Celeb-DF Deepfake Video Detection Using ResNet

Aldi Hilman Ramadhani[1]

[1]College of Computing and Data Science, Nanyang Technological University
50 Nanyang Ave, Singapore 639798
aldi001@e.ntu.edu.sg

Abstract

Deepfake video detection was performed using spatial approaches. Original and deepfake videos are taken from the Celeb-DF dataset and preprocessed with cropping and various data augmentations. Spatial approache using the ResNet models are executed. Different learning rate and learning rate scheduler configurations are used. The best performance with accuracy of 93.36% is retrieved using learning rate of 0.001 and the cosine annealing scheduler.

Introduction

The impact of deepfakes is real and profound, disturbing the integrity of digital media and contributing to misinformation, cyberbullying, political manipulation, and fraud. These manipulations are becoming increasingly indistinguishable from authentic content due to advancements in artificial intelligence, particularly through the use of deep learning technologies like deep neural networks (DNNs) (Yuezun Li and Lyu 2020). The realism and accessibility of deepfake technology have escalated the urgency for effective detection methods.

The necessity of employing deep learning approaches to combat deepfakes stems from their ability to analyze vast amounts of data and learn complex patterns. Some of the common models used for detection include Convolutional Neural Networks (CNNs), Generative Models like Autoencoders and Generative Adversarial Networks (GANs), and Recurrent Neural Networks (RNNs) like LSTM and GRU as the primary tools. Regarding the approaches, there are two main ways to detect deepfake videos: spatial and temporal. The spatial approach primarily utilizes CNNs to process and scrutinize individual frames for visual inconsistencies that are not typically perceptible to the human eye. Temporal techniques, on the other hand, use RNNs like LSTM and GRU to detect anomalies over sequences of frames, identifying unnatural movement or expression transitions that suggest manipulation (Passos et al. 2024).

In this research, we implement deepfake detection models and test them against the Celeb-DF dataset (Yuezun Li and Lyu 2020). The spatial approach is implemented using ResNet and analyzed which parameter performs best. Dataset selection, dataset pre-processing, and model implementation will be explained in the next section.

Dataset

Celeb-DF is a dataset designed to improve deepfake detection by providing high-quality, challenging videos (Yuezun Li and Lyu 2020). The motivation for of this dataset creation is that the common deepfake datasets usually suffer from low visual quality and do not resemble deepfake videos that are circulated on the internet. The published paper mentions that the dataset contains 590 original videos collected from YouTube, and 5,639 corresponding deepfake videos synthesized from the original videos. This dataset presents smoother and perceptually higher quality generated videos that intended to not only resemble circulated deepfake videos but also improve deepfake detection models.

In this research, we utilize a subset of the Celeb-DF dataset for our experiments. We retrieved the latest version of the dataset published by the original author, which contains slightly more videos from the published number on the paper. We randomly select a subset of the deepfake-generated videos due to the dataset consists of a larger number of deepfake videos and we want to prevent class imbalance. In the end, we used a total number of 1050 videos for training, 226 videos for validation, and 226 videos for testing.

Models

Other than ResNet, we use a face detection model to get and retrieved the face. The ResNet model is then used to differentiate those face to determined if a video is a deepfake video or not. If a video has more deepfake per total frame, then the video is labelled as deepfake video. In this section how those model works and integrate will be explained.

Face Detection

On top of the deepfake detection model, we use a face detection model to extract the face from the image. The model identifies the region of the frame with the face and extracts it from the rest of the frame. Finally, only the face image is used to train, validate, and test the model. The result of this process can be seen in Figure 1. Also important to notice this face detection model will not be trained and use a pre-existing weight. We consider this step as a preprocessing step before building the deepfake detection model.

The face detection model itself is called FaceNet (Schroff, Kalenichenko, and Philbin 2015). The FaceNet model works

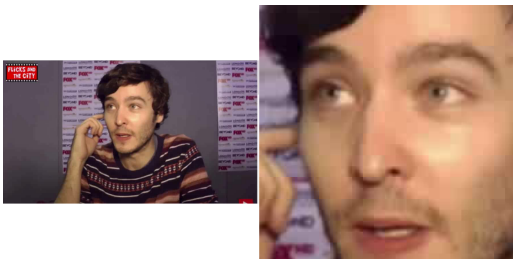

Figure 1: The original image (left) and the result after extracting face (right) using FaceNet model

by directly learning a mapping of face images to a compact Euclidean space. This mapping is accomplished using deep convolutional networks, where the distance between any two face embeddings accurately reflects their facial similarity. The model's strength lies in its ability to perform both face verification and clustering tasks with high accuracy. This is achieved through a training process that optimizes the embedding space so that distances correlate with face similarity, using a triplet loss function that compares a base image against both a positive (same identity) and a negative (different identity) example.

Residual Network

ResNet, short for Residual Network, is a type of convolutional neural network (CNN) architecture that was introduced to address the vanishing gradient problem and enable the training of much deeper networks (He et al. 2015). The main idea behind ResNet is the introduction of "residual blocks" with skip connections, where the input to a block is added to its output, allowing the network to learn identity functions. This helps in mitigating the problem of diminishing feature gradients, making it possible to efficiently train networks with depths of hundreds of layers.

ResNet50 is a version that is used in this report, a specific configuration of ResNet that consists of 50 layers. It's one of the variants available that provides a good balance between complexity and performance. In addition to the model architecture, the initial pre-trained weights were used called the IMAGENET1K_V2. An extra fully connected layer is also used to predict the image class, whether it's a deepfake or not.

Experiment Settings

This section outlines the experimental framework for deepfake identification using our models. The experimental process begins with the preparation of the dataset, which involves configuring and sampling it appropriately for model training. Following this, the experiments progress to evaluate the models on the test dataset, with the aim of ascertaining which model yields the best performance.

Dataset Preparation

We first sample a smaller subset of the dataset by randomly selecting the same amount of real and deepfake videos. We conduct several preprocessing steps. The first step is to crop the images to only include the face area of each frame. These frames are then resized to 224x224, and then several data augmentations are done when building the data loader in the form of random crop, rotation, horizontal flip, and affine and color-jittering. By default, we use a batch size of 32.

Training Settings

We train our models using a total of 1050 real and DeepFake videos. For each videos, we sample 10 images to be fed into the network. The spatial approach, in which the model predicts based on whether an image is deepfake or not, is applied with ResNet model. During training, the training and validation accuracy is calculated image-wise since the model is trained for processing a single image. However, in the testing phase the accuracy is calculated per video labels.

As we want to compare the model performance, we set several global settings with the same hyperparameters to have better comparisons. In the experiments, the Adam optimizer (Kingma and Ba 2014) was used with a learning rate of 0.001 or 0.005. We also use 2 learning rate schedulers, the constant scheduler and cosine annealing.

The loss function we use for training this model is cross-entropy loss. The cross-entropy loss for a classification problem with C classes can be defined as:

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y_i}) \tag{1}$$

where L is the loss function, y is a binary indicator (0 or 1) if class label c is the correct classification for observation o, and $\hat{y}$ is the predicted probability that observation o is of class c.

Testing Settings

We test our model's ability to classify videos as real or deepfake. Hence, we implement a simple voting mechanism. First, we sample 15 images from the video. We intentionally choose odd numbers to avoid the tie between the classes. We then predict those 15 images using the spatial model and choose the video label based on the most voted class. For the temporal approach, the accuracy during the testing calculation method is the same as the training.

Experiment Results

This Table 1 displays the performance of various hyperparameters for ResNet50, such as learning rates and learning rate schedulers. The models exhibit a notable increase in validation and testing accuracy when using a cosine annealing learning rate scheduler, indicating its effectiveness over a constant scheduler, especially with a lower learning rate of 0.001.

For the first ResNet configuration, with a learning rate of 0.005 and a constant scheduler, the model exhibits moderate performance with a training loss of 0.4821 and an accuracy of 82.31%. This performance slightly declines on the validation set, with a loss of 0.5038 and an accuracy of 78.93%, and maintains a similar trend in the testing phase, achieving an accuracy of 79.64%. This suggests that while the model

Table 1: Consolidated Loss and Accuracy Results

Model	Learning Rate		Training		Validation		Testing
	Value	Scheduler	Loss	Accuracy	Loss	Accuracy	Accuracy
ResNet50	0.005	Constant	0.4821	82.31%	0.5038	78.93%	79.64%
ResNet50	0.001	Constant	0.3779	93.24%	0.3911	91.78%	92.92%
ResNet50	0.005	Cosine Annealing	0.4497	85.76%	0.4979	80.84%	80.97%
ResNet50	0.001	Cosine Annealing	0.3237	98.95%	0.3848	92.62%	93.36%

is learning, it might be taking steps that are too large, not allowing it to finely adjust to the nuances of the dataset.

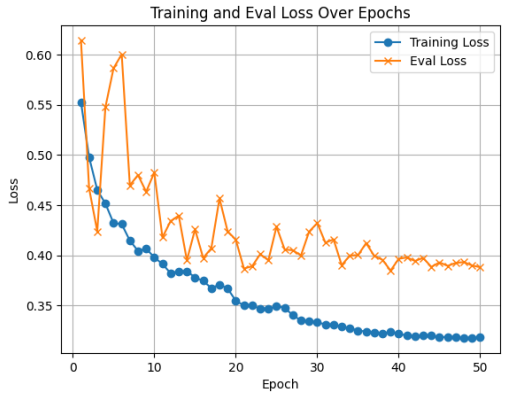

Figure 2: Loss graph for ResNet with 0.001 learning rate and cosine annealing

In contrast, when the learning rate is decreased to 0.001 with the same constant scheduler, the model's performance improves significantly. The training loss is reduced to 0.3779, and the accuracy climbs to 93.24%. This positive trend is also reflected in the validation set, with a lower loss of 0.3911 and a higher accuracy of 91.78%. The testing accuracy further confirms the model's improved generalization at 92.92%. This indicates that a smaller learning rate allows the model to make smaller, more precise updates to its weights, which improves performance.

The ResNet models employing cosine annealing with both learning rates present interesting results. With a higher learning rate of 0.005, the training loss is at 0.4497 with an accuracy of 85.76%, while the validation set shows a loss of 0.4979 and an accuracy of 80.84%. The testing accuracy drops slightly to 80.97%. However, reducing the learning rate to 0.001 while using cosine annealing results in the best performance among the ResNet configurations. The training loss falls to a remarkable low of 0.3237 as seen on Figure 2, paired with a high accuracy of 98.95%. This excellence carries over to the validation and testing sets, with losses of 0.3848 and 0.3926, and accuracies of 92.62% and 93.36%, respectively. These figures suggest that combining a lower learning rate with an adaptive scheduler like cosine annealing can significantly enhance the model's learning capability.

Conclusion

This report details the development and evaluation of deepfake detection models using the Celeb-DF dataset, focusing on spatial approaches with ResNet architectures. This study addresses the growing concern over deepfakes in digital media, which complicate the integrity of content across the internet. By leveraging advanced machine learning techniques, the research presents methods for recognizing deepfake videos by examining frame-by-frame visual inconsistencies. The models, tested under various configurations of learning rates and schedulers, showed that the best results were achieved using a learning rate of 0.001 with a cosine annealing scheduler, achieving an accuracy of 93.36%. This outcome underscores the effectiveness of the chosen approach and the importance of fine-tuning model parameters to optimize deepfake detection.

References

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Passos, L. A.; Jodas, D.; Costa, K. A. P.; Souza Júnior, L. A.; Rodrigues, D.; Del Ser, J.; Camacho, D.; and Papa, J. P. 2024. A review of deep learning-based approaches for deepfake content detection. *Expert Systems*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Yuezun Li, P. S. H. Q., Xin Yang; and Lyu, S. 2020. Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.