Detection of Deepfakes leveraging Video Motion Amplification

Abhilash Sampath , Amudheswaran Sankaranarayanan

**Abstract**— Deepfakes are a type of synthetic media in which a person in an existing image or video is replaced with someone else's likeness. They are created using artificial intelligence (AI) to manipulate the facial features of the original image or video. Deepfakes can be used for a variety of purposes, including entertainment, social engineering, and political propaganda.

In this paper, we propose a new method for detecting deepfakes that leverages a phase-based motion amplified representation of facial temporal dynamics. Our method, called PhaseForensics, is able to detect deepfakes with high accuracy, even when they are created using state-of-the-art AI techniques.

**Index Terms**— Deepfake Detection, Phase based Video Amplification, Video Motion Amplification

Keywords should be taken from the taxonomy (http://tbiom.ieee-biometrics.org). Keywords should closely reflect the topic and should optimally characterize the paper. Use about four key words or phrases in alphabetical order, separated by commas (there should not be a period at the end of the index terms)

CSE 666 Final Project Detection of Deepfakes leveraging Video Motion Amplification

—————————— ◆ ——————————

# 1 Introduction

Deepfakes are a growing threat to the integrity of media. They can be used to create fake news, spread misinformation, and damage the reputations of individuals. Deepfakes are also becoming increasingly difficult to detect.

Traditional deepfake detection methods rely on features such as facial alignment, skin texture, and eye movement. However, these features can be easily manipulated by deepfake creators.

In this paper, we propose a new method for detecting deepfakes that leverages a phase-based motion representation of facial temporal dynamics. Our method, called PhaseForensics, can detect deepfakes with high accuracy, even when they are created using state-of-the-art AI techniques.

# 2 Related Work

# 2.1 Deepfake Identification

Detailed submission As society begins to feel the harmful effects of deepfakes ranging from inter-topic shenanigans to wide-scale falsification of news and spreading false agendas like [[2]](https://www.npr.org/2022/03/16/1087062648/deepfake-video-zelenskyy-experts-war-manipulation-ukraine-russia) “*The video, which shows a rendering of the Ukrainian president appearing to tell his soldiers to lay down their arms and surrender the fight against Russia*”, there is a growing competition between their production and detection. Early research into deepfake detection concentrated on identifying pixel-level artifacts straight from the data, proposing "blind" detectors[[3]](https://nealmangaokar.com/static/media/noisescope.afabec2d.pdf). However, these methods tend to learn the specific artifacts of their training data sets, limiting their applicability and domain transferability to previously unseen videos. They also are more susceptible to adversarial attacks.

## 2.2 Deepfake Detection

For papers accepted for publication, Contrarily, modern deepfake detectors aim to identify unique authenticity cues from real videos, treating them as human watermarks. These may include aspects such as lip movements[[1]](https://arxiv.org/abs/2012.07657) head pose[[5]](https://arxiv.org/abs/2108.12715), heartbeats[[6]](https://arxiv.org/abs/2010.00400) and other natural or physical human features. Since the coherence and correlation of these understandable signals are disrupted in falsified videos, these strategies offer improved adaptability as long as the GAN does not use the specific prior as a loss.

If the above aspects were used as a loss that the GAN aims to minimize, we can leverage motion amplification[[4]](https://www.researchgate.net/publication/342081774_Exposing_AI-generated_videos_with_motion_magnification) to make these artifacts stand out.

# 3 Method

# 3.1 Data Preprocessing

We first preprocess our data to feed it to our model.

First, we select frames from a video file for deepfake detection. We extract the frames by calculating the frame interval, and iterate through the frames to extract the desired frames in a certain interval, out of all the intervals that we have segregated the video into . Then we return the selected frames as a list.

Next, we align and resize a face in an image. We take the input frames, a face detector, a facial landmark predictor, and some optional parameters such as the desired face width, height, and the position of the left eye. WE then align the faces to the same pose across all frames.

Now, for a video, we have a batch of selectively chosen frames and in each frame, we select sample intervals of ω frames from each video for training. These samples are selected uniformly from every percentile of the video the face has been detected, resized to our requirements, and aligned.

We use this batch of frames to feed it to our motion amplification module that amplifies the movements of the face.

To magnify the motion of the face in the selected frames, we used motion magnification techniques. This was done in the frequency domain, using CuPy for efficient computation on the GPU. We computed the phase difference between subsequent frames and magnified it by a factor k. We then created a new image with the original amplitude and the magnified phase, and bandpass filtered it around the desired frequency range.

The idea is that the abnormalities are amplified in a deepfake and in a regular video, this would add some noise artifacts that our model aims to distinguish apart.

Then the amplified frames of that video are resized again to the size of the input layer of the network and processed

# 3.2 Networks used

Our approach here was to use the two commonly used, state of the art , readily available neural net models ResNeXt-based model and an XceptionNet-based model.

# 3.2.1 Custom Resnet : *MyResNeXt*

We utilize ResNet as the base architecture and extends it through inheritance and customization to create a modified ResNet model (MyResNeXt) for Deepfake detection.

In the case of the ResNeXt-based model, we removed the original output layer and replaced it with a new fully connected layer with one output unit.

# 3.2.2 Custom XceptionNet with Pooling

In the case of the ResNeXt-based model, we removed the original output layer and replaced it with a new fully connected layer with one output unit.

The Pooling module performs adaptive pooling [[7]](https://ieeexplore.ieee.org/document/9645730) on the input feature map, producing a fixed-size representation that captures both the average and maximum activations across the spatial dimensions. This pooling operation helps in reducing the spatial dimensions of the feature map while retaining important information. The combination of adaptive average pooling and adaptive max pooling provides a balanced representation that can capture both global and local information from the input.

# 3.3 Algorithm

Our Algorithm now identifies each video to be tested upon, selects a batch of frames for processing. With each chosen face, we identify the face(s) present, normalize them, and feed it to the network to gain an estimate of whether it's a deepfake or an original video. In the case of multiple faces being present in the video, each face is probed whether they are modified, and the cumulative result is portrayed by calculating the mean.

# 3.4 DATASET:

The dataset used is Kaggle's Deepfake Detection Challenge's Deepfake dataset., with the videos compressed to the c23 encoding.

It consists of over 390 original YouTube videos that are used to generate over 1000 deepfakes. The model is tested on 1000 datapoints and tested on 400 samples.

# 4 EXPERIMENTS

# Finally, we made predictions on unseen videos and took the average prediction from all selected frames. We fused the predictions from both the models by taking a weighted average, with the weights being the reciprocals of the models' respective validation losses.

# 5 Results

# 5.1 ResNet Performance

The *MyResNeXt* model achieved an accuracy of 54% on the evaluated dataset. Let's analyse the performance based on the precision, recall, and F1-score for each class:

A picture containing text, screenshot, number, software

Description automatically generated

Fig. 1. The Confusion Matrix for Resnet

Our model has shown the following performance:

* **Precision**
  + Genuine videos (class 0): 0.58
  + DeepFake videos (class 1): 0.50
* **Recall**
  + Genuine videos (class 0): 0.59
  + DeepFake videos (class 1): 0.49
* **F1-score**
  + Genuine videos (class 0): 0.58
  + DeepFake videos (class 1): 0.50
* **Accuracy**: 0.54
* **Macro average F1-score**: 0.54
* **Weighted average F1-score**: 0.54

Our model demonstrates a slightly higher precision for authentic videos (0.58) compared to DeepFakes (0.50), indicating that when it classifies a video as genuine, it is more likely to be accurate than when it labels a video as a DeepFake. The recall for both classes is close, with a slight leaning towards genuine videos (0.59) compared to DeepFakes (0.49).

The F1-scores, which provide a balance between precision and recall, reflect a similar pattern with a higher score for genuine videos (0.58) compared to DeepFakes (0.50). The overall accuracy of the model is 0.54, indicating a performance level slightly better than a random guess.

# A picture containing text, screenshot, font, number Description automatically generated5.2 Xception net Performance

Fig. 2. The Confusion Matrix for Xception net

Looking at the provided metrics, the model demonstrates the following performance:

* **Precision**
  + Genuine videos (class 0): 0.60
  + DeepFake videos (class 1): 0.48
* **Recall**
  + Genuine videos (class 0): 0.36
  + DeepFake videos (class 1): 0.71
* **F1-score**
  + Genuine videos (class 0): 0.45
  + DeepFake videos (class 1): 0.58
* **Accuracy**: 0.52
* **Macro average F1-score**: 0.51
* **Weighted average F1-score**: 0.51

This means that our model has a higher precision for detecting genuine videos (0.60) than DeepFakes (0.48), meaning that when it predicts a video as genuine, it is more likely to be correct than when it predicts a video as a DeepFake. However, our model has a significantly higher recall for DeepFakes (0.71) than genuine videos (0.36), indicating that it is better at capturing all DeepFakes in the dataset at the expense of falsely classifying genuine videos as DeepFakes.

The F1-scores, which balance precision and recall, are relatively low for both classes, with the score for DeepFakes (0.58) being higher than for genuine videos (0.45). The overall accuracy of the model is slightly above a random guess at 0.52.

# A picture containing text, screenshot, font, number Description automatically generated5.3 Ensemble model Performance

Fig. 3. The Confusion Matrix for the Ensemble Model

The ensemble model combines the predictions of two individual models, a ResNeXt-based model and an XceptionNet-based model. The predictions are combined using weights proportional to their validation scores. This combination intends to leverage the strengths of both models to achieve better overall performance.

* **Precision**
  + Genuine videos (class 0): 0.61
  + DeepFake videos (class 1): 0.51
* **Recall**
  + A picture containing text, line, plot, diagram

    Description automatically generatedGenuine videos (class 0): 0.45
  + DeepFake videos (class 1): 0.67
* **F1-score**
  + Genuine videos (class 0): 0.52
  + DeepFake videos (class 1): 0.58
* **Accuracy**: 0.55
* **Macro average F1-score**: 0.55
* **Weighted average F1-score**: 0.54

In the context of the ensemble model, precision refers to the ratio of true positive predictions to the total predicted positives for each class. The model demonstrates higher precision in detecting genuine videos (0.61) compared to DeepFake videos (0.51).

The recall metric, which indicates the proportion of actual positives correctly identified, is higher for DeepFake videos (0.67) compared to genuine videos (0.45). This suggests that the ensemble model is more adept at catching DeepFake videos at the expense of missing some genuine ones.

The F1-scores, which balance the precision and recall, also reflect this pattern. The model achieves a slightly higher score for DeepFake videos (0.58) compared to genuine ones (0.52). The overall accuracy of the ensemble model stands at 55%.

# A picture containing text, line, plot, diagram Description automatically generated5.4 Comparison against the state of the art

Fig. 3. The ROC for the Ensemble Model

This report outlines a comparative analysis of two models used for the task of video deepfake detection - our current ensemble model, which scores a 0.57 on the Receiver Operating Characteristic (ROC) curve, and the state-of-the-art EfficientNet and Vision Transformers model by Coccomini et al.[[7]](https://arxiv.org/pdf/2107.02612v2.pdf), with a ROC score of over 0.95.

Performance Analysis:

1. **ROC Score:** The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. It is an effective method for comparing classifier performance. In this case, the EfficientNet and Vision Transformers model considerably outperforms our ensemble model, with a ROC

Fig. 4. The ROC for the State-of-the-art models

score of 0.95+ compared to 0.57. The significant difference in scores indicates that the SotA model has much better discriminative power to differentiate between real and deepfake videos.

1. **Precision, Recall, and F1 Score:** Our ensemble model shows an approximately balanced precision and recall for both classes, but they are both just above 0.5, which indicates the model struggles to correctly classify deepfakes and real videos. In comparison, a higher precision and recall score for the SotA model would suggest better performance on both fronts.

Possible Reasons for Superior Performance of the SotA Model:

1. **Model Architectures:** EfficientNet and Vision Transformers are advanced architectures that, compared to traditional CNNs, better handle the task of recognizing deepfakes. They use a more structured scaling method and are capable of focusing on significant pixels in the image.
2. **Combination of Techniques:** Coccomini et al. combined EfficientNet and Vision Transformers, thereby leveraging the strengths of both architectures. The ensemble model uses different models, which may not capture complementary information as effectively.
3. **Temporal Information:** The SotA model may make better use of temporal information across video frames, which is essential in deepfake detection.

Suggestions for Improvement:

1. **Use Advanced Architectures:** Consider utilizing more sophisticated architectures, such as EfficientNet or Vision Transformers, that are known for their superior performance in image and video processing tasks.
2. **Temporal Information:** Improve the way temporal information is captured and used in the model. Techniques like 3D convolutions, recurrent neural networks, or even transformers might prove beneficial.
3. **Feature Fusion:** Experiment with a hybrid model that combines features learned from both CNN and transformer-based models. This could help capture complementary information and improve the model's performance.
4. **Training Strategy:** Adopt advanced training strategies such as using a larger batch size, a lower learning rate, or different types of optimization algorithms. Techniques like dropout or batch normalization can also be incorporated to improve model performance.
5. **Data Augmentation:** Employing advanced data augmentation techniques can increase the diversity of training samples, making the model more robust to different deepfake techniques.
6. **Fine-Tuning:** Fine-tuning the model on a specific task or dataset can help improve its performance. Use methods such as early stopping and adaptive learning rates to avoid overfitting and improve generalization.

# Future Work

Our current model leverages video motion magnification to accentuate the temporal changes in the video. This process has shown to be somewhat effective in revealing deepfakes as they often exhibit unnatural or exaggerated motion patterns. However, this approach also introduces extra noise and can complicate the learning task for the model, potentially affecting overall performance.

For future work, we propose to refine our approach to video motion amplification:

* Noise Reduction: We plan to incorporate advanced noise reduction techniques to mitigate the noise introduced by motion magnification.
* Improved Motion Amplification: We will explore more advanced and adaptive motion amplification algorithms, aiming to reveal subtle inconsistencies in deepfakes without overwhelming the model with unnecessary information.
* Model Training with Amplified Videos: Training the model explicitly on a dataset containing motion amplified videos could help the model to better understand and learn from the patterns revealed by the amplification.

Additionally, we aim to implement this on other model architectures and focus on the following aspects:

* Enhanced Temporal Information Processing: We will explore advanced techniques such as 3D convolutions, recurrent neural networks, and transformers to improve our handling of temporal information across video frames. This is especially important in video-based tasks like deepfake detection where temporal consistencies and inconsistencies play a crucial role.
* Feature Fusion: The hybrid model combining the features learned from both CNN and transformer-based models will be one of our primary focuses. This approach might help us capture complementary information, which, in turn, might improve our model's performance.
* Advanced Training Strategy: We aim to adopt advanced training strategies like using a larger batch size, a lower learning rate, or advanced optimization algorithms. Additionally, we will also explore techniques like dropout or batch normalization to improve model robustness and generalization.
* Data Augmentation: We aim to experiment with more advanced data augmentation techniques. The objective is to increase the diversity of the training samples, thereby making the model more robust to different deepfake techniques.
* Fine-Tuning: We plan to fine-tune our model specifically on the deepfake detection task. We will leverage methods such as early stopping and adaptive learning rates to avoid overfitting and improve model generalization.

# Conclusion

The ensemble model exhibits a better performance than the individual models, with an improvement in overall accuracy and in the F1-score for DeepFake detection. It is worth noting that while the model shows higher precision for detecting genuine videos, it achieves higher recall for DeepFake videos, implying that it tends to lean more towards flagging videos as DeepFakes.

This might be a desirable characteristic considering the serious implications of failing to detect DeepFake videos. However, further enhancements could involve working on improving the recall for genuine videos to reduce the number of false alarms.

**References**

[1] [Lips Don't Lie,](https://arxiv.org/abs/2012.07657)

[2] [Deepfake video of Zelenskyy could be 'tip of the iceberg' in info war, experts warn](https://www.npr.org/2022/03/16/1087062648/deepfake-video-zelenskyy-experts-war-manipulation-ukraine-russia)

[3] [NoiseScope: Detecting Deepfake Images in a Blind Setting](https://nealmangaokar.com/static/media/noisescope.afabec2d.pdf)

[4] [Exposing AI-generated videos with motion magnification](https://www.researchgate.net/publication/342081774_Exposing_AI-generated_videos_with_motion_magnification)

[5] [DeepFake Detection with Inconsistent Head Poses: Reproducibility and Analysis](https://arxiv.org/abs/2108.12715)

[6] [DeepFakesON-Phys: DeepFakes Detection based on Heart Rate Estimation](https://arxiv.org/abs/2010.00400)

[7] [Combining EfficientNet and Vision Transformers for Video Deepfake Detection](https://arxiv.org/pdf/2107.02612v2.pdf)

[8] [**Xception + Resnext Ensemble Model**](https://www.kaggle.com/code/khoongweihao/xception-resnext-ensemble-inference)

**[9]** [How Do Deepfakes Move? Motion Magnification for Deepfake Source Detection](https://arxiv.org/pdf/2212.14033.pdf)