A Project Report

On

Video Source Identification and Tampering Detection

Submitted in partial fulfilment of the requirement of

University of Mumbai

For the Degree of

Bachelor of Engineering

in

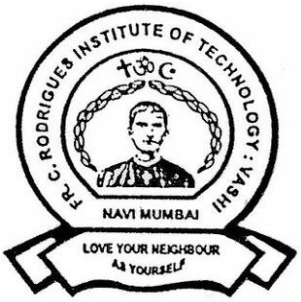
COMPUTER ENGINEERING

Submitted by

Madhavi Bhilegaokar
Vipul Borhade
Shreya Pai

Supervised by

Ms. Shweta Tripathi



Department of Computer Engineering

Fr. Conceicao Rodrigues Institute of Technology
Sector 9A, Vashi, Navi Mumbai - 400703

UNIVERSITY OF MUMBAI

2019-2020

APPROVAL SHEET

This is to certify that the project entitled

"Video Source Identification and Tampering Detection"

Submitted by

Madhavi Bhilegaokar	**101604**
Vipul Borhade	**101605**
Shreya Pai	**101637**

Supervisors : _____

Project Coordinator : _____

Examiners : **1.** _____

2. _____

Head of Department : _____

Date :

Place :

Declaration

We declare that this written submission for B.E. Declaration entitled "**Video Source Identification and Tampering Detection**" represent our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declared that we have adhere to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by institute and also evoke penal action from the sources which have thus not been properly cited or from whom paper permission have not been taken when needed.

Project Group Members:

1. Madhavi Bhilegaokar, 101604

_____

2. Vipul Borhade, 101605

_____

3. Shreya Pai, 101637

_____

Abstract

A video sequence provides more proof of an occurring event than a photo or a group of photos. Videos can be tampered in many ways: one may be interested in replacing or removing some frames (e.g., from a video-surveillance recording), replicating a set of frames, introducing, duplicating or removing some objects from the scene etc. Forensic analysis of multimedia data is being significantly researched upon in recent tines. A large part of the research activities in this field is devoted towards the analysis of still images, since digital photographs are largely used to provide objective evidence in legal, medical, and surveillance applications. There is already available significant research and literature on image forensics. However, in the case of video forensics, there are many research issues which remain unexplored, due to video signal peculiarities with respect to images. Also, there is a vast range of alterations possible that can be applied to this digital content. Several approaches target the possibility of validating, detecting alterations in video sequences, and recovering the chain of processing steps operated on digital videos. As a result, one can now determine if a video has cut and paste operations performed on it or is original, the video has been generated from which source (camera model, vendors), if there has been any artificial modifications in either the whole video or parts of it, also, how these modifications were done or what was the processing history of a video using video forensic techniques. Furthermore, the spatial resolution of videos is lower than images. For a video sequence, one can perform all potential modifications on both the single frames of the video and along the temporal direction. In this project, we explore the algorithms for the identification of the source camera and tampering detection.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Video forensics is an emerging field in this digital world. In this era, videos can be easily made and edited with the help of video editing tools like DaVinci Resolve, Adobe After Effects and Adobe Premiere Pro, even by a person with a non-technical background . A video can be made from varied diverse origins. Scenes of the same place can be taken from different angles from different sources which can then, be merged together. The amount of alterations that can be applied to a video is huge. This brings questions about the authenticity of a video.

1.2 Motivation

People often believe what they see in videos. Even if they don't believe it, it certainly does create a sense of doubt regarding the persons in the video. The image of the person/ persons in the video, is thereby manipulated among the general public. This manipulation may result in direct and indirect effects in stock prices, market value, votes etc. Also, videos can be a great source of proof in a court of law. The chain of acquisition of a video sequence should always be established while proving the authenticity of a video. To avoid false manipulations of story flows, persons or organisations, establishing the authenticity of the video is a necessity.

1.3 Aim and Objective

To avoid high repercussions in society due to the creation of fake videos or tampering of videos, we aim to find out if a given video is authentic or not. This will dampen the efforts of certain person or persons to manipulate the mindset of the society in favour of or against a person or organisation.

Chapter 2

Study Of the System

2.1 Video Forensics

Video forensics relates to the re-construction of the processing history of a given multimedia signal. The non-reversible operations applied to a signal leave some traces (footprints) that can be identified. These footprints are classified in order to reconstruct the possible alterations that have been operated on the source. One can say that this detection of footprints is a sort of reverse engineering.

2.1.1 Source Camera Identification

The Photo Response Non-Uniformity (PRNU) algorithm has been used to find out the authenticity of the source of the video.

In any acquisition pipeline, light enters the camera via the optical lens, and is filtered by the RGB Color Filter Array(CFA). It is then converted to a digital copy of light patterns by the internal CCD array. In-camera post processing of the acquired content is also done, after which the acquired content is compressed. This acquisition leaves individual fingerprints on the acquired content. It is these footprints which help determine the device from which the video was captured.

PRNU noise is a pattern noise that mainly depends on factors such as doping concentrations, the thickness of over layers, detector dimensions and the wavelength of illumination (spectral response). In other words, it is caused by the imperfections in the sensors. As PRNU is non-linear in nature, it becomes extremely difficult to remove and provides a much stronger and more reliable fingerprint than any other noise. Even the compression of videos will not affect this noise. If one capture this noise pattern, we can create distinctive connections between the source camera and the video.

2.1.2 Tampering Detection

For detecting whether any tampering has occurred in the video sequences, the Edge Detection Technique is being used. This technique can be used to find out if any frames have been inserted or deleted. Edge detection is an image processing technique for finding the boundaries of objects within images. There is a certain continuous flow in the movement of the edges of objects in one frame to another consequent frame in a video sequence. Thus, here, we compute the entering and exiting pixels of each frame from the video sequence and thereby find out whether any tampering has occurred.

2.2 Related Works

Emmanuel Kiegaing Kouokam el.al.[1] proposed a method for source identification for youtube videos. In this paper, the author takes into account the effects of video compression on PRNU noise in the frames. The author uses VISION dataset to check and verify the result. The experimental result showed that the PRNU method works more accurately and efficiently than the existing frame based method that uses either I frame or all (I-B-P) frames, especially on youtube videos.

In 2019 Enes Altinisik el.al.[2] discussed about the PRNU estimation of sensors for H.264 and H.265 videos. This work introduced different methods to mitigate widely deployed H.264 and H.265 video compression standards on PRNU estimation. The process uses decoding parameters to develop a weighting scheme and contribution of video frames at the macroblock level to the PRNU estimation process. The experimental results obtained on videos captured by 28 cameras show that our approach increases the PRNU matching metric more than five times over the conventional estimation method tailored for photos.

Enes Altinisik el.al.[3] discussed the estimation of PRNU noise in images and a procedure to estimate PRNU noise for H.264 video compression standard. They have discussed how they have reduced blockiness by removing a filtering procedure that had been applied at the decoder. For estimating PRNU noise pattern, they have also used macroblocks selectively. They have shown results to prove that the method they have used has improved matching performance significantly.

Dasara Shullani el.al.[4] discuss contribution of VISION dataset in PRNU estimation and the development of multimedia forensics. It also gives a detailed overview of the VISION dataset which contains 34427 images and 1914 videos, both in the native format and in their social version (Facebook, YouTube, and WhatsApp are considered), from 35 portable devices of 11 major brands.The paper gives possible application of dataset with experimental results like image source identification and video source identification with the help of dataset.

Author Yu Chen el.al.[5] studied Photo Response Non-Uniformity (PRNU) based image tampering detection methods and their applicability in real-world image tampering detection applications. The paper also provides a comparative study of Wavelet denoising filter and BM3D filter. It also presents a comparative study of 4 DSLR cameras with 80 images and 800 PRNU noise patterns.

Chapter 3

Proposed System

3.1 Problem Statement

Being a project in the cyber security domain, there are a wide range of methods and ways to conduct the forensic analysis of a video. One can rely purely on forensic tools to get the results of the forensic analysis. However, most of these tools are not open-source and are highly expensive.

Our system focuses on finding out the type of device that captured the video. Also, one can find out if a given video sequence has been tampered with. There are many ways in which a video can be tampered with, like, insertion, duplication and deletion of video frames, insertion, duplication and deletion of objects from a video scene, temporal expansion and contraction of a video sequence etc.

3.2 Scope

In our project, we focus mainly on finding if the query video has been acquired from the specified source. Also, we find out if the video has been tampered with. Our project focuses mainly on the insertion and deletion of frames in the video sequence.

3.3 Proposed System

Our research proposes using the Photo Response Non-Uniformity (PRNU) algorithm to help determine whether the query video, i.e, the video under observation is obtained from the source device as is specified. When a query video is submitted to the system by the user, the system finds out the most likely phone camera that could have captured the video. The possible models of the phone camera is included in the dataset that the system already has access to. Also, the system uses an Object-Edge Detection algorithm to determine if any video frame insertion or deletion has occurred in the video sequence. The type of forgery that occurs when video frames are inserted and deleted from a video sequence is called inter-frame video forgery. Upon selecting a video sequence and submitting it to the system, the system determines if the video sequence has any inter-frame tampering of video frames.

Chapter 4

Design Of the System

4.1 Requirement Engineering

4.1.1 Requirement Elicitation

Our project uses the VISION dataset. The VISION dataset is currently composed by 34,427 images and 1914 videos. We have used the video dataset of the VISION dataset. The videos that are in their native format and in their social version (Facebook, YouTube, and WhatsApp) are considered. These videos have been obtained from 35 portable devices of 11 major brands.

4.1.2 Software life-cycle model

Scrum process model is iterative and incremental. The system explained in the report has been developed based on the scrum process model. The team worked for a short period of time, i.e., sprint, and then presented a project increment at the end of each sprint.

4.1.3 Requirement Analysis

4.1.3.1 Activity Diagram

There are primarily three stages to check if a given video is authentic or has been tampered with. They are Preparation, Authentication and Presentation. Once a video is uploaded, then the PRNU Technique and Edge Detection Technique can be applied to check the authenticity of the source of the video, and to determine if the video has been tampered with or not. The Presentation stage includes displaying the outcomes of each of the algorithms.

4.1.3.2 Software requirement

The Software requirement for the project is as follows :

Table 4.1: Software Names and Their Versions

Name	Version
Python	python 3.5.0 or higher
Matlab	matlab2018b or higher
Django	2.2.11 or higher
Web Browser	Chrome 65X or Firefox 25x

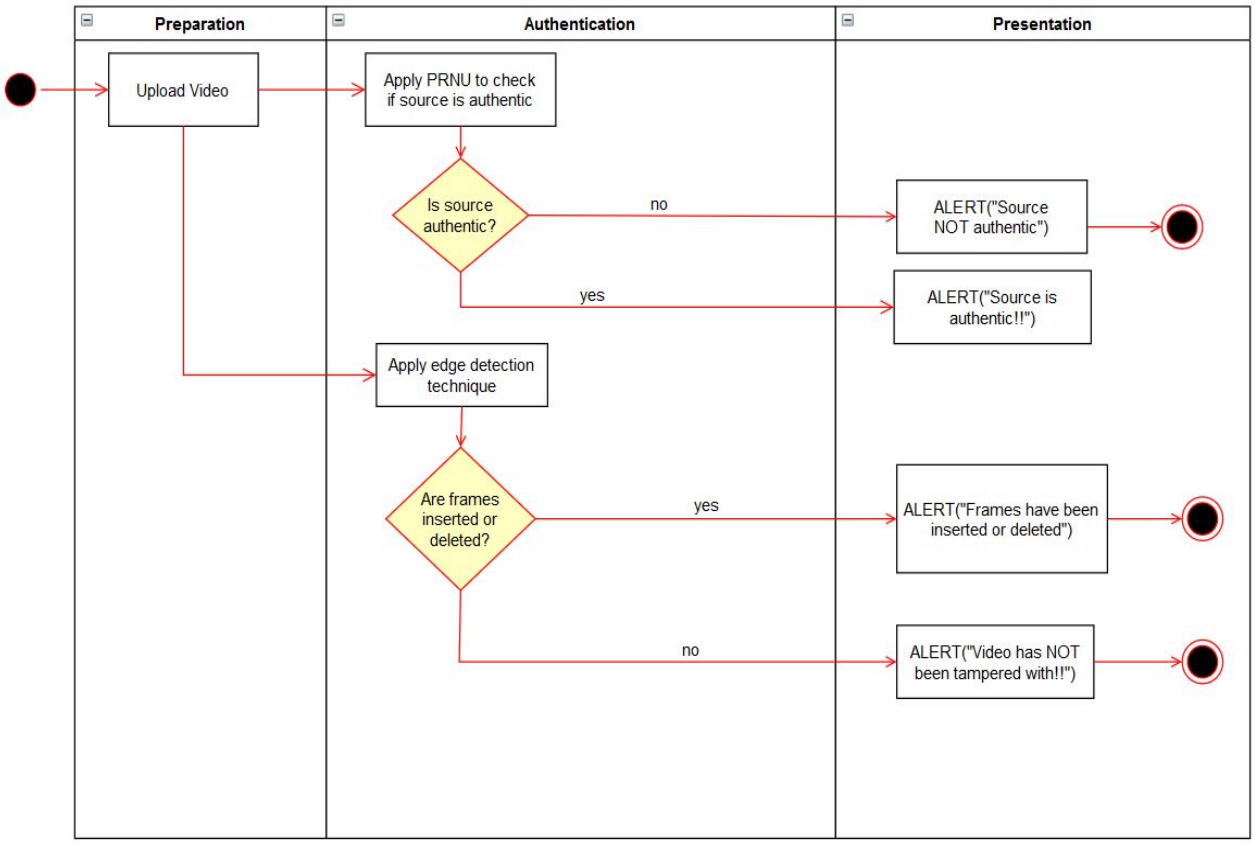


Figure 4.1: Activity Diagram of the System

4.2 System architecture

4.2.1 Flowcharts

The flowchart of calculating the PRNU fingerprint of the source camera is shown in Fig 4.2, and the flowchart of calculating the PRNU fingerprint of the query video and the identification of whether the query video has been generated from the source camera is shown in Fig 4.3.

The flowchart of the Edge Detection Technique to identify if any video frames have been inserted or deleted is shown if Fig 4.4.

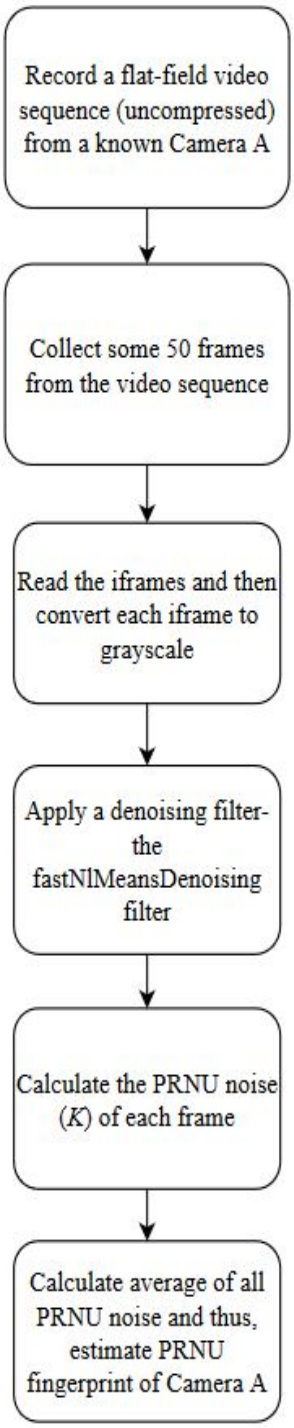


Figure 4.2: PRNU Fingerprint of the Source Camera

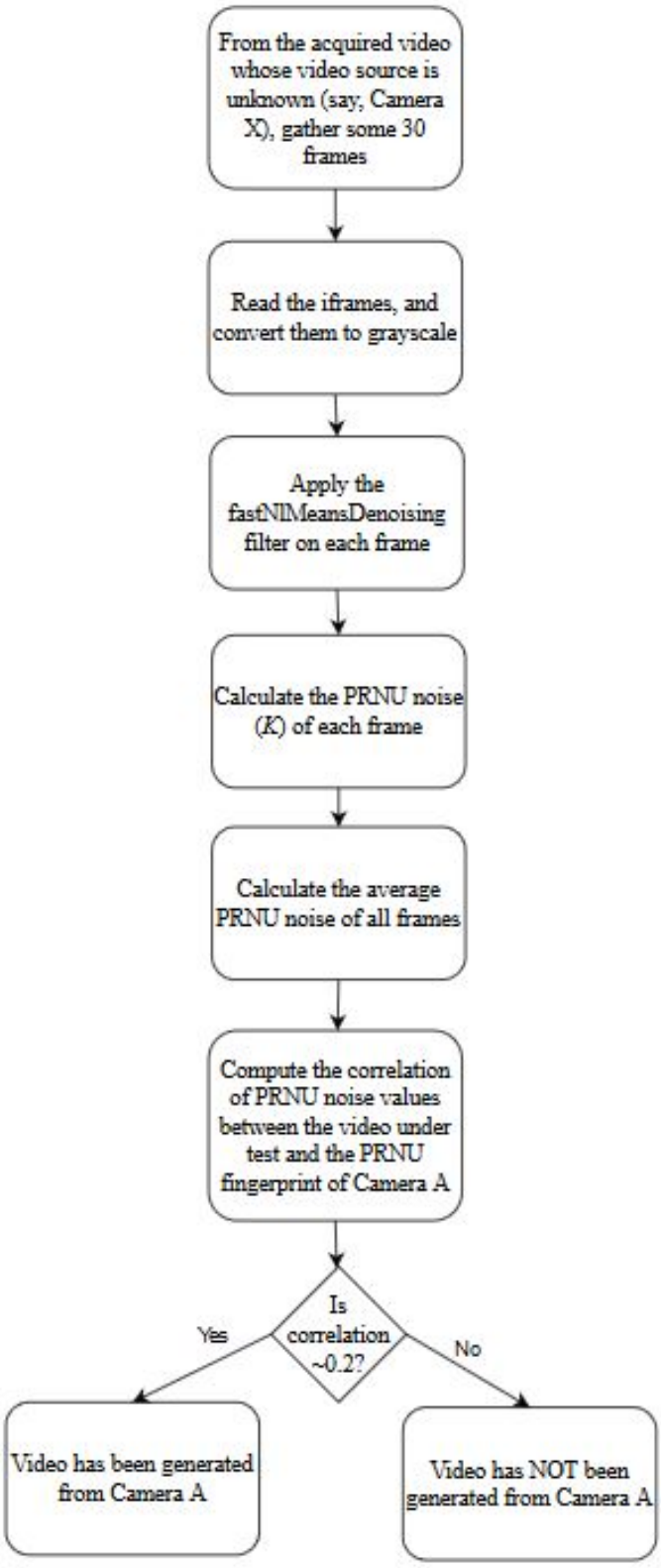


Figure 4.3: PRNU Fingerprint of the Query Video and Identification

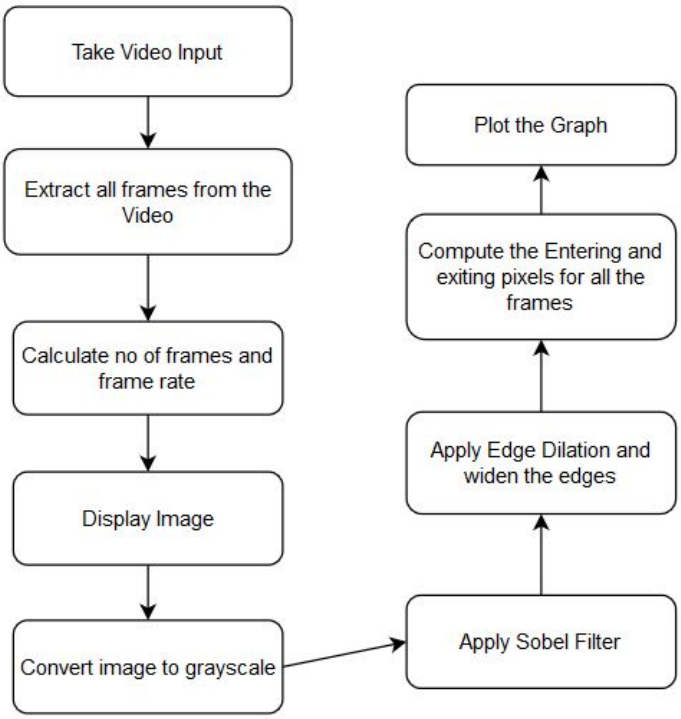


Figure 4.4: The Edge Detection Technique

4.2.2 Block Diagram

The block diagram of the entire system is shown in figure 4.5. The PRNU fingerprints of the source camera and the query video is considered for the identification of the source. On the other hand, the application of the edge detection technique is used to find whether a given query video has been tampered with or not.

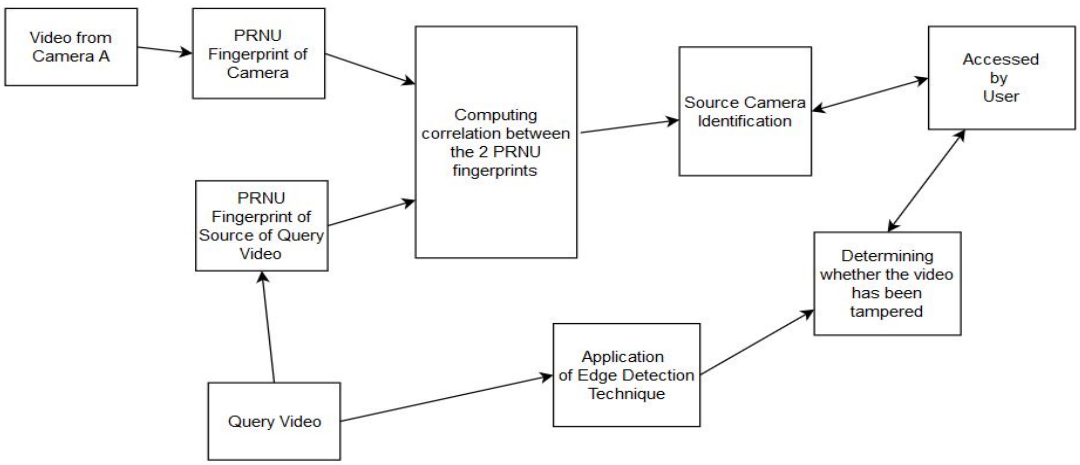


Figure 4.5: Block Diagram of the System

Chapter 5

Result and Discussion

5.1 Screenshots of the System

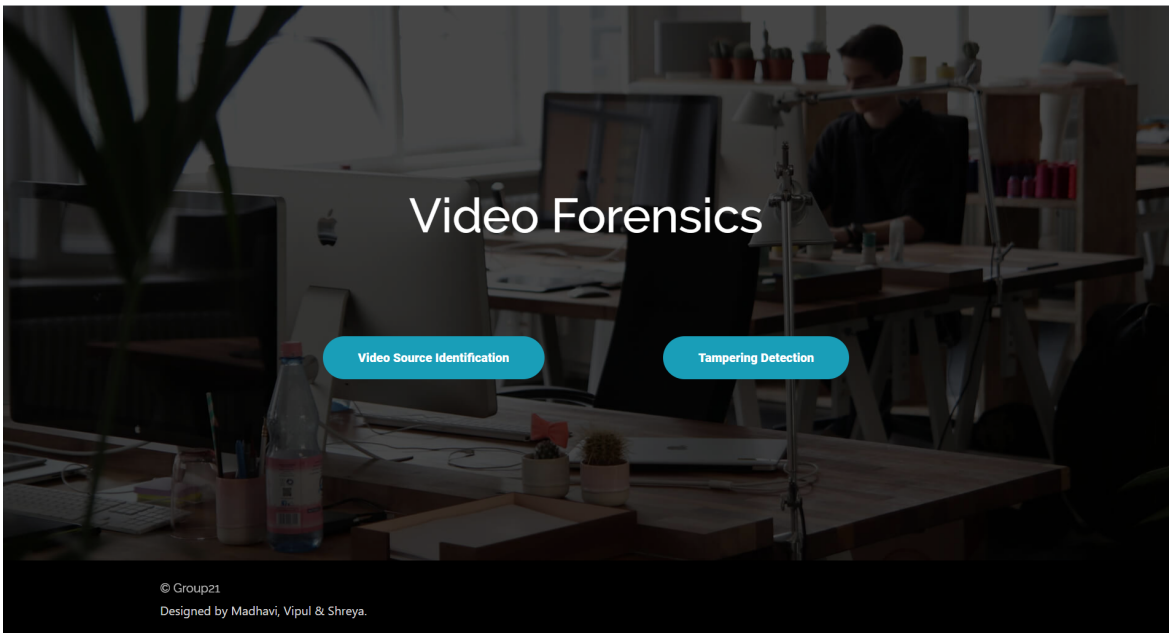


Figure 5.1: Main page

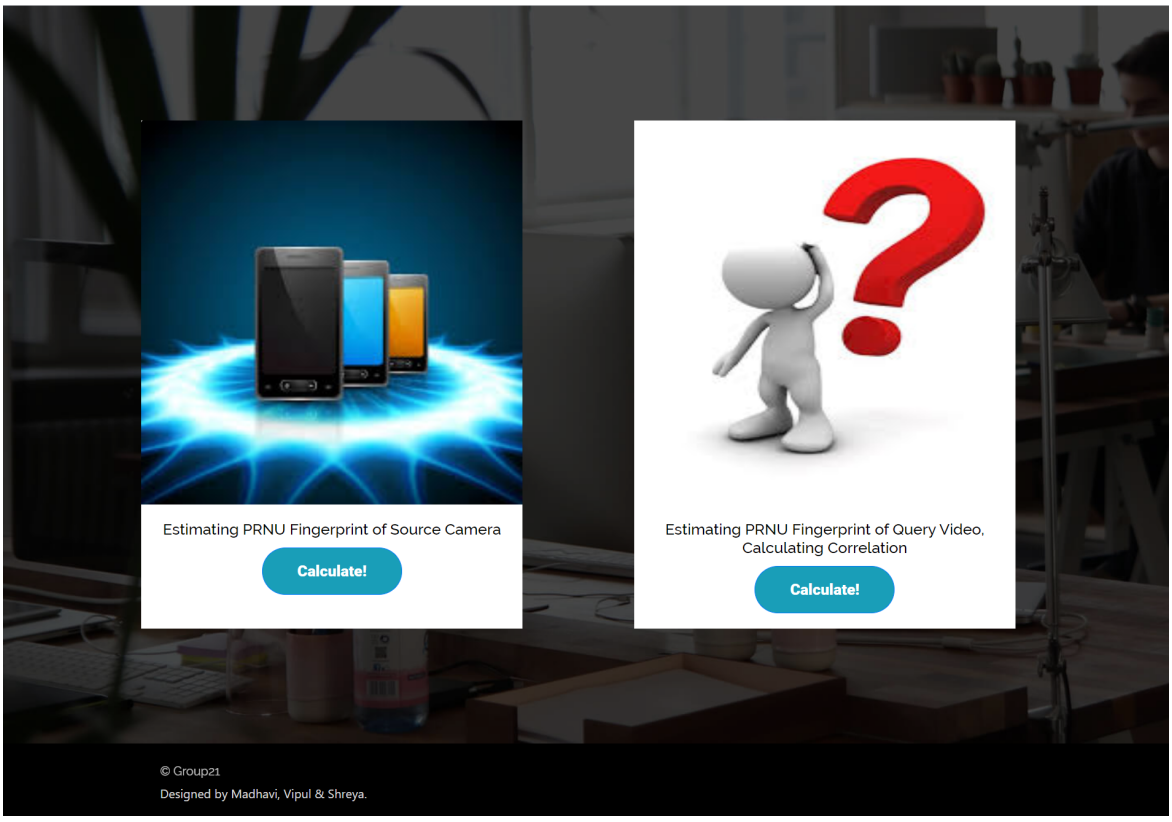


Figure 5.2: PRNU module

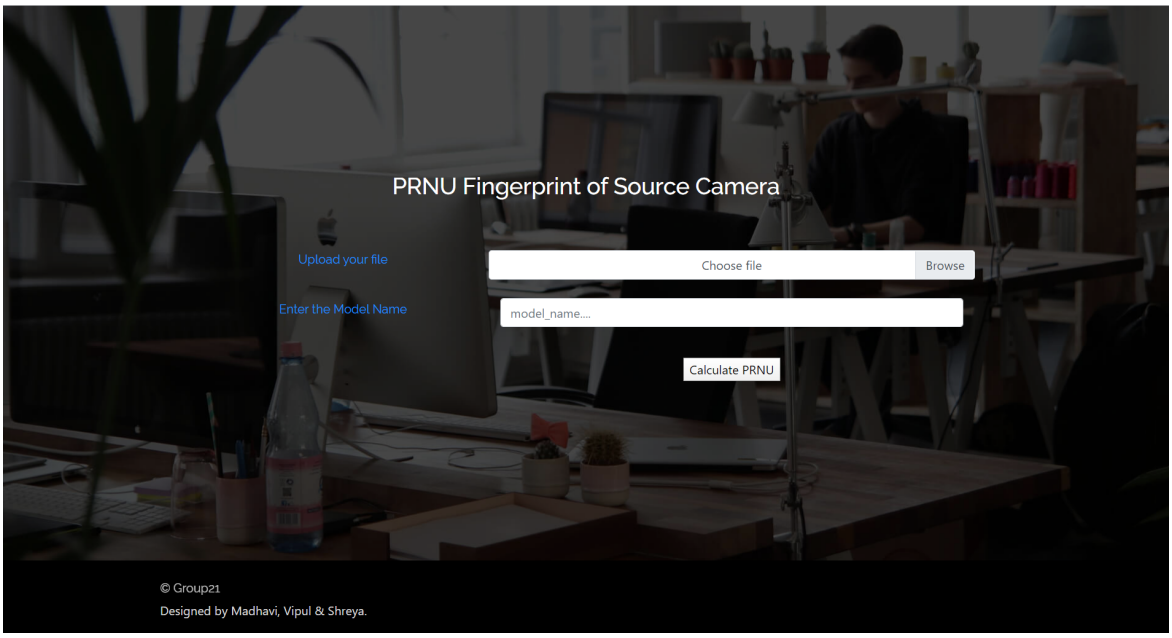


Figure 5.3: Fingerprint calculation of Source Camera

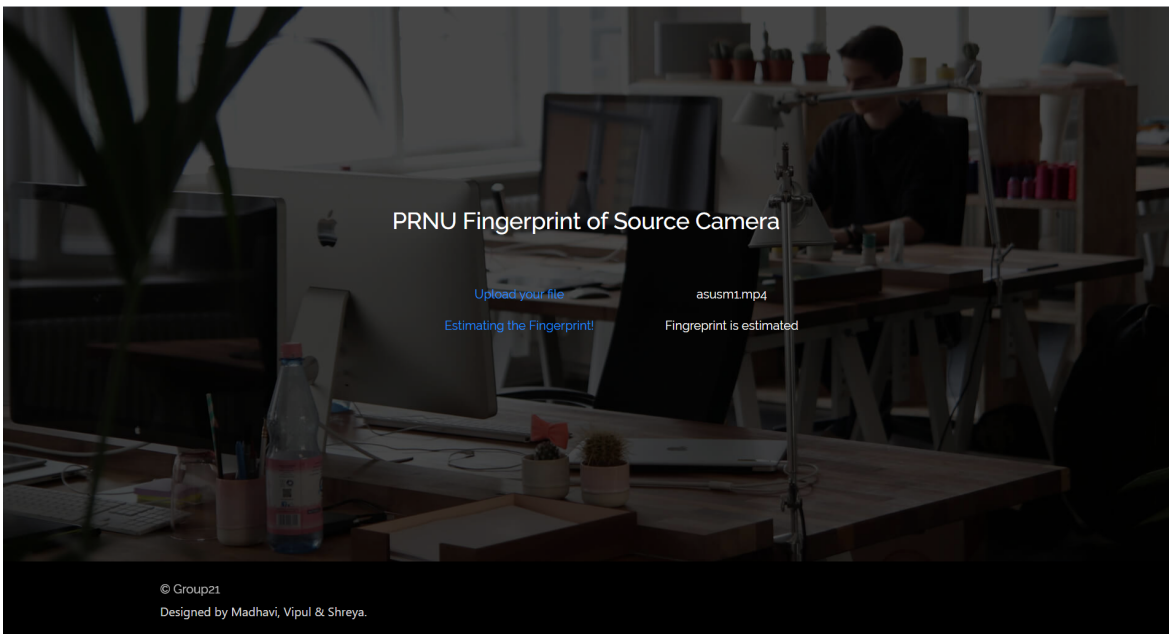


Figure 5.4: Fingerprint estimation of Source Camera

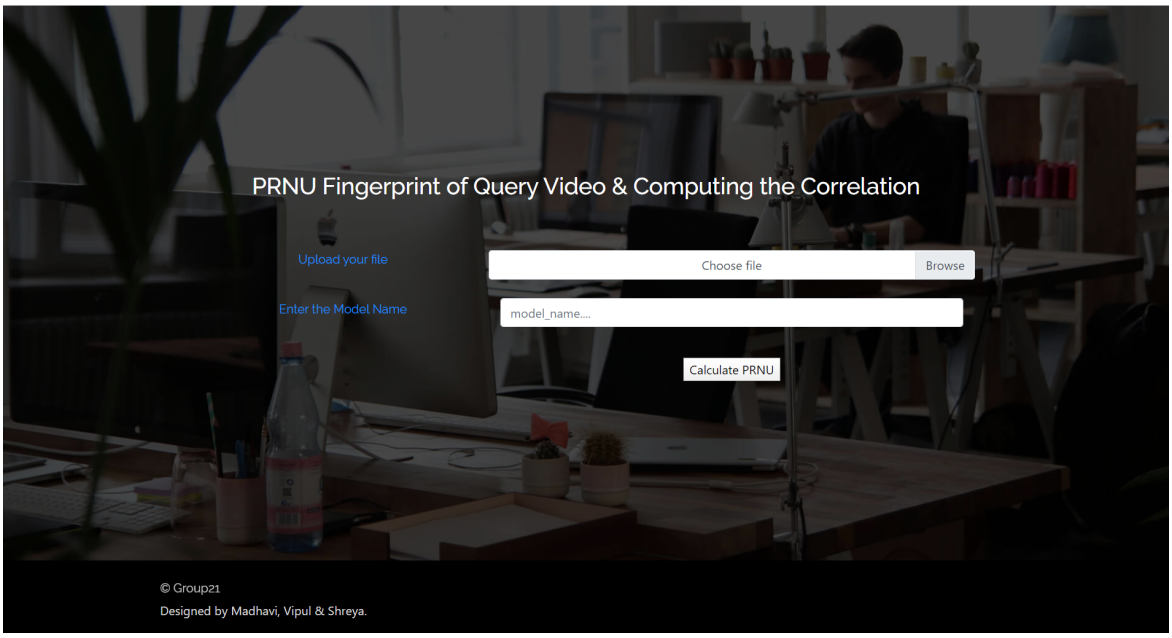


Figure 5.5: Fingerprint estimation of Query Video

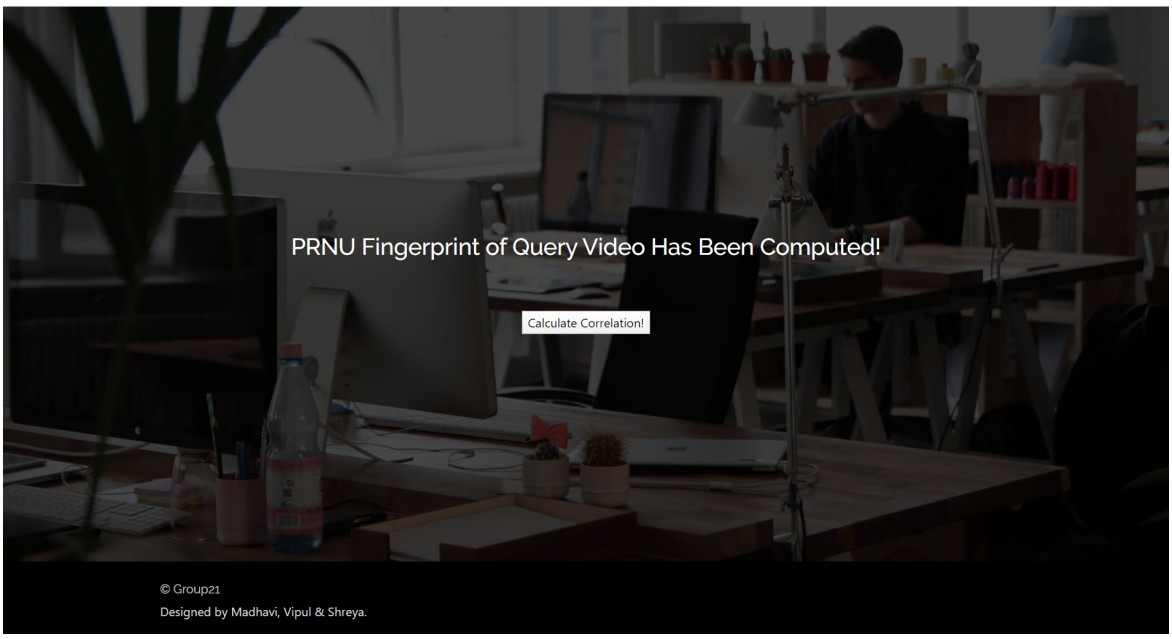


Figure 5.6: Correlation Estimation of Query video

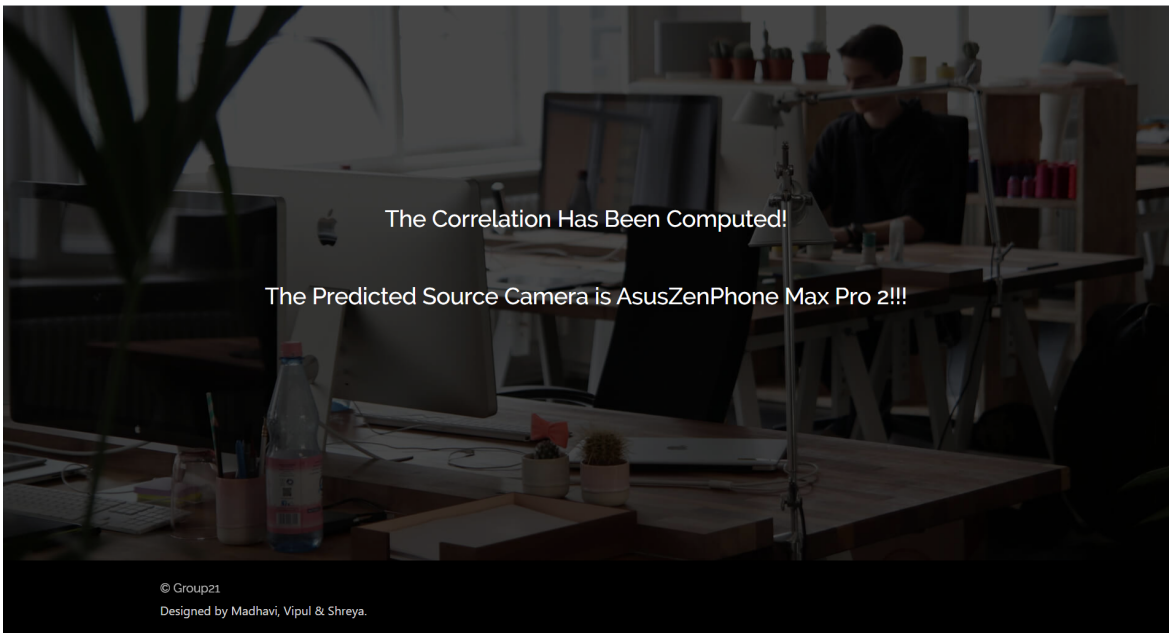


Figure 5.7: Source Camera Identification

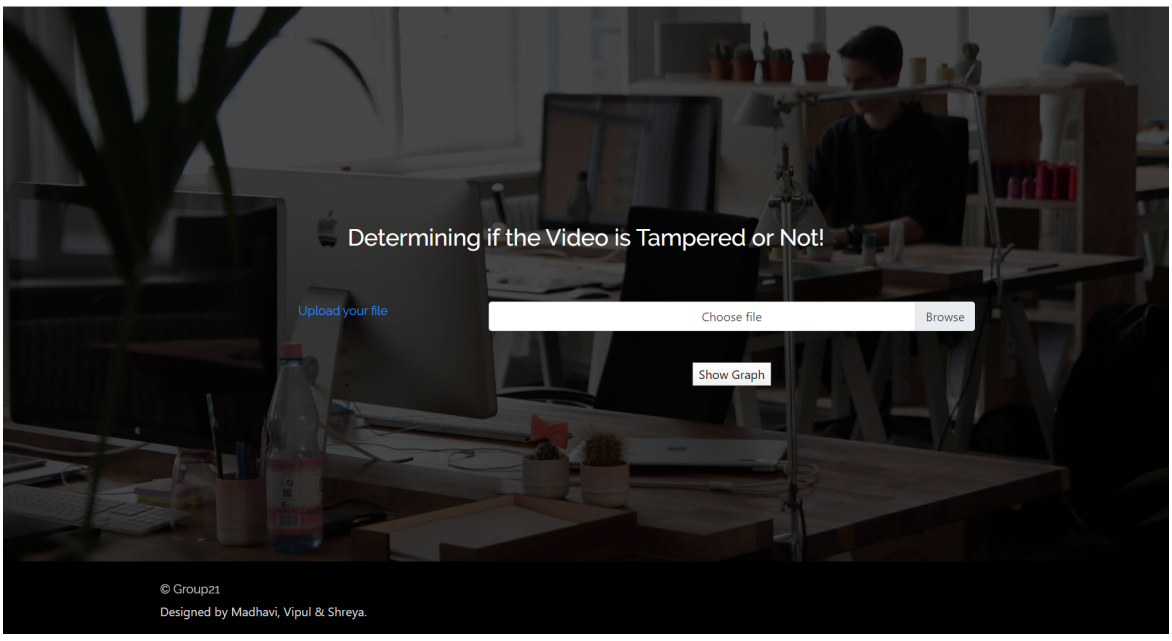


Figure 5.8: Main page of Tampering Detection Module

Fig 5.9 shows that there is spike in graph when frame is inserted or deleted from the graph Fig 5.10 shows that graph is nearly linear and there is not sudden increase in graph when video is authentic

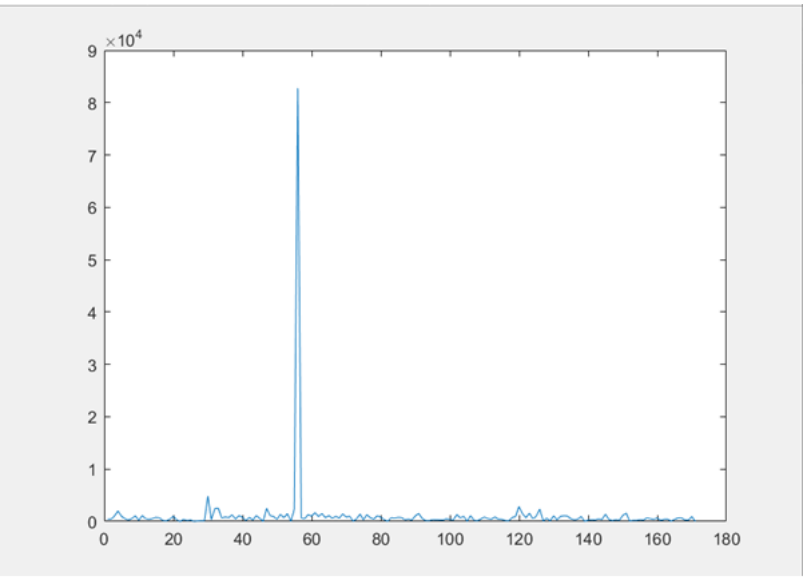


Figure 5.9: Frame Vs Changes in Pixels (video is tampered)

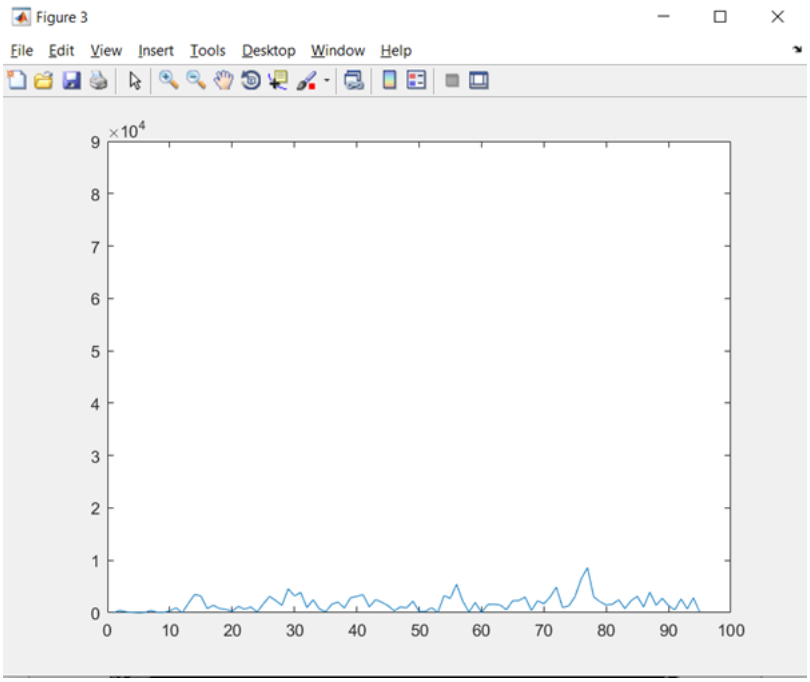


Figure 5.10: Frame Vs Changes in Pixels (video is authentic)

5.2 Sample Code (of imp part/ main logic)

5.2.1 PRNU

```
    images= []
    for filename in os.listdir(folder):
        img=cv.imread(os.path.join(folder,filename))
        if img is not None:
            images.append(img)
    print('block 1')
    imagegray=[]
    for img in images:
        imagegray.append(cv.cvtColor(img,cv.COLORBGR2GRAY))
    print('block 2')
    imagedenoise=[]
    for img in imagegray:
        dst=cv.fastNlMeansDenoising(img,None,9,13)
        imagedenoise.append(dst)
    print('block 3')
    for i in range(len(images)):
        temp1=(imagedenoise[i]*imagegray[i])
        temp2=(imagegray[i])2
    k=temp1/temp2
    print(k)
    return k
```

Listing 5.1: PRNU Fingreprint Calculation

5.2.2 Tampering Detection

```
    function [nFrames,a]= filetemp(filename)
    %............ TAMPER DETECTION ..........%
    %......... Reading Video into vid Object .........%
    obj = VideoReader(filename);
    image = read(obj, 1);
    binarization_threshold = 10;
    nFrames = obj.NumberOfFrames;
    nFrames
    rate = obj.FrameRate;
    rate
    %figure, imshow (image);
    %........ Morphological Operation on Image .......%
    image = rgb2gray(image);
    [x y] = size(image);
    [~, threshold] = edge(image, 'sobel');
    fudgeFactor = .8;
    image = edge(image,'sobel', threshold * fudgeFactor);
    figure, imshow (image);
    image = edge (image, 'sobel');
    se = strel('line',11,90);
    im2 = imdilate(image, se);
    figure, imshow (im2);
    imshow (im2);
    %........ Taking complement of the Image ..............%
    countin =0;
    countout = 0;
    ind = 0;
    a = zeros (1, nFrames - 2);
```

```
    for i = 1:x
            for j = 1:y
                im2(i,j) = 1- im2(i,j);
            end
    end
    maxTillNow = 0;
    %........ Processing Images for Tamper Detection ........%
    for t = 2:1:nFrames - 1
        countin =0;
        countout = 0;
        n = 0;
        nMinusOne = 0;
        im = read (obj, t);
        %[x, y] = size(image)
        im = rgb2gray(im);
        [~, threshold] = edge(im, 'sobel');
        fudgeFactor = .8;
        im = edge(im,'sobel', threshold * fudgeFactor);
        se = strel('line',11,90);
        im = imdilate(im, se);
        %figure,imshow (im);
        for i = 1:x
            for j = 1:y
                im(i,j) = 1 - im(i,j);

                if (im2(i, j) == 1 && im(i,j) == 0)
                    countout = countout + 1;
                end
                if (im2(i,j) == 0 && im(i, j) == 1)
                    countin = countin + 1;
                end
            end
        end
        if (mod (t,5) == 0)
        %figure,imshow (im);
        end
        %countout
        %countin
        im2 = im;
        %c = min (countout, countin);
        c = countout - countin;
        if (c < 0)
            c = -c;
        end
        ind = ind + 1;
        a(ind) = c;
    end

    maxTillNow
```

Listing 5.2: Frame relation graph generation

5.3 Testing

5.3.1 Unit Testing

Unit Testing is a software testing level where each individual component of the software is tested, in order to validate that each unit of the software performs as is originally intended.

Test Objective: There are basically two modules in our project- Video Source Identification and Tampering Detection.

Fig 5.11 shows that unit is working for calculating PRNU value for source camera.

Fig 5.12 shows that unit is working for calculating PRNU value for Query Video

```
F:\project\video-forencis\fyp1\model\qwerty
Successfully created the directory F:\project\video-forencis\fyp1\model\qwerty
Done
in prnu
block 1
block 2
block 3
[[0.07086614 0.07086614 0.07086614 ... 0.54189944 0.54189944 0.54189944]
 [0.07086614 0.07086614 0.07086614 ... 0.54189944 0.54189944 0.54189944]
 [0.07086614 0.07086614 0.07086614 ... 0.54189944 0.54189944 0.54189944]
 ...
 [0.48214286 0.48214286 0.48214286 ... 1.22516556 1.22516556 1.22516556]
 [1.49557522 1.49557522 0.48214286 ... 0.52702703 1.22516556 1.22516556]
 [0.23728814 1.49557522 1.49557522 ... 0.52702703 0.52702703 0.52702703]]
prnu finish
[17/Apr/2020 21:25:46] "POST /prnu11 HTTP/1.1" 200 4327
```

Figure 5.11: PRNU value of Source Camera

```
F:\project\video-forencis\fyp1\model\vid1
Successfully created the directory F:\project\video-forencis\fyp1\model\vid1
Done
in prnu
block 1
block 2
block 3
[[1.3015873  1.3015873  1.3015873  ... 1.21666667 0.17073171 0.23076923]
 [0.37190083 0.37190083 0.37190083 ... 1.11111111 1.21666667 0.17073171]
 [1.51666667 1.51666667 1.51666667 ... 0.07086614 1.11111111 1.11111111]
 ...
 [1.23611111 0.2244898  0.98630137 ... 2.11764706 1.17525773 1.17525773]
 [0.46206897 1.23611111 0.2244898  ... 0.125      0.125      0.125     ]
 [0.66887417 1.41333333 1.23611111 ... 1.67676768 1.67676768 1.67676768]]
prnu finish
[18/Apr/2020 20:42:02] "POST /prnu11 HTTP/1.1" 200 4325
```

Figure 5.12: PRNU value of Query Video

5.3.2 Integration Testing

Integration Testing is a software testing level where the individual components of the software are, one by one, integrated together, and tested as a group. This is done in order to expose faults in the interaction between the units that are combined together.

Fig 5.13 shows that model is calculating the correlation between source camera and query video and shows that it detecting source camera model

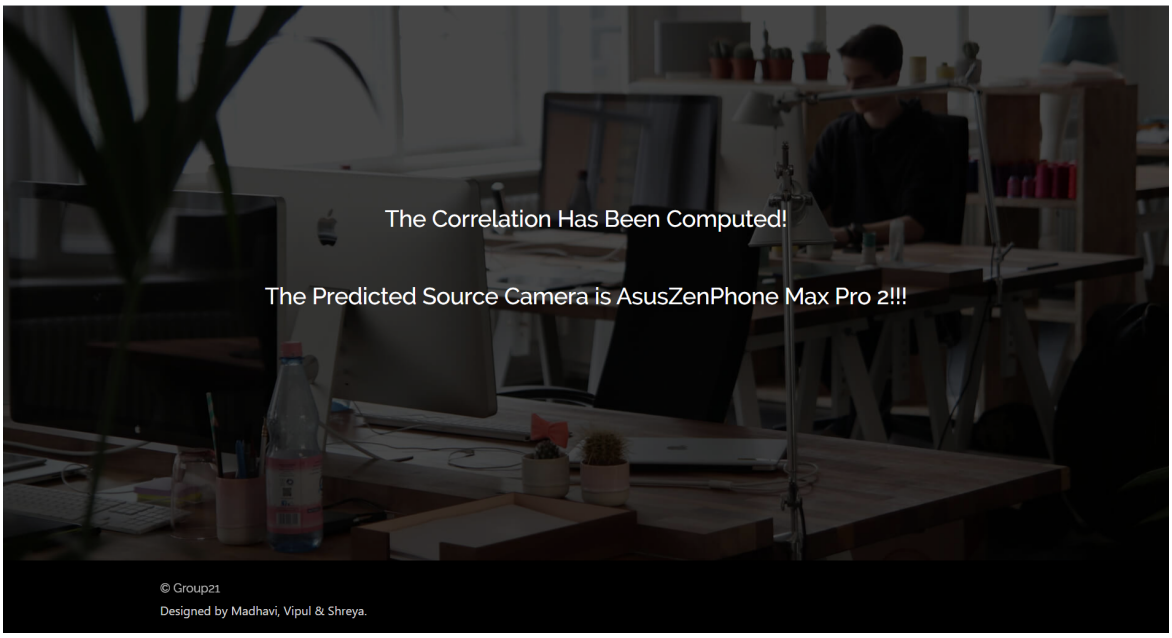


Figure 5.13: Source Camera Identification

5.3.3 Black-box Testing

Black-box Testing is a software testing method that only focuses on the input provided to the system and the output displayed by the system. It does not focus on the internal knowledge of the application or the code.

Table 5.1: Study of different video samples

	Mobile Brands		
	Asus Zenphone max	Realme 3 pro	Redmi Note 7 pro
video 1	0.035	-0.0027	-0.0043
video 2	-0.017	0.043	-0.23
video 3	-0.13	-0.035	0.06

For black box testing As shown in the 5.1 3 video samples are taken and correlation between fingerprint of video and fingerprint of 3 different cameras is calculated. Since video 1 ,2 and 3 are taken from Asus Zenphone max, Realme 3 pro and Redmi Note 7 pro respectively; the correlation in the respective columns is maximum and greater than 0.002. For example, for video 1, the correlation value for Asus Zenphone max is 0.035 (0.02) and is thus, the query video is considered to be taken from this phone camera

Chapter 6

Conclusion & Future Scope

In this report, the study of two different approaches of Video Source Identification and Tampering Detection in the domain of video forensics is presented. Two algorithms namely, the Photo Response Non-Uniformity (PRNU) Algorithm and the Edge Detection Algorithm have been used for the identification of the source phone camera and to determine whether the video has been tampered with, respectively. The comprehensive study of various techniques mentioned above is presented in this report. Upon uploading a video, the user can determine the source of the query video and if the video is tampered. The VISION dataset, which is a standard dataset used for the purpose of image and video forensics is used in our project. The techniques used in this project fall under the cybersecurity domain and can be used by the Police IT cell to provide the chain of acquisition of a video in a law court and to identify fake videos which circulate amongst the public. This project can be further extended by identifying further ways in which a video can be tampered, like object insertion/ duplication/ deletion. Also, the project can be extended further by identifying deep fakes in the video sequence. Further, one can look for ways to further optimize the algorithms we have used.

References

[1] E. K. Kouokam and A. E. Dirik, "Prnu-based source device attribution for youtube videos," *Digital Investigation*, vol. 29, pp. 91–100, 2019.

[2] E. Altınışık, K. Taşdemir, and H. T. Sencar, "Mitigation of h. 264 and h. 265 video compression for reliable prnu estimation," *arXiv preprint arXiv:1905.09611*, 2019.

[3] E. Altinisik, K. Tasdemir, and H. T. Sencar, "Extracting prnu noise from h. 264 coded videos," in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 1367–1371, IEEE, 2018.

[4] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "Vision: a video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 15, 2017.

[5] Y. Chen and V. L. Thing, "A study on the photo response non-uniformity noise pattern based image forensics in real-world applications," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, p. 1, The Steering Committee of The World Congress in Computer Science, Computer ..., 2012.

Acknowledgement

Success of a project like this involving high technical expertise, patience and massive support of guides, is possible when team members work together. We take this opportunity to express our gratitude to those who have been instrumental in the successful completion of this project. We would like to show our appreciation to **Ms. Shweta Tripathi** for their tremendous support and help, without them this project would have reached nowhere. We would also like to thank our project coordinator **Mrs. Rakhi Kalantri** for providing us with regular inputs about documentation and project timeline. A big thanks to our HOD **Dr. Lata Ragha** for all the encouragement given to our team. We would also like to thank our principal, **Dr. S. M. Khot**, and our college, **Fr. C. Rodrigues Institute of Technology, Vashi**, for giving us the opportunity and the environment to learn and grow.

Project Group Members:

1. Madhavi Bhilegaokar, 101604

_____

2. Vipul Borhade, 101605

_____

3. Shreya Pai, 101637

_____

Appendix A : Timeline Chart

Name	Begin date	End date
Abstract	1/7/19	12/7/19
Introduction	15/7/19	15/7/19
Literature Survey	16/7/19	22/7/19
Proposed System	23/7/19	26/7/19
Scope	29/7/19	29/7/19
H/W & S/W	30/7/19	5/8/19
Block diagram	6/8/19	23/8/19
Activity and deployment diagram	26/8/19	26/8/19
PRNU of Image	27/8/19	16/9/19
PRNU of source camera	17/9/19	23/9/19
PRNU of query video and correlation	24/9/19	13/1/20
Backend implementation for tampering	14/1/20	24/1/20
Frontend implementation	27/1/20	7/2/20
Django integration frontend	10/2/20	21/2/20
Django integration backend prnu	24/2/20	28/2/20
Django integration backend tampering	2/3/20	10/3/20

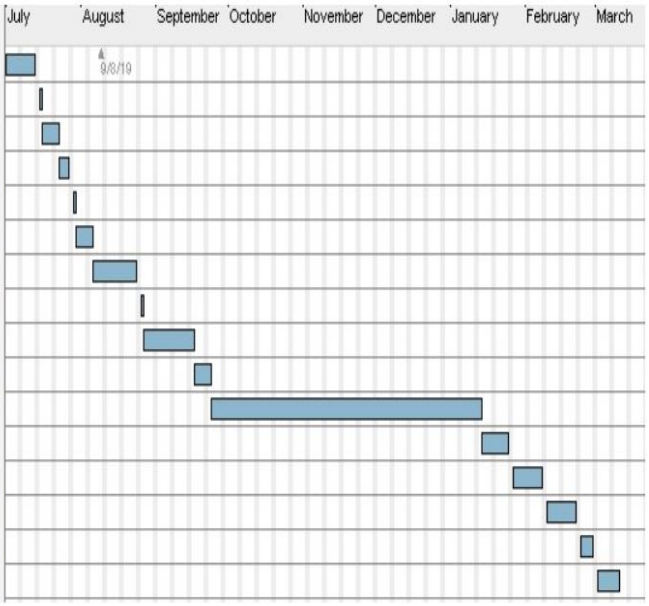


Figure 6.1: Timeline Chart