Deepfake Detection and Comparative Analysis

Aryan Taneja[1] ; Tarun Singal[1] ; Rajesh Dalal[1] and Kavita Saxena[2]

[1]Department of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India
[2]Assistant Professor, Department of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India

Abstract: Deep learning has been successfully applied to solve various complex problems ranging from big data analytics to computer vision and human-level control. Deep learning advances however have also been employed to create software that can cause threats to privacy, democracy and national security. One of those deep learning powered applications recently emerged is "deepfake". Deepfake algorithms can create fake images and videos that humans cannot distinguish them from authentic ones. The proposal of technologies that can automatically detect and assess the integrity of digital visual media is therefore indispensable. In this work, we present a Sequential Convolutional Neural Network model that can accurately detect deepfake images without using very high computational power. The architecture performs satisfactorily results in distinguishing fake images from their real counterparts and is light enough that in future it may even be possible for the model to be embedded on the web.

KEYWORDS: Deep Learning, Convolutional Neural Network, DeepFake.

DOI of the Article: https://doi.org/10.46501/IJMTST0706030

Available online at: http://www.ijmtst.com/vol7issue06.html

As per UGC guidelines an electronic bar code is provided to seure your paper

To Cite this Article: Aryan Taneja; Tarun Singal; Rajesh Dalal and Kavita Saxena. Deepfake Detection and Comparative Analysis. *International Journal for Modern Trends in Sceicen and Technology* 2021, *7*, 0706136, pp. 175-180. https://doi.org/10.46501/IJMTST0706030

Article Info. Received: 19 May 2021; Accepted: 10 June 2021; Published: 15 June 2021

Introduction

Deepfake (stemming from "Deep Learning" and "Fake"), is a technique to create fake media in which a person's features such as face, audio etc are replaced with another person's features. Photo Manipulation and art forgery can be considered as a precursor to deepfakes, with the exception that both were manual operations with limited success and required great expertise, while deepfakes are easier and have recently been more successful in fooling the world.

Deepfakes make use of some common computer vision models to analyse and recreate facial and motor movements, and require large training data of images and videos of same person to create realistic fakes. Hence, public figures have been the most common targets of deepfakes. Fabricated videos of politicians have become common on WhatsApp groups these days, manipulating public opinion and have emerged as a major threat to fair elections It's not like deepfakes can only have a negative influence on the world, and have shown to be very successful in spreading positive messages through the world, such as a message by Malaria Must Die campaign in which David Beckham appears to speak in nine different languages. Deepfakes have also pushed forward creation of deep-learning training environments, where using this technology we can create true-to-life artificial data. Hence, like any technology, how deepfake works depends on the user, and therefore it has become a necessity to be able to detect deepfakes with extreme accuracy, especially when deepfakes are improving at such a great speed. Deepfake detection methods were introduced as soon as the threat of deepfake was introduced. Early attempts included handcrafted features, while recent methods used end-to-end deep learning to automatically extract salient and discriminating features of deep fakes.

Deepfake detection suffers from the very common problem of deep learning, lack of data. Any method of deepfake detection requires a large dataset of both real and fake videos to train, and while deepfakes are increasingly available, but there is still a limit for bench-marking purposes. To address this issue, Korshunov and Marcel produced a notable deepfake data set consisting of 620 videos based on the GAN model using the open source code Faceswap-GAN. Popular face recognition methods, such as systems based on Facenet, proved to be ineffective in detecting deepfakes, while methods such as detection based on lip-syncing approaches also failed to make the cut. This raises concerns about the critical need of deepfake detection models, especially as deepfakes continue to improve and become more and more genuine.

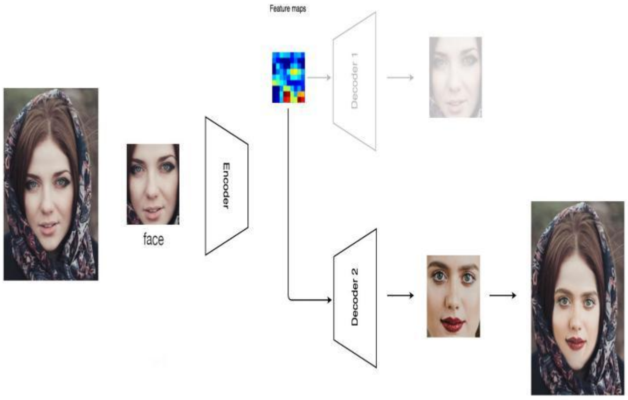


Figure1: Creation of DeepFake

Good deepfake detection models are the need of the hour, with social-media sites such as Facebook, twitter and internet giants such as google, Microsoft, all of them investing in the same. With the information manipulating capability of deepfakes, free will is at risk if the development of good deepfake detection models is postponed any further[1].

a. Objective of Study

- Recognize deepfakes satisfactorily
- Reduce the threat posed by deepfakes
- Differentiate between real and fake images
- Use considerably less computational power
- Compare the results and time taken with famous Deep learning models

b. Scope of Study

The scope of this project is generated due to the fact that there are already some really great Deep Learning Models available online but their size is so big that if we train them on a new Dataset from scratch it would take days or in some cases even weeks depending upon the size of the Dataset and not everybody has that kind of computational power so we decided to build a model that uses considerably less computational power than the models present out there and provides almost similar results when compared to these other Models

RELATED WORK

In Ian J. Goodfellow; et. al. [3], presented at the 2014 NeurIPS Conference, the authors describe generative adversarial networks, or GANs, the technology that makes deepfakes so realistic. GANs work by pitting two artificial intelligence computer models against each other. It's sort of like counterfeiting currency, the authors explain. Imagine a counterfeiter trying to create fake currency that looks real. There are also police, who try to detect the fake currency. The goal is to trick the police.

To produce deepfakes, one computer model acts like the counterfeiter and tries to create an artificial face based on example images. The other model acts like the police and compares the artificial productions to the real images and identifies places where they diverge. The models go back and forth many times until the artificial image is practically identical to the original.

The big breakthrough with GANs is that they allow computers to create. Before GANs, artificial intelligence algorithms could classify images, but had a harder time creating them. However, it was in late 2017 when Deepfakes actually burst on to the scene, when an anonymous Reddit user uploaded several adult videos on the internet under the pseudonym "Deepfakes". While photo and video manipulations have been prevalent since 19th century, there had never been seen more realistic fake videos. Since then, there has been an extensive focus on recognising deepfakes as soon as possible and reducing the damage caused by them. In Thanh Thi Nguyen; et. al. [4], we learn how the early explorers of the detection technology faced the problem of lack of reliable datasets. We also learn how the more prevalent networks of early deepfake era, such as VGG and Facenet were unable to effectively detect deepfakes, while lip-syncing approaches and image quality metrics using SVMs produce a high error rate. We also see a mention of deep fake detection models such as

ResoNet50, EfficientNetB7, XceptionNet etc.

As we saw in Thanh Thi Nguyen; et. al., deepfake detection is still in its infantry, and future research is necessary for it to beat the evil of deepfake. An approach to improve performance of detection methods is to create a growing updated benchmark dataset of deepfakes to validate the ongoing development of detection methods.

Recently, Facebook Inc teaming up with Microsoft Corp and the Partnership on AI coalition have launched the Deepfake Detection Challenge to catalyse more research and development in detecting and preventing deepfakes from being used to mislead viewers [5][6]. Google meanwhile also published its own deepfake dataset, DeepFakeDetection dataset [7].

METHODOLOGY

a. Deepfake Dataset

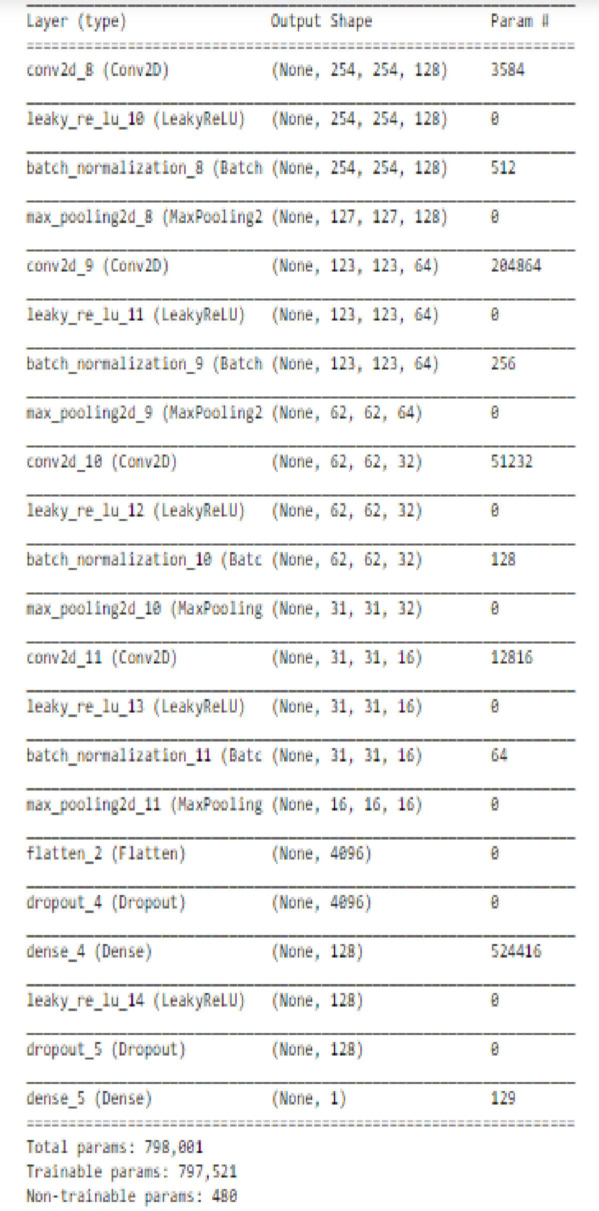
Now the major issue with us is to find a suitable dataset because the whole model is dependent on the quality of dataset we use, so for generating the deepfake dataset we have used videos provided in the DeeperForensics1.0 dataset. After that the another major thing is to extract the faces from the videos we have in the dataset, so which when we stumbled around we got to know some inbuilt face detection algorithms in OpenCV library of Python.So we took out frames from each video and then used face detection algorithm in OpenCV on them to take out the part of images which contains the faces. The next step is to search for Real Images for which we have used the dataset provided on GitHub where images where taken from Flickr-Face-HQ-Dataset[8] and resize them to a height of 600 pixels and width of 600 pixels. The total size of complete dataset is around 65,000 images shuffled properly which cannot be directly uploaded on Google Drive to make them use with Google Colab. So we have resize all the images to 170 pixels height and 170 pixels width to reduce the space they need to get stored and also divided them into 7 batches so that it will not consume all of the RAM storage.

Set	Fake Class	Real Class	Total
Train	26345	23654	49999
Val	7973	6965	14938
		Total	64937

Table1: Cardinality of each class in the studied dataset.

b. Algorithm

Our model is a simple CNN model, which uses convolutional layers, max-pool layers, dropout layers and dense layers finally culminating with a single dense layer of 1 unit with an activation of sigmoid function. The model in total has 798,001 parameters, and performs satisfactorily with a validation accuracy of 93%. Figure 2 represents a summary of the same model, showcasing each step and layer of the model, along with the trainable parameters in each layer



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 254, 254, 128)     3584
_________________________________________________________________
leaky_re_lu_10 (LeakyReLU)   (None, 254, 254, 128)     0
_________________________________________________________________
batch_normalization_8 (Batch (None, 254, 254, 128)     512
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 127, 127, 128)     0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 123, 123, 64)      204864
_________________________________________________________________
leaky_re_lu_11 (LeakyReLU)   (None, 123, 123, 64)      0
_________________________________________________________________
batch_normalization_9 (Batch (None, 123, 123, 64)      256
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 62, 62, 64)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 62, 62, 32)        51232
_________________________________________________________________
leaky_re_lu_12 (LeakyReLU)   (None, 62, 62, 32)        0
_________________________________________________________________
batch_normalization_10 (Batc (None, 62, 62, 32)        128
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 31, 31, 16)        12816
_________________________________________________________________
leaky_re_lu_13 (LeakyReLU)   (None, 31, 31, 16)        0
_________________________________________________________________
batch_normalization_11 (Batc (None, 31, 31, 16)        64
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 16, 16, 16)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 4096)              0
_________________________________________________________________
dropout_4 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense_4 (Dense)              (None, 128)               524416
_________________________________________________________________
leaky_re_lu_14 (LeakyReLU)   (None, 128)               0
_________________________________________________________________
dropout_5 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 129
=================================================================
Total params: 798,001
Trainable params: 797,521
Non-trainable params: 480
```

Figure2: Model Summary

c.VGG16 Model

VGG16[9] is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU's. The model in total has 14,739,777 parameters, and performs with a validation accuracy of 97%.

d.Xception Model

Xception Model[10] is proposed by Francois Chollet. Xception is an extension of the inception Architecture which replaces the standard Inception modules with depthwise Separable Convolutions. The model in total has 20,992,553 parameters, and performs with a validation accuracy of 95.5%.

e.ResNet50 Model

ResNet50[11] is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations. It is a widely used ResNet. Because of the framework that ResNets presented it was made possible to train ultra deep neural networks meaning that network can contain hundreds or thousands of layers and still achieve great performance. The model in total has 23,638,913 parameters, and performs with a validation accuracy of 97.6%.

f.EfficientNetB7 Model

For EfficientNet[12], its main building block is mobile inverted bottleneck MBConv, which was first introduced in MobileNetV2. By using shortcuts directly between the bottlenecks which connects a much fewer number of channels compared to expansion layers, combined with depthwise separable convolution which effectively reduces computation by almost a factor of k2, compared to traditional layers The model in total has 64,223,128 parameters, and performs with a validation accuracy of 96.5%.

RESULT

ModelName	ValidationAccuracy
Xception	0.955
EfficientNetB7	0.963
RestNet50	0.976
VGG16	0.973
OurCNNModel	0.930

Table2: Performance of Deepfake Detection Models

In table 2, we compare our Sequential CNN Model with the other Deep Learning models. As visible in Table 2, our model performs on the deefake_database in comparison to other major models. Our model clearly outputs a competitive validation accuracy of nearly 93%.

All the pre-trained models are large networks, which require millions of images for training before they are able to output good results. This drawback of theirs can easily explain the experimental results. Our model, in comparison, can output great accuracy on small dataset itself. This advantage means that for small scale detection, our model is clearly better.

However, due to the small dataset, we can't determine if it's similarly feasible for large scale detections, such as for being used on social media sites, etc.

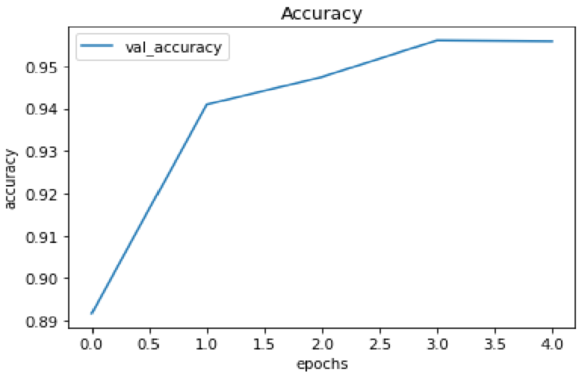


Figure3: VGG16 result

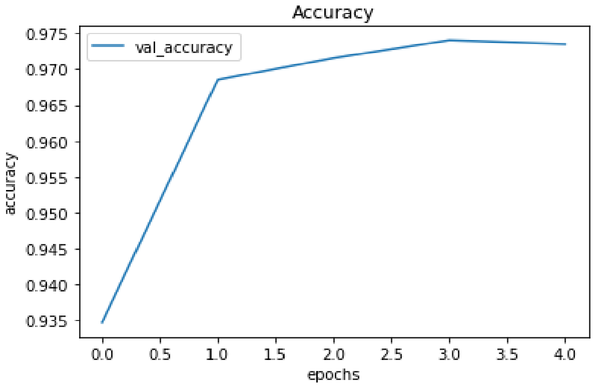


Figure4 : XCEPTION result

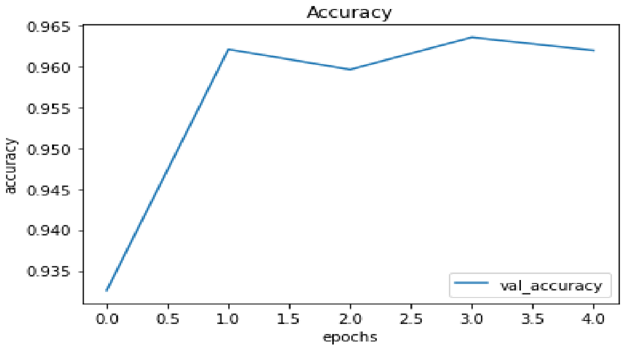


Figure5: RESNET50 result

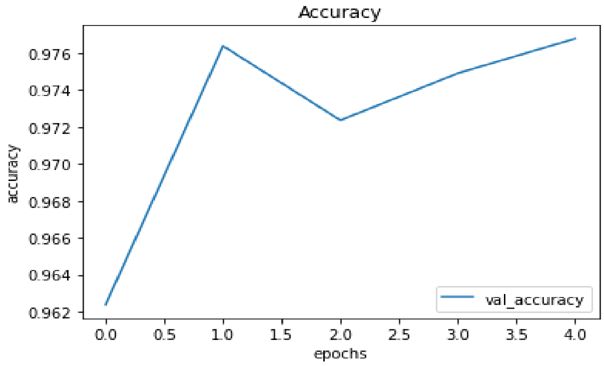


Figure6: EFFICIENTNETB7 result

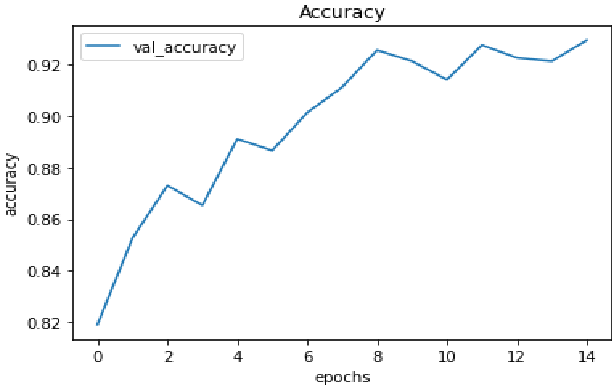


Figure7: Our Model result

CONCLUSION

Hence, here we show a small deepfake detection model which can accurately detect deepfakes. We also demonstrate how threatening deepfakes are to the society and must be stopped before they can really become a thing of concern. Deepfakes have repeatedly shown an astonishing capability of beating the detection models, while simultaneously being able to improve. While the detection models continue to seem to suffer in a way similar to how the black pieces in chess suffer against white, they can only be reactionary to the deepfakes. Lack of easily accessible datasets and hardware bottlenecks have also affected the research into the deepfakes detection and it is the need of the hour to properly look for their solutions.

FUTURE SCOPE

We are living in an age where people believe what they see on online platforms. These online platforms act as catalyst for people who wants to spread wrongful image of someone/group and deepfake can do that very effectively. This is the worst time as people are not much aware about the concept of deepfake and hence can be manipulated very easily. Now more than ever it is important for us to know the authenticity of something going online.

Therefore, these deepfake detection algorithms play a very crucial role in modern world. The Sequential CNN developed by us can act as stepping stone for future models to come. As this model has low overhead requirements, it may be able to merge with web apps in efficient manner.

Future possibilities in this research field are endless, and will continue to be so until the threat of deepfakes either becomes negligible or vanishes in its entirety.

REFERENCES

1. https://jonathan-hui.medium.com/how-deep-learning-fakes-videos-deepfakes-and-how-to-detect-it-c0b50fbf7cb9
2. M. Caldwell, J. T. A. Andrews, T. Tanay and L. Griffin "AI-enabled future crime," Crime Science, vol. 9, no. 1, p. 14 , 2020.
3. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets," Advances in neural information processing systems, pp. 2672-2680, 2014.
4. T. Nguyen, C. M. Nguyen, T. Nguyen, D. Nguyen and S. Nahavandi, "Deep Learning for Deepfakes Creation and Detection: A Survey," 2019.
5. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in IEEE, Honolulu, HI, USA, 2017.
6. C.C.Ferrer,B.Dolhansky, .Pflaum,J.Bitton,J.Pan and J.Lu, Deepfake Detection Challenge Results: An open initiative to advance AI," Facebook AI,12 June 2020.
7. G. R. a. A. G. J. Nick Dufour, N. Dufour, G. R. A. Gully and J. , "Contributing Data to Deepfake Detection Research," Google AI, 24 September 2019.
8. Flickr-Faces-HQ-Dataset: https://github.com/NVlabs/ffhq-dataset
9. VGG16: https://neurohive.io/en/popular-networks/vgg16/
10. Xception: https://medium.com/analytics-vidhya/image-recognition-using-pre-trained-xception-model-in-5-steps-96ac858f4206
11. ResNet50: https://iq.opengenus.org/resnet50-architecture/
12. EfficientNetB7: https://www.dlology.com/blog/transfer-learning-with-efficientnet/
13. Y. a. L. S. Li, "Exposing DeepFake Videos By Detecting Face Warping Artifacts," in IEEE, 2019.
14. G. Pierobon, "Visualizing intermediate activation in Convolutional Neural Net- works with Keras," Towards Data Science, 2 November 2018. :
15. C. Merrefield, "Deepfake technology is changing fast use these 5 resources to keep up," Journalist Resource, 27 June 2019.
16. https://lionbridge.ai/articles/three-types-of-deepfake-detection/