Chapter 1
Detecting manipulations in video

Grégoire Mercier, Foteini Markatopoulou, Roger Cozien, Markos Zampoglou, Evlampios Apostolidis, Alexandros I. Metsai, Symeon Papadopoulos, Vasileios Mezaris, Ioannis Patras, Ioannis Kompatsiaris

Grégoire Mercier, Dr, HdR
eXo maKina, Paris, France, e-mail: gregoire.mercier@exomakina.fr

Foteini Markatopoulou
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: markatopoulou@iti.gr

Roger Cozien, Dr, CTO
eXo maKina, Paris, France, e-mail: roger.cozien@exomakina.fr

Markos Zampoglou
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: markzampoglou@iti.gr

Evlampios Apostolidis
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece and School of Electronic Engineering and Computer Science, Queen Mary University, London, UK, e-mail: apostolid@iti.gr

Alexandros I. Metsai
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: alexmetsai@iti.gr

Symeon Papadopoulos
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: papadop@iti.gr

Vasileios Mezaris
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: bmezaris@iti.gr

Ioannis Patras
School of Electronic Engineering and Computer Science, Queen Mary University, London, UK, e-mail: i.patras@qmul.ac.uk

Ioannis Kompatsiaris
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: ikom@iti.gr

Abstract This chapter presents the techniques researched and developed within InVID for the forensic analysis of videos, and the detection and localization of forgeries within User-Generated Videos (UGVs). Following an overview of state-of-the-art video tampering detection techniques, we observed that the bulk of current research is mainly dedicated to frame-based tampering analysis or encoding-based inconsistency characterization. We built upon this existing research, by designing forensics filters aimed to highlight any traces left behind by video tampering, with a focus on identifying disruptions in the temporal aspects of a video. As for many other data analysis domains, deep neural networks show very promising results in tampering detection as well. Thus, following the development of a number of analysis filters aimed to help human users in highlighting inconsistencies in video content, we proceeded to develop a deep learning approach aimed to analyse the outputs of these forensics filters and automatically detect tampered videos. In this chapter we present our survey of the state of the art with respect to its relevance to the goals of InVID, the forensics filters we developed and their potential role in localizing video forgeries, as well as our deep learning approach for automatic tampering detection. We present experimental results on benchmark and real-world data, and analyse the results. We observe that the proposed method yields promising results compared to the state of the art, especially with respect to the algorithm's ability to generalise to unknown data taken from the real world. We conclude with the research directions that our work in InVID has opened for the future.

1.1 Introduction

Among the InVID requirements, a prominent one has been to provide state-of-the-art technologies to support video forensic analysis, and in particular manipulation detection and localization. Video manipulation detection refers to the task of using video analysis algorithms to detect whether a video has been tampered with video processing software, and if yes, to provide further information on the tampering process (e.g. where in the video the tampering is located and what sort of tampering took place).

InVID deals with online content, primarily User-Generated Content (UGC). The typical case concerns videos captured with hand-held devices (e.g. smartphones) by amateurs, although it is not uncommon to include semi-professional or professional content. These videos are presented as real content captured on the scene of a newsworthy event, and usually do not contain any shot transitions but instead consist of a single shot. This is an important aspect of the problem, as a video that contains multiple shots has by definition already been edited, which may lessen its value as original eyewitness material. The videos are typically uploaded on social media sharing platforms (e.g. Facebook, YouTube), which means that they are typically in H.264 format, and often suffer from low resolution, and relatively strong quantization.

When considering the task, we should keep in mind that image modifications are not always malicious. Of course such cases are possible, such as the insertion or removal of key people or objects, which may alter the meaning of a video, and these are the cases that InVID video forensics was mostly aimed at. However, there are many more types of tampering that can take place on a video, which can be considered innocuous. These may include for example whole-video operations such as sharpening or color adjustments for aesthetic reasons, or the addition of logos and watermarks on the videos. Of course, contextually such post-processing steps do partly diminish the originality and usefulness of a video, but in the case that such videos are the only available evidence on a breaking event, they become important for news organisations.

The detection of manipulations in video is a challenging task. The underlying rationale is that a tampering operation leaves a trace on the video -usually invisible to the eye and pertaining to some property of the underlying noise or compression patterns of the video- and that trace may be detectable with an appropriate algorithm. However, there are multiple complications in this approach. Overall, there are many different types of manipulation that can take place (object removal, object copy-paste from the same scene or from another video, insertion of synthetic content, frame insertion or removal, frame filtering or global color/illumination changes, etc.), each potentially leaving different sorts of traces on the video. Furthermore, we are dealing with the fact that video compression consists of a number of different processes, all of which may disrupt the tampering traces. Finally, especially in the case of online UGVs, these are typically published on social networks, which means that they have been repeatedly re-encoded and are often of low quality, either due to the resulting resolution or due to multiple compression steps. So, in order to succeed, detection strategies may often need to be able to detect very weak and fragmented traces of manipulation. Finally, an issue that further complicates the task is non-malicious editing. As mentioned above, occasionally videos are published with additional logos or watermarks. While these do not constitute manipulation or tampering, they are the result of an editing process identical to that of tampering and thus may trigger a detection algorithm, or cover up the traces of other, malicious modifications.

With these challenges in mind, we set out to implement the InVID video forensics component, aiming to contribute a system that could assist professionals in identifying tampered videos, or to advance the state of the art towards this direction. We began by exploring the state of the art in image forensics, based on the previous expertise of some of InVID partners (CERTH-ITI, eXo maKina) in this area. We then extended our research into video forensics, and finally proceeded to develop the InVID video forensics component. This consists of a number of algorithms, also referred to as *Filters*, aimed to process the video and help human users localise suspect inconsistencies. These filters are integrated in the InVID Verification Application and their outputs made visible to the users, to help them visually verify the videos. Finally, we tried to automate the detection process by training a deep neural network architecture to spot these inconsistencies and classify videos as authentic or tampered.

This chapter focuses on video tampering detection and does not deal with other forms of verification, e.g. semantically analyzing the video content, or considering metadata or contextual information. It is dedicated to the means that are adopted to track weak traces (or *signatures*) left by the tampering process in the encoded video content. It accounts for encoding integrity, space, time, color and quantization coherence. Two complementary approaches are presented, one dealing with tampering localization, i.e. using filters to produce output maps aimed to highlight where the image may have been tampered, and designed to be interpreted by a human user, and one dealing with tampering detection, aiming to produce a single-value output per video indicating the probability that the video is tampered.

The rest of the chapter is organised as follows. Section 1.3 briefly presents the necessary background, Section 1.3 presents an overview of the most relevant approaches that can be found in the literature. Section 1.4 details the methodologies developed in InVID for detecting tampering in videos. Specifically, subsection 1.4.1 presents the filters developed for video tampering localization, while subsection 1.4.2 presents our approach for automatic video tampering detection. Section 1.5 then presents and analyses the evaluation results from the automatic approach over a number of experimental datasets. Finally, section 1.6 presents our conclusions from our work in video forensics during the InVID project.

1.2 Background

Image and Video forensics are essentially sub-fields of image and video processing, and thus certain concepts from these fields are particularly important to the tasks at hand. In this section we will briefly go over the most relevant of these concepts, as necessary background for the rest of the chapter.

While an image (or video frame) can in our case be treated as a 2D array of (R,G,B) values, the actual color content of the image is often irrelevant for forensics. Instead, we are often interested in other less prominent features, such as the noise, luminance-normalised color, or acuity of the image.

The term **image noise** refers to the random variation of brightness or color information, and is generally a combination of the physical characteristics of the capturing device (e.g. lens imperfections) and the image compression (in the case of lossy compression which is the norm). One way to isolate the image noise is to subtract a low-pass filtered version of the image from itself. The residue of this operation tends to be dominated by image noise. In cases where we deal with the luminance rather than the color information of the image, we call the output **luminance noise**. Another high-frequency aspect of the image is the **acuity** or **sharpness**, which is is a combination of focus, visibility, and image quality, and can be isolated using high-pass filtering.

With respect to video, certain aspects of MPEG compression are important for forensics and will be presented in short here. MPEG compression in its variants (MPEG-1, MPEG-2, MPEG-4 Part 2, and MPEG-4 part 10, also known as AVC

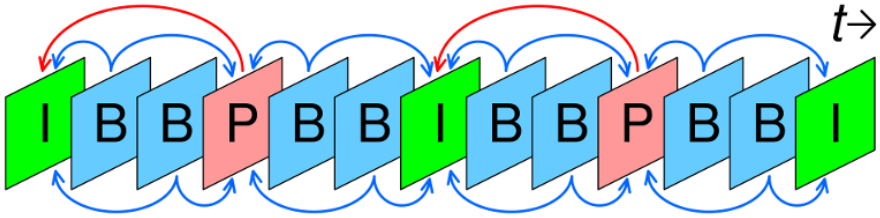


Fig. 1.1: Two example GOPs with I, P, and B frames. The GOP size in this case is 6 for both GOPs.

or H.264) is essentially based on the difference between frames that are encoded using only information contained within them, also known as intra-frame compression, and frames that are encoded using information from other frames in the video, known as inter-frame compression. Intra-frame compression is essentially image compression, and in most cases is based on algorithms that resemble JPEG encoding. The concept of inter-frame encoding is more complicated. Given other frames in the sequence, the compression algorithm performs block-matching between these frames and the frame to be encoded. The vectors linking these blocks are known as **motion vectors** and, besides providing a way to reconstruct a frame using similar parts from other frames, can also provide a rough estimate of the motion patterns in the video, by studying the displacements of objects through time. The reconstruction of a frame is done by combining the motion-compensated blocks from the reference frames, with a residue image which is added to it to create the final frame.

Frames in MPEG-encoded videos are labelled **I, P, or B frames**, depending on their encoding. I signifies intra-frame encoding, P signifies inter-frame encoding using only data from previous frames, while B signifies bi-directional inter-frame encoding using data from both previous and future frames. Within a video, these are organised in **Groups of Pictures (GOPs)**, starting with an I-frame and containing P- and B- frames (Fig. 1.1). The distance between two I-frames is the GOP length, which is fixed in earlier encodings but can vary in the modern formats. Similarly, modern formats allow much more flexibility in other aspects of the encoding, such as the block size and shape, which means that algorithms with strict assumptions on the workings of the algorithm (e.g. expecting a fixed GOP size) will not work on modern formats.

1.3 Related Work

1.3.1 Image Forensics

Multimedia forensics is a field with a long research history, and much progress has been achieved in the last decades. However, most of this progress concerned

the analysis of images rather than videos. Image forensics methods are typically organised in one of two categories: *active forensics*, where a watermark or similar (normally invisible) piece of information is embedded in the image at the time of capture, of which the integrity ensures that the image has not been modified since capture [38, 39, 43], and *passive forensics*, where no such prior information exists, and the analysis of whether an image has been tampered depends entirely on the image content itself. While the latter is a much tougher task, it is also the most relevant in the majority of use cases, where we typically have no access to any information about the image capturing process.

One important distinction in image forensics algorithms is between tampering *detection* and tampering *localization*. In the former case, the algorithm only reports knowledge on whether the image has been tampered or not, and typically returns a scalar likelihood estimate. In the latter case, the algorithm attempts to inform the user *where* the tampering has taken place, and returns a map corresponding to the shape of the image and highlighting the regions of the image that are likely to have been tampered -ideally, a per-block or per-pixel probability estimate.

Passive image forensics approaches can be categorised with respect to the type of modification they intend to detect and/or localise. Three main groups of modifications are copy-moving, splicing or in-painting, and whole-image operations. In the first case, a part of the image is replicated and placed elsewhere in it -for example, the background is copied to remove an object or person, or a crowd is duplicated to appear larger. Copy-move detection algorithms attempt to capture the forgery by looking for self-similarities within the image [56, 46]. In the case of splicing, a part of one image is placed within another. Splicing detection and localization algorithms are based on the premise that, on some possibly invisible level, the spliced area will differ from the rest of the image due to their different capturing and compression histories. The case with in-painting, i.e. when part of the image is erased and then automatically filled using an in-painting algorithm is in principle similar, since the computer-generated part will carry a different profile than the rest of the image. Algorithms designed to detect such forgeries may exploit inconsistencies in the local JPEG compression history [18, 25], in local noise patterns [29, 11], or in the traces left by the capturing devices' Color Filter Array (CFA) [15, 19]. It is interesting to note that, in many cases, such algorithms are also able to detect copy-move forgeries, as they also often cause detectable local disruptions. For cases where localization is not necessary, tampering detection algorithms combining filtering and machine learning have been proposed in the past, reaching very high accuracy within some datasets [10, 32]. Finally, whole-image operations such as rescaling, recompression, or filtering cannot be localised and thus are generally tackled with tampering detection algorithms [65, 6, 52].

Recently, with the advent of deep learning, new approaches began to appear, attempting to leverage the power of convolutional neural networks for tampering localization and detection. One approach is to apply a filtering step on the image, and then use a Convolutional Neural Network to analyse the filter output [7]. Other methods have attempted to incorporate the filtering step into the network, through the introduction of a Constrained Convolutional Layer, of which the parameters are

normalised at each iteration of the training process. This ensures that the first layer always operates as a high-pass filter, but is still trained alongside the rest of the network. Networks having this layer as their first convolutional layer were proposed for tampering detection [4] and resampling detection [5] with promising results, while a multi-scale approach was proposed in [28]. Recently, an integrated model was proposed, re-implementing an approach similar to [20], but exclusively using deep learning architectures [12].

A major consideration with image forensics, and especially in the use cases tackled through InVID, where we deal with online content from Web and social media sources, is the degradation of the tampering traces as the content circulates from platform to platform. The traces that most algorithms look for are particularly fragile, and are easily erased through resampling or recompression. Since most online platforms perform such operations on all images uploaded to them, this is a very important consideration for news-related multimedia forensics, and a recent study attempted to evaluate the performance of splicing localization algorithms in such environments [64].

1.3.2 Video Forensics

With respect to video-related disinformation, the types of tampering that we may encounter are, to an extent, similar to the ones encountered in images. Thus, we may encounter copy-moving, splicing, in-painting, or whole-video operations such as filtering or illumination changes. An important difference is that such operations may have a temporal aspect, e.g. splicing is typically the insertion of a second video consisting of multiple green-screen frames depicting the new object in motion. Similarly, a copy-move may be temporally displaced, i.e. an object of a video from some frames reappearing in other frames, or spatially displaced, i.e. an object from a frame reappearing elsewhere on the same frame. Furthermore, there exists a type of forgery that is only possible in videos, namely inter-frame forgery, which essentially consists of frame insertion or deletion.

Inter-frame forgery is a special type of video tampering, because it is visually identifiable in most cases as an abrupt cut or shot change in the video. There exist two types of videos where such a forgery may actually succeed to deceive viewers: One is the case of a video that already contains cuts, i.e. edited footage. There, a shot could be erased or added among the existing shots, if the audio track can be correspondingly edited. The other is the case of CCTV video or other video footage taken from a static camera. There, frames could be inserted, deleted, or replaced without being visually noticeable. However, the majority of InVID use cases concern UGV, which is usually taken by hand-held capturing devices and consists of unedited single shots. In those cases, inter-frame forgeries cannot be applied without being immediately noticeable. Thus, inter-frame forgery detection was not a high priority for InVID.

When first approaching video forensics, one could conceptualise the challenge as an extension of image forensics, which could be tackled with similar solutions. For example, video splicing could be detected based on the assumption that the inserted part carries a different capturing and compression history than the video receiving it. However, our preliminary experimentation showed that the algorithms designed for images do not work on videos, and this even applies to the most generic noise-based algorithms. It goes without saying that algorithms based specifically on the JPEG image format are even more inadequate to detect or localise video forgeries. The main reason for this is that a video is much more than a sequence of images. MPEG compression -which is the dominant video format today- encodes information by exploiting temporal interrelations between frames, essentially reconstructing most frames by combining blocks from other frames with a residual image. This process essentially destroys the traces that image-based algorithms aim to detect. Furthermore, the requantization and recompression performed by online platforms such as YouTube, Facebook, and Twitter is much more disruptive for the fragile traces of tampering than the corresponding recompression algorithms for images. Thus, video tampering detection requires the development of targeted, video-based algorithms. Even more so, algorithms designed for MPEG-2 will often fail when encountered with MPEG-4/H.264 videos [45], which are the dominant format for online videos nowadays. Thus, when reviewing the state of the art, we should always evaluate the potential robustness of the algorithm with respect to online videos.

When surveying the state of the art, a similar taxonomy that we used for image forensics can be used for videos-based algorithms. Thus, we can find a large number of active forensics approaches [44, 67, 17, 51, 47], which however are not applicable in most InVID use cases, where we have no control of the video capturing process. As mentioned above, passive video forensics can be organised in a similar structure as passive image forensics, with respect to the type of forgery they aim to detect: splicing/object insertion, copy-moving/cloning, whole-video operations, and inter-frame insertion/deletion. The following subsections present an overview of these areas, while two comprehensive surveys can be found in [37, 45].

1.3.2.1 Video splicing and in-painting

Video splicing refers to the insertion of a part of an image or video in some frames of a recipient video. Video in-painting refers to the replacement of a part of the video frames with automatically generated content, presumably to erase the objects depicted in those parts of the frame. In principle, video splicing detection algorithms operate similarly to image splicing detection algorithms, i.e. by trying to identify local inconsistencies in some aspect of the image, such as the noise patterns or compression coefficients.

Other strategies focus on temporal noise [33] or correlation behavior [27]. It is not clear if those methods could process video encoded in a constant bit rate strategy, since imposing a constant bit rate compression induces a variable quantization level over time, depending on the video content. Nevertheless, the noise estimation

induces a predictable feature shape or background, which imposes an implicit hypothesis such as a limited global motion (the fact is that those methods work better with still background). The Motion Compensated Edge Artifact is an interesting alternative to deal with temporal behavior of residuals between I, P and B frames without requiring strong hypotheses on the motion or background contents. Those periodic artifacts in the DCT coefficients may be extracted through a thresholding technique [48] or spectral analysis [16].

1.3.3 Detection of Double/Multiple Quantization

When we detect that a video has been requantised more than once, it does not mean that the video was tampered between the two compressions. In fact, it may well be possible that the video was simply re-scaled, or changed format, or was simply uploaded on a social media platform which re-encoded the video. Thus, detection of double/multiple quantization does not give tampering information as such, but gives a good indication that the video has been reprocessed and may have been edited. Of course, as InVID primarily deals with social media content, all analysed videos will have been quantised twice. Thus, from our perspective it is more important to know if the video has been quantised more than two times, and if yes, to know the exact number of quantizations it has undergone.

Video multiple quantization detection is often based on the quantization analysis of I frames. This is similar to techniques used for recompression analysis of JPEG images, although, as explained above, it should be kept in mind that an I frame can not always be treated as a JPEG image, as its compression is often much more complex, and a JPEG-based algorithm may be inadequate to address the problem.

In JPEG compression, the distribution of DCT coefficients before quantization follows the Generalised Gaussian distribution, thus its quantised representation is given by Benford's law and its generalised version [21]. The degree to which the DCT coefficient distribution conforms with Benford's law may be used as an indication on whether the image has been requantised or not. In a more video-specific approach, the temporal behavior of the parameters extracted from Benford's law may also be exploited to detect multi-compression of the video [31, 59].

Other approaches propose to detect multiple quantization of a video stream by considering the link between the quantization level and the motion estimation error, especially on the first P frame following a (requantised) I frame [54, 49]. However, such approaches are designed to work with fixed-size GOPs, which is more relevant for MPEG-2 or the simpler Part 2 of MPEG-4, rather than the more complex modern formats such as H.264/AVC/MPEG-4 Part 10.

1.3.4 Inter-frame Forgery Detection

This kind of tampering is characterised by the insertion (or removal) of entire frames in the video stream. Such cases arise for instance in video surveillance systems where, due to the static background, frames can be inserted, deleted, or replaced without being detectable, with malicious intent. Many approaches are based on the detection of inconsistencies in the motion prediction error along frames, the mean displacement over time, the evolution of the percentage of intra-coded macro blocks, or the evolution of temporal correlation of spatial features such as Local Binary Patterns (LBP) or velocity fields [42, 66, 22, 57].

However, inter-frame forgery is generally not very common in UGVs, as we have found through the InVID use cases. The Fake Video Corpus [34], a dataset of fake and real UGC videos collected during InVID shows that, on the one hand, most UGV content presented as original is typically unedited and single-shot, which means that it is hard to insert frames without them being visually detectable. On the other hand, multi-shot video by nature includes frame insertions and extractions, without this constituting some form of forgery. Thus, for InVID such methods are not particularly relevant.

1.3.5 Video Deep Fakes and their Detection

Recently, the introduction of deep learning approaches has disrupted many fields including image and video classification and synthesis. Of particular relevance has been the application of such approaches for the automatic synthesis of highly realistic videos with impressive results. Among them, a popular task with direct implications on the aims of InVID is face swapping, where networks are trained to replace human faces in videos with increasingly more convincing results [8, 2]. Other tasks include image-to-image translation [3, 26], where the model learns to convert images from one domain to another (e.g. take daytime images and convert them to look as if they were captured at night), and image in-painting [55, 62], where a region of the image is filled by automatically generated content, presumably with erasing objects and replacing them with background.

Those approaches are bringing new challenges in the field of video forensics, since in most of these cases the tampered frames are synthesised from scratch by the network. As a consequence, in these cases it is most likely that content inconsistencies are no longer relevant with respect to tampering detection. Thus, all strategies based on the statistical analysis of video parameters (such as quantization parameters, motion vectors, heteroscedasticity, etc.) may have been rendered obsolete. Instead, new tampering detection strategies need to account for scene, color and shape consistencies, or to look for possible artifacts induced by forgery methods. Indeed, detecting deep fakes may be a problem more closely linked to the detection of computer generated images (a variant of which is the detection of computer graphics and 3D rendered scenes) [13, 14, 53] than to tampering detection. Face

swaps are an exception to this, as in most cases the face is inserted on an existing video frame, thus the established video splicing scenario still holds. Recently, a study on face swap detection was published, testing a number of detection approaches against face swaps produced by three different algorithms, including one based on deep learning [41]. This work, which is an extension of a previous work on face swap detection [40], shows that in many cases image splicing localization algorithms such as XceptionNet [9] and MesoNet [1] can work, at least for the raw images and videos having undergone one compression step. During the course of the InVID project, the discourse on the potential of deep fakes to disrupt the news cycle and add to the amount of online disinformation has risen from practically nonexistent in 2016 to central in 2018. The timing and scope of the project did not allow to devote resources to tackling the challenge. Instead, the InVID forensics component was dedicated to analyzing forgeries committed using more traditional means. However, as the technological capabilities of generative networks increase, and their outputs become more and more convincing, it is clear that any future ventures into video forensics would have to take this task very seriously as well.

1.4 Methodology

1.4.1 Video Tampering Localization

The first set of video forensics technologies developed within InVID concerned a number of forensics filters aimed to be interpreted by trained human investigators in order to spot inconsistencies and artifacts, which may highlight the presence of tampering. In this work, we followed the practice of a number of image tampering localization approaches [29, 61] that do not return a binary map or a bounding box giving a specific answer to the question, but rather a map of values that need to be visualised and interpreted by the user in order to decide if there is tampering, and where.

With this in mind, eXo maKina developed a set of novel filters aimed at generating such output maps by exploiting the parameters of the MPEG-4 compression, as well as the optical and mathematical properties of the video pixel values. The outputs of these filters are essentially videos themselves, with the same duration as the input videos, allowing temporal and spatial localization of any highlighted inconsistencies in the video content. In line with their image forensics counterparts, these filters do not include direct decision making on whether the video is tampered or not, but instead highlight various aspects of the video stream that an investigator can visually analyse for inconsistencies. However, since these features can be used by analysts to visually reach a conclusion, we deduce that it is possible for a system to be trained to automatically process these traces and come to a similar conclusion without human help. This reduces the need for training investigators to analyse videos. Therefore, in parallel to developing filters for human inspection, we

also investigated machine learning processes that may contribute to decision making based on the outputs of those filters. Subsection 1.4 presents these filters and the type of tampering artifacts they aim to detect, while subsection 1.4.2 presents our efforts to develop an automatic system that can interpret the filter outputs in order to assist investigators.

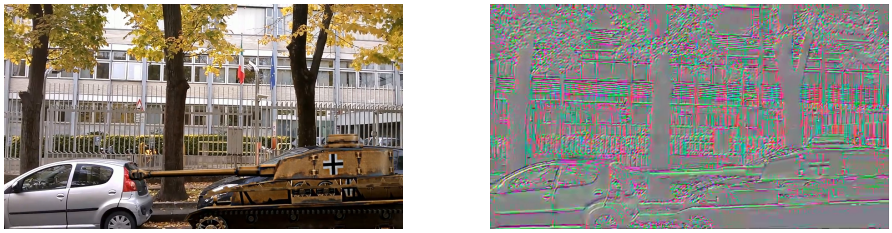
The filters developed by eXo maKina for the video forensics component of InVID are organised in three broad groups: Algebraic, optical, and temporal filters.

1. Algebraic filters: The term *algebraic filters* refers to any algebraic approaches that allow projecting information into a sparse feature space that makes forensic interpretation more easy.

   - The **Q4** filter is used to analyse the decomposition of the image through the Discrete Cosine Transform. The 2D DCT converts an $N \times N$ block of an image into a new $N \times N$ block in which the coefficients are calculated based on their frequency. Specifically within each block, the first coefficient situated at position (0,0) represents the lowest frequency information and its value is therefore related to the average value of the entire block, the coefficient (0,1) next to it characterises a slow evolution from dark to light in the horizontal direction, etc.
     If we transform all $N \times N$ blocks of an image with the DCT, we can build for example a single-channel image of the coefficients (0,0) of each block. This image will then be N times smaller per dimension. More generally, one can build an image using the coefficients corresponding to position (i, j) of each block for any chosen pair of i and j. Additionally, one may create false color images by selecting three block positions and using the three resulting arrays as the red, green, and blue channel of the resulting image, as shown in the following equation:

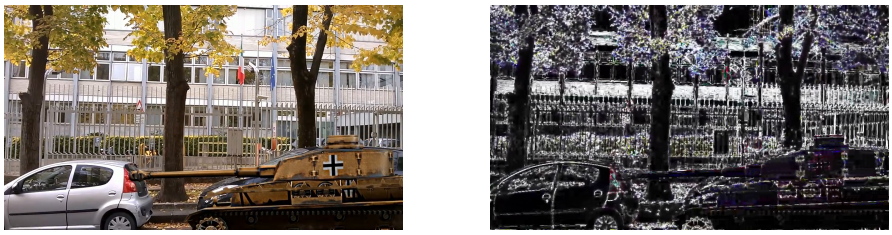
     $$\begin{pmatrix} \text{red} \\ \text{green} \\ \text{blue} \end{pmatrix} = \begin{pmatrix} \text{coefficients \#1} \\ \text{coefficients \#2} \\ \text{coefficients \#3} \end{pmatrix}. \tag{1.1}$$

     For the implementation of the Q4 filter used in InVID we chose to use blocks of size 2×2. Since the coefficient corresponding to block position (0,0) is not relevant for verification and only returns a low-frequency version of the image, we have the remaining three coefficients with which we can create a false color image. Thus, in this case the red channel corresponds to horizontal frequencies (0,1), the green channel corresponds to vertical frequencies (1,0), and the blue corresponds to frequencies along the diagonal direction (1,1).
   - The **Chrome** filter is dedicated to analyzing the luminance noise of the image. It highlights noise homogeneity, which is expected in a normal and naturally illuminated observation system. It is mainly based on a non-linear filter in order to capture impulsive noise. Hence, the Chrome filter is mainly based on the following operation applied on each frame of the video:



(a) (b)

Fig. 1.2: Output of the **Q4** filter on the edited tank video. (a) edited frame, (b) filter output. According to Equation (1.1), the image in (b) shows in red the strength of vertical transition (corresponding to transitions along the lines), in green the horizontal transitions and in blue the diagonal transitions (which can be mainly seen in the leafs of the trees).



(a) (b)

Fig. 1.3: Output of the **Chrome** filter on the edited tank video. (a) edited frame, (b) filter output. The image in (b) appears to be black and white but remains with color information. As it comes from Equation (1.2), it shows that the noise is of the same level independent of the input color bands.

$$I_{\text{Chrome}}(x) = |I(x) - \text{median}(W(I(x)))|, \tag{1.2}$$

where $I(x)$ signifies an image pixel, and $W(I(x))$ stands for a 3×3 window around that pixel.

This filter resembles the Median Noise algorithm for image forensics, implemented in the Image Forensics Toolbox[1], where the median filter residue image is used to spot inconsistencies in the image. Essentially, as it isolates high-frequency noise, this approach gives an overview of the entire frame where items with different noise traces can be spotted and identified as standing out from the rest of the frame.

2. Optical filters: Videos are acquired from an optical system coupled with a sensor system. The latter has the sole purpose of transforming light and optical information into digital data in the form of a video stream. A lot of information

[1] https://github.com/MKLab-ITI/image-forensics

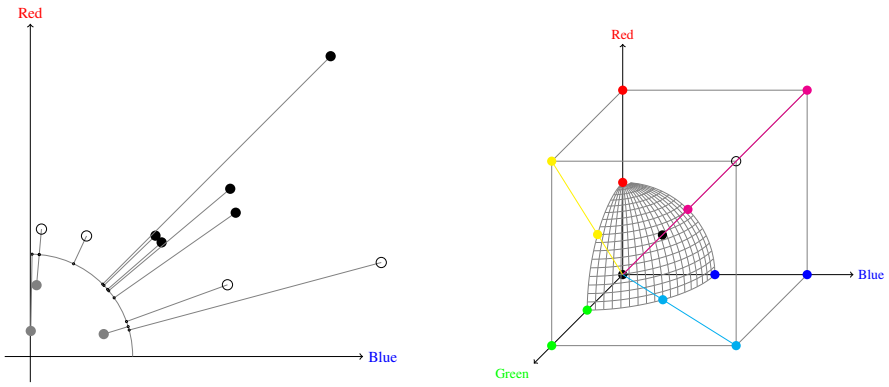


Fig. 1.4: Projection principle performed by the **Fluor** filter.

directly related to the light and optical information initially captured by the device is hidden in the structure of the video file. The purpose of **optical** filters is to extract this information and to allow the investigator to look for anomalies in the optical information patterns. It must be kept in mind that these anomalies are directly related to optical physics. Some knowledge of these phenomena is therefore required for an accurate interpretation of the results.

- The **Fluor** filter is used to study the colors of an image regardless of its luminance level. The filter produces a *normalised* image where the colors of the initial image have been restored independently of the associated luminance. The underlying transformation is the following:

$$\begin{pmatrix} \text{red} \\ \text{green} \\ \text{blue} \end{pmatrix} = \begin{pmatrix} \frac{\text{red}}{\text{red+green+blue}} \\ \frac{\text{green}}{\text{red+green+blue}} \\ \frac{\text{blue}}{\text{red+green+blue}} \end{pmatrix} \tag{1.3}$$

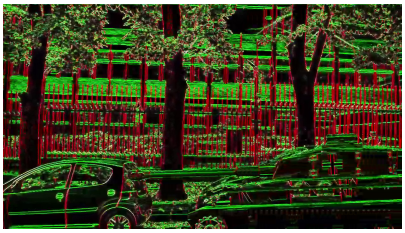
  As shown in Fig. 1.4, in 2D or 3D, colored pixels with Red, Green, and Blue components are projected on the sphere centered on the black color so that the norm of the new vector (red, green, blue) is always equal to 1.
  We see on the 2D image that the points in black represent different colors but their projections on the arc of a circle are located in the same region which induces the same hue of the image **Fluor**. On the other hand, dark pixels, drawn as points in gray in the image, may appear similar to the eye, but may actually have a different hue and their projection on the arc enhances these differences and may allow the user to distinguish between them. This normalization performed by the Fluor filter makes it possible to break the similarity of colors as it is perceived by the human visual system and to highlight colors with more pronounced differences based on their actual hue.

(a)

(b)

Fig. 1.5: Output of the **Fluor** filter on the edited tank video. (a) edited frame, (b) filter output. Image on (b) shows the colors of the original according to Equation (1.3).

(a)



(b)

Fig. 1.6: Output of the **Focus** filter on the edited tank video. (a) edited frame, (b) filter output. In the (b) image, vertical sharpness is shown in red and horizontal sharpness in green.

- The **Focus** filter is used to identify and visualise the sharp areas in an image or areas of stronger *acuity*. When an image is sharp, it has the characteristic of containing *abrupt* transitions as opposed to a *smooth* level evolution of color at the boundaries of an object. An image with high acuity contains a higher amount of high frequencies, while in contrast the high frequencies are insignificant when the object is blurred or out of focus. This sharpness estimation for the Focus filter is performed through the wavelet transform [30]. The **Focus** filter considers the wavelet coefficients only through a non-linear filtering based on the processing of the three RGB planes of each frame. It yields a false color composition where blurred low frequency areas remain in grey and the sharp contours appear in color.
- The **Acutance** filter refers to the physical term for the sharpness in photography. Normally, it is a simple measure of the slope of a local gradient but here it is normalised with the local value of the gray levels, which distinguishes it from the **Focus** filter. The Acutance filter is computed as the ratio between the outputs of a high-pass filter and a low-pass filter. In practice, we use two Gaussian filters with different sizes. Hence, the following equation characterises the **Acutance** filtering process:

(a) (b)

Fig. 1.7: Output of the **Acutance** filter on the edited tank video. (a) edited frame, (b) filter output. The image (b) stresses that the tank appears much more sharp than the rest of the image.

$$\text{frame}_{\text{Acutance}} = \frac{\text{frame}_{\text{HighPass}}}{\text{frame}_{\text{LowPass}}}. \tag{1.4}$$

3. Temporal filters: These filters aim at highlighting the behavior of the video stream over time. MPEG-4 video compression exploits temporal redundancy to reduce the compressed video size. This is the reason a compressed video is much more complex than a sequence of compressed images. Moreover, in many frames MPEG-4 mixes up the intra/inter predictions in one direction or in a forward/backward strategy, so that the frame representation is highly dependent on the frame contents and the degree of quantization. Thus, the analysis of the temporal behavior of the quantization parameters may help us detect inconsistencies in the frame representation.

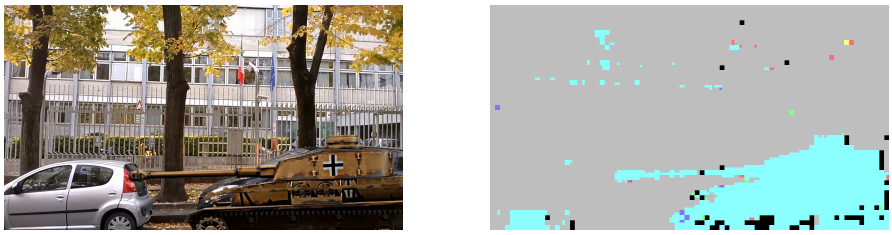
    - The **Cobalt** filter compares the original video with a modified version of the original video re-quantised by MPEG-4 with a different quality level (and a correspondingly different bit rate). The principle of the Cobalt filter is simple. One observes the video of errors[2] between the initial video and the video re-quantised by MPEG-4 with a variable quality level or a variable bit rate level. If the quantization level coincides with the quality level actually used on the small modified area, there will be no error right there. This practice is quite similar to the JPEG Ghosts algorithm [18] where a JPEG image is recompressed and the new image is subtracted from the original, to locally highlight inconsistencies ("ghosts") that correspond to added objects from images of different qualities. The ELA algorithm[3] follows a similar approach.
    - The **Motion Vectors** filter yields a color-based representation of block-motions as encoded into the video stream. Usually, this kind of representation uses arrows to show block displacements. It is worth noting that the encoding system does not recognise 'objects' but handles blocks only (namely

[2] Video of errors: a video constructed by per-pixel differences of frames between the two videos.

[3] https://fotoforensics.com/tutorial-ela.php

(a) (b)

Fig. 1.8: Output of the **Cobalt** filter on the edited tank video. (a) edited frame, (b) filter output.



(a) (b)

Fig. 1.9: Output of the **Motion Vectors** filter on the edited tank video. (a) edited frame, (b) filter output. Instead of a usual arrow-based representation of the motion vectors, image (b) shows macro-blocks displacements according to a vector orientation definition that uses the Hue angular definition of the HLS color representation. This representation allows better visualization for human investigators, and potential processing by automatic systems.

macro-blocks). The motion vectors are encoded in the video stream to reconstruct all frames which are not keyframes (i.e. not intra-coded frames but inter-coded frames that are essentially encoded by using information from other frames). Then, an object of the scene has a set of motion-vectors associated to each macro-block inside it. These motions as represented by the Motion Vectors filter have to be homogeneous and coherent, otherwise there is a high likelihood that some suspicious operation has taken place.

- The **Temporal Filter** is used to apply temporal transformation on the video, such as smoothing or temporal regulation. It should also be used to make a frame-to-frame comparison to focus on the evolution of the luminance in time only. The Temporal Filter is computed as the frame-to-frame difference over time as stated by the following equation:

$$\text{frame}_{\text{Temporal Filter}}(t) = \text{frame}(t) - \text{frame}(t-1)$$

which is applied on each color channel of the frames so that the output of the filter is also a color image.

(a) (b)

Fig. 1.10: Output of the **Temporal Filter** on the edited tank video. (a) edited frame, (b) filter output. The frame-to-frame difference shown in image (b) highlights the tank displacement as well as the light shift of the camera.

1.4.2 Tampering Detection

Besides the development of video-specific forensic filters, we also dedicated effort towards developing an automatic detection system, which would be able to assist investigators in their work. Similar to other tampering detection approaches, our methodology is to train a machine learning system using a set of input features to distinguish between tampered and non-tampered items. The approach, presented in [63], is based on image classification. Since the filters produce colorised sequences of outputs in the form of digital videos, we decided to use image classification networks in order to model the way a human investigator would look for inconsistencies in the filter outputs.

Deep networks generally require very large training sets. This is the reason the authors of [40] resorted to (semi-)automatically generated face-swap videos for training and evaluation. However, for the general case of video tampering, such videos do not exist. On the other hand, in contrast to other methods which are based on filter outputs that are not readable by humans, the outputs produced by our filters are designed to be visually interpreted by users. This means that we can treat the task as a generic image classification task, and refine networks that have been pre-trained on general image datasets.

Even in this case, there is need for a large number of items, which was not available. Similar to [41], we decided to deal with the problem at the frame level. Thus, each frame was treated as a separate item, and accordingly the system was trained to distinguish between tampered and untampered frames. There are admittedly strong correlations between consecutive video frames, which reduces the variability in the training set, but operating at the frame level remains the only viable strategy given the limited available data. Of course, during training and evaluation, caution needs to be applied so as to ensure that all the frames from the same video remain exclusively either in the training or test set, and that no information leak takes place.

For classification, we chose Convolutional Neural Networks (CNNs) which are currently the dominant approach for this type of task. Specifically, we chose GoogLeNet [50] and ResNet [24], which are two very successful models for image

classification. In order to apply them to tampering detection, we initialise the models with pre-trained weights from the ImageNet dataset, and fine-tune them using annotated filter outputs from our datasets.

To produce the outputs, we chose the Q4 and Cobalt filters for classification, which represent two complementary aspects of digital videos: Q4 provides us with frequency analysis through the DCT transform, and Cobalt visualises the requantization residue. The CNNs are designed to accept inputs in a fixed resolution of 224×224 pixels. We thus rescaled all filter outputs to match these dimensions. Generally, in multimedia forensics, rescaling is a very disruptive operation that tends to erase the -usually very sensitive- traces of tampering. However, in our case, the forensic filters we are using are designed to be visually inspected by humans and, as a result, exhibit no such sensitivities. Thus, we can safely adjust their dimensions to the CNNs.

One final note on the CNNs is that, instead of using their standard architecture, we extend them using the proposed approach of [36]. The work of [36] shows that, if we extend the CNN with an additional Fully Connected (FC) layer before the final FC layer, the network classification performance is improved significantly. We chose to add an 128-unit FC layer to both networks, and we also replaced the final 1000-unit layer, aimed at the 1000-class ImageNet task, with a 2-unit layer appropriate for the binary (tampered/untampered) task.

1.5 Results

1.5.1 Datasets and Experimental Setup

This section is dedicated to the quantitative evaluation of the proposed tampering detection approach. We drew videos from two different sources to create our training and evaluation datasets. One source was the NIST 2018 Media Forensics Challenge[4] and specifically the annotated development datasets provided for the *Video Manipulation Detection* task. The development videos provided by NIST were split in two separate datasets, named Dev1 and Dev2. Out of those datasets, we kept all tampered videos, plus their untampered sources, but did not take into account the various distractor videos included in the sets, which would lead to significant class imbalances, and also because we decided to train the video using corresponding pairs of tampered videos and their sources, which are visually similar to a large extent. The aim was to allow the network to ignore the effects of the visual content -since it would not allow it to discriminate between the two- and focus on the impact of the tampering.

In our experiments Dev1 consists of 30 tampered videos and their 30 untampered sources, while Dev2 contains 86 tampered videos, and their 86 untampered sources. The two datasets contain approximately 44,000 and 134,000 frames respectively,

[4] https://www.nist.gov/itl/iad/mig/media-forensics-challenge-2018

which are generally evenly shared between tampered and untampered videos. It should be kept in mind that the two datasets originate from the same source (NIST), and thus, while during our experiments we treat them as different sets, it is very likely that they will exhibit similar feature distributions.

The other source was the InVID Fake Video Corpus [35], a collection of real and fake videos developed in the course of the InVID project. The version of the FVC used in these experiments consists of 110 "real" and 117 "fake" news-related, user-generated videos from various social media sources. These are videos that convey factual or counterfactual information, but the distinction between tampered and untampered is not clear, since many "real" videos contain watermarks or logos, which means they should be detected as tampered, and in contrast many "fake" videos are untampered user-captured videos that were circulated out of context. Out of that collection, we selected 35 "real", unedited videos, and 33 "fake" videos that were tampered with the aim of deceiving viewers, but with no obvious edits such as logos, watermarks, or cuts/transitions. In total, the subset of the FVC dataset we created contains 163,000 frames, which are approximately evenly shared between tampered and untampered videos.

Fig. 1.11: Indicative videos from the FVC dataset. Top (tampered videos): "Bear attacks cyclist", "Lava selfie", "Bear attacks snowboarder", "Eagle drops snake". Bottom (untampered videos): "Stockholm attack", "Hudson landing", "Istanbul attack" and "Giant aligator in golf field".

One major problem with the dataset is that we do not have accurate temporal annotations for most videos. That is, in many cases where only part of the video contains tampered areas, and the rest is essentially identical to the untampered version, we do not have specific temporal or per-frame annotations. As an approximation in our experiments, we labelled all the frames that we drew from tampered videos as tampered, and all the frames we drew from untampered videos as untampered. This is a weak assumption, and we can be certain that a percentage of our annotations will be wrong. However, based on manual inspection, we concluded that it is indeed true for the majority of videos -meaning, in most cases the tampering appears on the frame from the beginning to the end of the video-, and thus we consider the quality of annotations adequate for the task.

1.5.2 Experimental Setup

For our evaluation experiments, we first applied the two chosen filters, namely Q4 and Cobalt, on all videos, and extracted all frames of the resulting output sequences to use as training and test items. Then, each of the two chosen networks -GoogLeNet and ResNet- was trained on the task using these outputs. For comparison, we also implemented three more features from related approaches, to be used for classification in a similar manner. These features are:

- *rawKeyframes* [40]. The video is decoded into its frames and the raw keyframes (without any filtering process) are given as input to the deep network.
- *highPass frames* [20]. The video is decoded into its frames, each frame is filtered by a high-pass filter and the filtered frame is given as input to the deep network.
- *frameDifference* [60]. The video is decoded into its frames, the frame difference between two neighboring frames is calculated, the new filtered frame is also processed by a high-pass filter and the final filtered frame is given as input to the deep network.

As explained, during training each frame is treated as an individual image. However, in order to test the classifier, we require a per-video result. To achieve this, we extract the classification scores for all frames, and calculate the average score separately for each class (tampered, untampered). If the average score for the "tampered" class is higher than the average score for the "untampered" class, then the video is classified as tampered.

We ran two types of experiments. In one case, we trained and evaluated the algorithm on the same dataset, using 5-fold cross validation, and ensuring that all frames from a video are placed either in the training or in the evaluation set to avoid information leak. In the other case, we used one of the datasets for training, and the other two for testing. These cross-dataset evaluations are important in order to evaluate an algorithm's ability to generalise, and to assess whether any encouraging results we observe during within-dataset evaluations are actually the result of overfitting on the particular dataset's characteristics, rather than a true solution to the task. In all cases, we used three performance measures: Accuracy, Mean Average Precision (MAP), and Mean Precision for the top-20 retrieved items (MP@20). A preliminary version of these results has also been presented in [63].

1.5.2.1 Within-dataset Experiments

For the within-dataset evaluations, we used the two NIST datasets (Dev1, Dev2) and their union. This resulted in three separate runs, the results of which are presented in Table 1.1.

As shown on the Table 1.1, Dev1 consistently leads to poorer performance in all cases, for all filters and both models. The reason we did not apply the MP@20 measure on Dev1 is that the dataset is so small that the test set in all cases contains

Table 1.1: Within-dataset evaluations

Dataset	Filter-DCNN	Accuracy	MAP	MP@20
Dev1	cobalt-gnet	**0.6833**	0.7614	-
	cobalt-resnet	0.5833	0.6073	-
	q4-gnet	0.6500	**0.7856**	-
	q4-resnet	0.6333	0.7335	-
Dev2	cobalt-gnet	0.8791	**0.9568**	**0.82**
	cobalt-resnet	0.7972	0.8633	0.76
	q4-gnet	**0.8843**	0.9472	0.79
	q4-resnet	0.8382	0.9433	0.76
Dev1 + Dev2	cobalt-gnet	**0.8509**	0.9257	0.91
	cobalt-resnet	0.8217	0.9069	0.87
	q4-gnet	0.8408	**0.9369**	**0.92**
	q4-resnet	0.8021	0.9155	0.87

less than 20 items, and thus is inappropriate for the specific measure. Accuracy is between 0.58 and 0.68 in all cases in Dev1, while it is significantly higher in Dev2, ranging from 0.79 to 0.88. MAP is similarly significantly higher in Dev2. This can be explained by the fact that Dev2 contains many videos that are taken from the same locations, so we can deduce that a degree of leakage occurs between training and test data, which leads to seemingly more successful detections.

We also built an additional dataset by merging Dev1 and Dev2. The increased size of the Dev1+Dev2 dataset suggests that cross-validation results will be more reliable than for the individual sets. As shown in Table 1.1, Mean Average Precision for Dev1+Dev2 falls between that for Dev1 and Dev2, but is much closer to Dev2. On the other hand, MP@20 is higher than for Dev2, although that could possibly be the result of Dev2 being relatively small. The cross-validation Mean Average Precision for Dev1+Dev2 reaches 0.937 which is a very high value and can be considered promising with respect to the task. It is important to note that, for this set of evaluations, the two filters yielded comparable results, with Q4 being superior in some cases and Cobalt in others. On the other hand, with respect to the two CNN models there seems to be a significant difference between GoogLeNet and ResNet, with the former yielding much better results.

1.5.2.2 Cross-dataset Experiments

Within-dataset evaluations using cross-validation is the typical way to evaluate automatic tampering detection algorithms. However, as we are dealing with machine learning, it does not account for the possibility of the algorithm actually learning specific features of a particular dataset, and thus remaining useless for general application. The most important set of algorithm evaluations for InVID automatic tampering detection concerned cross-dataset evaluation, with the models being trained on one dataset and tested on another.

Table 1.2: Cross-dataset evaluations (Training set: Dev1)

Training	Testing	Filter-DCNN	Accuracy	MAP	MP@20
Dev1	**Dev2**	cobalt-gnet	0.5818	0.7793	0.82
		cobalt-resnet	0.6512	0.8380	0.90
		q4-gnet	0.5232	0.8282	0.90
		q4-resnet	0.5240	0.8266	**0.93**
		rawKeyframes-gnet [40]	0.5868	0.8450	0.85
		rawKeyframes-resnet [40]	0.4512	0.7864	0.75
		highPass-gnet [20]	0.5636	0.8103	0.88
		highPass-resnet [20]	0.5901	0.8026	0.84
		frameDifference-gnet [60]	**0.7074**	**0.8585**	0.87
		frameDifference-resnet [60]	0.6777	0.8240	0.81
	FVC	cobalt-gnet	0.5147	0.5143	0.48
		cobalt-resnet	0.4824	0.5220	0.50
		q4-gnet	0.5824	0.6650	0.64
		q4-resnet	**0.6441**	**0.6790**	**0.69**
		rawKeyframes-gnet [40]	0.5265	0.5261	0.49
		rawKeyframes-resnet [40]	0.4882	0.4873	0.44
		highPass-gnet [20]	0.5441	0.5359	0.51
		highPass-resnet [20]	0.4882	0.5092	0.49
		frameDifference-gnet [60]	0.5559	0.5276	0.46
		frameDifference-resnet [60]	0.5382	0.4949	0.51

The training-test sets were based on the three datasets we described above, namely Dev1, Dev2, and FVC. Similar to subsection 1.5.2.1, we also combined Dev1 and Dev2 to create an additional dataset, named Dev1+Dev2. Given that Dev1 and Dev2 are both taken from the NIST challenge, although different, we would expect that they would exhibit similar properties and thus should give relatively better results than when testing on FVC. In contrast, evaluations on the FVC correspond to the most realistic and challenging scenario, that is training on benchmark, lab-generated content, and testing on real-world content encountered on social media. Given the small size and the extremely varied content of the FVC, we opted not to use it for training, but only as a challenging test set.

The results are shown in Tables 1.2, 1.3, and 1.4. Using Dev1 to train and Dev2 to test, and vice versa, yields comparable results to the within-dataset evaluations for the same dataset, confirming our expectation that, due to the common source of the two datasets, cross-dataset evaluation for these datasets would not be particularly challenging. Compared to other approaches, it seems that our proposed approaches do not yield superior results in those cases. Actually, the *frameDifference* feature seems to outperform the others in those cases.

The situation changes in the realistic case where we are evaluating on the Fake Video Corpus. In that case, the performance drops significantly. In fact, most algorithms drop to an Accuracy of almost 0.5. One major exception, and the most notable finding in our investigation, is the performance of the Q4 filter when used to train a GoogLeNet model. In this case, the performance is significantly higher than in any other case, and remains promising with respect to the potential of real-world

Table 1.3: Cross-dataset evaluations (Training set: Dev2)

Training	Testing	Filter-DCNN	Accuracy	MAP	MP@20
Dev2	**Dev1**	cobalt-gnet	0.5433	0.5504	0.55
		cobalt-resnet	0.5633	0.6563	0.63
		q4-gnet	0.6267	0.6972	**0.71**
		q4-resnet	0.5933	0.6383	0.63
		rawKeyframes-gnet	**0.6467**	0.6853	0.65
		rawKeyframes-resnet	0.6200	0.6870	0.62
		highPass-gnet [20]	0.5633	0.6479	0.66
		highPass-resnet [20]	0.6433	0.6665	0.65
		frameDifference-gnet [60]	0.6133	**0.7346**	0.70
		frameDifference-resnet [60]	0.6133	0.7115	0.67
	FVC	cobalt-gnet	0.5676	0.5351	0.58
		cobalt-resnet	0.5059	0.4880	0.49
		q4-gnet	**0.6118**	**0.6645**	**0.70**
		q4-resnet	0.5000	0.4405	0.39
		rawKeyframes-gnet [40]	0.5206	0.6170	0.66
		rawKeyframes-resnet [40]	0.5971	0.6559	0.69
		highPass-gnet [20]	0.4794	0.5223	0.47
		highPass-resnet [20]	0.5235	0.5541	0.58
		frameDifference-gnet [60]	0.4882	0.5830	0.64
		frameDifference-resnet [60]	0.5029	0.5653	0.59

Table 1.4: Cross-dataset evaluations (Training set: Dev1+Dev2)

Training	Testing	Filter-DCNN	Accuracy	MAP	MP@20
Dev1 + Dev2	**FVC**	cobalt-gnet	0.5235	0.5178	0.54
		cobalt-resnet	0.5029	0.4807	0.47
		q4-gnet	**0.6294**	**0.7017**	**0.72**
		q4-resnet	0.6000	0.6129	0.64
		rawKeyframes-gnet	0.6029	0.5694	0.53
		rawKeyframes-resnet	0.5441	0.5115	0.52
		highPass-gnet	0.5147	0.5194	0.53
		highPass-resnet	0.5294	0.6064	0.70
		frameDifference-gnet	0.5176	0.5330	0.55
		frameDifference-resnet	0.4824	0.5558	0.54

application. Being able to generalise into new data with unknown feature distributions is the most important feature in this respect, since it is very unlikely at this stage that we will be able to create a large-scale training dataset to model any real world case.

Trained on Dev1+Dev2, the Q4 filter combined with GoogLeNet yields a MAP of 0.711. This is a promising result and significantly higher than all competing alternatives. Still, however, it is not sufficient for direct real-world application, and further refinement would be required to improve this.

1.6 Conclusions and Future Work

We presented our efforts toward video forensics, and the development of the tampering detection and localization components of InVID. We explored the state of the art in video forensics, identified the current prospects and limitations of the field, and then proceeded to advance the technology and develop novel approaches.

We first developed a series of video forensics filters aimed to analyse videos from various perspectives, and highlight potential inconsistencies in different spectrums that may correspond to traces of tampering. These filters are aimed to be interpreted by human investigators and are based on three different types of analysis, namely algebraic processing of the video input, optical features, and temporal video patterns.

With respect to automatic video tampering detection, we developed an approach based on combining the video forensics filters with deep learning models designed for visual classification. The aim was to evaluate the extent to which we could automate the process of analyzing the filter outputs using deep learning algorithms. We evaluated two of the filters developed in InVID, combined with two different deep learning architectures. The conclusion was that, while alternative features performed better in within-dataset evaluations, the InVID filters were more successful in realistic cross-dataset evaluations, which are the most relevant in assessing the potential for real-world application.

Still, more effort is required to reach the desired accuracy. One major issue is the lack of accurate temporal annotations for the datasets. By assigning the "tampered" label on all frames of tampered videos, we are ignoring the fact that tampered videos may also contain frames without tampering, and as a result the labelling is inaccurate. This may be resulting in noisy training, which may be a cause of reduced performance. Furthermore, given the per-frame classification outputs, currently we calculate the per-video score by comparing the average "tampered" score with the average "untampered" score. This approach may not be optimal, and different ways of aggregating per-frame to per-video scores.

Currently, given the evaluation results, we cannot claim that we are ready for real-world application, nor that we have exhaustively evaluated the proposed automatic detection algorithm. In order to improve the performance of the algorithm and run more extensive evaluations, we intend to improve the temporal annotations of the provided datasets and continue collecting real-world cases to create a larger-scale evaluation benchmark. Finally, given that the current aggregation scheme may not be optimal, we will explore more alternatives in the hope of improving the algorithm performance and should extend our investigations into more filters and CNN models, in order to improve performance, including the possibility of using feature fusion by combining the outputs of multiple filters in order to assess each video.

References

1. Afchar, D., Nozick, V., Yamagishi, J., Echizen, I.: MesoNet: a compact facial video forgery detection network. CoRR **abs/1809.00888** (2018)
2. Baek, K., Bang, D., Shim, H.: Editable generative adversarial networks: Generating and editing faces simultaneously. CoRR **abs/1807.07700** (2018)
3. Bansal, A., Ma, S., Ramanan, D., Sheikh, Y.: Recycle-GAN: Unsupervised Video Retargeting. In: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (eds.) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proc., Part V, *Lecture Notes in Computer Science*, vol. 11209, pp. 122–138. Springer (2018)
4. Bayar, B., Stamm, M.C.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: Proc. of the 4th ACM Workshop on Information Hiding and Multimedia Security, pp. 5–10. ACM (2016)
5. Bayar, B., Stamm, M.C.: On the robustness of constrained convolutional neural networks to JPEG post-compression for image resampling detection. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2152–2156. IEEE (2017)
6. Birajdar, G.K., Mankar, V.H.: Blind method for rescaling detection and rescale factor estimation in digital images using periodic properties of interpolation. AEU-International Journal of Electronics and Communications **68**(7), 644–652 (2014)
7. Chen, J., Kang, X., Liu, Y., Wang, Z.J.: Median filtering forensics based on convolutional neural networks. IEEE Signal Processing Letters **22**(11), 1849–1853 (2015)
8. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
9. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proc. of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258 (2017)
10. Cozzolino, D., Gragnaniello, D., Verdoliva, L.: Image forgery detection through residual-based local descriptors and block-matching. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 5297–5301. IEEE (2014)
11. Cozzolino, D., Poggi, G., Verdoliva, L.: Splicebuster: A new blind image splicing detector. In: 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6. IEEE (2015)
12. Cozzolino, D., Poggi, G., Verdoliva, L.: Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In: Proc. of the 5th ACM Workshop on Information Hiding and Multimedia Security, pp. 159–164. ACM (2017)
13. Dehnie, S., Sencar, H.T., Memon, N.D.: Digital image forensics for identifying computer generated and digital camera images. In: Proc. of the 2006 IEEE International Conference on Image Processing (ICIP 2006), pp. 2313–2316. IEEE (2006). URL `http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4106439`
14. Dirik, A.E., Bayram, S., Sencar, H.T., Memon, N.D.: New features to identify computer generated images. In: Proc. of the 2007 IEEE International Conference on Image Processing (ICIP 2007), pp. 433–436. IEEE (2007). URL `http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4378863`
15. Dirik, A.E., Memon, N.: Image tamper detection based on demosaicing artifacts. In: Proc. of the 2009 IEEE International Conference on Image Processing (ICIP 2009), pp. 1497–1500. IEEE (2009)
16. Dong, Q., Yang, G., Zhu, N.: A MCEA based passive forensics scheme for detecting frame based video tampering. Digital Investigation pp. 151–159 (2012)
17. Fallahpour, M., Shirmohammadi, S., Semsarzadeh, M., Zhao, J.: Tampering detection in compressed digital video using watermarking. IEEE Transactions on Instrumentation and Measurement **63**(5), 1057–1072 (2014)
18. Farid, H.: Exposing digital forgeries from JPEG ghosts. IEEE Transactions on Information Forensics and Security **4**(1), 154–160 (2009)

19. Ferrara, P., Bianchi, T., De Rosa, A., Piva, A.: Image forgery localization via fine-grained analysis of CFA artifacts. IEEE Transactions on Information Forensics and Security **7**(5), 1566–1577 (2012)
20. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security **7**(3), 868–882 (2012)
21. Fu, D., Shi, Y., Su, W.: A generalized Benford's law for JPEG coefficients and its applications in image forensics. In: Proc. of SPIE, Security, Steganography and Watermarking of Multimedia Contents IX, vol. 6505, p. 39–48 (2009)
22. Gironi, A., Fontani, M., Bianchi, T., Piva, A., Barni, M.: A video forensic technique for detecting frame deletion and insertion. In: ICASSP (2014)
23. Grana, C., Cucchiara, R.: Sub-shot summarization for MPEG-7 based fast browsing. In: Post-Proc. of the Second Italian Research Conference on Digital Library Management Systems (IRCDL 2006), Padova, 27th January 2006 [23], pp. 80–84
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 770–778 (2016). DOI 10.1109/CVPR.2016.90
25. Iakovidou, C., Zampoglou, M., Papadopoulos, S., Kompatsiaris, Y.: Content-aware detection of JPEG grid inconsistencies for intuitive image forensics. Journal of Visual Communication and Image Representation **54**, 155–170 (2018)
26. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: Proc. of the European Conference on Computer Vision (ECCV), pp. 35–51 (2018)
27. Lin, C.S., Tsay, J.J.: A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis. Digital Investigation **11**(2), 120–140 (2014)
28. Liu, Y., Guan, Q., Zhao, X., Cao, Y.: Image forgery localization based on multi-scale convolutional neural networks. In: Proc. of the 6th ACM Workshop on Information Hiding and Multimedia Security, pp. 85–90. ACM (2018)
29. Mahdian, B., Saic, S.: Using noise inconsistencies for blind image forensics. Image and Vision Computing **27**(10), 1497–1503 (2009)
30. Mallat, S.: A Wavelet Tour of Signal Processing, Third edn. Academic Press (2009)
31. Milani, S., Bestagini, P., Tagliasacchi, M., Tubaro, S.: Multiple compression detection for video sequences. In: MMSP, pp. 112–117. IEEE (2012). URL `http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6331800`
32. Muhammad, G., Al-Hammadi, M.H., Hussain, M., Bebis, G.: Image forgery detection using steerable pyramid transform and local binary pattern. Machine Vision and Applications **25**(4), 985–995 (2014)
33. Pandey, R., Singh, S., Shukla, K.: Passive copy-move forgery detection in videos. In: IEEE International Conference On Computer and Communication Technology (ICCCT), pp. 301–306 (2014)
34. Papadopoulou, O., Zampoglou, M., Papadopoulos, S., Kompatsiaris, I.: A corpus of debunked and verified user-generated videos. Online Information Review (2018)
35. Papadopoulou, O., Zampoglou, M., Papadopoulos, S., Kompatsiaris, Y., Teyssou, D.: Invid Fake Video Corpus v2.0 (version 2.0). Dataset on Zenodo (2018)
36. Pittaras, N., Markatopoulou, F., Mezaris, V., Patras, I.: Comparison of Fine-Tuning and Extension Strategies for Deep Convolutional Neural Networks. In: Proc. of the 23rd International Conference on MultiMedia Modeling (MMM 2017), pp. 102–114. Springer, Reykjavik, Iceland (2017)
37. Piva, A.: An overview on image forensics. ISRN Signal Processing pp. 1–22 (2013)
38. Qi, X., Xin, X.: A singular-value-based semi-fragile watermarking scheme for image content authentication with tamper localization. Journal of Visual Communication and Image Representation **30**, 312–327 (2015)
39. Qin, C., Ji, P., Zhang, X., Dong, J., Wang, J.: Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. Signal Processing **138**, 280–293 (2017)

40. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Faceforensics: A large-scale video dataset for forgery detection in human faces. CoRR **abs/1803.09179** (2018). ArXiv:1803.09179v1
41. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Faceforensics++: Learning to detect manipulated facial images. arXiv preprint arXiv:1901.08971 (2019)
42. Shanableh, T.: Detection of frame deletion for digital video forensics. Digital Investigation **10**(4), 350 – 360 (2013). DOI https://doi.org/10.1016/j.diin.2013.10.004. URL `http://www.sciencedirect.com/science/article/pii/S1742287613001102`
43. Shehab, A., Elhoseny, M., Muhammad, K., Sangaiah, A.K., Yang, P., Huang, H., Hou, G.: Secure and robust fragile watermarking scheme for medical images. IEEE Access **6**, 10,269–10,278 (2018)
44. Singh, R., Vatsa, M., Singh, S.K., Upadhyay, S.: Integrating SVM classification with svd watermarking for intelligent video authentication. Telecommunication Systems **40**(1-2), 5–15 (2009)
45. Sitara, K., Mehtre, B.M.: Digital video tampering detection: An overview of passive techniques. Digital Investigation **18**, 8–22 (2016)
46. Soni, B., Das, P.K., Thounaojam, D.M.: CMFD: a detailed review of block based and key feature based techniques in image copy-move forgery detection. IET Image Processing **12**(2), 167–178 (2017)
47. Sowmya, K., Chennamma, H., Rangarajan, L.: Video authentication using spatio temporal relationship for tampering detection. Journal of Information Security and Applications **41**, 159–169 (2018)
48. Su, L., Huang, T., Yang, J.: A video forgery detection algorthm based on compressive sensing. Multimedia Tools and Applications **74**, 6641–6656 (2015)
49. Su, Y., Xu, J.: Detection of double compression in MPEG-2 videos. In: IEEE 2nd International Workshop on Intelligent Syetems and Application (ISA) (2010)
50. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), pp. 1–9 (2015)
51. Tong, M., Guo, J., Tao, S., Wu, Y.: Independent detection and self-recovery video authentication mechanism using extended NMF with different sparseness constraints. Multimedia Tools and Applications **75**(13), 8045–8069 (2016)
52. Vázquez-Padín, D., Comesana, P., Pérez-González, F.: An SVD approach to forensic image resampling detection. In: 2015 23rd European Signal Processing Conference (EUSIPCO), pp. 2067–2071. IEEE (2015)
53. Wang, J., Li, T., Shi, Y.Q., Lian, S., Ye, J.: Forensics feature analysis in quaternion wavelet domain for distinguishing photographic images and computer graphics. Multimedia Tools Appl **76**(22), 23,721–23,737 (2017)
54. Wang, W., Farid, H.: Exposing digital forgery in video by detecting double MPEG compression. In: Proc. of the 8th workshop on multimedia and security. ACM, pp. 37–47 (2006)
55. Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J.: Image inpainting via generative multi-column convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 331–340 (2018)
56. Warif, N.B.A., Wahab, A.W.A., Idris, M.Y.I., Ramli, R., Salleh, R., Shamshirband, S., Choo, K.K.R.: Copy-move forgery detection: Survey, challenges and future directions. Journal of Network and Computer Applications **100**(75), 259–278 (2016)
57. Wu, Y., Jiang, X., Sun, T., Wang, W.: Exposing video inter-frame forgery based on velocity field consistency. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2014)
58. Xu, J., Mukherjee, L., Li, Y., Warner, J., Rehg, J.M., Singh, V.: Gaze-enabled egocentric video summarization via constrained submodular maximization. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [58], pp. 2235–2244. URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2015.html#XuMLWRS15`
59. Xu, J., Su, Y., liu, Q.: Detection of double MPEG-2 compression based on distribution of dct coefficients. International J. Pattern Recognition and Artificial Intelligence **27**(1) (2013)

60. Yao, Y., Shi, Y., Weng, S., Guan, B.: Deep learning for detection of object-based forgery in advanced video. Symmetry **10**(1), 3 (2017)
61. Ye, S., Sun, Q., Chang, E.C.: Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In: 2007 IEEE International Conference on Multimedia and Expo, pp. 12–15. IEEE (2007)
62. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5505–5514 (2018)
63. Zampoglou, M., Markatopoulou, F., Mercier, G., Touska, D., Apostolidis, E., Papadopoulos, S., Cozien, R., Patras, I., Mezaris, V., Kompatsiaris, I.: Detecting tampered videos with multimedia forensics and deep learning. In: International Conference on Multimedia Modeling, pp. 374–386. Springer (2019)
64. Zampoglou, M., Papadopoulos, S., Kompatsiaris, Y.: Large-scale evaluation of splicing localization algorithms for web images. Multimedia Tools and Applications **76**(4), 4801–4834 (2017)
65. Zhang, Y., Li, S., Wang, S., Shi, Y.Q.: Revealing the traces of median filtering using high-order local ternary patterns. IEEE Signal Processing Letters **3**(21), 275–279 (2014)
66. Zhang, Z., Hou, J., Ma, Q., Li, Z.: Efficient video frame insertion and deletion detection based on inconsistency of correlations between local binary pattern coded frames. Security and Communicatin networks **8**(2) (2015)
67. Zhi-yu, H., Xiang-hong, T.: Integrity authentication scheme of color video based on the fragile watermarking. In: 2011 International Conference on Electronics, Communications and Control (ICECC), pp. 4354–4358. IEEE (2011)