Electrical and Electronics Engineering
Semester Project
Bachelor's Exchange Semester - Spring 2022

Deep Learning for DeepFake Detection

Student: Berkay Güler

Supervised by: Yuhang Lu
Prof. Dr. Touradj Ebrahimi

June 5, 2022

Multimedia Signal Processing Group
EPFL

Abstract

Deepfakes are media contents created to deceive or mislead a target audience. These involve manipulation of faces of source identities with target identities, which is commonly referred to as face swap. While there are many methods to generate face swap deepfake content, a growing number of solutions are being developed by researchers and scientists to detect face swap deepfake content. One initiation that accelerated this development was Deep Fake Detection Challenge (DFDC), hosted on Kaggle, and created by Facebook in collaboration with industry leaders and academic experts. An analysis of the five best models submitted to this challenge is included in this research, as well as their key points. Moreover, the performance of the top three models submitted to this challenge is re-evaluated on another dataset with different metrics that are not used in DFDC. We particularly focused on metrics that reflect the models' usefulness in real-life deep fake detection. It is observed that the ranking of the models has been different from DFDC with the new dataset and evaluation metrics. We used a challenging test set with heavier augmentations to compare the robustness of each model. In this research, rankings of the models in different settings, their strengths and weaknesses are presented to provide a deeper analysis of DFDC.

Contents

1 Introduction

Deepfakes are media contents designed to disseminate fallacious information to a specific audience. One of the most common deepfake methods is face swap. Face swap is a term used to describe the process of combining the faces of source and target identities. While there are numerous approaches for creating face swap deepfake material, an increasing number of solutions for detecting face swap deepfake content are being created by researchers and scientists. The Deep Fake Detection Challenge [8], hosted on Kaggle and launched by Facebook in partnership with industry leaders and academic specialists, was one initiative that sped this development process.

There are two ways in which deepfakes might result in unintended consequences. The first involves deepfakes being mistaken for real videos, while the second involves real films being misidentified as fake. Humans, with their perceptual and cognitive abilities, are up to a point able to differentiate some deepfakes from real videos but the quality of the deepfake content plays a great role in that. Carefully fine-tuned and professionally made deepfake videos are becoming more and more difficult to detect for humans. [1]. In this research, the performance of deep learning detectors submitted to DFDC is re-evaluated. It is discovered and concluded that depending on which evaluation metric and test set is used, rankings of these models change greatly.

2 Deepfake Detection

Deepfake detection is a binary classification problem. Given a video or an image, the detector system aims to classify the content as being real or fake. Two main approaches can be used to design a detector system. One of them is first extracting some features from a video and then using a classifier [4]. The chosen features should have low intraclass, high interclass variance such that it would be possible to find suitable hyperplanes in n-dimensional space that could separate this space into two sub-spaces for real and fake videos. The other approach is to use deep learning models that perform both feature extraction and classification as a black-box detector system [2, 3].

3 Deepfake Detection with Deep Learning

One of the challenges a deep learning-based deepfake detector faces in the wild is that the test data does not most of the time come from the same distribution as the training data. Since the model is trained on a specific training set, even though the training set is purposefully chosen to reflect the distribution of deepfakes in reality, the performance of the detectors on unforeseen data will be almost always worse than the training set performance [4]. As the models become more robust they can generalize well enough to other distributions without over-fitting and under-fitting, and the model might produce decent results on the unforeseen test set. Creating robust models was a difficult challenge for participants of the DFDC.

Deep learning methods for deepfake detection can be analyzed under two main categories.

3.1 Frame Based DL Models

Frame-based models classify frames as containing deepfake or not. If the detector aims to classify images, then this approach can perform the task without any loss of information. However, if these models are used to classify videos, then it is inevitable that temporal information embedded in videos will be lost [4]. In that case, the detector system has to use an averaging function that takes the classification confidence of chosen frames of the video as input and a real number as output.

3.2 Sequence Based DL Models

Sequence-based models differ from frame-based models in that they can process the information gathered from multiple frames at the same time.

3D convolutional neural networks (3D-CNNs) are commonly used to design sequence-based models [5]. These models input a sequence of frames and process multiple frames at the same time depending on the volume of the filters between layers.

Another common way of using temporal information is to use various deep learning architectures which are variants of recurrent neural networks such as LSTMs, GRUs, and transformers [6, 7].

4 Deepfake Detection Challenge

Deepfake Detection challenge (DFDC) is hosted on Kaggle and attracted many researchers all around the globe. This challenge comes with its dataset with over 100.000 videos [9]. All videos of DFDC are labeled as fake if they contain face swaps, or real. One particular challenge of this competition was to train models robust enough to generalize to unforeseen data as the private test set contained heavy augmentations and distractors unlike the training set without any augmentations. Morover, %50 of the private test set were videos taken from the internet.

DFDC dataset is the largest dataset for fake swap deepfake detection. Also, it is seen as a good representative of the distribution of deepfake videos in real-life with its versatility of deepfake generation methods.

4.1 Ranking

2114 participants submitted more than 35.000 models and 5 winners are chosen depending on the performance in the average log-loss metric on the private test set that consists of 10.000 videos [8].

Leaderboard		
Name	Average Log-loss Score	Prize
Selim Seferbekov	0.42798	$ 500.000
/VM\	0.42842	$ 300.000
NtechLab	0.43452	$ 100.000
Eighteen years old	0.43476	$ 60.000
The Medics	0.43711	$ 40.000

Table 1: Private Test Set Ranking in DFDC

$$\text{average log-loss} = -\sum_{i=1}^{N} \left(y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})\right)$$

where N is the cardinality of the test set y_i is the output of the detector, and $\hat{y_i}$ is the ground truth labels for i^{th} input.

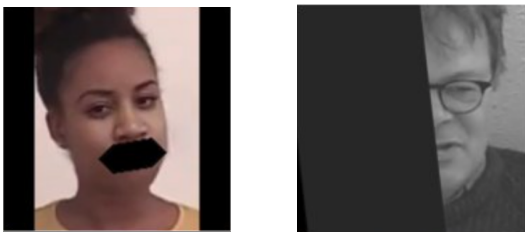
5 Analysis of Top Models

In this section, a brief overview of top 5 models in DFDC are given with an analysis focusing on their key points.

5.1 Selim Seferbekov

At the heart of this frame-based detector [22] lies an EfficientNet-B7 with MTCNN [18] as the face detector. Seferbekov used heavy augmentations supported by Albumentations [12] package

as well as dropout and cutout augmentations to increase the robustness of the detector for unseen test data. During inference time, 32 frames are cropped for each video and these frames are fed into the EfficientNet-B7 network. After obtaining the results for each frame, the detector outputs a real number $p \in [0, 1]$.



(a) dropout (b) cutout

Figure 1: Sample of augmentations used by Seferbekov

The EfficientNet-B7 network is pre-trained on ImageNet with Noisy Student [21].

5.2 /VM\

/VM\also used a purely frame-based detector [14] comprising of an ensemble of three networks.

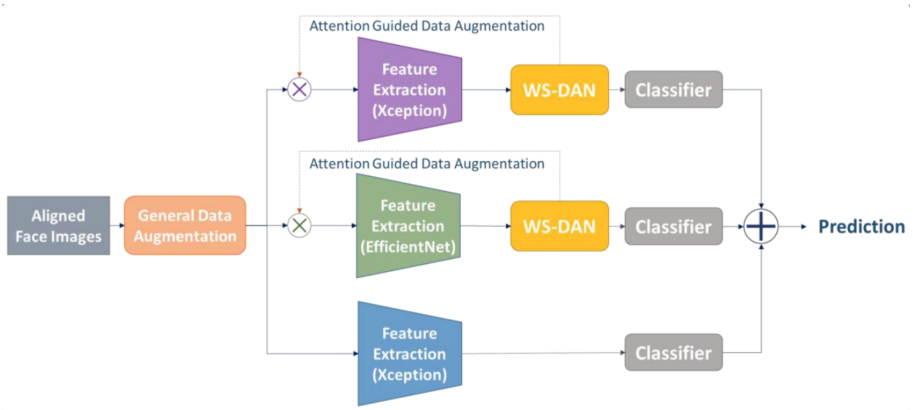
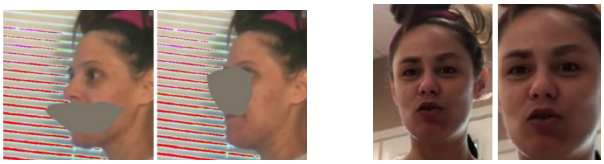


Figure 2: Flowchart of /VM\'s model

Two of these networks are trained with WS-DAN (Weakly Supervised Data Augmentation Network) [23] which performs data augmentation with help of attention maps. The third model is an Xception [10] network.



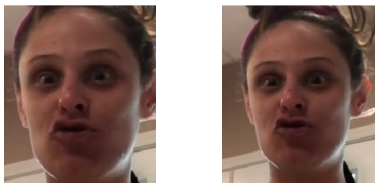
(a) attention drop (b) attention crop

Figure 3: Sample of augmentations performed by WS-DAN

5.3 NtechLab

NtechLab's solution [19] is a mixture of frame-based and sequence-based models that consist of an ensemble of three detection networks, all of which are based on EfficientNet-B7 architecture. These networks are trained on ImageNet with Noisy Student [21].

Although they share the same architecture, these models differ in their inputs and augmentations applied to the data they are trained on. Input to the first frame-based model is a frame that has almost only the face of interest with less background, whereas the input to the second frame-based model is a frame with more background information present.



(a) first model (b) second model

Figure 4: Samples of input to frame-based model

The third network has 3D convolution filters at each layer and inputs a sequence of seven frames for each video.

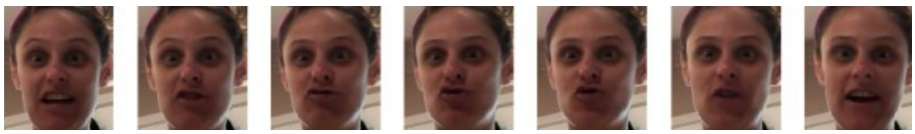


Figure 5: Sample of input to 3D-CNN

To increase the robustness of the network NtechLab used a mix-up technique [11] and video compression augmentations. The mix-up technique enables the generation of new training data as a linear combination of aligned fake and real faces. A parameter α is used to determine the amount of contribution of the real face in the generated linear combination using the formula below.

$$i^{th}\ mixed\ image = (1 - \alpha_i) real_i + \alpha_i fake_i$$

$$i^{th}\ mixed\ image\ label = \alpha_i$$

where α_i's are sampled from the beta distribution below with $\alpha = \beta = 0.5$

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\, du}$$

Figure 6: Illustration of mix-up technique on a sample data

A final prediction transformation function is applied to the prediction of the ensemble of three models.

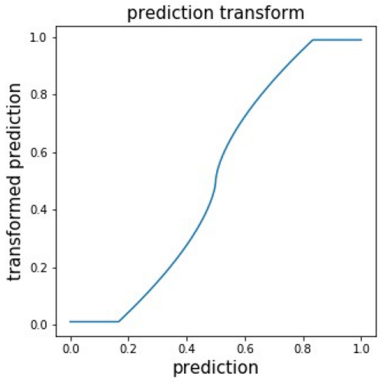


Figure 7: Tranformation for final prediction

5.4 Eighteen years old

Eighteen years old's detector is an ensemble of 3D-CNNs and 2D-CNNs with RetinaFace [17] as the face detector. This detector is a mixture of frame-based and sequence-based models. Xception, ResNet, and EfficientNet architectures are used for frame-based models while four different SlowFast networks [15] are used as 3D-CNN architectures.

5.5 The Medics

The Medics' detector [20] is a mixture of frame-based and sequence-based models. They used 7 3D-CNNs with 4 different architectures (I3D, 3D ResNet34, MC3, R2+1D), and 2 different resolutions (224 x 224, 112 x 112) together with a 2D-CNN based on SE-ResNeXT50. SE-ResNeXT50 is trained with heavy augmentations. Like many other methods, they used video compression augmentation to increase the detectors robustness.

Inputs to 3D-CNNs were short clips in which the background remained unchanged but the face inside the clip floated. This ensured having a lower number of false positives and improved performance greatly in that the model could focus more on the face with the background remaining unchanged.

6 Reassessment of Top-3 Models

Seferbekov's, NtechLab's, and /VM\'s detectors are implemented and their performance is evaluated on two different test sets with a range of metrics. The original weights that are used for the evaluation of these models on the DFDC private test set are used without further training.

6.1 Test Set 1: DFDC Preview Test Set

This test set consists of fake swap deepfake videos and original videos with total number of videos being 780. Low FPS, low resolution, and reduced encoding quality augmentations are applied to videos in this test set [13]. Videos are encoded in H.264 codec.

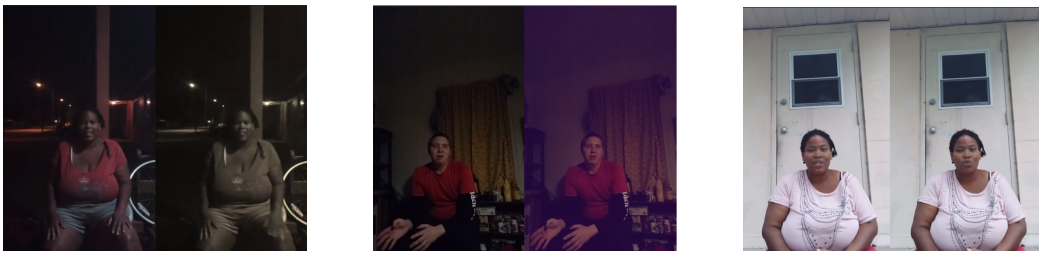
(a) (b)

Figure 8: Sample Fake Swaps from DFDC Preview

6.2 Test Set 2: Processed DFDC Preview Test Set

Another test set is generated with FFmpeg [16] from the original DFDC Preview test set with heavier augmentations in addition to already present ones. To generate the new dataset, all videos from the DFDC Preview test set are encoded in H.265. Moreover, one of the following filters/effects are applied to each video: vintage filter, sepia filter, conversion to greyscale, darkening, lightening, adding Gaussian blur, changing hue values, adding uniform and temporal noise, and sharpening.

These augmentations are chosen in that they reflect the distribution of common filters and effects applied to videos and images on social media platforms.



(a) greyscale filter (b) vintage filter (c) gaussian blur

Figure 9: Samples of applied filters to DFDC preview test set

7 Results

7.1 Evaluation Metrics

The deepfake detection problem comes with an extreme class imbalance. The number of real videos available, in reality, is much more than the number of fake videos. This results in some of the metrics used to evaluate binary classification models becoming potentially misleading [9]. Among those metrics accuracy, F1-score, and F2-score can be mentioned.

As an illustration, even a model that outputs the label "real" for any given input can produce an accuracy of more than %99 given the huge number of real videos this model will come across. One solution to this issue is to use metrics that are not affected by the class imbalance.

The average log-loss score is the only evaluation metric used originally to rank models submitted to DFDC. However, it is barely relatable to real-life performance of a detector because, in reality, the aim is to label videos correctly as fake or real in a binary domain rather than evaluating their output in a continuous domain. However, from another perspective, it might be also invaluable to know how confident the output of the model is.

In this section performance of each detector is assessed in average log-loss, weighted precision, recall, the area under the ROC curve, and accuracy.

7.1.1 Receiver Operating characteristic (ROC) Curve

ROC curves are especially useful to see the performance of each detector with varying threshold values. Here, detectors are assessed in their closeness to the ideal detector which can produce the least number of false positives and the maximum number of true positives at the same time.

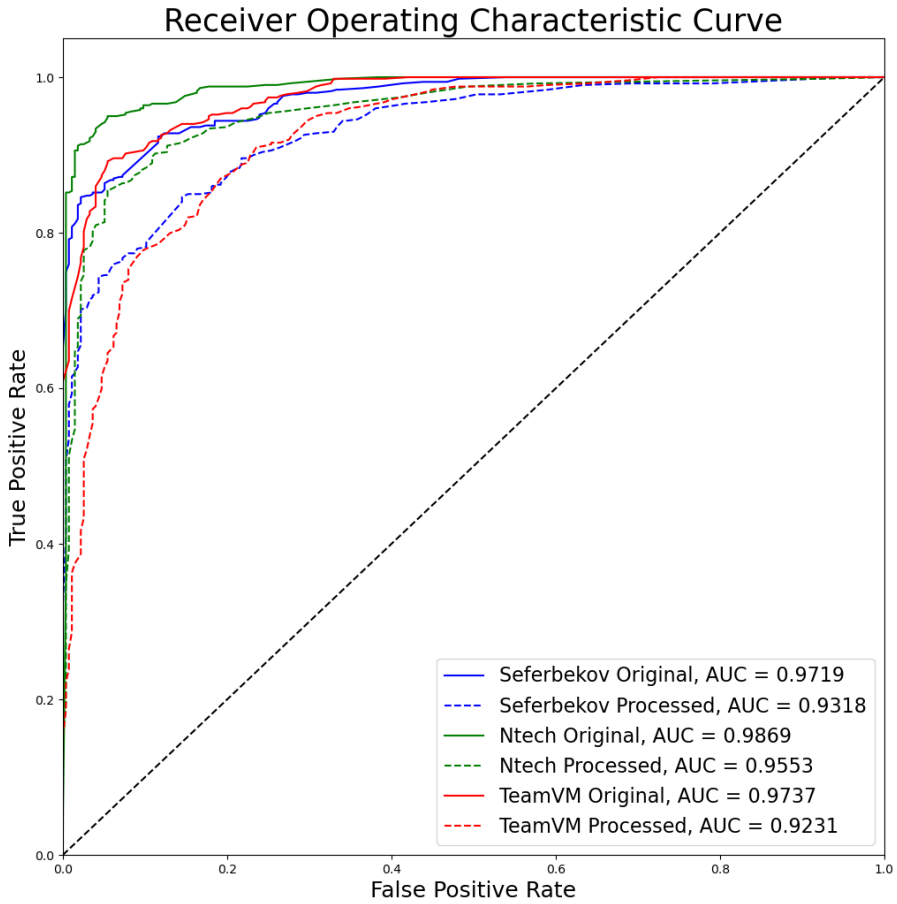


Figure 10: ROC Curve of each detector on both test sets

It is observed that the most robust detector is NtechLab's Model and the least robust one is /VM\'s detector if the difference in AUC score between the original DFDC preview and processed one is taken into account.

7.1.2 Weighted Precision vs. Recall Curve

The precision-recall curve is a common metric used in the assessment of classifiers. However, for this classification problem, weighted precision is used instead of precision.

$$weighted\ precision = \frac{TP}{TP+\alpha FP}$$

where TP is the number of true positives, FP is the number of false positives, and α is a parameter to penalize the model with a larger number of false positives. Because the ratio of $\frac{\#real}{\#fake}$ is around 2 in the DFDC Preview test set but much larger, in reality, weighting the number of false positives with $\alpha > 1$ helps the metric produce similar results to those it would produce when it was used in real-life.

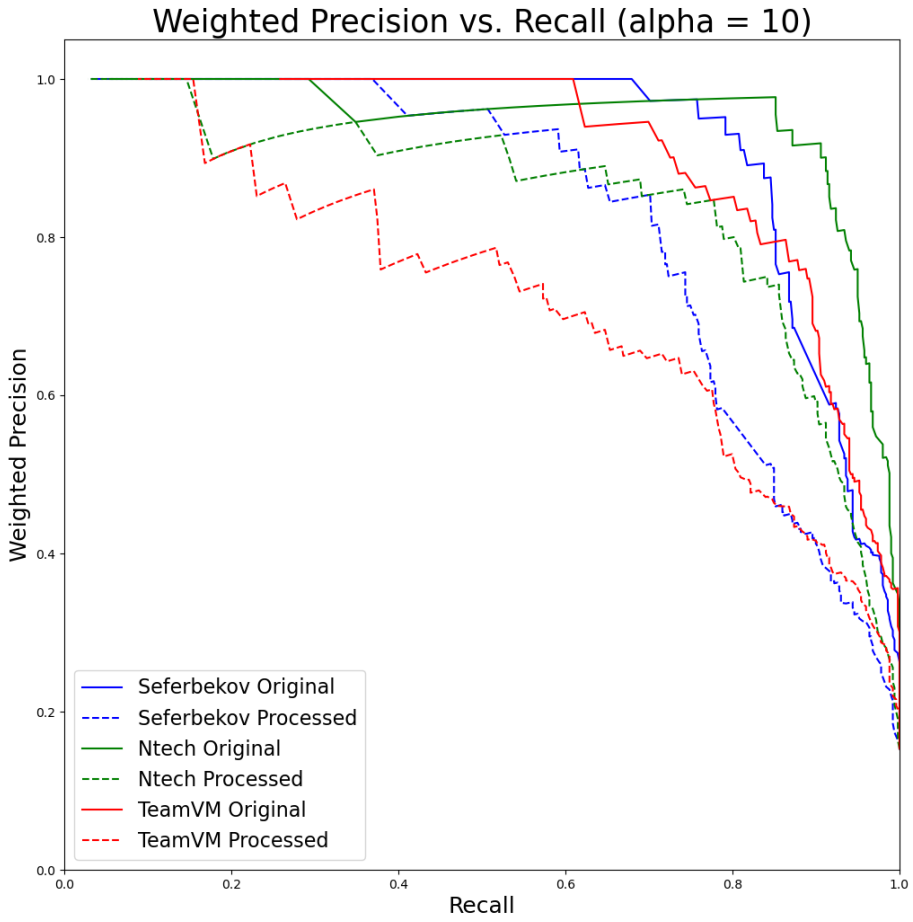


Figure 11: PR Curve of each detector on both test sets

It is observed that the NtechLab's detector performs better than other detectors at many threshold values. However, Seferbekov's detector beats that of NtechLab for precision values close to 1. /VM\'s detector performs poorly in terms of robustness while those of Seferbekov and NtechLab are performing similarly.

7.1.3 Average Log-loss Score

Even though detectors produce somewhat similar average log-loss scores, their ranking is different from that in DFDC private test set.

Test Set	Ranking/Model	Log-loss
Original DFDC Preview	1. Ntech	0.187
	2. /VM\	0.210
	3. Seferbekov	0.213
Processed DFDC Preview	1. Ntech	0.347
	2. Seferbekov	0.353
	3. /VM\	0.392

Table 2: Ranking in average log-loss score

We can conclude that Seferbekov's detector is the most robust one with /VM\being the least considering the difference between the log-loss scores obtained in the processed and original test set.

7.1.4 Accuracy

The ranking is different from that in the original DFDC if the metric used is accuracy.

Test Set	Ranking/Model	Accuracy
Original DFDC Preview	1. Ntech	0.930
	2. /VM\	0.908
	3. Seferbekov	0.893
Processed DFDC Preview	1. Ntech	0.852
	2. Seferbekov	0.827
	3. /VM\	0.822

Table 3: Ranking in accuracy

We obtain almost the same decrease in accuracy for each model when they are tested on the processed DFDC preview test set.

7.1.5 Area Under ROC Curve

The ranking is different from that in the original DFDC if the metric used is the area under the ROC curve.

Test Set	Ranking/Model	ROC AUC
Original DFDC Preview	1. Ntech	0.987
	2. /VM\	0.974
	3. Seferbekov	0.972
Processed DFDC Preview	1. Ntech	0.955
	2. Seferbekov	0.932
	3. /VM\	0.923

Table 4: Ranking in area under ROC curve

NtechLab's detector displays a more robust performance by obtaining the smallest decrease in area under the ROC curve among all models, while /VM\'s detector ranking last in robustness.

7.1.6 Weighted Precision

The ranking of detectors is different from that in the original DFDC if the metric used is weighted precision with $\alpha \in 10, 100, 250$.

Test Set	Ranking/Model	wP ($\alpha = 10$)	wP ($\alpha = 100$)	wP ($\alpha = 250$)
Original DFDC Preview	1. Ntech	0.918	0.529	0.310
	2. Seferbekov	0.686	0.179	0.080
	3. /VM\	0.569	0.117	0.050
Processed DFDC Preview	1. Ntech	0.798	0.283	0.136
	2. /VM\	0.590	0.126	0.054
	3. Seferbekov	0.584	0.123	0.053

Table 5: Ranking in weighted precision

Though NtechLab's detector performs best in both test sets it does not seem to be the most robust one. /VM\'s results, unexpectedly, improve on the processed test set, unlike other detectors, hence it is the most robust detector in weighted precision metric.

7.1.7 Recall

The ranking of detectors is different from that in the original DFDC if the metric used is recall.

Test Set	Ranking/Model	Recall
Original DFDC Preview	1. /VM\	0.928
	2. Ntech	0.900
	3. Seferbekov	0.874
Processed DFDC Preview	1. Ntech	0.790
	2. Seferbekov	0.788
	3. /VM\	0.778

Table 6: Ranking in recall

/VM\'s detector appears to be the least robust model in recall metric while Seferbekov's detector is the most robust one.

8 Conclusion

After the tests made in 10 different settings with 5 different metrics and 2 different test sets, it is observed that the ranking of the models has been different almost for each setting. Moreover, ranking obtained in the original DFDC has never occurred in the tests made in 10 settings.

The rankings of detectors are summarized for all settings below. The rankings used to create the table below are rankings in accuracy, weighted precision, area under the ROC curve, average log-loss score, and recall on both the original DFDC Preview test set and processed DFDC Preview test set.

Test Set	Ranking/Model	#times 1^{st}	#times 2^{nd}	#times 3^{rd}
Original DFDC Preview	1. Ntech	4	1	0
	2. /VM\	1	3	1
	3. Seferbekov	0	1	4
Processed DFDC Preview	1. Ntech	5	0	0
	2. Seferbekov	0	4	1
	3. /VM\	0	1	4

Table 7: Summary of rankings

#times n^{th} is the number of time a certain detector is ranked n^{th} in 5 different metrics

According to obtained results, /VM\'s detector suffers from a robustness issue and falls below other detectors when it encounters unforeseen data. NtechLab's detector outperforms other detectors in both test sets except for the recall metric on the original DFDC preview test set, proving it to be the best model on these test sets. As for Seferbekov's detector, it appears that, though it does not perform brilliantly, being relatively more robust than /VM\'s detector, it is ranked as the second-best model in the processed test set.

With the help of the results presented above, the importance of developing more robust models becomes prominent for providing sustainable solutions to the deepfake detection problem. More importantly, it can be concluded that ranking deepfake detectors on a single test set with one evaluation metric does not necessarily show the genuine success or quality of the detectors. To have a deeper insight into the capabilities and weaknesses of each detector, it is necessary, but still not sufficient, to use a realistic assessment framework with as many test sets as possible. On the other hand, if one knows or at least has a strong forecast of the data that a detector will mainly encounter, it can be sufficient to use the metric and test set that would reflect best the performance of candidate detectors.

References

[1] A. Deshmukh and S. B. Wankhade, "Deepfake detection approaches using Deep Learning: A Systematic Review," Intelligent Computing and Networking, pp. 293–302, 2020.

[2] "A review of deep learning-based approaches for deepfake content detection." [Online]. Available: https://www.researchgate.net/publication/358603507. [Accessed: 11-Jun-2022].

[3] H. A. Khalil and S. A. Maged, "Deepfakes creation and detection using Deep Learning," 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), 2021.

[4] "Deep learning for deepfakes creation and detection: A survey." [Online]. Available: https://www.researchgate.net/publication/336058980 [Accessed: 13-Jun-2022].

[5] "A video is worth more than 1000 lies. comparing 3DCNN approaches for detecting deepfakes," IEEE Xplore. [Online]. Available: https://ieeexplore.ieee.org/document/9320165. [Accessed: 16-Jun-2022].

[6] I. Amerini and R. Caldelli, "Exploiting prediction error inconsistencies through LSTM-based classifiers to detect deepfake videos," Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security, 2020.

[7] Y. Al-Dhabi and S. Zhang, "Deepfake video detection by combining Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN)," 2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE), 2021.

[8] "Deepfake Detection Challenge Dataset," Meta AI. [Online]. Available: https://ai.facebook.com/datasets/dfdc/. [Accessed: 13-Jun-2022].

[9] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (DFDC) dataset," arXiv.org, 28-Oct-2020. [Online]. Available: https://arxiv.org/abs/2006.07397. [Accessed: 13-Jun-2022].

[10] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," arXiv.org, 27-Apr-2018. [Online]. Available: https://arxiv.org/abs/1710.09412. [Accessed: 16-Jun-2022].

[12] Albumentations. [Online]. Available: https://albumentations.ai/. [Accessed: 13-Jun-2022].

[13] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (DFDC) preview dataset," arXiv.org, 23-Oct-2019. [Online]. Available: https://arxiv.org/abs/1910.08854. [Accessed: 13-Jun-2022].

[14] Cuihaoleo, "Cuihaoleo/Kaggle-DFDC: 2nd place solution for Kaggle Deepfake Detection Challenge," GitHub. [Online]. Available: https://github.com/cuihaoleo/kaggle-dfdc. [Accessed: 13-Jun-2022].

[15] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SLOWFAST networks for video recognition," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[16] FFmpeg. [Online]. Available: https://ffmpeg.org/. [Accessed: 13-Jun-2022].

[17] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[18] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, 2016.

[19] NTech-Lab, "NTech-Lab/deepfake-detection-challenge: 3rd place solution for the Deepfake Detection Challenge on kaggle," GitHub. [Online]. Available: https://github.com/NTech-Lab/deepfake-detection-challenge. [Accessed: 13-Jun-2022].

[20] Jphdotam, "JPHDOTAM/DFDC: Training codebase for our solution to Kaggle's Deepfake Detection Challenge," GitHub. [Online]. Available: https://github.com/jphdotam/DFDC. [Accessed: 16-Jun-2022].

[21] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves ImageNet Classification," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[22] Selimsef, "Selimsef/dfdcdeepfakechallenge: A prize winning solution for DFDC Challenge," GitHub. [Online]. Available: https://github.com/selimsef/dfdcdeepfakechallenge. [Accessed: 13-Jun-2022].

[23] T. Hu, H. Qi, Q. Huang, and Y. Lu, "See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification," arXiv.org, 23-Mar-2019. [Online]. Available: https://arxiv.org/abs/1901.09891. [Accessed: 13-Jun-2022].