Deepfake detection model based on fake attributes shown in images/videos

Ran Heo[1], Eunjin Cho[2]
[1]Dept. of Chemical & Biomolecular Engineering, Dongyang Mirae University,
hran9462@gmail.com
[2]Dept. of Mechanical Engineering, RMIT University, jintazcho@gmail.com

Abstract

Deepfake refers to the fake images/videos created with deep learning. Deepfake video caused social problems as it has spread through the internet, including Social Network Service(SNS). [4,5,6] Therefore, the importance of deepfake detection models has emerged, so many detection models have been suggested. However, these models are not useful in the real world because they can only detect a few parts of deepfake creation algorithms. (Figure 2) In this paper, we suggest the model that uses CNN(Convolutional Neural Network) for fake attribute detection and RNN(Recurrent Neural Network) for fake/real judgment to cope with various deepfake creation algorithms real-world. Our purpose is to prove that the suggested model is useful in the real-world. Our experiment focused on the improved model's detection performance to detect an increasing number of various deepfake creation models. Researchers' participation is required to cope with multiple deepfake creation algorithms as well. This paper trained deepfake detection models with the presented architecture to prove that the detection performance is similar to compare with the models with high-end GPU[37, 38]. The evaluation result was that Recall increased from 2% to 75% after the detection model improvement, the FPR rate decreased 1.6% to 0.1 and AUC increased from 0.02 to 0.77, which is similar to the model with high-end GPU[37, 38]. [1)]

Keywords - Deepfake Detection, Face synthesis Detection, Convolutional Neural Network, Recurrent Neural Network

1. Introduction

Fake face generation algorithms are becoming more complicated and diversified since late 2017 when Reddit user is called 'Deepfakes' proposed face generation method with deep learning. The most representative fake face generation methods are Face synthesis, which generates a whole human head, including face and Face-swap, which swaps faces and facial expressions of two different people. There are deepfake videos of politicians like Obama, Putin, Hillary, and celebrities' face-synthesized pornos created with the Face-swap method. [1, 2, 3] Such videos may cause social problems such as fake news creation, defamation[4, 5, 6], so deep fake related videos legislation[7, 8, 9] and fake face detection have become critical issues to solve these problems.

Deepfake creation algorithm	Method
Began[11]	Face synthesis
CausalGAN[12]	Face synthesis
faceswap/deepfake[13]	Face swap
StarGAN[14]	Face synthesis
Enrique Sanchez and Michel Valstar[15]	Face synthesis
MWGAN[16]	Face synthesis
ALAE[17]	Face synthesis
StyleGAN[18]	Face synthesis
MSG-GAN[19]	Face synthesis
FQGAN[20]	Face synthesis
ProGAN[21]	Face synthesis
StyleGAN v2[22]	Face synthesis
COCO-GAN[23]	Face synthesis
VAEGAN[24]	Face synthesis
HoloGAN[25]	Face synthesis
SPA-GAN[26]	Face synthesis
FTGAN[27]	Face synthesis
SEGAN[28]	Face synthesis
StarGAN V2[29]	Face synthesis
LSGAN[30]	Face synthesis
DCGAN[31]	Face synthesis
WGAN[32]	Face synthesis
GAN2play[33]	Face synthesis
Glow[34]	Face synthesis
GANnotation[35]	Face synthesis
deferred neural rendering[36]	Face synthesis
neural texture[36]	Face synthesis

Table 1. Various deepfake creation algorithms

1) github (Entire model code, instruction, Process Video are available): https://github.com/teamnova-ailab/Deepfake-detection-model-based-on-fake-attributes-shown-in-image-video/

Current detection models stated in Table 2, shows a high detection rate of specific fake images created by the algorithm, which created a training image. However, such detection models specialized in just one algorithm, so they can not guarantee the detection performance of fake faces created by different algorithms.[10] As Fake face creation algorithms are various continuously, existing detection models cannot detect fake faces in the real world. Therefore detection model which guarantee detection accuracy of new fake face creation algorithms is required.

We defined images from figure 1 and awkward images due to the flaws of the fake face creation process as fake attributes presented images. In this paper, the detection model comprises of CNN models that detect each fake attribute and an RNN model which judge Real/Fake in the basis of those attributes. This model is useful when a new fake face creation method appears because it is possible to add a new CNN model that trains a new fake feature. Model structure and code are available in GitHub so researchers can contribute this model to improve detection accuracy.

The proposed detection model is enough to train with Nvidia RTX 2070 SUPER when other detection models used Nvidia Titan X GPU[37], Nvidia

(a)

(b)

(c)

Figure 1. Example of Image with fake features (a) Face blur Images not showing eyes, nose and mouth (b) Face noise showing Images (c) Images Glasses without glass legs

Detection model	Data set used
McCloskey and Albright (2018) [39]	NIST MFC2018
Yu et al. (2019) [40]	Own (ProGAN, SNGAN, CramerGAN, MMDGAN)
Wang et al. (2019) [38]	FF++, DFDC, Own (PGGAN, StyleGAN2, StarGAN, STGAN, StyleGAN, STGAN)
Stehouwer et al. (2019) [41]	DFFD (ProGAN, StyleGAN)
Nataraj et al. (2019) [42]	100K-Faces (StyleGAN)
Neves et al. (2019) [43]	100K-Faces (StyleGAN) FSRemovalDB (StyleGAN)
Marra et al. (2019) [44]	Own (CycleGAN, ProGAN, Glow, StarGAN, StyleGAN)
Zhou et al. (2018) [45]	Own
Afchar et al. (2018) [46]	Own
Güera and Delp (2018) [47]	Own
Yang et al. (2019) [48]	UADFV
Li et al. (2019) [49]	UADFV DeepfakeTIMIT
Rössler et al. (2019) [50]	FF++
Matern et al. (2019) [51]	Own
Nguyen et al. (2019) [52]	FF++
Agarwal and Farid (2019) [53]	Own (FaceSwap, HQ)
Sabir et al. (2019) [54]	FF++
Bharati et al. (2016) [55]	Own (Celebrity Retouching, ND-IIITD Retouching)
Tariq et al. (2018) [56]	Own (ProGAN, Adobe Photoshop)
Wang et al. (2019) [57]	Own (InterFaceGAN/StyleGAN)
Jain et al. (2019) [58]	Own (ND-IIITD Retouching, StarGAN)
Marra et al. (2019) [59]	Own (Glow/StarGAN)
Zhang et al. (2019) [60]	Own (StarGAN/CycleGAN)
Amerini et al. (2019) [61]	FF++

Table 2. Detection model and Dataset used in Detection model

Tesla P40 GPU[38]. It is because of training with filter applied images that are suitable for each fak e attribute, so unnecessary attribute extraction is reduced, and necessary attribute extraction becom es easy. Therefore it is possible to train with 100, 200 images, which are relatively small amounts of images, and possibly build models despite the har dware not being high-end. This process disentang les the restriction which occurs during the deepfa ke detection study and paves the way for more re searchers to develop the model.

2. Related Works

2.1. Fake face generation method

2.1.1. Face synthesis

Face synthesis is a GAN(Generative Adversarial Netw orks) based method to generate a whole face, including hair. GAN consists of Generator, which generates imag es and Discriminator, which judges Real/Fake. A gener ator consists of Encoder and Decoder, where Encoder learns image attributes and decoder reconstructs image based on those attributes. Specifically, there are two face datasets called A and B. Encoder trains facial features in each dataset, then A decoder creates A face, and B decoder creates B face. (Figure 2) Studies that focus on maintaining hairstyle, hair color, eye color, mustache, and facial expressions[63, 64, 65] are actively processed these days.

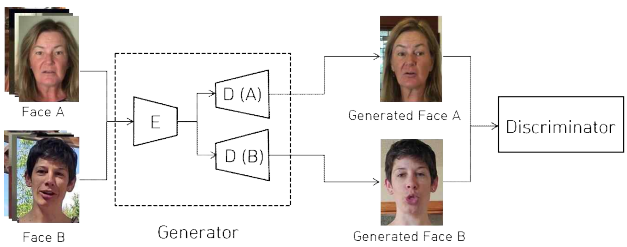


Figure 2. face synthesis algorithm E : Encoder , D (A) : A Decoder, D (B) : B Decoder

2.1.2. face swap

Face swap is a face generation method that swaps a Source face to a Target face in original images/videos. GAN generates faces to convert to Target Face during this process. In the first step, it detects and aligns the Source face in original images/videos, which is an input value of Encoder. The decoder generates Targe t's Face based on the attributes extracted from Encoder. The generated face is adjusted and inserted into the original image, then smooth the boundary of it. (Figure 3) We can implement this method to both images/Video s and generate faces using various GAN algorithms in Figure 3 (c). However, it may display face color mismatch or resolution mismatch and unnatural synthe sized boundary part between the generated face and original image. (Figure 4)

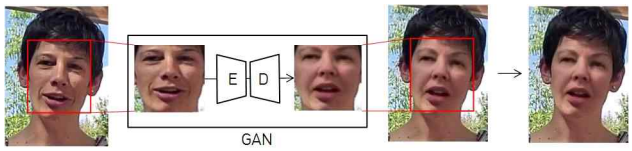
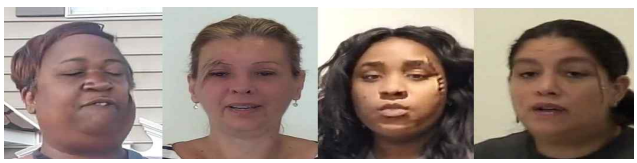


Figure 3. face swap algorithm (a) Face detection (b) Face Crop and Face Align (c) GAN (d) Wraping face (e) Boundary smoothing

(a)



(b)

(c)

Figure 4. Fake features shown in face swap (a) Face color mismatch between original image and generated face (b) Unnatural synthesized boundary (c) resolution mismatch and unnatural synthesized boundary part between generated face and original image

2.2. Detection Model

2.2.1. Fake face Detection model with similar archit ecture to this paper

Figure 2 states various fake face detection model s. Li et al.[66], Güera and Delp[47], Sabir et al. [4 8] use CNN and RNN to attempt fake face detecti

on in fake videos. Li et al.[66] attempts fake vide o detection with eye blink interval based on the f act that fake video has less number of eye blink that real video. To do this, CNN extracts features from eye images then RNN checks eye blink inter val.

Güera and Delp[47] attempt to detect fake videos based on attributes shown in videos. CNN extract s features per frame, then RNN detects fake vide os in time flow to train video features.

Sabir et al.[48] has a similar architecture to Gü era and Delp[47], but the difference is that it adds Face crop and Face align preprocess to make CN N easy to extract features. Those three detection models incorporate the CNN models and the RNN models to utilize characteristics in videos to detect unnatural part in temporal flow.

2.2.2. Face detection model with better performanc e GPU than proposed method

Dang et al.[37] intends to build a model which detects fake face well even if data imbalance problem occurs because there are more real images/videos than fake images/videos. It proposes HF-MANFA, which incorp orates MANFA and XGBoost, then attempts to prove the performance through the experiment.

Wang et al.[38] offers a model that can detect w ell in four types of deformation(noise, blur, compr ession, resize) in fake images and real images. It proposes a fake face detection model that recogniz es neuron activity during learning, and it perform s experiments on how well detect those four type s of deformations. It also confirms how much the proposed model can identify new data that does n ot use in learning.

2.2.1. Techniques used in Detection model

2.2.1.1. CNN (Convolutional Neural Network)

We use CNN detector to capture fake attributes, a combination of SSD[67] and Faster-RCNN[68]. Faster -RCNN is 2 Stage-Detector, and it detects fake charact eristics in 2 stages. In the first stage, it uses RPN(Regio n Proposal Network) to draw bounding boxes in an area with fake features while image comes as an input value. The second stage checks if there is any fake feature in bounding boxes through ROI pooling. Faster-RCNN[68] is accurate because it goes the Convolution process twice but has a disadvantage of a slow process.

SSD[67] is a 1 Stage-Detector that detects fake features in only one stage. It confirms if there is any fake feature in pre-sized Default boxes durin g each Convolution process while the input image comes in. It has an advantage of speed because t here is only one convolution process, but the ima ge with less Convolution process has low-dimensi onal features(straight-line, curve, do), so detection rate possibly decreases in images that require hig h-dimensional features(eyes, nose, mouth, face).

Considering such attributes, we use the SSD mo del[67] for features discovered in eyes, nose, mout h then uses the Faster-RCNN model[68] for only a few features found in the face that results in d etection model speed increase. It is because the fe atures discovered in eyes, nose, mouth must detec t fake features again after eyes, nose, mouth dete ction. Where there are features found in the face, we used SSD on low-dimensional features and Fa ster-RCNN on high-dimensional features.

2.2.1.2. RNN (Recurrent Neural Network)

We use BERT(Bidirectional Encoder Representations from Transformers)[69] for the Real/Fake judgment RNN model and BERT's fake and real image judgment-making process. It judges Real/Fake based on the relati onship in fake features through Self-Attention(1), (2) of Transformer[70], and Feed-Forward Network(3).

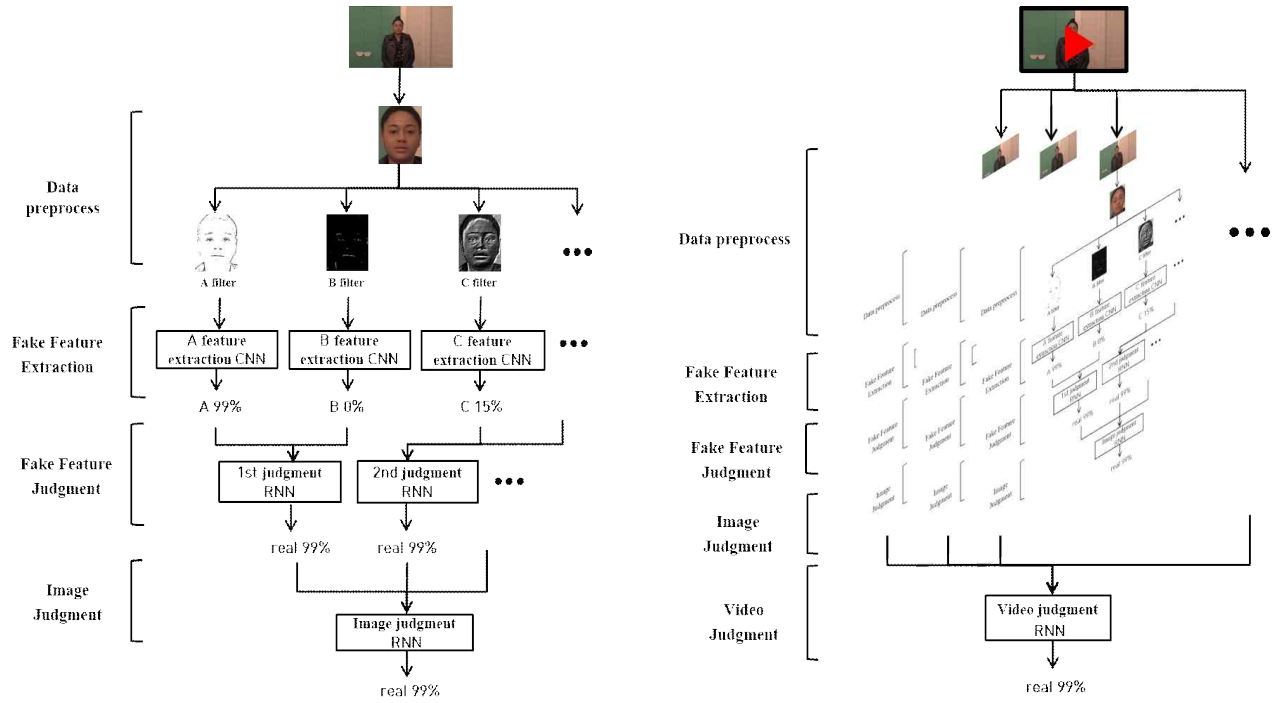
$$Attention(Q,\ K,\ V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

$$MultiHead(Q,\ K,\ V\ ) = Concat(head_1, \cdots, head_h)W^O \quad (2)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V\ )$$

$$FFN(x) = max(0,\ xW_1\ + b_1\ )W_2\ + b_2 \quad (3)$$

Explicitly, the fake features detected by CNN Detector are computed 'i' times Scaled Dot-Product Attention(1) and determined which features are related. It merges the result of 'i' times computation and multiplies the weight matrix then consolidate (Multi-Head Attention



(a) (b)

Figure 5. Whole detection model architecture (a) Image detection process (b) Video detection process

(2)) results. We use the Multi-Head Attention result value of each feature as Position-Wise Feed-Forward Network(3) input value to figure out how fake features are related.

3. Methods

This paper suggests the detection model consists of five stages: preprocess, detection of fake features, fake features determination, image, and video judgment. The overall model architecture presented in Figure 5, and the model architecture briefly explained step by step.

3.1. Data preprocess

We use face detection to crop facial part when images/videos come in. The most optimized filters for each feature applied in the face images, and then such images are moved to the fake feature detection step. Researchers observe and classify fake feature data to find the most optimized filter combination then find the most suitable filter combination for each feature. The filter combination consists of one filter to multiple filters. When all the real images and fake images are in a sortable state, then nominate the filter as suitable.

3.2. Fake Feature Extraction

We extract fake features discovered in images/videos with the CNN model in this stage. Then we classify the data in each fake feature and construct CNN models that detect such features. Images that applied filter for each feature are used as the input value for the CNN model to classify through whether the feature exists or not. Considering process speed, each model that detects fake features processed in parallel, and the results returns. Various generate methods, and fake features are various in each generation method, so real-time detection is difficult.

3.3. Fake Feature Judgment

The RNN model judges Real/Fake based on the fake feature information which is previously detected. Fake feature information is bounded to related one and used as RNN model input value. This process is essential because there is a single fake feature that judges Real/Fake and a combination of multiple fake features that judges Real/Fake.

3.4. Image Judgment

Image judgment uses 3.3(Fake Feature Judgment) result as the input value. Fake feature judgments divide into multiple cases, so the entire decision can be made when image judgment with RNN. When images come in as input value, image judgment finishes. When the input value is a video, iterates as many times as the number of video frames times the process of 3.1(Data Preprocess) to 3.4(Image Judge) and video judgment starts.

3.5. Video Judgment

Video judgment uses the result value of the selected number of frame images. Video is a set of multiple images, so each image's information is gathered to judge the Real/Fake of video.

4. Experiments

4.1. Experimental Environment

4.1.1. Dataset

The Models in Table 3 generates data to construct the detection model training data. Since there are fake images created with various real-world models, more fake feature detection makes a more useful detection model available in reality. Therefore we set detection model training data to include various data in more generate algorithms. We excluded less than 100 generated data or less than 100 numbers of fake features and created CNN Detector using 17 of the total 28 data generated by the algorithms presented in Table 3.

For a straightforward approach and prevent biased results, we use public data called generated photo[71] and celeb-A[72] as fake face data set and real face data..

4.1.2. Evaluation

We used FPR(False positive rate), Recall, AUC(Area Under Curve) for evaluation. FPR, the index of the detection probability where it detects real data as fake data. The Recall is an index of the likelihood of detecting fake data as fake. AUC is an index of the ROC curve(Receiver Operating Characteristic curve) area that its x-axis indicates FPR(1), and the y-axis indicates Recall (2). The model's FPR, Recall represented in the coordinate plane by Threshold, is called the ROC curve. AUC is the calculation of the base area of the graph to compare this graph with other models quantitatively.

$$FPR = \frac{FP}{FP + TN} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

The critical part of deepfake detection is to detect fake and real faces, respectively, accurately. Therefore, a high Recall (2) model indicates the probability where it detects the fake data to be fake and a low FPR (1), indicating the probability that real data predicted to be fake is an excellent detection model. Besides, if AUC is closer to 1, it is an excellent detection model because AUC is the ROC curve area, and the ROC curve comprehensively represents by the Threshold.

4.1.3. Training and Test Environment

Training and test environment is stated below.

OS : Ubuntu 18.04
CPU : AMD Ryzen 5 2400G
GPU : Nvidia RTX 2070 SUPER
RAM : DDR4 32GB

4.2. Test for detection rate improvement when fake features are added.

In reality, there are images created with various generate algorithms, and new generate algorithms continuously spread. On the other hand, detection models have lower detection rates when features that have not been learning appear. Therefore, the detection rate should be guaranteed when the detection model improved to prepare for images of new features. To this end, we tested whether the detection rate increased when CNN Detector added to the detection model. The test methods are as follows.

a. Conduct detection of test sets with existing detection models learned from data provided by Facebook [73].
b. Proceed CNN Detector training with the data presented in Table 1 and supplement the model to conduct the test again.

c. Compare the detection rate before and after adding fake features and prove that the proposed model's detection rate increases.

Deepfake creation algorithms	Total number of data	Number of data used to train
Began[11]	512 Images	512
CausalGAN[12]	300 Images	140
faceswap/deepfake[13]	1,576 Images	1,576
StarGAN[14]	2,240 Images	1,747
Enrique Sanchez and Michel Valstar[15]	9,216 images	8,694
MWGAN[16]	1,281 Images	500
ALAE[17]	184 Images	184
StyleGAN[18]	1,000 Images	1,000
MSG-GAN[19]	1,000 Images	1,000
FQGAN[20]	1,000 Images	581
ProGAN[21]	1,000 Images	501
StyleGAN v2[22]	1,000 Images	579
COCO-GAN[23]	1,024 Images	941
VAEGAN[24]	700 Images	637
HoloGAN[25]	640 Images	-
SPA-GAN[26]	401 Images	-
FTGAN[27]	310 Images	-
SEGAN[28]	300 Images	-
StarGAN V2[29]	362 Images	-
LSGAN[30]	100 Images	-
DCGAN[31]	100 Images	-
WGAN[32]	100 Images	-
GAN2play[33]	64 Images	
Glow[34]	3 Images	-
GANnotation[35]	1 Video	-
deferred neural rendering[36]	Webcam	-
neural texture[36]	Webcam	-

Table 3. Total number of data created by deepfake algorithms and data with fake features

Figure 6 states the test result of the trained model with the Facebook dataset and Facebook dataset and dataset presented in Table 3. Recall, which indicates whether fake face data was detected well, was low at 2%, and FPR, which suggests whether real face data was detected incorrectly, was low at 1.6%. AUC, which calculates the base area of the ROC curve that shows Recall and FPR of the detection model according to the Threshold, is low at 0.002. However, Recall increased to 75%, FPR decreased to 0.1%, AUC increased to 0.77 after the model supplement by adding a CNN detector with data stated in Table 3.

The detection rate increases when there are more fake features that can be detected based on test results. The Facebook model can identify 14 fake features, but the model that added CNN Detector as the data in Table 3 can discover 35 counterfeit features. Therefore it proves that detection rate is guaranteed when the CNN detector increases, identifying the fake features.

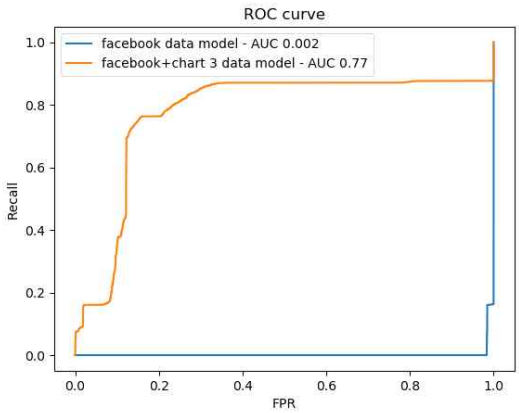


Figure 6. Detection model test result

4.3. Comparison to the model used high-end GPU

We aim to prove that the detection model, which does not use high-end hardware, can perform a high detection rate by comparing them with high-end GPUs. Various researchers need to participate In detecting various fake images in reality, as there are limitations in research if detection models can be learned only with high-end GPUs. Therefore, we aim to ease the constraints of deepfake detection studies and allow more researchers to participate in fake image detection studies.

In the experiment, we compare models using GPUs that performed better than the RTX 2070 GPUs used for model learning [37, 38]. Each model for comparison, as outlined in Table 4, does not experiment using a unified dataset. Moreover, the code is not public, so the comparison is conducted based on AUC, the performance evaluation index listed in the paper. (Table 4)

In the comparison result of the AUC evaluation index of our proposed detection to the detection models in [37] and [38], our proposed model performs approximately 0.13 lower. (Table 5) However, as 4.2 states that

the detection model performs better as it adds more fake features than [37], [38].

Therefore, the additional CNN Detector can lead to higher detection rates than detection models using high-end GPUs and requires several researchers to detect various images that appear in real life.

Detection models	GPU	Dataset
Dang et al. [11]	Titan X 12GB GPU	- MANFA dataset - SwapMe and FaceSwap dataset
Wang et al. [12]	Tesla P40 GPU	REAL - CelebA - Flicker-FacesHQ(FFHQ) - FaceForensics++ - DFDC - Celeb-DF FAKE - PGGAN - StyleGAN2 - StarGAN - STGAN - StyleGAN - FF++ - DFDC

Table 4. GPUs and datasets used by detection models to compare

Detection models	AUC
Dang et al. [11]	0.93
Wang et al. [12]	0.906
Proposed model in this paper	0.77

Table 5. Comparison of AUC values between models using high-end GPUs and the model presented in the paper

5. Conclusion

Efforts have made to prevent the abuse of deepfake continuously From 2007 to the present. In this paper, the detection model can be used in practice by a continuous supplement in the detection model consistently, CNN models learn fake features and the detection model structure that final judgment made with RNN models proposed. Test results for the detection model showed that Recall achieved 85%, FPR 0.1%, and AUC 0.77.

The proposed model mainly focuses on fake features, so there is a limitation when there is a new deepfake creation algorithm. To create a detection model with a high detection rate in new generation algorithms requires the continuous addition of CNN Detector with the participation of researchers.

It is also essential to store and share data, while RNN integrated training requires previously trained data. If data is not shared, there is a limit to the integrated model creation; otherwise, it creates a non-shared model. To overcome this limitation, we aim to enable data sharing around Github, which has released the detection model code to share data among researchers. We attach the Google drive address to GitHub to allow researchers to download data and upload new data through requests easily.

github (Entire model code, instruction, Process Video are available): https://github.com/teamnova-ailab/Deepfake-detection-model-based-on-fake-attributes-shown-in-image-video/

References

[1] https://www.youtube.com/watch?v=cQ54GDm1eL0 , Accessed : 2020-07-07

[2] https://www.youtube.com/watch?v=RWZmLKw7PG8 , Accessed : 2020-07-07

[3] https://www.youtube.com/watch?v=hKxFqxCaQcM , Accessed : 2020-07-07

[4] Deepfakes porn has serious consequences, https://www.bbc.com/news/technology-42938529 , Accessed : 2020-07-07

[5] How deepfakes undermine truth and threaten democracy, https://www.youtube.com/watch?v=pg5WtBjox-Y , Accessed : 2020-07-07

[6] Fake videos could be the next big problem in the 2020 elections, https://www.cnbc.com/2019/10/15/deepfakes-could-be-problem-for-the-2020-election.html , Accessed : 2020-07-07

[7] https://www.congress.gov/bill/376th-congress/senate-bill/2065/text , Accessed : 2020-07-07

[8] https://www.congress.gov/bill/376th-congress/house-bill/3230/text , Accessed : 2020-07-07

[9] http://www.moj.go.kr/bbs/moj/182/521437/artclView.do , Accessed : 2020-07-07

[10] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, Javier Ortega-Garcia, "Deep

Fakes and Beyond: A Survey of Face Manipulation and Fake Detection", arXiv:2001.00179, 2020, p. 4
[11] David Berthelot, Thomas Schumm, Luke Metz, "Began: Boundary equilibrium generative adversarial networks", arXiv preprint arXiv:1703.10717, 2017
[12] Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, Sriram Vishwanath, "CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training", arXiv preprint arXiv:1709.02023, 2017
[13] deepfake/faceswap, https://github.com/deepfakes/faceswap , Accessed : 2020-07-07
[14] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", arXiv preprintarXiv:1711.
09020v3, 2018
[15] Enrique Sanchez, Michel Valstar, "Triple consistency loss for pairing distributions in GAN-based face synthesis", arXiv preprint arXiv:1811.03492v1, 2018
[16] Jiezhang Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, Mingkui Tan, "Multi-marginal Wasserstein GAN", arXiv preprint arXiv:1911.00888v1 , 2019
[17] Stanislav Pidhorskyi, Donald Adjeroh, Gianfranco Doretto, Adversarial Latent Autoencoders, arXiv preprint arXiv:2004.04467, 2020
[18] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks”, in Proc. Conference on Computer Vision and Pattern Recognition, 2019.
[19] Animesh Karnewar, Oliver Wang, "MSG-GAN: Multi-Scale Gradient GAN for STable Image Synthesis", arXiv preprint arXiv:1903.06048v3, 2019
[20] Yang Zhao, Chunyuan Li, Ping Yu, Jianfeng Gao, Changyou Chen, "Feature Quantization Improves GAN Training", arXiv preprint arXiv:2004.02088v1, 2020
[21] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation", arXiv preprint arXiv:1710.101
96v3, 2018
[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila, "Analyzing and Improving the Image Quality of StyleGAN", arXiv preprint arXiv:1912.04958v2, 2020
[23] Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, Hwann-Tzong Chen, "COCO-GAN: Generation by Parts via Conditional Coordinating", arXiv preprint arXiv:1904.00284v4 , 2020
[24] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, Ole Winther, "Autoencoding beyond pixels using a learned similarity metric", arXiv preprint arXiv:1512.09300, 2016
[25] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, Yong-Liang Yang, "HoloGAN: Unsupervised learning of 3D representations from natural images", arXiv preprint arXiv:1904.01326v2, 2019
[26] Hajar Emami, Majid Moradi Aliabadi, Ming Dong, Ratna Babu Chinnam, "SPA-GAN: Spatial Attention GAN for Image-to-Image Translation", arXiv preprint arXiv:1908.06616 , 2020
[27] Xiang Chen, Lingbo Qing, Xiaohai He, Xiaodong Luo, Yining Xu, "FTGAN: A Fully-trained Generative Adversarial Networks for Text to Face Generation", arXiv preprint arXiv:1904.05729, 2020
[28] Santiago Pascual, Antonio Bonafonte, Joan Serrà, "SEGAN: Speech Enhancement Generative Adversarial Network", arXiv preprint arXiv:1703.0945
2, 2017
[29] Yunjey Choi, Youngjung Uh, Jaejun Yoo, Jung-Woo Ha, "StarGAN v2: Diverse Image Synthesis for Multiple Domains", arXiv preprint arXiv:1912.018
65v2, 2020
[30] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, Stephen Paul Smolley, "Least Squares Generative Adversarial Networks", arXiv preprint arXiv:1611.04076, 2017
[31] Alec Radford, Luke Metz, Soumith Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", arXiv preprint arXiv:1511.06434, 2016
[32] Martin Arjovsky, Soumith Chintala, Léon Bottou, "Wasserstein GAN", arXiv preprint arXiv:1701.07875, 2017
[33] GAN2Play, https://github.com/JimmyYing/GAN2Play , Accessed : 2020-07-07
[34] Diederik P. Kingma, Prafulla Dhariwal, "Glow:

Generative Flow with Invertible 1x1 Convolutions", arXiv preprint arXiv:1807.03039v2, 2018
[35] GANnotation, https://github.com/ESanchezLozano/GANnotation , Accessed : 2020-07-07
[36] Justus Thies, Michael Zollhöfer, Matthias Nießner, "Deferred Neural Rendering: Image Synthesis using Neural Textures", arXiv preprint arXiv:1904.12356v1, 2019
[37] L. Minh Dang, Syed Ibrahim Hassan, Suhyeon Im, Hyeonjoon Moon, "Face image manipulation detection based on a convolutional neural network.", Expert Systems with Applications 389, 156 - 168, 2019., pp. 159,160,166
[38] Run Wang, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yihao Huang, Jian Wang, Yang Liu, "FakeSpotter: A Simple yet Robust Baseline for Spotting AI-Synthesized Fake Faces", arXiv preprint arXiv:1909.06382v2, 2020, pp. 3,4,5
[39] S. McCloskey and M. Albright, "Detecting GAN-Generated Imagery Using Color Cues", arXiv preprint arXiv:1812.08247, 2018.
[40] N. Yu, L. Davis, and M. Fritz, "Attributing Fake Images to GANs: Analyzing Fingerprints in Generated Images", in Proc. International Conference on Computer Vision, 2019.
[41] J. Stehouwer, H. Dang, F. Liu, X. Liu, and A. Jain, "On the Detection of Digital Face Manipulation", arXiv preprint arXiv:1910.01717, 2019.
[42] L. Nataraj, T. Mohammed, B. Manjunath, S. Chandrasekaran, A. Flenner,J. Bappy, and A. Roy-Chowdhury, "Detecting GAN Generated Fake Images Using Co-Occurrence Matrices," arXiv preprint arXiv:1903.06836, 2019.
[43] J. Neves, R. Tolosana, R. Vera-Rodriguez, V. Lopes, and H. Proenc¸a, "Real or Fake? Spoofing State-Of-The-Art Face Synthesis Detection Systems", arXiv preprint arXiv:1911.05351, 2019
[44] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, "Incremental Learning for the Detection and Classification of GAN-Generated Images", in Proc. International Workshop on Information Forensics and Security, 2019
[45] P. Zhou, X. Han, V. Morariu, and L. Davis, "Two-Stream Neural Networks for Tampered Face Detection", in Proc. Conference on Computer Vision and Pattern Recognition Workshops, 2017.
[46] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: a Compact Facial Video Forgery Detection Network," in Proc. International Workshop on Information Forensics and Security, 2018.
[47] D. Güera and E. Delp, "Deepfake Video Detection Using Recurrent Neural Networks", in Proc. International Conference on Advanced Video and Signal Based Surveillance, 2018.
[48] X. Yang, Y. Li, and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses," in Proc. International Conference on Acoustics, Speech and Signal Processing, 2019.
[49] Y. Li and S. Lyu, "Exposing DeepFake Videos By Detecting Face Warping Artifacts," in Proc. Conference on Computer Vision and Pattern Recognition Workshops, 2019
[50] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images", in Proc. International Conference on Computer Vision, 2019.
[51] F. Matern, C. Riess, and M. Stamminger, "Exploiting Visual Artifacts to Expose DeepFakes and Face Manipulations", in Proc. IEEE Winter Applications of Computer Vision Workshops, 2019.
[52] H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task Learning For Detecting and Segmenting Manipulated Facial Images and Videos", arXiv preprint arXiv:1906.06876, 2019.
[53] S. Agarwal and H. Farid, "Protecting World Leaders Against Deep Fakes", in Proc. Conference on Computer Vision and Pattern Recognition Workshops, 2019.
[54] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, Prem Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos", arXiv:1905.00582v3 , 2019
[55] A. Bharati, R. Singh, M. Vatsa, and K. Bowyer, "Detecting Facial Retouching Using Supervised Deep Learning", IEEE Transactions on Information Forensics and Security, vol. 11, no. 9, pp. 1903 - 1913, 2016.
[56] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. Woo, "Detecting Both Machine and Human Created Fake Face Images in the Wild," in Proc. International Worksh

op on Multimedia Privacy and Security, 2018, pp. 81 - 87
[57] S. Wang, O. Wang, A. Owens, R. Zhang, and A. Efros, “Detecting Photoshopped Faces by Scripting Photoshop,” arXiv preprint arXiv:1906.05856, 2019.
[58] A. Jain, R. Singh, and M. Vatsa, “On Detecting GANs and Retouching based Synthetic Alterations”, in Proc. International Conference on Biometrics Theory, Applications and Systems, 2018.
[59] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, “Incremental Learning for the Detection and Classification of GAN-Generated Images,” in Proc. International Workshop on Information Forensics and Security, 2019.
[60] X. Zhang, S. Karaman, and S. Chang, “Detecting and Simulating Artifacts in GAN Fake Images”, arXiv preprint arXiv:1907.06515, 2019.
[61] I. Amerini, L. Galteri, R. Caldelli, and A. Bimbo, “Deepfake Video Detection through Optical Flow based CNN”, in Proc. International Conference on Computer Vision, 2019.
[62] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Networks", arXiv preprint arXiv:1406.2661, 2014
[63] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks”, in Proc. Conference on Computer Vision and Pattern Recognition, 2019.
[64] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", arXiv:1711.09020v3, 2018
[65] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks",arXiv preprint arXiv:1703.10593v6, 2018
[66] Y. Li, M. Chang, and S. Lyu, “In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking”, in Proc. International Workshop on Information Forensics and Security, 2018.
[62] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Networks", arXiv preprint arXiv:1406.2661, 2014
[63] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks”, in Proc. Conference on Computer Vision and Pattern Recognition, 2019.
[64] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", arXiv:1711.09020v3, 2018
[65] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks",arXiv preprint arXiv:1703.10593v6, 2018
[66] Y. Li, M. Chang, and S. Lyu, “In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking”, in Proc. International Workshop on Information Forensics and Security, 2018.
[67] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector", arXiv preprint arXiv:1512.02325, 2016
[68] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", arXiv preprint arXiv:1506.01497, 2016
[69] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv preprint arXiv:1810.04805v2, 2019
[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention Is All You Need", arXiv preprint 1706.03762v5, 2017
[71] https://generated.photos/ , Accessed : 2020-07-07
[72] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html , Accessed : 2020-07-07

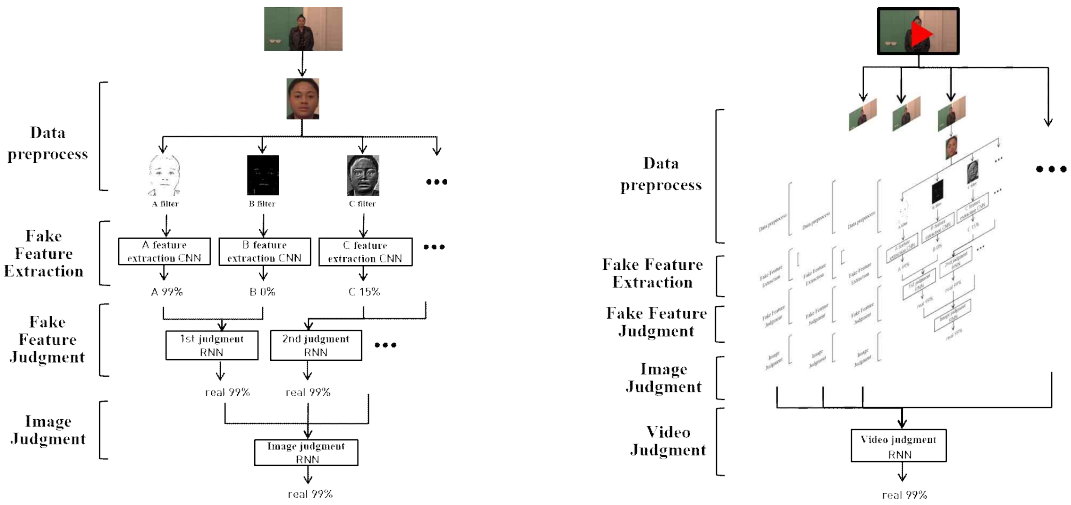
SUMMARY OF THIS PAPER

Deepfake refers to fake images created using deep learning. As deepfake videos spread on the internet including SNS, it causes social problems including fake news and defamation. Numerous detection models are purposed as the importance of the deepfake detection model is emphasized. However, those algorithms are not useful in the real world because they detect only a few parts of deepfake creation algorithms. Moreover, existing detection models have a limitation to study because they use high-end GPUs.

Deepfake images represents various fake features

To solve such problems, We propose a detection model that uses CNN(Convolutional Neural Network) to detect fake features and RNN(Recurrent Neural Network) to judge whether it is fake or not. This model can add CNN Detector to cope with the various deepfake creation algorithms in the real world. In addition, we designed the model suitable to reduce unnecessary features and extract necessary features easily during the Convolution process. It presents that model creation is possible without high-end hardwares.

Our model consists of 5-stages that are preprocess, fake feature extraction, fake feature judgment, image judgment, video judgment. The most optimal filter is applied in each feature during the preprocessing. We extract fake features discovered in images/videos using the CNN model during the fake feature extraction. RNN model judges Real/Fake based on the fake feature information extracted in the fake feature extraction during the fake feature judgment. In the image judgment stage, it judges images Real/Fake on the basis of result value from the previous stage. In the video judgment stage, it judges videos Real/Fake on the basis of judgment result value in each frame.[2)]



(a) (b)

Detection Model Architecture (a) Image Detection Process (b) Video Detection Process

In order to prove whether becoming constantly various algorithms is detectable, we conducted an experiment to see if the detection model improves the detection rate. We also conducted the experiment to prove that the detection performance of the proposed architecture model is similar to high-end GPU used models after training. The experimental result states that Recall increased from 2% to 75%, FPR decreased from 1.6% to 0.1%, AUC increased from 0.02 to 0.77 which is 0.13 lower than the high-end GPU model. However, the proposed detection model can perform a higher detection rate if it adds more fake features than the high-end GPU models.

---

2) github (Entire model code, instruction, Process Video are available) : https://github.com/teamnova-ailab/Deepfake-detection-model-based-on-fake-attributes-shown-in-image-video/