Light Weight Digital Watermarking Method
for Live Videos Forgeries Detection

By
Mohammed Kamal Al Arabid

Supervised by
Dr. Tawfiq Barhoom
Associate Prof. of Applied Computer Technology

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Information Technology

July/2018
I understand the nature of plagiarism, and I am aware of the University’s policy on this. The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted by others elsewhere for any other degree or qualification.

| Student’s name:  | محمد كمال سالم العرابيد | اسم الطالب: |
| Signature:       |                          | التوقيع: |
| Date:            |                          | التاريخ: |
نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة جامعة البحث العلمي والدراسات العليا بجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ محمد مسلم العبار في كلية تكنولوجيا المعلومات والموارد الرقمية، ورغم المسار الأخرى، برنامج تكنولوجيا المعلومات والموارد الرقمية، برنامج تكنولوجيا المعلومات والموارد الرقمية، ورئيس لجنة الحكم.

طريقة قليلة تقلل تعقيد العلامات المائية للكشف عن التزوير في الفيديوهات المباشرة

Light Weight Digital Watermarking Method for Live Videos Forgeries Detection

بعد المداولات التي تمت الباحة في عشيرة صباحاً، في قاعة مبنى القسم، اجتمعت لجنة الحكم على أطروحة الباحث/ محمد مسلم العبار في كلية تكنولوجيا المعلومات، ورغم المسار الأخرى، برنامج تكنولوجيا المعلومات.

وبعد مداخلة أوصت لجنة بمنح الباحث درجة الماجستير في كلية تكنولوجيا المعلومات، برنامج تكنولوجيا المعلومات، وللباحثة طلبت من أن يتم تدشينها بقوة الله تعالى وأززوم، طاعته وأن يسخر علمه في خدمة دينه ووطنه.

وأي الله ولن يفوته...

أ.د. مازن إسماعيل هنية

P.O. Box 188, Gaza, Palestine Tel: +970 (8) 264 4400 Fax: +970 (8) 264 4400 public@lugaza.edu.ps www.lugaza.edu.ps
الموضوع: استلام النسخة الإلكترونية لرسالة علمية

قامت إدارة المكتبات بالجامعة الإسلامية بأستلام النسخة الإلكترونية من رسالة الطالب.

رقم جامعي: 498701124
قسم: علوم إسلامية
كلية: نظم الحاسب

وتم الإطلاع عليها، ومتابعتها بالنسخة الورقية للرسالة نفسها، ضمن المحددات المبينة أدناه:

- تم إجراء جميع التعديلات التي طالبتها لجنة المناقشة.
- تم توقيع المشرفين على النسخة الورقية لاعتمادها كنسخة معدلة ونهائية.
- تم وضع ختم "عمادة الدراسات العليا" على النسخة الورقية لاعتماد توقيع المشرفين.

(WORD PDF وWORD PDF)

- وجود فهرس الرسالة، والملخصين باللغتين العربية والإنجليزية بملفات منفصلة.
- تطابق النص في كل صفحة ورقية مع النص في كل صفحة نقالة في الصفحات الإلكترونية.
- تطابق التنسيق في جميع الصفحات (نوع وحجم الخط) بين النسخة الورقية والإلكترونية.

ملاحظة: ستقوم إدارة المكتبات بنشر هذه الرسالة كملف بصيغة (PDF) على موقع المكتبة الإلكتروني.

إدارة المكتبة المركزية

توقيع الطالب
Abstract

With the advancement of the digital video and digital image editing tools, thus increases the difficulty for humans to identify visually the authentic video from the forged copy, this needs a powerful method that investigates various video from tempering attempts.

There have been several previous kinds of research which propose methods for video forgery detection. However, there are only a few methods that discuss detecting forgeries on live streaming and recorded videos. Therefore, these works have limitations, that they are used a huge amount of memory to store images, consequently, this requires much comparing process and increase the execution time of detecting forgery in such video streams.

This research has been proposed a lightweight method this means reduce memory usage and execution time for detecting forgery in video streams, based on measuring pixel intensity to explore various forgeries type on streaming videos, so a method has been suggested to enhance the process of detecting video forgeries in an efficient and effective way to protect the integrity and authenticity of videos.

Experiments have been conducted on a dataset of 27 videos covering different cases using local IP camera, cases like using compression algorithms and some tempering effects. The result indicated that the system achieved high relevant measures with 97% accuracy on detecting duplication on a case of lossless compression and 95% on the other compression algorithms; on the other hand on detecting cloning forgeries our system achieved a highly relevant measure with 86% on a case of lossless compression.

**Keywords:** Digital Tampering, Digital Forensics, Video Forgery, Streaming attacks, Digital Watermarking, Streaming Security.
الملخص

مع تقدم أدوات تحرير الصور الرقمية، فذلك يزيد من صعوبة تعرف البشر على الفيديو الأصلي من النسخة المزورة، وهذا يحتاج إلى طريقة قوية للتحقق في مختلف أنواع محولات التزوير في الفيديو.

هناك العديد من الإبحارات السابقة التي تقترح أساليب للكشف عن التزوير في الفيديوهات، مع العلم أنه يوجد عدد قليل من الأبحاث التي تناقش عمليات التزوير في مقاطع الفيديو المسجلة بواسطة البث المباشر، إضافة إلى ذلك تلك الأبحاث تحتوي على قيود حيث انها تستخدم كما هنالك من الذاكرة لتخزين الصور، وهذا يتطلب الكثير من الوقت في عملية المقارنة واكتشاف التزوير في الفيديوهات المباشرة.

في هذا البحث تم إقتراح طريقة قليلة الثقل للكشف عن نوع التزوير و أماكنها في الفيديوهات المباشرة، حيث ان هذه الطريقة تقلل من استخدام الذاكرة ووقت التنفيذ للكشف عن التزوير في الفيديوهات المباشرة استنادًا إلى قياس كثافة البكسل و العلامات المائية المنشورة في الفيديو المصور في مرحلة التصبير، ولذلك تم اقتراح طريقة لتعزيز عملية الكشف عن التزوير بطريقة فعالة لحماية سلامة وموثوقياً مقاطع الفيديو.

أجريت التجارب على 27 متقطع فيديو تغطي حالات مختلفة باستخدام كاميرا IP، وأشارت النتائج إلى أن النظام حقق نتائج عالية الدقة في الكشف عن التزوير في معظم التجارب.

الكلمات المفتاحية: العباث الرقمي، التحاليل الجنائية الرقمية، تزوير الفيديو، هجمات الفيديوهات المباشرة، العلامات الرقمية المائية، أمن الفيديوهات المباشرة.
Dedication

This research is dedicated to my father Kamal, my mother, sisters, brothers, my wife and my beloved son Kamal, friends and all one who encourage me to complete my study.
Acknowledgment

My great thanks to Allah the most Merciful, the lord of the words for his help and guidance to finish my research, and the great thanks to our messenger Mohammad.

Firstly, I would like to express my sincere gratitude to my advisor Dr. Tawfiq S.M. Barhoom, associate professor of Information Technology in the Islamic university for the continuous support of my master study, for his patience, motivation, and immense knowledge. He consistently allowed this thesis to be my own work, and steered me in the right direction.

I would like to thank my dear father Kamal, my mother, everybody in my family and my dear wife for their love and support during my study, they have always encouraged me towards excellence.

I would also like to thank my wife family for their supporting and encouraging me for the better.

Deepest thanks go to all one who took part in making this thesis.
# Table of Contents

Abstract ......................................................................................................................... iv

ملخص ............................................................................................................................. vi

Dedication ........................................................................................................................ vii

Acknowledgment .......................................................................................................... viii

Table of Contents ......................................................................................................... viii

List of Equations ........................................................................................................... xi

List of Tables ................................................................................................................ xiii

List of Figures ................................................................................................................ xiv

List of Abbreviations .................................................................................................... xv

Chapter 1 Introduction .................................................................................................. 2

1.1 Statement of the Problem ....................................................................................... 4

1.2 Objectives ............................................................................................................... 5

1.2.1 Main Objectives ................................................................................................. 5

1.2.2 Specific Objectives ............................................................................................ 5

1.3 Importance of the Research .................................................................................. 5

1.4 Scope and Limitation of the Research .................................................................. 6

1.4.1 Scopes ................................................................................................................ 6

1.4.2 Limitations .......................................................................................................... 6

Chapter 2 Background ................................................................................................ 8

2.1 Raspberry PI .......................................................................................................... 8

2.2 Image Forgery Detection Techniques ..................................................................... 10
2.2.1 Pixel-based image forgery detection ........................................... 110
2.2.2 Format-based image forgery detection ....................................... 11
2.2.3 Camera-based image forgery detection ...................................... 11
2.2.4 Physical environment-based image forgery detection .................. 12
2.2.5 Geometry-based image forgery detection ................................... 12
2.3 OpenCV-Python Library .................................................................. 13
2.3.1 OpenCV .................................................................................. 13
2.3.2 OpenCV-Python ...................................................................... 14

Chapter 3 Related Works ....................................................................... 16
3.1 Video Forgery Detection ................................................................. 16
3.2 Video Tempering Indicators ............................................................ 18
3.3 Summary ...................................................................................... 19

Chapter 4 Methodology ....................................................................... 21
4.1 Introduction .................................................................................. 21
4.2 Configure Detector Streaming System: ........................................... 21
4.3 The proposed method: ................................................................. 25
  4.3.1 Embedding stage: ................................................................. 26
  4.3.2 Distortion/ Tempering Stage: .................................................. 29
  4.3.3 Detection/ Retrieval Stage: ..................................................... 30
4.4 Summery: ................................................................................... 32

Chapter 5 Experiments ....................................................................... 34
5.1 Introduction .................................................................................. 34
Chapter 5

5.2 Dataset .............................................................................................................. 34
5.3 Experiment Settings .......................................................................................... 37
5.4 Experiments Process ......................................................................................... 38
  5.4.1 Experiment: Making Duplication Forgeries .............................................. 38
  5.4.2 Experiment: Making Cloning (copy-move) Forgeries ......................... 44
5.5 Summary ............................................................................................................. 47

Chapter 6

Chapter 6 Result discussion ...................................................................................... 50
  6.1 Introduction ........................................................................................................ 50
  6.2 Explanation of the Idea ..................................................................................... 50
  6.3 Results ................................................................................................................. 55
    6.3.1 Result Discussion for Duplication: ............................................................. 55
    6.3.2 Result Discussion for Cloning: ................................................................. 61
  6.4 Summary: ........................................................................................................... 63

Chapter 7

Chapter 7 Conclusion and Future Works ................................................................. 65
REFERENCES: ......................................................................................................... 67
List of Equations

Equation 4.1: Average of RGB Pixel................................................................. 24
Equation 5.1: Time needed to upload frames ..................................................... 40
Equation 6.1: Accuracy of detector streaming system ....................................... 55
List of Tables

Table 4.1: Comparison between MPEG-4 and H.264 Features .................................................. 23
Table 4.2: Parameters of 36 watermarks that saved on excel file ............................................. 29
Table 5.1: Detector streaming excel file .................................................................................... 35
Table 5.2: Original videos properties ....................................................................................... 35
Table 5.3: Describe original video properties .......................................................................... 36
Table 5.4: Machine specifications ........................................................................................... 37
Table 6.1: Depicts the value of watermark number 35 on different compression format for the same video on Exp1 .................................................................................. 51
Table 6.2: Depicts the value of watermark number 18 on different compression format for the same video on Exp2 .................................................................................. 52
Table 6.3: Depicts the value of watermark number 7 on different compression format for the same video on Exp 3 .................................................................................. 52
Table 6.4: Depicts the datum of experiment 1 and show the false positive of these video ................................................................................................................................. 55
Table 6.5: Depicts the datum of forged portion that used to carry out duplication. .......................... 56
Table 6.6: Depicts various versions of forged videos that have duplication forgery and its accuracy .......................................................................................................................... 56
Table 6.7: Depicts the datum of experiment 2 and show the false positive of these video ................................................................................................................................. 57
Table 6.8: Depicts the datum of forged portion that used to carry out duplication. .......................... 57
Table 6.9: Depicts various versions of forged videos that have duplication forgery and its accuracy .......................................................................................................................... 58
Table 6.10: Depicts the datum of experiment 3 and show the false positive of these video ................................................................................................................................. 59
Table 6.11: Depicts the datum of forged portion that used to carry out duplication. .......................... 59
Table 6.12: Depicts various versions of forged videos that have duplication forgery and its accuracy .......................................................................................................................... 60
Table 6.13: Depicts the datum of forged portion that used to carry out cloning. ............................. 61
Table 6.14: Depicts various versions of forged videos that have cloning forgery and its accuracy .......................................................................................................................... 61
Table 6.15: Depicts the datum of forged portion that used to carry out cloning. ............................. 62
Table 6.16: Depicts various versions of forged videos that have cloning forgery and its accuracy. ................................................................. 62
Table 6.17: Depicts the datum of forged portion that used to carry out cloning. ...... 62
Table 6.18: Depicts various versions of forged videos that have cloning forgery and its accuracy. ............................................................................................................. 63
List of Figures

Figure 2.1: Raspberry pi device ................................................................. 8
Figure 2.2: categories of image forgery detection ...................................... 10
Figure 2.3: pixel based forgery detection .................................................... 11
Figure 2.4: format based forgery detection ................................................... 11
Figure 2.5: format based forgery detection ................................................... 12
Figure 2.6: physical environment based forgery detection ....................... 12
Figure 2.7: geometry environment based forgery detection ..................... 13
Figure 4.1: Proposed methodology of detector streaming system. ............. 22
Figure 4.2: Watermark embedding algorithm ................................................. 23
Figure 4.3: Digital watermarking detector stages ...................................... 24
Figure 4.4: Watermark Detection Process .................................................... 25
Figure 4.5: Embedding process of detector streaming system ................ 26
Figure 4.6: Hidden watermarks distribution on frame ............................... 27
Figure 4.7: Flow chart for process of the detector system in detecting forgeries ..... 31
Figure 5.1: VideoPad interface and timeline ............................................. 38
Figure 5.2: Making duplication in VideoPad timeline first experiment .......... 40
Figure 5.3: Image before and after duplication forgery .............................. 41
Figure 5.4: Making duplication in VideoPad timeline on second experiment ... 42
Figure 5.5: Image before and after duplication forgery .............................. 43
Figure 5.6: Making duplication and cloning in VideoPad timeline on third experiment ................................. 44
Figure 5.7: Making cloning in VideoPad timeline on first experiment .......... 45
Figure 5.8: Image before and after cloning forgery .................................... 46
Figure 5.9: Image before and after cloning forgery .................................... 47
Figure 6.1: Watermarks 10, 20, 30 for original video exp2 .......................... 54
Figure 6.2: Watermarks 10, 20, 30 for Forged Lossless exp2 ...................... 54
List of Abbreviations

**IP camera**  An Internet Protocol camera is a type of digital video camera commonly employed for surveillance.

**CCTV**  (closed-circuit television) is a TV system in which signals are not publicly distributed but are monitored, primarily for surveillance and security purposes.

**ARP spoofing**  Address resolution protocol (ARP) spoofing is a technique that causes the redirection of network traffic to a hacker.

**CDIS**  Cross Domain Image Streaming.

**HTTP**  Hypertext Transport Protocol.

**DVR**  (Digital Video Recorder) A device that records video from up to a dozen or more surveillance cameras onto a hard disk.

**Raspberry PI**  is a series of small single-board computers.

**MPEG**  is a digital multimedia container format most commonly used to store video and audio.

**HOG**  histogram of oriented gradients algorithm.

**SIFT**  The scale-invariant feature transform algorithm.
Chapter 1

Introduction
Chapter 1
Introduction

Forgery is the process of making adapting or imitating objects. Forgeries are not new to mankind but are a very old problem. In the past, it was limited to art and literature but did not affect the general public. Nowadays, due to the advancement of digital video processing software and editing tools, a video can be easily manipulated and modified (Redi, Taktak, and Dugelay (2011)). This is mainly due to the availability of low-cost hardware and photo editing software which makes it easy to manipulate and alter digital videos without leaving any obvious trace. Therefore there is a rapid increase in digitally manipulated forgeries in mainstream media and on the Internet (J. Wang, Liu, Zhang, Dai, & Wang, 2009), in addition, the huge amount of possible alterations operated on it. Nowadays, surveillance systems are used to control crimes. Therefore, the authenticity of digital video increases the accuracy of deciding to admit the digital video as legal evidence or not. Inter-frame duplication forgery is the most common type of video forgery methods. However, many existing methods have been proposed for detecting this type of forgery and these methods require high computational time and impractical(Fadl, Han, & Li, 2018).

On the other hand, over the last several years IP security cameras have gained wide acceptance for business and home use. IP surveillance cameras are used typically for surveillance purposes in organizations and public places. IP surveillance cameras play a critical role in providing evidence against crimes and unlawful activities. The surveillance technologies have become mature and sophisticated with the passage of time. The IP surveillance cameras are serving as an important defensive tool in today’s environment, there is no doubt, IP surveillance cameras can serve as one of the best defense against crime. But, IP surveillance cameras are also prone to security vulnerabilities. In the last couple of years, we have noticed an explicit set of network attacks(Hacking CCTV’s) on CCTV cameras such as injecting video streams through ARP spoofing, another form of hacking which is cross-domain image streaming (CDIS), which is an attacker exploits the web
interface of IP surveillance camera to render another fake or non-legitimate image stream from third-party domain in the context of legitimate domain where the IP surveillance camera is hosted, this can happen by not validating the input that read image streams from camera. And don’t validate the content type of HTTP header in order to understand the images that are rendered by the running server.

In general the video is a series of images, the forgery can take place on image level, image forgery detection is classified as active and passive. In the active approach, the digital image requires preprocessing of image such as watermark embedding or signature generation, which limits their application in practice (Tyagi, 2010), on the other hand, the passive techniques do not need any digital signature to be generated or to embed any watermark. One of the common used image forgery detection techniques is pixel based forgery detection technique; Pixel-based techniques detect statistical anomalies introduced at the pixel level, that can be classified into four categories. Resampling technique which means resize, stretch or rotate the image; splicing technique that splicing two or more images into one composite image; statistical that base on using statistic techniques on pixel level to detect the tempering; the last type is cloning (copy-paste) which is most commonly used, The copy-paste image forgeries can be classified into two different categories, spatial tampering, and temporal tampering. In spatial tampering, a region may be copied from a location in a frame and pasted to a different location on the same frame or other frames possibly after some modification. While in temporal tampering, a complete frame may be duplicated. In addition, a region may also be duplicated across the frames at the same spatial location (Subramanyam & Emmanuel, 2012). Since an action or object may be occurring in multiple frames, in order to create a convincing forgery the temporal tampering.

Many researches have been conducted for forgery detection related to the image, however, there is only few image forgery duplication detections works among the image copy paste forgery detection system, Furthermore, those researches proposed methods that require a huge amount of storage to detect image forgery.
In this research we proposed a new method; this method is designed to be integrated in surveillance cameras system, in order to detect the duplication in live video and offline video stream by using a random watermark (pixel), in specific slice of the image, so we can compare these slice of image series not whole the image to ensure that it is not forged regions in the specific area with an efficient and effective way. The aim of our method is protect private and sensitive Areas that have important streaming content; these areas should have quality security cameras software to detect image series forgery on the live videos, on an effective and fast way to prevent crime by identifying potential criminal activity and help responsible to respond more quickly to incidents.

Our methodology conducted by three steps to makes this process more efficient first step: collecting the watermarked video streams from IP surveillance camera, second step: Making forgeries on the given videos from step one, third step: detecting the forged video and the forged part on the given video from the second step. In this research, we will discuss the process with details, results ,and evaluation with all measurements taken for videos that have forgeries or affected by compression format and videos that haven’t any forgeries to approve the accuracy of our detector system.

As far as we aware of, this is the first effort that aims to offer detector forgeries system on video streaming, the proposed system is expected to act as a new version of DVR system that can explore different forgeries type on video. The detector streaming system is evaluated over meaningful videos from recorded live streaming camera, the result indicated that the system achieved high relevant measures on detecting duplication snd cloning forgeries.

1.1 Statement of the Problem

Due to easy availability of advanced tools that make it possible to manipulate and alter the digital image integrity especially on videos captured by surveillance system to hide crime, one of the famous way of hacking the integrity of the image is make forgery of the image in video stream to hide any object from the video (Inter-frame duplication). On the other hand the process of detecting image forgery in video requires huge storage and consequently is time-consuming (CPU
process), furthermore many proposed method was set to detect forgeries on offline or recorded videos, but it needs many resources and executing time.

1.2 Objectives

1.2.1 Main Objectives

In our research, we propose a new method to detect forgeries in image series that coming from live videos and stored video that improve the process on a term of efficiencies like process speed and accuracy.

The main objective of this research is providing new efficient and fast method for video forgeries detection that reduces the amount of process require to detect forgeries by using a hidden watermarked matrix that spread on video frames and define the start and end of the forged part on the concerned videos.

1.2.2 Specific Objectives

The specific objectives of this research are:

- Use the best ways of image processing techniques.
- Develop a program to handle the movement detection sensor camera process.
- Collect data from different places which are recorded video and live video.
- Apply our idea of detecting forgeries on videos by spreading the hidden matrix of watermarks on video frames.
- Make experiment to asses our proposed method on a term of efficiency and accuracy by conducting tests on a number of videos.
- Validate videos from duplication and cloning forgeries.

1.3 Importance of the Research

Because of the wide availability of powerful image processing tools has made imperceptible image modifications possible, as a result protecting integrity and authenticity becomes very important needs especially on videos that captured by surveillance systems, not only but also need a method for video forgeries detection
that explore any change on the data stream and define the effect types that carried out on the video file like duplication and cloning effects or normal compression change, not only but also it helps to define start and end forgery frame series and get the origin frames if required.

1.4 Scope and Limitation of the Research

1.4.1 Scopes
- Compression algorithm has been used to compress and store video over the network bandwidth to increase the recording time.
- 36 hidden pixels or watermarks have been used to be inserted into each frame reads from the streaming to be compared with the forged videos to calculate the efficiency of our detector streaming.
- Excel file has been used to save the position and the value of each watermark embedded in the captured frame to be used in detecting forgery in the recorded video file.

1.4.2 Limitations
- Experiments have been conducted on videos that captured from existing IP camera in the same network, not from another network, and the forgeries effect carried out after recording the videos.
- Experiments on this thesis explain only temporal duplication and cloning effects.
- Experiments will be conducted by using 27 videos that captured from IP surveillance camera; the forgeries have been carried out on the video after recording.
- Another detector streaming system version has been developed to detect the online streaming attacks but it didn’t test by real attacks.
- We will use sensor camera movement detection this sensor will work only when detecting movement in order to explore the origin forged frames.
- Testing will be carried out on the watermarked videos that captured on different places and there compressed video versions and forged compressed versions.
Chapter 2

Background
Chapter 2

Background

2.1 Raspberry PI

In our proposed method we will use Raspberry PI device as specified in figure 2.1, this device is a camera sensor for movement detection, to start capturing video streams when detect any moving object in order to reduce the storage needed to complete our process.

![Raspberry Pi Device](image)

**Figure 2.1**: Raspberry pi device

**Raspberry pi features**

- Assembled Pi Cobbler Plus - The best way to break out your Pi's pins into a breakout board. We also carry the assembled T-Cobbler Plus that makes your breakout pins a bit easier to read but is less compact.
- Adafruit Pi Box Plus for the Raspberry Pi Model A+ - this is a great case to keep your Raspberry Pi Model A+ safe and sound. We also carry the Pi Protector for Model A+ - our pocket-sized Pi Case.
- 8GB Card with NOOBS 1.39 - This is the fastest way to have a variety of operating systems on your pi.
- Bare 4 GB SD card - you'll need this to load the Raspbian/Debian image and is not included in a Pi computer purchase. Tested and works perfectly! We also have a wide range of pre-burned SD cards
- 5V @ 2A Switching Power Supply w/ 6’ MicroUSB Cable - An all-in-one 5V 2A + MicroUSB cable power adapter that's perfect for powering your pi
- HDMI cable - required to connect a Pi to a monitor and is not included in a Pi computer purchase. We also carry the flat version for a more sleek/stylish look.
- Ethernet cable - required if you want to connect your Pi to the Internet via a router
- USB WiFi Module - This is our recommended WiFi adapter, works out of the box with Pi with good range and speed
- Keyboards and Mice - We have a range of wired and wireless options
- A wide range of accessories and essentials
- Various small displays as well as nice Pi-compatible HDMI monitors in a range of sizes, from 2.5" to 10"

Technical details:

- Dimensions: 65mm x 56mm / 2.5" x 2.25"
- Processor: ARM 700Mhz CPU
- USB: One USB 2.0 Port - Keeping it compact means minifying the footprint as much as possible
- Video Output: HDMI and Composite
- Sound: L/R Stereo (via 3.5mm 4 Pole Cable)
- Memory: 512MB Ram
- Operating System: Uses microSD card slot to load O/S
- Digital interfaces: 2 onboard ribbon slots for Camera and Display
- GPIO: 40 GPIO pins, backwards compatible to B board
- No Ethernet jack
## 2.2 Image Forgery Detection Techniques

Digital image forgery detection techniques are classified into active and passive approaches. In the active approach, the digital image requires preprocessing of image such as watermark embedding or signature generation, which limits their application in practice. Active techniques such as digital watermarking (Khan et al., 2012; Shih, 2017) and digital signature (Lu & Liao, 2003; Tzeng & Tsai, 2001). The passive techniques do not need any digital signature to be generated or to embed any watermark. Passive image forgery detection techniques (Asghar, Habib, & Hussain, 2017; Qazi et al., 2013) can be divided into five categories as shown in Figure 2.2. Pixel-based techniques detect statistical anomalies introduced at the pixel level; format-based techniques leverage the statistical correlations introduced by a specific lossy compression scheme; camera-based techniques exploit artifacts introduced by the camera lens, sensor, or on-chip post-processing; physical environment-based techniques explicitly model and detect anomalies in the three-dimensional interaction between physical objects, light, and the camera; and geometry-based techniques make measurements of objects in the world and their positions relative to the camera.

![Diagram of categories of image forgery detection](image)

**Figure 2.2:** categories of image forgery detection

### 2.2.1 Pixel-based image forgery detection

Pixel-based techniques emphasize on the pixels of the digital image. These techniques are roughly categorized into four types. We are focusing only two types of techniques copy-move and splicing in this research. This is one of the most common forgery detection techniques. Figure 2.3 shows categorization of pixel based forgery detection techniques.
2.2.2 **Format-based image forgery detection**

Format based techniques are another type of image forgery detection techniques. These are based on image formats and work mainly in the JPEG format. These techniques can be divided into three types (Figure 2.4). If the image is compressed then it is very difficult to detect forgery but these techniques can detect forgery in the compressed image.

![Format Based Image Forgery Detection Diagram](image)

**Figure 2.4:** format based forgery detection

2.2.3 **Camera-based image forgery detection**

Whenever we capture an image from a digital camera, the image moves from the camera sensor to the memory and it undergoes a series of processing steps, including quantization, color correlation, gamma correction, white balancing, filtering, and JPEG compression. These processing steps from capturing to saving the image in the memory may vary on the basis of camera model and camera artifacts. These techniques work on this principle. These techniques can be divided into four categories as shown in Figure 2.5.

![Camera Based Image Forgery Detection Diagram](image)

**Figure 2.5:** camera based forgery detection
2.2.4 Physical environment-based image forgery detection
Consider the creation of a forgery showing two movie stars, rumored to be romantically involved, walking down a sunset beach. Such an image might be created by splicing together individual images of each movie star. In so doing, it is often difficult to exactly match the lighting effects under which each person was originally photographed. Differences in lighting across an image can then be used as evidence of tampering. These techniques work on the basis of the lighting environment under which an object or image is captured. Lighting is very important for capturing an image. These techniques are divided into three categories as shown in Figure 2.6.

2.2.5 Geometry-based image forgery detection
Grooves made in gun barrels impart a spin onto the projectile for increased accuracy and range. These grooves introduce somewhat distinct markings to the bullet fired, and can therefore be used to link a bullet with a specific handgun. In the same spirit, several image forensic techniques have been developed that specifically model artifacts introduced by various stages of the imaging process. Geometry-based
techniques make measurement of objects in the world and their position relative to the camera. Geometry-based image forgery techniques are divided into two categories Figure 2.7.

![Geometry Based Image Forgery Detection](image)

**Figure 2.7:** geometry environment based forgery detection

### 2.3 OpenCV-Python Library

This is the library that has been used to develop and apply the proposed detector streaming system.

#### 2.3.1 OpenCV

OpenCV was started at Intel in 1999 by Gary Bradsky and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel’s Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle who won 2005 DARPA Grand Challenge. Later its active development continued under the support of Willow Garage, with Gary Bradsky and Vadim Pisarevsky leading the project. Right now, OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day.

Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations.

OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language.
2.3.2 OpenCV-Python

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works; it is a Python wrapper around original C++ implementation.
Chapter 3

Related Works
Chapter 3
Related Works

In the previous research, many approaches have been proposed to detect forgery in video/image, here we list the most related to our proposed method:

3.1 Video Forgery Detection

Mathews et al. (Mathews & Sreedharan, 1 June 2015) proposed a new algorithm for detecting copy-move forgery in videos for both spatial and temporal tampering. The Correlation coefficient is used as a measure of similarity. The proposed method based on spatial correlation calculation, and frame duplication classification. This method based on the discontinuity in the optical flow variation sequence can be used for detecting forgeries in videos. The limitation of this technique is that it is only effective when the second compression quality is higher than the first compression quality. Also, the computational cost of the algorithm is the drawback.

Wan Wang et al. (Wan Wang, Jiang, Wang, Wan, & Sun, 2013) proposed a method to identify inter-frame forgery process in surveillance video, they investigate the optical flow extracted from frame series, the optical flows change almost continuously in the original videos; they find when forgery process occurred will introduce detectable discontinuity points, which have unique characteristics depending on the type of forgery. They calculated the optical flow variation sequence and adopt anomaly detection schema to locate discontinuity points to explore the different type of forgeries. They used a small forgery dataset of two original videos with 3000 frames. Also, the optical flow estimation method needs some improvement on a case the forgery less correlated to normal changes in original videos. Moreover, the frames after forgery were re-encoded with the same coding standard (MPEG-2).
Weihong Wang et al. (Weihong Wang & Farid, 2007) have described two techniques for detecting a common form of tampering in video. The first technique detects entire frame duplication and the second detects if only a portion of one or more frames was duplicated. In each proposed techniques, the proposed approach is to partition a full-length video sequence into short overlapping sub-sequences. A compact and efficient to compute representation that embodies both the temporal and spatial correlations in each sub-sequence is then extracted and compared throughout the entire video. Similarities in the temporal and spatial correlations are then used as evidence of duplication. Throughout they proposed to use the correlation coefficient as a measure of similarity. They used small forgery dataset two original videos with 10000 frames one from a rarely movement place and the second from little movement places, these videos were subjected to various forms of duplication. Also the run-time of the proposed method requires 45 minutes for processing a 10,000 frame sequence. The algorithm have issue of computational cost since even a video of modest length can run into thousands of frames; they didn’t take care about the huge storage that they will use to accomplish their works.

Randeep Kaur et al. (Randeep Kaur, December 2016) detected the editing digital multimedia content by using some technique like optical flow to detect the flow of the moving objects and the forgery object, to verify the video they used Invariant Feature Transforms (SIFT); It displays the number of keypoints extracted from input image, to detect the matches’ keypoints consecutantly the forged part on the image. The major improvement in this work is to detect the forgery part with the help of Key point features and the optical flow algorithm.

Subramanyam et al. (Subramanyam & Emmanuel, 2012) proposed algorithm for copy-paste forgery detection is based on Histogram of Oriented Gradients (HOG) feature matching and video compression properties. The benefit of using HOG features is that they are robust against various signal processing manipulations, the parameter cellsize of the Hog feature generation is set adaptively to reduce the false positive rate and increase the detection accuracy for spatial forgery detection. The
frames with high correlation for duplicated regions compared to authentic regions are selected for detection purpose.

Mehdi Fallahpour et al. (Fallahpour, Shirmohammadi, Semsarzadeh, & Zhao, 2014) proposed a practical system of digital video watermarking that suggested for authenticating and tampering detection of compressed videos, the embedding and extracting of watermarks are integrated with the coding and decoding routines of the video codec. To assure transparency to the human visual system, Furthermore, they distinguish malicious attacks from common video processing operations, such as H.264/AVC recompression, noise, and brightness increasing, analysis of the error is used to detect tampering, they use the watermark signals, watermark signal on his concept represent the macroblock’s and frame’s indices, and are embedded into the nonzero quantized discrete cosine transform value of blocks, mostly the last nonzero values, enabling our method to detect spatial, temporal, and spatiotemporal tampering, this practical system is targeted only for the offline video not on the live recording video in addition they also detect different type of image forgery detection except duplication.

3.2 Video Tempering Indicators

Kim et. al. (Kim & Han, 2012) developed a security model to ensure a safe and secure operation of an intelligent VSS. The model is represented by a set of particularly desired security functions which are grouped into several groups: video gathering group, video storage group, video control group, video application group. In their model, authors relate each identified security threat with particular security functions within defined groups, but their proposed model require time processing because of the huge debugging function and data.

Coole et. al. (Coole, Woodward, & Valli, 2012) described a subset of the security issues related to networked, and especially Wi-Fi based, surveillance devices. The authors also discuss the significance of vulnerability exploitation of such devices in the context of confidentiality, integrity, availability. They conclude
with a framework for implementing controls to reduce risk associated with Wi-Fi based CCTV systems.

Obermaier and Hutle (Obermaier & Hutle, 2016) provided a practical analysis of the security and privacy of four major cloud-based video surveillance systems. They reverse-engineered the security implementation and discovered several vulnerabilities in every of the tested systems. The authors considered two attacker models, namely local network attacker and remote network attacker. They demonstrated how these attackers can exploit VSS vulnerabilities to blackmail users and companies by DoS attacks, by injecting forged video streams, and by eavesdropping private video data, even without physical access to the systems. Their main findings, however, relate to classical weaknesses such as fallback to unsecured function, proprietary security protocols, weak passwords, and insecure authentication.

### 3.3 Summary

This chapter presents the related works to our proposed technique, which presented different type of digital image/videos forgeries detection techniques that classified into active and passive techniques; active technique requires preprocessing to image which limited to recorded videos, passive techniques don’t require preprocessing but it need much resources and execution time to apply these techniques.

Many researches have been presented to passive technique they used optical flow and HOG algorithm that already used to detect moving objects to measure the difference luminances between original and forged image, and show that they have perfect accuracy on detecting temporal and spatial forgeries. Moreover, they proposed techniques that need much resources and execution time to be configured, not like our proposed lightweight method that can detect spatio-tempering forgery on time of capturing video by injecting forged video streams or captured watermarked videos with perfect accuracy and less detecting runtime.
Chapter 4
Methodology
Chapter 4

Methodology

4.1 Introduction

This chapter presents the algorithm of detector streaming system that utilizes Active/intrusive/non-blind method for detecting different type of forgeries. It clarifies the detailed steps of detector streaming system which include: embedding/tempering stages. Second, used motion detection algorithm. Third, detection streaming forgeries stages. And finally, a case study is presented to view the functional steps of detector streaming system process.

Detector streaming system can detect many types of cloning techniques; the most popular and common photo tampering technique is the copy-move (cloning) technique because it’s easy to happen, it carried out by copying of some region in an image and replaces it to another region in the image. Since the copied region has been copied from the same image so the color of the remains region is compatible with the rest of the image, therefore this system detects more technique like duplication forgery that duplicates some image copies of such image instead of some sequence frames to hide an object from the streaming.

4.2 Configure Detector Streaming System

This section briefly explains the configuration needed for our forgery detector streaming system. The description of the system is depicted in (Figure 4.1).

Duplication forgery, can be used to remove people or objects from a video sequence or simply remove an undesired event from a video, for example, portion of frames on certain videos are duplicated to remove undesired objects or events, This type of manipulation is fairly easy to perform and can be difficult to detect visually particularly in a video taken from an IP surveillance camera, and second one is detecting the cloning (copy-move) forgery, in this type the attacker uses portion of the original base image as its source. Therefore, the source and the destination of the modified image originate from the same image, to hide some part of such frame in a video.
Figure 4.1: Proposed methodology of detector streaming system.

Our methodology has been designed for using in IP cameras and network video recorders (NVR) as shown in the figure above, for sure it should increase the number of days video recording, for this the top consideration is to use compression algorithm to compress and store video over the network bandwidth in NVR, that’s mean, IP camera video compression is performed inside the camera then transmitted to the NVR in the compressed format.

There are two groups of compression technologies currently used in video surveillance: frame-by-frame encoding (MPEG-4) and temporal encoding. In our method we use (H.264) temporal compression, to compress the video file to increase the streaming time on hand and decrease the video file size on the other hand. This depends on the network bandwidth. The following table Table 4.1 shows comparison between these two compression algorithms that mentioned above. Without some type of compression, most images would be too cumbersome and impractical for use.

Watermark embedding algorithm has been applied to the compressed frames captured by the camera, to provide identity to the continues captured frame, on the other hand these watermarked frames have been passed through motion detection algorithm to check if there are movement detection in the continuous video stream or not, on a positive case these frames should be sending to the cloud, these live frames captured only when detects movement in such area of object to save the original frame.
Table 4.1: Comparison between MPEG-4 and H.264 Features

<table>
<thead>
<tr>
<th>MPEG-4</th>
<th>H.264</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame-by-Frame compression technologies.</td>
<td>Temporal Compression technologies.</td>
</tr>
<tr>
<td>Intra-frame, compression technologies compress video by applying a compression algorithm to each frame captured by a camera.</td>
<td>Temporal compression technologies rely both on compressing data within a single frame and on analyzing changes between frames</td>
</tr>
<tr>
<td>The end result is a series of individually compressed images.</td>
<td>The result is a stream of video that is compressed over multiple frames rather than a series of individual frames.</td>
</tr>
</tbody>
</table>

After passing these frames into the motion detection algorithm, excel file has been generated to save the position and the value of each watermark embedded in the captured frame to be used in detecting forgery in the recorded video file.

In our proposed methodology two types of digital image forgery have been detected using digital watermarking method which is an application of the digital image processing, we make preprocessing for the digital image series by embedding watermarks on each image to verify the integrity of live video captured by IP surveillance camera. The proposed digital watermarking method include two algorithms: one as embedding algorithm and the other as detecting algorithm like other watermarking methods, Figure 4.2 show watermark embedding algorithm.

![Image of Watermark Embedding Algorithm](image)

**Figure 4.2:** Watermark embedding algorithm
Generally in embedding stage, we use a predefined pixels value to be embedded into each frame captured by the module camera on the region of (640,480) as the watermark, the embedded pixels which are used to identify the authenticity of the original image in video streams. We develop this pixel to be as much as invisible by the unaided eye.

Suppose we want to embed hidden pixel in a specific location represented in \( P(x,y) \), where \( P(x,y) = (\text{red}, \text{green}, \text{blue}) \), we use equation 4.1 to calculate the average in the colored frame.

**Equation 4.1:** Average of RGB Pixel

\[
\text{Average} = \sqrt[3]{\left(\sqrt{r_0 + r_1 + r_2}\right)^2 + \left(\sqrt{g_0 + g_1 + g_2}\right)^2 + \left(\sqrt{b_0 + b_1 + b_2}\right)^2}
\]

for each captured frame \( F(R,G,B) \), represent the new pixel color average value, that we insert as a hidden pixel, and the value of \( R_0,G_0,B_0 \) represented respectively as the following sequence \( z_n = z_{n-1} + 1 \) where \( n = 0 \) and \( n \leq 255 \), \( R_1,G_1,B_1 \) represent the pixel value on the left and \( R_2,G_2,B_2 \) represent the pixel value on the right.

Digital watermarking method is used to embedding hidden information into multimedia data (Mathews & Sreedharan, 1 June 2015), that information usually hidden and need detector or extractor to show and extracts that information. Digital image watermarking use digital image for embedding the hidden information, after embedding the watermarked image is generated and the watermarked image is more robust against attacks. Figure 4.3 show three stages of digital watermarking detector happen.

![Figure 4.3: Digital watermarking detector stages](image)

24
On detecting or extracting method, inverse digital watermarking method is used to read the registered video frame by frame, and compare the embedded watermarks position and their values with the origin value that saved on the excel file sheets that generated on the embedding stage. Figure 4.4 show the watermark detection process in which the embedded watermark will be verify by using the detection algorithm to show the forged part or forged frames.

![Watermark Detection Process](image)

**Figure 4.4: Watermark Detection Process**

### 4.3 The proposed method

In the proposed method, the streaming is passed into three stages to detect different types of forgeries, stage one which is embedding stage that make preprocessing for the streaming and apply the proposed method, stage two which is tempering stage, on it different type of forgeries and cases are applied to the saved streaming to be detected in the third stage which is detecting forgery process. The following flow chart illustrated in Figure 4.5 shows in details all the embedding process of detector system.
**4.3.1 Embedding stage**

In this stage as shown in Figure 4.5, the image pass through different steps, in the first step while the streaming capture is open, loop is used to capture frame by frame to insert the watermarks in the second step.

36 pixels or watermarks are inserted into each frame reads from the streaming loop, we divide the 36 pixels into two sub group of 20 pixels, the first group will shifting up by 5 pixels on each frame reads and we assign the average of each these pixels in the frame using the Equation 4.1 mentioned above, the same for the second group, 16 pixels but shifting down with average assigning to these pixels to generate the embedded frame.

The distribution of these watermarks is designed to capture as much as forgery cases, the following Figure 4.6 describe how the distribution of these watermark in each frame that positioned in loop:
After reaching the last captured frame on the loop, the embedding algorithm will start assigning the watermarks again that showing in the first matrix. The result
of this step is embedded watermarked frame that passed to the movement detection approach, to be known, the number of frame per second is fixed to 25 because when streaming for security purpose on case we have movement detection, to ameliorate the shown streaming and to be more clear to the reviewer on case of movement detection that expected to be tempered, but on the opposite case, it’s sufficient to opt for a frame rate of 10 FPS then save some networking bandwidth during nighttime for example and to increase the streaming time.

Most common motion detection approaches are to compare the current frame with the previous one. To estimate changes and to write only the difference, The embedded watermarked frame called current frame a gray scale copy of it is generated to be compared with the previous image. after that the algorithm should find the regions where these two frames are difference a bit, for this purpose thresholds filters are used.

Most cameras produce a noisy image, so we’ll get motion in such places, where there is no motion at all. To remove random noisy pixels, we can use an Erosion filter, for example. So, we’ll get now mostly only the regions where the actual motion was.

On this step we’ll get an image with white pixels on the place where the current frame is different from the previous frame on the specified threshold value. It’s already possible to count the different pixels, and if the amount of it will be greater than 1000 pixels a predefined alarm level we can signal about a motion event. Now the motion detector is ready, the motion regions are highlighted. If the motion detector found difference between compared frames more than 1000 pixels it will write the origin image and send it by threading to the cloud in our experiment Drop Box is used. Furthermore, if the detector didn’t found any changes it continue without sending to drop box, on all the cases the captured images are saved to the streaming before showing to the windows.

Before writing the image into the streaming file, the values of the predefined pixel positions of each frame captured are read, all pixels or watermarks information’s saved into excel files that have 36 sheets, each sheet contains the watermark information separately that shown below in Table 4.2, these information are used in detection stage later. Note that maximum total number of rows and
columns on a worksheet is 1,048,576 rows by 16,384 columns. For this Detector streaming system generate every 12 hours new excel file to save it’s watermarks information’s, because of the worst case on the streaming of our system is recording movement for 12 hours, We propose that we have continues movement for the 12 hours on case of movement detection the streaming camera will send 25 image per second, so (25FPS*60second *60minutes *12hours) will equal 1,080,000 approximately equals to 1,048,000.

Table 4.2: Parameters of 36 watermarks that saved on excel file

<table>
<thead>
<tr>
<th>X axis</th>
<th>Row Coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y Axis</td>
<td>Column Coordinate</td>
</tr>
<tr>
<td>Total = Red +Green +Blue</td>
<td>Summation of RGB</td>
</tr>
<tr>
<td>Frame number</td>
<td>The frame number that have given pixel</td>
</tr>
</tbody>
</table>

You can access a pixel value by its row and column coordinates. For RGB image, it returns an array of Blue, Green, Red values. The array of numbers represents the image pixel average as a series of numbers that calculated using Equation 4.1. Notice that the numbers are arranged as sets of three values, each ranging from 0 to 255. Each set of three represents the color of an individual pixel, as expressed in blue, green, and red values, respectively.

4.3.2 Distortion/ Tempering Stage

Video data has become more popular with the advancement of digital cameras and networking technologies with high speed bandwidths, in this context, the image editing software tools increased day by day leading to the forgery of digital images. There is much software available all over the internet that facilitates video editing. With these resources, video editing has become increasingly easier to alter video stream within minutes. This can introduce many security concerns. So detecting video forgery has become a critical requirement to ensure integrity of video data.
There are too many free video editing programs available on the market like (Avidemux 2.7, Movavi Video Editor 14, VideoPad Video Editor, HitFilm express 2017). The following two paragraphs briefly describe the two types of forgeries, in our proposal VideoPad tool has been used in this stage to apply the two type of forgeries.

4.3.2.1 Cloning

Cloning detection to clone or copy and paste a part of the image to conceal an object or person is one of the most commonly used image manipulation techniques. When it is done with care, it becomes almost impossible to detect the clone visually and since the cloned region can be of any shape and size and can be located anywhere in the image, it is not computationally possible to make an exhaustive search of all sizes to all possible image locations. According to (Gavade1 & Chougule2, 2015).

4.3.2.2 Duplication

Duplication forgeries carried out by replacing sequence of images especially in time of moving, by copies of the same of any image in the series that haven’t any motion on it. To hide any object from that video series.

4.3.3 Detection/ Retrieval Stage

Detector streaming system in the first inform that there are streaming editing that happen, in which frame and detect the type of forgery, the following two paragraphs describe how detector system determine the forgeries and the according type of forgeries. The following flow chart Figure 4.7, illustrates in details all the process that used in our detector streaming system to find forgeries in the saved video file.

Detector streaming system trait this type of forgery, by check if there are some regions in the video series have watermarks value different to the saved value plus 5, moreover if the number of different watermarks is equal to 30, so the type of forgery is consider as duplication forgery, if the differance number of watermarks more than
15 so the type consider cloning forgery, because the concerned person will make the duplication with the whole image not a part of the image like copy-move forgery.

Figure4.7: Flow chart for process of the detector system in detecting forgeries

In detection stage video files and the watermarks information files are saved to the recipient, these file are considered as digital signature to demonstrate the authenticity of the saved video files, Some operations can be manipulated on the image like scaling, rotating, blurring, brightness adjusting, change in contrast, etc. or combination of these operations are performed on an image.

In the proposed system, prior information about the image is indispensable to the process of authentication or detection saved in excels files. It is concerned with watermarks data hiding where some code is embedded into the image at the time of generation. Verifying this code authenticates the originality of image.

In the first step in detection stage, watermarked video file is decomposed into frames series using while loop.
In each frame grabbed from the loop, its watermarks positions and values is compared with the corresponding excel file that saved in the embedding stage, if the values are not equals, an alarm inform the responsible person about the forgeries, moreover identify the forged frame and the original frame that uploaded before in the cloud also the forged region in forged frame, else loop will continue comparing until finding new forged fame or finish reading.

4.4 Summery

This chapter presented the methodology of our detector streaming system, which discuss three important stages that need to asses our proposed method, Stage One: Embedding: this stage presented the proposed embedding algorithms that applied to the system that spreads 36 watermarks on each frame grapped from IP camera and save its value to cover the whole image. Stage Two: Tempering: this stage presented the tempering process that carried out on the captured video from stage one, temporal duplication, cloning effects applied to the watermarked video. Stage Three: Detecting: this stage presented on details the proposed detecting method that compares the values of each watermark on the video with saved value, moreover detect the temporal forgeries like cloning, duplication or compression effects.
Chapter 5
Experiments
Chapter 5

Experiments

5.1 Introduction

This chapter presents the system we utilized to detect duplication and cloning forgeries, in the video streaming that captured from IP surveillance camera, and the experiments established to assess the accuracy and reliability of our system. The main objective of the evaluation is to assess the reliability of the detector streaming system: we aim to explore the extent to which the proposed system can accurately suggest suitable and correct forgeries detection to the input saved video file.

Similar approaches from the state of the art have been evaluated by being compared to other approaches. However, we are not aware of any similar approach that considers the motion detection to send the origin forged series to Cloud, moreover, our system is recommended to be used by Raspberry PI card.

5.2 Dataset

The dataset is a set of three videos recorded using our detector system, captured from three different places, first place is video captured in place that has moving objects like place with traffic crowded for one hour, the second video is captured in place that has little moving objects like farm and moving tree, the third video is captured in place that has not any moving objects like store that store valuable things or objects. These three videos are forged by duplication and cloning forgeries.

Moreover, for each produced video file there is an excel file that saves all watermarks information’s about the related video file. This contains 36 excel sheets each sheet represents all needed information’s for one watermark in each captured frame from the streaming. This file will be used later to check the integrity of the related video file. The collected information’s are depicted in (Table 5.1).
Table 5.1: Detector streaming excel file

<table>
<thead>
<tr>
<th>X-axis</th>
<th>This column represents x-coordination of given watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y-axis</td>
<td>This column represents y-coordination of given watermark</td>
</tr>
<tr>
<td>Red Color</td>
<td>This column represents red value color of given pixel</td>
</tr>
<tr>
<td>Green Color</td>
<td>This column represents green value color of given pixel</td>
</tr>
<tr>
<td>Blue Color</td>
<td>This column represents blue value color of given pixel</td>
</tr>
</tbody>
</table>

So, the aim is to assess how the results of forgeries detection are correct and assess the time of detecting forgeries, moreover to assess the suitable resources that needed to be used with our detecting system.

The three videos recorded using our detector streaming system, captured on different places, size, and different forgery places, to measure the accuracy and execution time, and to determine the suitable resources needed to run our system. The following table (Table 5.2) explains in details our dataset properties that have been used in the experiments, which consider as a factor in determining the goals of our experiments.

Table 5.2: Original videos properties

<table>
<thead>
<tr>
<th>Video place</th>
<th>File size</th>
<th>Frame number</th>
<th>Frames on cloud</th>
<th>Upload speed</th>
<th>Duration</th>
<th>Excel file size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rarely movement</td>
<td>92.0 MB</td>
<td>20250</td>
<td>102</td>
<td>0.37 Mbps</td>
<td>00:22:30.067</td>
<td>18.2 MB</td>
</tr>
<tr>
<td>Little movement</td>
<td>430.0 MB</td>
<td>52208</td>
<td>310</td>
<td>0.27 Mbps</td>
<td>00:58:00.00</td>
<td>85.0 MB</td>
</tr>
<tr>
<td>Lots movement</td>
<td>768.0 MB</td>
<td>57176</td>
<td>1865</td>
<td>0.45 Mbps</td>
<td>01:03:30.00</td>
<td>125.0 MB</td>
</tr>
</tbody>
</table>

As shown on Table 5.2, video place column illustrate the place where the three videos are captured, this will affect consequently on the other properties like frames
number on the cloud, that’s means more movements we have on the video and more frames will be uploaded to the cloud.

The following table (Table 5.3), illustrates on details each property mentioned on Table 5.2

<table>
<thead>
<tr>
<th>Video place</th>
<th>Illustrate the place that we record the video on it using our detector streaming system.</th>
</tr>
</thead>
<tbody>
<tr>
<td>File size</td>
<td>Illustrate the size of the produced file after embedding the watermarks.</td>
</tr>
<tr>
<td>Frame number</td>
<td>This column shows the total frame number on each file.</td>
</tr>
<tr>
<td>Frames on cloud</td>
<td>This column shows the total number of frames that will upload to the cloud, these frame represent the origin motion recording on each video file.</td>
</tr>
<tr>
<td>Upload speed</td>
<td>Represents internet uploading speed that used to upload to Cloud.</td>
</tr>
<tr>
<td>Duration</td>
<td>This column represents recording duration of each file.</td>
</tr>
<tr>
<td>Excel file size</td>
<td>Illustrate the size of the produced excel file for each video file this file will be used to detect different type of forgeries.</td>
</tr>
</tbody>
</table>
5.3 Experiment Settings

The experiments were carried out in computer that executes the code of detector streaming system that written using python programming language and save the produced videos and corresponding excel files, this laptop connected to 5.0 Megapixel USB PC Camera Webcam, the specifications of this computer are depicted in (Table 5.4).

Table 5.4: Machine specifications

<table>
<thead>
<tr>
<th>Machine</th>
<th>HP laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel(r) core(tm) i3-4005u CPU</td>
</tr>
<tr>
<td></td>
<td>@ 1.70GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>4.00 GB</td>
</tr>
<tr>
<td>OS</td>
<td>Windows 10 Education</td>
</tr>
</tbody>
</table>

Video editing has become more and more popular due to the popularity of camcorders, to save money; you can use some free video editing tools. But it seem to be difficult to find a good free video creator as there are too many free video editing programs available on the market like (Avidemux 2.7, Movavi Video Editor 14, VideoPad Video Editor, HitFilm express 2017).

On our experiments, forging stage has used VideoPad; it has an intuitive user interface that you can learn to use in few minutes, Available for Windows and Mac OS X. 64-bit operating systems only, it doesn’t need much hardware racecourses to run this tool. The following Figure 5.1 show user interfaces about the tool.
5.4 Experiments Process

The evaluation process had two experiments. The first experiment aimed to determine the reliability of the detector streaming system for detecting duplication forgery. The second experiment was for determining the reliability of our system on detecting cloning forgery for the three placed as mentioned (refer to section 4.2). We ran the detector streaming system on the dataset and record the results. In the next sections, we discuss in details the two experiments and their properties to be used later in the evaluation process.

5.4.1 Experiment: Making Duplication Forgeries

On the first types of experiments which are making duplication forgeries on the videos that captured by live streaming camera. On the other words, duplications forgery on this stage means, remove portions from the concerned video and replace those portions by another portions from the same video that haven’t the object to be hidden.
Duplication forgeries did by removing portion of frames particularly in time of moving, by duplicates of the same of any amount of portion that haven't any movement on it. To conceal any object from that video. Detector streaming system process this kind of by check if there are some regions in the video series have watermarks value different to the saved value moreover if the number of different value watermarks is between 30 and 36 (number of embedded watermarks in each frame), if the number is equal to 36, so the system consider the forgery type as duplication forgery, because the duplication will carried out by removing frame portions and replace it with the same amount from existing video files to hide what he want to hide. In the following sections after the videos have been recorded by the detector streaming system, we discuss in details the duplication forgeries process using VideoPad Tool for each video files.

5.4.1.1 Rarely Movement Video
This video has been captured in a place that has rarely movements, the main purpose of this video to test our proposed system and conform the reality, the place about a store that store valuable mobile phone, the surveillance camera focus the recording on that mobile in all the time, suddenly and at specific time the mobile has been stolen from its place, and we on our turn we hide the mobile robbery from the recorded video using VideoPad tool and return the video with the same duration and quality to prevent the responsible to viewing the mobile robbery.

As shown in Table 5.2, the recording duration is 22 minutes and 30 seconds, this video consists of 20250 frames to construct video file with size 92.0 Mega Byte which saved on the machine that run the detector streaming system, moreover, excel file has been saved that contains all information about the embedded watermarks with size of 18.2 Mega Byte, this file by its role will be used on comparing process and discovering different types of forgeries.

From the other hands, 102 frames has been uploaded to the cloud, on our experiments DropBox has been used for this purpose, these frames has been captured on case of movements and named by the frame number and saved under folder on the Dropbox named by the current recording date. The average size of the captured frames on case of movement that should be uploaded to the cloud is 315 Kilo Byte, The system approximately takes six minutes to upload these frames to DropBox by
0.37 Mbps of uploading speed, and this uploading time calculated by the following equation that calculate the expected uploading time.

\[
\text{Total uploading time} = \frac{\text{frame number} \times \text{frame size}}{1024 \times \text{upload speed} \times 60}
\]

**Equation 5.1:** Time needed to upload frames

The mobile robbery on this video has been noticed in minute 16 and 33 seconds, where the thief steal the mobile after minute 16 and the mobile didn’t appear any more on the video till the end of recording time, here the role of the forged person has been carried out by using videoPad tool to temper the video file to hide the mobile robbery. This mean the tempering has been carried out between minute 16 and 22 on the video file.

Making duplication forgery in video file is very simple using VideoPad tool; the following Figure 5.2 shows how the duplication forgery carried out using VideoPad timeline.

![Figure 5.2: Making duplication in VideoPad timeline first experiment.](image)

The video file has been imported into the tool timeline, split the portion of frames that show mobile robbery as shown in the figure we named it portion2, delete this portion and make copy from the same file with same duration we named portion1 this portion didn’t show mobile robbery, paste portion1 in the place of portion2, the
last step we export the file and we choose export as lossless video without recompression to prevent losing the quality of the video file because it was compressed on time of recording.

5.4.1.2 Little Movement Video

On this experiment, the video is captured on a place that has a few motions detected on it, the place was a garden, and the IP Surveillance camera was recording to protect moving kids from kidnaping on that garden, the two forgeries types have been applied to hide the criminal from the recorded video using VideoPad Tool, the following two figures show a frame before and after making duplication forgeries.

![Figure 5.3: Image before and after duplication forgery](image)

As shown in table 5.2, the recording duration is 58 minutes, this video consists of 52208 frames to construct video file with size 430.0 Mega Byte which saved on the machine that run the detector streaming system, moreover, excel file has been saved that contains all information about the embedded watermarks with size of 85 Mega Byte, this file by its role will be used on comparing process and discovering different types of forgeries.

From the other hands, 310 frames has been uploaded to the cloud. The average size of the captured frames on case of movement that should be uploaded to the cloud is 360 Kilo Byte, The system approximately takes four minutes to upload these frames to DropBox by 0.27 Mbps of uploading speed, and this uploading time calculated by equation 2 that calculate the expected uploading time.
Figure 5.4: Making duplication in VideoPad timeline on second experiment.

As shown in Figure 5.4, The video file has been imported into the VideoMap timeline named copy1, another copy has been dragged on the VideoMap timeline named copy2, the next step is to fine a portion from the origin video file that have not any movements, that will be used to forge the video, the portion in minute 25:00 to 27:30 record that the kids stay on his place without any movements, this portion will be used to hide any movements of the criminal on the video file we named duplication part, the next step is to extract this portion from the video by split this durations from the right and the left, and delete the portion that placed on the right and the lift of this duration to keep the portion that will make the forgeries.

The criminal has been recorded in three different places on the video file, we make the forgeries depending on the place and the recorded object, duplication has been applied from minute 15:00 to minute 17:30 and from minute 56:30 to minute 58:00 by making two copy of duplication part above these two places, which capture the criminal on it, another place has been captured the criminal but cloning technique is applied and will discussed on details on cloning part of this experiment, the last step we export the file and we choose export as lossless video without recompression to
prevent losing the quality of the video file because it was compressed on time of recording.

5.4.1.3 Lots Movement Video
This experiment has been recorded on a university that have much moving people, and there were students that setting on public chairs, the forgeries carried out on the students that set on the public chairs, by hiding these students from that chairs on some places in the recorded video. The following two figures show a frame before and after making duplication forgeries.

![Image before and after duplication forgery](image)

**Figure 5.5**: Image before and after duplication forgery

As shown in Table 5.2, the recording duration is 63 minutes, this video consists of 57176 frames to construct video file with size 768.0 Mega Byte which saved on the machine that run the detector streaming system, moreover, excel file has been saved that contains all information about the embedded watermarks with size of 125 Mega Byte, this file by its role will be used on comparing process and discovering different types of forgeries.

For the other hands, 1865 frames have been uploaded to the cloud. The average size of the captured frames on case of movement that should be uploaded to the cloud is 435 Kilo Byte, The system approximately takes 30 minutes to upload these frames to DropBox by 0.45 Mbps of uploading speed, and this uploading time calculated by equation 2.

The following Figure 5.6 shows how the duplication forgery carried out using VideoPad timeline.
As illustrated in Figure 5.6, the video file has been imported into the tool timeline, split the portion of frames that show the empty public chair between minute 15:56 to 17:10, we use this portion to carry out the duplication forgery, we notice between minute 26:30 to 27:00 and between minute 36:14 to 37:14 that there were some students that set on that chair, for this purpose duplication carried out on these two location to hide the students, so when we play back the video again, the students on these two location are disappear, the last step we export the file and we choose export as lossless video without recompression to prevent losing the quality of the video file because it was compressed on time of recording.

5.4.2 Experiment: Making Cloning (copy-move) Forgeries

On the first types of experiments which are making duplication forgeries on the videos that captured by live streaming camera. On the other words, cloning forgery on this stage means, remove parts of portions from the concerned video and replace those parts of portions by another parts of portions from the same video that haven’t the object to be hidden.

Cloning forgeries did by clone or copy and paste a part of the image to hide an object or person is one of the most commonly used image manipulation
techniques. When it is done with care, it becomes almost impossible to detect the clone visually.

5.4.2.1 Rarely Movement Video

Making cloning forgery in video file is very simple using VideoPad tool; the following Figure 5.7 shows how the cloning forgery carried out using VideoPad timeline.

![Diagram](image)

**Figure 5.7:** Making cloning in VideoPad timeline on first experiment.

The video file has been imported into the tool timeline named portion1, another portion copied with approximately 8 minutes form the origin video file has been imported to the same timeline, this portion has been copied from a place that record existing object named portion2, next step is using crop tool to crop only the object from portion2, so when the video is played again from the beginning to the last the mobile robbery has been hidden by merging portion 2 with portion 1, the last step we export the file and we choose export as lossless video without recompression to prevent losing the quality of the video file because it was compressed on time of recording.
5.4.2.2 Little Movement Video

Making cloning forgery in video file is very simple using VideoPad tool; As Shown in Figure 5.8 the cloning forgery carried out using VideoPad timeline between minute 24:00 to 26:30, the following figures show a frame before and after cloning forgeries.

![Figure 5.8: Image before and after cloning forgery](image)

Part of the criminal has been captured on the video from minute 24:18 to 24:38, to hide that object from the video some steps carried out on the file using VideoPad tool. The video file has been imported into the tool timeline, another portion copied with approximately 2.5 minutes form the origin video file has been imported to the same timeline named cloning part, this portion has been copied from a place that record existing object that means the kid stay on his place without movements, next step is using crop tool to crop only the criminal from cloning part, so when the video is played again from the beginning to the last the mobile kidnaping has been hidden by merging cloning part with origin video timeline, the last step we export the file and we choose export as lossless video without recompression to prevent losing the quality of the video file because it was compressed on time of recording.

5.4.2.3 Lots Movement Video

As Shown in Figure 5.9 the cloning forgery carried out using VideoPad timeline from minute 44:21 to 47:00, and on another place from 55:07 and 58:17 the following figures show a frame before and after cloning forgeries.
As illustrated in the figure above, the student that set on the chair is hidden by cropping only empty chair from other place on the video, to accomplish this, The video file has been imported into the tool timeline, another copy has been cut from the origin file, on this copy cropping effect has been carried out to crop only the empty chair from that sequence, then this copy has been imported into the tool timeline above the origin file in the two places from minute 44:21 to 47:00, and on another place from 55:07 and 58:17, so when the video is play back again the chair appear empty on these two places, the last step we export the file and we choose export as lossless video without recompression to prevent losing the quality of the video file because it was compressed on time of recording.

5.5 Summary

This chapter presents the experiments that carried out to our system that will be used in the evaluation process, the experiments carried out on three videos captured on different palaces and different size, duplication and cloning forgeries have been applied on each video from the dataset. We explain how the forgery process can happen on each video and define the row data that will be used in the evaluation process.

Furthermore, we explain on details the steps of making forgeries and the forgeries places on each video using VideoPad tool, first video recorded (rarely movement) mobile robbery and we explain how it disappeared from the video, the
second video (little movement) show kidnapping kids, on it we explain on details the forgeries places and the process of hiding kid kidnapping, the last video (lots movement) recorded in university to hide setting students on public chair.
Chapter 6
Result Discussion
Chapter 6

Result discussion

6.1 Introduction

This chapter presents the results of experiments that explored in chapter 5 (Experiments), these experiments show the different type of possible forgeries that can be occurred in the original captured video file, or original recorded video file, to assess the effectiveness of our detector streaming on different cases of use. The main objective of this chapter is to show the results and discuss the results of using our detector streaming to detect the forgeries or any changes that can be occurred on the video file; on the other hand, we will measure the accuracy of the proposed system.

On this chapter the results have been explored in two sections, the first section discusses the result of making duplication forgeries for the three experiments; the second section discusses the result of making cloning forgeries on the three experiments (mentioned in chapter 5 Experiments). On each section, a table has been explored the results of detecting forgeries and it’s accuracy in a different way and places, moreover, on different types of compressions not only lossless compression.

6.2 Explanation of the Idea

This section presents the proposed idea that found to construct our detector streaming system after applying duplication and cloning forgeries on each experiment as mentioned in chapter 5.

Our detector streaming system can detect and find if there are any change occurred on the videos, by comparing the video with the original values of 36 watermarks that spread on video file and saved on excel file(as discussed on chapter 4 methodology). That’s mean when we applied any compression algorithms like MP4, AVI and MOV to the origin videos except lossless compression algorithms, consequently, the values of watermarks will be changes from beginning of video to the end.

Detector streaming system will detect the difference and inform that there are changes occurred to the original video, but the main objective of our research is to
define if the video have forgeries like duplication and cloning not only compression changes, moreover, define type and place on any video that have forgeries.

For this reason, different type of compression algorithm like MP4, AVI and MOV have been applied to each forged videos to analyses the values of watermarks on each forged video compressed, by comparing watermarks values in compressed videos and original forged video. The following tables, from Table 6.1 to Table 6.3 depict the value of specific watermark on different compression format for the same video in all experiments.

**Table 6.1:** Depicts the value of watermark number 35 on different compression format for the same video on Exp1

<table>
<thead>
<tr>
<th>Frame number</th>
<th>Watermark X-coordinate</th>
<th>Watermark Y-coordinate</th>
<th>Total RGB original (MP4)</th>
<th>Total RGB (AVI)</th>
<th>Total RGB (MOV)</th>
<th>Total RGB (Lossless)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11900</td>
<td>415</td>
<td>400</td>
<td>114</td>
<td>108</td>
<td>108</td>
<td>114</td>
</tr>
<tr>
<td>11901</td>
<td>410</td>
<td>400</td>
<td>108</td>
<td>106</td>
<td>106</td>
<td>108</td>
</tr>
<tr>
<td>11902</td>
<td>405</td>
<td>400</td>
<td>111</td>
<td>108</td>
<td>108</td>
<td>111</td>
</tr>
<tr>
<td>11903</td>
<td>400</td>
<td>400</td>
<td>111</td>
<td>109</td>
<td>109</td>
<td>111</td>
</tr>
<tr>
<td>11904</td>
<td>395</td>
<td>400</td>
<td>111</td>
<td>106</td>
<td>106</td>
<td>111</td>
</tr>
<tr>
<td>20055</td>
<td>440</td>
<td>400</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>111</td>
</tr>
<tr>
<td>20056</td>
<td>435</td>
<td>400</td>
<td>108</td>
<td>104</td>
<td>108</td>
<td>108</td>
</tr>
<tr>
<td>20057</td>
<td>430</td>
<td>400</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>114</td>
</tr>
<tr>
<td>20058</td>
<td>425</td>
<td>400</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>20059</td>
<td>420</td>
<td>400</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>112</td>
</tr>
</tbody>
</table>
Table 6.2: Depicts the value of watermark number 18 on different compression format for the same video on Exp2

<table>
<thead>
<tr>
<th>Frame number</th>
<th>Watermark X-coordinate</th>
<th>Watermark Y-coordinate</th>
<th>Total RGB (MP4)</th>
<th>Total RGB (original AVI)</th>
<th>Total RGB (MOV)</th>
<th>Total RGB (Lossless)</th>
</tr>
</thead>
<tbody>
<tr>
<td>49890</td>
<td>385</td>
<td>400</td>
<td>33</td>
<td>34</td>
<td>33</td>
<td>40</td>
</tr>
<tr>
<td>49891</td>
<td>460</td>
<td>400</td>
<td>51</td>
<td>52</td>
<td>51</td>
<td>65</td>
</tr>
<tr>
<td>49892</td>
<td>455</td>
<td>400</td>
<td>68</td>
<td>65</td>
<td>68</td>
<td>58</td>
</tr>
<tr>
<td>49893</td>
<td>450</td>
<td>400</td>
<td>123</td>
<td>125</td>
<td>123</td>
<td>122</td>
</tr>
<tr>
<td>49894</td>
<td>445</td>
<td>400</td>
<td>111</td>
<td>112</td>
<td>111</td>
<td>117</td>
</tr>
<tr>
<td>49895</td>
<td>440</td>
<td>400</td>
<td>130</td>
<td>129</td>
<td>130</td>
<td>127</td>
</tr>
<tr>
<td>49896</td>
<td>435</td>
<td>400</td>
<td>142</td>
<td>147</td>
<td>142</td>
<td>148</td>
</tr>
<tr>
<td>49897</td>
<td>430</td>
<td>400</td>
<td>117</td>
<td>125</td>
<td>117</td>
<td>126</td>
</tr>
<tr>
<td>49898</td>
<td>425</td>
<td>400</td>
<td>93</td>
<td>90</td>
<td>93</td>
<td>111</td>
</tr>
<tr>
<td>49899</td>
<td>420</td>
<td>400</td>
<td>72</td>
<td>71</td>
<td>72</td>
<td>76</td>
</tr>
<tr>
<td>49900</td>
<td>415</td>
<td>400</td>
<td>96</td>
<td>96</td>
<td>96</td>
<td>96</td>
</tr>
</tbody>
</table>

Table 6.3: Depicts the value of watermark number 7 on different compression format for the same video on Exp 3

<table>
<thead>
<tr>
<th>Frame number</th>
<th>Watermark X-coordinate</th>
<th>Watermark Y-coordinate</th>
<th>Total RGB (MP4)</th>
<th>Total RGB (original AVI)</th>
<th>Total RGB (MOV)</th>
<th>Total RGB (Lossless)</th>
</tr>
</thead>
<tbody>
<tr>
<td>23297</td>
<td>200</td>
<td>320</td>
<td>493</td>
<td>463</td>
<td>496</td>
<td>463</td>
</tr>
<tr>
<td>23298</td>
<td>205</td>
<td>320</td>
<td>762</td>
<td>757</td>
<td>762</td>
<td>757</td>
</tr>
<tr>
<td>23299</td>
<td>210</td>
<td>320</td>
<td>751</td>
<td>745</td>
<td>751</td>
<td>745</td>
</tr>
<tr>
<td>23300</td>
<td>215</td>
<td>320</td>
<td>240</td>
<td>234</td>
<td>231</td>
<td>234</td>
</tr>
<tr>
<td>23301</td>
<td>140</td>
<td>320</td>
<td>98</td>
<td>102</td>
<td>101</td>
<td>102</td>
</tr>
<tr>
<td>23302</td>
<td>145</td>
<td>320</td>
<td>77</td>
<td>93</td>
<td>109</td>
<td>93</td>
</tr>
<tr>
<td>Frame number</td>
<td>Watermark X-coordinate</td>
<td>Watermark Y-coordinate</td>
<td>Total RGB (MP4)</td>
<td>Total RGB original (AVI)</td>
<td>Total RGB (MOV)</td>
<td>Total RGB (Lossless)</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------</td>
<td>------------------------</td>
<td>-----------------</td>
<td>--------------------------</td>
<td>-----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>23303</td>
<td>150</td>
<td>320</td>
<td>417</td>
<td>398</td>
<td>458</td>
<td>398</td>
</tr>
<tr>
<td>23304</td>
<td>155</td>
<td>320</td>
<td>741</td>
<td>722</td>
<td>754</td>
<td>722</td>
</tr>
<tr>
<td>23305</td>
<td>160</td>
<td>320</td>
<td>634</td>
<td>662</td>
<td>636</td>
<td>662</td>
</tr>
<tr>
<td>23306</td>
<td>165</td>
<td>320</td>
<td>314</td>
<td>283</td>
<td>305</td>
<td>283</td>
</tr>
</tbody>
</table>

These tables compare the value of specific watermark on various frames in the three experiments, and changes because of compression method, that means for example watermark number 35 in frame 11900 with coordination (415,400) have total 114 on the origin video, when same video compressed by AVI algorithm it have 108, 108 for MOV and lossless 114 compression algorithm. At most watermark in the rate of approximately 80%, we found the change value between the original compressed video and other compressed format from the same video didn’t exceed 4 bits on each watermark.

This amount of change consider normal change and this change occurred by compression algorithms, on the other hand, if amount of change is more than or equal 5 bits, this case will consider as abnormal change and this amount of change were occurred because of forgeries.

Detector streaming system developed based on the information that there are abnormal changes when affected by Spatio-Temporal Tempering (Frame sequences are altered as well as visual contents of the frames are modified in the same video.) (Upadhyay & Singh, 2012; Yin & Yu, 2001), Figures 6.1 and 6.2 illustrate the difference between original captured video and the same lossless forged video.

By this analysis main objective of this research have been achieved by determining if there were forgeries carried out, not only but also determine the place and types of forgeries in the given video.
Figure 6.1: Watermarks 10, 20, 30 for original video exp2

Figure 6.2: Watermarks 10, 20, 30 for Forged Lossless exp2
6.3 Results

The following sections demonstrate the validity of our analysis and proposed information; moreover measure the accuracy of our detector streaming on various types of forgeries by equation 6.1.

\[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN}
\]

**Equation 6.1:** Calculate the accuracy of detector streaming system

6.3.1 Result Discussion for Duplication

This section explores on details the result and discusses it, and finds the logical reasons to validate the ability of detecting forgery for each experiments established that have duplication forgeries on it. We found that approximately not exceed 20% from the watermarks values were changes as abnormal change after compression at all experiments established, and there were 36 watermarks spread on each frame on the video, for this reasons we decide if the number of changed watermarks exceed 30 watermarks in such frame, this frame consider have duplication forgery on rate of 80%.

6.3.1.1 Result Discussion for Duplication on Experiment 1

The following Table 6.4 depicts the versions of the first experiment on various formats, and show the number of frames that consider as have duplication forgery but on fact it didn’t have duplication forgery, knowing that these versions were not forged by duplications.

**Table 6.4:** Depicts the datum of experiment1 and show the false positive of these video.

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>Duration</th>
<th>Frame number</th>
<th>Frame number of abnormal change (FP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4 original video</td>
<td>00:22:30</td>
<td>20251</td>
<td>61</td>
</tr>
<tr>
<td>AVI</td>
<td>00:22:30</td>
<td>20151</td>
<td>59</td>
</tr>
<tr>
<td>MOV</td>
<td>00:22:30</td>
<td>20151</td>
<td>61</td>
</tr>
<tr>
<td>Lossless</td>
<td>00:22:30</td>
<td>20251</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 6.5 shows on details the portion of video that have been prepared to make duplication forgery on the origin video as mentioned on (section 4.4.1.1 rarely movement video chapter 4).

**Table 6.5:** Depicts the datum of forged portion that used to carry out duplication.

<table>
<thead>
<tr>
<th>Portion number</th>
<th>Start duration</th>
<th>End duration</th>
<th>Total duration</th>
<th>Total Frame number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0:01:00.106</td>
<td>0:05:24.802</td>
<td>0:04:24.180</td>
<td>3965</td>
</tr>
</tbody>
</table>

The portion illustrated on Table 6.5 were used to construct the forged versions, the following Table 6.6 depicts on details the various forged videos version that passed into the detector streaming system to calculate the accuracy of by equation 6.1

**Table 6.6:** Depicts various versions of forged videos that have duplication forgery and it accuracy

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>Start forgery frame</th>
<th>End forgery frame</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4 original</td>
<td>913</td>
<td>4948</td>
<td>20251-2692=17559</td>
<td>2692</td>
<td>61</td>
<td>3965-2692=1273</td>
<td>0.930</td>
</tr>
<tr>
<td>AVI</td>
<td>908</td>
<td>4951</td>
<td>20151-2684=17467</td>
<td>2684</td>
<td>59</td>
<td>3965-2684=1281</td>
<td>0.929</td>
</tr>
<tr>
<td>MOV</td>
<td>908</td>
<td>4950</td>
<td>20151-2725=17426</td>
<td>2725</td>
<td>61</td>
<td>3965-2725=1240</td>
<td>0.931</td>
</tr>
<tr>
<td>Lossless</td>
<td>913</td>
<td>4875</td>
<td>20151-2917=17234</td>
<td>2917</td>
<td>2</td>
<td>3965-2914=1048</td>
<td><strong>0.950</strong></td>
</tr>
</tbody>
</table>

The previous Table 6.6 illustrates on details the result values of using detector streaming system to detect duplication forgery on experiment1, TP column show number of frame that consider as not forged and on fact not forged, TN column show number of frames that consider forged and on fact forged, FP column show number of frames that consider forged and on fact not forged, FN column show frame
number that consider not forged but on fact forged frames, as shown the accuracy of
detector system approximately 93% for the different versions and for lossless 97%.

6.3.1.2 Result Discussion for Duplication on Experiment 2
The following Table 6.7 depicts the versions of the second experiment on various
format, and show the number of frame that consider as have duplication forgery but
on fact it didn’t have duplication forgery, knowing that these versions were not
forged by duplications or any effects.

Table 6.7: Depicts the datum of experiment2 and show the false positive of these
video.

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>Duration</th>
<th>Frame number</th>
<th>Frame number of abnormal change (FP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>0:58:00.600</td>
<td>51951</td>
<td>2334</td>
</tr>
<tr>
<td>AVI original video</td>
<td>0:58:00.600</td>
<td>51951</td>
<td>2297</td>
</tr>
<tr>
<td>MOV</td>
<td>0:58:00.600</td>
<td>51951</td>
<td>2297</td>
</tr>
<tr>
<td>Lossless</td>
<td>0:58:00.600</td>
<td>51951</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.8 shows on details the portion of videos that have been used to make
duplication forgery on the origin video as mentioned on (section 5.4.1.2 little
movement video chapter 5).

Table 6.8: Depicts the datum of forged portion that used to carry out duplication.

<table>
<thead>
<tr>
<th>Portion number</th>
<th>Start duration</th>
<th>End duration</th>
<th>Total duration</th>
<th>Total Frame number</th>
<th>Forgery Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0:16:17.518</td>
<td>0:22:23.278</td>
<td>0:06:02.522</td>
<td>5440</td>
<td>Duplication</td>
</tr>
<tr>
<td>3</td>
<td>0:33:59.193</td>
<td>0:40:01.716</td>
<td>0:06:02.522</td>
<td>5440</td>
<td>Duplication</td>
</tr>
</tbody>
</table>
The portions illustrated on Table 6.8 were used to construct the forged versions; the following Table 6.9 depicts on details the various forged videos version that passed into the detector streaming system to calculate the accuracy of by equation 6.1

**Table 6.9:** Depicts various versions of forged videos that have duplication forgery and its accuracy.

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>Start and End portion1</th>
<th>Start and End portion2</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>146/98</td>
<td>79/17</td>
<td>51951-2334=49 617</td>
<td>5411+5 938 =11349</td>
<td>23</td>
<td>34</td>
<td>0.956</td>
</tr>
<tr>
<td>AVI original</td>
<td>146/88</td>
<td>304/86</td>
<td>51951-2297=49 654</td>
<td>5961+5 544 =11505</td>
<td>22</td>
<td>97</td>
<td>0.954</td>
</tr>
<tr>
<td>MOV</td>
<td>146/88</td>
<td>304/87</td>
<td>51951-2297=49 654</td>
<td>5960+5 543 =11503</td>
<td>22</td>
<td>97</td>
<td>0.954</td>
</tr>
<tr>
<td>Lossless</td>
<td>147/61</td>
<td>306/78</td>
<td>51951-9463-14 =42502</td>
<td>9463 14</td>
<td>14</td>
<td>10875-9463=1 412</td>
<td>0.973</td>
</tr>
</tbody>
</table>

The previous Table 6.9 illustrates on details the result values of using detector streaming system to detect duplication forgery on experiment2, TP column show number of frame that consider as not forged and on fact not forged, TN column show number of frames that consider forged and on fact forged, FP column show number of frames that consider forged and on fact not forged, FN column show frame number that consider not forged but on fact forged frames, as shown on Table (6.9)
the accuracy of detector system approximately 95% for the different versions and for lossless got 97% of accuracy.

6.3.1.3 Result Discussion for Duplication on Experiment 3
The following Table 6.10 depicts the versions of the third experiment on various format, and show the number of frame that consider as have duplication forgery but on fact it didn’t have duplication forgery, knowing that these versions were not forged by duplications.

**Table 6.10:** Depicts the datum of experiment3 and show the false positive of these video.

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>Duration</th>
<th>Frame number</th>
<th>Frame number of abnormal change (FP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>01:03:31</td>
<td>56893</td>
<td>3400</td>
</tr>
<tr>
<td>AVI original video</td>
<td>01:03:31</td>
<td>57177</td>
<td>2586</td>
</tr>
<tr>
<td>MOV</td>
<td>01:03:31</td>
<td>56893</td>
<td>2560</td>
</tr>
<tr>
<td>Lossless</td>
<td>01:03:31</td>
<td>57177</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.11 shows on details the portion of videos that have been used to make duplication forgery on the origin video as mentioned on (section 5.4.1.3 lots movement video chapter 5).

**Table 6.11:** Depicts the datum of forged portion that used to carry out duplication.

<table>
<thead>
<tr>
<th>Portion number</th>
<th>Start duration</th>
<th>End duration</th>
<th>Total duration</th>
<th>Total Frame number</th>
<th>Forgery Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0:16:18.664</td>
<td>0:23:45.752</td>
<td>0:07:27.026</td>
<td>6709</td>
<td>Duplication</td>
</tr>
<tr>
<td>2</td>
<td>0:42:28.752</td>
<td>0:49:34.718</td>
<td>0:07:05.965</td>
<td>6392</td>
<td>Duplication</td>
</tr>
</tbody>
</table>
The portions illustrated on Table 6.11 were used to construct the forged versions; the following table 6.12 depicts on details the various forged videos version that passed into the detector streaming system to calculate the accuracy of by equation 6.1.

**Table 6.12:** Depicts various versions of forged videos that have duplication forgery and its accuracy.

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>Start and End portion1</th>
<th>Start and End portion2</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>42 03</td>
<td>432 12</td>
<td>56893-8626=48 267</td>
<td>3189+1</td>
<td>186</td>
<td>4375</td>
<td>13101-4375=8 726</td>
</tr>
<tr>
<td>AVI original</td>
<td>42 77</td>
<td>432 11</td>
<td>57177-8632=48545</td>
<td>3219+1</td>
<td>195</td>
<td>4144</td>
<td>13101-4120=8 937</td>
</tr>
<tr>
<td>MOV</td>
<td>42 04</td>
<td>432 03</td>
<td>56893-8626=48 270</td>
<td>3189+1</td>
<td>176</td>
<td>4365</td>
<td>13101-4261=8 840</td>
</tr>
<tr>
<td>Lossless</td>
<td>42 64</td>
<td>431 34</td>
<td>57177-5057=52 120</td>
<td>336+11</td>
<td>90</td>
<td>1526</td>
<td>13101-3531=9 570</td>
</tr>
</tbody>
</table>

The previous Table 6.12 illustrates on details the result values of using detector streaming system to detect duplication forgery on experiment1, TP column show number of frame that consider as not forged and on fact not forged, TN column show number of frames that consider forged and on fact forged, FP column show number of frames that consider forged and on fact not forged, FN column show frame number that consider not forged but on fact forged frames, as shown the accuracy of detector system approximately 94% for the different versions.
6.3.2 Result Discussion for Cloning

This section explores on details the result discussion on the experiments that have cloning forgeries on it, moreover

6.3.2.1 Result Discussion for Cloning on Experiment 1

Table 6.13: Depicts the datum of forged portion that used to carry out cloning.

<table>
<thead>
<tr>
<th>Portion number</th>
<th>Start duration</th>
<th>End duration</th>
<th>Total duration</th>
<th>Total Frame number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0:01:00.106</td>
<td>0:05:24.802</td>
<td>0:04:24.180</td>
<td>3965</td>
</tr>
</tbody>
</table>

The portion illustrated on Table 6.13 were used to construct the forged versions, the following Table 6.14 depicts on details the various forged videos version that passed into the detector streaming system to calculate the accuracy of by equation 6.1

Table 6.14: Depicts various versions of forged videos that have cloning forgery and it accuracy.

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>Start forgery frame</th>
<th>End forgery frame</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lossless</td>
<td>939</td>
<td>4669</td>
<td>20251-143-3965=16143</td>
<td>127</td>
<td>16</td>
<td>3965-143=3822</td>
<td>0.809</td>
</tr>
</tbody>
</table>
6.3.2.2 Result Discussion for Cloning on Experiment 2

Table 6.15: Depicts the datum of forged portion that used to carry out cloning.

<table>
<thead>
<tr>
<th>Portion number</th>
<th>Start duration</th>
<th>End duration</th>
<th>Total duration</th>
<th>Total Frame number</th>
<th>Forgery Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0:16:17.518</td>
<td>0:22:23.278</td>
<td>0:06:02.522</td>
<td>5440</td>
<td>Cloning</td>
</tr>
<tr>
<td>3</td>
<td>0:33:59.193</td>
<td>0:40:01.716</td>
<td>0:06:02.522</td>
<td>5440</td>
<td>Cloning</td>
</tr>
</tbody>
</table>

The portions illustrated on Table 6.15 were used to construct the forged versions; the following Table 6.16 depicts on details the various forged videos version that passed into the detector streaming system to calculate the accuracy of by equation 6.1

Table 6.16: Depicts various versions of forged videos that have cloning forgery and it accuracy.

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>Start and End portion1</th>
<th>Start and End portion2</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lossless</td>
<td>1476 1 2020 9</td>
<td>3064 9 3656 9</td>
<td>5195-9320=426 31</td>
<td>4484+47 21 =9205</td>
<td>20+9 5 =115</td>
<td>10880-9320=15 60</td>
<td><strong>0.968</strong></td>
</tr>
</tbody>
</table>

6.3.2.3 Result Discussion for Cloning on Experiment 3

Table 6.17: Depicts the datum of forged portion that used to carry out cloning.

<table>
<thead>
<tr>
<th>Portion number</th>
<th>Start duration</th>
<th>End duration</th>
<th>Total duration</th>
<th>Total Frame number</th>
<th>Forgery Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0:16:18.664</td>
<td>0:23:45.752</td>
<td>0:07:27.026</td>
<td>6709</td>
<td>Cloning</td>
</tr>
<tr>
<td>2</td>
<td>0:42:28.752</td>
<td>0:49:34.718</td>
<td>0:07:05.965</td>
<td>6392</td>
<td>Cloning</td>
</tr>
</tbody>
</table>
The portions illustrated on Table 6.17 were used to construct the forged versions; the following Table 6.18 depicts on details the various forged videos version that passed into the detector streaming system to calculate the accuracy of by equation 6.1

**Table 6.18:** Depicts various versions of forged videos that have cloning forgery and its accuracy.

<table>
<thead>
<tr>
<th>Compression algorithm</th>
<th>Start and End portion1</th>
<th>Start and End portion2</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lossless</td>
<td>426 0</td>
<td>1100 6</td>
<td>4304 6</td>
<td>49050 0</td>
<td>57177-7820=493 57</td>
<td>1693+31 25</td>
<td>=4818 70</td>
</tr>
</tbody>
</table>

### 6.4 Summary

This chapter presented the evaluation of the system. And also discussed the results besides the strengths and weaknesses of the system.

We have formulated a dataset of 12 videos to assess the system. Results were calculated by using accuracy equation. The results indicated that our system achieved a high relevant measure with 97% accuracy on detecting duplication on case of lossless compression and 95% on the other compression algorithms; on the other hand on detecting cloning forgeries our system achieved a high relevant measure with 86% on case of lossless compression.
Chapter 7
Conclusion and Future work
Chapter 7

Conclusion and future work

In this work, we have developed a detector streaming system to make streaming and detecting forgery at the same time, the system can detect some effects that carried out like, if the original video compressed by any of compression methods, make duplication that means when replacing portion of the origin video by another portion from the same video that has a same duration like the replaced portion.

Another effect that can be detected by our detector streaming is cloning effect, that means clone yourself on the same video, in another word, make two or more copy of yourself in the concerned video, there are much videos and tools that easy learning how to clone yourself in recorded video, scientifically the cloning effect can be carried out by replacing a part of the desire frame portion not whole the frame portion like duplication effects, it's known that the video consist of a series of frames, on duplication effect the whole frame replaced by the forged portion but on cloning effect a part of the frame was replaced by the forged portion.

The system process consists of the following stages: first, watermark embedding, in this stage predefined watermarks positions have been hidden on the video stream, this information was used as a signature to validate the integrity of the concerned video, second, video tempering, in this stage the captured video from the first stage were subjected to some forgeries effects like duplication and cloning effects, to assess the accuracy of our proposed method, third, detecting forgery, in this stage the forged videos that processed on the second stage were subjected to our detector streaming system to detect the forged parts in the forged videos.

The detector streaming system is evaluated over 27 videos that have different formats and have different forgery types like duplication and cloning, result indicated that the system achieved high relevant measures with 97% accuracy on detecting duplication on case of lossless compression and 95% on the other compression algorithms, on the other hand on detecting cloning forgeries our system achieved a highly relevant measure with 86% on case of lossless compression, we evaluate the
system using a lossless algorithm because there were some tools that make effects to video without compression.

To our knowledge, this is the first work to explore forgeries like duplication and cloning in video streams and define the places in the given video that have forgery, that means it take care about demonstrating the integrity of the given video not only demonstrating the authenticity of video stream like the previous research, Current research in forgery detection is mainly limited to image tampering detection technique (not video), it can be some sort of cryptographic that produce mathematical scheme for demonstrating the authenticity and integrity of concerned video.

Our system is one of few works that utilize motion detector on it, if the motion detector found difference between compared frames, it will write the original image and send it by threading to the cloud, these live frames captured only when detects movement in such area of object to save the original frame, not only but also our system is especially designed for using in IP cameras, network video recorders and Raspberry Pi cart due to its code effectiveness, that didn’t require much resources to be executed.

Our work proposes a simple encryption-decryption algorithm that can be used on the live streaming or on a recorded video; it’s spatially designed to the sensitive videos, it accessing the data stream and investigate the original video data, to explore any change on the original video.

The results show that the system help explores any change on the data stream and define the effect types that carried out on the video file like duplication and cloning effects or normal compression change, not only but also it helps to define the start and end forgery frame series and get the origin frames if required.

**Future work**

We proposed an in-depth evaluation of our detector streaming system and explore the potential shortcomings and strengths of each involved stage on our work, this details evaluation can inform the related field researchers for designing similar approaches to explore more video effects.
REFERENCES:


Hacking CCTV’s, h. e. c. d. c. f. a.-s. c. h. p.


