COMPARISON OF WATERMARK METHODS TO DETECT DEEPFAKES

COMPARISON OF WATERMARK METHODS TO DETECT DEEPFAKES

A MS Course Project submitted in partial
fulfillment of the requirements for the degree of
Master of Science

By

Odyssey Villagomez
Bachelor of Science in Biology

December 2022

ABSTRACT

Fake imaging and videos have exploded into popularity over the past 5 years since the release of the first deepfake video in 2017. These deepfake media increasingly pose a threat to society by questioning the integrity and authenticity of images. This project compares two existing watermarking techniques that have developed to not only detect deepfake images, but to also defend against them proactively This study compares the two watermarking methods to test which performs better on image manipulation detection as well as their ability to prevent deepfake images. The first study, "Neekhara, et al., FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes" had an AUC score of 0.9992, when classifying real or fake images, while the second study, "CMUA-Watermark: A Cross-Model Universal Adversarial Watermark for Combating Deepfakes" had an AUC score of 0.5. The first study demonstrates how a pre-trained watermark encoder-decoder model can detect deepfake images with high accuracy. This mechanism allows for anyone to recognize fake or real imaging by looking for a watermark.

https://github.com/OdysseyV/MS_Project_Comparison_Watermarking_Deepfakes

This Project Report is approved for recommendation to the Graduate Committee.

Project Advisor:

_____

Haadi Jafarian, Ph.D.

TABLE OF CONTENTS

1. Introduction 1

1.1 Problem 1

1.2 Project Report Statement 2

1.3 Approach 2

1.4 Organization of this Project Report 4

2. Background 5

2.1 Key Concepts 5

2.1.1 Deepfakes 5

2.1.2 Watermarking 7

2.2 Related Work 8

2.2.1 Semi-Fragile Watermarks 8

2.2.2 Disruptive Watermarks 9

3. Architecture 11

3.1 High Level Design 11

3.2 Study Specific Metrics 13

3.3 Implementation 15

4. Methodology, Results and Analysis 17

4.1 Methodology 17

4.2 Results 18

4.3 Analysis 19

5. Conclusions 23

5.1 Summary 23

5.2 Contributions 23

5.3 Future Work 24

References 25

Appendix A. Additional Figures 28

LIST OF FIGURES

Figure 1: Different L2 values and their corresponding distortion 12

Figure 2: The testing pipeline used for each watermarking method..... 11

Figure 3: The embedding, transformation, and recovery of a watermark 12

Figure 4: The embedding, transformation, and recovery of a watermark 13

Figure 5: Calculation of L2 score 15

Figure 6: Confusion matrices for FaceSigns watermark classification 19

Figure 7: Confusion matrices for CMUA-watermark classification..... 19

Figure 8: The original, watermarked, and watermark for an image using FaceSigns 28

Figure 9: The original, watermarked, and watermark for an image using CMUA..... 28

Figure 10: Benign vs. malicious watermark decoding using FaceSigns 28

Figure 11: Target image, original image and deepfake using CMUA..... 29

Figure 12: Benign image transformations on a CMUA watermarked image 29

Figure 13: Target image, original image with watermark and deepfake using CMUA ..29

1. INTRODUCTION

1.1 Problem

Deepfake videos and imaging have garnered attention with cybersecurity professionals due to their recent advancements in generating fake media (Yu, Xia, Fei, & Lu, 2020). Deepfake technology uses deep learning methods like generative adversarial networks (GANs) to create realistic fake images and videos of a target subject (Yu, Xia, Fei, & Lu, 2020). This is usually done by feeding the deep learning model images of the target victim, or in some cases using a model pretrained on image datasets to generate a completely fake person in an image or video. Deepfakes first surfaced when a malicious video targeting a celebrity was released on Reddit in 2017 (Yu, Xia, Fei, & Lu, 2020). The artificial intelligence (AI) used to generate deepfakes has been quickly advancing, with frameworks becoming available for non-professionals to generate deepfakes (Yu, Xia, Fei, & Lu, 2020), extending the damage that can be done. Some of these frameworks include Deepfakes and DeepNude. Meanwhile, the detection methods against deepfakes have been mainly isolated to AI detection models where real or fake images are fed into a trained model to identify if the media is real or fake. These AI detection models are prone to overfitting due to their approach to detection as a classification problem (Yu, Xia, Fei, & Lu, 2020). These deepfake media present many challenges to cybersecurity professionals, like violating the integrity and authenticity of original images.

Watermarking techniques have recently been evaluated for their effectiveness against preventing images from being used to generate deepfake media (Lv, 2021), (Neekhara, et al., FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes, 2022), (Yang, Liang, He, Cao, & Zhenqiang Gong, 2021).

Watermarking is interesting in its approach against deepfakes due to its proactive instead of reactive nature. Watermarking, historically used to protect digital media from copyright infringement, is the practice of embedding information into media that cannot be removed without altering the original media (Lian & Zhang, 2009). The watermarked media is then recovered or examined later when the authenticity or integrity of an image is questioned. The embedded watermark is recovered, and if it matches the original watermark, the media is said to be authentic. Watermarking provides a unique advantage against deepfakes because of its proactive defense as well as its ability to detect completely new faces, as they will not have any watermark on them (Yang, Liang, He, Cao, & Zhenqiang Gong, 2021).

1.2 Project Statement

This article compares two watermarking methods used to defend against deepfakes. These methods use deep learning to train a model to create a watermark that is embeddable in any image. The special property of these watermarks are that they should be robust to common image changes like color and light adjustments while also being sensitive to malicious facial changes (Neekhara, et al., FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes, 2022), (Lv, 2021), (Yang, Liang, He, Cao, & Zhenqiang Gong, 2021).

1.3 Approach

Two studies will be analyzed for their effectiveness in detecting deepfake manipulations. These studies include “FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication*”* (Neekhara, et al., FaceSigns: Semi-Fragile Neural Watermarks

for Media Authentication and Countering Deepfakes, 2022) and "Countering Deepfakes, and CMUA-Watermark: A Cross-Model Universal Adversarial Watermark for Combating Deepfakes*"* (Huang, et al., 2021).

Each study uses its own method to create a unique watermark, which is then embedded in each original image. The pre-trained watermarks will be tested against common color and light changes as well as input into a deepfake engine to produce malicious changes. To evaluate the robustness and fragility of the watermark, the watermark must be recovered or undisturbed during benign image transformations and unrecoverable or distorted during malicious image transformations. Whether the watermark is undisturbed or not will be used to classify the image as real or fake. This is a classification problem, so Area under the ROC Curve (AUC) and confusion matrices will be used to evaluate each watermarking model; the benign image watermark is classified as real and the malicious image watermark is classified as fake. The ability of the watermark to respond to image transformations correctly drives the resultant image classification. Recovery of the watermark during benign transformations measures robustness, while non-recovery of the watermark during malicious transformations measures fragility (Neekhara, et al., FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes, 2022).

Not only should the watermark display appropriate robustness and fragility, it also should be secure against copying, meaning that the watermark should provide authenticity to images. To protect the integrity of the watermark against modification, the watermark should be imperceptible to the human eye, so that it cannot easily be

replicated. The watermark should also be recoverable to provide authentication that the image is in fact an original.

1.4 Organization of this Project Report

Chapter 2 provides important background information that discusses the history of deepfakes, how they are generated, and what current methods exist to detect them, along with what specific watermarking methods have been discussed in the literature. Chapter 3 describes the approach taken by this paper to analyze two deepfake watermarking techniques . Chapter 4 discusses the methodology used to test the water marking models. It describes how the watermarks are generated and embedded into images, as well as how benign and malicious changes are applied. Finally, it discusses the metrics used to evaluate the models and their results. Chapter 5 examines what conclusions can be drawn from these results.

2. BACKGROUND

2.1 Key Concepts

The following sections cover key concepts the reader should understand for this report. Section 2.1.1 covers deepfakes, and Section 2.1.2 covers watermarking technology.

2.1.1 Deepfakes

Deepfakes as a technology pose many threats to the cybersecurity landscape by attacking the integrity and authenticity of images or videos. Deepfake technology has rapidly expanded over the past 5 years since the release of the first deepfake video in 2017 on Reddit. As artificial intelligence, specifically deep learning models, advance, so does the quality of deepfake technology. This is because deepfakes rely on deep learning generative adversarial networks (GANs); the better these networks become, the more realistic the deepfakes. In a GAN, there are two models that are trained to compete against each other. The goal of this adversarial process is for the generator model (creating fake media) to eventually fool the discriminator model, which is trained on detecting if media is real or fake. The result of this process is a model that can take images and create fake or sometimes completely new media that can't be differentiated from real images by either humans or machines. While this deepfake technology is not inherently malicious, it has been used for malicious purposes. A good example of this is the application Deepnude, which undresses a person in a photo.

Since the launch of deepfakes, companies and researchers alike have taken steps to analyze and combat them. Facebook recently partnered with Microsoft, MIT, and the

University of Oxford, to host a deepfake detection competition (Dolhansky, et al., 2020). This competition focused on creating a dataset of deepfake videos for the community to study and hosted a Kaggle competition for automatic deepfake detection models. The top five detection methods for this competition used a similar framework. They trained a machine learning model on facial detection features or frame-by-frame feature extractions and used ensemble methods to combine multiple machine learning models into one (Dolhansky, et al., 2020). This study highlights the trend in the community to approach the deepfake detection problem as a machine learning task.

Further research into the current literature on deepfakes also showed a significant focus on machine learning classification models as the main detection technique. These detection techniques train networks to discriminate between fake and real images, in what is defined as a binary classification problem. These methods focus on detecting inconsistencies generated during the deepfake creation process (Yu, Xia, Fei, & Lu, 2020). Yu et al., categorize these models based on the type of inconsistency being identified (Yu, Xia, Fei, & Lu, 2020). These categorizations include General-networks, temporal-consistency, visual-artefacts, camera-fingerprints, and biological-signals (Yu, Xia, Fei, & Lu, 2020). General-networks are a frame-level classification model, temporal-consistency detects inconsistencies between adjacent frames, visual-artifacts detect deepfake artifacts generated through blending manipulations, camera-fingerprints detect traces left in the image and differences between the background and the subjects' faces, and biological-signals detect signals that are hard to fabricate, like blinking frequently (Yu, Xia, Fei, & Lu, 2020) Out of these 5 categories, general-networks show the most promise, with Dang et al., showing significant improvement to general-networks

by using multi-task learning in addition to attention mechanisms to focus on detecting facial manipulation (Dang, Liu, Stehouwer, Liu, & Jain, 2020). While the Yu et al., study focused on deepfake video techniques, other surveys of deepfake technology have confirmed that image deepfake detection methods also focus on the same image abnormalities to train machine learning detection models (Zhang, 2022).

Unfortunately, most of these techniques are retrospective, analyzing photos for authenticity after they have been created, after which significant social damage has already been done. Additionally, any of these detection methods use the same deepfake GAN techniques, which creates a race between the defenders and attackers, to see who can train the better model. Some interesting work by Google Brain and the University of California, Berkeley has shown that deepfake detection systems can be attacked through white and black box attacks, where the models are fooled into misclassifying images. This is an attack specifically on a machine learned model, and the study resulted in the reduction of the AUC from 0.95 to 0.0005 for white box attacks and 0.95 to 0.22 for black-box attacks (Carlini & Farid, 2020). White box attacks are where the attacker has access to the specific model and black box attacks are when the attacker only has knowledge of the type of classifier used. This study highlights how the use of only machine learning detection methods can be vulnerable to specific attacks.

2.1.2 Watermarking

More recently, watermarking methods to proactively authenticate images as real or fake have surfaced. Watermarking has historically been used for copyright protection on digital media (Lian & Zhang, 2009), but recent literature evaluates its use in detecting manipulations used to generate deepfakes. For watermarks to be effective, Lian & Zhang,

who extensively cover techniques to protect digital media, suggest that the ideal watermark should be permanent and robust enough to respond to common signal transformations or deliberate attacks (Lian & Zhang, 2009). Additionally, many of the watermarking methods to detect deepfakes add that the watermark should hold up to common image transformations, like compression, color and light adjustments while resisting attempts to remove it (Neekhara, et al., FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes, 2022).

2.2 Related Work

This section covers important related work for deepfake watermarking techniques. There have been a small number of studies that have used watermarking to defend against deepfake engines. Across the studies, the watermark techniques have differed, with Tancik et al., and Zhu et al. embedding hidden watermarks to be recovered later and Ruiz et al., embedding a disruptive watermark that will prevent a deepfake engine from generating a deepfake, by blurring the output. Section 2.2.1 takes a closer look at embedding hidden information as semi-fragile and robust watermarks, and Section 2.2.2 describes how deepfake engines can be disrupted using watermarks.

2.2.1 Semi-Fragile Watermarks

There has been previous research developing semi-fragile watermarks. A semi-fragile watermark is sensitive to malicious changes but robust to benign ones. Zhu et al., propose HiDDeN, a deep neural network that can be used to encode and decode hidden messages in images (Zhu, Kaplan, Johnson, & Fei-Fei, 2018). They also add that the hidden message is robust to blurring, cropping and image compression. Tancik et al.

proposed StegaSamp, a recoverable hyper-link embedded imperceptibly in images (Tancik, Mildenhall, & Ng, 2019). StegaStamp is also built on a deep neural network to encode and decode the watermarks.

For the study analyzed in this paper, Neekhara et al. propose a watermark that not only is recoverable during benign image changes, but also unrecoverable during deepfake image changes. To measure how *unrecoverable* a watermark is, Neekhara et. al proposed embedding a secret value into the watermark, to later be recovered. Upon recovery, a bit recovery rate was calculated, and these values were used to train a generative adversarial model. To accomplish this, they trained an encoder network to accept a watermark and a 128-bit string to produce a visually imperceptible watermark. They also trained a decoder that was encouraged to minimize distortion when an input image was labeled as benign and maximize distortion when the image was labeled as malicious. This resulted in what they labeled as a semi-fragile watermark that would be sensitive to benign transformations while robust to malicious ones (Neekhara, et al., FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes, 2022).

2.2.2 Disruptive Watermarks

Ruiz et al., were the first to propose an adversarial watermark to attack generative adversarial networks (Ruiz, Bargal, & Sclaroff, 2020). Previous work had been done on adversarial attacks on deep neural networks, but this work didn't focus on disrupting conditional image-to-image translation, which Ruiz et al., specifically study. Image-to-image translation happens when the same networks are trained for the same task, but on different datasets (Isola, Zhu, Zhou, & Efros, 2018). The result is that the network learns to map one image to another given different input (Isola, Zhu, Zhou, & Efros, 2018),

essentially learning how to create a specific type of image given something similar. Ruiz et al., proposed a watermark that was trained to specifically disrupt a GAN model to produce a different output from the one it intends, i.e., a blurred photo instead of a deepfake. In their study, they measured the distortion of the output using L2, also known as the mean squared error. Figure 1 shows the different L2 values they obtained and the resulting distortion. They concluded that images with L2 values greater than or equal to 0.05 produced visible distortions (Ruiz, Bargal, & Sclaroff, 2020). In the CMUA-Watermark study, which will be compared in this paper, the authors, Huang et al. propose the same mechanism but propose that their watermark methodology can be applied to any image, and their strategy of automatic step size tuning can train a model to generate a specific watermark that will protect all images from that deepfake engine.

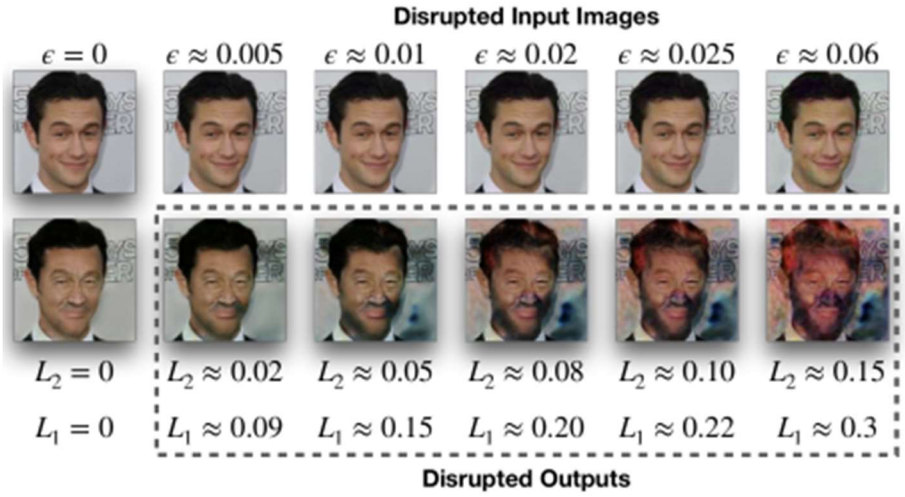

Figure 1: Different L2 values and their corresponding distortion between original images and disruptive watermarks (Ruiz, Bargal, & Sclaroff, 2020)

3. APPROACH

3.1 High Level Design

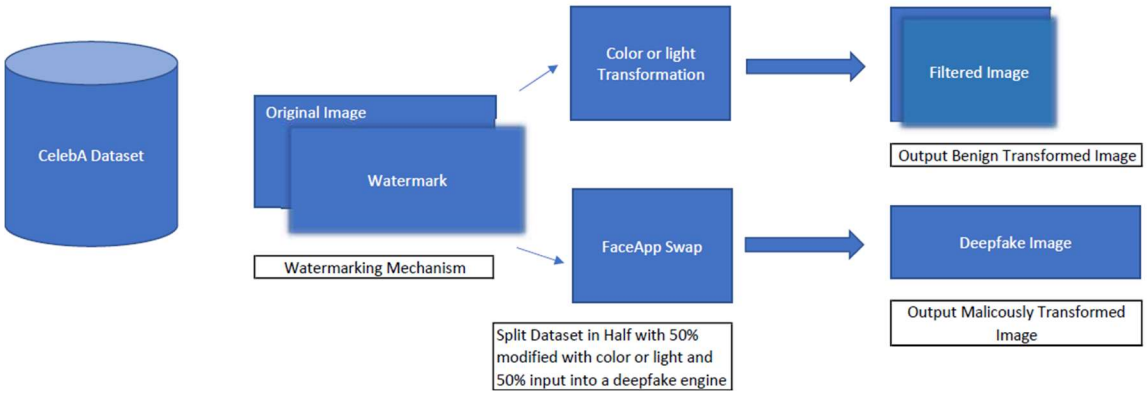

Figure 2: The testing pipeline used for each watermarking method tested in this paper

Two papers were selected to analyze and compare. These studies are "FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication" and "Countering Deepfakes, and CMUA-Watermark: A Cross-Model Universal Adversarial Watermark for Combating Deepfakes". Both studies developed a method to embed a watermark in an image. The watermarks created in these studies will be tested and analyzed. These studies were selected due to their similarity in using deep learning methods to train a model to add a watermark to an image, as well as the availability of their code repository and pre-trained model. The goal of both studies is to prevent a deepfake engine from using the original images to create fake images. In the first study, the use of the image to create a deepfake results in a watermark that is unrecoverable. In the second study, the use of the image in a deepfake engine results in image distortion, or blurring. Figures 3 and 4 show the respective models for each watermarking method.

To compare the watermarks, an AUC metric is calculated. AUC measures how the watermark classification perform across all classification thresholds. An image is determined to be fake if the watermark from the deepfake engine is unrecoverable or, in the case of the second study, if the resulting deepfake image is blurred. Recoverable watermarks, and unblurred images are considered benign and labeled as authentic images. Unrecoverable watermarks and blurred images are considered malicious and labeled as fake images. Additional metrics used for each study individually are discussed in the next sections.

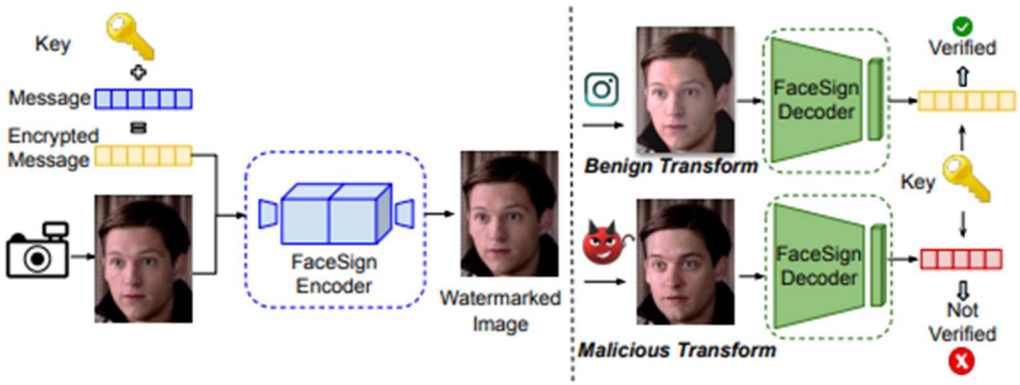

Figure 3: The embedding, transformation, and recovery of a watermark in the first study. The watermarked image is then transformed and the decoder classifies the image as real or fake.

(Neekhara, et al., FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes, 2022)

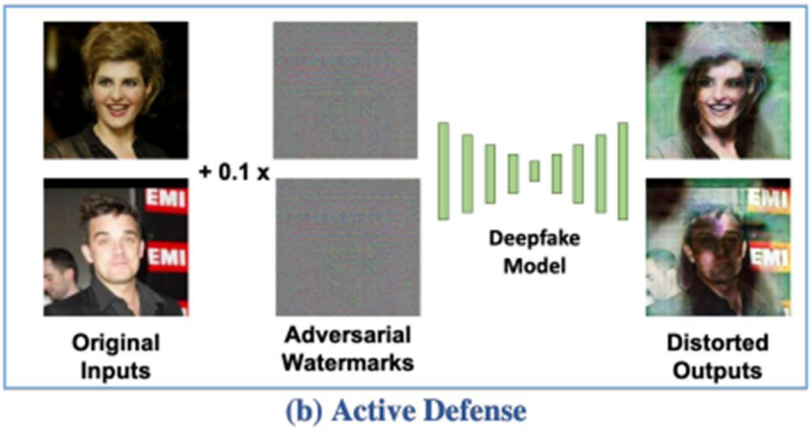

Figure 4: The embedding, transformation, and distortion of deepfake watermarked images . The original images are embedded with the same watermark and when fed into a deepfake engine, produce a distorted output.

(Huang, et al., 2021)

3.2 Study Specific Metrics

In the first study, "FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication", Neekhara et al., use Bit Recovery Rate (BRA) and AUC to measure how well their watermark performed. For this study, the bit recovery rates are calculated and are used as a boundary to classify the image as real or fake. A BRA greater than or equal to 0.95 is classified as real and a lower value is classified as fake. An AUC metric is also calculated, comparing the ground truth of real or fake. Finally, a confusion matrix will be generated to analyze the end classification of the watermark recovery model, also known as the decoder. Because this is a classification problem, the confusion matrix will have values for true positives, true negatives, false positives, and false negatives. Confusion matrices also allow us to see if there is a subset of images that the model struggles to classify

For the second study, “Countering Deepfakes, and CMUA-Watermark: A Cross-Model Universal Adversarial Watermark for Combating Deepfakes” Huang et al., calculate an L2 score to compare the original deepfake image without a watermark to the watermarked deepfake image, to see if there is more distortion in the watermarked deepfake image. They use the boundary of L2 greater than 0.05 to signify visual distortion, which is the same boundary used by Ruiz et al., in the previously mentioned Related Works Section 2.2.2.

To evaluate the second study, L2 scores were calculated between the original image and deepfake and watermarked image and deepfake. This is so that we can see if the addition of the watermark caused enough increased distortion in the deepfake. Figure 5 shows the model for this calculation. To calculate an AUC score, the images were predicted as benign or malicious. If enough distortion was caused, the image was classified as malicious, if there wasn’t any significant distortion, the image was classified as benign.

L2 values that were greater than 0.05 for the watermarked images, were predicted as malicious, while values lower were predicted as benign. If the watermark fails to cause enough disruption to the image to be noticeable, the watermark will not have a high L2 value. It’s important to note that simply comparing the distortion between the watermarked image and its resultant deepfake would tell us how much the watermarked image was changed by the deepfake, which would be inflated because the faces were changed. This wouldn’t tell us how much more protection the watermark adds compared to the original image.

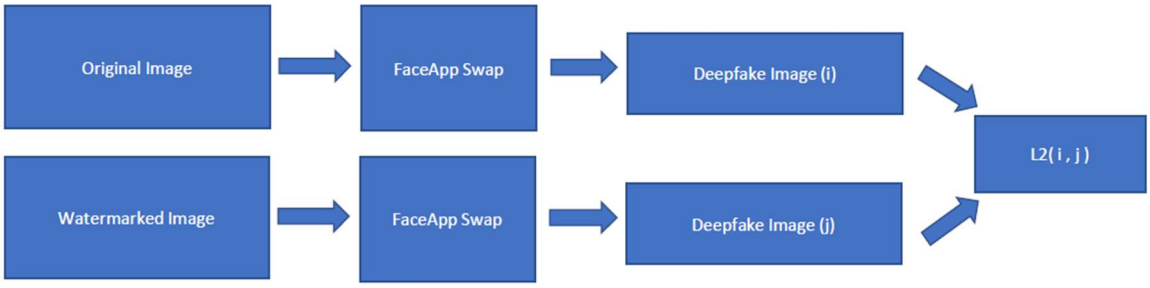

Figure 5: Calculation of L2 score

3.3 Implementation

The code for both studies came from their respective authors GitHub pages (Neekhara, et al., github.com, 2022), (Huang, et al., 2022), and was supplemented with additional code to load the new dataset, calculate metrics, and analyze the AUC score. The CelebFaces (CelebA) Dataset of images used was sourced from Kaggle (Liu, Luo, Wang, & Tang, 2015). The dataset consists of 202,603 images of celebrities.

The overall design for this study was implemented in Jupyter Notebook, where the dataset was loaded, standardized, watermarked, transformed, and analyzed.

For each study individually, the entire dataset of images was embedded with their respective watermarks. The dataset was then split in half, with each set of images modified in either a benign or malicious way. The benign modifications were common light and color transformations, which were implemented by applying popular image filters found on social media, like aden, or toaster as well as image compression to two different sizes. The filters were applied using the Image Module Python library (Lundh, Clark, Sphix, & Furo, 2010 - 2022). The malicious transformations were implemented by sending the watermarked images through the FaceSwap deepfake engine (Ghosn, Leite Guilherme, & Rosenzvaig, 2020) . Both transformations were previously implemented by

Neekhara et al., in their GitHub repository. After the images were benignly and maliciously transformed, the resulting output image was then analyzed for its authenticity.

4. METHODOLOGY, RESULTS AND ANALYSIS

To directly compare the robustness and fragility of each pre-trained watermark, an AUC score was calculated based on the correct classification of a benign image as benign, and a maliciously transformed image as fake. The FaceSigns watermark had an AUC of 0.9992 while the CMUA watermark had an AUC of 0.5.

4.1 Methodology

The CelebA dataset was used to test the effectiveness of each watermark. This dataset consisted of 202,603 celebrity images. Running all images through the testing framework, shown in Figure 2, proved to be changeling, due to limitations in time and memory. To combat this problem, images were selected in random batches of 500 and the metrics were averaged across each batch. For each batch, the dataset of images was split in half, with 50% being benignly transformed and 50% maliciously transformed. Three batches of images were analyzed. See Appendix A for the result of each batch.

For the first study, the images were encoded using a pre-trained pytorch encoder model that accepts an image and a bit string, and outputs watermarked images. The model was downloaded from the paper's GitHub, which provided the code to load, watermark, apply benign and malicious transformations, and decode the watermark (Neekhara, et al., github.com, 2022). See Appendix A for an example of the original image, its watermarked version, and the watermark itself.

For the second study, the images were watermarked using a pre-trained model that accepts an image and outputs a watermarked image. This model was provided by the original authors, on their GitHub page (Huang, et al., 2022). The same image

transformations used on the first study were applied, and the output images were classified as either benign or malicious.

For this second study, the FaceSwap engine should produce a blurred image when input a watermarked image. To measure how well the deepfake image was distorted, a Mean Squared error L2 was calculated using pytorch's built in `mse_loss` function . The original images were fed into the deepfake engine without a watermark, and the L2 values were calculated. Then the watermarked images were fed into the deepfake engine and the L2 values were calculated. This was done to compare the distortion in the deepfake engine between the non-watermarked and watermarked images. Interestingly, there were no significant differences in the L2 values between the watermarked and non-watermarked images. This means that the watermarked images didn't produce any extra distortion when run through the deepfake engine, compared with the original images. The L2 between the deepfake original images and deepfake watermark images was 0.00013, which is not enough to produce a visible distortion. See Figure 2 for L2 distortions. Qualitatively, we can also see that the watermarked images that were deepfakes didn't appear visually distorted. See Appendix A for additional images. As a result of none of the images were classified as malicious because the watermark distortion didn't reach the threshold required to classify it as malicious.

4.2 Results

Metric (average)	Semi-fragile watermark	CMUA watermark
AUC score	0.9992	0.5
L2	--	0.00013

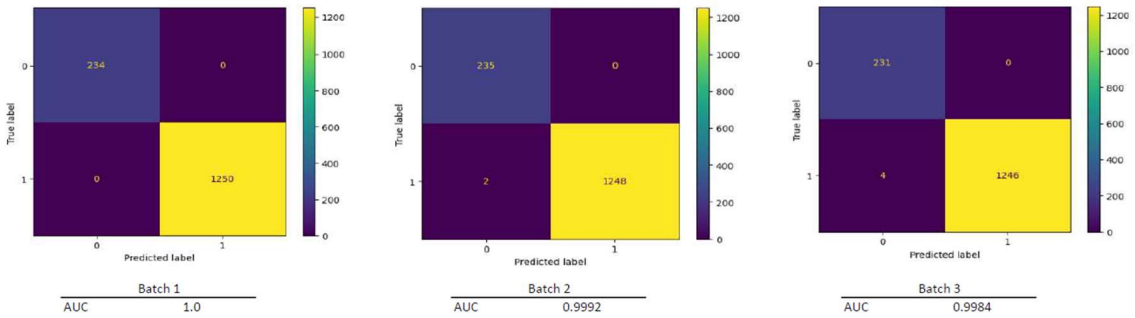

Figure 6: Confusion matrices for FaceSigns watermark classification

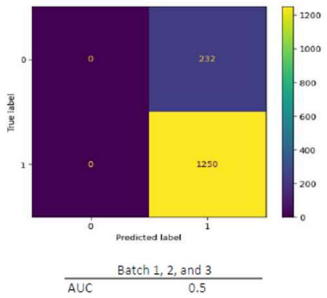

Figure 7: Confusion matrices for CMUA-watermark classification

4.3 Analysis

While both studies pre-trained a watermark, the first model designed a universal watermark that doesn't need any additional training to transfer as a defense to any deepfake technique. In comparison, the second study trained a universal watermark, but they also they added additional training specific to the deepfake engine. However, in the sense of comparing apples to apples, this paper compared only the individual watermarks, without any additional training specific to the deepfake engine. The watermarks were not additionally trained on the specific deepfake engine prior to being tested on it. Notably, Neekhara et al., mention that they specifically didn't train their watermark on a deepfake

engine to avoid the lack of generalization from the watermark on foreign deep fake techniques. This reasoning was further justified by the results of this study, which show that watermarks trained for a specific model did not generalize well against an unseen deepfake model, FaceSwap.

The CMUA watermark, without training on the specific deepfake engine, was unable to disrupt any of the deepfake engine images, and likely requires additional step size training on the FaceSwap model to be effective. The AUC from this watermark was 0.5, with 50% true positives, and 50% false negatives, seen in Figure 6 above. The watermark was unable to distort malicious images more than benign images, meaning that it couldn't differentiate between malicious or benign changes. As a result, all images were classified as benign. The watermark didn't have any fragility. In contrast, the FaceSigns watermark was unable to be recovered for deepfake photos, protecting the image. By analyzing the BRA accuracy for deepfake images, the watermark classified images above 0.95 BRA as benign and those that fell under this threshold were correctly identify as malicious, and even further, the secret embedded string was unable to be recovered from the image. Appendix A includes images and their corresponding BRA score. In one of the batches, the FaceSigns watermark had 2 false negatives, where it predicted an image as malicious when it was benign. False negatives are ok with image classification because the original owner of the image can themselves confirm if the image is authentic or not. Overall, the FaceSigns watermark had an AUC of 0.9992, which is the same as what the authors reported.

By studying the techniques used in each different watermarking method, we can see how the studies differed, and what that means for defensive strategies against

deepfake technology. Both studies watermarked images, but the way in which the watermark was created was different. In the first study by Neekhara et al, the watermark was embedded with a secret 128-bit string upon when accurately recovered, can be checked by the owner of the image and key for authenticity, proving images are real and detecting when they are fake. This study is more secure against adversarial attacks where the attacker attempts to train a model to recover the watermark (Neekhara, et al., FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes, 2022). Even if the watermark is recovered, because it is embedded with a 128-bit encrypted key, the adversary still wouldn't be able to duplicate the watermark correctly. This provides extra security against attacks on the watermarked image. In contrast, Huang et al, in the second study, create a general watermark that is the same for every image, and is specialized to one deepfake model. This training on one deepfake model potentially caused overfitting, evidenced by the inability of the general watermark to respond to new deepfake changes. Additionally, their watermark was a general watermark that could be more vulnerable to copying attacks, where adversaries attempt to copy the watermark.

It's important to note that the authors of the second study develop an additional method to hyper-tune the watermark by training it on a specific deepfake model. This step was not replicated in this study, which only compares the two watermarks. In the second study, the authors were able to achieve high success rate protecting facial images by hyper-tuning the watermark to four different deepfake engines (Huang, et al., 2021). The authors demonstrated that training a watermark to distort a specific deepfake engine

is possible, however this also leads to overfitting the watermark, and needing to train a watermark for every deepfake engine.

5. CONCLUSIONS

5.1 Summary

By comparing both watermarking techniques we were able to test how well each watermarking image was able to detect deepfake images by analyzing the watermark before and after changes. In the first study, “FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering”, the watermark was able to hide the secret string when input in a deepfake enigne, while recovering the string when filters or image size changes were applied. This watermark had good results, with an AUC of 0.9992. The second study, “CMUA-Watermark: A Cross-Model Universal Adversarial Watermark for Combating Deepfakes”, applied an imperceptible watermark to the images, but was unable to disrupt the FaceSwap engine by blurring the photos. This study had an AUC of 0.5 and the distortion for the photos was unnoticeable.

This project demonstrates that it is possible to protect any image from multiple deepfake engines using the embedded pre-trained watermark, in combination with a 128-bit secret, developed by Neekhara et al.,. This project also demonstrates how single adversarial watermarks, trained on one deepfake technique, do not generalize well to new deepfake techniques.

5.2 Contributions

This Project Report contributes to the deepfake field in a few ways:

- It compares two different watermarking methods, in which the first method trained without a deepfake engine and the second method was trained with 4 of them and evaluates their performance against an unseen deepfake engine.

- Demonstrates that some watermarks trained without access to a specific deepfake engine can generalize well to new deepfake engines.
- Demonstrates poor performance of a single adversarial watermark against a new deepfake engine.

5.3 Future Work

This project provides preliminary evidence that deepfake prevention mechanisms can be created so that they generalize well to unseen deepfake technology. More work is needed to

- Analyze the watermark for biases and see if it performs more accurately on different subsets of data.
- Test the watermarking methods to see how much computer resources they consume.

REFERENCES

Carlini, N., & Farid, H. (2020). Evading Deepfake-Image Detectors with White- and Black-Box Attacks. *arXiv.or Cornell University*.

Dang, H., Liu, F., Stehouwer, J., Liu, X., & Jain, A. (2020). *On the Detection of Digital Face Manipulation.*

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., & Ferrer, C. C. (2020). The DeepFake Detection Challenge (DFDC) Dataset. *arXiv:2006.07397*.

Ghosn, E., Leite Guilherme, L., & Rosenzvaig, R. (2020, December). *Single image face swap using dlib and OpenCV*. Retrieved from GitHub: https://github.com/guipleite/CV2-Face-Swap

Huang, H. W., Chen, Z., Zhang, Y., Li, Y., Tang, Z., Chu, W., . . . Ma, K.-K. (2021). CMUA-Watermark: A Cross-Model Universal Adversarial Watermark for. *arXiv:2105.10872v2*.

Huang, H., Wang, Y., Chen, Z., Zhang, Y., Li, Y., Tang, Z., . . . Ma, K.-K. (2022). *github.com*. Retrieved from CMUA-Watermark: https://github.com/VDIGPKU/CMUA-Watermark

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2018). Image-to-Image Translation with Conditional Adversarial Networks. *arXiv:1611.07004v3*.

Lalonde, D. (2021). *Policy Options on Non-Consensual Deepnudes an Secual Deepfakes.* London, Ontario: Learning Network Brief 39 ISBN: 978-1-988412-49-8.

Lian, S., & Zhang, Y. (2009). Handbook of Research on Secure Multimedia Distribution. In *Handbook of Research on Secure Multimedia Distribution* (pp. 257 - 314).

Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). *Kaggle.com*. Retrieved from CelebFaces Attributes CelebA Dataset: https://www.kaggle.com/datasets/jessicali9530/celeba-dataset

Lundh, F., Clark, A., Sphix, & Furo, p. (2010 - 2022). *Image Module*. Retrieved from pillow: https://pillow.readthedocs.io/en/stable/reference/Image.html

Lv, L. (2021). Smart Watermark to Defend against Deepfake Image Manipulation. *IEEE*.

Neekhara, P., Hussain, S., Zhang, X., Huang, K., McAuley, J., & Koushanfar, F. (2022). FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes. *arXiv:2204.01960*.

Neekhara, P., Hussain, S., Zhang, X., Huang, K., McAuley, J., & Koushanfar, F. (2022). *github.com*. Retrieved from FaceSignsDemo: https://github.com/paarthneekhara/FaceSignsDemo

Ruiz, N., Bargal, S. A., & Sclaroff, S. (2020). Disrupting Deepfakes: Adversarial Attacks Against Conditional Image Translation Networks and Facial Manipulation Systems. *Bartoli, A., Fusiello, A. (eds) Computer Vision – ECCV 2020 Workshops. ECCV 2020. Lecture Notes in Computer Science(), vol 12538* (pp. 236 - 251). Springer, Cham.

Tancik, M., Mildenhall, B., & Ng, R. (2019). StegaStamp: Invisible Hyperlinks in Physical Photographs. *arXiv:1904.05343*.

Yang, Y., Liang, C., He, H., Cao, X., & Zhenqiang Gong, N. (2021). FaceGuard: Proactive Deepfake Detection. *arXiv:2109.056773v1*.

Yu, P., Xia, Z., Fei, J., & Lu, Y. (2020). A Survey on Deepfake Video Detection. *The Institution of Engineering and Technology*.

Zhang, T. (2022). Deepfake generation and detection, a survey. *Multimed Tools Appl 81*, 6259–6276.

Zhu, J., Kaplan, R., Johnson, J., & Fei-Fei, L. (2018). HiDDeN: Hiding Data With Deep Networks. *European Conference on Computer Vision*, 682-697.

APPENDIX A. ADDITIONAL FIGURES

Figure 8: The original, watermarked, and watermark for an image using FaceSigns

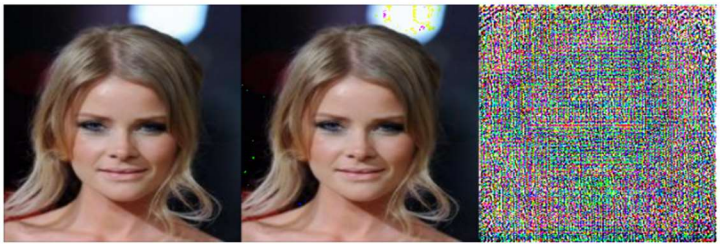

Figure 9: The original, watermarked, and watermark for an image using CMUA

Figure 10: Benign vs. malicious watermark decoding using FaceSigns

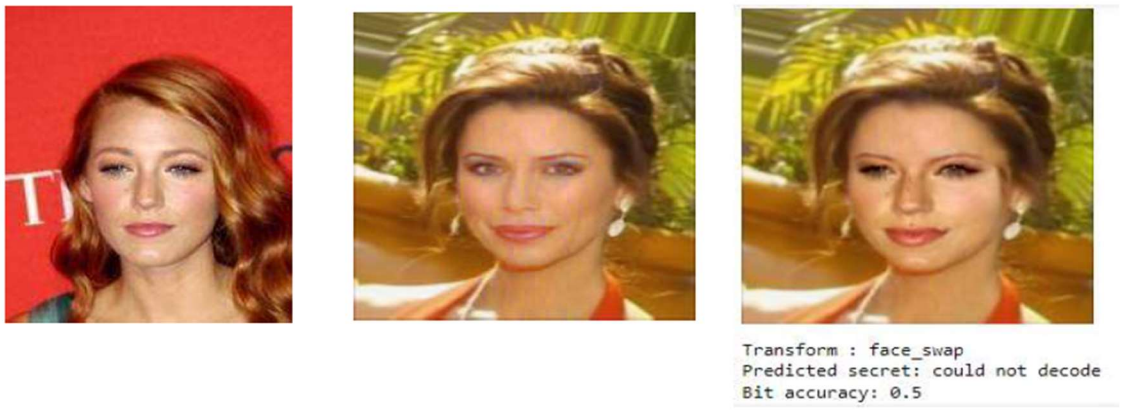

Figure 11: Target image, original image with watermark and deepfake using FaceSigns

Figure 12: Benign image transformations on a CMUA watermarked image

Figure 13: Target image, original image with watermark and deepfake using CMUA