A FACIAL RECOGNITION-BASED VIDEO ENCRYPTION APPROACH TO PREVENT FAKEDEEP VIDEOS

Alex Liang[1], Yu Su[2] and Fangyan Zhang[3]

[1]St. Margaret's Episcopal School, San Juan Capistrano, CA 92675
[2]Department of Computer Science, California State Polytechnic University,
Pomona, CA, 91768
[3]ASML, San Jose, CA, 95131

ABSTRACT

Deepfake is a kind of technique which forges video with a certain purpose. It is in urgent demand that one approach can defect if a video is deepfaked or not. It also can reduce a video to be exposed to slanderous deepfakes and content theft. This paper proposes a useful tool which can encrypt and verify a video through proper corresponding algorithms and defect it accurately. Experiment in the paper shows that the tool has realized our goal and we can put it into practice.

KEYWORDS

Video Encryption, Video Verification, Encryption Algorithm, Decryption algorithm

1. INTRODUCTION

Deepfakes [3][4][5] have become a more prevalent problem, with no solution. Not only is there an inability to determine whether or not videos are deepfaked, the amount of people researching deepfake detection are far outnumbered by those researching deepfake synthesis. Deepfakes have real world implications. For example, a video of Gabon's president was decried as a deepfake, nearly resulting in a military coup. deepfakes have begun to extend to America as well, as videos of politicians and influential figures are being falsified, like a video which slows Nancy Pelosi's speech. Some researchers are trying to use A.I. to detect if a video has been edited, others wish to use blockchain to detect deepfakes.

In order to verify digital content for authenticity and avoid slanderous deepfakes [6][7]. Gyfcat has implemented a system in which, after flagging a video under suspicion that it is a deepfake, it combs its database to search for similar videos.

However, software that can spot AI-manipulated videos will only ever provide a partial fix to this problem. As with computer viruses or biological weapons, the threat from deepfakes is now a permanent feature on the landscape. And although it's arguable whether or not deepfakes are a huge danger from a political perspective, they're certainly damaging the lives of women here and now through the spread of fake nudes and pornography.

Natarajan Meghanathan et al. (Eds) : CSEIT, CMLA, NeTCOM, CIoT, SPM, NCS, WiMoNe, Graph-hoc - 2019
pp. 203-208, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91317

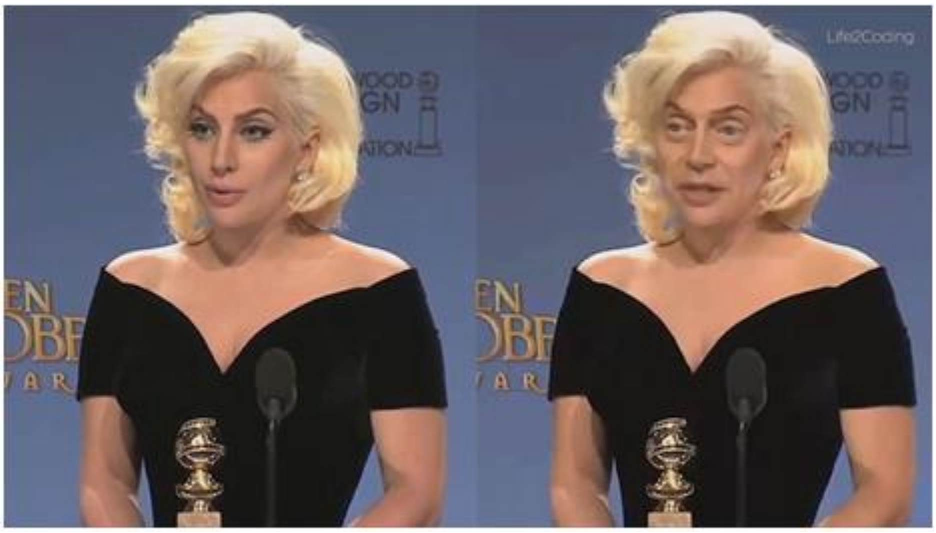

Figure 1: an example of deepfake (left: real, right: deep faked)

We are trying to do something similar; our goal is to add a mark/encrypted message onto the video before it is released, so we can detect if it has been edited later on. First, we change some pixels to certain values, before utilizing facial-detection to change a pixel, relative to the face. When we are trying to detect, the software will check if the 'face pixel' and 'static pixels' are the correct color, and if they are, the video is considered 'real' [10][11].

Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. Challenges

To implement this system, there are mainly two challenges that we have to overcome.

2.1. Choose Proper Algorithm to Encrypt

Currently, there are various algorithms can encrypt a video. Considering the efficiency and effects, we should to compare those existing encryption algorithms and choose optional solution. For example, a 2 second video takes around 15 seconds to encrypt. Users can not accept so long time to finish encryption for a large video. When choosing encryption algorithm, we have to consider all aspects, such as accuracy, quality, and efficiency.

2.2. Video Verification

After encrypting a video and finish transferring over internet, how to decrypt and verify the video is equally important. However, facial recognition does not have 100% consistence for encrypting. We have to overcome this challenge and make it more consistent in encryption. In addition, a video usually is compressed when sending out, which may fail in verification due to compression issue.

3. Solution

Figure 2 shows an overview of the proposed approach. First, we change some pixels to certain values, before utilizing facial-detection to change a pixel, relative to the face. When we are trying to detect, the software will check if the 'face pixel' and 'static pixels' are the correct color, and if they are, the video is considered 'real'.

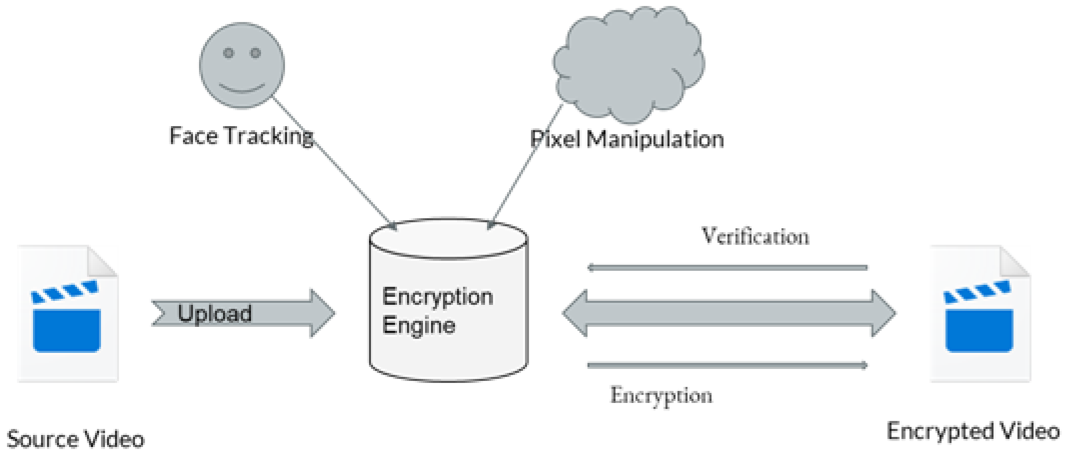

Figure 2: Overview of the solution

3.1. OpenCV

About. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products [8].

To build our face recognition system, we will first perform face detection, extract face embeddings from each face using deep learning, train a face recognition model on the embeddings, and then finally recognize faces in both images and video streams with OpenCV [9].

Like a series of waterfalls, the OpenCV cascade breaks the problem of detecting faces into multiple stages. For each block, it does a very rough and quick test. If that passes, it does a slightly more detailed test, and so on. The algorithm may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.

The advantage is that the majority of the picture will return a negative during the first few stages, which means the algorithm won't waste time testing all 6,000 features on it. Instead of taking hours, face detection can now be done in real time.

3.2. Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Two HTTP APIs have been implemented using the Flask: 1) encrypt the video; 2) verify the video.

3.3. Frontend

Figure 3 shows the basic frontend UI. It has been implemented using HTML and CSS.

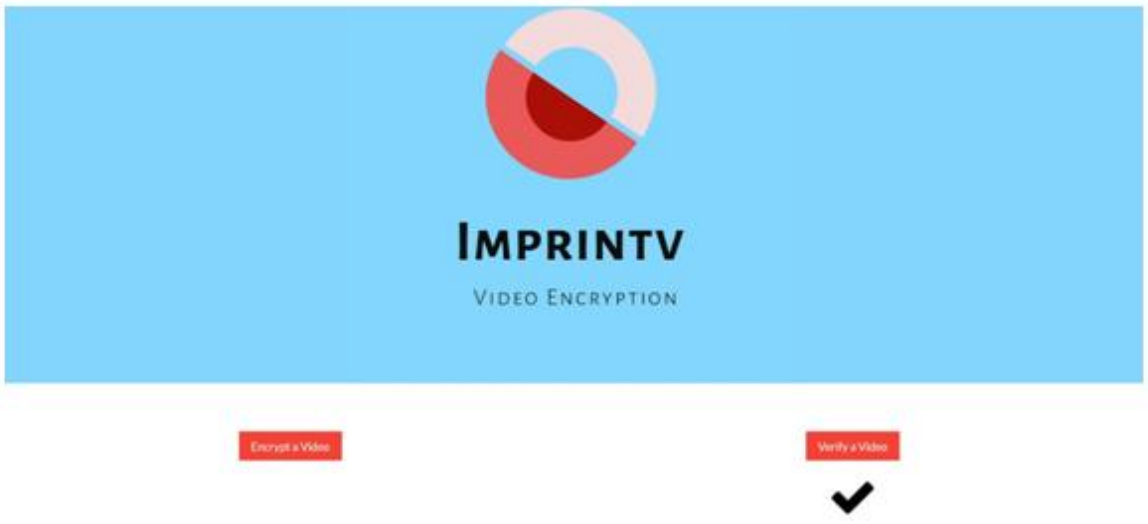

Figure 3. The Frontend Web UI

HTML - HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

CSS - Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. Although most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

3.4. Video Encryption

Lossless compression techniques, as their name implies, involve no loss of information. If data have been losslessly compressed, the original data can be recovered exactly from the compressed data. Lossless compression is generally used for applications that cannot tolerate any difference between the original and reconstructed data.

We created a simple encryption algorithm to serve as a placeholder, but we are planning on using steganography to create a well-hidden message. The file size is too big from lossless compression, and, the file type is limited to .avi files. We're working on utilizing lossy compression, but that's something for later.

4. Experiments

This tool allows us to encrypt videos before release and verify them in other side to detect whether it has been edited or not. In this experiment, we applied the tool to one sample video, encrypting and verifying it (see Figure 3). We marked the frame, as well as faces every time in progress.

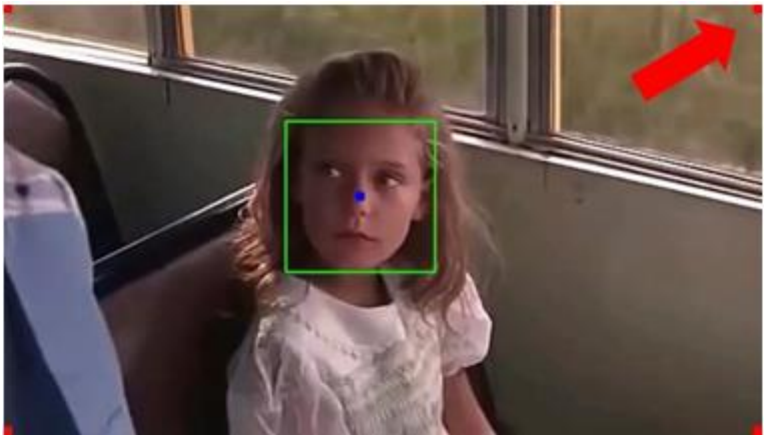

Figure 3: An experiment of encryption and decryption

5. Related Work

D. Güera and E. J. Delp [1] proposed an AI-based approach to detect deepfake videos. Just like other approaches, the major disadvantage of this approach is that once the hackers understand the algorithm, they can always create a counter approach to skip the detection process. Yuezun Li [2] focuses on using an image processing technique to detect the obvious edges formed by the deepfake videos. However, as the deepfake video become more and more sophisticated, it is very difficult to guarantee the accuracy of this approach in the long run, as a number of deepfake videos cannot be judged by human eyes.

6. Conclusions and Future Work

In this project, we proposed a video encryption approach to secure video after release. This tool uses a high efficiency video encryption algorithm to encrypt videos before release. After release, they can be verified through decryption algorithm to defect if the video has been edited or not. This tool helps us test the "authenticity" of those videos and avoid slanderous deepfakes and prevent content theft to some extent. In addition, it also secure online identity.

In the future, we will investigate other video encryption algorithms to keep improving the accuracy and efficiency. We also would like to explore the possibility of automaticity in video encryption and decryption.

REFERENCES

[1] D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.

[2] Exposing DeepFake Videos By Detecting Face Warping Artifacts Yuezun Li, Siwei Lyu Computer Science Department University at Albany, State University of New York, USA

[3] Ruchansky, N., Seo, S., & Liu, Y. (2017, November). Csi: A hybrid deep model for fake news detection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (pp. 797-806). ACM.

[4] Polletta, F., & Callahan, J. (2019). Deep stories, nostalgia narratives, and fake news: Storytelling in the Trump era. In Politics of meaning/meaning of politics (pp. 55-73). Palgrave Macmillan, Cham.

[5] Singhania, S., Fernandez, N., & Rao, S. (2017, November). 3han: A deep neural network for fake news detection. In International Conference on Neural Information Processing (pp. 572-581). Springer, Cham.

[6] Güera, D., & Delp, E. J. (2018, November). Deepfake video detection using recurrent neural networks. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). IEEE.

[7] Citron, D. K., & Chesney, R. (2018). Deep Fakes: A Looming Crisis for National Security, Democracy and Privacy?. Lawfare.

[8] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.".

[9] Pulli, K., Baksheev, A., Kornyakov, K., & Eruhimov, V. (2012). Real-time computer vision with OpenCV. Communications of the ACM, 55(6), 61-69.

[10] Li, Y., & Lyu, S. (2018). Exposing deepfake videos by detecting face warping artifacts. arXiv preprint arXiv:1811.00656, 2.

[11] Cozzolino, D., Thies, J., Rössler, A., Riess, C., Nießner, M., & Verdoliva, L. (2018). Forensictransfer: Weakly-supervised domain adaptation for forgery detection. arXiv preprint arXiv:1812.02510.

[12] Dolhansky, B., Howes, R., Pflaum, B., Baram, N., & Ferrer, C. C. (2019). The Deepfake Detection Challenge (DFDC) Preview Dataset. arXiv preprint arXiv:1910.08854.