DeepfakeNet, an Efficient Deepfake Detection Method

Dafeng Gong[1]
Wenzhou Polytechnic, Wenzhou, China
Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka
Melaka, Malaysia

Yogan Jaya Kumar[2], Ong Sing Goh[3]
Zi Ye[4], Wanle Chi[5]
Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka
Melaka, Malaysia

***Abstract*—Different CNNs models do not perform well in deepfake detection in cross datasets. This paper proposes a deepfake detection model called DeepfakeNet, which consists of 20 network layers. It refers to the stacking idea of ResNet and the split-transform-merge idea of Inception to design the network block structure, That is, the block structure of ResNeXt. The study uses some data of FaceForensics++, Kaggle and TIMIT datasets, and data enhancement technology is used to expand the datasets for training and testing models. The experimental results show that, compared with the current mainstream models including VGG19, ResNet101, ResNeXt50, XceptionNet and GoogleNet, in the same dataset and preset parameters, the proposed detection model not only has higher accuracy and lower error rate in cross dataset detection, but also has a significant improvement in performance.**

***Keywords*—DeepfakeNet; deepfake detection; data enhancemen t; CNNs; cross dataset**

I. INTRODUCTION

Face is one of the most representative features in human beings' biometrics, with high recognition. At the same time, with the rapid development of face synthesis technology, the security threat brought by face tampering is becoming more and more serious. Especially in the era of mobile phones highly popular and social networks increasingly mature, the deepfake video using deep network model to replace face spreads rapidly in social media and the Internet, such as deepfacelab, faceswap [1].

At the end of 2018, the Dutch Deeptrace laboratory released a Deepfake development report[2] showing: Deepfake's global search volume has stabilized to as much as 1 million by December 2018, with at least 14678 fake videos, including 96% of pornography, and risk content such as violence, political sensitivity, advertising, contraband, etc., disguised in the video. The appearance of AI face changing is undoubtedly a great impact on its objective authenticity.

The threat of fake face video to society is increasing, and it has attracted widespread attention from academia and industry. There have been some related studies, and there are also international competitions for face-changing video detection[3]. According to the features used, the existing fake face video detection technologies are roughly divided into three categories[4]: based on traditional manual features, based on biological features, and based on neural network extraction of features. The first type of method mainly draws on the idea of image forensics and analyzes a single frame of images. Typical methods include the use of image quality measurement and principal component analysis[5] and the use of local binary pattern (LBP) features[6]. The second type of method mainly uses the unique biological information of the face. X. Yang et al. [7] proposed a model that can divide the facial landmarks into two groups according to the degree of influence during the tampering process, and use different landmarks to estimate the head posture direction and compare the differences. As a basis for discrimination, F. Matern et al. [8] found that the diffuse reflection information presented by the pupils of the two eyes in the fake face is inconsistent; The study by P. Korshunov and S. Marcel [9] uses both the video image and audio information to compare the lips in the true and false video. The difference between action and voice matching distinguishes whether there is tampering; S. Agarwal et al. [10] pointed out that every person has a unique movement pattern, and changing faces leads to a mismatch between the target object and the source object's movement patterns, which can be measured from the forehead, cheeks, nose, etc. The features are extracted from the movement changes of the region for classification decision. The third type of method mainly learns human faces by constructing a convolutional neural network, and extracts higher-dimensional semantic features for classification. Some researchers regard it as a conventional classification problem. The study by A. Khodabakhsh et al. [6] uses classic classification models such as AlextNet[11], VGG-19[12], ResNet[13], Inception and Xception[14] for image recognition to detect. D. Afchar et al. [15] Built Meso-4 and MesoInception-4, and S. Tariq et al. [16] built ShallowNet to detect single-frame images; B. Bayar and M. C. Stamm [17] pointed out that in the problem of tamper detection, tampering traces are more important than image content information. The MISLnet of the constrained convolutional layer suppresses the image content when extracting the shallow features; D. Guera et al. [18] considers the time domain information in the video and combines the convolutional neural network with the sequential neural network to find the continuous frame features in the fake face video inconsistency; S.-Y. Wang et al. [19] uses the ResNet-50[13] network model to detect different GAN composite images and Deepfake face images.

From the experimental results given in the above research, the high performance of the algorithm based on neural network extraction features often depends on the dataset used. In the

cross-dataset, due to the large domain offset, that is, the size of the dataset target is different, most of the algorithms and models performs poorly, such as VGG19, GoogleNet, XceptionNet, ResNet50. Background complexity, resolution, and the quality of the synthetic fake face are different, which makes the data distribution vary greatly, which leads to the model's inability to make correct judgments and poor detection results, resulting in a significant drop in performance during cross-dataset [6].

This paper proposes a solution to the generalization performance of face-swapping video detection. Different from the above-mentioned method based on feature detection, and starts from the image and believes that fake face tampering is a special splicing tampering problem. According to the fact that face changing mainly operates on part of the face area without modifying the content of other images, DeepfakeNet is proposed, which is a detection method based on image segmentation and deep residual network. The key contributions in this study are: (1) Extracting video data from multiple data sources, and creating a unified experimental data set through data enhancement methods; (2) Proposing an improved structure based on ResNeXt[20], as whether face change occurs judgment basis for tampering; (3) A model parameter is trained to obtain better detection results.

This paper first introduces the generation principle of deepfake and the current detection technology of deepfake; Then, the DeepfakeNet detection algorithm proposed in this paper is described, and the network structure of the model is explained; Then set up the experimental environment, expand the data set using data enhancement technology, and use relevant standards to compare the mainstream models to draw conclusions; Finally, the possible direction of further efforts in this field is put forward.

In this study, Prof. & Dr. Goh put forward the overall idea; Dr. Yogan gives guidance and improvement; Dafeng Gong designs algorithm, implementation, and analysis conclusion; Zi Ye preprocesses the dataset; Wanle Chi verifies the model.

II. Related Technology

A. The Basic Architecture of Deepfake

Deepfake is a neural network trained with an unsupervised learning method. It regenerates the original input after encoding the distorted face image, and expects this network to have the ability to restore any face. The overall process of implementing Deepfake to change the face is shown in Fig. 1, that is, the final realization of FaceA to replace FaceB in a video or image. First, obtain images with A/B faces from the video or image collection, and perform face detection and face alignment. Specifically, on the basis of face detection, the location of key facial features is performed, and the detected faces are normalized and aligned through affine transformation, which can intercept half-face or full-face facial images; then The intercepted A/B face image is sent to the neural network for training and conversion to obtain A face with B expression, action, environment and other conditions, and then the output face image is overlaid on the original image (Fig. 1), and the edges are smoothed then re-synthesize the video.

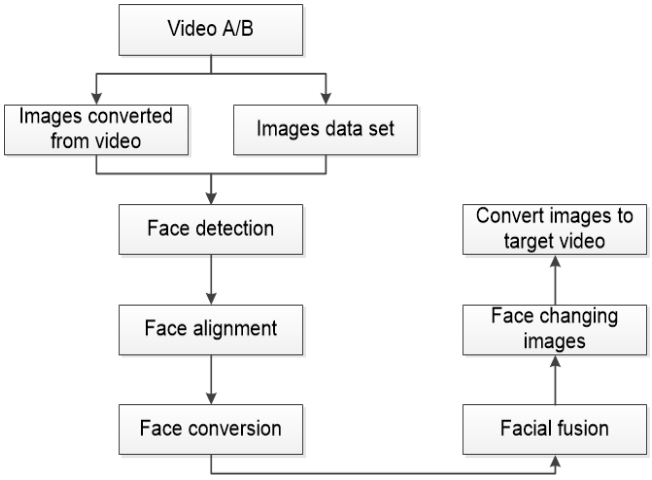

Fig. 1. Face-Swapping Process through Deepfake.

The above process is based on GAN network, and the generation process of data can be represented by the following two public announcements. Goodfellow, I. J et al. [21] proposed data is generated in the following way:

$$\lim_{\sigma\to 0} \nabla_x \, E_{\epsilon\sim N(0,\sigma^2 I)} f(x+\epsilon) = \nabla_x f(x) \quad (1)$$

The function of optimization of multilayer perceptrons is:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)} [\log D(x)] + E_{z\sim p_z(z)} \left[\log\left(1 - D(G(z))\right)\right] \quad (2)$$

In the above equation, D(x) represents the probability of D judging x from real data, D(G(z)) represents the probability of D judging G(z) as real data, Z represents random noise, and G(z) is the probability of generating data, that is, the probability of D judging as fake, and 1-D(G(z)) is the probability of D judging as true; for D, the probability of other judging as true is maximized, while for G, the objective function is minimized.

B. Deepfake Detection

The development of Deepfake technology also gave birth to the corresponding detection technology. Yuezun Li et al. [22] used a deep neural network model to combine the convolutional neural network (CNN) and the recurrent neural network (RNN) to form a long and short-term memory network (LRCN). It is believed that fake videos generally do not show the characteristics of blinking, breathing, and eye movement, so that it can detect whether the face in the image or video is real or generated by AI; Matern et al. [8] proposed using image artifacts to detect facial forgery; Yuezun Li et al. [23] trained CNN to detect this facial artifact in face-changing videos; Haodong Li et al. [24] used the difference in color components between real images and forged facial images to distinguish between authenticity and fake images; Yang et al. [7] proposed an SVM support vector machine classification method based on the mismatch between the head pose and the position of the important facial features; Koopman et al. [25] predicted that the tampering of the facial region would affect the local illumination response inconsistency noise in the video frame, and proved that the noise can be used distinguish between original video and face-changing video, but a larger

data set must be studied to determine the correlation between the two; because FakeApp generates Deepfake fake video, it will cause inconsistencies within and between frames, Guera et al. [18] proposed the use of recurrent neural networks (RNN) to detect this anomaly in videos; Afchar et al. [15] proposed two low-level neural networks for detection; Korshunov et al. [5] evaluated the reliability of several detection methods and proposed: Advanced VGG and Facenet-based neural network face recognition algorithms cannot distinguish face-changing videos from original videos. Detection algorithms based on lip shape and voice inconsistency cannot distinguish Deepfake fake videos, while image quality detection based on support vector machine classification. The technology can detect high-quality (128×128) face-changing images with low error.

However, algorithms based on neural network extraction of features can often achieve higher accuracy in cross-dataset detection, and the main drawback is that the performance of cross-dataset detection drops sharply, and there is a problem of insufficient generalization performance[6].

III. DEEPFAKE DETECTION ALGORITHM

This study refers to the ResNeXt network [20], using ResNet's stacking ideas [13] and Inception's split-transform-merge ideas [26]. The calculation process can be expressed as follows:

$$y = x + \sum_{i=1}^{C} T_i(x) \quad (3)$$

Where x is a short-cut, each feature undergoes a linear transformation, C is the cardinality of simple Inception, and T_i is any transformation, such as a series of convolution operations.

Its basic block structure is the same as ResNeXt, as shown in Fig. 2. Each box represents a layer, and the meanings of the three data representations are: input data channel, filter size, output data channel. The advantage is to improve the accuracy through a wider or deeper network under the premise of ensuring the amount of FLOPs and parameters. For each block structure, the convolution kernel is grouped by channel to reduce the dimensionality of the data to form 32 parallel branches, and then respectively perform convolution transformation and feature change on 32 low-latitude data, and then aggregate them back to the original by addition dimension. The final network structure is shown in Table I, consisting of a 20-layer network, here called DeepfakeNet. This network architecture is shown in Fig. 3.

There are five groups of convolutions in this network. The input image size of 1st group is 224x224, and the size of output data of 5th group is 7x7, which is reduced by 32 (2^5) times. Each time, the stride is set to 2 on the first layer of each group of convolutions, and each time is reduced by a factor of 2, a total of 5 times is reduced.

The overall architecture and process of the detection model is mainly composed of 3 main parts: preprocessing data module, extracting feature module and deep learning model module. In the data preprocessing module, the video data set is first processed. For frame images, then enhance the data set. Since only the face of the person in the video frame is the detection target, the extracted video frame is intercepted, and the features of face images are extracted by CNN. At the same time, the powerful extracting feature ability of CNN can be used to more accurately judge images whether it has been modified. After sufficient training and verification, the DeepfakeNet model is continuously improved to obtain better results.

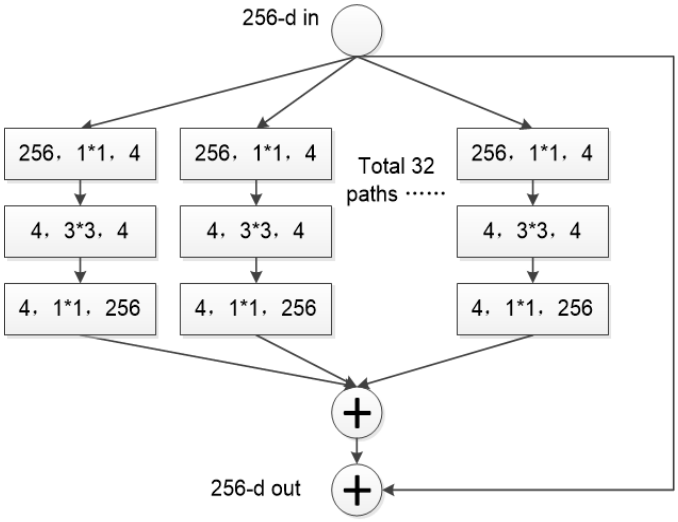

Fig. 2. A Block of ResNeXt with Cardinality = 32, with Roughly the Same Complexity.

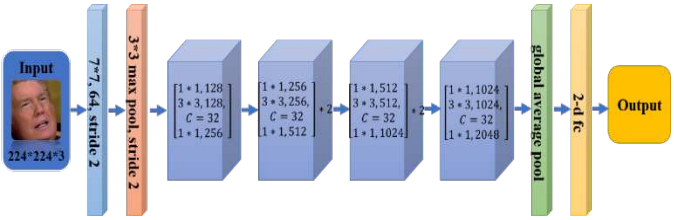

Fig. 3. DeepfakeNet Architecture.

TABLE I. DEEPFAKENET WITH A 32*4D TEMPLATE.

state	output	DeepfakeNet (32*4d)
Conv1	112*112	7*7, 64, stride 2
Conv2	56*56	3*3 max pool, stride 2
		$\begin{bmatrix} 1*1, 128 \\ 3*3, 128, C=32 \\ 1*1, 256 \end{bmatrix} * 1$
Conv3	28*28	$\begin{bmatrix} 1*1, 256 \\ 3*3, 256, C=32 \\ 1*1, 512 \end{bmatrix} * 2$
Conv4	14*14	$\begin{bmatrix} 1*1, 512 \\ 3*3, 512, C=32 \\ 1*1, 1024 \end{bmatrix} * 2$
Conv5	7*7	$\begin{bmatrix} 1*1, 1024 \\ 3*3, 1024, C=32 \\ 1*1, 2048 \end{bmatrix} * 1$
	1*1	global average pool 2-d fc, softmax
params		10.87 * 106
FLOPs		2.05 * 109

IV. Experimental Setup

A. Lab Environment

In order to verify the performance and effectiveness of the detection model, this study selects videos from the deepfake open source datasets such as FaceForensics++[27], TIMIT[5] and Kaggle competitions[3] for experiments, as shown in Table II. Since the deep learning model in this article reads the training set according to a sequence of consecutive frames, the video must be converted into a sequence of frame images.

In order to obtain an input image of a uniform size, the videos in each dataset are divided into frames, and the face is located by the convolutional neural network detector in the Dlib library frame by frame, and k (k>1) times the face is taken as the center of the face frame. The size of the image area, sampled to the size of 224 × 224, as the input image. In order to effectively train and test the network, this paper expands the training samples to enhance the diversity of the data, so that the model can adapt to a wide range of application environments, and has a wide range of applications in target recognition and target detection.

Common methods of enhancing dataset include stretching, rotating, flipping, etc., as shown in Table III. Perform the following operations on each image: compress or stretch to between 0.75 and 1.25 times; each image is generated from 30 degrees left to 30 degrees right, every 2 degrees; brightness changes from the original brightness 0.75 to 1.25 times of, each 0.1 times difference generates one; at the same time, each image is flipped horizontally and vertically. These operations are performed in order to obtain a sufficiently large set of enhanced data[15][28].

This experiment was done on an ubuntu server with 2 Intel Xeon Silver 4214 CPUs, 192GB RAM, and NVIDIA Tesla V100 32GB PCIe GPU. The deep learning model is implemented using Python language and Pytorch framework.

In order for the model to fully learn the feature information of the data set, this study sets the number of iterations of the model epoch to 300, and sets the loss function of the deep learning model to the mean-square error (MSE) loss function. The calculation method is as follows:

$$L_{loss} = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 \quad (4)$$

In the formula, m is the number of samples, y_i is the label of the sample, and $\hat{y}_i$ is the predicted value of the model. The loss function L_{loss} can well show the degree of fit between the true label and the predicted results. The smaller the value, the better the degree of fit. In deep learning, the optimization algorithm is used to optimize the model, so that the value of L_loss tends to the minimum, and the fitting ability of the model is increased. The dropout used in the training of this paper is set to 0.8, the learning rate is set to 1×10-5, and the optimization algorithm is Adam [20]. The experimental data set is divided into a training dataset and a verification dataset. Each experiment was randomly assigned to generate them, corresponding to 80% and 20% respectively. The model is trained on the training dataset and verified on the verification dataset.

TABLE II. Composition of Data Sources

Data Source	Total Quantity	Number of Videos Selected
FaceForensics++	4000	3200
Kaggle	117812	21767
TIMIT	1199	960

TABLE III. Approaches of Enhancing Data

Types	stretching	rotating	Brightness change	Flip horizontally/ vertically
Params	[0.75, 1.25]	[-30, 30]	[0.75, 1.25]	Yes/Yes

B. Evaluation Measures

In order to evaluate the performance based on the temporal and spatial feature consistency detection model, this paper selects a variety of metrics to evaluate the model. First of all, for classification models, accuracy is often used to evaluate the global accuracy of a model. The higher the accuracy, the better the accuracy of the model. The calculation formula of the accuracy rate is as follows:

$$A_{accuracy} = \frac{T_{TP}+T_{TN}}{T_{TP}+T_{TN}+F_{FP}+F_{FN}} \quad (5)$$

In this formula, T_{TP} is true positive (TP), which refers to the number of fake images that are correctly classified; T_{TN} is true negative (TN), which refers to the number of real images that are correctly classified; F_{FP} is False positive (FP) , which refers to the number of fake images that have been misclassified; F_{FN} is false negative (FN), which refers to the number of real images that have been misclassified.

In order to evaluate the model more comprehensively, in addition to selecting the accuracy rate, this article also selects the area under roc curve (AUC) as the evaluation index in addition to the receiver operating characteristic curve (ROC). The ROC curve is based on the predicted results of the model. Sort the samples by size, use the predicted probability of each sample as the threshold value one by one in this order, calculate the false positive rate (false positive rate, F_{PR}) and the true case rate, that is, the recall rate (true positive rate, T_{PR}), and Take F_{PR} as the horizontal axis and T_{PR} as the vertical axis. Among them, the calculation formulas of F_{PR} and T_{PR} are as follows:

$$F_{PR} = \frac{F_{TP}}{T_{TN}+F_{FP}} \quad (6)$$

$$T_{PR} = \frac{T_{TP}}{F_{FN}+T_{TP}} \quad (7)$$

ROC curve can well represent the generalization performance of a model. AUC is the area under the ROC curve. The larger the AUC value, the better the performance. The calculation formula of AUC is as follows:

$$A_{AUC} = \frac{1}{2}\sum_{i=1}^{m}\left(F_{PR}^{(i+1)} - F_{PR}^{(i)}\right) \times \left(T_{PR}^{(i)} - T_{PR}^{(i+1)}\right) \quad (8)$$

In this formula, m is the number of samples.

V. Experimental Result

A. Analysis of Algorithm Effectiveness

After a lot of experiments with the same parameters, such as Table IV (all models uses the same preset parameters), we get the data curves as shown in Fig. 4 and Fig. 5. It can be seen from Fig. 4 that as the number of training increases, the Loss function value of the model on the validation dataset and the training dataset gradually decreases, indicating that the model's fitting ability is getting stronger and stronger, which fully illustrates the effectiveness of the experimental model in this paper.

Fig. 5 shows that as the number of model iterations increases, the accuracy of the model's classification prediction on the training set and validation set also gradually increases, indicating that the model's effect is getting better and better. It can also be seen from Fig. 5 that the accuracy of the model tends to be stable around the 160th epoch. The training accuracy of the final model can reach about 97.13%, and the accuracy of the verification data set is about 96.69%, indicating that the model is good; classification and detection results. With the increase in the number of model iterations, the model's fitting ability has been slightly improved, but due to the strong fitting ability of the deep learning model, it is prone to overfitting the training dataset. In order to avoid over-fitting, this paper adopts an early stopping strategy in the experiment. The early stopping strategy is often used in deep learning model training, that is, when the loss function value of the model does not improve for a period of time, the training of the model is terminated.

As shown in Fig. 6, it is part of our experimental results. The number on the top of each small image represents the probability that the operation result is true or false, the final prediction result and the labelled value. For example, the number of the small image in the upper right corner is (0.02, 0.98 | 1 | 1), which indicates the result of operation with DeepfakeNet model. The probability of 0.02 is true, and the probability of 0.98 is false, so the prediction result is false (1), which is consistent with the actual labelled data (also false (1)).

In order to verify the effectiveness of the model, this paper uses common models to carry out comparative experiments based on the same dataset and the same preset parameters, and the comparative results are shown in Table V. It can be seen from Table V that the model proposed in this study has better accuracy than VGG, GoogleNet, XceptionNet, ResNet, ResNeXt and so on.

TABLE IV. Preset Parameters

Params	**Value**
batch_size	128
epochs	300
dropout	0.8
max_lr	0.00001
sample_ratio	2.0

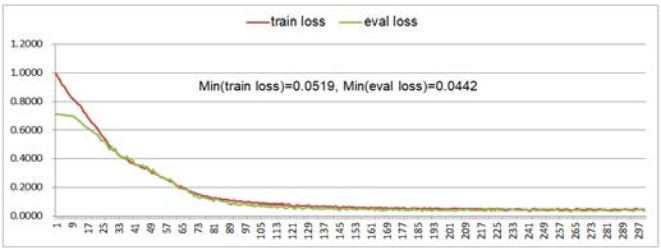

Fig. 4. Curve of Loss Value Changing with Training Times.

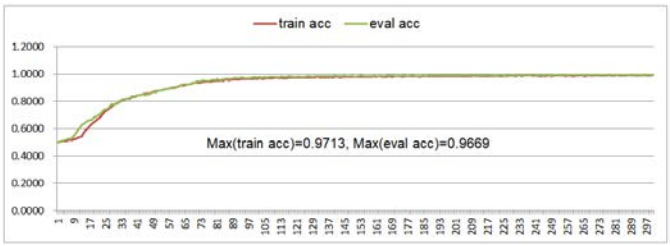

Fig. 5. The Curve of Accuracy with Training Times.

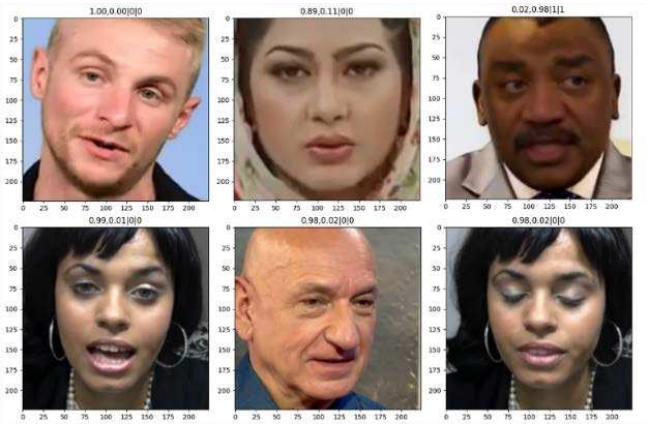

Fig. 6. Samples of Experimental Results.

TABLE V. Accuracy Comparison of Experimental Results

Models	Accuracy (%)
VGG19	80.22
GoogleNet	88.94
XceptionNet	92.03
ResNet101	93.78
ResNeXt50	94.36
Model of this paper (DeepfakeNet)	96.69

In order to compare the performance of the models in many aspects, AUC was selected as the comparison index. The comparison results are shown in Table VI. The AUC value can not only reflect the detection effect of models, but also represent the generalization ability of the model. From Table VI, it can be seen that the model in this paper has advantages over other models, indicating that the detection effect and generalization performance of the model in this paper are better than other models.

TABLE VI. AUC VALUE COMPARISON OF EXPERIMENTAL RESULTS

Models	AUC
VGG19	0.83
GoogleNet	0.92
XceptionNet	0.92
ResNet101	0.93
ResNeXt50	0.94
Model of this paper (DeepfakeNet)	0.96

B. Analysis of Algorithm Performance

This section discusses the computational complexity of each algorithm, and compares it according to the number of floating-point operations (FLOPs) and the number of parameters for each model. Generally speaking, if the same effect is achieved, the smaller the number of FLOPs and parameters, the better. Table VII shows FLOPs and the number of parameters of some current mainstream network architectures. Although Table VII shows that GoogleNet is superior in the number of FLOTs and parameters, combined with Table V and Table VI, its accuracy (88.94) and AUC (0.85) are compared with the corresponding data of DeepfakeNet (98.69 and 0.98, respectively). Compared with ResNeXt50, the FLOPs of this model (2.05) is about 48% of it (4.27), and the parameters number of this model (10.87) is about 43% of it (25.08). Overall, performance of the network structure proposed in this study is better.

TABLE VII. COMPARISON OF NUMBER OF FLOPS AND PARAMS

Models	FLOPs	Params
VGG19	19.67 * 109	145.77 * 106
ResNet101	7.85 * 109	44.6 * 106
ResNeXt50	4.27 * 109	25.08 * 106
XceptionNet	3.81 * 109	22.8 * 106
GoogleNet	1.51 * 109	6.13 * 106
Model of this paper (DeepfakeNet)	2.05 * 109	10.87 * 106

VI. CONCLUDING REMARKS

At present, most of the popular fake face video detection algorithms use deep network to extract features. The main reason for the poor cross-dataset performance is that the deep network is easy to learn too many features in the dataset, resulting in poor generalization performance. This paper treats fake face video detection as a special image mosaic tampering detection problem, and uses image segmentation and deep residual network to predict the tampered area, reduces the impact of different training datasets, and improves the generalization performance of the detection algorithm. The experimental results on multiple popular face-swapping video dataset show that compared with other similar algorithms, the method in this paper greatly reduces the average error rate of cross-dataset detection while maintaining high accuracy in the dataset detection. The algorithm has good generality. The method in this paper can obtain good fake face video detection performance in different data sources, which shows that the idea of improving generalization performance in this paper is general. Future improvements include expanding the scope of the training set, solving the precise detection of faces with different video quality, optimizing the network model, and developing a more complex and effective face tampering video detection network to improve usage.

ACKNOWLEDGMENT

This work was supported by Wenzhou basic scientific research project of Wenzhou Science and Technology Bureau in 2020 (No. G2020033). We would also like to thank Universiti Teknikal Malaysia Melaka for the collaboration.

REFERENCES

[1] Z. Zhang and Q. Liu, “Detect Video Forgery by Performing Transfer Le arning on Deep Neural Network,” Int. Conf. Nat. Comput. Fuzzy Syst. Knowl. Discov., pp. 415–422, 2020, doi: 10.1007/978-3-030-32591-6_4 4.D. B. V, “The state of Deepfakes: reality under attack,” 2019.

[2] “DeepFake Detection Challenge.” [Online]. Available: https://www.kag gle.com/c/deepfake-detection-challenge.

[3] H. Yongjian, “Deepfake Videos Detection Based on Image Segmentatio n with Deep Neural Networks,” J. Electron. Inf. Technol., vol. 43, no. 1, pp. 162–170, Jan. 2021, doi: 10.11999/JEIT200077.

[4] P. Korshunov and S. Marcel, “DeepFakes: a New Threat to Face Recogn ition? Assessment and Detection,” arXiv Prepr., pp. 1–6, 2018.

[5] A. Khodabakhsh, R. Ramachandra, K. Raja, P. Wasnik, and C. Busch, “ Fake Face Detection Methods: Can They Be Generalized?,” in 2018 Inte rnational Conference of the Biometrics Special Interest Group, BIOSIG 2018, 2018, pp. 1–6, doi: 10.23919/BIOSIG.2018.8553251.

[6] X. Yang, Y. Li, and S. Lyu, “Exposing Deep Fakes Using Inconsistent H ead Poses,” in ICASSP, IEEE International Conference on Acoustics, Sp eech and Signal Processing - Proceedings, 2019, doi: 10.1109/ICASSP.2 019.8683164.

[7] F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose deepfakes and face manipulations,” Proc. - 2019 IEEE Winter C onf. Appl. Comput. Vis. Work. WACVW 2019, pp. 83–92, 2019, doi: 1 0.1109/WACVW.2019.00020.

[8] P. Korshunov and S. Marcel, “Speaker inconsistency detection in tamper ed video,” in European Signal Processing Conference, 2018, pp. 2375–2 379, doi: 10.23919/EUSIPCO.2018.8553270.

[9] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, “Protecting world leaders against deep fakes,” Proc. IEEE Conf. Comput. Vis. Patter n Recognit. Work., pp. 38–45, 2019.

[10] G. E. KRIZHEVSKY, Alex, SUTSKEVER, Ilya, HINTON, “ImageNet Classification with Deep Convolutional Neural Networks,” Adv. Neural Inf. Process. Syst., vol. 25, pp. 1097–1105, 2012.

[11] Z. A. Simonyan K, “Very Deep Convolutional Networks for Large-Scal e Image Recognition,” arXiv Prepr., 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image r ecognition,” in Proceedings of the IEEE Computer Society Conference o n Computer Vision and Pattern Recognition, 2016, doi: 10.1109/CVPR. 2016.90.

[13] F. Chollet, “Xception: Deep learning with depthwise separable convoluti ons,” in Proceedings - 30th IEEE Conference on Computer Vision and P attern Recognition, CVPR 2017, 2017, pp. 246–253, doi: 10.1109/CVPR .2017.195.

[14] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “MesoNet: A comp act facial video forgery detection network,” 2018 IEEE Int. Work. Inf. F orensics Secur., pp. 1–7, 2018, doi: 10.1109/WIFS.2018.8630761.

[15] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo, “Detecting both machi ne and human created fake face images in the wild,” in Proceedings of th e ACM Conference on Computer and Communications Security, 2018, p p. 81–87, doi: 10.1145/3267357.3267367.

[16] B. Bayar and M. C. Stamm, “Constrained Convolutional Neural Networ ks: A New Approach Towards General Purpose Image Manipulation Det ection,” IEEE Trans. Inf. Forensics Secur., vol. 13, no. 11, pp. 2691–270 6, 2018, doi: 10.1109/TIFS.2018.2825953.

[17] D. Guera and E. J. Delp, “Deepfake Video Detection Using Recurrent N eural Networks,” in Proceedings of AVSS 2018 - 2018 15th IEEE Intern ational Conference on Advanced Video and Signal-Based Surveillance, 2019, pp. 1–6, doi: 10.1109/AVSS.2018.8639163.

[18] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “CNN-gen erated images are surprisingly easy to spot... for now,” arXiv Prepr., 201 9.

[19] K. Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, “Aggregated Residual Transformations for Deep Neural Networks,” IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 1492–1500, 2017.

[20] I. J. Goodfellow et al., “Generative Adversarial Networks,” in Advances in Neural Information Processing Systems, 2014, pp. 2672–2680, doi: 1 0.3156/jsoft.29.5_177_2.

[21] Y. Li, M.-C. Chang, H. Farid, and S. Lyu, “In ictu oculi: Exposing ai ge nerated fake face videos by detecting eye blinking,” 2018 IEEE Int. Wor k. Inf. Forensics Secur., pp. 1–7, 2018.

[22] Y. Li and S. Lyu, “Exposing DeepFake Videos By Detecting Face Warp ing Artifacts,” arXiv Prepr., 2019.

[23] J. Li, H., Li, B., Tan, S., Huang, “Detection of deep network generated i mages using disparities in color components,” arXiv Prepr., 2018.

[24] M. Koopman, A. M. Rodriguez, and Z. Geradts, “Detection of Deepfake Video Manipulation,” 20th Irish Mach. Vis. Image Process. Conf., pp. 1 33–136, 2018.

[25] and A. R. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelo v, D. Erhan, V. Vanhoucke, “Going deeper with convolutions,” IEEE Co mput. Soc., 2014.

[26] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießn er, “FaceForensics: A large-scale video dataset for forgery detection in h uman faces,” arXiv. 2018.

[27] Y. Gao, Y. Hu, Z. Yu, Y. Lin, and B. Liu, “Evaluation and Comparison of Five Popular Fake Face Detection Networks,” Yingyong Kexue Xueb ao/Journal Appl. Sci., vol. 37, no. 5, pp. 590–608, 2019, doi: 10.3969/j.i ssn.0255-8297.2019.05.002.