Ben-Gurion University of the Negev

Faculty of Engineering Science

Department of Software and Information Systems Engineering

Project in Offensive Artificial Intelligence Course

OAI Final Project - Robustness of Real Time Deepfakes

Author:
Amit Kama

Author:
Oren Shvartzman

June 20, 2022

Contents

1 Introduction

Deepfake is a general term that encompasses the use of deep learning algorithms in order to create synthetic media, in which one subject in an existing visual and/or audio content is usually replaced with another's likeness. While fraudulent content has been around for some time, recent advances in machine vision have posed a major threat to the trust and transparency of the media. Using powerful machine-learning and deep-learning techniques, deepfakes can now manipulate or generate visual and audio content that can be more easily misleading.

In recent years, deepfakes have garnered widespread attention for their uses in spreading fake news, committing financial fraud, creating pornographic materials, and many other disturbing uses. This has led to a significant need to identify and restrict their use.

Ever since the introduction of deepfakes, researchers in deep learning have increasingly focused on this area of research. In particular, they propose methods, as well as practical implementations of deepfakes is in various fields. Among other methods, in [1], Siarohin et al. propose the first order motion model for image animation. Their framework enables generating a video sequence, in which an object in a source image is animated according to the motion of a driving video, without using any annotation or prior information about the specific object to animate. According to them, once trained on a set of videos depicting objects of the same category, the method can be applied to any object of it. Based on this method, a number of real time photorealistic avatars have recently been developed, one of which we will explore in this work.

However, real time avatars are far from perfect, as they are not robust when it comes to edge cases. This includes facial gestures in the driving video, objects in the source or target media that make it difficult to identify facial boundaries, and many more.

Note that these limitations can be create visual glitches and distortions that can be detected with the naked eye as well as by software, and therefore can be utilized for deepfakes detection. This observation drives a growing amount of research dealing with those inaccuracies for dual use – correcting them in order to improve the deepfakes' credibility, or exploiting them to distinguish deepfakes from real content.

In this work, we evaluate the robustness of real time deepfakes by im-

plementing first order motion model-based avatarify and examining various edge cases on it. After demonstrating failures in the implementation, we also provide a method to utilize them for deepfakes detection.

2 Methods

In this section we will briefly introduce the First Order Motion Model, on which the avatarify implementation whose robustness we evaluated, consists of. In addition, we will describe our proposed method for failures detection.

2.1 First Order Motion Model

In [1], Siarohin et al. propose the First Order Motion Model for Image Animation, which addresses the task of generating a video sequence so that an object in a source image is animated according to the motion of a driving video.

Once their proposed method is trained on a set of videos depicting objects in the same category, it can be applied to any object in this category, without using annotations or prior knowledge about the object to be animate. This is done using a self-supervised formulation, for detaching appearance information and motion information from each other. The method also support complex motions, using representation consisting of a set of learned keypoints along with their local affine transformations.

The model's mathematical formulation describes motion between two frames and is efficiently computed by deriving a first order Taylor expansion approximation. Thus, motion is represented as a set of keypoints displacements and local affine transformations. Finally, a generator network combines the appearance extracted from the source image with the motion representation of the driving video. First Order Motion Model outperforms state of the art on all the benchmarks on a variety of object categories.

In the literature, most methods tackle image animation task by assuming strong priors on the object representation and resorting to computer graphics techniques. These approaches can be referred to as object-specific methods, as they assume knowledge about the model of the specific object to animate.

Recently, deep generative models have emerged as effective techniques for image animation and video retargeting [2, 41, 3, 42, 27, 28, 37, 40, 31, 21]. In particular, Generative Adversarial Networks (GANs) [14] and Variational

Auto-Encoders (VAEs) [20] have been used to transfer facial expres- sions [37] or motion patterns [3] between human subjects in videos. Nevertheless, these approaches usually rely on pre-trained models in order to extract object-specific representations such as keypoint locations. Unfortunately, these pre-trained models are built using costly ground-truth data annotations [2, 27, 31] and are not available in general for an arbitrary object category. To address this issues, recently Siarohin et al. [28] introduced Monkey-Net, the first object-agnostic deep model for image animation. Monkey-Net encodes motion information via keypoints learned in a self-supervised fash- ion. At test time, the source image is animated according to the corresponding keypoint trajectories estimated in the driving video. The major weakness of Monkey-Net is that it poorly models object appearance transformations in the keypoint neighborhoods assuming a zeroth order model (as we show in Sec. 3.1). This leads to poor generation quality in the case of large object pose changes (see Fig. 4).

To tackle this issue, we propose to use a set of self-learned keypoints together with local affine transformations to model complex motions. We therefore call our method a first-order motion model. Second, we introduce an occlusion-aware generator, which adopts an occlusion mask automatically estimated to indicate object parts that are not visible in the source image and that should be inferred from the context. This is especially needed when the driving video contains large motion patterns and occlusions are typical. Third, we extend the equivariance loss commonly used for keypoints detector training [18, 44], to improve the estimation of local affine transformations. Fourth, we experimentally show that our method significantly outperforms state-of-the-art image animation methods and can handle high-resolution datasets where other approaches generally fail. Finally, we release a new high resolution dataset, Thai-Chi-HD, which we believe could become a reference benchmark for evaluating frameworks for image animation and video generation.

2.2 Failures Detection Method

2.2.1 Facial Detection with Dlib

3 Experiments and Results

In this section we introduce the experiments and results. A detailed explanation of the visual glitches and distortions that we managed to create and detect is provided, as well as the results of our proposed method to utilize them for deepfakes detection.

3.1 First Order Model Robustness

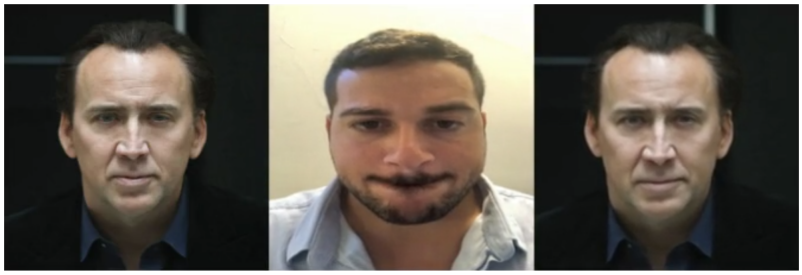

Figure 1: The effect of...

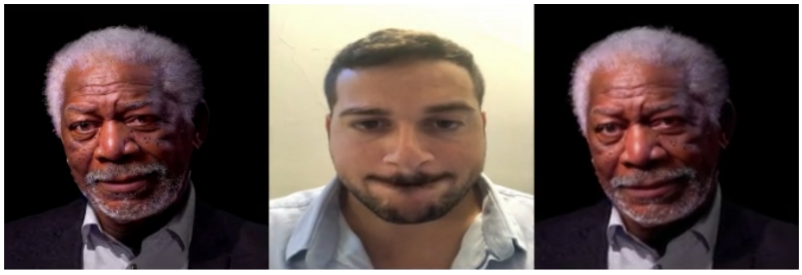

Figure 2: The effect of...

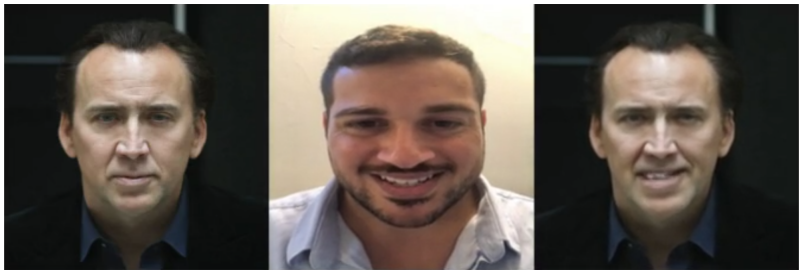

Figure 3: The effect of...

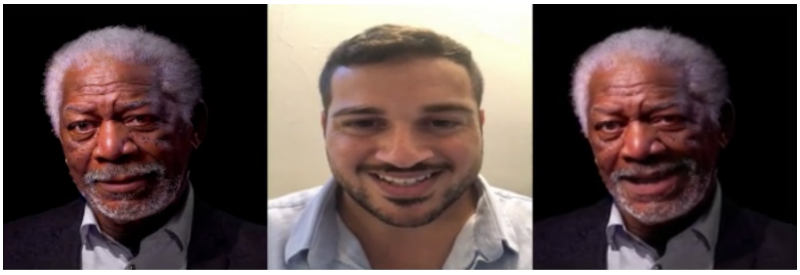

Figure 4: The effect of...

3.2 Failures Detection

4 Discussion

Today, deepfakes are bla bla bla..

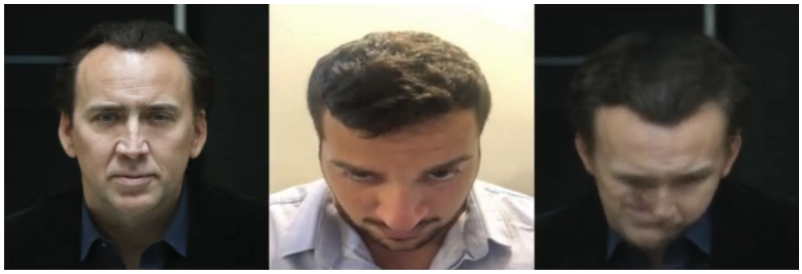

Figure 5: The effect of...

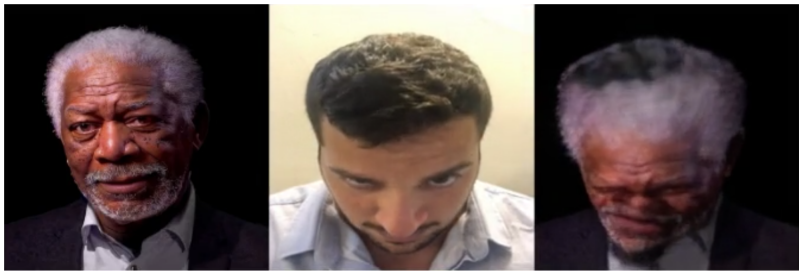

Figure 6: The effect of...

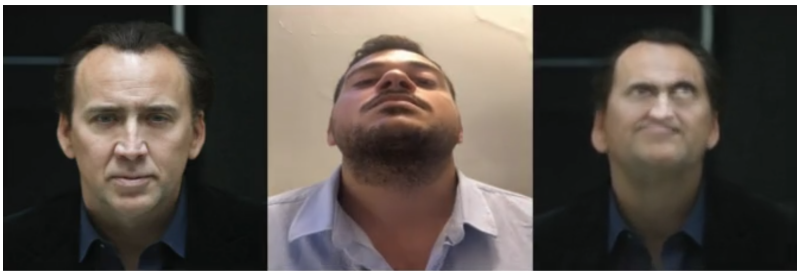

Figure 7: The effect of...

References

[1] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *CoRR*, abs/2003.00196, 2020.

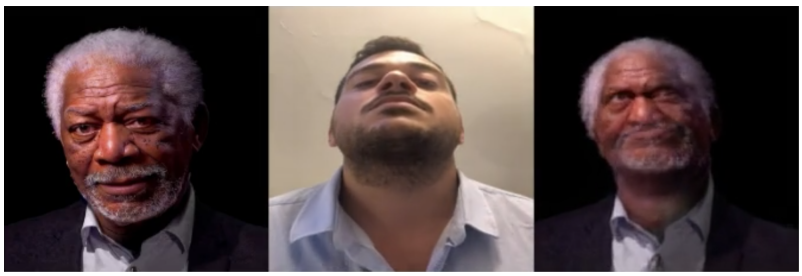

Figure 8: The effect of...

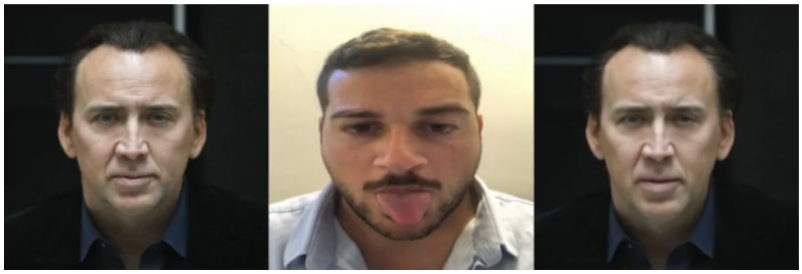

Figure 9: The effect of...

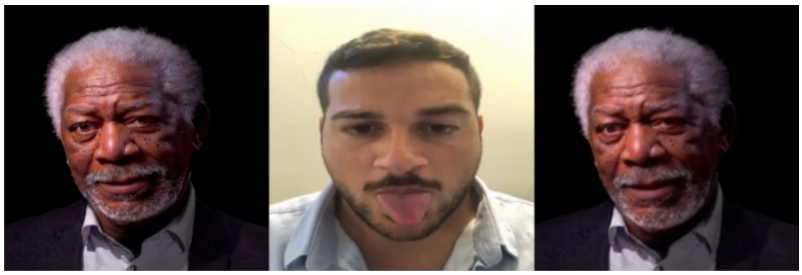

Figure 10: The effect of...

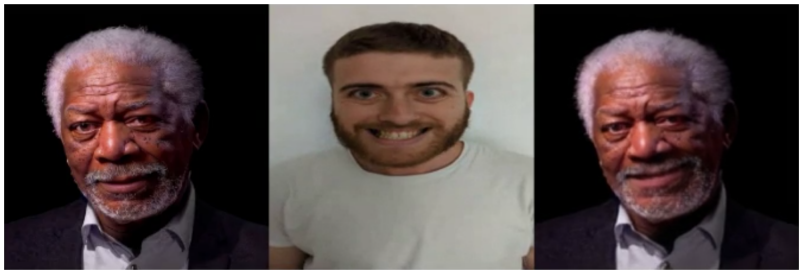

Figure 11: The effect of...

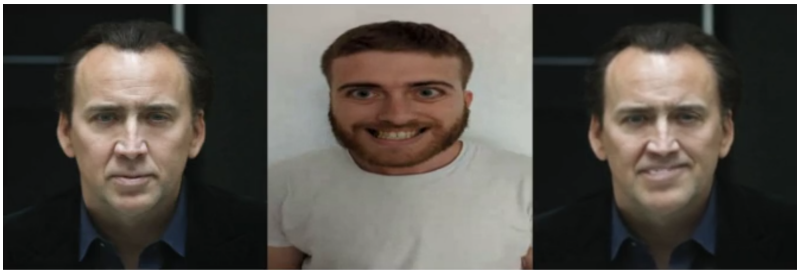

Figure 12: The effect of...

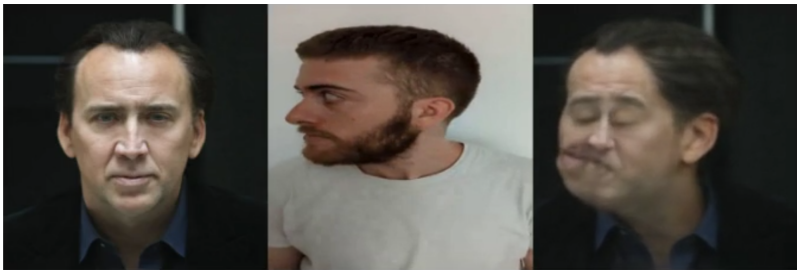

Figure 13: The effect of...

Figure 14: The effect of...

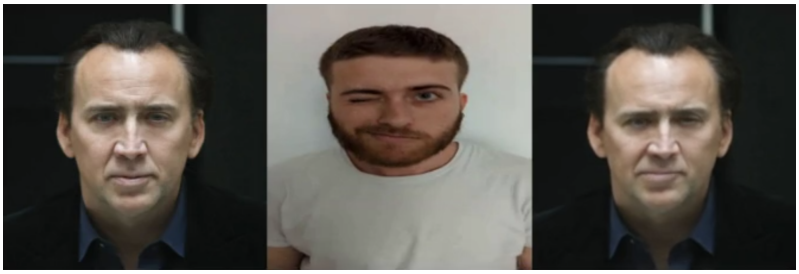

Figure 15: The effect of...

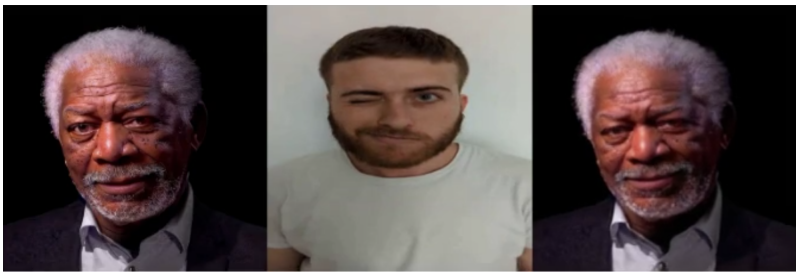

Figure 16: The effect of...