Deepfake Video Detection Using Neural Networks

CS3244 Group 5
Cai Ruicong[1](A0189237R), Choo Ze Yuan[2](A0194613A)
Huang Linhang[3](A0177833U), Ye Chenchen[4](A0177915R),
Zhang Xinran[5](A0177827M), Zhou Tianyu[6](A0187444W)
National University of Singapore
21 Lower Kent Ridge Rd, Singapore 119077
e0324821@u.nus.edu[1],e0379456@u.nus.edu[2]
e0261886@u.nus.edu[3] e0261968@u.nus.edu[4]
e0261880@u.nus.edu[5],e0323028@u.nus.edu[6]

Abstract

Deepfake videos are edited videos generated with face swapping technologies through which the faces appeared in the original video are replaced with others' faces. This kind of video editing approach has been widely considered with great potential to facilitate cybercrimes. In this paper we present three neural network models as candidates to detect deepfake videos. The models are evaluated against open-source face forensic dataset. The experimental result and limitation of each model are also discussed.

Introduction

The name Deepfake, is a portmanteau of the two terms, "deep learning" and "fake" (Wikipedia 2020) coined in 2017. Deepfakes are media generated by machine learning, and more commonly, deep learning. They come in the form of images or videos where a person's face is fitted over another person's face. Audio can also be deepfaked. For example, a voice skin, which is like a voice modulator, allows a person to mimic another person's voice (Sample 2020).

Since its inception, Deepfakes have garnered a lot of attention due to their potential to be immensely damaging. In September 2019, cybersecurity company Deeptrace reported that 14,698 deepfake videos were found online (Metz 2019), up drastically from 7,948 in december 2018. This has prompted several Internet companies such as Facebook, AWS, and Microsoft to build the Deepfake Detection Challenge to spur researchers to build innovative technologies to detect Deepfakes (Metz 2019).

Technological advances have made Deepfake generation tools easily accessible to the mass, and this ease of ability to fabricate content has wide implications for Singapore. Firstly, a major concern is with regards to privacy and identity. Netizens were reported to have allegedly inserted faces into pre-existing pornographical videos (Banks 2018) using face-swapping applications such FakeApp to create pornography of celebrities (Romano 2018). Cyber attackers could also employ such techniques to impersonate and penetrate authentication systems to gain illegitimate access (Ikeda 2019). Thus, if the use of Deepfake were to become prevalent in Singapore, it could severely undermine the privacy of Singaporeans.

Secondly, Deepfakes pose a threat to security as they can be employed to falsify speeches and attack political figures. Such abuse can stimulate political or even religious tensions between countries, inflame the public, and lead to violence or even war (Ng and Tang 2019). For instance, a recent Deepfake video of Trump has demonstrated its ability to incite anger, attract attention, and redirect people to petition on the Belgian government to take more urgent climate actions (Schwartz 2018). Singapore is a country that maintains important racial harmony and bilateral diplomatic relations with other countries. Malicious Deepfakes could compromise these relations that were built over years of hard work.

Moreover, Deepfakes has also provided a new approach to make convincing fake media coverages, which can cause large potential social impact (Christian 2018). Deepfakes could heighten disinformation, cause distress to those targeted, and erode the trust of people in media contents as seeing them is no longer commensurate with believing in them.

Such implications can be generalized to the context of Singapore. The recent COVID-19 situation has shown that Singaporeans respond quickly and decisively to announcements made by our leaders. If a Deepfake video of Lee Hsien Loong falsely declaring certain measures were to spread and go viral, it can lead to social unrest and disorder, escalating the severity of the situation and furthering the spread of the virus before it could be taken down,

Considering the aforementioned points that makes our society vulnerable, we propose an application that is capable of classifying whether videos are deepfaked.

Exploiting flaws of Deepfake generation

Deepfake falls into the category of identity manipulation by facial forgeries (Rossler et al. 2019), which works by replacing the face of a person with another (also known as face swapping). Such synthesized faces are generated by deep learning models such as generative adversarial networks (GAN) and autoencoders, which capture facial features such as angle, skin tone, facial expression and lighting and use them to generate feature maps. These feature maps

are then used to create the synthesized faces in deepfakes (Xu 2019).

However, many specific artifacts, such as facial warping, inconsistent eye blinking([44] of Rossler et al. (2019), color texture,shape cues([24,23] of Rossler et al. (2019) and resolution inconsistencies, can arise from the synthesis process, which can be exploited by manipulation detection methods.

In addition, since video manipulation is carried out on a frame-by-frame basis, the above artifacts produced by face manipulations are believed to further manifest themselves as temporal artifacts with inconsistencies across frames. Hence, deepfake videos contain both intra-frame inconsistencies and temporal inconsistencies between frames that can be exploited.

In the following sections, we will expound the architectures that our models are built on and further discuss our experiment and findings. We will require models that allow us to process video data and output a binary classification (Real/Fake), while exploiting the above flaws.

Models and Techniques

The use of Convolutional Neural Network (CNN)

Why use CNN? CNNs are popular deep learning algorithms used for image classification tasks. As videos are basically a sequence of frames, they can be separated into multiple images and are compatible with CNN.

Theoretically, image classification can also be performed by using basic feed-forward neural networks, such as a multilayer perceptron (MLP). After all, an image is just a 2D / 3D matrix of individual pixel values, depending on black-and-white or colour images. Flattening it into a 1D vector would allow us to utilize a normal feed-forward network. However, one would face a few problems in doing so.

The first problem is the extreme scalability of the parameters and weights of the neural network with increasing dimensions, which would become computationally intensive once the image reaches a certain dimension. For illustration, flattening a single (780, 1280) RGB image would result in a (2995200, 1) vector.

Secondly, spatial structure is super important for image data. Collapsing images into one-dimensional structures loses this spatial information, which would be detrimental to the performance of the classifier.

CNN overcomes both shortcomings through the convolution and pooling process, which will be explained in greater details in the next subsection.

How CNN works The main mechanic of CNN is the convolution step, where images are processed in patches of (n * n) squares called filters. Since each patch is still a mini-image, the spatial structure of the original image is retained. The entire image can then be represented by applying the sliding window principle. A layer may have multiple filters, and outputs a feature map, which is a highlight of areas in the input that most activates the filter, per filter. The purpose of these convolution operations is to learn the extraction of high-level features from the images.

The next part is the pooling step, which usually comes after each convolutional layer and is responsible for reducing the spatial size of the result of the convolution step. Pooling neurons have no weights, they simply aggregate inputs using functions such as max or mean. For example, a (3 * 3) kernel of max pooling will look through a 3 by 3 region of the input, and reduce the output to just a single number that is the largest of the region. This helps to reduce the number of parameters (thereby limiting the risk of overfitting), decrease the computational power required to process data through dimensionality reduction, and possibly serves as a noise suppressant.

Finally, a dense fully connected layer is added at the very end, by which dimensionality would have been drastically reduced. This layer then learns the non-linear combinations of the high-level features that are outputted by the previous convolutional layers before giving the classification of the image.

Hence, through these processes, CNN reduces images into a simpler form that is easier to process, in doing so reducing the number of parameters involved, but without losing much information that is critical for getting a good prediction. Overall, CNN is a better fit for working with image datasets as it can be trained to extract features, and thereby better "understand" an image.

Our application greatly benefits from these properties because we are working with massive datasets of images extracted from videos. Multiple CNN layers can also be stacked to learn increasing levels of features, and hopefully pick out the intra-frame inconsistencies created by the Deepfake generation.

Xception Net In early developed CNN-based solutions, the main body focused on detecting manipulations of low effort, such as inconsistencies in features, dropped or duplicated frames. However, such methods are sensitive to operations like resizing and compression (Rossler et al. 2019).

Xception net is based on depthwise separable convolution layers, inspired by InceptionNet. The model has great ability to extract and learn features from face manipulated videos, and is robust to quality of the videos (Rossler et al. 2019). This architecture outperforms Inception V3 especially in larger image classification dataset without increasing the number of parameters, indicating that parameters are used more efficiently (Chollet 2017). Compared to Inception V3 and other earlier algorithms like VGG-16 and ResNet-152, experimentally Xception has shown better classification accuracy on all raw, high and low quality videos dataset (Chollet 2017). As such, we decided to implement this algorithm which is built based on CNN. Additionally, this algorithm briefly is a linear stack of depth convolution layers with residual connections, which is simple to implement using a high-level library like Keras and Tensorflow.

MesoNet MesoNet is another state-of-the-art CNN model that can be used to detect deepfake videos. The name, MesoNet, is derived from its deep learning approach of focusing on the mesoscopic properties of the input images. Mesoscopic analysis is an intermediate approach between microscopic analysis and macroscopic / semantic analysis (human vision).

Deepfake video detection pre-processing requires greatly

compressing these videos in order to keep the number of input parameters manageable. However, this degrades the frame data to an extent that traditional image forensics methods which focus on microscopic analysis are rendered ineffective.

MesoNet models are said to be robust to such compression. In generating deepfakes, the extraction and then reintegration of faces can fail because some frames can end up with no facial reenactment, or with a large blurred area or double face contours. Autoencoders tend to poorly reconstruct fine details because of compression of input data and limited encoding space, resulting in blurry images generated. Autoencoders are restricted to such small encoding space because larger encoding space will result in the output face resembling input face and losing realism.

The use of Recurrent Neural Network (RNN)

Why use RNN? A video is not just a collection of unrelated frames but a continuous sequence of frames, with each frame having a temporal relationship with the previous frames. Since CNN works on each frame individually, it ignores the information encoded between multiple frames of the video, and cannot exploit the temporal relation between frames. Hence, extracting temporal characteristics in videos would be challenging for such neural networks that are designed for individual images, and we need a network more suitably designed for such sequential data, like RNN

Before we explore RNN, it must be said that in many cases, video classification would work just fine by analysing each frame in the video individually using CNN. In such cases, the classification of the video can be easily determined from just a snapshot of a single frame. For example, we could easily classify a video as a dance video or a video of a basketball game just from one frame only, as they all have distinct spatial features within one frame. Hence, for those applications, using only spatial features is sufficient for achieving high accuracy.

However, in the case of Deepfake videos, it is harder to label a video as real or fake based on just a single frame. Especially with the advancement of deepfake generation techniques such as GAN and autoencoders, the quality of Deepfake has improved and has become increasingly harder to detect. The use of these deep learning models has made swapped face images more challenging for models to detect as they are generally more realistic and of higher quality.

That being said, as mentioned in previous sections, they still leave traces such as temporal inconsistencies between frames. CNN cannot exploit this as they cannot accept a sequence of images without computationally intensive 3D convolution layers. Hence, there is a need for a network more suitable for sequential data to allow us to better understand and analyse each video frame, relative to the frames that came before it.

While one may expect a lot of weights to be in play, since different weights are applied to different parts of the input and we have a sequence of many images inputted to the RNN, however, due to parameter sharing, we are applying the same weights to each frame in the sequence. Hence, the number of weights are kept manageable.

How RNN works Recurrent neural networks are networks with loops in them, allowing information from previous inputs to persist. This chain-like nature is also another reason why recurrent neural networks are intimately related to sequences and should be the natural architecture to use for sequential data like videos.

Information is persisted in the network by passing the output of one training step to the input of the next training step, along with the new frames. Hence, by remembering information from past frames, it might be able to connect previous information to the present task, such as using previous video frames might better the understanding of the present frame.

Long Short Term Memory (LSTM) networks, which are a particular type of Recurrent Neural Network (RNN), further enhances this by deciding what we want to remember as well as forget about previous inputs via weighted gates, and thus can learn long-term dependencies in data sequences.

Employing a temporal-aware pipeline method that uses CNN with an additional long short term memory (LSTM) would allow us to exploit the advantages of both neural networks. CNN is employed to first extract the features from each frame, which are then fed into the LSTM (for temporal feature analysis) to create a temporal sequence descriptor for detecting inconsistencies between frames. A fully-connected network is used afterwards to classify doctored videos from real ones based on the sequence descriptor.

Support Vector Machine (SVM)

Apart from the neural nets algorithms mentioned above, the study conducted by Yang, Li, and Lyu (2019) also suggest the use of SVMs to classify deepfake and real videos. Based on the fact that many deepfake generation algorithms only change the central face regions instead of the whole face, the researchers experimentally confirmed that the estimated head pose of the subject constructed from the central face and that from the whole face are inconsistent. Following which, they used this difference in estimated headposes as a feature vector to train a SVM classifier to differentiate and identify deepfakes.

Unlike neural nets, SVM's strength is in its "whiteboxness" and high interpretability. By looking at post-trained parameters, we are able to further the analysis and validation of the results. Also, training and predicting processes of such models are significantly faster than neural nets.

However, the accuracy of the SVM model is relatively lower as compared to neural net models. In this regard, we will be focusing on the performance of neural network models in this study.

Enhancements — Transfer Learning and Pre-trained Weights

In our methods, we loaded our models with pretrained weights so as to reduce the time needed to train the model.

In addition, we used transfer learning for some of our models. This is because using appropriate pre-existing architecture can to a great extent reduce the parameters that need to be trained without necessarily sacrificing performance. Transfer learning would work in our case because

the task that the reused layers are trained on are quite similar, and maybe even overlapping. This is especially helpful in increasing the efficiency of models that would take longer to train, such as RNN. The CNN part in convolutional RNN structure usually has more complicated architecture than that of the RNN part. By using pre-trained CNN models like InceptionV3 in the CNN-RNN structure and therefore solely focusing on the weights in RNN, the training process is significantly sped up.

Implementation of Neural Networks

CNN

Implementation of Xception Net The xception model is trained on extracted frames from downloaded videos. The Xception architecture has 36 convolutional layers for feature extraction followed by fully-connected layer(s) for classification, which is replaced by a logistic regression layer with 2 outputs of 'real' or 'fake' for deepfake detection (Chollet 2017). The layers are initialized with the ImageNet weights.

To speed up training, we apply fine tune techniques to the original methods, which include using 2 training stages and a smaller learning rate for the whole model compared to scratched training. The first stage trains the custom top classifier only for 1 epoch and the second stage trains the whole model for 5 epochs. Using Adam as an optimizer, the learning rate for the top classifier is 1e-3, and for the whole model is 1e-4.

Implementation of Meso4 The features of Meso-4 model include the usage of ReLU activation to improve generalisation of the model by introducing non-linearity. Although a more robust approach to deepfake detection would involve an ensemble of neural networks with different model configurations. However, due to the nature of the problem and data, training and tuning multiple models would require too much time and resources. To that end, Meso-4 makes use of a technique known as Dropout which randomly drops out nodes in the architecture to prevent overfitting

Implementation of MesoInception MesoInception-4 makes use of a variant of the Inception module, which is a hallmark technique used for CNN models in recent years, which features a small number of hyper-parameters and low computation time. However, instead of the 5x5 convolutions used in standard Inception modules, dilated convolution was used in order to deal with multi-scale information. Both models made use of ADAM, a method for stochastic optimisation in order to optimise the weights of the model. It is a first-order gradient based optimisation which is computationally inexpensive to train with low memory requirements.

RNN

Implementation of Convolutional LSTM with Pre-trained CNN Based on the general idea of Güera and Delp (2018), we have used a pre-trained InceptionV3 model with weights extracted from ImageNet, which was combined with a 1024-node LSTM layer followed by a 512-node dense layer with 0.5 dropout rate. Softmax function was used as output activation function. The model was trained through Adam optimiser with learning rate 1e-5. The inputs were 40-frame videos with 299x299 image size. Images were sample-wise centered. Note that we strictly used equal amounts of real and fake videos for training to reduce bias.

Methodologies and Results

Dataset for Training and Testing

DeepFakeDetection (Rossler et al. 2019) dataset by FaceForensics++, which comprises 363 originally source actor videos and 3068 manipulated videos. This dataset claims to cover varied realistic scenarios by including videos that are manipulated and compressed to different extent. In this report, videos with compressed rate 23 (relatively high quality) are used for training and testing.

Computation Power[1]

NUS High Performance Computing (HPC) is deployed for the preprocessing and training of the FaceForensics++ datasets. It is capable of processing big data and performing complex tasks at much higher speed compared to local computers. By taking advantage of gpu-supported parallel processing across multiple computing nodes, it greatly improves the efficiency and reliability of our algorithms by enabling training on large-scale datasets.

Training and Testing Results

Xception Net Using original extracted frames from the video without face tracking, the training accuracy is 0.8379 and the testing accuracy is 0.84 training with a batch size of 16 and 1 epoch for the pre-trained stage and a batch size of 32 and 5 epochs for the fine-tuned stage. During the pre-trained stage which trains on the top classifier, the loss drops from 1.1085 to 0.4455. During the second stage to train the whole model, the loss drops to 0.4377, which shows reasonable convergence.

MesoNet With the help of an image generator, we managed to remove some distortion by rescaling, zooming, and changing the brightness of our input images. Our meso4 model utilised pretrained weights. We used 4 epochs with freezing of lower layers and slower learning rate to train our output layer while letting lower layers adjust to the type of inputs in our domain. We then ran a few epochs for fine-tuning which resulted in a model capable of correctly classifying 1042 out of 1241 videos correctly. This means an accuracy of approximately 0.8396.

Due to limited time to run the models, we did not manage to experiment to determine when our model would be overfitted. We also chose to omit the use of MesoInception4 as previous studies suggest that it had similar yet not much better performance (Afchar et al. 2018). Figure 1 and Figure 2 illustrate that for both models, at round 6 epochs, there would be a large improvement in performance on the training set. Hence there is a possibility that epochs around that

[1]Special thanks to Mr. Kuang Hao from NUS IT for providing instructions on the usage of HPC.

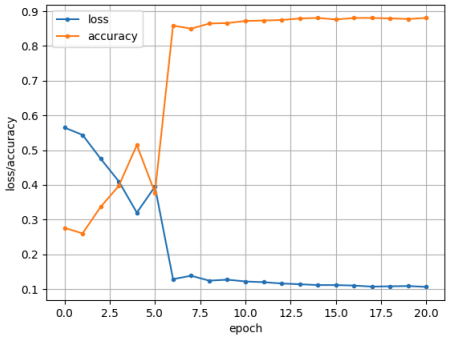

Figure 1: Plot of loss and accuracy of Meso4 against number of epochs

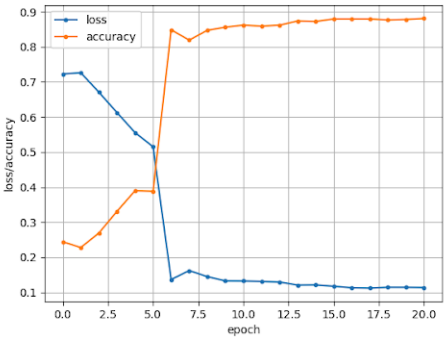

Figure 2: Plot of loss and accuracy of MesoInception4 against number of epoch

number would allow the models to predict optimally on the test set as well.

Convolutional LSTM Without specifically extracting the facial part of each video for the training, convolutional LSTM model failed to converge and generated final training accuracy of 0.49 in the training process with a batch size of 32 video and 3 epoches. As a result, the model generated a testing result of 0.53 accuracy, which is not different from random guessing at statistically significant level. This result was notably inconsistent with Güera and Delp (2018) where a similar convolutionalLSTM structure was used. The discussion on this discrepancy is included in the next section

Discussion and Conclusion

Future Improvements: Extracting faces from frames

To further improve the performance of our model, face tracking and extraction of face region with conservative cropping and enlargement can be performed during preprocessing. So that most of the unnecessary background will be removed as they can cause confusion to our models. It is crucial that we leave some margin so that the model can recognise the boundaries of faces.

Xception Net

The output of Xception Net shows a fairly good result, but can be improved if using frames extracted with face tracking and larger epochs (for example 5 epochs for the first stage and 50 epochs for the second stage).

MesoNet

Apart from extracting the faces from the frames, a possible improvement would be aligning the faces in the frames using a facial landmark detection module such as dlib which detects the key facial structures in the ROI (Regions of interest).

Also, as we are using extracting frames from videos, extra care could have been taken towards ensuring that there is a balanced distribution of different faces (i.e. not having all the repeated frames of the same face under the same camera angle and illumination environment).

These suggestions would greatly improve the quality of the dataset, and by ensuring that the dataset is well balanced, we are able to avoid possible biases that may arise from training the model with our training data.

Convolutional LSTM

The algorithm does not give a very satisfactory result, which may be due to the following reasons. First, the frames extracted are not high-resolution enough due to resizing and compressing for RNN to detect temporal differences between frames. Second, some videos have several scenes while the CNN-RNN structure requires consistent motions. In addition, the model might be too complex to converge through a relatively small number of epochs of training on the dataset. Also, it is probable that features extracted by pre-trained CNN are not significant enough for our detection.

To improve the performance of convolutionalLSTM, there are several possible ways, including performing face extraction on extracted frames, using more epochs, and choosing more delicately selected pre-trained models.

Roles and Remarks

The codes for this project can be found here.

https://github.com/MackyMaguire/CS3244_Project

We divided into three subgroups to implement 3 different models to detect deepfake videos. Chenchen and Xinran took charge of Xception net model; Tianyu and Linhang wrote codes for convolutional LSTM structure while MesoNet part was done by Ze yuan and Ruicong. Chenchen and Xinran also helped run the models on hpc provided by NUS IT and contact professionals for the technological issues we encountered. The parts regarding the results and implementations of each model in this report were written by the team in charge of each model, while the rest parts were mainly organised and edited by Tianyu, Ruicong and Ze yuan. Converting the report into AAAI required format was done by Linhang using open-source LaTex editor Overleaf.

References

Afchar, D.; Nozick, V.; Yamagishi, J.; and Echizen, I. 2018. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–7. IEEE.

Banks, A. 2018. Op-Ed: Deepfakes and why the future of porn is terrifying. https://www.highsnobiety.com/p/what-are-deepfakes-ai-porn/.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251–1258.

Christian, J. 2018. Experts fear face swapping tech could start an international showdown. https://www.highsnobiety.com/p/what-are-deepfakes-ai-porn/.

Güera, D., and Delp, E. J. 2018. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 1–6. IEEE.

Ikeda, S. 2019. The cutting edge of AI cyber attacks: Deepfake audio used to impersonate senior executives. https://www.cpomagazine.com/cyber-security/the-cutting-edge-of-ai-cyber-attacks-deepfake-audio-used-to-impersonate-senior-executives/.

Metz, C. 2019. Internet companies prepare to fight the 'deepfake' future. https://www.channelnewsasia.com/news/cnainsider/deepfakes-deepnude-porn-videos-threat-to-national-security-12183798.

Ng, D., and Tang, H. H. 2019. Deepfakes, nudes and the threat to national security. https://www.channelnewsasia.com/news/cnainsider/deepfakes-deepnude-porn-videos-threat-to-national-security-12183798.

Romano, A. 2018. Why Reddit's face-swapping celebrity porn craze is a harbinger of dystopia. https://www.vox.com/2018/1/31/16932264/reddit-celebrity-porn-face-swapping-dystopia.

Rossler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; and Nießner, M. 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE International Conference on Computer Vision*, 1–11.

Sample, I. 2020. What are deepfakes – and how can you spot them? https://www.theguardian.com/technology/2020/jan/13/what-are-deepfakes-and-how-can-you-spot-them.

Schwartz, O. 2018. You thought fake news was bad? Deep fakes are where truth goes to die. https://www.theguardian.com/technology/2018/nov/12/deep-fakes-fake-news-truth.

Wikipedia. 2020. Deepfake — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Deepfake&oldid=953485354.

Xu, A. 2019. AI, truth, and society: deepfakes at the front of the technological cold war. https://medium.com/gradientcrescent/ai-truth-and-society-deepfakes-at-the-front-of-the-technological-cold-war-86c3b5103ce6.

Yang, X.; Li, Y.; and Lyu, S. 2019. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8261–8265. IEEE.