

FaceGuard: Proactive Deepfake Detection

Yuankun Yang^{1*}, Chenyue Liang^{2*}, Hongyu He³, Xiaoyu Cao³, Neil Zhenqiang Gong³

¹Fudan University, 17307110068@fudan.edu.cn

²Chinese Academy of Sciences, llcy_cheryl@outlook.com

³Duke University, {hongyu.he, xiaoyu.cao, neil.gong}@duke.edu

Abstract

Existing deepfake-detection methods focus on *passive* detection, i.e., they detect fake face images via exploiting the artifacts produced during deepfake manipulation. A key limitation of passive detection is that it cannot detect fake faces that are generated by new deepfake generation methods. In this work, we propose *FaceGuard*, a *proactive* deepfake-detection framework. FaceGuard embeds a watermark into a real face image before it is published on social media. Given a face image that claims to be an individual (e.g., Nicolas Cage), FaceGuard extracts a watermark from it and predicts the face image to be fake if the extracted watermark does not match well with the individual’s ground truth one. A key component of FaceGuard is a new deep-learning-based watermarking method, which is 1) robust to normal image post-processing such as JPEG compression, Gaussian blurring, cropping, and resizing, but 2) fragile to deepfake manipulation. Our evaluation on multiple datasets shows that FaceGuard can detect deepfakes accurately and outperforms existing methods.

1 Introduction

As deep learning becomes more and more powerful, deep learning based *deepfake generation methods* can produce more and more realistic-looking deepfakes [8, 18, 19, 20, 30, 35, 41, 42, 51, 56]. In this work, we focus on fake faces because faces are key ingredients in human communications. Moreover, we focus on *manipulated* fake faces, in which a deepfake generation method replaces a target face as a source face (known as *face replacement*) or changes the facial expressions of a target face as those of a source face (known as *face reenactment*). For instance, in the well-known Trump-Cage deepfakes example [34], Trump’s face (target face) is replaced as Cage’s face (source face). Fake faces can be used to assist the propagation of fake news, rumors, and disinformation on social media (e.g., Facebook, Twitter, and Instagram). Therefore, fake faces pose growing concerns to the integrity of online information, highlighting the urgent needs for deepfake detection.

Existing deepfake detection mainly focuses on *passive* detection, which exploits the artifacts in fake faces to detect them after they have been generated. Specifically, given a face image, a passive detector extracts various features from it and classifies it to be real or fake based on the features. The features can be manually designed based on some heuristics [2, 14, 22, 23, 27, 50] or automatically extracted by a deep neural network based feature extractor [1, 6, 11, 14, 28, 29, 37, 38, 48, 54]. Passive detection faces a key limitation [7], i.e., it cannot detect fake faces that are generated by new deepfake generation methods that were not considered when training the passive detector. As new deepfake generation methods are continuously developed, this limitation poses significant challenges to passive deepfake detection.

Our work: In this work, we propose *FaceGuard*, a *proactive* deepfake-detection framework. FaceGuard addresses the limitation of passive detection via proactively embedding watermarks into real face images before they are manipulated by deepfake generation methods. Figure 1 illustrates the difference between passive detection and FaceGuard. Specifically, before posting an individual’s real face image on social media, **FaceGuard embeds a watermark (i.e., a binary vector in our work) into it.** The watermark is human imperceptible, i.e., a face image and its watermarked version look visually the same to human eyes. For instance, the watermark can be embedded into an individual’s face image using the individual’s smartphone. Suppose a face image is claimed to be an individual, e.g., the manipulated

*The first two authors made equal contributions. They performed this research when they were remote interns in Gong’s group.