DeepFake Detection Using Recurrent Neural Networks

Mrs.Pallavi N R
Assistant Professor
Computer Science and Engineering
BGS Institute of Technology
Adichunchanagiri University

Akanksh G S
20CSE036
Computer Science and Engineering
BGS Institute of Technology
Adichunchanagiri University

Abstract:

In recent months, the proliferation of machine learning-based software tools has facilitated the creation of convincing face swaps in videos, resulting in what are commonly referred to as "deepfake" videos. These videos can be highly deceptive, leaving minimal traces of manipulation and posing significant risks in various scenarios, including political manipulation, blackmail, and the fabrication of terrorism events.To address this challenge, this paper proposes a temporal-aware pipeline for the automatic detection of deepfake videos. Our approach utilizes a convolutional neural network (CNN) to extract frame-level features from the videos. These features are then employed to train a recurrent neural network (RNN), enabling the model to classify whether a video has undergone manipulation or not.We evaluate our method using a diverse dataset of deepfake videos collected from multiple online sources. Our results demonstrate the effectiveness of our approach, achieving competitive performance in detecting manipulated videos while employing a simple architecture.By leveraging the temporal information present in videos, our system offers a robust solution for identifying deepfake content, thereby mitigating the potential risks associated with the proliferation of deceptive media..

While benign uses of deepfake technology exist, the majority of applications have been malicious, including the creation of fake celebrity pornography and revenge porn. The realistic nature of deepfake videos also poses risks for generating pedopornographic material, fake news, surveillance footage, and hoaxes, contributing to political tensions and security concerns.

Given the potential for misuse, researchers in artificial intelligence must consider the dual-use nature of their work. In response to the malicious applications of deepfake technology, this paper presents a novel solution for detecting deepfake videos.

The main contributions of this work include a two-stage analysis comprising a CNN for frame-level feature extraction and a temporally-aware RNN for capturing temporal inconsistencies introduced by face-swapping. The proposed method is evaluated on a dataset of 600 videos, half of which are deepfakes collected from various video hosting websites. Experimental results demonstrate the effectiveness of the approach, achieving a 94% higher accuracy in detecting deepfake manipulations compared to a random detector baseline in a balanced setting.

1. INTRODUCTION

The practice of swapping faces in images dates back to as early as 1865, as evidenced by an iconic lithograph featuring U.S. President Abraham Lincoln. Following Lincoln's assassination, the high demand for lithographs led to the rapid creation of images featuring his head superimposed on various bodies, including that of Southern politician John Calhoun.

Recent advancements in image and video manipulation, driven by tools like TensorFlow and Keras, have revolutionized the field. Convolutional autoencoders and generative adversarial network (GAN) models have democratized image and video tampering, making it accessible to individuals with basic computer skills. Applications such as FaceApp and FakeApp leverage these technologies to generate realistic facial transformations and deepfake videos, respectively.

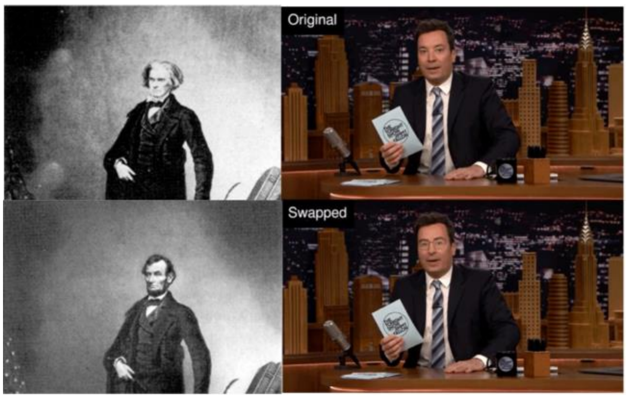

Figure 1. Face swapping is not new. Examples such as the swap of U.S. President Lincoln’s head with politician John Calhoun’s body were produced in mid-19th century (left). Modern tools like FakeApp [2] have made it easy for anyone to produce “deepfakes”, such as the one

swapping the heads of late-night TV hosts Jimmy Fallon and John Oliver (right).

2. Related Work

The field of digital media forensics focuses on developing technologies to assess the integrity of images and videos.

Both feature-based and CNN-based methods have been explored for integrity analysis. In video-based forensics, solutions often target computationally cheap manipulations like dropped frames or copy-move operations. Face-based manipulation detection techniques distinguish between computer-generated and natural faces. Biometric studies have proposed methods to detect morphed faces using pretrained deep CNNs. A new dataset by Rossler et al. contains half a million edited images generated with feature-based face editing.

Several approaches for face manipulation in videos have been proposed, including real-time expression transfer and facial reenactment systems like Face2Face. Deep learning techniques, such as generative adversarial networks (GANs), have been used for face attribute alterations like aging or skin color changes. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, are used for learning long-term dependencies in data sequences. Combining LSTMs with CNNs enables rich visual description and long-term temporal memory, making them effective for various computer vision tasks involving sequences like activity recognition and human re-identification in videos.

3. Deepfake Videos Exposed

The generation process of a deepfake video, as implemented by FakeApp, introduces both intra-frame and temporal inconsistencies that can be leveraged for detection purposes. By understanding how deepfake videos are generated, we can better comprehend why these anomalies occur and how they can be utilized for detection.FakeApp employs deep learning techniques, such as generative adversarial networks (GANs), to synthesize realistic facial transformations in videos. These transformations involve replacing the face of one individual in a video with the face of another individual. To achieve this, the algorithm learns from a large dataset of images and videos to generate realistic facial expressions and movements.During the generation process, intra-frame inconsistencies may arise due to discrepancies in facial features, lighting conditions, or facial expressions between the original and manipulated frames. Additionally, temporal inconsistencies can occur between frames, resulting from discrepancies in facial movements and expressions over time.Exploiting these anomalies involves analyzing the discrepancies between consecutive frames in the video. By identifying inconsistencies in facial features, expressions, or movements within individual frames and across frames, we can effectively distinguish between authentic videos and deepfake manipulations. This understanding of the generation process enables us to develop detection algorithms that exploit these anomalies to identify deepfake videos accurately.

3.1 Creating Deepfake Videos

Deep learning techniques, particularly autoencoders, have been effectively utilized to improve image compression performance. Autoencoders are employed for dimensionality reduction and generating compact representations of images, resulting in enhanced compression compared to existing standards. These compressed representations, or latent vectors, serve as a crucial component in the faceswapping capabilities of deepfake technologies.

The faceswapping process involves two key insights. Firstly, convolutional autoencoders are adept at extracting compressed representations of images while minimizing loss functions, leading to better compression performance. Secondly, the use of two sets of encoder-decoder networks with shared weights facilitates the faceswapping process.

In the training and generation phases of deepfake video creation, these ideas are employed to ensure both latent faces are encoded using the same features. This is achieved by having two networks share the same encoder but use separate decoders. During a faceswapping operation, the input face is encoded and decoded using the decoder associated with the target face, enabling seamless transformation.

3.1.1 Training

To facilitate the training process of autoencoders for faceswapping, two sets of training images are required. The first set comprises samples of the original face to be replaced, which can be obtained from the target video. This set can be augmented with images from other sources to enhance realism. The second set contains the desired face to be swapped into the target video. Ideally, both the original and target faces should have similar viewing and illumination conditions, though this is often not the case due to various factors such as multiple camera views and differences in lighting conditions or video codecs. These discrepancies can result in swapped faces that appear visually inconsistent with the rest of the scene, providing an opportunity for detection.It is crucial to note that training two autoencoders separately can lead to incompatible results. If autoencoders are trained independently on different sets of faces, their latent spaces and representations may differ. To address this, the autoencoders can share weights for the encoder networks while utilizing separate decoders. During training, each decoder is trained with faces from only one subject, but all latent faces are produced by the same encoder. This forces the encoder to identify common features in both faces, leveraging the inherent shared traits of all human faces, such as the number and position of eyes and nose. This approach ensures compatibility between the autoencoders and enables more realistic faceswapping.

3.1.2 Video Generation

After completing the training process, we can use the decoder network trained on faces of the subject we want to insert into the video to reconstruct a face from the latent representation of the original subject present in the video. This process, depicted in Figure 2, is repeated for each frame in the video where faceswapping is desired. However, before passing the face region to the trained autoencoder, a face detector is typically employed to extract only the relevant facial region. This step can introduce scene inconsistencies between the swapped face and the rest of the scene, as the encoder lacks awareness of skin tones and other scene details, often resulting in boundary effects due to seamed fusion.A third weakness we exploit is inherent to the generation process of the final video itself. Since the autoencoder operates frame-by-frame, it lacks temporal awareness and is unaware of any previously generated faces. This can lead to anomalies such as inconsistent choice of illuminants between frames, resulting in a flickering phenomenon in the face region commonly observed in fake videos. While this phenomenon may be imperceptible to the naked eye in well-tuned deepfake manipulations, it can be detected by a pixel-level CNN feature extractor. The issue of incorrect color constancy in CNN-generated videos remains an ongoing research challenge in computer vision. Therefore, it is not surprising that autoencoders trained on constrained data fail to render illuminants correctly.

4. Recurrent Network for Deepfake Detection

In this section, we introduce our end-to-end trainablerecurrent deepfake video detection system (refer to Figure 3). The proposed system consists of a convolutional LSTM structure designed to process sequences of frames. There are two key components within the convolutional LSTM:

1. Convolutional Neural Network (CNN) for frame feature extraction.
2. Long Short-Term Memory (LSTM) for temporal sequence analysis.

When presented with an unseen test sequence, we utilize the CNN to generate a set of features for each frame. Subsequently, we concatenate the features extracted from multiple consecutive frames and feed them into the LSTM for further analysis. Finally, the system produces an estimate of the likelihood that the sequence is either a deepfake or an unmanipulated video.

4.1 Convolutional LSTM

In the proposed approach, a convolutional LSTM is utilized to generate a temporal sequence descriptor for identifying image manipulation within the frame sequence (refer to Figure 3). To enable end-to-end learning, fully-connected layers are integrated to map the high-dimensional LSTM descriptor to a final detection probability. Specifically, the shallow network comprises two fully-connected layers and one dropout layer to mitigate training overfitting. The convolutional LSTM can be divided into a CNN and an LSTM, each of which is described separately below.

CNN for Feature Extraction: Drawing inspiration from its success in the IEEE Signal Processing Society Camera Model Identification Challenge, we adopt InceptionV3 with the fully-connected layer at the top of the network removed to directly produce a deep representation of each frame using the ImageNet pre-trained model. Consistent with prior work, we refrain from fine-tuning the network. The resulting 2048-dimensional feature vectors obtained after the last pooling layers serve as the sequential LSTM input.

LSTM for Sequence Processing: Assuming a sequence of CNN feature vectors of input frames as input and a 2-node neural network with probabilities indicating whether the sequence belongs to a deepfake video or an untampered video, our challenge is to design a model that can recursively process a sequence meaningfully. To address this, we employ a 2048-wide LSTM unit with a dropout probability of 0.5, which adequately fulfills our requirements. During training, our LSTM model accepts a sequence of 2048-dimensional ImageNet feature vectors, followed by a 512 fully-connected layer with a dropout probability of 0.5. Subsequently, a softmax layer computes the probabilities of the frame sequence being either pristine or deepfake. It is important to note that the LSTM module serves as an intermediate unit in our pipeline, trained entirely end-to-end without the need for auxiliary loss functions.

5. Experiments

In this section we report the details about our experiments. First, we describe our dataset. Then, we provide details of the experimental settings to ensure reproducibility and end up by analyzing the reported results.

5.1 Dataset

For this study, we curated a dataset comprising 300 deepfake videos sourced from various video-hosting platforms. Additionally, we augmented this dataset with 300 videos randomly extracted from the HOHA dataset [25], resulting in a total of 600 videos for analysis. We opted to utilize the HOHA dataset as a source of untampered videos due to its comprehensive collection of realistic sequences from well-known movies, particularly focusing on human actions. Given that a significant portion of deepfake videos are generated using segments from mainstream films, incorporating videos from the HOHA dataset ensures that our system learns to discern manipulation cues specific to deepfake videos, rather than merely memorizing semantic content from the two distinct classes of videos present in our final dataset.

5.2 Parameter Settings

Initially, we employed a random split of 70/15/15 to create three distinct sets for training, validation, and testing, respectively. This split was performed in a balanced manner, ensuring an equal distribution of videos from both classes in each set. Specifically, we first split the 300 deepfake videos and then repeated the process for the 300 non-manipulated videos. This approach guarantees that each final set comprises precisely 50% of videos from each class, enabling us to report our results in terms of accuracy without introducing biases related to the frequency of appearance of each class or the necessity of employing regularization techniques during training.

Regarding the preprocessing of the video sequences, the following steps were undertaken:

- Channel mean subtraction for each channel.
- Resizing of each frame to dimensions of 299×299.
- Sub-sequence sampling to control the length of the input sequence, with lengths of N=20, 40, and 80 frames. This exploration allows us to determine the optimal number of frames required per video for accurate detection.

For the optimization process during end-to-end training of the complete model, the Adam optimizer was employed with a learning rate of 1e−5 and a decay rate of 1e−6.

5.3 Results

To address the presence of deepfake manipulations occurring in only a small portion of videos, where the target face may appear briefly, we adopted a strategy to extract continuous subsequences of fixed frame length from every video in the training, validation, and test splits. These subsequences served as the input for our system, enabling us to effectively capture and analyze localized manipulations.

In Table 1, we provide an overview of our system's performance in terms of detection accuracy using sub-sequences of varying lengths, specifically N=20, 40, and 80 frames. These frame sequences were extracted sequentially, without any frame skips, from each video. The entire pipeline underwent end-to-end training until we reached a 10-epoch loss plateau in the validation set.

Model	Training acc. (%)	Validation acc. (%)	Test acc. (%)
Conv-LSTM, 20 frames	99.5	96.9	96.7
Conv-LSTM, 40 frames	99.3	97.1	97.1
Conv-LSTM, 80 frames	99.7	97.2	97.1

Table 1. Classification results of our dataset splits using video subsequences with different lengths.

As we can observe in our results, with less than 2 seconds of video (40 frames for videos sampled at 24 frames per second) our system can accurately predict if the fragment being analyzed comes from a deepfake video or not with an accuracy greater than 97%.

Conclusion

In this paper, we have introduced a temporal-aware system designed to automatically identify deepfake videos. Through our experiments utilizing a substantial collection of manipulated videos, we have demonstrated that employing a straightforward convolutional LSTM structure enables accurate prediction of whether a video has undergone manipulation with as little as 2 seconds of video data.

We contend that our research provides a potent initial defense mechanism for identifying fake media generated through the methodologies outlined in this paper. Our findings underscore the effectiveness of our system in achieving competitive results for this task, despite its utilization of a simple pipeline architecture. Moving forward, our future endeavors will focus on enhancing the robustness of our system against manipulated videos employing novel techniques that were not encountered during the training phase.

Acknowledgments: This content is derived from research supported by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. Government holds the rights to reproduce and distribute reprints for Governmental purposes, regardless of any copyright notation present. The opinions and conclusions expressed herein belong to the authors and do not necessarily reflect the official policies or endorsements of DARPA, AFRL, or the U.S.

Government.

References

[1] Faceapp. https://www.faceapp.com/. (Accessed on 05/29/2018). 1

[2] Fakeapp. https://www.fakeapp.org/. (Accessed on 05/29/2018). 1, 2

[3] IEEE's Signal Processing Society - Camera Model Identification — Kaggle. https://www.kaggle.com/c/ sp-society-camera-model-identification/ discussion/49299. (Accessed on 05/29/2018). 4

[4] TheOutline: Experts fear face swapping tech could start an international showdown. https://theoutline.com/post/3179/ deepfake-videos-are-freaking-experts-out?zd=1&zi=hbmf4svs. (Accessed on 05/29/2018). 1

[5] What are deepfakes & why the future of porn is terrifying. https://www.highsnobiety.com/p/ what-are-deepfakes-ai-porn/. (Accessed on 05/29/2018). 1

[6] M. Abadi et al. Tensorflow: A system for large-scale machine learning. *Proceedings of the USENIX Conference on Operating Systems Design and Implementation*, 16:265–283, Nov. 2016. Savannah, GA. 1

[7] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. *arXiv:1702.01983*, Feb. 2017. 1, 2

[8] H. Averbuch-Elor et al. Bringing portraits to life. *ACM Transactions on Graphics*, 36(6):196:1–196:13, Nov. 2017. 2

[9] P. Bestagini et al. Local tampering detection in video sequences. *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, pages 488–493, Sept. 2013. Pula, Italy. 2

[10] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. *Proceedings of the ACM Annual Conference on Computer Graphics And Interactive Techniques*, pages 353–360, Aug. 1997. Los Angeles, CA.