Classification of Deepfake Videos Using Pre-trained Convolutional Neural Networks

Conference Paper · May 2021

DOI: 10.1109/ICoDT252288.2021.9441519

6 authors, including:

Momina Masood

35 PUBLICATIONS **2,262** CITATIONS

SEE PROFILE

Marriam Nawaz

University of Engineering and Technology Taxila

67 PUBLICATIONS **3,055** CITATIONS

SEE PROFILE

Ali Javed

University of Engineering and Technology Taxila

186 PUBLICATIONS **4,595** CITATIONS

SEE PROFILE

Tahira Nazir

Riphah International University

61 PUBLICATIONS **2,283** CITATIONS

SEE PROFILE

2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2)

Classification of Deepfake Videos Using Pre-trained Convolutional Neural Networks

Momina Masood, Marriam Nawaz, Ali Javed, Tahira Nazir*, Awais Mehmood, Rabbia Mahum
Department of Computer Science, UET, Taxila, Pakistan
{momina.masood, marriam.nawaz, ali.javed}@uettaxila.edu.pk,
tahira.nazir77@gmail.com,
(awais.mehmood, rabbia.mahum)@uettaxila.edu.pk.

2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2) | 978-1-6654-1285-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICoDT252288.2021.9441519

***Abstract*— The advancement of Artificial Intelligence (AI) has brought a revolution in the field of information technology. Furthermore, AI has empowered the new applications to run with minimum resources and computational cost. One of such applications is Deepfakes, which produces extensively altered and modified multimedia content. However, such manipulated visual data imposed a severe threat to the security and privacy of people and can cause massive sect, religious, political, and communal stress around the globe. Now, the face-swapped base visual content is difficult to recognizable by humans through their naked eyes due to the advancement of Generative adversarial networks (GANs). Therefore, identifying such forgeries is a challenging task for the research community. In this paper, we have introduced a pipeline for identifying and detecting person faces from input visual samples. In the second step, several deep learning (DL) based approaches are employed to compute the deep features from extracted faces. Lastly, a classifier namely SVM is trained over these features to classify the data as real or manipulated. We have performed the performance comparison of various feature extractors and confirmed from reported results that DenseNet-169 along with SVM classifier outperforms the rest of the methods.**

Keywords*— *deepfakes, deep-learning, visual manipulations, convolutional neural networks.

I. INTRODUCTION

The easier accessibility to computationally efficient smart gadgets i.e. mobile phones, tabs, computers, and digital cameras has caused a serious increase in multimedia data (i.e. images and videos) over the internet. Moreover, the advancement in the usage of social media enables people to share audio-visual content which ultimately results in easier accessibility to multimedia content without the consent of users. Furthermore, great enhancement in the field of Machine Learning (ML) is evident due to the introduction of innovative techniques that can easily change this digital data for spreading false information via social websites. The recent era is known as "post-truth" in which a chunk of information and disinformation is controlled by malicious actors to affect people's beliefs. Deepfakes has the power to originate severe damages such as affecting the elections, generation of warmongering situations, affecting the reputation of famous personalities, etc. The easier access to many ML-based tools and apps i.e. Zao [1], REFACE[2], FaceApp[3] can assist humans to make their content more pleasing and charming. Because of the simple creation and spread of false data, now it has become very difficult to know what to trust and what to not. Especially, it is evident for those cases where these images and videos can be used in investigating a criminal case or processing other legal claims. Videos utilized as evidence must be trustworthy (i.e., it must contain the detail of all events since its generation), and its truthfulness (i.e., identification of disjointedness in the visual content, precise operations, e.g., insertion, compression, etc.) must be confirmed. Due to the sophisticated creation of deepfakes, now, it is becoming a challenging task to verify the authenticity of videos [4].

There are several positive usages of deepfakes i.e. deepfakes can deliver economical solutions to several problems, for example, deepfakes can produce audio sounds for the persons who have lost their speech, artists can employ them to exhibit their creativity. Moreover, movie makers can utilize deepfakes generation techniques to update the scenes without the need to reshoot them [5]. However, the negative usages of deepfakes are more dominant. As in the past, deepfakes have been created to make celebrities controversial to their followers, i.e., in 2017 a female celebrity was confronted with a situation in which her deepfake pornographic video was widely shared on the internet [6]. Hence, deepfakes can be used to influence people's goodwill for a variety of purposes, including character assassination of well-known personalities to insult them [6], inciting religious or political strife by threatening religious scholars or politicians with manipulated videos or speeches [7], and so on. Furthermore, with alter news creation, deepfakes can have a substantial influence on the stock markets around the globe. Moreover, with the introduction of the few-shot deepfakes generation, now, its impact is not limited to celebrities only. For example, the apps like Zao [8] enables the common people to swap his/her face with celebrities and visualize himself/herself in the clips of famous dramas and movies. Hence, such apps can be employed to cause severe confidentiality harms for not only well-known persons but common people as well.

Researchers have introduced several methods to identify fake digital content. The methods used for deepfakes detection are broadly categorized as either ML-based methods or deep-learning (DL) based approaches. Yang et al. [9] proposed a deepfake detection technique that used 2D facial landmarks to compute the 3D head position. To train the SVM classifier, the calculated difference between the head poses was employed as a keypoints vector. This approach shows better performance for detecting deepfakes however, unable to compute the landmark alignment from the blurred samples which reduces the robustness of this framework under these cases. The work in [10] utilized the Image Quality Metric (IQM) together with the principal

* Corresponding Author

component analysis (PCA) and linear discriminant analysis (LDA) for keypoints computation. The SVM classifier was used to train the obtained and classify the input as fake or real. The results show that the face recognition approaches i.e., Facenet [11] and Visual Geometry Group (VGG) [12] are incompetent to identify visual manipulations. In [13] authors introduced a deepfake detection technique to protect famous personalities. Initially, deepfakes were generated by employing GAN, on which OpenFace2 [14] toolkit was applied to compute the face features and head movements. The binary SVM was trained to distinguish between genuine and forged faces using the measured landmarks.

Recently, DL-based approaches are getting the attention of researchers for deepfakes detection. In [15] authors presented an approach to exploit deepfakes by using the concept that the forged faces in the manipulated videos lack accurate eye blinking. A Spatio-temporal network based on CNN/RNN was employed to locate an absence of eye blinking from the visual content to identify the altered video. This approach exhibits good deepfakes identification accuracy, however, it just utilizes the absence of eye blinking as a signal to identify deepfakes that can be easily evaded by the deepfake generation algorithms. In [16] authors computed pixel co-occurrence matrices from the three color channels of the input sample to detect manipulation. Then, a CNN was used to extract a representative set of features from it. In [17], authors observed that the manipulation algorithms often fail to effectively enforce temporal coherence in the forged video during the synthesis process. The work used a recurrent convolutional neural network to examine the temporal inconsistencies for the identification of altered faces in the frames. These methods [16], [17] acquired improved detection accuracy, however, applicable to static samples only. Afchar et al. [18] presented a methodology that performs a mesoscopic level analysis of forged videos by using CNN namely Meso-4 and MesoInception-4. This approach is computationally efficient, however, at the expense of performance degradation for deepfakes identification. In [19] a multi-task learning-based CNN framework is applied to simultaneously identify and locate the altered regions in the forged visual content. The network comprises of an autoencoder for the binary classification of encoded features, and a y-shaped decoder for the reconstruction of input and segmentation of manipulated region. This method exhibits better deepfakes detection performance; however, the accuracy decreases significantly in real-world scenarios.

Although extensive work has been resented in the field of deepfakes detection, however, still there is a room for performance improvement both in terms of efficiency and effectiveness. As day by day, the deepfakes generation techniques are getting enhanced which in turn produce more challenging datasets, on which the existing techniques may not perform well. The major contributions of presented framework are:

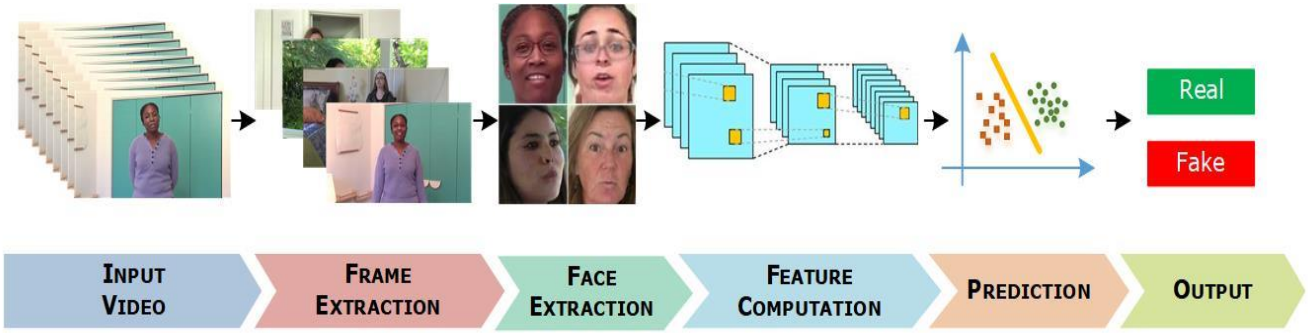

Fig. 1. Workflow diagram of the proposed methodology

- A generic pipeline for deepfakes detection by using pre-trained models and employing the concept of transfer learning to deal with the problem of data over-fitting.
- We compared the performance of ten popular deep learning models i.e., VGG-16 [20], VGG-19 [20], ResNet101 [21], Inception V3 [22], XceptionNet [23], InceptionResV2 [24], DenseNet-169 [25], MobileNetv2 [26], EfficientNet [27] and NASNet-Mobile [28] for deepfake detection.
- We performed performance evaluation over a challenging database namely Deepfake Detection Challenge Dataset (DFDC) to show the robustness of deep-learning models.

The remaining of the manuscript is structured as: Section II presents the introduced methodology; section III provides experiment details and a discussion of obtained results. Finally, section IV presents the conclusion of the presented work.

II. METHODOLOGY

This section describes the proposed workflow employed for deepfakes detection. Fig. 1 shows the workflow diagram of our evaluation framework. The proposed system comprises three main steps (i) face detection and extraction (ii) feature computation and (iii) prediction.

i) Face detection and extraction:

The first step is face identification and extraction from video frames. In deepfake videos, the facial region is the main portion of the frame that is manipulated. So, we are concerned with the face area only. A face detector is employed to locate and extract the face region from the rest of the frame. In our implementation, we considered OpenFace2[14] toolkit for face detections. OpenFace uses 2D and 3D facial landmarks to detect the face and is capable of estimating the head posture, eye-gaze, and identification of facial action units. It is important to highlight that the OpenFace is robust against variations in facial pose, light condition, and camera position and thus it

can effectively detect and crop face under such varying conditions [29]. Furthermore, instead of using all frames, we have considered only 20 frames from each video sample to reduce the computational cost.

ii) Feature computation:

This step involves the computation of features from the extracted face images which are later used by the classifier to classify them as fake or real. We have considered ten different state-of-the-art models for feature extraction. The pre-trained CNNs models used in our study are VGG-16 [20], VGG-19 [20], ResNet101 [21], Inception V3 [22], XceptionNet [23], InceptionResV2 [24], DenseNet-169 [25], MobileNetv2 [26], EfficientNet [27] and NASNet-Mobile [28]. The motivation to employ pre-trained frameworks is that these models have been trained on massive, publically accessible databases, i.e. ImageNet, and are therefore capable to learn important keypoints. During training the initial layers learn low-level features, as the network goes up the layers can learn task-specific patterns. Thus, when the pre-trained models are trained for a new task, such as the classification of deepfake videos, the training speed, and accuracy of the new model increase. As the significant sample keypoints have already been computed do not have to be learned again and are transferred to the new task. This process is known as 'transfer learning'. Fig. 2 shows a flow diagram of fine-tuning the pre-trained deep-learning model using transfer learning. The input images are resized to 224x224 before feeding them into these CNN models. The employed models learn important features from the face samples i.e., face texture, eyes, nose orientation and lips size etc.

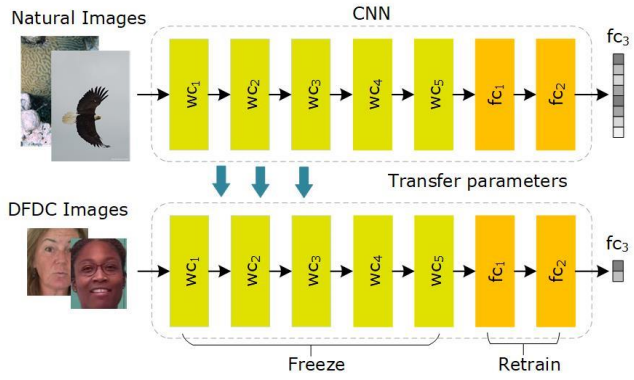

Fig. 2. Flow diagram of fine-tuning the model on DFDC dataset using transfer learning

VGG [20] is a CNN model introduced by Visual Geometry Group and is among the top-performing models on the ImageNet database. The model is known for its simplicity and efficiency. It comprises several 3x3 convolutional and 2x2 max-pooling layers stacked on top of each other forming the depth of the network to 16 and 19 layers respectively. VGG16 model predicts with an output dimension of (1,512)

ResNet-101 (Residual Neural Network) [21] introduces an identity shortcut connection that skips one or more layers. The shortcut connection applies identity mapping, and the computed outcomes are passed to the outputs of the stacked layers. It follows the assumption that only building a stack of identity mappings will achieve similar results as a shallow structure. Thus, explicitly permitting the numerous stacked layers to fit a residual mapping is effective than directly fit a required fundamental mapping.

Most of the CNN architectures simply stack the convolutional layers to develop a deeper model for improvement in accuracy. The deeper model can compute richer and more representative keypoints and works well on unknown samples. However, this increases the computational overhead and learning error.

InceptionV3 [22] is proposed to improve performance in terms of both computational cost and accuracy. They are based on sparsely connected network architecture. A single inception block performs convolution on an input using multiple filter sizes (i.e., 1x1, 3x3, and 5x5) in the same layer and concatenates all output into a single output vector forming the input for the next stage. The 1x1 filter is used to reduce the dimensionality before applying another layer, hence reduces the computation cost and the number of parameters. 4096

XceptionNet [23] is an improved form of the Inception-V3 model. In the XceptionNet model, the depthwise separable convolution is introduced instead of using inception modules that are followed by pointwise convolutional layers. A stack of depthwise separable convolution layers with residual connection is applied independently over each input data channel. The pointwise convolutional layer (filter size of 1x1) projects the channel output through a depth-wise convolution into a new channel space.

InceptionResV2 [24] incorporates the benefits of both the Inception and ResNet models. The inception model better learns the features at different resolutions within the same convolution layer while the ResNet model supports deeper CNN for learning feature without compromising the performance.

DenseNet-169 [25] is extension to ResNet architecture. As the depth of the network increases, it suffers from vanishing gradient issues in the training procedure. Both ResNet and DenseNet models are designed to resolve this problem. The DenseNet architecture is based on all layer connections where each layer receives input from all the earlier layers and passes its output to every layer ahead. Thus, the resulting connections are dense that improves the accuracy with relatively fewer parameters.

MobileNet [26] is a lightweight architecture designed for embedded or mobile devices. It employs depth-wise separable convolutional layers to minimize both the network size and convolution cost of the framework. MobileNetv2 is an extended version of MobileNetv1, which introduced two new features i) linear bottleneck between layers and ii) inverted residual blocks where the shortcut connection is inserted between bottleneck layers.

NasNet-Mobile [28] is a scalable CNN architecture, designed automatically by using NAS (Network Architecture Search) algorithm searching through a space of neural network configurations. The structure comprises basic building blocks (cells) that are enhanced via utilizing reinforcement learning. A cell contains fewer methods (many independent convolutions and pooling) and is reiterated several times in accordance with the needed size of the framework.

EfficientNet [27] uses mobile inverted bottleneck convolution similar to MobileNetV2. The baseline network has been optimized by NAS algorithms using the AutoML MNAS framework, which causes to generates a more smart and efficient framework. This network has outperformed previous state-of-the-art approaches in terms of accuracy and efficiency on the ImageNet dataset [38] with a small number of parameters.

iii) Prediction

Finally, the features obtained from DL models are classified as real or fake. We have used an SVM classifier for feature classification. The SVM classifier generates hyperplanes to locate the decision boundary for classification. The reason to use the SVM classifier is that it can efficiently deal with the problem of the curse of dimensionality as compared to other classifiers e.g., Nave Bayes, KNN. Further, it minimizes the amount of empirical error besides maintaining the complexity level of the mapping function. The main motivation to employ SVM for deepfakes classification is its robustness and capability to deal with the over-fitted training data. These abilities of SVM allow it to accurately generalize its prediction behavior and perform well for unseen data samples. We utilized obtained features to train the SVM classifier and classify each input sample into two classes, i.e., real and fake. The training data comprises of N visual feature vectors organized as: (x(i), y(i)), i=1,.N, where y(i) ∈ {1, −1} shows the real and fake classes. For each feature vector x(i), SVM generates a hyperplane that linearly separates the two classes as:

$$w^T.x^{(i)} + \beta \geq 1 \; if \; y^{(i)} = +1 \tag{1}$$

$$w^T.x^{(i)} + \beta < 1 \; if \; y^{(i)} = -1 \tag{2}$$

Where w is the weight vector and β is the bias. The objective is to maximize the distance between two support vectors by minimizing the norm ||w|| which can be defined as a quadratic optimization problem as shown in Eq. (3):

$$\min \|w\|, \text{ such that } y^{(i)}\left(w^T.x^{(i)} + \beta\right) \geq 1 \tag{3}$$

The two classes (real and fake) can be determined by using the discriminant function f(x) = sign (wT. x(i) + β) as follows:

$$\begin{cases} \text{real}, f(x^{(i)}) = +1, \\ \text{fake}, f(x^{(i)}) = -1 \end{cases} \tag{4}$$

III. Experiments and Results

A) Dataset:

We tested the proposed model on the DFDC database released by Facebook and is publicly accessible on Kaggle competition [30]. The dataset is produced by employing two unfamiliar AI approaches. It includes 19,000 original videos and 100,000 fake samples. The provided dataset is divided randomly into 70-30 parts. We utilize 70% data for training while the remaining 30% data for evaluation.

B) Evaluation Matrices

The obtained results are quantitatively evaluated using parameters such as precision (P), recall (R), accuracy (Acc), true positive rate (TPR), and F1-score. We computed these parameters as follows:

$$P = \frac{\alpha}{(\alpha + \gamma)} \tag{5}$$

$$R = \frac{\alpha}{(\alpha + \eta)} \tag{6}$$

$$Acc = \frac{(\alpha + \beta)}{(\alpha + \beta \; + \gamma + \eta)} \tag{7}$$

$$F1 - score = \frac{2PR}{\mathrm{P} + \mathrm{R}} \tag{8}$$

where α= true positive, β=true negative denotes the number of positive (fake) and negative (real) samples respectively, that are classified correctly. The γ= false positive and η= false negative denotes the number of negative (real) and positive (fake) samples respectively, that are misclassified.

C) Implementation details

All networks are implemented using Python with TensorFlow and run on Nvidia GTX1070 GPU based system. Moreover, the SVM classifier is trained using various feature extraction networks and applied to locate fakes from the DFDC dataset with 60 epochs and a 0.001 learning rate.

D) Results and Discussion

Here, we have discussed the evaluation results of SVM with varying feature extractors i.e. VGG-16 [20], VGG-19 [20], ResNet101 [21], Inception V3 [22], XceptionNet [23], InceptionResV2 [24], DenseNet-169 [25], MobileNetv2 [26], EfficientNet [27] and NASNet-Mobile [28]. To perform the training of the SVM classifier, we have employed about 8000 image samples, while for system testing, we utilized 3000 samples. Fig. 3 presented the results obtained from the SVM classifier with all feature extractors in terms of evaluation metrics.

Fig. 3a exhibits the performance analysis in terms of accuracies values for all feature calculators. Among all the employed models, the DenseNet-169 obtained the highest accuracy with the value of 98% while the XceptionNet exhibits the second highest accuracy value of 97.2%. VGG-16 model shows the lowest accuracy with a value of 89%.

In recognizing visual manipulations, incorrectly detecting a real sample as Deepfakes is less costly than misclassifying a deepfake sample as the original. Therefore, the objective of deepfakes detection techniques is to minimize the occurrence of false negatives, hence, optimizing recall is the main priority. Fig. 3c demonstrates the performance comparisons of all techniques in terms of obtained values of recall. The DenseNet-169 framework shows the highest recall value of 97.6%, while InceptionResV2 achieves the second-highest recall value of 96.4%. In the case of recall evaluation matric, VGG-19 shows the least value of 85.6%.

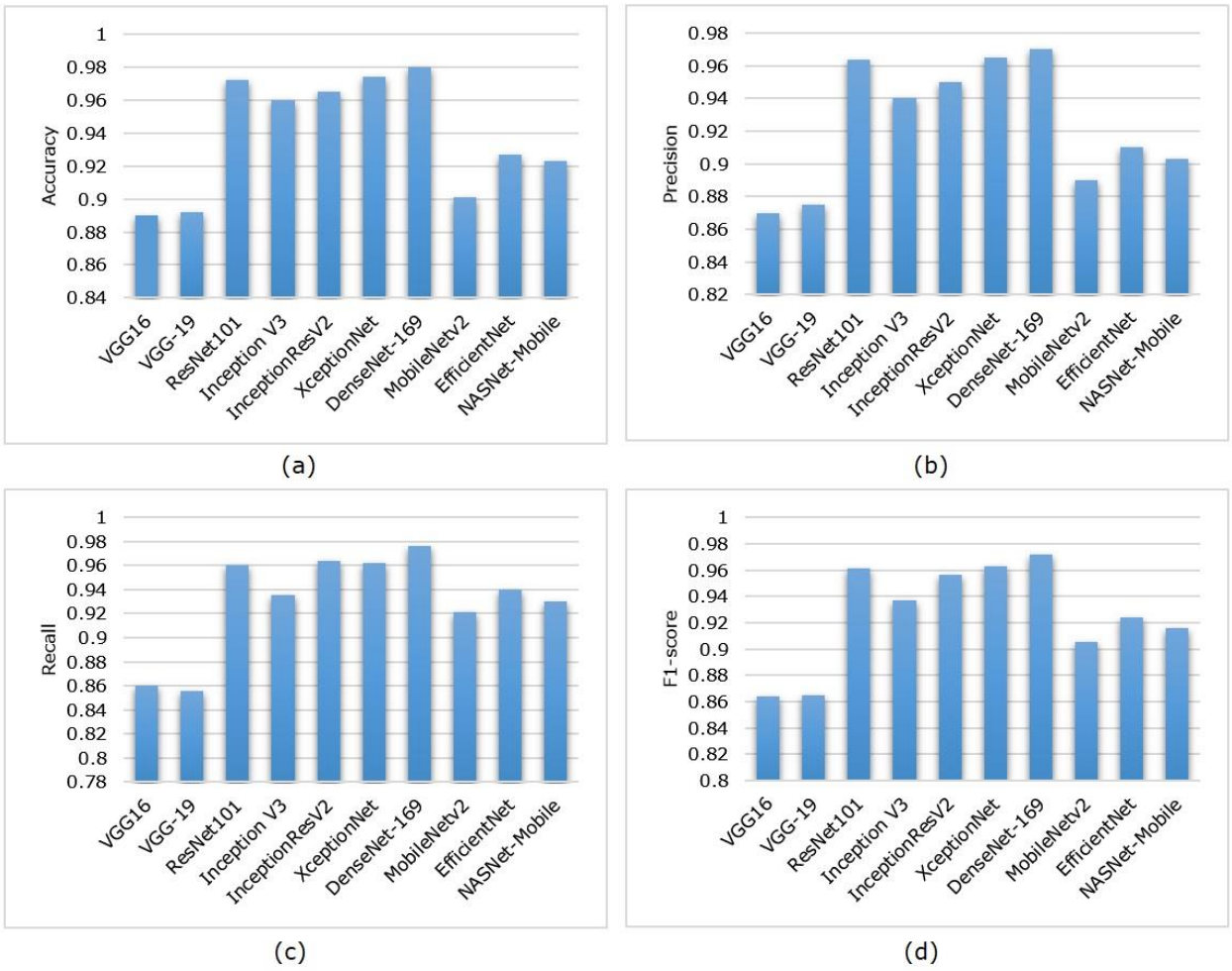

Fig. 3 Performance comparion of varios feature extraction models using different parameters (a) Accuracy (b) Precision (c) Recall (d) F1-score

After reducing the occurrence of false negatives, optimizing false positives is mandatory to minimize misclassification of original content as forged. Therefore, exhibiting robust precision is another main goal of deepfakes detection models. The obtained precision values of all ten models are presented in Fig. 3b. DenseNet-169 attains the highest precision of 97%, while VGG-16 obtains the lowest precision value of 87%.

The F1-Score gives an overall analysis of the robustness of the classifier. The more the value of F1-sore, the better will be the performance of the classifier. Fig. 3d represented the obtained F1-score values for all ten feature extractors with the SVM classifier. The DenseNet-169 framework shows the highest F1-score value of 97.2%, while the VGG-16 shows the lowest F1-score value of 86.4%.

From Fig. 3, it can be visualized that the DenseNet-169 framework along with the SVM classifier exhibits robust performance on all evaluation parameters for classifying the deepfakes samples. Therefore, DenseNet-169 is highly preferred for deepfakes detection.

IV. Conclusion

The advancement of GAN has led to the creation of convincing deepfakes which is resulting in a serious threat to the privacy and security of individual data. This works aims to provide an effective end-to-end framework, which can be employed as a benchmark framework for the research community that wants to work in the area of visual manipulation detection. The work also demonstrates the concept of using transfer learning to compute the representative set of features from the suspected samples. Several performance evaluation metrics i.e. accuracy, precision, recall, and F1-score, have been utilized in this work to perform an in-depth analysis of how various DL-based feature extractors work. After comparison, it is concluded that DenseNet-169 together with the SVM classifier performs well than the rest of the approaches. However, it is also a critical analysis that the remaining networks are not far behind the DenseNet-169, therefore, there is a possibility that the other frameworks can be further tweaked with different classifiers to show more accurate classification results.

References

[1] *ZAO*. Accessed on: September 09, 2020. Available: https://apps.apple.com/cn/app/zao/id1465199127.
[2] *Reface App*. Accessed on: September 11, 2020. Available: https://reface.app/
[3] *FaceApp*. Accessed on: September 17, 2020. Available: https://www.faceapp.com/
[4] *Audacity*. Accessed on: September 09, 2020. Available: https://www.audacityteam.org/

[5] B. Uga, "Towards Trustworthy AI: A proposed set of design guidelines for understandable, trustworthy and actionable AI," ed, 2019.
[6] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, "Deep Learning for Deepfakes Creation and Detection," *arXiv preprint arXiv:1909.11573,* 2019.
[7] R. Chesney and D. Citron, "Deepfakes and the New Disinformation War: The Coming Age of Post-Truth Geopolitics," *Foreign Aff.,* vol. 98, p. 147, 2019.
[8] S. D. Olabarriaga and A. W. Smeulders, "Interaction in the segmentation of medical images: A survey," *Medical image analysis,* vol. 5, no. 2, pp. 127-142, 2001.
[9] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8261-8265: IEEE.
[10] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *arXiv preprint arXiv:1812.08685,* 2018.
[11] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815-823.
[12] A. J. O'Toole *et al.*, "Face recognition algorithms surpass humans matching faces over changes in illumination," vol. 29, no. 9, pp. 1642-1646, 2007.
[13] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting World Leaders Against Deep Fakes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 38-45.
[14] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1-10: IEEE.
[15] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking," *arXiv preprint arXiv:1806.02877,* 2018.
[16] L. Nataraj *et al.*, "Detecting GAN generated fake images using co-occurrence matrices," *arXiv preprint arXiv:1903.06836,* 2019.
[17] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos," *Interfaces (GUI),* vol. 3, p. 1, 2019.
[18] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: a compact facial video forgery detection network," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018, pp. 1-7: IEEE.
[19] H. 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.
[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556.,* 2014.
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.
[23] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258.
[24] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, vol. 31, no. 1.
[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.
[26] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:.04861,* 2017.
[27] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105-6114: PMLR.
[28] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697-8710.
[29] A. Fydanaki and Z. Geradts, "Evaluating OpenFace: an open-source automatic facial comparison algorithm for forensics," *Forensic sciences research,* vol. 3, no. 3, pp. 202-209, 2018.
[30] B. Dolhansky *et al.*, "The DeepFake Detection Challenge Dataset," *arXiv preprint arXiv:2006.07397,* 2020.