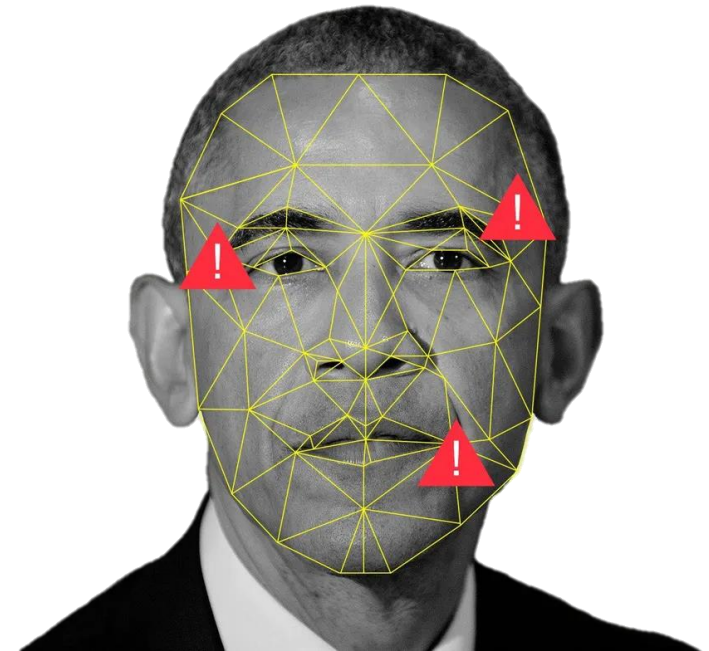


Robustness of Real Time deepfakes

Evaluating the robustness of real time deepfakes by implementing avatarify and examining edge cases on it

Deepfakes

- ✓ 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.

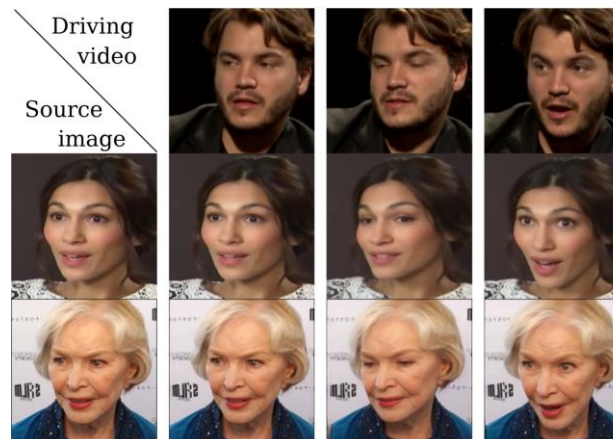


Deepfakes

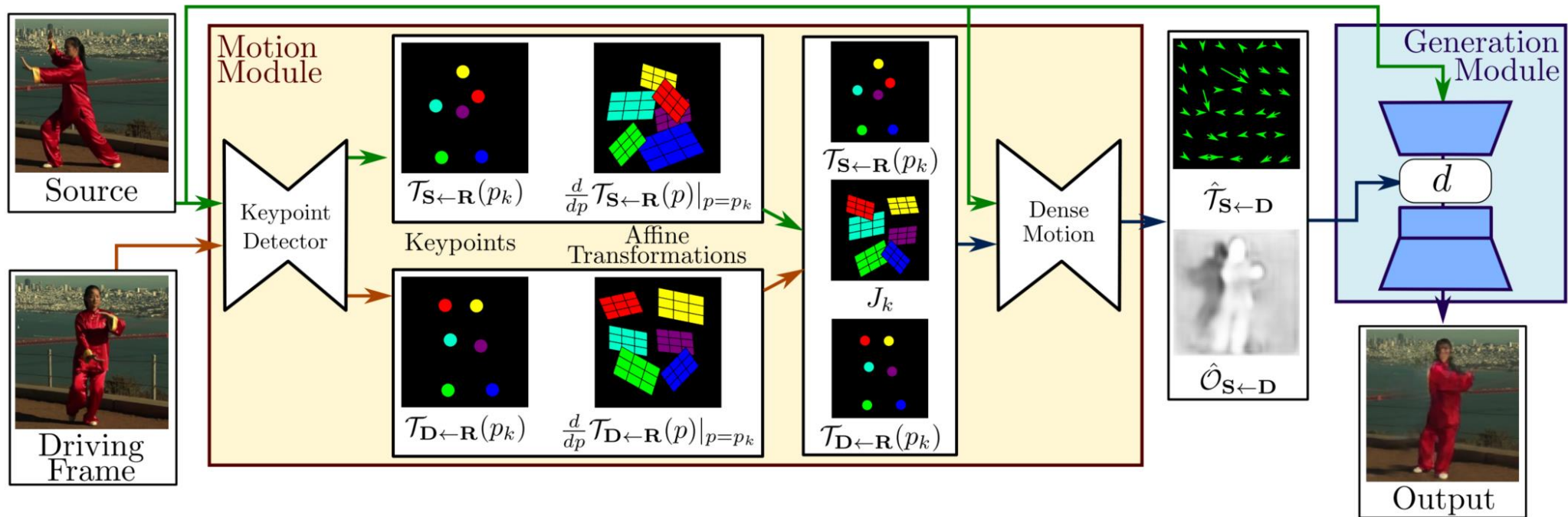
- ✓ Ever since the introduction of deepfakes, researchers in deep learning have increasingly focused on this area of research. They propose methods, as well as practical implementations of deepfakes in various fields. Among other methods, Siarohin et al. propose the first order motion model for image 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.
- ✓ 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, etc.
- ✓ These limitations can create visual glitches and distortions that can be detected with the naked eye as well as by software, and therefore can be utilized for failures 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 implementing first order motion model-based avatarification and examining various edge cases on it. After demonstrating failures in the implementation, we also provide a method to utilize them for failures detection.

First Order Motion Model

- ✓ 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 supports 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.



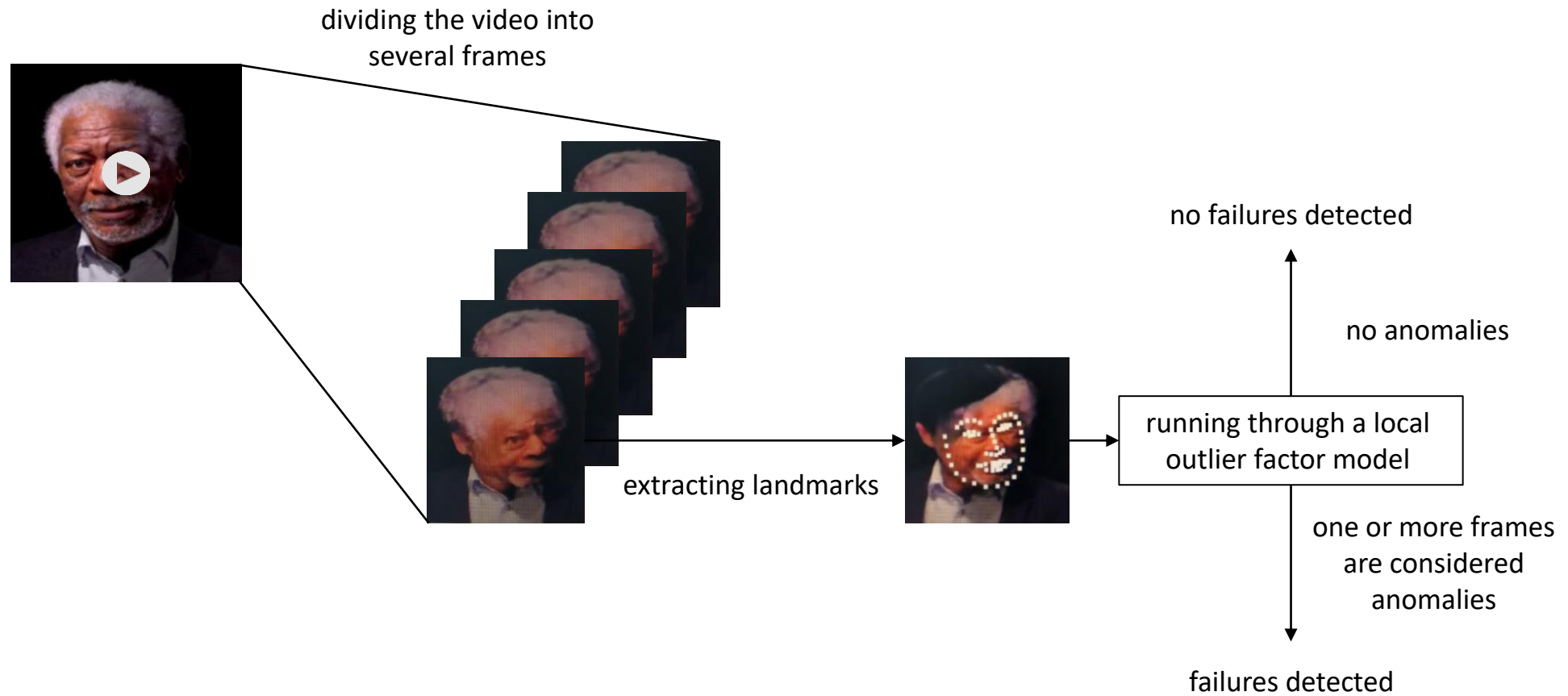
First Order Motion Model



Failures Detection Method

- ✓ At the basis of our method for failures detection is the idea according to which we can detect the avatarify failures using anomaly detection methods. To address the task of detecting avatarify failures in an output video, we simplified it to the problem of detecting avatarify failures in a target image. To do so, the method starts with dividing the video into several frames. For each frame we extract the facial landmarks, calculate Euclidean distance between all the points.
- ✓ We then normalize and run the results through the local outlier factor model with 20 neighbors. The test was done using the Yale Face Database, which contains 165 grayscale images in GIF format of 15 individuals, when for each subject, there are 11 images, one for each different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. This is in practice using novelty local outlier factor, since the dataset contains only ordinary faces.
- ✓ The test includes making prediction on each frame of the video. If one or more frames are considered anomalies, we declare that the video contains failures.

Failures Detection Method



First Order Model Robustness

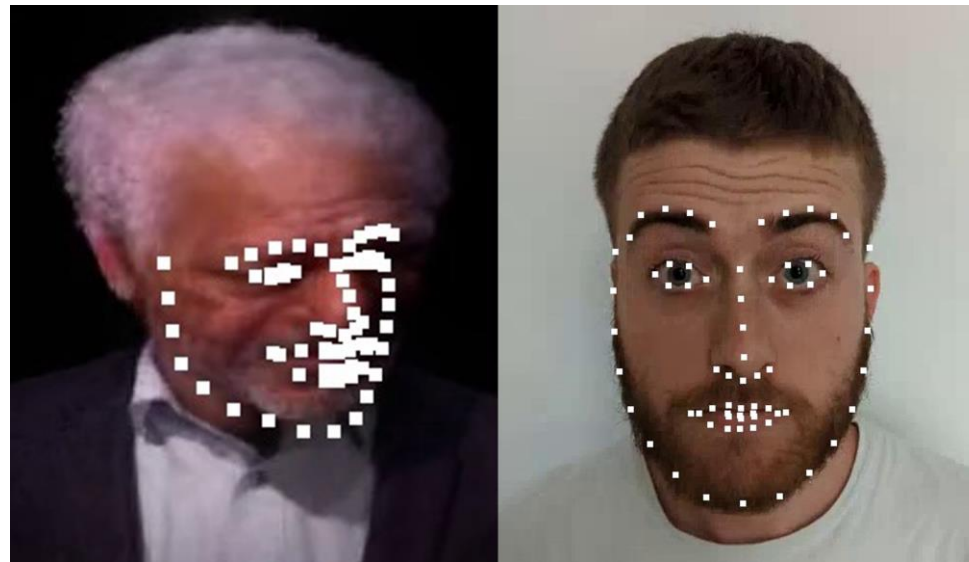
- ✓ In order to evaluate the robustness of real time deepfakes, we started by implementing the First Order Motion Model avatarify and then examining edge cases on it.
- ✓ Since we focused on the implementation of avatarify which work in a way that a face in a source image is animated according to the motion of a driving video, we tried to test its limitations by passing driving videos with various facial gestures and head tilt as input. This includes winks, tongue out, smiles, eyebrow raise, different head tilts, etc.
- ✓ From the output videos, it was clear that the avatarify does not work on certain artifacts. For example, avatarify is not able to create videos with a tongue out at all, as well as creating a proper dental structure. However, even when it fails to copy certain facial gestures to a source image perfectly, the outputs still look relatively believable.
- ✓ The more significant limitation we identified in the implementation, is the creation of a video that includes head tilts, or more precisely, that the extraction of the keypoints from the faces in the input videos is more complex.

First Order Model Robustness



Failures Detection

- ✓ In order to test our proposed method, we fed it with two videos – authentic video and a deepfake of Morgan Freeman, derived from the authentic video as a driving video using the First Order Motion model. When we ran the method on each of the two videos, we found that indeed no failures were detected in the original video, while in the deepfake they were.
- ✓ Note that in some cases the face in the target image is distorted enough that it is not possible to identify the face, and therefore to determine the landmarks. This fact can be exploited for further improvements in the failure detection task.



Conclusions

- ✓ As the impact of social media on our world grows moment by moment, along with the negligible cost and lack of need for experience enabling almost anyone to create high quality deepfakes, this emerging technology poses a serious threat to society.
- ✓ In this work we experimented with an implementation of real time deepfakes, the first order motion model-based avatarify, proposed by Siarohin et al., and evaluated its robustness by examining various end cases on it.
- ✓ We have learned that avatarify implementations are not robust, and that various facial gestures and artifacts can cause significant failures in the target videos. It seems that most failures caused by head tilt and rotations in the driving video, as well as gestures which include objects that the model does not recognize in the source image, such as tongue or teeth.
- ✓ After demonstrating failures in the implementation, we also provided a method to utilize them for failures detection. We were able to successfully demonstrate our method on two videos – authentic video and a deepfake. No failures were detected in the original video, while in the deepfake they were.



Thank You