www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 9 Issue 10 October 2020, Page No. 25199-25207
ISSN: 2319-7242 DOI: 10.18535/ijecs/v9i10.4533

Deep Learning Algorithms and Frameworks for Deepfake Image and Video Detection: A Review

Worku Muluye

Department of Information Systems, Wolkite University, Wolkite, Ethiopia

Abstract

Deep learning algorithm is used to detect as well as create deepfake images and videos. Images and videos are often used as evidence in police investigations and courts to resolve legal cases since they are considered to be reliable sources. However, deepfake technology increases the development of fake videos, and this may lead to image or video evidence unreliable. This paper aims to qualitatively compare deep learning algorithms and frameworks. To detect real and fake images or videos, various detection algorithms have been proposed after deepfakes were introduced. The current deepfake detection algorithms detect the deepfakes by eye blinking, eye teach and facial texture, head poses, face warping artifacts, eye color, lip movements, audio speakers, reflections in the teeth, spatiotemporal features, and capsule forensics. Deepfake detection algorithms and deep learning frameworks are selected and compared. Deep learning frameworks with different performance and features such as TensorFlow, CNTK, Caffe, Torch, Chainer, and Theano are compared. This helps to use appropriate deep learning algorithms and frameworks for deepfake detection.

Keywords: Detection Algorithms, Deepfake Detection, Deep Learning, Image Detection, Video Detection.

1. Introduction

Deep learning is the arrangement of algorithms that can learn on a dataset and make intelligent decisions on their own [1]. Deep learning algorithms are used to detect and create deepfakes, social network filtering, image and speech recognition, fraud detection, computer vision, audio recognition, image processing, and customer relationship management [2]. Deepfake images and video created by deepfake algorithms becomes a great public issue recently [3]. Deepfake is a deep learning-based technology to create deepfake videos or images by manipulating the face or full-body of one person in video or image by the face or full-body of another person [4]. Images and videos are often used as evidence in police investigations and courts to resolve legal cases since they are considered to be reliable sources. However, deepfake increases the development of fake images, and videos that have possibly made images or video evidence are untrustworthy [5]. Generative Adversarial Networks (GANs) [6] is an image or video manipulating deep learning algorithm to create high-quality deepfake videos and images, and the media increases the fast distribution of these fake images and videos [7]. GAN is an algorithm used for unsupervised learning, cybersecurity, natural language processing, health

diagnostics, and speech processing [2]. The GAN models were trained on a large image or video datasets, and it can generate genuine faces or full-body that can be seamlessly spliced into the original video, and the generated video can lead to forgery of the subject's identity in the image or video [8], [9]. The combination of Convolutional Neural Networks (CNNs) and GANs can create deepfakes that the detection algorithms cannot detect them [4].

The existence of, open software mobile applications increasing to everyone to generate fake videos and images [4]. The smartphone availability, advancement of cameras, and social media popularity have made the editing, creation, and dissemination of images and videos more than ever. This increases the altering of videos and makes it effective to share falsified information [8]. After deepfake technology were introduced to detect deepfakes, various deepfake detection algorithms have been proposed. Deepfake detections are algorithms that is trained on a dataset to detect fake and real images or videos. Currently, the detection algorithms can detect the deepfakes by using facial texture, eye blinking, head poses, eye color, face warping artifacts, audio speakers, lip movements, reflections in the teeth, capsule forensics, and spatiotemporal features [10]-[14].

A lot of deepfake detection algorithms are proposed to detect fake images and videos. The common deepfake detection algorithms such as Convolutional Neural Network (CNN) to extract frame feature, LRCN to capture the eye blinking temporal patterns, Recurrent Neural Network (RNN) to discover temporal discrepancies across frames, and Long Short-Term Memory (LSTM) for temporal sequence analysis [5]. CNN with Error Level Analysis, CNN and LSTM [15], Hybrid CNN and SVM, CNN with CFFN [10], CNN, RCN, capsule networks, logistic regression, and neural network [9]. TensorFlow, CNTK, Caffe, Torch, Chainer, and Theano are deep learning frameworks with different performance characteristics and features. Each deep learning framework tries different methods to optimize its application of deep learning algorithms. Comparing deep learning frameworks is very important to enable people who are interested in applying deep learning in their research to identify suitable frameworks for their works [16]. Today machine learning and deep learning-based research are needed a high-performance computer to train the models on large datasets. The most common free online GPU, CPU, and storage services for training models are Google Colab, Kaggle Kernel, Jupiter Notebook, Amazon SegeMaker, Azure Notebooks, etc. [17]. In this study, the deepfake, deepfake image and video detection, deepfake detection algorithms, deep learning frameworks, public image, and video datasets and, and online free computational power services have been investigated.

2. Related Works

Ali Shatnawi et al has quantitatively and qualitatively compared TensorFlow, Theano, and CNTK deep learning frameworks [16]. In [18] has compared RNNs and CNNs for natural language processing tasks. They find that CNTK's framework is better than other frameworks. Brian Dolhansky, et.al. [19] studied deepfake image or video detection challenges and features facial manipulation algorithms. In [20] states the challenges and opportunities of fake news and detecting fake news by proposing algorithms which detect fake news form the web services. Weihong Deng, et.al. [21] reviewed recent developments on deep face recognition including algorithm designs, databases, protocols, and application scenes. This paper [4] presents a review of face image manipulation techniques, deepfake methods, and methods to detect manipulations. In this [14] has compared deepfake detection algorithms that are used to create deepfakes and, detect deepfakes. Also, they discussed the challenges, directions, and research trends to deepfake technologies and deepfake methods. The researchers examine how deepfake photos videos are created and present deepfake impacts on society. After deepfake technology, several deepfakes detection algorithms have been developed such as convolutional neural networks (CNNs), face detection, multimedia forensics, and watermarking. Each method uses machine learning and deep learning, to detect any kind of manipulation in photos and videos [5], [7], [10].

3. Deepfake Detection

Deepfake technology leads to distrust that what we see and hear from social media. Regarding this, currently different detection algorithms and detection methods have been proposed to detect deepfakes images and videos. The deepfake detection methods need a huge dataset of real and fake images or videos to train the machine. A lot of deepfake images and video algorithms are developed to detect falsified information [15]. Deepfake is created using Generative Adversarial Networks (GANs), in which two machine learning models exit. One model train on a fake and real dataset and then creates image or video forgeries, and the other model tries to detect the forgeries. The forger creates fake images or videos until the other model cannot detect fake images or videos. The larger the training dataset, the easier it is for the forger to create a believable deepfakes [14], [22]. The first deepfake was created by using an autoencoder-decoder structure. The autoencoder extracts hidden features of face images and the decoder reconstruct the face images. To manipulate faces images, two encoder-decoder pairs train on a dataset [4], [14]. This method allows the encoder to encode and learn the image datasets. In the deepfake creation process, DFaker, DeepFaceLab, Face2Face, and NeuralTextures, and TensorFlow-based deepfakes creates the deepfake by swapping face [14]. FaceSwap is a graphics-based method to alter the face image from a source video to a target video by using two encoder-decoder pairs [23]. DFaker reconstructs the face and is implemented based on the Keras library. Also, the TensorFlow-based deepfake is the same as with DFaker but the TensorFlow-based deepfake is applied based on the TensorFlow library. The NeuralTextures with rendering network uses the video dataset to learn the texture of the target person [22], [24], [25].

3.1.Deepfake Images Detection

The authors in [24] aimed to detect manipulated facial images by using different face tracking methods. The DARPA advances of fake digital visual media detection algorithms [14]. The swapped face images made by CNN and GAN deep learning are more challenging for forensics models to detect the forgeries [14]. In [26] uses deep transfer learning for face swapping detection methods of deepfake images. The fake face image detection is the most serious challenging issue in the fake face image detection field. Fake face images are created fake identities on social media networks, so theft personal information illegally [10]. Huy H. Nguyen, et. al. [12] developed a multitask learning algorithms to do classification and segmentation of altered facial images concurrently.

3.2.Deepfake Videos Detection

In the detection process, the temporal consistency of the video is not imposed efficiently and is enforced to use the Spatiotemporal contents of the video to detect deepfakes [14]. The integration of conventional networks and recurrent unit manipulate the temporal inconsistencies of the frames [27]. Using both Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) able to extract temporal features of the video, that is denoted through the sequence of frames. The detection network takes the frame sequence and detects original or fake frames by estimating the frame sequences. CNN is used to extract features of the frame and then fed them into the LSTM to generate temporal sequence frames [4], [14], [28], [29]. Using physiological signal such as eye blinking, to detect deepfakes has introduced in which a person in deepfakes has fewer eye blinking than in altered videos. A normal person's eye will blink between 2 to 10 seconds, and each eye blink will take between 0.1 and 0.4 seconds. Mostly, deepfake creation tools cannot create fake eye blinking that can blink like a normal person. Most of the time eye blinking speed is manipulated videos are slower than in real videos [5]. The deepfake detection technique uses the audio-visual dataset to detect inconsistencies between lip-movements, speech, and numerous differences of slow images. The deepfake detection technique through lip-synching differentiates the real videos from altered videos, where the lip-movement and the speech are matched [30], [31]. Long-Term Recurrent Convolutional Networks (LRCN) is an algorithm that predicts eye area sequences dynamically. Also, it contains a feature extractor that extracts depend on CNN, a sequence learning through LSTM, and to predict the possibility of open and closed eye state. The LSTM captures the eye-blinking temporal patterns effectively. The extremely frequent eye blinking may also be the criteria of altered videos [14]. Deep recurrent network models can differentiate deepfake videos by using temporal patterns across video frames. The other approach that fragment videos into frames and identify visual objects inside single frames to get discriminant features. The resolution discrepancy between the nearby and the changing face area can be detected by CNN models [4], [14].

3.3.Deepfake Detection Algorithms

Deepfake detection algorithms are a step by step procedure to detect fake images and videos. If we are unable to detect deepfakes, we may distrust everything that we see and hear. The common deepfake detection algorithms such as CNN to extract frame feature, LRCN to capture the eye blinking temporal patterns, Recurrent Neural Network (RNN) to discover temporal discrepancies across frames, and LSTM for temporal sequence analysis [5]. The deepfake detection algorithms detect the deepfake videos and images by eye-blinking, eye teach and facial texture, head poses, lip-synching, face warping artifacts, physiological signal, reflections in the teeth, spatiotemporal features, and capsule forensics [5], [9], [14].

Table 1: Deep learning algorithms for deepfake image or video detection [8]- [11], [32].

Detection Algorithms	Description	Detection methods
CNN with Error Level Analysis [33]	CNN extracts the forged feature and detect whether images are fake and error level analyses detect altered images and different image compression ratios.	Real and fake image face area/features
Hybrid CNN and SVM [34]	The LeNet-5 CNN model detects the input image background, open eye, and closed eye states. The SVM selects an area in which an eye cannot be found.	Eye blinking
Multimedia forensics [5]	Detects the detailed history of images and determines whether images have been altered.	Detect altered images
Multi-task [35]	It uses a CNN model to concurrently detect altered images and segment altered areas as a multi-task learning problem.	Detect altered images
CNN with CFFN [10]	CNN performs classification, and CFFN makes two-phase feature extraction by using an image dataset.	Pairwise learning
CNN [11, 23, 32]	Detection technique for face tampering in videos generated through Deepfake and Face2Face apps [11]. It is a powerful image analysis to detect any minor modifications that have been made to an image or video. Detect fake video by using face warping artifacts.	Face warping artifacts, facial artifacts, and noise patterns
PRNU [9]	Detect the original and fake video patterns since face-swapping is trusted to alter the local patterns.	PRNU Analysis
LRCN [8]	LRCN captures eye blinking temporal patterns in the videos. The eye blinking frequency of fake video is slower or faster than normal videos.	Eye blinking, lip movement
RCN [37]	RCN discovered temporal inconsistencies across the frame. It uses Spatiotemporal features of videos.	Using spatiotemporal features
CNN and LSTM [28]	CNN detect frame features then disseminate to LSTM and LSTM detect temporal inconsistencies in videos to set frame sequences for classification. VGG16, and ResNet50 are CNN models that used to detect resolution inconsistency between the nearby and the changed face area from video [32].	Eye blinking [15], Intra-frame, and temporal Inconsistencies
Capsule Networks	The capsule network classifies the hidden features and dynamic routes the 3 capsules output to 2 capsules output, one output for fake and the other output for real images/videos. It deals with videos/images dataset.	Capsule-Forensics
Logistic regression and	Logistic regression and neural network are discovered facial texture and missing reflections of eye and teeth areas and	Eye teaches and facial

neural network [9]	classifying videos.	texture

4. Datasets and Computational Power Online Free Services

Deepfake detection algorithms required a large training dataset, and the trained dataset needs to be evaluated. As such, there is an increasing need for a huge amount of deepfake image and video datasets to evaluate the performance of detection techniques [35]. There are deepfake datasets to train the deepfake detection models [14], [38]. UADFV [14], [35] is a publically available video dataset created by using the DNN model with FakeAPP. UADFV contains 98 videos of which 49 fake and 49 real videos and these videos contain 32,752 frames. DeepfakeTIMIT [19], [39] is a dataset that contains low and high-quality videos with a total of 10537 real and 34,023 fake images extracted from 320 total videos. It is created by FaceSwap-GAN. VTD [24] Video Tampering is a large, manipulated video dataset that contains 1000 videos. FaceForensics [13] is created from 1004 videos collected from YouTube. The video was altered by using the Face2Face approach. DFD [35] is the Google or Jigsaw deepfake video dataset generated based on 3068 real videos and 28 consented individuals of different ages, genders, and ethnic groups. DeepFakeDetection [39] It contains 363 original sequences and more than 3000 altered videos by using deepfakes and their corresponding binary masks. FaceForensics++ [38] is an extended dataset from FaceForensics which contains real and fake videos generated using FaceSwap. DFDC [19] is the Facebook deepfake detection challenge that has 4113 deepfake videos created based on 1131 real videos of 66 consented individuals of different ages, genders, and ethnic groups. Celeb-DF is a public dataset that contains 590 real and 5639 fake videos that contain above two million video frames [35].

Today machine learning and deep learning-based research are needed high computation power to train the models on large datasets. However, this challenges the researchers who have a machine with less computational power. To solve this proble0m various online free computation power services such as online free CPU, GPU, TPU, and storage spaces are available to train deep learning models. The most common free online GPU and CPU services (free computational power cloud services) are Google Colab, Kaggle Kernel, Jupiter Notebook, Amazon SegeMaker, and Azure Notebooks. These are online free computation power services to train models [17].

5. Deep Learning Frameworks for Deepfake Detection

Training time, testing time, and adversarial robustness are some critical parameters for deep learning frameworks. The training time is a time spent on building a model over the training dataset. Even if pre-trained models are made available, training is still critical for new model development. Testing time is the time spent on testing the trained model using a valid dataset. Testing time shows the latency of the trained model for prediction or learning. Prediction/learning accuracy measures for the effectiveness of the trained model on a training dataset at the testing phase. Accuracy and training time is critical for both system-specific and data-specific parameters, such as the type of the datasets, the learning rate, the number of prediction classes, the number of training samples per-class, the number of iterations, the batch size computations, the type of deep learning frameworks and library used. Adversarial robustness is designed to measure the resilience of the deep learning framework and its trained model beside adversarial during the testing time [40], [41]. Today there are a lot of deep learning frameworks that can be used to execute our codes when we train the models to detect fake images and videos [42].

Table 2: Deep learning frameworks for deepfake image and video detection [42]- [46].

Criteria	TensorFlow	CNTK	Caffe	PyTorch	Chainer	Theano
License	Apache 2.0	MIT	BSD	BSD	Open source	BSD
Written in	C++, Python	C++	C++	Python, C, CUDA	Python	Python
Computation graph	Static with mall support dynamic graph	Static	Static	Dynamic	Dynamic	Static
Interface Support	Python, Go, R, Java, C/C++	Python, C++, BrainScript, ONNX	C++, Python, MatLab	Python, ONNX	Python	Python
Popularity	Very High	Medium	High	Medium	Low	Medium-low
Usage	Academic Industrial	Academic Industrial Limited mobile solution	Academic Industrial	Academic Industrial	Academic Industrial	Academic Industrial
Created by	Google	Microsoft	Y. Jia, BAIR	A. Paszke, S. Gross, S. Chintala, G. Chanan	Preferred Networks	Y. Bengio University of Montreal
CPU	Yes	Yes	Yes	Yes	Yes	Yes
Multi-threaded CPU	Yes (Eigen)	Yes	Yes (Blas)	Yes	Yes	Yes
GPU	Yes	Yes	Yes	Yes	Yes	Yes
Multi-GPU	Yes (Most flexible)	Yes	Yes	Yes	Yes	Yes
NVIDIA cuDNN	Yes	Yes	Yes	Yes	Yes	Yes
Platform	Linux, Mac OS X, Windows	Linux, Windows	Ubuntu, OS X, Windows and Android	Linux, Android, iOS, Windows, Mac OS	Window	Cross-Platform
Deep Learning Models	DBN, CNN, RNN	CNN, RNN, FFNN,	CNN, RNN	DBN, CNN, RNN	CNN, RNN, RL	DBN, CNN, RNN
CNN & RNN Support	Yes	Yes	Yes	Yes	Yes	Yes
DBN Support	Yes	NO	NO	Yes	Yes	Yes
Fault Tolerance	Checkpoint and Recovery	Checkpoints and resume	Not applicable	Checkpoints and resume	Not applicable	Checkpoints and resume
Visualization	Graph (interactive), Training monitoring	Graph (static)	Summary Statistics	Plots	Computational graph	Graph (static)
Synchronization Model	Sync or async	Sync	Sync	Sync	Async	Async
Growth speed	Very fast	Fast	Fast	Very fast	Low	Low

The deep learning framework's performance was evaluated through the running time, CPU and GPU utilization, memory consumption, and the number of epochs [47]. Deep learning frameworks such as TensorFlow, CNTK, Caffe, PyTorch, Chainer, and Theano are selected and compared. TensorFlow supports python and R programming language and uses dataflow graphs to process the data. TensorFlow is easy to build deep learning and machine learning models. It uses a tensor-board for data visualization. The TensorFlow based data flow makes an ideal API and implementation tool for CNNs and RNNs. Caffe is a deep learning framework that supports CNN, RCNN, LSTM, and other deep learning algorithms. TensorFlow, Theano, and CNTK frameworks allow Keras to runs on top of these frameworks. Keras is user-friendly, easily extensible, and provides modularity [41], [42].

PyTorch provides dynamic computation graphs, which are useful when working with RNNs. PyTorch has lots of modular pieces that are easy to combine, lots of pre-trained models, and less plug and play. Chainer is an open-source neural network framework with a Python API for dynamic computation graphs and NLP tasks. It is faster than other python oriented deep learning frameworks [41], [42], [43], [48]. PyTorch has less plug and play, and poor documentation than TensorFlow. TensorFlow runs much fatter than PyTorch and slower than other frameworks; not support many pre-trained models. Caffe is not good for recurrent networks, not extensible, probably dying, and slow development. error messages can be unhelpful, large models can have long compile times, much fatter than Torch, and occasional support for pre-trained models [2], [49].

TensorFlow supports mobile computing, high scalable, open-source, very fast developing by Google. TensorFlow has lower-level application interfaces difficult to use directly for creating deep learning models and the computational flow constructed as a static graph. Caffe is easy to code, suitable for image processing, and has pre-trained networks. The Caffe development is not fast, the static model does not fit with many RNN applications and the custom layers must be written in C++. PyTorch supports NumPy and SciPy, open neural network exchange which allows to easily transform models between frameworks such as CNTK, PyTorch, Caffe, and Caffe2. Chainer supports dynamic computational graph, and provide libraries. Chainer does no support higher-order gradients and dynamic computational graphs are generated every time also for fixed networks. Theano is an open-source, cross-platform, powerful numerical library, and the API supports to implement RNNs efficiently. Theano lack for mobile platform, and does not support lower-level API, and no in active development [2], [40], [42].

6. Conclusion

Deep learning algorithms are used to detect and create deepfake images and videos. Deepfake is a deep learning-based method to create deepfake images or videos by altering the face or full-body of a person in an image or video to another person. In this study, deepfake detection algorithms and deep learning frameworks are selected and compared. The common deepfake detection algorithms such as CNN to extract frame feature, LRCN to capture the eye blinking temporal patterns, RNN to discover temporal discrepancies across frames, and LSTM for temporal sequence analysis. CNN with Error Level Analysis, CNN and LSTM, Hybrid CNN and SVM, CNN with CFFN, CNN, RCN, capsule networks, logistic regression, and neural network. After deepfake technology were introduced various deepfake detection algorithms have been proposed. The detection algorithms detect the deepfakes by eye blinking, head poses, eye color, facial texture, lip movements, reflections in the teeth, audio speakers, capsule forensics, and spatiotemporal features. Deep learning frameworks with different performance characteristics and features such as TensorFlow, CNTK, Caffe, Torch, Chainer, and Theano are compared. Since comparing deep learning frameworks is very important to enable people who are interested in applying deep learning in their research to identify suitable frameworks for their works. This helps the researchers to use the most suitable deep learning algorithms and frameworks for deepfake detection. Today machine learning and deep learning-based research are needed high-performance machines to train the models on large datasets. The most common free online GPU, CPU, and storage services for training models are Google Colab, Kaggle Kernel, Jupiter Notebook, Amazon SegeMaker, and Azure notebooks.

Future work will focus on the quantitative comparison of deep learning frameworks. The deep learning framework's running time, CPU and GPU utilization, memory consumption, and the number of epochs will be evaluated. Following, the training time, testing time, and adversarial robustness will be tested. The prediction or learning accuracy of the detection algorithms will be tested. Deepfake is the new challenging technology that may negatively affect religions, culture, communities, national security, and individuals across the world. The advancements of the deepfakes will be difficult to detect by the existing detection algorithms. So, cross-platform, accurate, robust, and reliable deepfake detection algorithms need to be developed in the future.

References

[1.] G. Shao, "What 'deepfakes' are and how they may be dangerous," 13 October 2019. [Online]. Available: https://deepfakedetectionchallenge.ai/.

[2.] Simplilearn, "Deep Learning Algorithms You Should Know About," Simplilearn, 28 May 2020. [Online]. Available: https://www.simplilearn.com/deep-learning-algorithms-article. [Accessed 10 June 2020].

[3.] D. D. Zahid Akhtar, "A Comparative Evaluation of Local Feature Descriptors for DeepFakes Detection," 07 November 2019.

[4.] R. V.-R. J. F. A. M. J. O.-G. Ruben Tolosana, "DeepFakes and Beyond A Survey of Face Manipulation and Fake Detection," pp. 1-15, 2020.

[5.] J. A. Marwan Albahar, "Deepfakes Threats and Countermeasures Systematic Review," JTAIT, vol. 97, no. 22, pp. 3242-3250, 30 November 2019.

[6.] B. P. W. W. J. D. Xinsheng Xuan, "On the Generalization of Generative Adversarial Networks Image Forensics," pp. 1-8, 2019.

[7.] S. M. Pavel Korshunov, "Vulnerability assessment and detection of Deepfake videos," pp. 1-6, 2019.

[8.] M.-C. C. a. S. L. Yuezun Li, "In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking," pp. 1-7, 2018.

[9.] M. R. Z. G. Marissa Koopman, "Detection of Deepfake Video Manipulation," in Proceedings of the 20th Irish Machine Vision and Image Processing conference, Belfast, 2018.

[10.] Y.-X. Z. a. C.-Y. L. Chih-Chung Hsu, "Deep Fake Image Detection Based on Pairwise Learning," Applied science, pp. 1-14, 2020.

[11.] V. N. J. Y. I. E. Darius Afchar, "MesoNet a Compact Facial Video Forgery Detection Network," pp. 1-7, 2018.

[12.] F. F. J. Y. I. E. Huy H. Nguyen, "Multi-task Learning For Detecting and Segmenting Manipulated Facial Images and Videos," pp. 1-8, 2019.

[13.] I.-S. N. S.-H. K. Nhu-Tai Do, " Forensics Face Detection From GANs Using Convolutional Neural Network," pp. 1-5, 2019.

[14.] C. M. N. D. T. N. D. T. N. S. N. Thanh Thi Nguyen, "Deep Learning for Deepfakes Creation and Detection," pp. 1-16, 2019.

[15.] W. M. Wubet, "The Deepfake Challenges and Deepfake Video Detection," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 6, pp. 1-8, 2020.

[16.] G. A.-B. R. A.-Q. M. A.-A. Ali Shatnawi, "A Comparative Study of Open Source Deep Learning Frameworks," pp. 1-7, 2018..

[17.] G.-Y. W. D. B. Yu (Emma) Wang, "Benchmarking TPU, GPU, and CPU Platforms for Deep Learning," pp. 1-19, 2019.

[18.] K. K. M. Y. H. S. u. Wenpeng Yin, "Comparative Study of CNN and RNN for Natural Language Processing," pp. 1-7, 2017.

[19.] R. H. B. P. N. B. C. C. F. Brian Dolhansky, "The Deepfake Detection Challenge (DFDC) Preview Dataset," Deepfake Detection Challenge, pp. 1-4, 2019.

[20.] L. O. Alvaro Figueira, "The current state of fake news: challenges and opportunities," in ICHSCIST, Barcelona, 2017.

[21.] W. D. Mei Wang, "Deep Face Recognition: A Survey," pp. 1-26, 2019.
[22.] J. D. Britt Paris, "Deepfakes and Cheap Fakes the Manipulation of Audio and Visual Evidence," Media Manipulation, Los Angeles, 2018.
[23.] M. V. R. S. Prabhat Kumar, "Detecting Face2Face Facial Reenactment in Videos," pp. 1-9, 2020.
[24.] R. D. C. L. V. C. R. J. T. M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," pp. 1-14, 2019.
[25.] D. D. B. B. Zahid Akhtara, "Face Authenticity An Overview of Face Manipulation Generation, Detection and Recognition," in ICCIP-2019, 2019.
[26.] Z. R. E. C. L. E. V. O. P. K. M. H. Xinyi Ding, "Swapped Face Detection using Deep Learning and Subjective Assessment," pp. 1-8, 2019.
[27.] J. C. A. J. W. A. I. M. P. N. Ekraam Sabir, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos," pp. 80-87, 2019.
[28.] E. J. D. David Guera, "Deepfake video detection using recurrent neural networks," AVSS, pp. 1-6, 2018.
[29.] J. Y. I. E. Huy H. Nguyen, "Capsule-Forensics Using Capsule Networks to Detect Forged Images snd Videos," pp. 1-5, 2018.
[30.] [30] S. M. Pavel Korshunov, "DeepFakes A New Threat to Face Recognition, Assessment and Detection," pp. 1-5, 2018.
[31.] [31] L. Z. V. L. L. T. Ying Zhang, "Automated face swapping and its detection," ICSIP, pp. 15-19, 2017.
[32.] S. L. Yuezun Li, "Exposing DeepFake Videos By Detecting Face Warping Artifacts," New York,, 2019.
[33.] C. Z. Y. L. Weiguo Zhang, "A Novel Counterfeit Feature Extraction Technique for Exposing Face-Swap Images Based on Deep Learning and Error Level Analysis," pp. 1-17, 2020.
[34.] W. K. J.-S. P. Young-Joo Han, "Efficient Eye-Blinking Detection on Smartphones: A Hybrid Approach Based on Deep Learning," Mobile Information Systems, pp. 1-9, 2018.
[35.] X. Y. P. S. H. Q. S. L. Yuezun Li, "Celeb-DF (v2): A New Dataset for DeepFake Forensics," pp. 1-6, 2019.
[36.] Y. L. S. L. Xin Yang, "Exposing Deepfakes Using Inconsistent Head Poses," ICASSP, pp. 8261- 8265, 2019.
[37.] E. C. J. J. A. A. W. M. I. N. P. Sabir, "Recurrent convolutional strategies for face manipulation detection in videos," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 80-87, 2019.
[38.] D. C. L. V. C. R. J. T. M. N. Andreas Rössler, "FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces," Computer Vision and Pattern Recognition, pp. 1-21, 2018.
[39.] M. K. F.-J. P. J. K. Ricard Durall, "Unmasking DeepFakes with simple Features," pp. 1-8, 2019.
[40.] S. C. X. X. L. M. Q. H. H. L. Y. L. J. Z. X. L. Qianyu Guo, "An Empirical Study towards Characterizing Deep Learning Development and Deployment across Different Frameworks and Platforms," pp. 1-13, 2019.
[41.] Y. W. W. W. W. C. S. S. Q. Z. Ling Liu, "Benchmarking Deep Learning Frameworks: Design Considerations, Metrics and Beyond," in IEEE 38th International Conference on Distributed Computing Systems, Georgia, 2018.
[42.] S. D. M. B. V. T. Á. L. G. I. H. P. M. L. H. Giang Nguyen, "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey," Artificial Intelligence Review, pp. 77-124, 2019.
[43.] N. R. L. S. M. S. Soheil Bahrampour, "Comparative Study of Deep Learning Software Frameworks," pp. 1-8, 2016.
[44.] J. C. Aniruddha Parvat, "A Survey of Deep-learning Frameworks," International Conference on Inventive Systems and Control (ICISC-2017), pp. 1-7, 2017.
[45.] Y. Z. J. Q. James Fox, "Software Frameworks for Deep Learning at Scale," pp. 1-7, 2018.
[46.] S. S. Y. Y. H. T. Y. T. M. P. R. M.-L. S. S.-. Samira Pouyanfar, "A Survey on Deep Learning: Algorithms, Techniques, and Applications.," ACM Computing Surveys, vol. 51, no. 5, pp. 92:1-36, 2018.

[47.] R. A.-Q. M. A.-A. A. S. Ghadeer Al-Bdour, "A DETAILED COMPARATIVE STUDY OF OPEN SOURCE DEEP LEARNING FRAMEWORKS," A PREPRINT, pp. 1-25, 2020.
[48.] Pathmind, "Comparison of AI Frameworks," Pathmind Inc., 2019. [Online]. Available: https://pathmind.com/wiki/comparison-frameworks-dl4j-tensorflow-pytorch. [Accessed 17 June 2020].
[49.] Q. W. P. X. X. C. Shaohuai Shi, "Benchmarking State-of-the-Art Deep Learning Software Tools," pp. 1-14, 2017.