aujourd'hui.

Avec la transition du support photographique analogique vers les technologies numériques, ce type de médias a vu son utilisation croître de manière exponentielle. Cette forte croissance s'explique notamment par la simplicité de partage et d'accès de ces médias sur internet. On constate aujourd'hui que la plupart des journaux proposent maintenant une version numérique de leurs articles ou même de courtes vidéos largement diffusées sur les réseaux sociaux. En parallèle, les technologies de retouche photographique ont largement évoluées. Si la retouche photographique était un domaine de niche à l'ère de la photographie argentique, c'est maintenant une pratique commune et accessible au plus grand nombre.

Les outils de retouche tels que Photoshop, Affinity Photo ou encore Gimp permettent aujourd'hui de produire simplement des retouches photographiques d'une grande qualité. L'utilisation massive des images et vidéos ainsi que l'évolution des logiciels de retouche ont progressivement fait grandir les craintes concernant la falsification d'image et la propagation des fausses informations.

Récemment, des régulations visant à lutter contre le blanchiment d'argent on fait leurs apparitions en Europe. De ces régulations est né le concept de «Know Your Customer» (KYC) ou «Connaître ses Utilisateurs». L'idée du KYC est d'imposer les institutions financières à mettre en place des mécanismes permettant de vérifier l'identité de ses clients afin de limiter les risques de blanchiment d'argent. Quand le client est un individu, cela revient en la vérification d'une pièce d'identité officielle. Initialement ces vérifications se faisaient en présentiel, mais la crise du

COVID19 a favorisé le développement d'alternative distancielle à la fois plus simple pour l'utilisateur et pour l'institution financière. Ce nouveau mode de KYC est communément appelé «Electronic-KYC» ou eKYC. Lors d'un eKYC, l'utilisateur est amené à envoyer une photo de son document d'identité ainsi qu'une photo de lui-même. L'institution financière cherche alors à déterminer si le document est authentique et si la personne en est le propriétaire.

Dans ce contexte, l'authentification des images et vidéos est essentielle puisque la sécurité du reste du système reposera intégralement sur cette vérification. Un système de vérification d'identité à distance peut être sujet à de multiples attaques telles que le vol d'identité, la création de fausse identité, la modification de tout ou partie de l'identité, etc.

Dans cette thèse, nous nous concentrons sur la sécurisation de tels systèmes de vérification d'identité à distance. En particulier, nous étudierons les falsifications numériques qui peuvent être effectuées sur les images et vidéos et proposerons des méthodes permettant d'authentifier ces médias.

Dans un premier temps, nous présenterons la base de données DEFACTO qui a été développée afin de favoriser la recherche sur la détection de falsifications numériques. Ensuite, nous étudierons les falsifications de type Copier-Coller qui permettent notamment de falsifier les textes d'un document d'identité. Nous aborderons ensuite la suppression d'objet qui est souvent nécessaire pour produire une falsification. Puis nous étudierons les attaques de type Face Morphing qui présentent une menace particulière contre les systèmes de vérification d'identité à distance. Enfin nous aborderons la détection de la double compression vidéo afin de sécuriser le flux vidéo après son acquisition.

A.2 Base de données DEFACTO

Afin d'étudier et de comparer différentes méthodes de détection de falsification d'images, des bases de données représentatives sont nécessaires. Cependant, la création manuelle de telles bases de données nécessite un travail conséquent et seule une petite quantité de données pourraient être produites dans un temps raisonnable. Pendant de nombreuses années, seules des bases de relativement

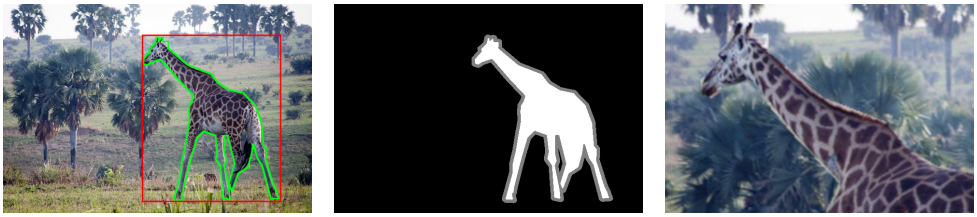
petites tailles étaient disponibles. La plus grande d'entre elles, CASIA [69] ne contient par exemple qu'un peu plus de 3200 images. Pour des méthodes de détection dîtes sans apprentissage ceci ne représentait pas un problème puisque l'intégralité des 3200 images pouvaient être utilisée pour évaluer les algorithmes. Cependant les méthodes basées sur l'apprentissage ont commencé à devenir de plus en plus populaires et nécessitent quant à elle une base d'apprentissage et une base d'évaluation distincte. En particulier, les méthodes basées sur l'apprentissage profond nécessitent des bases d'apprentissage extrêmement large pour réaliser l'apprentissage.

Dans le cadre du projet ANR-16-DEFA-0002, il était envisagé d'utiliser des méthodes basées sur l'apprentissage profond. Dès lors, il est apparu nécessaire de constituer un corpus d'apprentissage conséquent. Les objectifs de ce corpus étaient multiples. La base devait tout d'abord contenir un nombre d'images falsifié conséquent afin de permettre l'apprentissage d'un réseau de neurones profond. Tout type de falsification devait être présent dans ce corpus. Et finalement, les falsifications devaient être le plus réalistes possible.

Pour produire un nombre d'images conséquent, il n'était pas envisageable de créer les images manuellement. Nous avons donc mis au point un algorithme permettant de générer automatiquement des falsifications. Pour que les falsifications soient le plus réalistes possible, nous nous sommes basés sur la base de données MSCOCO [78] contenant de nombreuses images annotées. Ces annotations étaient nécessaires pour la production de falsifications réalistes. Dans la section suivante, nous allons détailler le fonctionnement de l'algorithme de falsification automatique.

A.2.1 Algorithme de falsification automatique

Une falsification réaliste implique d'insérer, de dupliquer ou encore de supprimer un objet significatif dans une image. En effet, la simple insertion ou duplication d'élément aléatoire n'aurait pas de sens sémantique et serait simplement perceptible à l'oeil. Il est donc nécessaire d'avoir accès à des images annotées afin de produire des falsifications réalistes. Cependant la création de telles annotations représente un travail tout aussi conséquent que la création d'images falsifiées. Nous avons donc utilisé les images de la base MSCOCO [78] afin d'obtenir des annotations



(a) Annotations brutes MSCOCO

(b) Trimap générée

(c) Copier-coller sans alpha matting

(d) Copier-coller avec alpha matting

Figure A.1 : Amélioration des annotations de MSCOCO

d'objets significatifs (Personnes, Avions, Paneaux, etc.).

Cependant les annotations de la base MSCOCO n'étaient pas assez précises et ne permettaient donc pas de créer des falsifications convaincantes. Il était donc nécessaire d'améliorer automatiquement ces annotations afin de produire des falsifications réalistes. Nous avons eu recours à un algorithme d'Alpha Matting [79] permettant de générer automatiquement une segmentation précise. Cet algorithme d'Alpha Matting nécessite une trimap. Une trimap comporte trois régions (voir Fig. A.1b) délimitant l'arrière-plan en noir, le premier plan en blanc et une zone d'incertitude en gris. L'objectif de l'algorithme d'alpha matting et de déterminer à quel plan appartient chaque pixel de la zone d'incertitude.

On peut voir en Fig. A.1 la nette amélioration de la qualité visuelle de la falsification grâce à l'utilisation d'une méthode d'alpha matting. Cette méthode d'Alpha matting est nécessaire pour la production de falsification de type insertion ou copier-coller. Pour la création de suppression, un masque précis n'est pas nécessaire. Les annotations de MSCOCO ont alors simplement été dilatées pour s'assurer que le masque contient bien l'intégralité de l'objet à supprimer.

La création d'un masque de bonne qualité n'est pas suffisante pour garantir un

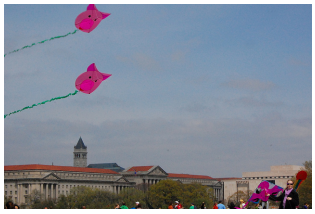
(a) Authentique (b) Falsifiée

Figure A.2 : Exemple d'insertion

résultat final convaincant. En effet, une des problématiques majeures est de placer l'élément dupliqué ou inséré à un endroit cohérent dans l'image finale. Dans le cas d'une falsification de type insertion, cette tache peut se révéler particulièrement ardue, voire même impossible. Effectivement, si l'élément inséré n'a pas été photographié selon le même angle de vue que l'image cible, alors il peut être tout bonnement impossible d'obtenir un résultat réaliste. Pour relever ce défi, nous avons donc limité nos falsifications à un ensemble d'objets ne variant que peu en fonction de l'angle de vue. Par exemple, une balle de foot sphérique aura toujours le même aspect, quel que soit notre point de vue. Il est alors aisé de l'insérer dans une autre image. Un exemple de ce type de falsification est donné en Fig. A.2.

De la même manière, lors de la réalisation d'un copier-coller, nous avons contraint le placement de l'élément dupliqué. En plaçant celui-ci sur le même axe que l'objet initial, on s'assure que les perspectives sont respectées ce qui permet d'obtenir des résultats plus convaincants. Des exemples sont donnés en Fig. A.3.

Pour la suppression, les contraintes sont moins importantes et surtout liées à l'algorithme de suppression. Nous avons utilisé une méthode basée sur l'exemple [84] qui nécessite que l'objet à supprimer se trouve sur un fond relativement simple. Nous avons donc limité la suppression aux objets se trouvant sur un fond simple en appliquant un seuil sur la variance de l'image dans un voisinage proche de l'objet. Un exemple est donné en Fig. A.4



(a) Copier-Coller sur l'axe y

(b) Copier-Coller sur l'axe x

Figure A.3 : Exemple de Copier-Coller

(a) Authentique

(b) Falsifiée

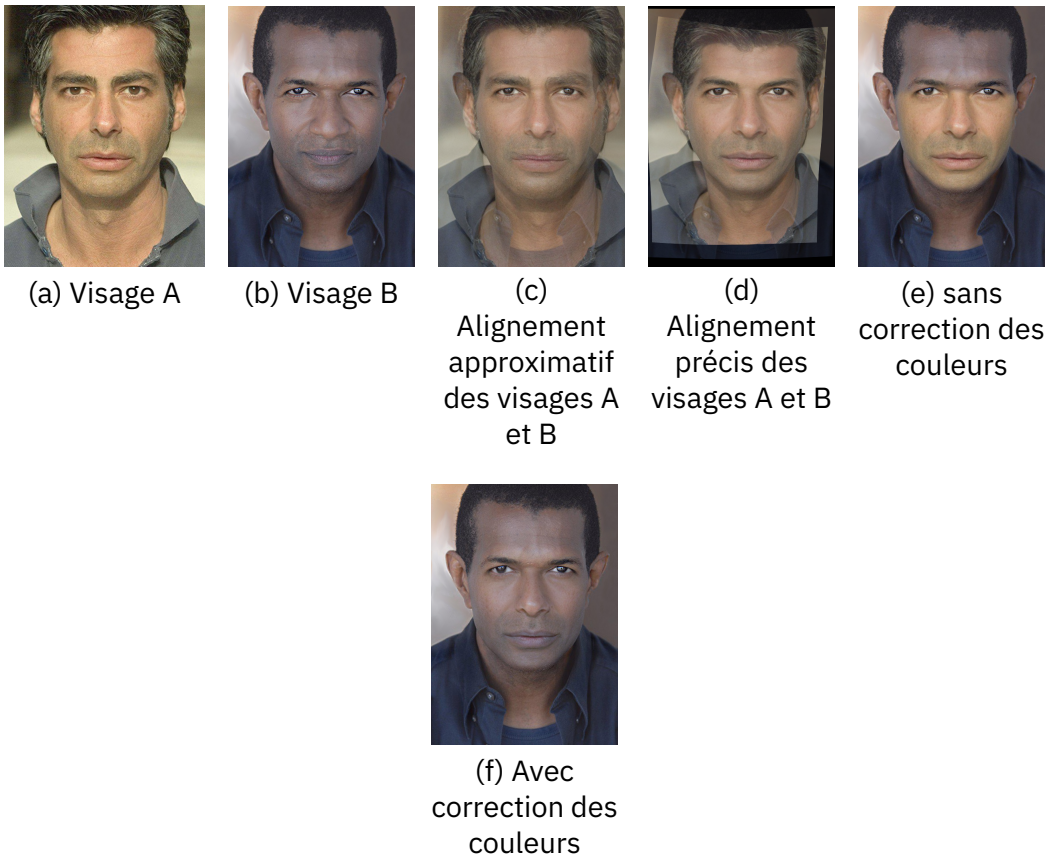
Figure A.4 : Exemple de suppression

Enfin, nous avons aussi généré deux types de falsification supplémentaire. Les morphoses et les remplacements de visage. La génération de ces falsifications peut être vue en Fig. A.5. Dans un premier temps les deux visages sont approximativement alignés. Puis le visage cible est déformé (ou non selon la falsification voulue). Enfin une étape de correction de couleur et de mélange permet de produire la falsification finale.

A.2.2 Résultats

Grâce à l'algorithme de falsification présenté précédemment, il a été possible de générer une grande quantité d'images falsifiées. Ceci a permis de construire une base de données conséquente et significative permettant d'entraîner des réseaux de neurones profonds.

Le contenu de la base de données est détaillé en Table. A.1.



(a) Visage A (b) Visage B (c) Alignement approximatif des visages A et B (d) Alignement précis des visages A et B (e) sans correction des couleurs

(f) Avec correction des couleurs

Figure A.5 : Étape de création de morphose ou remplacement de visage

Table A.1 : Contenu de la base DEFACTO

Falsifications	Copier-Coller	Suppression	Insertion	Morphose
Images	19000	25000	105000	80000

A.3 Détection du Copier-Coller

Une méthode très commune de falsification des images numériques est couramment appelée le copier-coller. Cette falsification élémentaire consiste en la duplication d'une portion de l'image. L'élément dupliqué n'est pas contraint en taille et peut subir une déformation affine avant d'être à nouveau collé dans l'image. Il est intéressant de pouvoir détecter un tel type de falsification puisque cette opération basique est couramment utilisée dans les photomontages.

Cette méthode peut servir à la simple duplication d'un élément pour tromper sur une quantité. Ou encore pour la suppression d'un élément sur une image en le cachant derrière un ou plusieurs éléments dupliqués.

Dans le cadre des systèmes de vérification d'identité à distance, le copier-coller sera particulièrement efficace pour falsifier les textes du document. En effet, un contrefacteur pourra simplement copier-coller certaines lettres déjà présentes afin d'altérer un ou plusieurs champs de textes. La détection du copier-coller est un sujet déjà bien étudié. Ici nous effectuons la détection à l'aide des points clés SIFT et proposons l'utilisation d'une carte de dissimilarités locales [117] pour permettre une détection plus précise. Une telle amélioration est nécessaire, car les méthodes de détections de copier-coller sont très sensibles aux éléments similaires et tendent à produire de nombreux faux positifs en présence de ceux-ci. Or les lettres sont extrêmement similaires dans un document d'identité ce qui rend la plupart des approches de l'état de l'art inutilisable dans un cadre pratique.

Dans un premier temps nous présenterons une nouvelle approche pour la détection du copier-coller. Puis, nous montrerons qu'il est possible de grandement réduire le taux de faux positifs à l'aide de cartes de dissimilarités locales. Enfin, nous proposerons une nouvelle base de données contenant de nombreux documents d'identité falsifiés afin d'encourager la recherche dans ce contexte difficile.

A.3.1 Principe de la méthode

La méthode de détection peut être découpée en deux phases distinctes. Dans un premier temps, les falsifications sont détectées de manière grossière à l'aide de points clés SIFT. Les points clés sont extraits de l'image et mis en correspondance. Ils sont ensuite partitionnés afin de grouper les points appartenant à un même objet de l'image, mais aussi de supprimer les faux positifs évidents. Cette première détection est efficace, mais très sensible à la présence d'objets similaires.

Une deuxième étape vise donc à supprimer les faux positifs issus de la mise en correspondance et ayant résisté à l'étape de partitionnement. Dans cette étape, les points clés de chaque partition sont filtrés un à un à l'aide de carte de dissimilarités locales. À l'issue de cette étape, seules les partions contenant suffisamment de points clés sont finalement considérés comme des zones falsifiées.

Dans les sections suivantes, nous présenterons chaque étape de la détection plus en détail.

A.3.2 Extraction des points clés

L'extraction des points clés est effectuée à l'aide du détecteur de la méthode SIFT [113]. Plutôt que d'effectuer une détection globale des points clés sur l'image, la détection est faite au travers d'une fenêtre glissante non superposée. Les méthodes de détection du copier-coller basées sur les points clés nécessitent que les régions dupliquées soient couvertes par un nombre suffisant de points clés. Pour pallier ce problème, le seuil de rejet sur le contraste est ajusté pour extraire une quantité de points clés SIFT suffisante pour chaque portion de l'image. Ceci permet de conserver les points clés sur les contours et dans les zones de gradient faible.

A.3.3 Mise en correspondance

La mise en correspondance classique de descripteur SIFT décrit dans [113] est la méthode $2NN$. Pour un descripteur donné, les deux voisins les plus proches avec des distances d_1 et d_2 sont trouvés. On considère qu'une mise en correspondance est positive si le rapport $\frac{d_1}{d_2}$ est inférieur à un seuil δ.

Une seule mise en correspondance est considérée par le test $2NN$. Dans le cadre de la détection du copier-coller, l'hypothèse qu'un élément ne sera dupliqué que de manière unique n'est pas raisonnable. Pour cette raison les auteurs de [93] proposent $g2NN$, une généralisation de $2NN$. Le test $2NN$ est itéré pour les k plus proches voisins d'un point clé donné. On teste alors les rapports $\frac{d_i}{d_{i+1}}$ tant que ceux-ci sont inférieurs à un seuil δ fixé. Nous avons utilisé ce test généralisé qui permet une détection du copier-coller multiple.

A.3.4 Partitionnement

À l'étape de mise en correspondance, le seuil δ choisi ne peut être trop strict si l'on veut pouvoir détecter un copier-coller ayant subi un quelconque post-traitement (réechantillonage, ajustement des couleurs ou du contraste ...). Ce qui entraîne

l'obtention de faux positifs à ce stade. Un deuxième filtrage est nécessaire pour supprimer un maximum de fausses alarmes.

Nous effectuons un partitionnement à l'aide de la méthode décrite dans [116] qui permet de grouper des éléments selon des règles d'équivalences. Soit un objet O de l'image, et O_D sa copie (figure A.6). Soit A et C des points clés dans l'objet O et les points clés B et D dans O_D. Soit la mise en correspondance M_{AB} entre A et B formant le vecteur $\overrightarrow{AB}$ et la mise en correspondance M_{CD} entre C et D formant le vecteur $\overrightarrow{CD}$. Les mises en correspondance M_{AB} et M_{CD} sont considérées équivalentes et sont groupées si :

$$\|\overrightarrow{AB} - \overrightarrow{CD}\| < \delta_1 \tag{A.1}$$

$$\|AC\| < \delta_2 \text{ et } \|BD\| < \delta_2, \tag{A.2}$$

$$\text{et } \|AB\| > \delta_3 \text{ et } \|CD\| > \delta_3. \tag{A.3}$$

Sur la figure A.6 l'objet O est dupliqué en O_D avec une légère rotation. Dans le cas d'une duplication avec une déformation simple (rotation, échelle), toutes les mises en correspondance ont une orientation et une norme similaire.

Le seuil δ_1 permet de limiter l'écart entre deux vecteurs formés par deux paires de points mis en correspondance. Ceci permet de grouper les mises en correspondance d'orientation proches. Un seuil δ_1 faible tendra à augmenter le nombre de partitions. Un objet dupliqué subissant une rotation sera décomposé en plusieurs partitions de plus petite taille. Un seuil δ_1 important tendra à diminuer le nombre de partitions. De faux positifs risquent alors d'être inclus dans les partitions des objets dupliqués. Ce seuil ne dépend pas de la taille de l'image analysée. Dans toutes nos expérimentations, δ_1 est fixé à 10.

La distance entre A et C est limitée par la taille de l'objet O dupliqué. Le seuil δ_2 fixe la taille maximale de l'objet dupliqué que l'on pourra détecter. Un seuil δ_2 trop grand tendra à ajouter des faux positifs dans les partitions des objets dupliqués. Celui-ci ne peut être trop faible non plus en raison de la densité parfois plus faible de point SIFT extrait.

A et B sont nécessairement à une distance supérieure à un seuil donné , de même pour C et D, si l'on veut dupliquer O intégralement. Le seuil δ_3 fixe la distance minimale entre l'objet O et sa copie O_D que l'on souhaite détecter. La

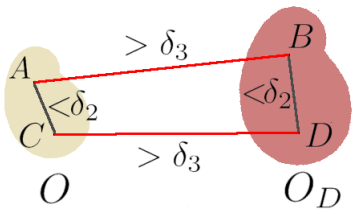


Figure A.6 : Duplication de O et règles d'équivalences

valeur de δ_3 sera principalement choisie en fonction de la taille de l'image analysée. Dans nos expérimentations, les valeurs de δ_2 et δ_3 sont fixées à 50.

Le choix des valeurs δ_1, δ_2 et δ_3 n'est pas critique si le seuil sur le contraste en A.3.2 est choisi arbitrairement bas.

À l'issue du partitionnement, les partitions comportant trop peu éléments (moins de trois paires de points mises en correspondance) sont écartées.

A.3.5 Filtrage avec la carte de dissimilarité locale

Le descripteur SIFT est l'histogramme des gradients orientés autour du point clé. Ce descripteur ne contient que peu d'information concernant la structure de l'image autour du point clé.

Dans le cadre de la détection du copier-coller, de nombreux faux positifs peuvent être produits dans le cas d'une image avec des structures répétitives (façade d'un bâtiment, textes ...). L'utilisation des informations de couleurs et de structures permettrait de rejeter de faux positifs évidents. Nous proposons d'utiliser la carte de dissimilarité locale (CDL) afin de filtrer ces faux positifs.

La carte de dissimilarité locale [117] permet de mesurer les écarts locaux entre deux images binaires. Pour deux images binaires A et B, la CDL est définie de $\mathbb{R}^2 \times \mathbb{R}^2$ dans $\mathbb{R}^2$ par

$$\mathrm{CDL}_{\mathrm{bin}}(A, B)(p) = \mid A(p) - B(p) \mid \max(d_A(p), d_B(p)) \tag{A.4}$$

avec $p = (x, y)$ et $d_X(p)$ la transformée en distance de X au point p.

Une extension de la CDL aux images en niveau de gris est utilisée [118]. Les images sont dans un premier temps découpées en un ensemble d'images binaires.

La CDL en niveau de gris est alors l'accumulation des CDL entre chacune de ces images binaires. Soient A et B deux images en niveau de gris, CDL_N est alors définie de $\mathbb{R}^2 \times \mathbb{R}^2$ dans $\mathbb{R}^2$ par

$$\mathrm{CDL}(A, B)(p) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{CDL}_{\mathrm{bin}}(\mathrm{A_i}, \mathrm{B_i})(\mathrm{p}) \tag{A.5}$$

Où N est le nombre de coupes, A_i (resp. B_i) est une version binaire de A (resp. B_i), obtenue par seuillage global $A > s_i$. Les seuils s_i sont régulièrement espacés entre 0 et le maximum m de chaque image. Par exemple, pour A : $s_i = \frac{i}{N} m_A, i \in [1..N]$.

Pour le copier-coller, nous utilisons une extension directe de la CDL pour des images avec C canaux. Dans notre cas les images sont converties dans l'espace colorimétrique CIE XYZ, qui propose une répartition des couleurs se rapprochant de celle du système visuel humain. Soit A^k le canal $k \in (\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ de l'image A, la CDL est définie comme :

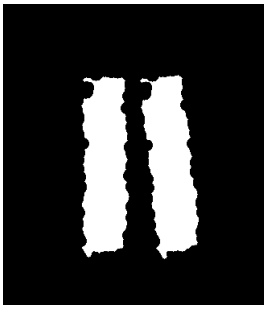
$$\mathrm{CDL}^{\mathrm{XYZ}}(A, B)(p) = \frac{1}{3} \sum_{k} \mathrm{CDL}_N(A^k, B^k)(p). \tag{A.6}$$

Pour supprimer les derniers faux positifs, les partitions issues du partitionnement (section A.3.4) vont être validées à l'aide des CDL. Pour chaque paire de points clés mis en correspondances de la partition, deux fenêtres, F_1 et F_2, sont extraites pour chacun des points clés. Les fenêtres sont centrées sur les points clés et leurs tailles sont fixées par l'échelle des points clés associés. Le contenu des deux fenêtres est aligné en fonction de l'angle des points clés associés. Finalement, la paire de points est supprimée de la partition si

$$\|\mathrm{CDL}^{\mathrm{XYZ}}(F_1, F_2)\|_2 > \delta_{\mathrm{CDL}}. \tag{A.7}$$

Comme pour A.3.4, la partition est finalement supprimée si elle contient moins de trois paires de points clés. On peut voir le résultat de la détection après filtrage dans la figure A.7e.

(a) Image authentique (b) Image falsifiée (c) Vérité terrain (d) Détection sans CDL^{XYZ}



(e) Detection avec CDL^{XYZ}

Figure A.7 : Images de COVERAGE, CDL^{XYZ} et exemple de détection

A.3.6 Résultats

La méthode a été évaluée sur un ensemble de base de données et comparé à différentes approches de l'état de l'art. Le seuil δ_{CDL} est fixé à 7 et les seuils δ_1, δ_2 et δ_3 sont fixés pour chacune des bases pour maximiser le taux de vrais positifs.

Les différents résultats sont présentés en Table. A.2. On peut voir que l'ajout de la CDL permet de surpasser les méthodes de l'état en diminuant très significativement le taux de faux positifs tout en maintenant un taux de vrai positif satisfaisant.

Table A.2 : TPR, FPR, F_1 (niveau image) sur FAU [71] GRIP [94] et COVERAGE [73] et F_P (niveau pixel) sur GRIP

Méthode	COVERAGE [73]			FAU [71]			GRIP [94]			
	TPR	FPR	F_1	TPR	FPR	F_1	TPR	FPR	F_1	F_P
Amerini [93]	85.71	54.95	71.23	66.67	10.42	75.29	70	20	73.68	-
Cozzolino [94]	59.34	21.98	65.45	97.92	8.33	94.95	98.75	8.75	95.18	92.99
J. Li [96]	87.91	63.74	69.87	72.92	22.92	74.47	83.75	35	76.57	27.24
Y. Li [99]	80.22	41.76	72.28	100	2.08	98.97	100	0	100	94.66
Proposed	70.02	7.89	78.73	100	0	100	100	0	100	82.40

(a) Image d'un document (b) Binarisation des lettres (c) Détection des lettres (d) Falsification

Figure A.8 : Étape de falsification du document

Type de document	Authentique	Falsifié	Taille des images
16	304	893	1342×943

Table A.3 : Contenu de la base CMID

A.3.7 Base de données CMID

On a vu que la méthode permettait de faire diminuer significativement le taux de faux positif. Cependant les résultats ne sont toujours pas satisfaisants en présence d'objets hautement similaires. On peut voir dans la Table. A.2 que le meilleur taux de vrais positifs atteint est d'environ 88% pour un taux de faux positif d'environ 64%. Notre méthode permet d'atteindre le meilleur compromis avec un taux de vrai positif d'environ 70% pour un taux de faux positif d'environ 8%. En pratique ce taux de vrai positif n'est pas satisfaisant. Malheureusement, COVERAGE est actuellement la seule base de données permettant d'évaluer les algorithmes dans ce contexte défavorable et ne contient que 100 images falsifiées. De plus, même si les images falsifiées contiennent des objets similaires et authentiques. Les falsifications sont relativement larges et ne représentent pas un usage réaliste du copier-coller.

Nous avons donc constitué une nouvelle base de données plus conséquente et représentant un cas d'usage pratique du copier-coller. La base CMID est constituée de document d'identité falsifié. Pour la constituer, nous avons mis au point un algorithme permettant de détecter les lettres du document, puis de dupliquer l'une d'entre elles (voir Fig. A.8). Ainsi nous avons pu produire 893 images falsifiées. Le contenu exact de la base de données est donné en Table. A.3.

Méthode	TPR	FPR	MCC
SURF [71]	0.7919	0.7697	0.0236
SIFT [71]	**0.9676**	0.9145	0.1104
BusterNet [109]	0.1601	0.1607	-0.0006
FE-CMFD [99]	0.0246	**0.0066**	0.0561
SIFT-LDM [2]	0.7917	0.0197	**0.6847**

Table A.4 : Score niveau image

Méthode	TPR	FDR	F_1	MCC
SURF [71]	0.2155	0.9792	0.0378	0.0606
SIFT [71]	**0.6004**	0.9610	0.0731	0.1471
BusterNet [109]	0.0016	0.9979	0.0018	0
FE-CMFD [99]	0.0341	0.3114	0.0650	0.1530
SIFT-LDM [2]	0.2555	**0.0541**	**0.4024**	**0.4912**

Table A.5 : Score niveau pixel

A.3.8 Résultats sur la base CMID

Nous avons évalué plusieurs méthodes de l'état de l'art ainsi que notre approche sur cette nouvelle base de données. On peut voir les résultats au niveau image en Table. A.4. On constate que les méthodes ont du mal à maintenir un taux de vrais positifs élevé tout en maintenant un taux de faux positifs faible. De la même manière, on peut voir en Table. A.5 que les résultats ne sont pas satisfaisants au niveau pixel.

On peut voir que notre approche qui a été conçue pour diminuer le taux de faux positif obtient les meilleurs résultats. Cependant ceux-ci ne sont pas encore suffisants pour une utilisation pratique. La base de données CMID est publiquement accessible afin de favoriser la recherche de méthode de détection plus robuste en présence d'objets similaires. Les résultats de notre approche, bien qu'insuffisants, montrent que ce défi peut être relevé s'il est pris en compte.

A.4 Détection de la suppression d'objet

Une manipulation d'image commune est la suppression d'objet. Un ou plusieurs éléments de l'image sont alors effacés en reconstituant l'arrière-plan. Des algorithmes d'inpainting sont souvent utilisés dans ce type de falsification et les bords de la falsification sont souvent floutés afin de rendre celle-ci visuellement imperceptible. En conséquence, la bordure de la texture reconstituée peut apparaître moins nette que le reste de la texture. Nous proposons une méthode de détection de la suppression en mettant en évidence ce phénomène. Pour se faire, nous avons développé une mesure de netteté relative d'un pixel par rapport au reste de l'image. Cette mesure permet de mettre en évidence des zones anormalement lisses de l'image.

Dans les sections suivantes, nous introduirons la méthode de détection proposée.

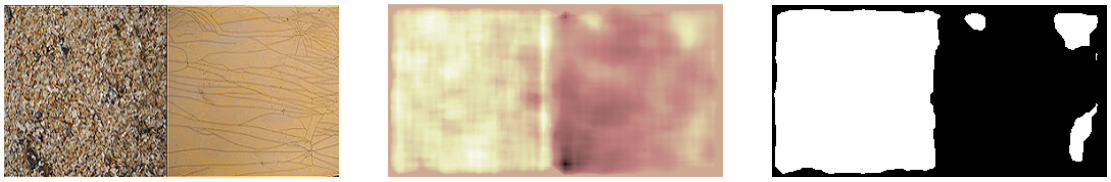
A.4.1 Mesure de netteté basée sur la réflectance

On peut simplifier le processus de formation d'une image en utilisant le modèle d'illumination-réflectance. La réflectance est la capacité d'une surface à réfléchir de la lumière tandis que l'illumination est la quantité de lumière parvenant sur cette surface. Le modèle illumination-réflectance modélise donc chaque pixel (x, y) de l'image comme étant le produire de l'illumination en $(x, y$ et de la réflectance en (x, y). On a donc :

$$I(x, y) = L(x, y) * R(x, y) \tag{A.8}$$

ou $I(x, y)$ est l'intensité du pixel en (x, y), $L(x, y)$ l'illumination et $R(x, y)$ la réflectance. Les variations de L sont typiquement très lentes par rapport à R, il est donc possible d'extraire la réflectance en utilisant un filtrage homomorphe [125]. Tout d'abord on applique une transformation logarithmique à l'image afin de séparer l'illumination et la réflectance en une somme :

$$\ln(I(x, y)) = \ln(L(x, y)) + \ln(R(x, y)). \tag{A.9}$$



(a) Image de deux textures (b) Mesure de netteté relative S (c) Segmentation de S

Figure A.9 : Utilisation de la mesure de netteté S pour segmenter une image

On peut ainsi isoler la réflectance en utilisant un filtre passe-haut :

$$r(u,v) = \mathcal{F}(\ln(I(x,y)))H(u,v) \tag{A.10}$$

avec $\mathcal{F}$ la transformée de Fourier, H un filtre passe-haut dans le domaine fréquentiel et $r(u,v)$ l'estimation de la réflectance dans le domaine fréquentielle.

On obtient l'approximation finale de la réflectance $\hat{R}$ en prenant l'exponentielle de la transformée de Fourier inverse de $r(u,v)$:

$$\hat{R}(x,y) = \exp(\mathcal{F}^{-1}(r(u,v))). \tag{A.11}$$

La réflectance estimée $\hat{R}$ est une approximation grossière, mais suffit comme mesure de netteté. En effet, plus une surface est rugueuse et plus l'estimation $\hat{R}$ sera variable. Ainsi on peut dériver une mesure de netteté locale d'un pixel (x,y) comme étant la variance dans une fenêtre autour de ce pixel. Chaque pixel est donc associé à une mesure de netteté locale $\sigma_{\hat{R}}$ pour obtenir une mesure de netteté relative au reste de l'image, les pixels de celle-ci sont partitionnés en fonction de leur intensité. Pour chaque pixel, sa netteté est calculée comme étant la déviation absolue médiane de sa netteté locale $\sigma_{\hat{R}}$ par rapport aux autres pixels de la même partition. On obtient une carte de netteté relative S dans laquelle les grandes valeurs positives représentent des zones perceptiblement plus nettes tandis que de grandes valeurs négatives indiquent des zones perceptiblement plus lisses. On peut voir en Fig. A.9 un exemple de segmentation basé sur cette mesure S permettant de séparer une texture rugueuse d'une texture plus lisse.

A.4.2 Application à la détection de falsification

Il est commun pour un contrefacteur de flouter les bords de la falsification lors d'une suppression. On espère donc que cet artefact pourra être mis en évidence à l'aide de la mesure de netteté proposée.

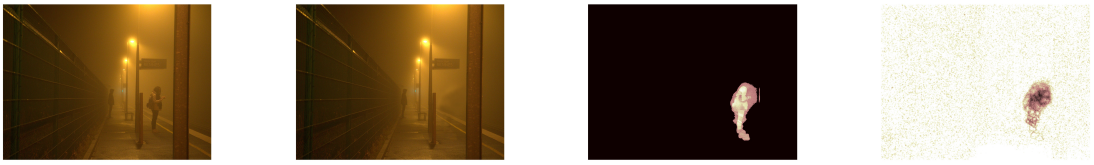
On peut voir sur la figure A.10d, une zone qui apparaît effectivement beaucoup plus lisse que le reste de l'image à l'endroit de la falsification. L'exemple donné en Fig. A.10b a été créé spécifiquement pour évaluer notre approche. Afin de valider cette méthode sur des cas concrets, nous l'avons appliquée sur des images falsifiées (Fig. A.10f et Fig. A.10j) provenant du forum internet «PhotoshopBattle» ou des utilisateurs réalisent des photomontages à partir d'une image donnée. On peut voir que la méthode permet effectivement de mettre en évidence les suppressions pour ces exemples. De plus, on peut voir sur la Fig. A.10h que l'insertion de l'enfant est aussi mise en évidence par notre méthode. Ceci est dû à la forte différence de qualité entre l'image originale et l'image insérée qui fait apparaître l'élément inséré comme beaucoup plus flou que le reste de l'image. Le même phénomène est visible pour la Fig. A.10p. On voit donc que notre méthode peut aussi mettre en évidence des falsifications de types insertion sous certaines conditions.

A.5 Détection du Face Morphing

Dans ce chapitre nous nous intéressons à la détection des attaques de type Morphose de visage. Une morphose de visage consiste en un mélange de deux visages ou plus. Les deux visages sont mélangés comme expliqué en Fig. A.5. L'intérêt d'une telle attaque est de créer un visage synthétique partageant la biométrie de deux individus. Un système de reconnaissance facial est alors trompé par ce visage synthétique et authentifiera les deux personnes ayant servi à générer cette morphose.

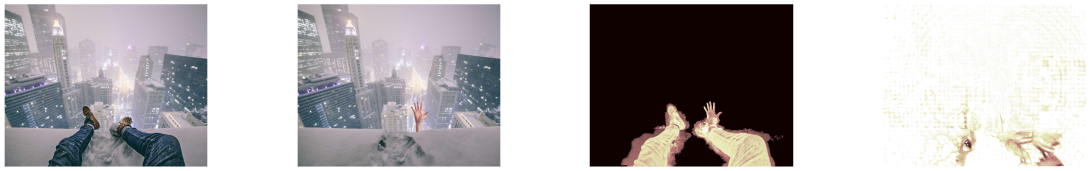
Sur la Fig. A.11, on peut voir au centre une morphose de visage entre le visage de gauche et de droite. Le système de reconnaissance faciale libre de la librairie Dlib [88] authentifie alors les deux visages si l'on utilise le seul recommandé de $0,6$.

Pour un système de vérification d'identité à distance les morphoses de visage



(a) Image authentique (b) Image falsifiée (c) Vérité terrain (d) Carte de netteté S

(e) Image authentique (f) Image falsifiée (g) Vérité terrain (h) Carte de netteté S



(i) Image authentique (j) Image falsifiée (k) Vérité terrain (l) Carte de netteté S

(m) Image authentique (n) Image falsifiée (o) Vérité terrain (p) Carte de netteté S

Figure A.10 : De gauche à droite : Les images authentiques, les images falsifiées, les vérités terrain, la mesure de netteté S

sont une réelle menace puisqu'elles permettraient la création d'identité partagée. Deux contrefacteurs pourraient en effet la photo du document d'identité par une morphose afin de pouvoir plus tard s'authentifier à l'aide d'un même document.

Dans ce chapitre nous proposons une méthode à l'aveugle et sans référence de détection des attaques de types morphoses de visage par analyse du bruit. Notre

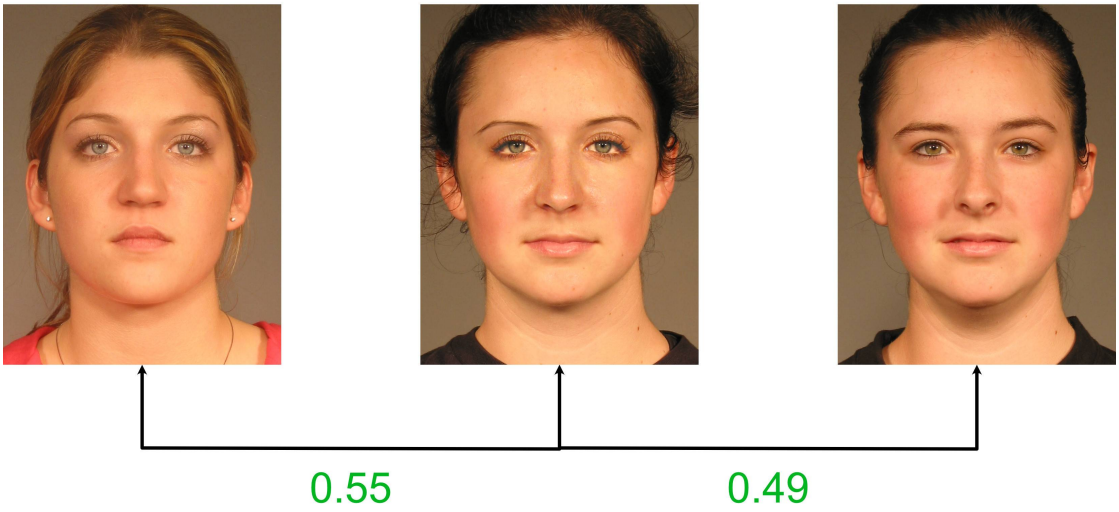
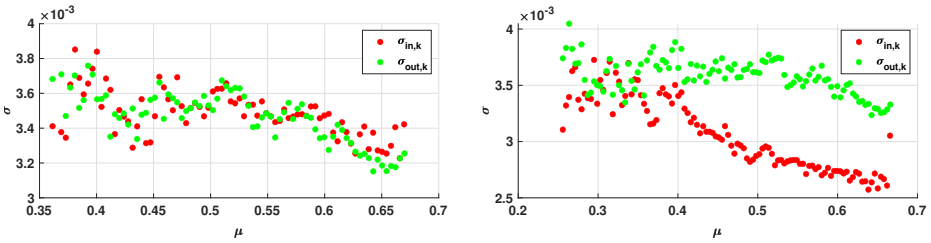


Figure A.11 : Le système de reconnaissance facial authentifie les visages de gauche et de droite avec la morphose au centre pour un seuil de $0,6$.

méthode est à l'aveugle, car elle ne fait pas recourt à un marquage actif de l'image. Elle est sans référence, car la détection est faite uniquement à l'aide du visage synthétique. Nous montrons ensuite qu'il est possible de tromper notre approche à l'aide d'une attaque de contre forensique ciblée. Nous montrons que cette même attaque permet de tromper d'autres méthodes de l'état de l'art à l'aveugle et sans références. Nous montrons ensuite les difficultés de certaines de ces méthodes vis-à-vis d'un changement de qualité des images authentiques. Ces difficultés ainsi que la possibilité de créer des algorithmes de contre forensique efficace mettent en doute la possibilité d'utilisé des méthodes à l'aveugle et sans référence pour certains scénarii d'usage des morphoses de visages.

A.5.1 Détection des Morphoses par analyse du bruit

La morphose d'un visage A et d'un visage B peut être formalisée de la manière suivante. Étant donné un alpha matte $\boldsymbol{\alpha_b}$ avec $\boldsymbol{\alpha_b}_{i,j} \in [0,1], \forall(i,j)$. L'image finale



(a) $\sigma_{int,k}$ et $\sigma_{ext,k}$ pour un visage authentique

(b) $\sigma_{int,k}$ et $\sigma_{ext,k}$ pour une morphose

Figure A.12 : $\sigma_{int,k}$ et $\sigma_{ext,k}$

M^{AB} est donnée par

$$M^{AB} = \boldsymbol{\alpha_b} \circ B^A + (1 - \boldsymbol{\alpha_b}) \circ I^B \tag{A.12}$$

avec B^A l'image du visage A déformé et dont les couleurs ont été corrigées et I^B l'image du visage B déformé.

Dans l'image résultante M^{AB} le comportement du bruit sera donc celui d'une image naturelle en tout point ou $\boldsymbol{\alpha_b} = 0$. Pour $\boldsymbol{\alpha_b} \neq 0$, M^{AB} est donc une moyenne pondérée de deux images interpolées.

On peut montrer que pour des pixels appartenant à une partition k d'intensité proche, la variance $\sigma^2_{int,k}$ des pixels de la partition k à l'intérieur du visage est donnée par :

$$\sigma^2_{int,k} = (\alpha_b \sigma'_{int,k})^2 + ((1 - \alpha_b)\sigma_{ext,k})^2. \tag{A.13}$$

On s'attend donc à ce que la variance du bruit soit plus faible dans la région ou $\boldsymbol{\alpha_b} \neq 0$.

On peut voir en Fig. A.12 la chute de variance des pixels à l'intérieur du visage pour une morphose.

A.5.2 Résultats

Nous avons créé une première base de données contenant 100 visages authentiques et 171 morphoses de visages à partir de la base de données PUT [147]. Nous avons ensuite évalué et comparé notre approche sur cette base de données. On peut voir

Table A.6 : **EER** sur la base de données PUT en variant la qualité JPEG des morphoses de visage et en appliquant le Contre-Forensique (CF)

EER at JPEG	100	95	90	85	80	75	CF
LBP	11.20	37.98	59.98	77.82	85.67	73.26	**22.86**
BSIF	26.86	5.86	**1.19**	5.86	11.21	**13.87**	41.63
LPQ	4.26	54.01	33.07	37.05	58.17	53.06	27.43
Méthode proposée	**0.59**	**0.37**	2.35	**4.73**	**8.48**	21.08	35.86

Table A.7 : **EER** sur la base de données FERET en variant la qualité JPEG des images authentiques

EER pour JPEG	Raw	100	95	90	85	80	75
LBP	3.93	3.42	4.87	8.07	11.71	16.19	18.42
BSIF	**1.63**	**1.75**	**2.23**	**3.87**	6.24	8.75	11.90
LPQ	2.64	2.78	2.72	4.80	5.82	8.06	8.86
Méthode proposée	7.59	7.79	7.24	6.59	**6.02**	**5.45**	**5.20**

les résultats en Table. A.6.

On voit que notre approche surpasse les méthodes de l'état de l'art pour des facteurs de qualité standard. Cependant, on peut voir qu'il est possible de développer une approche de contre-forensique permettant de tromper.

Dans la Table. A.7, nous avons cette fois fait varier la qualité JPEG des images authentiques et en prenant les morphoses de visage de meilleure qualité. Idéalement, on s'attend à ce que la détection soit aisée, quelle que soit la qualité des images authentiques puisque les morphoses sont de bonne qualité et donc simplement détectables. En réalité on observe que les performances des détecteurs de l'état de l'art chutent lorsque la qualité des images authentique diminue. Ceci montre que ces détecteurs se comportent comme des détecteurs de qualité d'images plutôt que des détecteurs de morphose. À l'inverse, les performances de notre détecteur augmentent puisque le résidu des images fortement compressées tend à être nul. Notre détecteur ne parvient donc plus à faire de différence entre le résidu à l'intérieur et à l'extérieur du visage. Cette faiblesse n'est pas gênante puisqu'elle est explicable est qu'elle est valable quelle que soit l'image étudiée (authentique ou falsifiée).

Ces résultats soulèvent des questionnements concernant l'utilisation pratique

des méthodes de détection à l'aveugle et sans référence dans certains scénarii. En effet, certaines méthodes visent à être utilisées dans des cas d'usage non contrôlé. L'image étudiée provient d'une source inconnue est pourrait ainsi avoir été imprimée et rescannée par exemple. Or nous avons montré les difficultés dont les détecteurs faisaient preuve en présence de variation de qualité des images authentiques ou falsifiées. De plus, nous avons montré qu'un algorithme de contre forensique conçu pour tromper notre méthode permettait aussi de tromper d'autres méthodes. Nous pensons que l'accumulation de ces défis ne permet pas la détection des morphoses de visage par des méthodes à l'aveugle et sans références dans des contextes non contrôlés. Pour ces applications, des méthodes basées sur une référence (c.-à-d. autre acquisition sûre de la personne) sont préférables. Les méthodes à l'aveugle et sans références restent cependant intéressantes dans un contexte contrôlé, mais une évaluation rigoureuse des performances face à une variation de qualité des images est nécessaire.

A.6 Détection de la double compression H.264

Dans un contexte de vérification d'identité à distance, l'acquisition du document et de la personne sous forme d'une vidéo se présente comme une solution naturelle. En effet, les personnes ou les éléments de sécurité standard tels que les hologrammes et encres variables sont par nature dynamiques et nécessitent de fait plus d'une image pour être pleinement caractérisés. Un autre argument en faveur de l'authentification via une vidéo plutôt qu'une image et la complexité ajoutées pour falsifier le document numériquement. Contrairement à la falsification d'une image, le contrefacteur fait en ici face à la contrainte du temps réel, mais aussi développé des méthodes plus évoluées. Pour falsifier les champs de textes d'un document sur une vidéo par copier-coller, l'algorithme devrait déjà être en mesure de détecter et suivre chaque champ du document. Une alternative serait l'usage de méthode de type «shape from template» afin de venir synthétiser l'intégralité du document. Quelle que soit la méthode choisie, on perçoit aisément le challenge pour le contrefacteur en comparaison d'une simple image. Ces raisons ont notamment privilégié le choix de la vidéo dans le cadre de la vérification d'identité pour

la certification de ces systèmes [14].

La complexité de telles falsifications ne devrait pas pour autant justifier une confiance aveugle dans les médias vidéos. En particulier dans le contexte de vérification qui nous intéresse. Une des attaques les plus pressenties dans le cadre d'un contrôle d'identité est la vérification de la biométrie de la personne. En particulier, de la biométrie faciale. Si la détection et le suivi en temps réel d'un document d'identité quelconque ne sont pas un sujet très étudié, il n'en va pas de même pour les visages. La détection et le suivi de ceux-ci sont des problématiques bien étudiées. Certaines technologies permettent même aujourd'hui la détection et l'approximation d'un maillage en 3D du visage en temps réel dans un navigateur sur un téléphone mobile [26, 27]. De même, exclure la possibilité pour le contrefacteur d'injecter un flux vidéo déjà falsifié au sein du système n'est pas raisonnable.

Il est donc nécessaire de justifier de l'intégrité du média vidéo en amont de la vérification de l'identité de la personne. De nombreuses recherches ont été menées concernant la détection du vivant. En pratique, les méthodes de détections du vivant combinées à l'utilisation de challenge aléatoire (clignement des yeux, sourire ...) permettent d'écarter raisonnable l'hypothèse d'un flux vidéo préparé en avance et injecté dans le système. Pour autant, ces systèmes ne suffisent pas à garantir l'intégrité du média final. Dans ce chapitre, on suppose donc que le flux est falsifié en temps réel. En particulier, on suppose que l'appareil d'acquisition est contrôlé et sûr. Dans ce cas d'usage, le contrefacteur intercepte le flux et le falsifie avant l'envoi sur le serveur. Si le flux est compressé, il est nécessaire pour le contrefacteur de le décompresser puis de le recompresser après sa falsification. Une première approche pour vérifier l'intégrité du flux est alors de détecter la double compression. On s'intéressera en particulier à la compression vidéo H.264 qui est, avec VP8, le seul codec imposé par la norme WebRTC [153].

Dans ce chapitre une approche de détection basée sur la distribution des coefficients DCT sera présentée.

A.6.1 Modélisation des coefficients DCT

Dans ce chapitre, nous considérons les coefficients $\mathbf{C}$ distribués selon une loi de Laplace

$$\mathbf{C} \sim \text{Laplace}(0, b). \tag{A.14}$$

En figure A.13 on peut constater que pour des vidéos simplement compressées, le paramètre b semble relativement stable d'une vidéo à l'autre.

La loi de Laplace est un choix populaire pour la modélisation des coefficients DCT. Cette modélisation n'est pas tout à fait exacte [180] mais à l'avantage d'être une bonne approximation avec une forme analytique plus simple.

La densité de probabilité des coefficients $\mathbf{C}$ sera donc donnée par :

$$f(x|b) = \frac{1}{2b} \exp(\frac{-|x|}{b}). \tag{A.15}$$

On suppose que b sera modifié par le processus de double compression. On propose donc deux tests d'hypothèse. Dans un premier temps, on suppose que la valeur théorique pour une simple compression b_0 ainsi que la valeur b_1 d'une double compression sont connues. Ensuite nous proposerons un test ou seule la valeur b_0 pour une simple compression est connue.

A.6.2 Test d'hypothèse simple

Pour vérifier si la vidéo est doublement compressée nous proposons donc le test suivant :

$$\begin{cases} \mathcal{H}_0 : \mathbf{C} \sim \text{Laplace}(0, b_0) \\ \mathcal{H}_1 : \mathbf{C} \sim \text{Laplace}(0, b_1), b_1 \neq b_0. \end{cases} \tag{A.16}$$

Le rapport de vraisemblance est défini comme :

$$\Lambda(\mathbf{C}) = \frac{\mathcal{L}_1(\mathbf{C})}{\mathcal{L}_0(\mathbf{C})}. \tag{A.17}$$

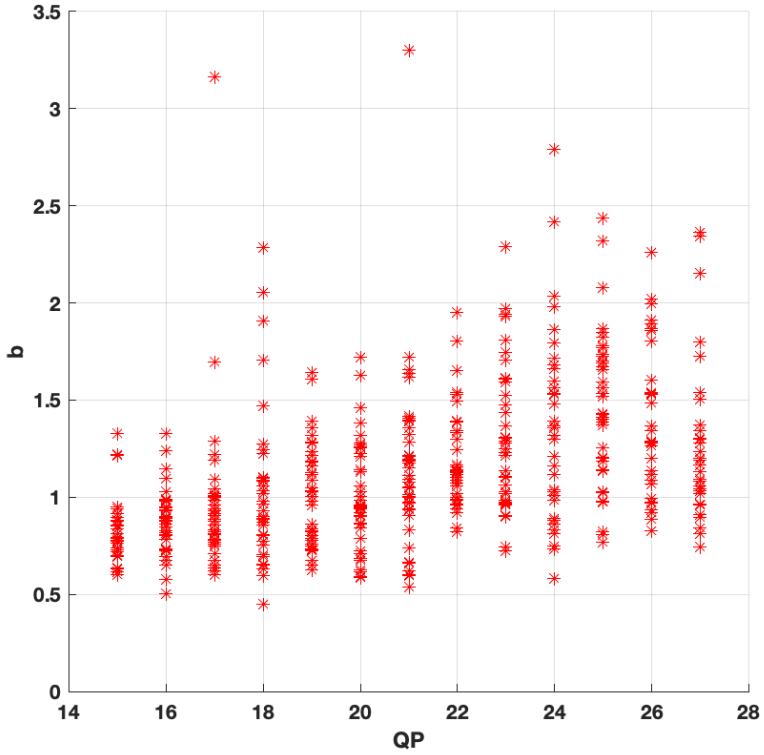


Figure A.13 : Distribution de b pour différente valeur de QP sur 40 vidéos.

On a donc le rapport de log-vraisemblance qui est donné par :

$$\Lambda(\mathbf{C}) = N \log \frac{b_0}{b_1} + \frac{b_1 - b_0}{b_0 b_1} \sum_{i=0}^{N} |c_i|. \tag{A.18}$$

On peut montrer que sous $\mathcal{H}_h, h \in \{0, 1\}$ on a :

$$\Lambda_h(\mathbf{C}) \sim \mathcal{N}(\mu_h, \sigma_h), \tag{A.19}$$

avec

$$\mu_h = N \log \frac{b_0}{b_1} + N b_h \frac{b_1 - b_0}{b_0 b_1}, \tag{A.20}$$

$$\sigma_h = |N \frac{b_1 - b_0}{b_0 b_1}| \frac{b_h}{\sqrt{N}}. \tag{A.21}$$

On pose finalement

$$\Lambda^{\star}(\mathbf{C}) = \frac{\Lambda_h(\mathbf{C}) - \mu_0}{\sigma_0} \sim \mathcal{N}(\frac{\mu_h - \mu_0}{\sigma_0}, \frac{\sigma_h}{\sigma_0}). \tag{A.22}$$

Sous $\mathcal{H}_0$ on a donc $\Lambda^{\star}(\mathbf{C})$ qui suit une loi normale centrée réduite.

En vertu du Lemme de Neyman-Pearson, le test le plus puissant est le test du rapport de vraisemblance. On définit donc le test :

$$\delta^{\star}(\mathbf{C}) = \begin{cases} \mathcal{H}_0 \text{ si } \Lambda^{\star}(\mathbf{C}) < \tau^{\star} \\ \mathcal{H}_1 \text{ si } \Lambda^{\star}(\mathbf{C}) \geq \tau^{\star} \end{cases}. \tag{A.23}$$

Pour ce test la puissance β est donnée par la probabilité α de rejeter l'hypothèse nulle sous $\mathcal{H}_1$:

$$\beta(\delta^{\star}) \triangleq \mathbb{P}_{\mathcal{H}_1}[\delta^{\star}(\mathbf{C}) = \mathcal{H}_1] = 1 - \alpha. \tag{A.24}$$

On peut définir un seuil $\tau^{\star}$ permettant d'obtenir un taux de fausses alarmes α_0 en résolvant :

$$\mathbb{P}_{\mathcal{H}_0}[\Lambda^{\star}(\mathbf{C}) \geq \tau^{\star}] = \alpha_0. \tag{A.25}$$

La puissance de notre test est finalement donnée par :

$$\beta(\delta^{\star}) = \mathbb{P}_{\mathcal{H}_1}[\Lambda^{\star}(\mathbf{C}) \geq \tau^{\star}]. \tag{A.26}$$

A.6.3 Test d'hypothèse composé

Pour le test composé, seul le paramètre b_0 est connu. Le paramètre b_1 est quant à lui remplacé par le maximum de vraisemblance estimé à partir des données. On a donc le rapport de vraisemblance donné par

$$\Lambda(\mathbf{C}) = \frac{N\overline{\mathbf{C}}}{b_0} - N\log(\overline{\mathbf{C}}) + N(\log(b_0) - b_0) \tag{A.27}$$

avec

$$\overline{\mathbf{C}} = \frac{1}{N}\sum_{i=0}^{N}|c_i|. \tag{A.28}$$

Table A.8 : Comparaison à l'état de l'art

Paramètre de quantification QP		Méthode		
QP_1	QP_2	Proposée	SVM-DCT [173]	G-VPF [165]
25	18	**1**	0.9781	0.9580
25	20	**1**	0.7503	0.9170
25	23	**0.9512**	0.6629	0.8926
23	18	**0.9990**	0.8241	0.9561
23	20	**0.9863**	0.7575	0.9111
20	18	**0.9570**	0.5078	0.9268
23	25	0.4453	0.4724	**0.8936**
25	30	0.3028	0.3632	**0.7432**

On peut montrer que :

$$\hat{\Lambda}(\mathbf{C}) = 2(\Lambda(\mathbf{C}) - a_0) \sim \chi^2(1) \tag{A.29}$$

avec

$$a_0 = N(1 - b_0). \tag{A.30}$$

A.6.4 Résultats

Nous avons constitué une base de données de vidéos simplement et doublement compressée afin d'évaluer notre approche et de la comparer à l'état de l'art. Les résultats sont donnés en Table. A.8.

On peut voir que notre approche surpasse les méthodes de l'état de l'art pour tout recomperssion ou $QP_1 > QP_2$. Cependant notre méthode n'est pas applicable dans le cas ou $QP_2 > QP_1$. En effet, une seconde compression plus forte que la compression initiale tend à effacer toutes traces de celle-ci. Ainsi, le paramètre b estimé correspond à la valeur attendue dans le cas d'une simple compression. Il est intéressant de noter que la méthode SVM-DCT est aussi inapplicable dans ce scénario. Ce résultat est attendu puisque la détection est basée sur les coefficients DCT. À l'inverse, la méthode G-VPF est plus robuste que les approches basées sur les coefficients DCT. On voit cependant que les performances chutent lorsque l'écart entre QP_2 et QP_1 grandit.

A.7 Conclusion

Dans cette thèse nous avons abordé les problématiques d'authentifications des images et vidéos. Ces médias font partie de notre vie de tous les jours et nous ne questionnons que rarement leurs authenticités. Cependant, les technologies de retouches photographiques et vidéos permettent aujourd'hui de produire des falsifications d'une très grande qualité.

Avec la pandémie liée au COVID19, l'utilisation des systèmes de vérification d'identité à distance s'est intensifiée. De tels systèmes reposent en grande partie sur des images et vidéos. Il est donc essentiel de pouvoir garantir l'intégrité de ces médias afin de sécuriser ces systèmes.

Nous avons abordé différentes attaques qui pourraient être menées par un contrefacteur tout en essayant de traiter des problématiques plus larges liées à ces attaques.

Dans un premier temps, nous avons constitué une base de données d'images falsifiées en tout genre afin de permettre d'étudier les approches par réseaux de neurones profonds.

Ensuite, nous avons étudié la problématique du copier-coller. Le copier-coller est une technique de falsification très simple et très efficace pour modifier les textes d'un document d'identité. Nous avons vu que ces méthodes sont très sensibles à la présence d'objet hautement similaire. En effet, les méthodes de l'état de l'art n'abordent que très peu cette problématique en se concentrant majoritairement sur la détection de copier-coller ayant subi de multiples transformations. Nous avons montré que ceci rendait leur utilisation impossible dans un contexte de détection de documents retouchés à cause des lettres fortement similaires. Nous avons donc proposé une première approche pour réduire le taux de faux positifs dans ce scénario ainsi qu'une nouvelle base de données de copier-coller sur des documents d'identité afin de favoriser la recherche dans cette direction.

Nous avons ensuite abordé la problématique de la suppression d'objet. Ce type de falsification est souvent utilisé en amont d'une autre attaque telle qu'une insertion ou un copier-coller. Nous avons montré que les contrefacteurs floutaient légèrement les bords de leur falsification afin d'obtenir un résultat visuel plus convaincant. Nous avons montré qu'il était possible de mettre en évidence cette

opération en étudiant la réflectance de l'image. En effet, cette opération tend à produire des textures plus lisses qui se traduisent par une chute localisée de la variance de la réflectance. Nous avons montré que ceci pouvait aussi mettre en évidence des falsifications de types insertions dans le cas ou la qualité de l'image insérée diffère largement de l'image cible.

Nous nous sommes ensuite intéressés à la problématique de la Morphose de visage qui peut permettre la création d'une biométrie partagée. Nous avons d'abord montré que la morphose de visage induit une chute de la variance à l'intérieur du visage retouché. Nous avons montré qu'il était possible de détecter ce type de falsification en étudiant la différence de niveau de variance à l'intérieur et à l'extérieur du visage. Nous avons aussi montré que les méthodes à l'aveugle et sans référence pouvaient être sensibles à la qualité des morphoses, mais aussi des images authentiques. Il est donc nécessaire de bien définir le cadre d'application pour de telles méthodes.

Enfin, nous avons étudié la double compression vidéo. En particulier nous nous sommes intéressés à l'algorithme H.264 qui est le plus répandu. En supposant que les coefficients sont distribués selon une loi de Laplace. Nous avons proposé deux tests d'hypothèses permettant la détection de la double compression. Dans un premier temps, nous avons proposé un test ou tous les paramètres sont connus. Puis nous avons proposé un test généralisé pour lequel l'un des paramètres est inconnu. Nous avons montré que ce test généralisé permet d'obtenir de meilleurs résultats de détection que les méthodes de l'état de l'art. Nous avons aussi montré que notre méthode n'est pas applicable dans le cas où la seconde compression est plus forte que la première.

APPENDIX B

Appendix

B.1 Maximum Likelihood Estimator for parameter b

We suppose that $\mathbf{C} \sim \text{Laplace}(0, \text{b})$. We can then define the likelihood function of a given parameter b as :

$$\mathcal{L}_b(c_i) = \frac{1}{2b} \exp(\frac{-|c_i|}{b}). \tag{B.1}$$

For $\mathbf{C}$ we then have

$$\begin{aligned} \mathcal{L}_b(\mathbf{C}) &= \prod_{i=0}^{N} \mathcal{L}_b(c_i) \\ &= \frac{1}{(2b)^N} \exp(\frac{-\sum_{i=0}^{N}|c_i|}{b}). \end{aligned} \tag{B.2}$$

The log-likelihood function for $\mathbf{C}$ is finally given by

$$\begin{aligned} \ell_b(\mathbf{C}) &= \log(\mathcal{L}_b(\mathbf{C})) \\ &= -N\log(2b) + \frac{-\sum_{i=0}^{N}|c_i|}{b}. \end{aligned} \tag{B.3}$$

The maximum likelihood estimate is thus give for

$$\frac{\partial \ell_b(\mathbf{C})}{\partial b} = 0, \tag{B.4}$$

With

$$\frac{\partial \ell_b(\mathbf{C})}{\partial b} = \frac{-N}{b} + \frac{\sum_{i=0}^{N}|c_i|}{b^2}. \tag{B.5}$$

We finally derive the maximum likelihood estimator $\hat{b}$ as

$$\hat{b} = \frac{\sum_{i=0}^{N}|c_i|}{N}. \tag{B.6}$$

Bibliographie

[1] G. Mahfoudi et al. “DEFACTO: Image and Face Manipulation Dataset”. In: *2019 27th European Signal Processing Conference (EUSIPCO)*. 2019, pp. 1–5.

[2] G. Mahfoudi et al. “Copy and Move Forgery Detection Using SIFT and Local Color Dissimilarity Maps”. In: *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2019, pp. 1–5.

[3] M. Pic, G. Mahfoudi, and A. Trabelsi. “Remote KYC: Attacks and Counter-Measures”. In: *2019 European Intelligence and Security Informatics Conference (EISIC)*. IEEE. 2019, pp. 126–129.

[4] G. Mahfoudi et al. “Object-Removal Forgery Detection through Reflectance Analysis”. In: *2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE. 2020, pp. 1–6.

[5] G. Mahfoudi et al. “CMID: A New Dataset for Copy-Move Forgeries on ID Documents”. In: *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2021, pp. 3028–3032.

[6] M. Pic et al. “Face Manipulation Detection in Remote Operational Systems”. In: *Handbook of Digital Face Manipulation and Detection*. Ed. by R. Christian et al. Springer International Publishing, 2022.

[7] G. Mahfoudi et al. “Statistical H.264 Double Compression Detection Method Based on DCT Coefficients”. In: *IEEE Access* 10 (2022), pp. 4271–4283. DOI: 10.1109/ACCESS.2022.3140588.

[8] G. Mahfoudi et al. *Method for processing a candidate image*. French Patent FR2000395. Jan. 2020.

[9] M. M. Pic et al. *Image processing method for an identity document*. French Patent FR19004228. Feb. 2019.

[10] G. Mahfoudi. *Procédé de vérification d'une image numérique*. French Patent FR2012868. Oct. 2020.

[11] M. M. Pic et al. *Procédé d'authentification d'un élément optiquement variable*. French Patent FR2006419. Feb. 2020.

[12] G. Mahfoudi, M. M. Pic, and A. Trabelsi. *Method for automatically detecting facial impersonation*. French Patent FR3088457, World Patent WO2020099400. May 2020.

[13] G. Mahfoudi and A. Ouddan. *Digital image processing method*. French Patent FR3091610B1. May 2020.

[14] ANSSI. *Publication du référentiel d'exigences applicables aux Prestataires de Vérification d'Identité à Distance (PVID)*. `https://www.ssi.gouv.fr/uploads/2021/08/anssi-requirements_rule_set-pvid-v1.1.pdf`. Accessed: 2020-09-27. Mar. 2021.

[15] G. S. Kc and P. A. Karger. *Security and Privacy Issues in Machine Readable Travel Documents (MRTDs)*. 2005.

[16] *Identification cards — Physical characteristics*. Standard. Geneva, Switzerland: International Organization for Standardization, Dec. 2019.

[17] *Machine Readable Travel Documents : Part 3 — Specifications Common to all MRTDs*. Specification. Geneva, Switzerland: International Civil Aviation Organization, 2021.

[18] S. Baechler. "Document Fraud: Will Your Identity Be Secure in the Twenty-First Century?" In: *European Journal on Criminal Policy and Research* 26.3 (Sept. 2020), pp. 379–398. ISSN: 0928-1371, 1572-9869. DOI: `10.1007/s10610-020-09441-8`. URL: `https://link.springer.com/10.1007/s10610-020-09441-8` (visited on 10/20/2021).

[19] L. Verdoliva. “Media Forensics and DeepFakes: An Overview”. In: *IEEE Journal of Selected Topics in Signal Processing* 14.5 (2020), pp. 910–932. DOI: 10.1109/JSTSP.2020.3002101.

[20] L. Zheng, Y. Zhang, and V. L. L. Thing. “A Survey on Image Tampering and Its Detection in Real-World Photos”. In: *Journal of Visual Communication and Image Representation* 58 (2019), pp. 380–399. ISSN: 1047-3203. DOI: 10.1016/j.jvcir.2018.12.022.

[21] N. A. Shelke and S. S. Kasana. “A Comprehensive Survey on Passive Techniques for Digital Video Forgery Detection”. In: *Multimedia Tools and Applications* 80.4 (Feb. 2021), pp. 6247–6310. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-020-09974-4. URL: https://link.springer.com/10.1007/s11042-020-09974-4 (visited on 10/20/2021).

[22] P. Joshi and N. Shrivastav. “A Review Paper on Image Inpainting and Their Different Techniques”. English. In: (2018), p. 4.

[23] E. M. Newton, L. Sweeney, and B. Malin. “Preserving Privacy by De-Identifying Face Images”. In: *IEEE transactions on Knowledge and Data Engineering* 17.2 (2005), pp. 232–243.

[24] R. Gross et al. “Integrating Utility into Face De-Identification”. In: *International Workshop on Privacy Enhancing Technologies*. Springer. 2005, pp. 227–242.

[25] O. Gafni, L. Wolf, and Y. Taigman. “Live Face De-Identification in Video”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9378–9387.

[26] Y. Kartynnik et al. “Real-time facial surface geometry from monocular video on mobile GPUs”. In: *arXiv preprint arXiv:1907.06724* (2019).

[27] Google. *Face Mesh - MediaPipe*. https://google.github.io/mediapipe/solutions/face_mesh.html. Accessed: 2020-09-27. 2020.

[28] J. Nakamura, ed. *Image Sensors and Signal Processing for Digital Still Cameras*. Boca Raton, FL: Taylor & Francis, 2006. 336 pp. ISBN: 978-0-8493-3545-7.

[29] C. Chen, S. Mccloskey, and J. Yu. "Analyzing Modern Camera Response Functions". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). Waikoloa Village, HI, USA: IEEE, Jan. 2019, pp. 1961–1969. ISBN: 978-1-72811-975-5. DOI: 10.1109/WACV.2019.00213. URL: https://ieeexplore.ieee.org/document/8658243/ (visited on 10/20/2021).

[30] B. Peng et al. "Optimized 3D Lighting Environment Estimation for Image Forgery Detection". In: *IEEE Transactions on Information Forensics and Security* 12.2 (Feb. 2017), pp. 479–494. ISSN: 1556-6021. DOI: 10.1109/TIFS.2016.2623589.

[31] B. Peng et al. "Improved 3D Lighting Environment Estimation for Image Forgery Detection". In: *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*. Nov. 2015, pp. 1–6. DOI: 10.1109/WIFS.2015.7368587.

[32] A. Mazumdar, J. Jacob, and P. K. Bora. "Forgery Detection in Digital Images through Lighting Environment Inconsistencies". In: *2018 Twenty Fourth National Conference on Communications (NCC)*. Feb. 2018, pp. 1–6. DOI: 10.1109/NCC.2018.8600175.

[33] A. Mazumdar and P. K. Bora. "Estimation of Lighting Environment for Exposing Image Splicing Forgeries". In: *Multimedia Tools and Applications* 78.14 (July 2019), pp. 19839–19860. ISSN: 1573-7721. DOI: 10.1007/s11042-018-7147-2.

[34] J.-B. Lee et al. "Detecting Digital Forgeries Using of a Lighting Direction". In: *2007 Digest of Technical Papers International Conference on Consumer Electronics*. Jan. 2007, pp. 1–2. DOI: 10.1109/ICCE.2007.341520.

[35] W. Fan et al. "3D Lighting-Based Image Forgery Detection Using Shape-from-Shading". In: *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. 2012, pp. 1777–1781.

[36] T. Carvalhoa, H. Faridb, and E. Keec. *Exposing Photo Manipulation From User-Guided 3-D Lighting Analysis*.

[37] X. Yang, Y. Li, and S. Lyu. "Exposing Deep Fakes Using Inconsistent Head Poses". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2019, pp. 8261–8265. DOI: 10.1109/ICASSP.2019.8683164.

[38] O. Mayer and M. C. Stamm. "Accurate and Efficient Image Forgery Detection Using Lateral Chromatic Aberration". In: *IEEE Transactions on Information Forensics and Security* 13.7 (July 2018), pp. 1762–1777. ISSN: 1556-6021. DOI: 10.1109/TIFS.2018.2799421.

[39] M. K. Johnson and H. Farid. "Exposing Digital Forgeries through Chromatic Aberration". In: *Proceeding of the 8th Workshop on Multimedia and Security - MM&Sec '06*. Geneva, Switzerland: ACM Press, 2006, p. 48. ISBN: 978-1-59593-493-2. DOI: 10.1145/1161366.1161376.

[40] Y.-F. Hsu and S.-F. Chang. "Image Splicing Detection Using Camera Response Function Consistency and Automatic Segmentation". In: *2007 IEEE International Conference on Multimedia and Expo*. July 2007, pp. 28–31. DOI: 10.1109/ICME.2007.4284578.

[41] Y.-f. Hsu and S.-f. Chang. "Detecting Image Splicing Using Geometry Invariants and Camera Characteristics Consistency". English. In: *2006 IEEE International Conference on Multimedia and Expo*. Toronto, ON, Canada: IEEE, July 2006, pp. 549–552. ISBN: 978-1-4244-0367-7. DOI: 10.1109/ICME.2006.262447.

[42] C. Chen, S. McCloskey, and J. Yu. "Image Splicing Detection via Camera Response Function Analysis". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 1876–1885. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.203.

[43] Y.-F. Hsu and S.-F. Chang. "Camera Response Functions for Image Forensics: An Automatic Algorithm for Splicing Detection". In: *IEEE Transactions on Information Forensics and Security* 5.4 (Dec. 2010), pp. 816–825. ISSN: 1556-6021. DOI: 10.1109/TIFS.2010.2077628.

[44] P. Ferrara et al. “Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts”. English. In: *IEEE Transactions on Information Forensics and Security* 7.5 (Oct. 2012), pp. 1566–1577. ISSN: 1556-6013, 1556-6021. DOI: 10.1109/TIFS.2012.2202227.

[45] C.-H. Choi, H.-Y. Lee, and H.-K. Lee. “Estimation of Color Modification in Digital Images by CFA Pattern Change”. In: *Forensic Science International* 226.1 (Mar. 2013), pp. 94–105. ISSN: 0379-0738. DOI: 10.1016/j.forsciint.2012.12.014.

[46] A. Singh, G. Singh, and K. Singh. “A Markov Based Image Forgery Detection Approach by Analyzing CFA Artifacts”. In: *Multimedia Tools and Applications* 77.21 (Nov. 2018), pp. 28949–28968. ISSN: 1573-7721. DOI: 10.1007/s11042-018-6075-5.

[47] A. Popescu and H. Farid. “Exposing Digital Forgeries in Color Filter Array Interpolated Images”. In: *IEEE Transactions on Signal Processing* 53.10 (Oct. 2005), pp. 3948–3959. ISSN: 1941-0476. DOI: 10.1109/TSP.2005.855406.

[48] N. Le and F. Retraint. “An Improved Algorithm for Digital Image Authentication and Forgery Localization Using Demosaicing Artifacts”. In: *IEEE Access* 7 (2019), pp. 125038–125053. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2938467.

[49] A. E. Dirik and N. Memon. “Image Tamper Detection Based on Demosaicing Artifacts”. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. Nov. 2009, pp. 1497–1500. DOI: 10.1109/ICIP.2009.5414611.

[50] F. Marra et al. “Blind PRNU-Based Image Clustering for Source Identification”. In: *IEEE Transactions on Information Forensics and Security* 12.9 (Sept. 2017), pp. 2197–2211. ISSN: 1556-6021. DOI: 10.1109/TIFS.2017.2701335.

[51] P. Korus and J. Huang. “Multi-Scale Analysis Strategies in PRNU-Based Tampering Localization”. In: *IEEE Transactions on Information Forensics*

and Security 12.4 (Apr. 2017), pp. 809–824. ISSN: 1556-6021. DOI: 10.1109/TIFS.2016.2636089.

[52] A. J. Cooper. "Improved Photo Response Non-Uniformity (PRNU) Based Source Camera Identification". In: *Forensic Science International* 226.1 (Mar. 2013), pp. 132–141. ISSN: 0379-0738. DOI: 10.1016/j.forsciint.2012.12.018.

[53] G. Chierchia et al. "Guided Filtering for PRNU-Based Localization of Small-Size Image Forgeries". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2014, pp. 6231–6235. DOI: 10.1109/ICASSP.2014.6854802.

[54] G. Chierchia et al. "A Bayesian-MRF Approach for PRNU-Based Image Forgery Detection". In: *IEEE Transactions on Information Forensics and Security* 9.4 (Apr. 2014), pp. 554–567. ISSN: 1556-6021. DOI: 10.1109/TIFS.2014.2302078.

[55] J. Lukas, J. Fridrich, and M. Goljan. "Digital Camera Identification from Sensor Pattern Noise". In: *IEEE Transactions on Information Forensics and Security* 1.2 (June 2006), pp. 205–214. ISSN: 1556-6021. DOI: 10.1109/TIFS.2006.873602.

[56] X. Pan, X. Zhang, and S. Lyu. "Exposing Image Forgery with Blind Noise Estimation". In: *Proceedings of the Thirteenth ACM Multimedia Workshop on Multimedia and Security - MM&Sec '11*. Buffalo, New York, USA: ACM Press, 2011, p. 15. ISBN: 978-1-4503-0806-9. DOI: 10.1145/2037252.2037256.

[57] B. Mahdian and S. Saic. "Using Noise Inconsistencies for Blind Image Forensics". English. In: *Image and Vision Computing* 27.10 (Sept. 2009), pp. 1497–1503. ISSN: 02628856. DOI: 10.1016/j.imavis.2009.02.001.

[58] T. Julliand, V. Nozick, and H. Talbot. "Automated Image Splicing Detection from Noise Estimation in Raw Images". In: *6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15)*. July 2015, pp. 1–6. DOI: 10.1049/ic.2015.0111.

[59] N. Le and F. Retraint. “Statistical Detector of Resampled TIFF Images”. In: *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. Dec. 2018, pp. 398–401. DOI: 10.1109/ISSPIT.2018.8642747.

[60] T. N. C. Doan, F. Retraint, and C. Zitzmann. “Image Tampering Detection Based on a Statistical Model”. In: *Multimedia Tools and Applications* (Aug. 2021). ISSN: 1573-7721. DOI: 10.1007/s11042-021-11213-3.

[61] B. Soni, P. K. Das, and D. M. Thounaojam. “CMFD: A Detailed Review of Block Based and Key Feature Based Techniques in Image Copy-Move Forgery Detection”. In: *IET Image Processing* 12.2 (2018), pp. 167–178.

[62] G. Liu et al. “Detect Image Splicing with Artificial Blurred Boundary”. In: *Mathematical and Computer Modelling* 57.11-12 (2013), pp. 2647–2659.

[63] J. Stanton, K. Hirakawa, and S. McCloskey. “Detecting Image Forgery Based on Color Phenomenology.” In: *CVPR Workshops*. 2019, pp. 138–145.

[64] P. Sun et al. “Exposing Splicing Forgery Based on Color Temperature Estimation”. In: *Forensic Science International* 289 (Aug. 2018), pp. 1–11. ISSN: 03790738. DOI: 10.1016/j.forsciint.2018.04.049. URL: https://linkinghub.elsevier.com/retrieve/pii/S037907381830207X (visited on 10/12/2021).

[65] S. Ye, Q. Sun, and E.-C. Chang. “Detecting Digital Image Forgeries by Measuring Inconsistencies of Blocking Artifact”. In: *2007 IEEE International Conference on Multimedia and Expo*. Ieee. 2007, pp. 12–15.

[66] T. Bianchi and A. Piva. “Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps”. In: *IEEE transactions on Information Forensics and Security* 7.2 (2011), pp. 842–848.

[67] C. Kumawat and V. Pankajakshan. “A JPEG Forensic Detector for Color Bitmap Images”. In: *IEEE Open Journal of Signal Processing* (2021).

[68] T.-T. Ng and S.-F. Chang. *A Data Set of Authentic and Spliced Image Blocks*. Tech. rep. Columbia University, June 2004.

[69] J. Dong, W. Wang, and T. Tan. “CASIA Image Tampering Detection Evaluation Database”. English. In: *2013 IEEE China Summit and International Conference on Signal and Information Processing*. Beijing, China: IEEE, July 2013, pp. 422–426. ISBN: 978-1-4799-1043-4. DOI: 10.1109/ChinaSIP.2013.6625374.

[70] I. Amerini et al. “A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery”. English. In: *IEEE Transactions on Information Forensics and Security* 6.3 (Sept. 2011), pp. 1099–1110. ISSN: 1556-6013, 1556-6021. DOI: 10.1109/TIFS.2011.2129512.

[71] V. Christlein et al. “An Evaluation of Popular Copy-Move Forgery Detection Approaches”. In: *IEEE Transactions on Information Forensics and Security* 7.6 (2012), pp. 1841–1854. DOI: 10.1109/TIFS.2012.2218597.

[72] D. Tralic et al. “CoMoFoD - New Database for Copy-Move Forgery Detection”. en. In: (), p. 7.

[73] B. Wen et al. “COVERAGE — A Novel Database for Copy-Move Forgery Detection”. In: *2016 IEEE International Conference on Image Processing (ICIP)*. Sept. 2016, pp. 161–165. DOI: 10.1109/ICIP.2016.7532339.

[74] D. Cozzolino, G. Poggi, and L. Verdoliva. “Copy-Move Forgery Detection Based on PatchMatch”. In: *2014 IEEE International Conference on Image Processing (ICIP)*. Oct. 2014, pp. 5312–5316. DOI: 10.1109/ICIP.2014.7026075.

[75] H. Guan et al. “MFC Datasets: Large-Scale Benchmark Datasets for Media Forensic Challenge Evaluation”. In: *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 2019, pp. 63–72.

[76] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris. “Detecting Image Splicing in the Wild (WEB)”. English. In: *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. Turin, Italy: IEEE, June 2015, pp. 1–6. ISBN: 978-1-4799-7079-7. DOI: 10.1109/ICMEW.2015.7169839.

[77] S. Heller, L. Rossetto, and H. Schuldt. “The PS-Battles Dataset - an Image Collection for Image Manipulation Detection”. English. In: *arXiv:1804.04866 [cs]* (Apr. 2018). arXiv: 1804.04866 [cs].

[78] T.-Y. Lin et al. “Microsoft COCO: Common Objects in Context”. English. In: *arXiv:1405.0312 [cs]* (May 2014). arXiv: 1405.0312 [cs].

[79] K. He et al. “A Global Sampling Method for Alpha Matting”. English. In: *CVPR 2011*. Colorado Springs, CO, USA: IEEE, June 2011, pp. 2049–2056. ISBN: 978-1-4577-0394-2. DOI: 10.1109/CVPR.2011.5995495.

[80] J. K. Chhabra and M. V. Birchha. “Detailed Survey on Exemplar Based Image Inpainting Techniques”. English. In: 5 (2014), p. 5.

[81] K. s Mahajan. “Image in Painting Techniques: A Survey”. English. In: *IOSR Journal of Computer Engineering* 5.4 (2012), pp. 45–49. ISSN: 22788727, 22780661. DOI: 10.9790/0661-0514549.

[82] N. Pandya and B. Limbasiya. “A Survey on Image Inpainting Techniques”. English. In: *International Journal of Current Engineering and Technology* (2013), p. 4.

[83] R. Suthar and M. K. R. Patel. “A Survey on Various Image Inpainting Techniques to Restore Image”. English. In: 4.2 (2014), p. 4.

[84] M. Daisy et al. “A Smarter Exemplar-Based Inpainting Algorithm Using Local and Global Heuristics for More Geometric Coherence”. English. In: *2014 IEEE International Conference on Image Processing (ICIP)*. Paris, France: IEEE, Oct. 2014. ISBN: 978-1-4799-5751-4. DOI: 10.1109/ICIP.2014.7025937.

[85] C. GraDhics. “Feature-Based Image Metamorphosis”. English. In: *Computer Graphics* (1992), p. 8.

[86] A. Makrushin and A. Wolf. “An Overview of Recent Advances in Assessing and Mitigating the Face Morphing Attack”. English. In: (2018), p. 5.

[87] M. Ferrara, A. Franco, and D. Maltoni. “The Magic Passport”. English. In: *IEEE International Joint Conference on Biometrics*. Clearwater, FL, USA: IEEE, Sept. 2014, pp. 1–7. ISBN: 978-1-4799-3584-0. DOI: 10.1109/BTAS.2014.6996240.

[88] D. E. King. “Dlib-Ml: A Machine Learning Toolkit”. In: *Journal of Machine Learning Research* 10 (2009), pp. 1755–1758.

[89] F. Bookstein. “Principal Warps: Thin-Plate Splines and the Decomposition of Deformations”. English. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.6 (June 1989), pp. 567–585. ISSN: 01628828. DOI: 10.1109/34.24792.

[90] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš. “Detection of Copy-Move Forgery in Digital Images”. In: *In Proceedings of Digital Forensic Research Workshop*. Citeseer. 2003.

[91] A. C. Popescu and H. Farid. “Exposing Digital Forgeries by Detecting Duplicated Image Regions”. In: (2004).

[92] B. Mahdian and S. Saic. “Detection of Copy–Move Forgery Using a Method Based on Blur Moment Invariants”. In: *Forensic Science International* 171.2 (2007), pp. 180–189. ISSN: 0379-0738. DOI: 10.1016/j.forsciint.2006.11.002.

[93] I. Amerini et al. “A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery”. In: *IEEE Transactions on Information Forensics and Security* 6.3 (2011), pp. 1099–1110.

[94] D. Cozzolino, G. Poggi, and L. Verdoliva. “Efficient Dense-Field Copy Move Forgery Detection”. In: *IEEE Transactions on Information Forensics and Security* 10.11 (Nov. 2015), pp. 2284–2297. ISSN: 1556-6013. DOI: 10.1109/TIFS.2015.2455334.

[95] E. Ardizzone, A. Bruno, and G. Mazzola. “Copy–Move Forgery Detection by Matching Triangles of Keypoints”. In: *IEEE Transactions on Information Forensics and Security* 10.10 (Oct. 2015), pp. 2084–2094. ISSN: 1556-6013. DOI: 10.1109/TIFS.2015.2445742.

[96] J. Li et al. “Segmentation-Based Image Copy-Move Forgery Detection Scheme”. In: *IEEE Transactions on Information Forensics and Security* 10.3 (Mar. 2015), pp. 507–518. ISSN: 1556-6013. DOI: 10.1109/TIFS.2014.2381872.

[97] S. Bravo-Solorio and A. K. Nandi. “Exposing Duplicated Regions Affected by Reflection, Rotation and Scaling”. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, pp. 1880–1883. DOI: 10.1109/ICASSP.2011.5946873.

[98] B. Soni, P. K. Das, and D. M. Thounaojam. “Keypoints Based Enhanced Multiple Copy-Move Forgeries Detection System Using Density-Based Spatial Clustering of Application with Noise Clustering Algorithm”. In: *IET Image Processing* 12.11 (2018), pp. 2092–2099. ISSN: 1751-9659. DOI: 10.1049/iet-ipr.2018.5576.

[99] Y. Li and J. Zhou. “Fast and Effective Image Copy-Move Forgery Detection via Hierarchical Feature Point Matching”. In: *IEEE Transactions on Information Forensics and Security* 14.5 (2019), pp. 1307–1322.

[100] A. Shahroudnejad and M. Rahmati. “Copy-Move Forgery Detection in Digital Images Using Affine-SIFT”. In: *2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS)*. Dec. 2016, pp. 1–5. DOI: 10.1109/ICSPIS.2016.7869896.

[101] Y. Fan, Y.-S. Zhu, and Z. Liu. “An Improved SIFT-Based Copy-Move Forgery Detection Method Using T-Linkage and Multi-Scale Analysis”. en. In: (2016), p. 10.

[102] T. Du, L. Tian, and C. Li. “Image Copy-Move Forgery Detection Based on SIFT-BRISK”. en. In: *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*. Hangzhou: IEEE, Oct. 2018, pp. 141–145. ISBN: 978-1-5386-6020-1. DOI: 10.1109/ICCAIS.2018.8570513.

[103] E. Baykal, B. Ustubioglu, and G. Ulutas. “Image Forgery Detection Based on SIFT and K-Means++”. In: *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*. June 2016, pp. 474–477. DOI: 10.1109/TSP.2016.7760924.

[104] C. S. Prakash et al. “Detection of Copy-Move Forgery Using AKAZE and SIFT Keypoint Extraction”. In: *Multimedia Tools and Applications* 78.16 (Aug. 2019), pp. 23535–23558. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-019-7629-x. URL: http://link.springer.com/10.1007/s11042-019-7629-x (visited on 10/12/2021).

[105] X. Bi, C.-M. Pun, and X.-C. Yuan. “Multi-Scale Feature Extraction and Adaptive Matching for Copy-Move Forgery Detection”. en. In: *Multimedia Tools and Applications* 77.1 (Jan. 2018), pp. 363–385. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-016-4276-3.

[106] E. Silva et al. “Going Deeper into Copy-Move Forgery Detection: Exploring Image Telltales via Multi-Scale Analysis and Voting Processes”. In: *Journal of Visual Communication and Image Representation* 29 (2015), pp. 16–32. ISSN: 1047-3203. DOI: 10.1016/j.jvcir.2015.01.016.

[107] A. Islam et al. “DOA-GAN: Dual-Order Attentive Generative Adversarial Network for Image Copy-Move Forgery Detection and Localization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 4676–4685.

[108] Y. Zhu et al. “AR-Net: Adaptive Attention and Residual Refinement Network for Copy-Move Forgery Detection”. In: *IEEE Transactions on Industrial Informatics* 16.10 (2020), pp. 6714–6723.

[109] R. Y. Wu, W. Abd-Almageed, and P. Natarajan. “BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization”. en. In: (2018), p. 17.

[110] E. Kafali et al. “RobusterNet: Improving Copy-Move Forgery Detection with Volterra-Based Convolutions”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. 2021, pp. 1160–1165. DOI: 10.1109/ICPR48806.2021.9412587.

[111] M. A. Elaskily et al. “A Novel Deep Learning Framework for Copy-Moveforgery Detection in Images”. In: *Multimedia Tools and Applications* 79.27-28 (July 2020), pp. 19167–19192. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-

020-08751-7. URL: http://link.springer.com/10.1007/s11042-020-08751-7 (visited on 10/12/2021).

[112] Y. Wu, W. Abd-Almageed, and P. Natarajan. "Image Copy-Move Forgery Detection via an End-to-End Deep Neural Network". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). Lake Tahoe, NV: IEEE, Mar. 2018, pp. 1907–1915. ISBN: 978-1-5386-4886-5. DOI: 10.1109/WACV.2018.00211. URL: https://ieeexplore.ieee.org/document/8354315/ (visited on 10/12/2021).

[113] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". en. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94.

[114] H. Bay et al. "Speeded-Up Robust Features (SURF)". In: *Computer Vision and Image Understanding* 110.3 (June 2008), pp. 346–359. ISSN: 10773142. DOI: 10.1016/j.cviu.2007.09.014. URL: https://linkinghub.elsevier.com/retrieve/pii/S1077314207001555 (visited on 10/12/2021).

[115] E. Rublee et al. "ORB: An Efficient Alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[116] B. A. Galler and M. J. Fisher. "An Improved Equivalence Algorithm". en. In: *Communications of the ACM* 7.5 (May 1964), pp. 301–303. ISSN: 00010782. DOI: 10.1145/364099.364331.

[117] É. Baudrier et al. "Binary-Image Comparison with Local Dissimilarity Quantification". In: *Pattern Recognition* 41.5 (2008), pp. 1461–1478. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2007.07.011.

[118] Y. Ech-Choudany et al. "Méthodes de Reconnaissance de Formes Basées Sur La Carte de Dissimilarité Locale Pour La Classification Des Images". In: *GRETSI* (2017).

[119] N. Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66.

[120] Y. Li and J. Zhou. "Image Copy-Move Forgery Detection Using Hierarchical Feature Point Matching". In: *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. 2016, pp. 1–4.

[121] C. Barnes et al. "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing". In: 28.3 (July 2009). ISSN: 0730-0301. DOI: 10.1145/1531326.1531330.

[122] A. Criminisi, P. Perez, and K. Toyama. "Object Removal by Exemplar-Based Inpainting". In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 2. 2003, pp. II–II.

[123] Z. Liang et al. "An Efficient Forgery Detection Algorithm for Object Removal by Exemplar-Based Image Inpainting". In: *Journal of Visual Communication and Image Representation* 30 (2015), pp. 75–85.

[124] D. T. Trung, A. Beghdadi, and M. Larabi. "Blind Inpainting Forgery Detection". In: *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2014, pp. 1019–1023.

[125] A. V. Oppenheim, R. W. Schafer, and T. G. Stockham. "Nonlinear Filtering of Multiplied and Convolved Signals". In: *Proceedings of the IEEE* 56.8 (1968), pp. 1264–1291.

[126] M. Ferrara, A. Franco, and D. Maltoni. "The Magic Passport". In: *IEEE International Joint Conference on Biometrics*. Sept. 2014, pp. 1–7. DOI: 10.1109/BTAS.2014.6996240.

[127] D. J. Robertson et al. "Detecting Morphed Passport Photos: A Training and Individual Differences Approach". In: *Cognitive research: principles and implications* 3.1 (2018), pp. 1–11.

[128] M. Ferrara, A. Franco, and D. Maltoni. "Decoupling Texture Blending and Shape Warping in Face Morphing". In: *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*. 2019, pp. 1–5.

[129] N. Damer et al. “Detecting Face Morphing Attacks by Analyzing the Directed Distances of Facial Landmarks Shifts”. In: *Pattern Recognition*. Ed. by T. Brox, A. Bruhn, and M. Fritz. Cham: Springer International Publishing, 2019, pp. 518–534. ISBN: 978-3-030-12939-2.

[130] S. Banerjee and A. Ross. “Conditional Identity Disentanglement for Differential Face Morph Detection”. In: *2021 IEEE International Joint Conference on Biometrics (IJCB)*. 2021, pp. 1–8. DOI: 10.1109/IJCB52358.2021.9484355.

[131] B. Chaudhary et al. “Differential Morph Face Detection Using Discriminative Wavelet Sub-Bands”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2021, pp. 1425–1434.

[132] S. Soleymani et al. “Mutual Information Maximization on Disentangled Representations for Differential Morph Detection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2021, pp. 1731–1741.

[133] U. Scherhag et al. “Deep Face Representations for Differential Morphing Attack Detection”. In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3625–3639. DOI: 10.1109/TIFS.2020.2994750.

[134] G. Borghi et al. “A Double Siamese Framework for Differential Morphing Attack Detection”. In: *Sensors* 21.10 (2021). ISSN: 1424-8220. DOI: 10.3390/s21103466. URL: https://www.mdpi.com/1424-8220/21/10/3466.

[135] R. Raghavendra et al. “Transferable Deep-CNN Features for Detecting Digital and Print-Scanned Morphed Face Images”. English. In: IEEE, July 2017, pp. 1822–1830. ISBN: 978-1-5386-0733-6. DOI: 10.1109/CVPRW.2017.228.

[136] R. Ramachandra et al. “Detecting Face Morphing Attacks with Collaborative Representation of Steerable Features”. In: *Proceedings of 3rd International Conference on Computer Vision and Image Processing*. Ed. by B. B. Chaudhuri et al. Vol. 1022. Advances in Intelligent Systems and Computing. Singapore: Springer Singapore, 2020, pp. 255–265. ISBN: 978-981-329-087-7 978-981-329-088-4. DOI: 10.1007/978-981-32-9088-4_22. URL:

http://link.springer.com/10.1007/978-981-32-9088-4_22 (visited on 10/14/2021).

[137] C. Seibold et al. *Accurate and Robust Neural Networks for Security Related Applications Exampled by Face Morphing Attacks*. 2018. arXiv: 1806.04265.

[138] C. Seibold, A. Hilsmann, and P. Eisert. "Reflection Analysis for Face Morphing Attack Detection". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. 2018, pp. 1022–1026. DOI: 10.23919/EUSIPCO.2018.8553116.

[139] L. Wandzik, G. Kaeding, and R. V. Garcia. "Morphing Detection Using a General- Purpose Face Recognition System". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. 2018, pp. 1012–1016. DOI: 10.23919/EUSIPCO.2018.8553375.

[140] R. Raghavendra, K. B. Raja, and C. Busch. "Detecting Morphed Face Images". English. In: *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. Niagara Falls, NY, USA: IEEE, Sept. 2016, pp. 1–7. ISBN: 978-1-4673-9733-9. DOI: 10.1109/BTAS.2016.7791169.

[141] L. Zhang, F. Peng, and M. Long. "Face Morphing Detection Using Fourier Spectrum of Sensor Pattern Noise". In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*. July 2018, pp. 1–6. DOI: 10.1109/ICME.2018.8486607.

[142] L. Debiasi et al. "PRNU Variance Analysis for Morphed Face Image Detection". In: *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. Oct. 2018, pp. 1–9. DOI: 10.1109/BTAS.2018.8698576.

[143] C. Kraetzer et al. "Modeling Attacks on Photo-ID Documents and Applying Media Forensics for the Detection of Facial Morphing". English. In: *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security - IHMMSec '17*. Philadelphia, Pennsylvania, USA: ACM Press, 2017, pp. 21–32. ISBN: 978-1-4503-5061-7. DOI: 10.1145/3082031.3083244.

[144] U. Scherhag, C. Rathgeb, and C. Busch. “Performance Variation of Morphed Face Image Detection Algorithms across Different Datasets”. In: *2018 International Workshop on Biometrics and Forensics (IWBF)*. IEEE. 2018, pp. 1–6.

[145] V. Kazemi and J. Sullivan. “One Millisecond Face Alignment with an Ensemble of Regression Trees”. English. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH: IEEE, June 2014, pp. 1867–1874. ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.241.

[146] D. E. King. “Dlib-Ml: A Machine Learning Toolkit”. English. In: (), p. 4.

[147] A. Kasinski, A. Florek, and A. Schmidt. “The PUT face database”. In: *Image Processing and Communications* 13.3-4 (2008), pp. 59–64.

[148] *Information Technology—Biometric Presentation Attack Detection— Part 3: Testing and Reporting*. Standard ISO/IEC 30107-3:2017. Geneva, Switzerland: ISO/IEC JTC1 SC37 Biometrics, International Organization for Standardization, 2017.

[149] T. Hai Thai, F. Retraint, and R. Cogranne. “Generalized Signal-Dependent Noise Model and Parameter Estimation for Natural Images”. English. In: *Signal Processing* 114 (Sept. 2015), pp. 164–170. ISSN: 01651684. DOI: 10.1016/j.sigpro.2015.02.020.

[150] T. Ojala, M. Pietikainen, and T. Maenpaa. “Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns”. In: *IEEE Transactions on pattern analysis and machine intelligence* 24.7 (2002), pp. 971–987.

[151] J. Kannala and E. Rahtu. “BSIF: Binarized Statistical Image Features”. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. Nov. 2012, pp. 1363–1366.

[152] V. Ojansivu and J. Heikkilä. “Blur Insensitive Texture Classification Using Local Phase Quantization”. In: *International Conference on Image and Signal Processing*. Springer. 2008, pp. 236–243.

[153] A. Roach. *WebRTC Video Processing and Codec Requirements*. RFC 7742. Mar. 2016. DOI: 10.17487/RFC7742. URL: https://rfc-editor.org/rfc/rfc7742.txt.

[154] W. Chen and Y. Q. Shi. "Detection of double MPEG compression based on first digit statistics". In: *International Workshop on Digital Watermarking*. Springer. 2008, pp. 16–30.

[155] T. Sun, W. Wang, and X. Jiang. "Exposing video forgeries by detecting MPEG double compression". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2012, pp. 1389–1392.

[156] W. Wang and H. Farid. "Exposing digital forgeries in video by detecting double MPEG compression". In: *Proceedings of the 8th workshop on Multimedia and security*. 2006, pp. 37–47.

[157] Y. Su and J. Xu. "Detection of double-compression in MPEG-2 videos". In: *2010 2nd International Workshop on Intelligent Systems and Applications*. IEEE. 2010, pp. 1–4.

[158] P. He et al. "Frame-wise detection of relocated I-frames in double compressed H. 264 videos based on convolutional neural network". In: *Journal of Visual Communication and Image Representation* 48 (2017), pp. 149–158.

[159] Q. Xu et al. "Relocated I-Frames Detection in H. 264 Double Compressed Videos Based on Genetic-CNN". In: *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE. 2018, pp. 710–716.

[160] S.-H. Nam et al. "Two-stream network for detecting double compression of H. 264 videos". In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 111–115.

[161] Q. Li et al. "One-Class Double Compression Detection of Advanced Videos Based on Simple Gaussian Distribution Model". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2021), pp. 1–1. DOI: 10.1109/TCSVT.2021.3069254.

[162] H. Yao et al. “Detection of double-compressed H. 264/AVC video incorporating the features of the string of data bits and skip macroblocks”. In: *Symmetry* 9.12 (2017), p. 313.

[163] H. Yao, R. Ni, and Y. Zhao. “Double compression detection for H. 264 videos with adaptive GOP structure”. In: *Multimedia Tools and Applications* 79.9 (2020), pp. 5789–5806.

[164] D. Vazquez-Padin et al. “Detection of video double encoding with GOP size estimation”. In: *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE. 2012, pp. 151–156.

[165] D. Vázquez-Padín et al. “Video integrity verification and GOP size estimation via generalized variation of prediction footprint”. In: *IEEE Transactions on Information Forensics and Security* 15 (2019), pp. 1815–1830.

[166] S Chen et al. “Detecting double H. 264 compression based on analyzing prediction residual distribution”. In: *International Workshop on Digital Watermarking*. Springer. 2016, pp. 61–74.

[167] J. Zheng et al. “Double H. 264 compression detection scheme based on prediction residual of background regions”. In: *International Conference on Intelligent Computing*. Springer. 2017, pp. 471–482.

[168] P. Bestagini et al. “Codec and GOP identification in double compressed videos”. In: *IEEE Transactions on Image Processing* 25.5 (2016), pp. 2298–2310.

[169] P. He et al. “Double compression detection based on local motion vector field analysis in static-background videos”. In: *Journal of Visual Communication and Image Representation* 35 (2016), pp. 55–66.

[170] Z. Zhang et al. “Detecting multiple H. 264/AVC compressions with the same quantisation parameters”. In: *IET Information Security* 11.3 (2016), pp. 152–158.

[171] X. Jiang et al. “Detection of double compression with the same coding parameters based on quality degradation mechanism analysis”. In: *IEEE Transactions on Information Forensics and Security* 13.1 (2017), pp. 170–185.

[172] S. Ting, S. Tanfeng, and J. Xinghao. “Detection of Double Compression with the Same Quantization Parameter Based on Dual Encoding Parameter Model”. In: *Journal of Shanghai Jiaotong University* 53.3 (2019), p. 334.

[173] D. Liao et al. “Double H. 264/AVC compression detection using quantized nonzero AC coefficients”. In: *Media Watermarking, Security, and Forensics III*. Vol. 7880. International Society for Optics and Photonics. 2011, 78800Q.

[174] X. Jiang et al. “Detection of double compression in MPEG-4 videos based on Markov statistics”. In: *IEEE Signal processing letters* 20.5 (2013), pp. 447–450.

[175] I. E. Richardson. *The H. 264 advanced video compression standard*. John Wiley & Sons, 2011.

[176] W. K. Pratt. “Digital image processing john wiley & sons”. In: *Inc., New York* (1991).

[177] R. Reininger and J. Gibson. “Distributions of the two-dimensional DCT coefficients for images”. In: *IEEE Transactions on Communications* 31.6 (1983), pp. 835–839.

[178] J. D. Eggerton and M. D. Srinath. “Statistical distributions of image DCT coefficients”. In: *Computers & electrical engineering* 12.3-4 (1986), pp. 137–145.

[179] T. Eude et al. “On the distribution of the DCT coefficients”. In: *Proceedings of ICASSP’94. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 1994, pp. V–365.

[180] T. H. Thai, R. Cogranne, and F. Retraint. “Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm”. In: *IEEE Transactions on Image Processing* 23.5 (2014), pp. 1980–1993.

[181] Y. Altunbasak and N. Kamaci. “An analysis of the DCT coefficient distribution with the H. 264 video coder”. In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 3. IEEE. 2004, pp. iii–177.

[182] T. Qiao et al. “Steganalysis of JSteg algorithm using hypothesis testing theory”. In: *EURASIP Journal on Information Security* 2015.1 (2015), pp. 1–16.

[183] D.-T. Dang-Nguyen et al. “Raise: A raw images dataset for digital image forensics”. In: *Proceedings of the 6th ACM multimedia systems conference*. 2015, pp. 219–224.

[184] ISO. *Information technology — Coding of audio-visual objects — Part 10: Advanced video coding*. ISO 14496-10:2020. Geneva, Switzerland: International Organization for Standardization, 2020.

[185] *Derf's collection*. `https://media.xiph.org/video/derf/`. Accessed: 2021-08-17.

Gaël MAHFOUDI

Doctorat : Optimisation et Sûreté des Systèmes

Année 2021

Authentification des images et vidéos numériques

Les médias digitaux font partie de notre vie de tous les jours. Après des années de photojournalisme, nous nous sommes habitués à considérer ces médias comme des témoignages objectifs de la réalité. Cependant les logiciels de retouches d'images et de vidéos deviennent de plus en plus puissants et de plus en plus simples à utiliser, ce qui permet aux contrefacteurs de produire des images falsifiées d'une grande qualité. L'authenticité de ces médias ne peut donc plus être prise pour acquise. Récemment, de nouvelles régulations visant à lutter contre le blanchiment d'argent ont vu le jour. Ces régulations imposent notamment aux institutions financières de vérifier l'identité de leurs clients. Cette vérification est souvent effectuée de manière distantielle au travers d'un Système de Vérification d'Identité à Distance (SVID). Les médias digitaux sont centraux dans de tels systèmes, il est donc essentiel de pouvoir vérifier leurs authenticités. Cette thèse se concentre sur l'authentification des images et vidéos au sein d'un SVID. Suite à la définition formelle d'un tel système, les attaques probables à l'encontre de ceux-ci ont été identifiées. Nous nous sommes efforcés de comprendre les enjeux de ces différentes menaces afin de proposer des solutions adaptées. Nos approches sont basées sur des méthodes de traitement de l'image ou sur des modèles paramétriques. Nous avons aussi proposé de nouvelles bases de données afin d'encourager la recherche sur certains défis spécifiques encore peu étudiés.

Mots clés : criminalistique – traitement d'images – test d'hypothèses (statistique) – détection de falsifications d'images – statistique mathématique.

Authentication of Digital Images and Videos

Digital media are parts of our day-to-day lives. With years of photojournalism, we have been used to consider them as an objective testimony of the truth. But images and video retouching software are becoming increasingly more powerful and easy to use and allow counterfeiters to produce highly realistic image forgery. Consequently, digital media authenticity should not be taken for granted any more. Recent Anti-Money Laundering (AML) relegation introduced the notion of Know Your Customer (KYC) which enforced financial institutions to verify their customer identity. Many institutions prefer to perform this verification remotely relying on a Remote Identity Verification (RIV) system. Such a system relies heavily on both digital images and videos. The authentication of those media is then essential. This thesis focuses on the authentication of images and videos in the context of a RIV system. After formally defining a RIV system, we studied the various attacks that a counterfeiter may perform against it. We attempt to understand the challenges of each of those threats to propose relevant solutions. Our approaches are based on both image processing methods and statistical tests. We also proposed new datasets to encourage research on challenges that are not yet well studied.

Keywords: forensic sciences – image processing – statistical hypothesis testing – image forgery detection – mathematical statistics.

Thèse réalisée en partenariat entre :

anrt association nationale recherche technologie — Cifre — SURYS — utt Université de Technologie Troyes

Ecole Doctorale "Sciences pour l'Ingénieur"