DeepFake Videos: Image Animation

David Weinflash
Department of Computer Science
University of California, Santa Barbara
dweinflash@ucsb.edu

Introduction

Deepfake videos, or the synthetic videos derived from altering a person in original content, have become more ubiquitous as machine learning algorithms have become smarter and computer hardware more powerful. Currently, the most prevalent application of deepfake videos may be character replacement – where, for example, an actor's face in an original film is replaced with the face of another actor. An equally disturbing application in deepfakes, however, is blackmail. In deepfake blackmail, counterfeit yet convincing videos are generated to falsely incriminate a victim. Perhaps the most effective way to generate such counterfeit videos is by way of *image animation*, in which an object in a source image is animated according to the motion of a driving video. To evaluate the effectiveness and analyze the risks introduced by such techniques, it is necessary to critically evaluate the latest and greatest research. Thus, the following report examines the image generation techniques proposed in *First Order Motion Model for Image Animation*, a research paper introduced at the 2019 Advances in Neural Information Processing Systems (NeurIPS) conference. After creating and evaluating deepfake videos, the content is then used to aid the creation of forensic tools to help accurately distinguish between real and fake media content.

Keywords: DeepFake, Image Animation, Generators

CS281: Advanced Topics in Computer Vision, UC Santa Barbara

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 Overview

1.1 Problem

Image animation, or the task of automatically synthesizing videos by combining the contents of a source image with the motion patterns derived from a driving video, has often been achieved by utilizing Generative Adversial Networks (GANs) and Variational Auto-Encoders (VAEs). This approach, however effective, relies on pretrained models built upon ground-truth data annotations. Such required annotated datasets are not, unfortunately, widely available and apply to only a small subset of object categories. Thus, GANs and VAEs are limited in their scope and efficiency, constrained to animating only the images available in costly annotated datasets.

In order to address the data limitation issue constraining GANs and VAEs, the authors of *First Order Motion Model for Image Animation* propose a framework that achieves image animation without relying on prior information or annotated datasets. Specifically, the authors introduce a framework that does not depend on labels or custom training procedures specific to individual objects. And, unlike previous computer graphic solutions, the authors' proposed system tackles image animation without relying on prior information about the animated objects (*e.g.* 3D models). Indeed, the framework proposed by Siarohin *et al.* expands the scope of the state of the art in image animation, allowing any arbitrary object of the same object category to be animated according to the motions of a driving video.

1.2 Framework

The framework proposed by Siarohin *et al.* is composed of two main modules, or a *motion estimation module* and an *image generation module*. As a whole, the framework is tasked with animating an object depicted in a source image S based upon the motion of a similar object in a driving video D. In order to accomplish such a task, the framework is first trained to reconstruct a set of training videos, where an encoder-decoder network a part of the *motion estimation module* learns to

encode the motion between frame pairs using keypoint displacements and local affine transformations. After training, the framework is deployed at test time to use the motion learned in the *motion estimation module* as input to the *image generation module*, where a convolutional neural network renders the source image *S* moving in accordance with the motion learned in the driving video *D*. The following subsections provide more detail on each module and explain how the system as a whole achieves image animation without prior information or labeled datasets.

1.2.1 Motion Estimation Module. Primarily, the purpose of the *motion estimation module* is to predict a dense motion field from a frame *d* of the driving video *D* and align it to the source image *S*. By modeling the motion field with a function that maps each pixel location in *D* to its corresponding location in *S*, the *motion estimation module* successfully aligns the feature maps computed from *S* with the object pose in *D*. In order to estimate the corresponding keypoints between the driving video and the source image, an encoder-decoder network is employed with a U-Net architecture and final softmax layer. The output of the network is thus a collection of predicted heatmaps that can be interpreted as keypoint detection confidence maps. Finally, after aligning and outputing the dense motion field, the *motion estimation module* outputs an occlusion mask, allowing the generator in the *image generation module* to determine which image parts of the driving video can be reconstructed by warping the source image and which parts should be inpainted, or inferred by the generator.

1.2.2 Image Generation Module. Given the source image and the output of the *motion estimation module* – or a dense motion field and an occlusion mask – the *image generation module* produces an animated version of the source image that mimics the motion found in the driving video. Specifically, the *image generation module* utilizes a convolutional neural network to generate a moving version of the source image, using the derived keypoints of the source image and driving video to transform the source image. Based on the input of the occlusion mask, the generator network either uses existing features to warp the source image or infers and inpaints new image features to match the motion of the driving video. What results is a sequential collection of source image frames that imitate the motion found in the driving video.

2 Evaluation

Among all the advancements in image animation proposed by Siarhohin *et al.*, perhaps the most significant is the ability to achieve image animation without relying on prior information or annotated datasets. Indeed, accomplishing such a feat would save significant time and resources when performing image animation. In order to verify this and the work's other top claims, I performed a range of different experiments, deploying the framework proposed by Siarhohin *et al.* to perform image animation on my own custom dataset. What immediately became clear is that Siarhohin's framework is very successful at transferring motion from a driving video and applying it convincingly to a source video; the framework, trained without prior information or annotated datasets, is indeed remarkably effective. To justify the other claims set forth in the paper, I performed the following experiments, analyzing how well the paper's proclaimed accomplishments matched my own personal results.

Claim #1: *Our method significantly outperforms state-of-the-art image animation methods and can handle high-resolution datasets where other approaches generally fail.*

In order to verify this claim, I tested the *First Order Motion Model* on a dataset that included both high and low-resolution driving videos. Specifically, the bitrates of the input videos ranged from 110 Kbps on the low-end to 8 Mbps on the high-end. Qualitative methods were used to judge the effectiveness of the model in transferring motion from the driving video to the source image, where a better quality source video translates to a more convincing deepfake video. Suprisingly, the model performed best when analyzing mid-resolution videos, or videos with a bitrate of 1 Mbps. Output deepfake videos generated from very high or very low-resolution driving videos were less convincing, as demonstrated by Figure 1.

Claim #2: *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.*

According to Siarohin *et al.*, the image generator a part of the *First Order Motion Model* is capable of inferring and inpainting features in the source video not available in the original source image. The image generator accomplishes such a feat with the help of

Figure 1. The model performed best when analyzing mid-resolution videos. Driving video bitrates ordered clockwise from the top-left: 110 Kbps, 1 Mbps, 8 Mbps.

Figure 2. The model struggled to inpaint features not originally illustrated in the source image.

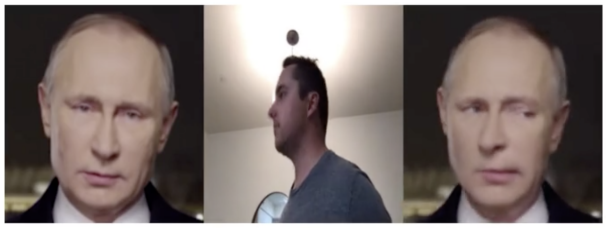

Figure 3. Model performance will suffer if the driving video and source image do not share similar poses.

an occlusion mask, indicating which features should be warped and which should be inferred. While the authors demonstrated the effectiveness of the occlusion mask in generating full-body animations with the *TaiChi* dataset, the mask was less effective in reconstructing facial gestures in my own experiments. Figure 2 demonstrates the image generator's struggle when attempting to inpaint facial features not originally illustrated in the source image.

Claim #3: *It's important to note that one limitation of transferring relative motion is that we need to assume that the objects in S_1 and D_1 have similar poses.*

As expected, the image generator does indeed perform poorly when attempting to transfer the motions from a video of one pose to a source image of another pose. To demonstrate this limitation, I attempted to transfer the video gestures from the profile of a face to an image of a face looking directly at the camera. The results were less than convincing, indicating the importance of pose when transferring the animation of one object to another. Figure 3 illustrates this limitation and suggests an avenue for improvement in image animation.

References

[1] Siarohin, Aliaksandr and Lathuliere, Stephane and Tulyakov, Sergey and Ricci, Elisa *First Order Motion for Image Animation.*

DeepFake Videos: Video Forensics

David Weinflash
Department of Computer Science
University of California, Santa Barbara
dweinflash@ucsb.edu

Introduction

Just as advances in machine learning gave rise to DeepFake videos, so to can modern machine learning models be put to use to realiably detect and segment fake media content. Such an application of machine learning is arguably more important, as the rise of DeepFake videos brings with it a long list of dangerous societal side effects – namely distrust in media, blackmail and cyber bullying, to name a few. Indeed, machine learning may be the solution best positioned to detect modern DeepFake videos, as traditional video forensic techniques oftentime rely on predictable and conventional forgery strategies to detect manipulated videos. Recognizing this opportunity, the authors of *Video Face Manipulation Detection Through Ensemble of CNNs* introduce their own specialized deep learning system, utilizing their framework to efficiently distinguish between real and altered images. Using an ensemble of state-of-the-art models, attention mechanisms and a siamese training strategy, Bonettini *et al.* demonstrate that computer vision models may be applied effectively to extricate DeepFake videos from original media content. By creatively analyzing the results of their framework, the authors uncover the footprints of altered videos and pave the way for future forensic techniques to more efficiently and effectively identify DeepFake videos.

Keywords: DeepFake, Deep Learning, Forensics

CS281: Advanced Topics in Computer Vision, UC Santa Barbara

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 Overview

1.1 Problem

Identifying DeepFake videos in real-world scenarios is no easy task. For one, the video manipulation footprints left behind by generative networks are often very subtle and difficult to detect. Just as generative networks are hard to formally model and explain, so too are their footprints difficult to reliably predict and anticipate. What is more, the initial DeepFake products output from generative networks often undergo many different modifications and alterations as they are shared across social media platforms. Resizing, reformatting and other coding steps all make detecting DeepFake content that much more difficult. Finally, detecting the ever growing production of DeepFakes across the internet requires a solution that is both efficient and scaleable, meaning that the most promising DeepFake detectors must be computationally inexpensive and lightweight.

Recognizing these challenges, the authors of *Video Face Manipulation Detection Through Ensemble of CNNs* propose a modern solution that makes use of the most accurate computer vision models available to detect facial manipulation artifacts in a minimal amount of time. Specifically, the framework proposed by Bonettini *et al.* is capable of analyzing at least 4,000 videos in less than 9 hours, requires at most a single NVIDIA P100 GPU for top performance and occupies less than 1GB of disk space. What follows is an analysis of the author's proposed framework and the effectiveness of their solution, where I experiment with deploying the system on my own custom dataset in an attempt to efficiently detect DeepFake videos and recreate the author's publshed results.

1.2 Framework

Built off the premise that model ensembling leads to better prediction performance, Bonettini *et al.* propose a system that combines multiple Convolutional Neural Network (CNN) models to perform binary classification, predicting whether or not an input image is the product of a deep neural network. Specifically,

convolutional neural networks obtained from the EfficientNet family of models are used to build the system, as EfficientNets have been shown to be among the most accurate (achieved 83.8% accuracy when categorizing the ImageNet dataset) and least computationally expensive (19 million parameters, 4.2 billion FLOPS) models in today's top performing class of neural networks. In order to effectively apply the EfficientNet models to DeepFake detection, the author's made a few key adjustments. In particular, Bonettini *et al.* made use of (i) attention layers and (ii) a siamese training strategy to deploy a system that most effectively and efficiently identifies DeepFake videos. The following sections explore these custom modifications in detail and suggest why the Bonettini *et al.* system is able to so accurately and convincingly identify fake media content.

1.2.1 Attention Layers The benefits of attention layers in a convolutional neural network are twofold: First, attention layers allow the network to learn what regions of the input's feature maps are most important to analyze when performing classification; second, attention layers shed light on the network's thinking, allowing network operators to understand what parts of an image are most informative to a convolutional neural network. Such a step towards explainability is helpful not only for image forensics but for the field of deep learning as a whole, as modern deep learning models are often interpreted as "black boxes" where the specific origin of results is difficult to derive.

1.2.2 Siamese Training During model training, a face is extracted from an input video frame and considered for DeepFake classification. Using facial features as the primary indicator of fake media content – as generated features have been shown to typically reside around the face – the model attempts to learn which features correspond to a generated video and which features correspond to original content. To assist with the learning process, the model is trained using a siamese training strategy, where additional information about the data is extrapolated and more generalizabilities are uncovered. Ultimately, the model learns the broadest representations of facial features to most accurately identify altered media content, categorizing images as either real or fake.

2 Evaluation

To evaluate the DeepFake detection system proposed by Bonettini *et al.*, I trained the network on my own collection of original and DeepFake videos, analyzing the predictions output by the network to measure the model's success in distinguishing between real and fake videos. Results from the network are output on a frame by frame basis, where each frame of an input video is assigned a score ranging from 0 to 1, with a score of 0 indicating "real content" and 1 indicating "fake content." Video prediction scores are then derived by averaging the scores of all frames in a video.

In each of my experiments, the network was able to accurately determine which of my videos was manipulated and which was legitimate. However, certain properties of a video played a large role in determining the confidence of the model's prediction. The following subsections – organized by the claims set forth in the paper – provide a critical analysis of the author's DeepFake detection system and demonstrate which video properties are most influential when trying to identify fake media content.

Claim #1: *Network fusion helps both the accuracy of the DeepFake detection and the quality of the detection.*

The group of convolutional neural networks ensembled by Bonettini *et al.* did indeed produce accurate results. In fact, for each one of my experiments, the EfficientNet models were able to correctly predict which video was an original and which was a fake. As Figure 1 demonstrates, such an accomplishment is no easy task, as both the real and fake videos analyzed by the model appear to be legitimate to the naked eye.

Claim #2: *During training and validation, to make our models more robust, we perform data augmentation operations (downscaling, horizontal flipping, noise addition, etc.) on the input faces.*

To measure the robustness of the authors' model, I performed a series of data augmentation operations on my own DeepFake videos before inputting them into the model. As Figure 2 illustrates, I tested the network on videos that had been downscaled, rotated, posed and blurred. Interestingly, the model most confidently predicted fraud on videos that had been blurred with a gaussian noise filter. Figure 3 indicates which data augmentation operations were the most impactful, where, in the end, no augmented videos were able to score lower than unaltered content.

Claim #3: *Roughly modeled eyes and teeth, showing excessively white regions, are the main trademarks of DeepFake generation methods.*

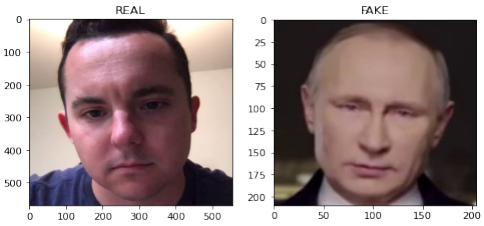

Figure 1. Example of a real video and fake video input into the network ensemble for classification.

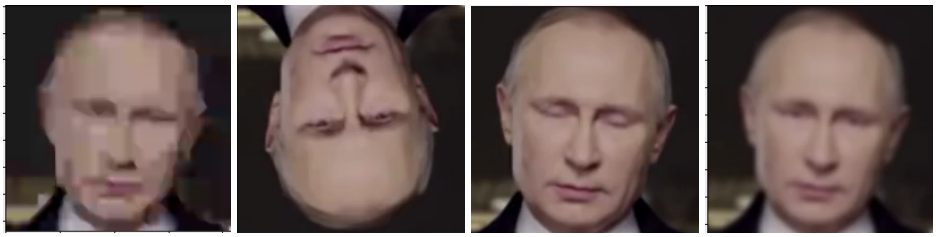

Figure 2. Videos were altered with various data augmentation techniques to test the robustness of the classification system. From the left: Downscale, Rotation, Closed Eyes & Mouth, Gaussian Blur

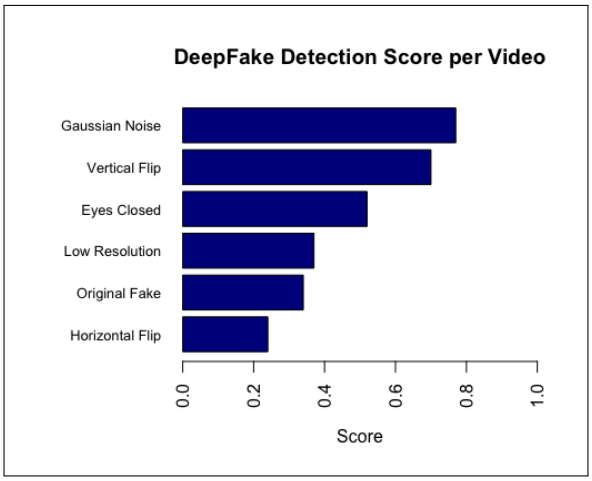

Figure 3. DeepFake detection scores for fake, augmented videos. A score close to 0 predicts REAL, while a score close to 1 predicts FAKE. In my experiments, real videos scored in the $0 - 0.05$ range.

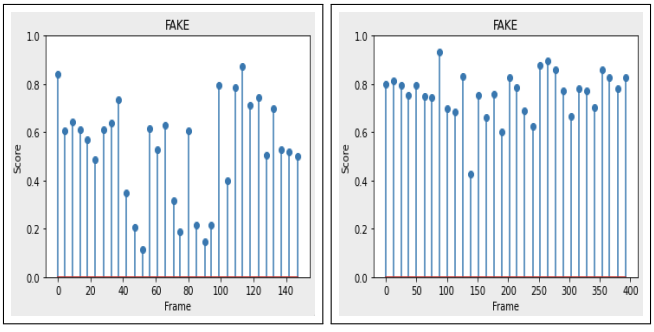

Figure 4. Videos augmented with random noise (right) were considered to be more fraudulent than videos that hid facial features (left).

In order to test this claim I generated a DeepFake video where the character in the video hid his teeth and eyes. While the network was able to correctly categorize the video as fake, it considered the video to be more real than other fake video counterparts. Fake videos augmented with gaussian blur, for instance, were considered to be more fraudulent than fake videos not showing teeth or eyes. Figure 4 demonstrates this effect and suggests what data augmentation operations future works should take into account when developing image forensic systems.

References

[1] Bonettini, Nicolo and Cannas, Edoardo Daniele and Mandelli, Sara and Bondi, Luca and Bestagini, Paolo and Tubaro, Stefano *Video Face Manipulation Detection Through Ensemble of CNNs.*