Exposing DeepFakes Using Convolutional Neural Networks and Transfer Learning Approaches

Shraddha Suratkar
COE - CNDS
Veermata Jijabai Technological Institute
Mumbai
sssuratkar@ce.vjti.ac.in

Faruk Kazi
COE - CNDS
Veermata Jijabai Technological Institute
Mumbai
fskazi@el.vjti.ac.in

Mukul Sakhalkar
COE - CNDS
Veermata Jijabai Technological Institute
Mumbai
mukul.sakhalkar@gmail.com

Nikhil Abhyankar
COE - CNDS
Veermata Jijabai Technological Institute
Mumbai
nikhil293@gmail.com

Mrunal Kshirsagar
COE - CNDS
Veermata Jijabai Technological Institute
Mumbai
kshirsagar.mrunal15@gmail.com

2020 IEEE 17th India Council International Conference (INDICON) | 978-1-7281-6916-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/INDICON49873.2020.9342252

***Abstract*—Advancements in Artificial Intelligence – oriented computing power and the ever-growing reach of social media have proven to be catalysts in emergence and spread of a new vein of AI generated fake videos known as 'DeepFake' videos. Such videos are synthesized using generative machine learning models like Generative Adversarial Networks or Variational AutoEncoders and they can achieve high degrees of realism. Spread of sensitive political or obscene content in form of such videos may lead to social distress to the target entity(s). This paper presents a study pertinent to the detection of DeepFake videos using Convolutional Neural Networks (CNNs) with transfer learning. A comparative study of the performance of various models in the detection of tampered videos has been presented. These models are trained (fine-tuned) and tested on a custom dataset encompassing randomly selected labelled frames from videos in the DeepFake Detection Dataset by Google AI and FaceForensics++ dataset.**

***Keywords*—Convolutional Neural Networks, Generative Adversarial Networks, Transfer Learning, Visual Geometry Group, DenseNet, Xception, Inception V3.**

I. Introduction

Creation of fake photos has been around for over a century, even since the time of absence of digital tools for manipulation. Immense tedious human hours and technological limitations in manual reconstruction and lithography substantially hindered the proliferation of such photos. However, with eventual advancements in technology and inception of powerful post-processing tools the time required to doctor photos manually reduced to a great extent. Since it became increasingly difficult for humans to localize such manipulations, developments aimed towards leveraging Artificial Intelligence to spot the tampering [1]. Since Convolutional Neural Networks (CNNs) outperformed all classical machine learning algorithms at the ImageNet challenge [2], they were used to detect such images [3], [4].

The more recent problem faced is on the grounds of using computer intelligence in the form of 'Generative Models' for the creation of such counterfeit content. These include the Generative Adversarial Networks (GANs) [5] and Variational Autoencoders (VAEs) [6], which have led to the latest phenomenon of the emergence of a class of faux videos called 'DeepFake' videos. 'DeepFake' is a portmanteau of the terms 'deep learning' and 'fake' [7]. This technology can be used in malignant ways, manipulating people into believing something is real when it is not. Generative models can produce persuasive counterfeits by training over photos and videos of a target person, and then emulating their behaviour and speech patterns [8], [9]. The field of Computer Vision has vastly leveraged generative models built using to solve various problems [10], [11].

Recent prominent occurrences of DeepFake videos have targeted former U.S. President Barack Obama [12], [13] and Facebook CEO Mark Zuckerberg making them say absurd things.

The commercialization of this technology can be instantiated by applications like FaceApp, the Chinese DeepFake app Zao, DeepNude as well as in acting for face swapping in movies like Solo: A Star Wars Story wherein Han Solo's face was replaced with Harrison Ford's face. Major video platforms like Reddit, YouTube, Twitter, Discord, etc. are banning such doctored content and updating their terms of use after accrual in occurrences of circulation of DeepFake videos. Thus, measures to curb the spread of such videos are imperative to prevent a gamut of problems from fake news and extortion to erosion of trust from videos as evidence in court. [13], [14].

The the models used in the proposed method have been fine-tuned and tested on a custom-made dataset to give results comparable to the contemporary technologies used. Posterior layers of these pre-trained models have been modified corresponding to the output which is a binary decision ('real'/ 'fake'). A set of real and DeepFake videos was shortlisted from the variety of samples available in the huge labelled repository of Google AI [15] and the FaceForensics++ dataset [16]. Since the primary focus of study has been the face, the videos were first decomposed into frames and later a frame by frame extraction of the facial zone has been achieved by using Dlib library [17]. The extracted faces that were that were labelled as 'real' or 'fake', served as the training set.

The ineffectiveness of generative models manifests in the form of some aberrations in the image (frame) [13] pertinent to, sharpness, pixel consistency, and pixel variance while trying to achieve a face swap. The Deep CNNs can effectively capture these inconsistencies in the deeper layers.

A. Generative Adversarial Networks (GANs)

GANs used for the synthesis of 'DeepFakes' were introduced in 2014 by Goodfellow et al [5]. As shown in Fig.1, GANs comprise of a discriminator (D) and a generator (G) which are deep neural networks. G tries to fool D by generating samples as close to the real data-distribution as possible. D, which is usually a binary classifier, detects these as 'real' or 'fake'. The initially generated samples mainly comprise of noise. With progressing iterations, G tunes its parameters based on the backpropagated loss from D. This 'min-max' optimization aims at maximizing the probability of generated samples being from the original distribution as well as minimizing the classification loss. A convincing 'DeepFake' video can thus be generated after multiple iterations of detection, back-propagation and forgery.

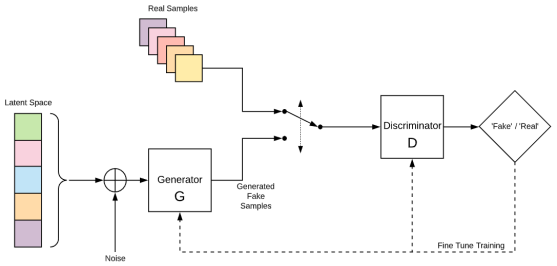


Fig. 1. Architecture of GANs [5]

The samples produced by a GAN can be made to achieve very high degrees of realism by starting to train the GAN from lower resolutions and increasing its depth as training progresses to capture finer higher resolution details [18]. Thus, GANs have a huge potential for both good and evil [19]–[22] because they can be effectively trained to closely model any data distribution.

B. Transfer Learning

Traditional machine learning approaches the task of training in an isolated manner; there is neither retention of knowledge nor consideration of existing information in learning. Transfer learning pertains to generalizing information in a particular setting, using what has previously been learned in another. Since Transfer Learning enables the use of prior information it often gives better results even in situations of scarce training data.

Statistically, transfer learning is applicable in a scenario with two domains, each having a task defined over it [23]. A Domain is usually a bipartite entity represented as $D = \{X,\ P(X)\}$. Here X is the feature space and $P(X)$ indicates the marginal probability, where $X = \{x_1, x_2, \ldots x_n\}$, $x_i \in X$. Similarly, a task can be defined as a duple of the label space Y, and the objective function η which relates the feature space with the label space. The predictive function η is learned from pairs of feature vector/label, (x_i, y_i) where, $x_i \in X$, $y_i \in Y$ which entails that $\eta(x_i) = y_i$. Thus, a task is mathematically expressed as $T = \{Y, P(Y|X)\} = \{Y, \eta\}$ where $Y = \{y_1, y_2, \ldots y_n\}$, $y_i \in Y$. Two such domains form part of the definition where information learned in domain D_s (source domain), having task T_s (source task) defined over it, is used in domain D_T (target domain) with corresponding task T_T (target task). Thus, Transfer Learning enables us to grasp the conditional distribution $P(Y_T|X_T)$ in D_T based on what is learned in D_S and T_S where $D_S \neq D_T$ or $T_S \neq T_T$. The training and testing data need not necessarily have same feature space distribution. As a result of insufficient training data, models can be

trained on one dataset and fine-tuned and tested on another.

Related studies suggest that a combination of CNNs and Transfer Learning can achieve higher accuracy over a shorter span of training time [24].

II. Related Work

Generative networks tend to lose details of the face while swapping it onto the target video. The concurrent works in the field of detection of 'DeepFakes' using machine learning models mostly narrow down to two approaches. Both approaches target inconsistencies in the fake video which are caused due to shortcomings of generative networks.

A. Temporal Inaccuracies

Inter-frame temporal contiguity of facial features are targetted under this approach. Recurrent convolutional models have been effective at exploiting temporal information from an image sequence as put forth by [25], thus making it possible to localize tampered zones. Guera and Delp [26] pipelined a CNN (for feature extraction) followed by a LSTM (for sequence processing) in order to target anomalies. A similar attempt was made by using a model combining CNNs with a RNN in order to exploit the irregularities in the process of eye blinking. To distinguish between pristine and manipulated videos, Li et al. [27] initially decomposed the videos into frames. After aligning the faces, facial zone and ocular areas were extracted using six landmarks as references. Then the bounding boxes of eye were scaled, hence creating new cropped sequences which were fed to a LRCN (Long Term Recurrent Convolutional Network) [28]. Promising results were obtained by resorting to this approach.

B. Visual Artifacts

Unlike in the temporal inaccuracies approach, this method focuses on a single frame rather than a sequence of them. Generative networks tend to lose the pixel variance in the process of affine warping of the facial mask in the generation stage of a fake video. The re-sampling involved in reconstruction has a blurring effect, which is a by-product of the generative models being trained on L2 loss. The inconsistency in sharpness between the facial mask and surrounding is detected using various pre-trained CNN models. Several methods for detection targeting face warping artifacts were proposed [29], [30]. An attempt to exploit inconsistencies in the head poses [31] was also made by suggesting two methods involving the MesoNet model [32], which forms the basis of the proposed method. [32]

III. Experiments

A. Dataset

For this work, a custom dataset has been created by referring to the FaceForensics++ [16] dataset and the DeepFake Detection Dataset by Google AI and Jigsaw [15]. Videos in the FaceForensics++ dataset have been generated using various techniques like Neural Textures [33], Face2Face [34] and Face Swap [35] techniques. Table.I provides information regarding the number of frames per dataset taken into account to make the custom dataset. Facial zones in these frames were extracted by implementing a face detection pipeline using the Dlib toolkit [17].

TABLE I
Preparation of Dataset

Dataset	Type	Training Set	Validation Set	Test Set
Google AI	Fake	203313	71898	33213
	Real	146618	82162	45651
Face Forensics++	Fake	42063	25192	16716
	Real	110387	38864	19638

B. Models Used

1) Inception V3: The Inception architecture [36] achieves computational efficiency owing to use of fewer parameters. The Fig. 2 shows the architecture implemented in the proposed system for detection.

INCEPTION V3 ARCHITECTURE	
Input (299 x 299 RGB image)	
Conv 32, 3 x 3, stride = 2	
Conv 32, 3 x 3	
Padded Conv 64, 3 x 3	
Max pooling 3 x 3, stride 2	
Conv 80, 1 x 1	
Conv 192, 3 x 3	
Inception Module A	X 3
Inception Module B	X 5
Inception Module C	X 2
Global Average Pooling 8 x 8	
Sigmoid	

Fig. 2. Inception V3 Architecture

The 42 layered network with lesser number of parameters achieves complexity comparable to the VGGNet, with a lower rate of error. The Inception V3 factorizes the convolutions in order to reduce connections without hampering efficiency. Factorisation consists of two types, with the upper layers making use of factorization with smaller convolutions (In Fig. 3a, two layers with 3x3 filters used over a layer with a

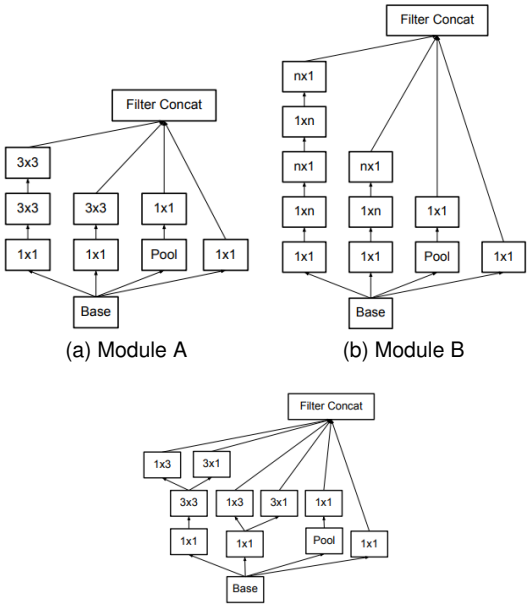


Fig. 3. Inception V3 Modules [36]

5x5 filter) and the lower architecture(Fig.3b and Fig. 3c) having asymmetric factorization.

2) VGG: The VGG model showcases the relation between accuracy and the depth of the network in case of large-scale image recognition.

VGG16 ARCHITECTURE
Input (224 x 224 RGB image)
Conv 64, 3 x 3 Conv 64, 3 x 3
Max Pool 2 x 2, stride = 2
Conv 128, 3 x 3 Conv 128, 3 x 3
Max Pool 2 x 2, stride = 2
Conv 256, 3 x 3 Conv 256, 3 x 3 Conv 256, 3 x 3
Max Pool 2 x 2, stride = 2
Conv 512, 3 x 3 Conv 512, 3 x 3 Conv 512, 3 x 3
Max Pool 2 x 2, stride = 2
Conv 512, 3 x 3 Conv 512, 3 x 3 Conv 512, 3 x 3
Max Pool 2 x 2, stride = 2
Global Average Pooling 7 x 7
Sigmoid

Fig. 4. VGG Architecture

The depth of a VGG model can be increased, without hurting the performance, owing to a small filter size (3x3) [4]. The modified VGG model used has been summarized in Fig.4. VGG-16 architecture consists of five blocks of Convolution. Similar to the previous case, staring from low level features at the shallow layers, as one goes deeper into the network, more complex features (entire facial region) are learned by the model.

3) Xception: The Xception [37] architecture as seen in Fig.5 is inspired by the Inception model [36] in which certain modules are substituted with depth-wise separable convolutions. The model is a translational step between regular and depth-wise separable convolutions. Xception has performed better than Inception V3 in the ImageNet challenge [37]. This is due to the efficient use of model parameters in Xception over Inception V3.

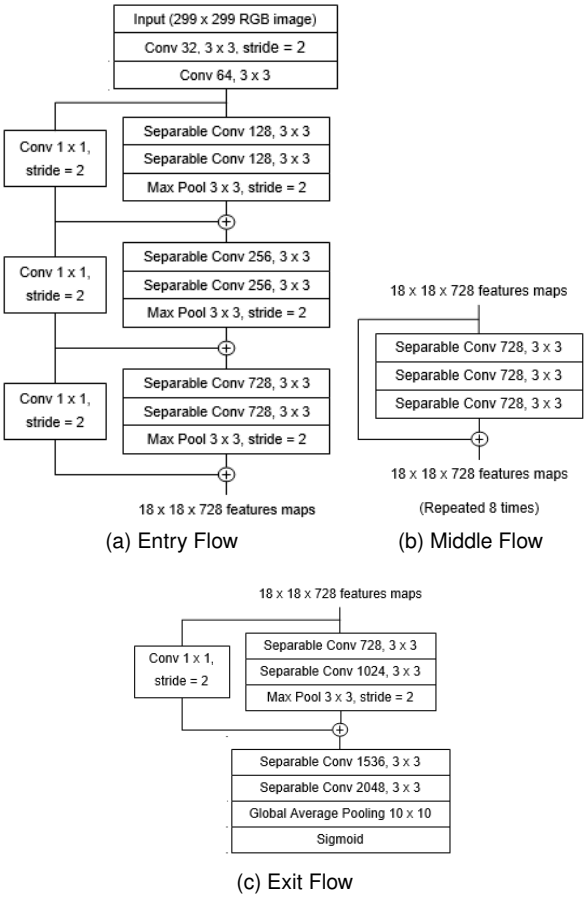


Fig. 5. Xception Architecture

The architecture can be divided into three sections: Entry flow 5a, Middle flow 5b, and Exit flow 5c. The learning which occurs within these three sections proceeds to the Optional Fully Connected Network. The output layer then uses sigmoid activation to map the input feature vectors to a probability used to distinguish between real and fake.

4) DenseNet: The DenseNet model connects every layer to every subsequent layer in a feed-forward manner. For a given layer, all of the preceding layers form its inputs and the layer works as an input for

the all of the following layers. The primary purpose is that having direct connections between the shallower and the deeper layers counters the vanishing gradient problem. DenseNet has $N(N+1)/2$ direct connections, unlike traditional networks which have a total of N connections. Thus, DenseNet reuses the features while reducing the number of trainable parameters [38]. Fig. 6 summarizes the architecture implemented in the proposed approach. The feature maps flowing in from all previous layers are concatenated and not summed.

Let f_m represent feature maps at the *m*-th layer. Hyperparameter f is also called a growth rate.

$$f_m = f_0 + f * (m-1) \tag{1}$$

DENSENET 121 ARCHITECTURE	
Input (224 x 224 RGB image)	
Conv 64, 7 x 7, stride = 2	
Max Pool 3 x 3, stride = 2	
Conv 128, 1 x 1 Conv 32, 3 x 3	X 6
Conv 1 x 1	
Average Pool 2 x 2, stride = 2	
Conv 128, 1 x 1 Conv 32, 3 x 3	X 12
Conv 1 x 1	
Average Pool 2 x 2, stride = 2	
Conv 128, 1 x 1 Conv 32, 3 x 3	X 24
Conv 1 x 1	
Average Pool 2 x 2, stride = 2	
Conv 128, 1 x 1 Conv 32, 3 x 3	X 16
Global Average Pooling 7 x 7	
Sigmoid	

Fig. 6. DenseNet Architecture

From Fig.6, blocks from the Convolution layer up to Dense Block4 are responsible for extracting facial features of real and fake images. Fully connected layers classify these images into either 'real' or 'fake' depending upon the features learned by Convolutional Layers. Layers closer to input learn to detect more general, low level features like edges, corners or curves which are already learnt in pre-trained Imagenet parameters. As a result, for fine-tuning by using the transfer learning approach, parameters of these layers do not need to be updated.

C. Training

The objective is to achieve binary classification of the data into two classes, 'Fake' and 'Real'. Each model has been trained over five epochs, calculating the loss in the form of Binary Cross Entropy (BCE).

$$BCE = -\sum_{i}^{c'=2} y_i log(\hat{y}_i) \tag{2}$$

$$= -y_1 log(\hat{y}_1) - ((1-y_1) log(1-\hat{y}_1)) \tag{3}$$

where 'y_i' represents actual output vector; '$\hat{y}_i$' represents the predicted output vector; 'c' represents classes.

Stochastic Gradient Descent (SGD) with momentum is used for optimization and the parameters are updated with the prediction loss of every input sample. Momentum prevents large fluctuations in the loss by utilizing exponentially weighted moving averages. Gradient jumping in SGD with momentum is damped as compared to Batch Gradient Descent and the amount of damping is dependent on the value of β. The standard value of hyper parameter β (0.9) has proved to be superior in the proposed approach.

$$V_t = \beta V_{t-1} + (1-\beta) \nabla_W L(W, X, y) \tag{4}$$

$$W = W - \alpha V_t \tag{5}$$

Higher accuracy results could be obtained after the first epoch because of utilizing pre-trained ImageNet weights instead of random initialization. The architectures have been tweaked by excluding the top layer and using average pooling with an added dense layer as the output layer.

For the confusion matrices in sub-section III-E, the classification threshold probability is set to 0.5; samples with probability less than 0.5 represent the 'real' class and those greater than 0.5 represent the 'fake' class. Learning Rate Scheduler, Early Stopping and Data Augmentation techniques like re-scaling, resizing, rotation, flipping, and addition of random noise [39] were employed to prevent data over-fitting.

Implementation of architectures DenseNet121, Xception required two GPUs, whereas VGG16 required a single GPU. Training process has been implemented using Tensorflow in Python 3.6 on the Nvidia DGX-1 AI supercomputer.

D. Performance Evaluation Metrics

1) Normalised Confusion Matrices: It is a tabular layout enabling the visualisation of the performance of the applied algorithm, representing binary classification. (Fig.7)

2) Accuracy: Accuracy is the measure of a model to correctly detect the negative and positive class. It is calculated by dividing the sum of the True Negatives (TN) and True Positives (TP) with the aggregate sum of all parameters shown in Fig. 7.

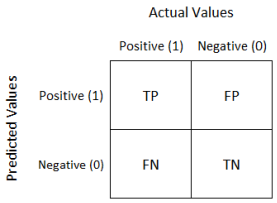


Fig. 7. Confusion Matrix

3) Area Under Curve (AUC): AUC is the area under the ROC curve along the x-axis. The probability of the classifier grading a arbitrarily selected positive occurrence higher than a arbitrarily selected negative occurrence.
The AUC is given by:
TPR(S): S $\rightarrow y(z)$: True Positive Rate
FPR(S): S $\rightarrow z$: False Positive Rate

$$Area = \int_{z=0}^{1} TPR(FPR^{-1}(z))dz \quad (6)$$

$$= \int_{-\infty}^{\infty} TPR(S)FPR'(S)dS \quad (7)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(S' > S)f_1(S')f_0(S)dS'dS \quad (8)$$

$$= P(Z_1 > Z_0) \quad (9)$$

Where Z_0 represents the score for a negative occurrence, and Z_1 shows the score for a positive occurrence and f_0 and f_1 are probability densities.

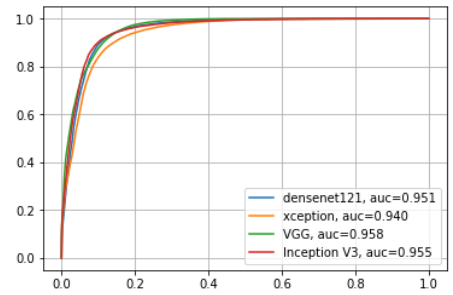
E. Results

TABLE II
ACCURACY AFTER THE FIFTH EPOCH

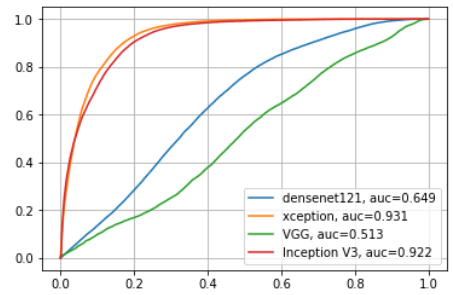
Models	Validation	Test
DenseNet121	0.8909	0.8966
Xception	0.8786	0.8453
VGG-16	0.8520	0.8916
Inception V3	0.8921	0.8750

With DenseNet 121, Inception V3 and Xception, a validation accuracy over 87% has been achieved while VGG achieved 85% validation accuracy in one epoch. After five epochs, a sweet spot of low loss and high accuracy is attained (Table. II).

Fig.8 represents the Receiver Operating Characteristic (ROC) curve which shows the performance of every model on the test dataset. Using Transfer Learning, DenseNet, Xception, VGG and Inception V3 achieved an AUC score of 0.951, 0.940, 0.958 and 0.955 (Fig. 8a). In contrast, AUC scores 0.649, 0.931, 0.513 and 0.922 were achieved by the respective models without Transfer Learning. (Fig. 8b)



(a) Transfer Learning



(b) Non Transfer Learning

Fig. 8. ROC Curves

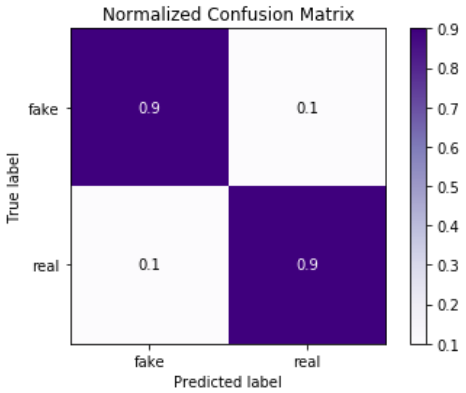


Fig. 9. DenseNet

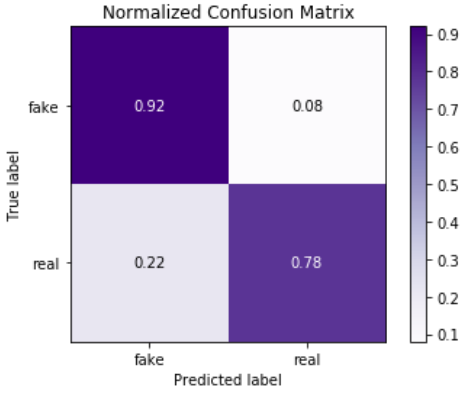


Fig. 10. Xception

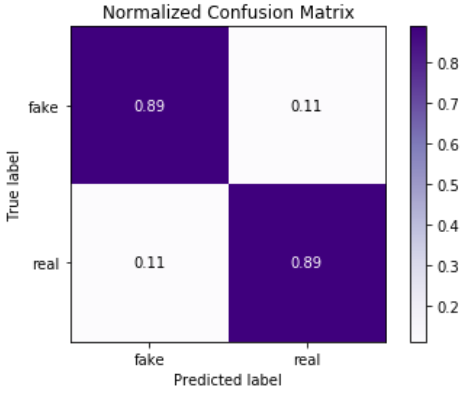


Fig. 11. VGG

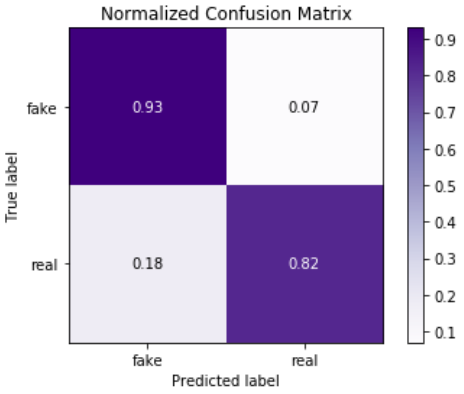


Fig. 12. Inception V3

The Normalized Confusion Matrices in Fig.9, Fig.10, Fig.11 and Fig.12 show the performances on the test dataset. The DenseNet and the VGG models achieved a high true positive and true negative score thereby minimizing errors of Type I error (rejection of a true null hypothesis (false positive)) and Type II error (non-rejection of a false null hypothesis (false negative)). It is thus important to minimize either or both of these errors. The Xception model, however, could not classify real images in the test dataset well enough as it showed an accuracy of 78%.

As a part of the comparative study, the method adopted is juxtaposed with other approaches undertaken to counter DeepFakes to present results indicating its performance. Table. III provides a synopsis of various methods in terms of their training datasets, metrics and performances.

IV. Conclusion

This paper presents an approach using transfer learning to detect DeepFake videos. The empirical results obtained by using pretrained weights and simple image augmentation techniques as precursors to training are comparable to existing approaches. Initializing pretrained weights for shallow layers of Deep CNNs yield superior results in shorter training spans than CNN models trained on randomly initialized weights. Thus, generic Deep Convolution Networks used with transfer learning have a lot of potential to correctly spot manipulated videos.

TABLE III
Comparative Study With Other Existing DeepFake Detection Models

Methods	**Dataset**	**Metrics**	**Results**	
LRCN [27]	CEW + EBV	AUC	0.99	
LogReg [29]	FaceForensics - Face2Face	AUC	0.866	
SVM classifier [31]	UADFV + DARPA GAN	AUC	0.890	
Proposed Approach				
VGG-16	FaceForensics++ + Google AI	AUC	**T.L**[1] 0.958	**Non - TL**[2] 0.513
DenseNet121	FaceForensics++ + Google AI	AUC	0.951	0.649
XceptionNet	FaceForensics++ + Google AI	AUC	0.940	0.931
Inception V3	FaceForensics++ + Google AI	AUC	0.955	0.922

[1]Transfer Learning [2]Non - Transfer Learning

The robustness of the proposed system can be increased by applying ConvLstm2D (Tensorflow) layers by passing a sequence of images rather than passing a single image to the network which would target temporal inconsistencies occurring in manipulated videos (LSTM) along with feature distortion (CNN) to provide more faithful results.

Acknowledgment

The authors are thankful to the Centre of Excellence (CoE) in Complex and Nonlinear Dynamical Systems (CNDS) VJTI, under the funding of TEQIP-III for providing cutting edge technology in the form of DGX-1 AI Supercomputer supported by NVIDIA's eight V-100 Tesla GPU accelerators.

References

[1] Y. Zhang, J. Goh, L. L. Win, and V. L. Thing, "Image region forgery detection: A deep learning approach." *SG-CRC*, vol. 2016, pp. 1–11, 2016.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[3] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2016, pp. 1–6.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[6] D. P. Kingma, M. Welling *et al.*, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[7] M. Westerlund, "The emergence of deepfake technology: A review," *Technology Innovation Management Review*, vol. 9, no. 11, 2019.

[8] 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.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[10] Y. Guo, L. Jiao, S. Wang, S. Wang, and F. Liu, "Fuzzy sparse autoencoder framework for single image per person face recognition," *IEEE transactions on cybernetics*, vol. 48, no. 8, pp. 2402–2415, 2017.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[12] A. Romano *et al.*, "Jordan peele's simulated obama psa is a double-edged warning against fake news," *Australasian Policing*, vol. 10, no. 2, p. 44, 2018.

[13] J. Hui, "How deep learning fakes videos (deepfakes) and how to detect it?" 2018.

[14] M.-H. Maras and A. Alexandrou, "Determining authenticity of video evidence in the age of artificial intelligence and in the wake of deepfake videos," *The International Journal of Evidence & Proof*, vol. 23, no. 3, pp. 255–262, 2019.

[15] N. Dufour, A. Gully, P. Karlsson, A. V. Vorbyov, T. Leung, J. Childs, and C. Bregler, "Deepfakes detection dataset by google & jigsaw."

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

[17] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1755–1758, 2009.

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[19] W. Yang, C. Hui, Z. Chen, J.-H. Xue, and Q. Liao, "Fv-gan: finger vein representation using generative adversarial networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2512–2524, 2019.

[20] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan++: Realistic image synthesis with stacked generative adversarial networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1947–1962, 2018.

[21] F. Liu, L. Jiao, and X. Tang, "Task-oriented gan for polsar image classification and clustering," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2707–2719, 2019.

[22] J. Cao, Y. Hu, B. Yu, R. He, and Z. Sun, "3d aided duet gans for multi-view face image synthesis," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2028–2042, 2019.

[23] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*. Springer, 2018, pp. 270–279.

[24] M. Huh, P. Agrawal, and A. A. Efros, "What makes imagenet good for transfer learning?" *arXiv preprint arXiv:1608.08614*, 2016.

[25] 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.

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

[27] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai created fake videos by detecting eye blinking," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018, pp. 1–7.

[28] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[29] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 2019, pp. 83–92.

[30] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," *arXiv preprint arXiv:1811.00656*, 2018.

[31] 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)*. IEEE, 2019, pp. 8261–8265.

[32] 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)*. IEEE, 2018, pp. 1–7.

[33] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[34] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.

[35] "Faceswap," 2018. [Online]. Available: https://github.com/MarekKowalski/FaceSwap/

[36] 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.

[37] 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.

[38] 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.

[39] T. Julliand, V. Nozick, and H. Talbot, "Image noise and digital image forensics," in *International Workshop on Digital Watermarking*. Springer, 2015, pp. 3–17.