SN Computer Science (2020) 1:212
https://doi.org/10.1007/s42979-020-00225-9

ORIGINAL RESEARCH

DeepFake Video Detection: A Time-Distributed Approach

Amritpal Singh[1] · Amanpreet Singh Saimbhi[1] · Navjot Singh[1] · Mamta Mittal[2]

Received: 15 May 2020 / Accepted: 10 June 2020 / Published online: 23 June 2020

Abstract
Recent developments in machine learning algorithms have led to the generation of forged videos having remarkable quality, which are indistinguishable from real videos. This can fatally affect the way in which one perceives the information available digitally. Thus, this paper aims to efficiently and holistically detect manipulated videos generated using DeepFake, which is the most effective deep learning powered technique developed so far by the researchers. Arduous efforts have been put to detect the forgery in still images, but the authors leveraged the spatio-temporal features of the videos by taking sequences of frames as input to the model. Furthermore, the authors have proposed an architecture which took advantage of lower-level features around regions of interest as well as discrepancies across multiple frames. Experiments have been performed on the Deep Fake Detection Challenge dataset of ≈ 470 GB in size, and it has been observed that the proposed approach yielded a test accuracy score of 97.6%.

Keywords DeepFake · Forgery detection · Deep Fake Detection Challenge · Artificial intelligence · Time distribution

Introduction

In this digital era, a profusion of smartphones and digital devices have led to the popularization of social media platforms. According to several reports, billions of pictures are uploaded daily on the internet. Since the rise in social media platforms, people have been interested in manipulating photos and videos, an example of such a Deepfake-generated image has been presented in Fig. 1. The spread of fake news is on the rise as well; several efforts have been made for the detection of the same [1–3].

The co-existence and co-development of computer graphics and computer vision, with the support of advanced and easily accessible hardware, have led to advancements in manipulation techniques. The aftermath of this advancement is that people with malicious purposes can easily use techniques to create fake images and videos to publish them on social networks or use them to commit white collar crimes, bypass facial authentication systems, identity theft, character assassination, disrupt a close election, create manipulated pornographic videos [4] and a threat to democracy [5]. This technique is used to substitute the face of a targeted person by the face of the source person in a video. Initially, this technique was released by a Reddit user to create manipulated pornographic videos of famous actors. Later, it was developed as a user-friendly application so that it could be accessed by anyone easily [6, 7].

DeepFake uses the concept of generative adversarial networks (GANs), in which two deep learning models compete. One model gets trained on real data and tries to create forgeries; meanwhile, the other strives to detect the forgery. The forger keeps on creating better and better fakes until the other model is unable to detect the forgery. To generate DeepFake, one has to cumulate the aligned faces of two discrete individuals X and Y, then auto-encoders E_x and E_y are trained to regenerate the faces from the dataset of the images of X and Y, respectively, as shown in Fig. 2. The magic lies in sharing the weights of the encoding part of the two auto-encoders, but their decoders are separated. After the training, any image comprising a face of X can be encoded through this shared encoder but decoded with the decoder of E_y.

This article is part of the topical collection "Deep learning approaches for data analysis: A practical perspective" guest edited by D. Jude Hemanth, Lipo Wang and Anastasia Angelopoulou.

Mamta Mittal
mittalmamta79@gmail.com

[1] Guru Gobind Singh Indraprastha University, Delhi 110078, India

[2] Department of CSE, G. B. Pant Government Engineering College, Delhi 110020, India

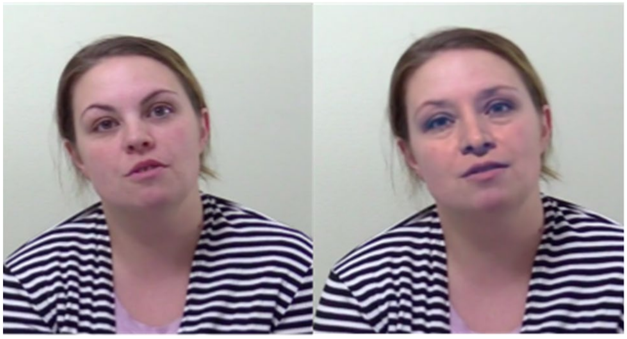


Fig. 1 A DeepFake-generated image (right) on the basis of a person (left)

The traditional state-of-the-art DeepFake detectors use convolutional neural network (CNN's) to predict from still images, thus missing the temporal features of the video. On experimentation, this has been found that temporal features can play an important role while detecting the DeepFake video because some spatio-temporal features can only be captured by a temporal model, like in DeepFake-generated videos, the eyes blink less often in comparison with the real videos. This observation could be made with our network [8].

In this paper, first and foremost, all the related works formerly present have been discussed. Subsequently, the method proposed by the authors has been illustrated meticulously. A technique is futile if it lacks the evidence to support itself. Therefore, the results and observations accumulated from the experiments have been exhibited. Conclusively, insights and ideas that are discerned have been mentioned.

Related Work

In this section, various state-of-the-art face forging creation and detection techniques have been discussed by the authors articulately. The approaches mentioned are not necessarily single folded, they could also be applied outside of their original scope. Thereafter, several datasets created by different techniques are alluded to. Face Manipulation is not a novel concept, it has been around for quite some time now. Earlier this decade, techniques that could generate videos that were indistinguishable from real videos were emerging. Dale et al. [9] introduced a technique to manipulate faces in a video with the help of a 3D multilinear model, in which they warp the source to the target face and retime the source so that the target performance is matched. Garrido et al. [10] used a novel image matching metric that merges the appearance and motion to choose candidate frames from the source video, and the user's identity is preserved as the face transfer uses a 2D warping technique.

More recent techniques have made it possible to re-enact the faces in real time like *Face2Face* technique by Thies et al. [11] in which face re-enactment is performed by taking dense photometric consistency measures while tracking the facial expressions of the source as well as targeted video. Modern methods leverage the use of deep learning techniques [12], especially GANs [13], to create forged videos. FaceSwap [14] is another technique that uses CNN to catch the appearance of the target entity from an amorphous collection of photos. The goal is to manifest an image in the style of the other by framing the

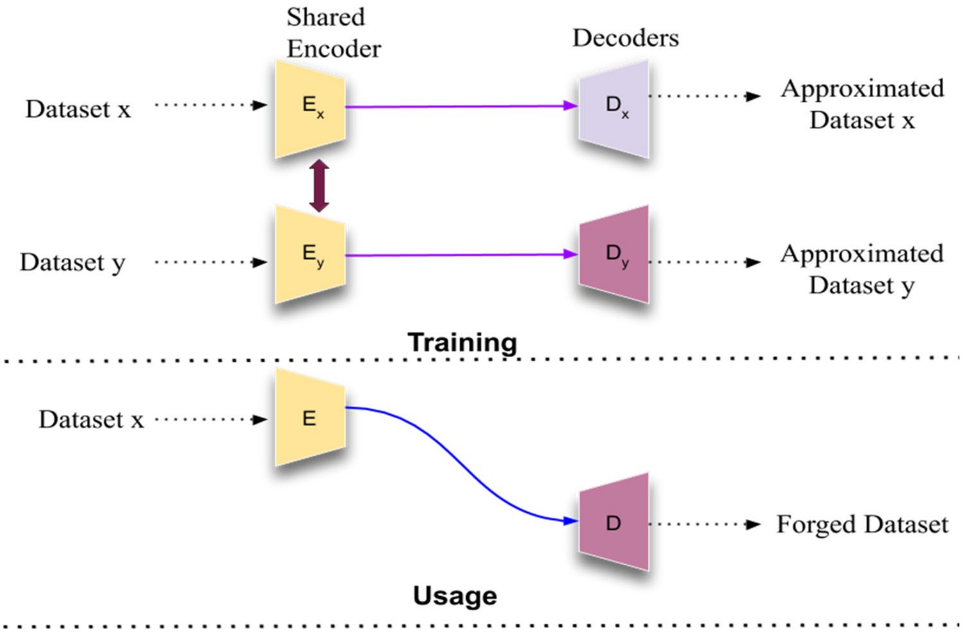


Fig. 2 A DeepFake creation model using two encoder–decoder pairs

FaceSwapping problem in terms of neural style transfer. DeepFake is one of the most widely used techniques due to the high quality of the tampered videos and easy to use, as well as accessible software, with its user-base ranging from novice to professional.

Due to state-of-the-art forging techniques, remarkable datasets came into existence. Rössler et al. [15] introduced a novel face manipulation dataset having nearly half a million edited images (from over 1000 videos). It transcended all existing video manipulation datasets by a notable order at that time. Later, Rössler et al. introduced FaceForensics++ [16], which is an extension of the previously introduced FaceForensic dataset. Recently, the Deep Fake Detection Challenge dataset [17, 18] has been made publicly available, which consists of 1,19,146 videos. The dataset is varied in several factors like gender, skin tone, age, etc. Various actors with arbitrary background have been considered, thus bringing visual volatility. Realistic GAN-generated images pose serious threats to society. Fortunately, several forgery detection techniques are on the rise due to the availability of the before mentioned datasets. Marra et al. [19] showed that each GAN leaves its specific fingerprints in the images it generates, this can play an important role while detecting forgeries. Mittal et al. [20] introduced an algorithm which uses multiple threshold approach (B-edge) for efficient edge detection. Afchar et al. [21] presented two networks, both having a low number of layers so that they could focus on the mesoscopic properties of images, which made their forgery detection technique fast and reliable. Image segmentation has been found useful for various purposes [22–24]. In addition, specific techniques utilized in the medical domain and watermarking [25–27] could be used as a reference for forgery detection.

Most of the manipulation techniques are targeted at certain domains and were not effectively applicable to other domains or new attacks. Nguyen et al. [28] introduced a capsule network that could detect various kinds of attacks with very few parameters than the traditional CNN with similar performance. Later, it became crucial to localize the manipulated regions and detect manipulated face images. Attention mechanism was proposed by Dang et al. [29] to improve the feature maps and to highlight the informative region to further improve the binary classification. Most methods are designed to detect only still forged images; therefore, it becomes strenuous to detect forged videos which have temporal characteristics [30].

Using CNNs for object localization/detection has been a customary practice. More often than not, these networks are felicitous in feature extraction from images. Some of the most widely used state-of-the-art CNNs are XceptionNet [31], InceptionV3 [32], SeNet [33], and ResNet [34] which could be used as a feature-extractor/backbone network.

Proposed Method

The authors proposed an architecture which leverages the use of spatio-temporal features to detect the DeepFake videos since several experiments demonstrate that the model is competent in selecting which spatio-temporal features are most relevant for the Deep Fake Detection Challenge (DFDC) dataset.

There is no hard and fast rule about the size of the input image as such. One can use a larger input size but at the expense of more computational power. The commonly used image sizes in the practice are 100 × 100, 128 × 128, 256 × 256, 299 × 299, 300 × 300 [35]. Keeping this in view, there is always a trade-off between the computational power and the input size; 240 × 240 images have been considered as it is an even number (easier to perform cropping and scaling) and large enough so that all the essential features can be detected. DeepFake detection problem has been framed as a binary classification problem using a CNN model wrapped in a time-distributed layer followed by a long short-term memory (LSTM) layer whose output is fed into dense layers as shown in Fig. 3. The authors aim to detect forged videos which may get unnoticed by the naked human eye but the model being spatio-temporal will be able to capture those minuscule details.

Dataset Pre-processing

The entire DFDC dataset (≈ 470 GB of data) has been used for the experimentation. The spatio-temporal approach has been utilized; thus, 30 frames per video have been considered by keeping in view the equilibrium between computational resources and a larger number of frames. After analyzing the dataset, it has been found that on an average there were 300 frames per video. DeepFake techniques manipulate mostly the face and regions around the face. Therefore, the face as the region of interest has been considered and extracted from the video as shown in Fig. 4. Video manipulation has been carried out on a frame-by-frame basis by retrieving the faces so

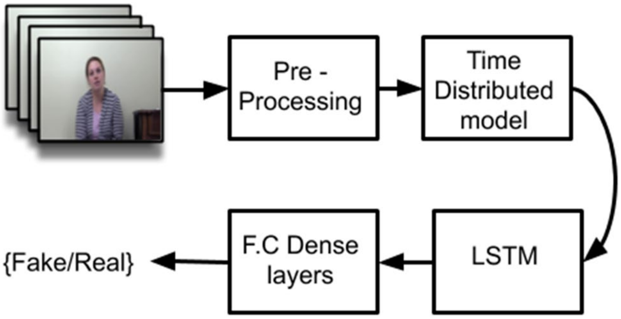


Fig. 3 Basic structure of the proposed method

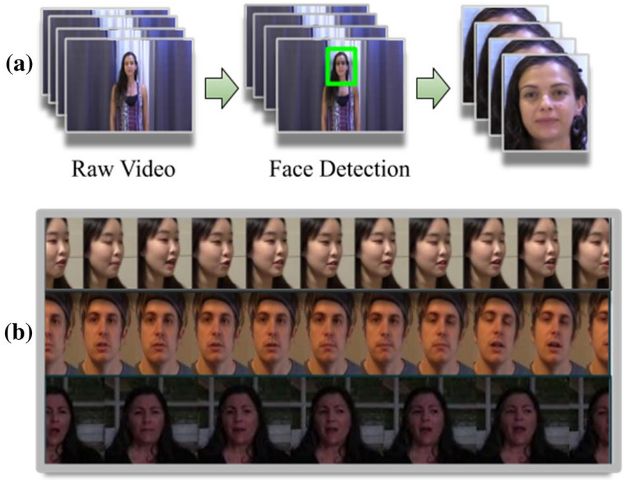


Fig. 4 **a** Pre-processing pipeline, **b** example faces in DFDC dataset

that low-level artifacts produced by face manipulation further manifest themselves as temporal artifacts with inconsistencies across frames.

Due to the humongous amount of data, there was a need for a face extractor which was fast as well as accurate. MobileNet-SSD [36, 37] provided a righteous trade-off between the two. A 35% extra margin around the faces has been added so that the distortions in that region could be detected.

Detailed Architecture

As EfficientNet uses compound scaling methods [38], that proportionately scales all dimensions of D, W and R using the manageable yet immensely effective compound coefficient. Neural Architecture Search (using the AutoML and MNAS framework) [39] is used to design a modern baseline network and to scale it up to procure a family of models called EfficientNet, which achieve much finer accuracy and efficiency than previous ConvNets. It makes it easier to optimize and create substantially efficient networks when scaling the baseline model, which results in a significant boost in performance on image classification tasks. EfficientNet wrapped in a time-distributed layer followed by an LSTM layer, extending the CNN Architecture to learn spatio-temporal features has been proposed.

Compound scaling method uses a compound coefficient α to uniformly scale network D, W and R in a principled way:

$$D = a^{\alpha}, \tag{1}$$

$$W = b^{\alpha}, \tag{2}$$

$$R = c^{\alpha}, \tag{3}$$

$$s.t.a \cdot b_2 \cdot c_2 \approx 2, \tag{4}$$

$$a \geq 1, b \geq 1, c \geq 1, \tag{5}$$

where D is depth, W is width and R is resolution.

The principle behind the LSTM architecture is a memory cell which can sustain its state over time [40], and non-linear gating units which manage the information flux into and out of the cell. A deep network of convolutional LSTM can be used to access the full spectrum [41] of temporal information at spatial scales of the data. This architecture can be applied to any application where spatio-temporal features play a key role without having to deliberately cater the design of the network for the particular spatio-temporal features existent within the problem.

As Greff et al. [42] mentioned, convolutional LSTM as averse to a basic convolution layer, incorporates an internal cell state c_t and calculates a hidden state h_t utilized as the output for the subsequent layers as well as for state-to-state transitions. While processing sequences of frames of a video, c_t and h_t can be viewed as images of appropriate size sustained by the network with relevant information based on what it has observed in the past. Learnable filters W_f with bias terms b_f are used to handle a new input x_t along with the past information being used by learnable filters Z_f.

$$\Gamma_f = \sigma(W_f * x_t + Z_f * h_{t-1} + b_f), \tag{6}$$

$$\Gamma_i = \sigma(W_i * x_t + Z_i * h_{t-1} + b_f), \tag{7}$$

$$\Gamma_o = \sigma\left(W_o * x_t + Z_o * h_{t-1} + b_0\right), \tag{8}$$

$$c_t = \Gamma_f \circ c_{t-1} + \Gamma_i \circ \tanh\left(W_c * x_t + Z_c * h_{t-1} + b_c\right), \tag{9}$$

$$h_t = \Gamma_o \circ \tanh\left(c_t\right). \tag{10}$$

The model is fed with 3D input data of shape $m \times 30 \times 240 \times 240 \times 3$ (where m is the batch size) into the EfficientNet-B1 wrapped inside a time-distributed layer [43], so that the backbone network could be applied to every temporal slice of the input. Subsequently, the 3D data get converted into feature vectors which act as time-steps to be fed into the LSTM layer which learns the features and long-term dependencies across all the frames and outputs a sequence descriptor. Finally, the detection network having fully connected layers is added to take the previous output as input and calculate the probabilities of the sequence of frames belonging to either fake or real class as illustrated in Fig. 5.

The advantage that this architecture has is that when multiple frames of the same video are fed-forward through

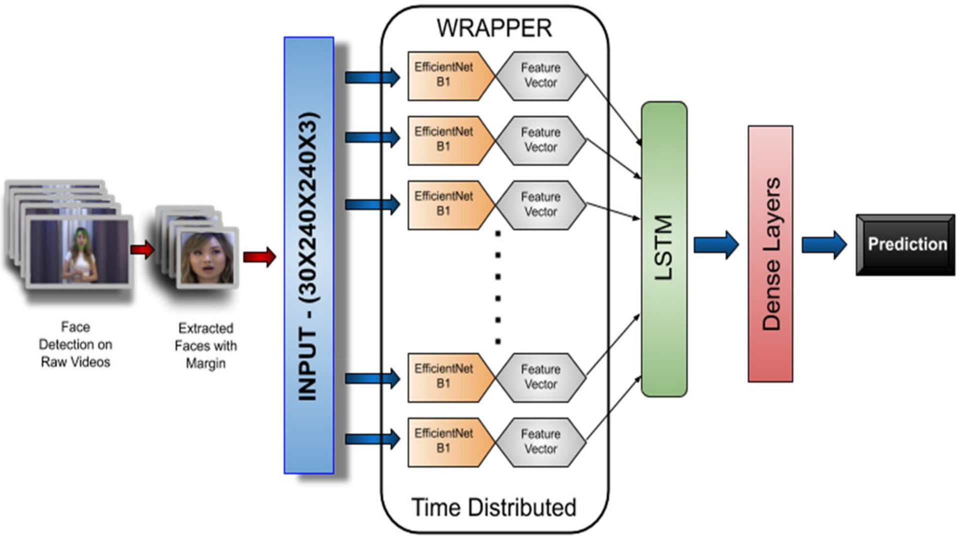


Fig. 5 Detailed architecture with visualization of time-distributed layer

Table 1 Distribution of DFDC dataset in accordance with our implementation

Set	Real class (videos)	Fake class (videos)
Training	16,142	85,123
Validation	832	4498
Testing	826	7111

the network, the chances of detecting anomalies become high due to discrepancies between frames.

Results and Discussions

In this section, the results and findings of the implementation of the proposed architecture to detect digital forgeries have been presented.

Experimental Setup

In total, pre-processing of 1,14,532 videos has been performed; within these videos, 17,800 real and 96,732 fake videos have been used, as shown in Table 1. Down-sampling of the fake videos has been done so that the imbalances in each category could be eliminated. 30 frames over the entire length, from an approximated average of 300 frames per video have been extracted. Random splitting of the data into training, validation, and test sets has been performed.

The learning rate has been set to $lr = 10^{-4}$ with a batch size of two (As the data was high dimensional and the authors had memory limitations). The Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ has been used. Python 3.6.6 and Tensorflow 2.1.0 have been used to implement the network.

For comparison amongst models, depending on the dataset, the following metrics have been used:

Accuracy: This calculates how often prediction matches the labels.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{FN} + \text{TN})},$$

where TP is true positives, TN is true negatives, FP is false positives and FN is false negatives.

Mean absolute error (MAE): This computes the mean absolute error between the labels and predictions.

$$\text{MAE} = \frac{\Sigma_{i=1}^{n} |y_i - x_i|}{n},$$

where y_i is the prediction and x_i is the true value.

F1 score: F1 score can be defined as the weighted average of precision and recall.

$$\text{F1Score} = \frac{2 \times (\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})},$$

where Precision $= \frac{\text{TP}}{\text{TP+FP}}$ and Recall $= \frac{\text{TP}}{\text{TP+FN}}$.

Area under the curve (AUC): This approximates AUC via a Reimann sum. Receiver operating characteristics (ROC) is the probability curve and AUC represents degree or measure of separability. It tells how much a model is capable of distinguishing between classes.

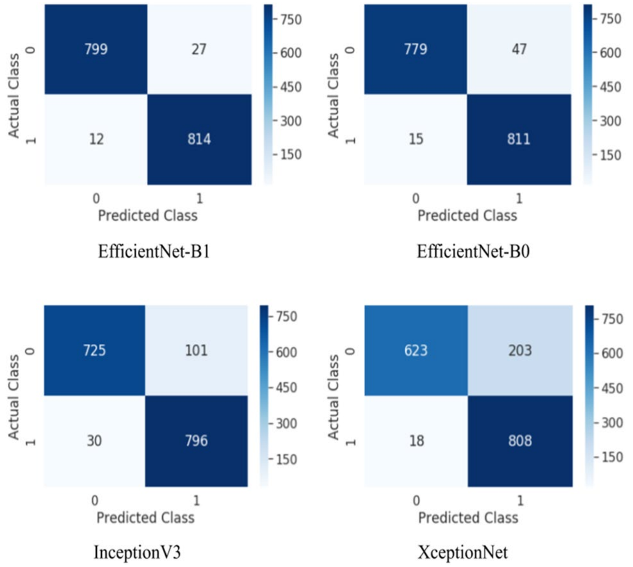
Forgery Detection Results

To test the proposed manipulation detection technique, the DFDC dataset has been used. This dataset makes use of DeepFake techniques to warp the videos. The dataset was divided into a training set, a validation set, and a test set. For testing, 30 frames from the input video were fed-forwarded to the trained model. Rigorous testing has been performed by shuffling and sampling the fake videos to the count of real videos. After testing multiple times in the above-mentioned way, the mode value of all the scores was considered the final score of that particular model. Experimentational insights have led to the consideration of different CNN architectures like *EfficientNet-B1*, *EfficientNet-B3*, *XceptionNet*, and *InceptionV3* to act as backbone networks for the complete model. Backbone network is used to extract features vectors from the input image**.** Larger models were eluded from consideration as they were more prone to overfit in relevance with the data used, also smaller networks were more successful in detecting some lower-level spatio-temporal features.

As it is evident from Table 2, the architecture consisting of EfficientNet-B1 outperformed the other analogous architectures. EfficientNet-B1 was able to learn and generalize the feature vectors in a slightly better manner. Due to its comparatively smaller size and better optimization, overfitting was prevented by the model to a larger extent. Furthermore, EfficientNet-B0 gave competitive results but was found insipid compared to its scaled successor. Best in class scores were unattainable using previous state-of-the-art models, XceptionNet and InceptionV3 as backbone networks. However, ideal a model may seem, it is substantially not impeccable. Thus, the comparison of the performances of the models has been done in an intuitive manner as shown in Fig. 6. Confusion matrices have been plotted to comprehend the errors made by the respective models and to calculate the F1 scores using the count of true positives, false negatives, false positives and true negatives.

Conclusions

Nowadays, malicious people use various techniques to manipulate images and videos and publish them on social networks or commit crimes which impose threats to society. Therefore, the authors have proposed an architecture which is enlightened to leverage the spatio-temporal features while detecting forged videos inexpensively with respect to computational cost. Among the backbone networks, *EfficientNet-B1* gives optimal results with an accuracy of 97.6% while having fewer parameters, making it a viable option for the problem. The authors aspire that the solution which has been bestowed will act as a premise for subsequent enhancements in the paramount techniques to detect digitally exploited videos.



EfficientNet-B1

EfficientNet-B0

InceptionV3

XceptionNet

Fig. 6 Visualization of the confusion matrix

Compliance with Ethical Standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Table 2 Comparison among several backbone networks

Backbone network	Accuracy (%)	AUC (%)	MAE	F1 score	No. of parameters
EfficientNet-B1	97.63	99.61	0.0390	0.976	7,304,961
EfficientNet-B0	96.24	99.60	0.0519	0.961	4,779,293
XceptionNet	86.62	96.41	0.1903	0.849	21,984,425
InceptionV3	92.07	97.67	0.1201	0.917	22,925,729

References

1. Duhan N, Mittal M. Opinion mining using ontological spam detection. In: 2017 international conference on Infocom technologies and unmanned systems (trends and future directions) (ICTUS). IEEE; 2017. p. 557–62.
2. Agarwal A, Mittal M, Pathak A, Goyal LM. Fake news detection using a blend of neural networks: an application of deep learning. SN Comput Sci. 2020;1:1–9.
3. Aggarwal A, Chauhan A, Kumar D, Mittal M, Verma S. Classification of Fake News by Fine-tuning Deep Bidirectional Transformers based Language Model. EAI Endorsed Transactions on Scalable Information Systems Online First; EAI: Ghent, Belgium; 2020.
4. https://www.businessinsider.in/tech/welcome-to-deepfake-hell-how-realistic-looking-fake-videos-left-the-uncanny-valley-and-entered-the-mainstream/articleshow/69906413.cms. Accessed 16 Apr 2020.
5. https://www.theguardian.com/technology/ng-interactive/2019/jun/22/the-rise-of-the-deepfake-and-the-threat-to-democracy. Accessed 17 Apr 2020.
6. https://www.github.com/deepfakes/faceswap. Accessed 20 Apr 2020.
7. https://www.malavida.com/en/soft/fakeapp. Accessed 20 Apr 2020.
8. Li Y, Chang MC, Lyu S. 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. p. 1–7.
9. Dale K, Sunkavalli K, Johnson MK, Vlasic D, Matusik W, Pfister H. Video face replacement. In: Proceedings of the 2011 SIGGRAPH Asia conference. 2011. p. 1–10.
10. Garrido P, Valgaerts L, Rehmsen O, Thormahlen T, Perez P, Theobalt C. Automatic face reenactment. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014. p. 4217–24.
11. Thies J, Zollhofer M, Stamminger M, Theobalt C, Nießner M. Face2face: Real-time face capture and reenactment of rgb videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 2387–95.
12. Tolosana R, Vera-Rodriguez R, Fierrez J, Morales A, Ortega-Garcia J. DeepFakes and beyond: a survey of face manipulation and fake detection. arXiv preprint arXiv:2001.00179 (2020).
13. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Advances in neural information processing systems. 2014. p. 2672–80.
14. Korshunova I, Shi W, Dambre J, Theis L. Fast face-swap using convolutional neural networks. In: Proceedings of the IEEE international conference on computer vision. 2017. p. 3677–85.
15. Rössler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Nießner M. Faceforensics: a large-scale video dataset for forgery detection in human faces. arXiv preprint. arXiv:1803.09179 (2018).
16. Rossler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Nießner M. Faceforensics++: learning to detect manipulated facial images. In: Proceedings of the IEEE international conference on computer vision. 2019. p. 1–11.
17. https://www.kaggle.com/c/deepfake-detection-challenge/data. Accessed 5 Feb 2020.
18. Dolhansky B, Howes R, Pflaum B, Baram N, Ferrer CC. The Deepfake Detection Challenge (DFDC) preview dataset. arXiv preprint. arXiv:1910.08854 (2019).
19. Marra F, Gragnaniello D, Verdoliva L, Poggi G. Do gans leave artificial fingerprints? In: 2019 IEEE conference on multimedia information processing and retrieval (MIPR). IEEE; 2019. p. 506–11.
20. Mittal M, Verma A, Kaur I, Kaur B, Sharma M, Goyal LM, Roy S, Kim TH. An efficient edge detection approach to provide better edge connectivity for image analysis. IEEE Access. 2019;13(7):33240–55.
21. Afchar D, Nozick V, Yamagishi J, Echizen I. Mesonet: a compact facial video forgery detection network. In: 2018 IEEE international workshop on information forensics and security (WIFS). IEEE; 2018. p. 1–7.
22. Yu CM, Chang CT, Ti YW. Detecting Deepfake-forged contents with separable convolutional neural network and image segmentation. arXiv preprint. arXiv:1912.12184 (2019).
23. Mittal M, Goyal LM, Kaur S, Kaur I, Verma A, Hemanth DJ. Deep learning based enhanced tumor segmentation approach for MR brain images. Appl Soft Comput. 2019;1(78):346–54.
24. Mittal M, Arora M, Pandey T, Goyal LM. Image segmentation using deep learning techniques in medical images. In: Advancement of machine intelligence in interactive medical image analysis. Singapore: Springer; 2020. p. 41–63.
25. Mittal A, Kumar D, Mittal M, Saba T, Abunadi I, Rehman A, Roy S. Detecting pneumonia using convolutions and dynamic capsule routing for chest X-ray images. Sensors. 2020;20(4):1068.
26. Goyal LM, Mittal M, Kaushik R, Verma A, Kaur I, Roy S, Kim T-H. Improved ECG watermarking technique using curvelet transform. Sensors. 2020;20:2941.
27. Mittal M, Kaushik R, Verma A, Kaur I, Goyal LM, Roy S, Kim TH. Image watermarking in curvelet domain using edge surface blocks. Symmetry. 2020;12(5):822.
28. Nguyen HH, Yamagishi J, Echizen I. Capsule-forensics: using capsule networks to detect forged images and videos. In: ICASSP 2019–2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; p. 2307–11.
29. Stehouwer J, Dang H, Liu F, Liu X, Jain A. On the detection of digital face manipulation. arXiv preprint. arXiv:1910.01717 (2019).
30. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. 2015. p. 4489–97.
31. Chollet F. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 1251–8.
32. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 2818–26.
33. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. p. 7132–41.
34. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770–8.
35. Rahmouni N, Nozick V, Yamagishi J, Echizen I. Distinguishing computer graphics from natural images using convolution neural networks. In: 2017 IEEE workshop on information forensics and security (WIFS). IEEE; 2017. p. 1–6.
36. Soviany P, Ionescu RT. Continuous trade-off optimization between fast and accurate deep face detectors. In: International conference on neural information processing. Cham: Springer; 2018. p. 473–85.
37. https://github.com/yeephycho/tensorflow-face-detection. Accessed 13 Feb 2020.
38. Tan M, Le QV. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint. arXiv:1905.11946 (2019).
39. He X, Zhao K, Chu X. AutoML: A Survey of the state-of-the-art. arXiv preprint. arXiv:1908.00709 (2019).
40. Mittal M, Arora M, Pandey T. Emoticon prediction on textual data using stacked LSTM model. In: International conference on

communication and intelligent systems. Singapore: Springer; 2019. p. 259–69.

41. Courtney L, Sreenivas R. Learning from videos with deep convolutional LSTM networks. arXiv preprint. arXiv:1904.04817 (2019).
42. Greff K, Srivastava RK, Koutník J, Steunebrink BR, Schmidhuber J. LSTM: a search space odyssey. IEEE Trans Neural Netw Learn Syst. 2016;28(10):2222–32.
43. https://www.keras.io/layers/wrappers. Accessed 13 Feb 2020.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.