*Artificial Intelligence and Social Computing,* Vol. 122, 2024, 137–144
https://doi.org/10.54941/ahfe1004650

# Development of Neural Networks for Deepfake Recognition

**Alina Latipova[1] and Maria Yadryshnikova[2]**

[1]University of Central Lancashire, Preston, PR1 2HE, UK
[2]South Ural State University (NRU), Chelyabinsk, Lenina Ave, 76, 454080, Russian Federation, Russia

## ABSTRACT

Today, deepfake technology is well developed and widely used for entertainment purposes in social networks, cinema and television. However, such falsification can be used maliciously and cause harm to the person depicted in the video. Therefore, tools for identifying such malicious deepfakes are needed, which can be built using neural networks. This paper discusses the architectures of neural networks that make it possible to identify forged videos. Public data sets, described in the article, were used as initial data for training the networks. The article also discusses the pre-processing of video data. Much attention is paid to the ensemble approach, which combines the results of several trained neural network models.

**Keywords:** Deepfake recognition, Deep learning, Artificial intelligence, Ensemble of neural networks

## INTRODUCTION

Deepfake recognition methods can be divided into two main groups (Passos et al., 2024):

- using spatial properties of video only (frame-dependence level)
- using spatial and temporal properties of video together.

For the first group, every frame of video is analyzed individually for further feature extraction and classification, then prediction can be obtained as combination of classification results of examined frames. This category of algorithms may fail in capturing unnatural artefacts because the whole video composition is not taken into account. The second group uses sequence of frame spatial properties to find inconsistencies over time by applying intercorrelation analysis (Passos et al., 2024). The problem is that extracting temporal features and processing them may require a lot of time and other computational resources.

Nowadays, neural networks are dominating other approaches of deepfake detection (e.g. image noise estimation or lighting analysis) (Yadryshnikova et al., 2023). Many researchers use combinations and ensembles of networks to improve effectiveness of recognition.

In (Afchar et al., 2018), the authors present a method to detection face tampering by Meso-4 and MesoInception-4 architectures which are

convolutional networks with pooling, dropout, ReLU activation function, batch normalization and fully connected layers using dropout for regularization. The authors tried to minimise number of parameters and simplify structure of networks without significant deterioration of quality, and their models have a high accuracy of over 90% for Deepfake and Face2Face datasets. Also, they used average score of frames for video classification, the same approach of scoring was used in our model.

The authors of (Zhou et al., 2017) propose to use a two-stream face tampering detection technique. The first stream is a GoogLeNet convolutional neural network (CNN) trained to detect tampering artifacts, and the second stream is a patch based triplet network to capture traces left by in-camera processing (e.g. CFA) and local noise residuals. This thread consists of steganalysis features extractor and support vector machine (SVM). This approach provided the best value above 0.92 of the area under the curve (AUC metric) for the Receiver Operating Characteristic Curve (ROC) over against other methods from the article.

In (Coccomini et al., 2022), combination of Vision Transformers with CNN EfficientNet B0 was introduced whereas EfficientNet B0 is a feature extractor. Their model achieved AUC over 95% for the DeepFake Detection Challenge.

## INITIAL DATA

There are two ways to obtain initial data for neural networks. First, you can make your own dataset by using generative adversarial networks (GANs), original and forged video and creating deepfakes out of them of various quality. But it takes a lot of time to build your own GAN and generate a large amount of data for training. Another way is to use ready-made datasets for deepfakes detection. Fortunately, a lot of such datasets are available (mostly as initial data for deepfake recognition contests). We decided to take the given below four public datasets to provide necessary level of variety and amount for data (Yadryshnikova et al., 2023).

### DeepfakeTIMIT Dataset (Korshunov et al., 2023)

This dataset (Korshunov et al., 2023) represents a database of videos with face swapping. The GAN-based approach based on the autoencoder-based Deepfake algorithm was used to swap faces.

This dataset contains 620 videos of swapped and original faces. We took 320 videos from it in dataset for this work.

### Celeb-DF Dataset (Li et al., 2020)

This dataset (Li et al., 2023) was created by contestants by the DeepFake Game Competition (DFGC) from publicly available YouTube video clips of 59 famous people of different ages, ethnic groups and genders.

Celeb-DF dataset includes 590 original YouTube videos with subjects, and 5639 corresponding deepfake videos. DeepFake synthesis method was used to generate deepfakes. We took 1000 photos with original real data and 1000 photos with fake faces.

### DeepFake Detection Dataset (Rössler et al., 2019)

This dataset (Rössler et al., 2023) was created by Google with collaboration with JigSaw.

It contains over 363 original sequences from 28 paid actors in 16 different scenes The dataset contains over 3000 manipulated videos from 28 actors in various scenes. Four methods of manipulation were used: Deepfakes, Face2Face, FaceSwap and NeuralTextures. We took 1000 photos with original real data and 1000 photos with fake faces.

### DeepFake Detection Challenge Dataset (Dolhansky et al., 2018)

This dataset (Dolhansky et al., 2023) was used in the DeepFake Detection Challenge by Kaggle from Google. The dataset was created by Facebook with 3,426 paid actors.

The dataset contains more than 100,000 clips with real original and manipulated videos. Swapping was produced with GAN-based and non-learned methods. We took 1000 photos with original real data and 1000 photos with fake faces.

## ENSEMBLE DEVELOPMENT

### Data Pre-Processing

Pre-processing technique depends on what kind of deepfake detection methods will be used (spatial or temporal) and structure of your data. We decided to use spatial frame-level approach that is why we must split the video into frames. We used the following algorithm:

- take one frame from each second,
- detect faces on these frames(by a pre-trained MTCNN Multi-task Cascaded Convolutional Neural Network).

MTCNN detector crops pictures so that they contain only a face. Then it puts these data into folders with names indicating their class. After that it splits the data into training, test and validation sets (Zhang et al., 2016; 2023).

Video pre-processing to a series of photos from the dataset is shown on Figure 1.

Characteristics of sample for the research is given in Table 1 which shows almost equal proportions of deepfakes and real photos after data pre-processing.

**Table 1.** Description of used dataset.

| Sample Type \ Parameters | Number of photos | Ratio |
|---|---|---|
| training sample | 21,632 | 80% |
| validation sample | 2,704 | 10% |
| test sample | 2,704 | 10% |

**Figure 1:** Making photos out of a video.

## NEURAL NETWORK ENSEMBLE ARCHITECTURE

It was decided to use two models Inception-ResNet-v2 and Xception, tune and combine them into an ensemble for better performance (see Figure 2).

Both models were modified by adding fully connected layers. The output layer was also changed to a fully connected layer. All layers (except the output one) have the ReLU activation function. The output layer has a sigmoid activation function to obtain result from 0 to 1, which can be treated as the probability of being related to any class for every frame. Then we calculate average value of frame outputs for the video, and closer the final value is to 1, the higher the probability of the video being related to the class of real original videos.

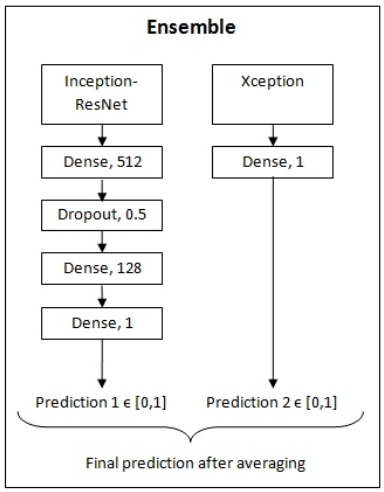


**Figure 2:** Ensemble structure.

There were 50 epochs to train neural networks. We used a callback to save the model with the best validation accuracy score. Binary cross-entropy was used as a validation loss or error. The validation loss of the Inception-ResNet-v2 model stopped falling after epoch 32, the validation loss of the Xception model stopped after epoch 31 (see Figure 3).

## ACCURACY EVALUATION FOR PHOTOS

The confusion matrix is given in Table 2 and it provides accuracy metrics (Chauhan et al., 2023):

- TP – true-positive,
- FP – false-positive,
- TN – true-negative,
- FN – false-negative.

These metrics can be used to calculate true-positive ratio (TPR) and true-negative ratio (TNR).

We used equal weights 0.5 for both models outputs of ensemble to calculate the metrics.

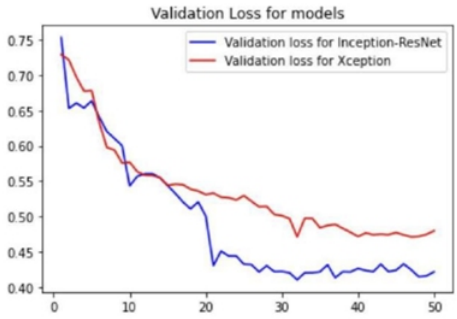


**Figure 3:** Validation analysis.

**Table 2.** Confusion matrix of ensemble on test data (average output).

| Metrics\Actual classes | | Positive | Negative |
|---|---|---|---|
| Predicted classes | Positive | TP = 1451 | FP = 46 |
| | Negative | TN = 198 | FN = 1009 |

It comes from Table 2 that TPR = 0.8799, TNR = 0, both value exceed 0.85 which indicates high level of accuracy for photos.

### Accuracy Metric for Video

Accuracy metric must be also calculated for video. E.g., there are two videos from the trained dataset: video 1 that is NOT a deepfake (actual class is 0) and video 2 that IS a deepfake (actual class is 1). Both video were split in six photos. Each photo was analyzed by Inception-ResNet-v2 and Xception.

Then two variants of ensembles which output is weighed sum of the both networks. Results are presented in Table 3 and Table 4.

As you can see from Table 3 and Table 4, final classification of video by ensemble depends on method of combining two networks and choosing of aggregation metric for the set of photos (e.g. average value or median).

Table 5 represents accuracy metrics for different settings of ensemble.

**Table 3.** Video1 output analysis (class 0).

| Output \ Photo | 1 | 2 | 3 | 4 | 5 | 6 | Average | Median |
|---|---|---|---|---|---|---|---|---|
| Inception-Resnet-v2 | 0.3097 | 0.2997 | 0.4552 | 0.4290 | 0.3269 | 0.0793 | 0.3166 | 0.3183 |
| Exception | 0.2456 | 0.5557 | 0.4137 | 0.1276 | 0.6725 | 0.2461 | 0.3769 | 0.3299 |
| Ensemble (weight for both is 0.5) | 0.2776 | 0.4277 | 0.4345 | 0.2783 | 0.4997 | 0.1627 | 0.3468 | 0.3241 |
| Ensemble (weight for Inception is 0.6 and for Exception is 0.4) | 0.2841 | 0.4021 | 0.4386 | 0.3084 | 0.4651 | 0.1460 | 0.3407 | 0.3229 |

**Table 4.** Video2 output analysis (class 1).

| Output \ Photo | 1 | 2 | 3 | 4 | 5 | 6 | Average | Median |
|---|---|---|---|---|---|---|---|---|
| Inception-Resnet-v2 | 0.8866 | 0.7541 | 0.8954 | 0.5994 | 0.6067 | 0.8239 | 0.7610 | 0.7890 |
| Exception | 0.7923 | 0.6139 | 0.3633 | 0.9530 | 0.6674 | 0.7086 | 0.6831 | 0.6880 |
| Ensemble (weight for both is 0.5) | 0.8395 | 0.6840 | 0.6294 | 0.7762 | 0.6371 | 0.7663 | 0.7221 | 0.7385 |
| Ensemble (weight for Inception is 0.6 and for Exception is 0.4) | 0.8489 | 0.6980 | 0.6826 | 0.7408 | 0.6310 | 0.7778 | 0.7298 | 0.7486 |

**Table 5.** Accuracy metrics for test dataset.

| Setting of ensemble | Accuracy |
|---|---|
| Ensemble (weight for both is 0.5) with average aggregation function | 0.9105 |
| Ensemble (weight for both is 0.5) with median aggregation function | 0.9243 |
| Ensemble (weight for Inception is 0.6 and for Exception is 0.4) with average aggregation function | 0.8986 |

It comes from Table 5 that median aggregation function may be a better option than average. It can be explained that median is more resistant to outliers than average value. Also it might be efficient to assign equal weights to networks outputs in ensemble since it avoids the dominance of one of the neural networks.

## Software Implementation

The given ensemble approach was implemented in software as bot for Telegram messenger (see Figure 4). The user can upload his/her video to the bot, this video must be pre-processed and then classified by ensemble of neural networks as original or deepfake. The main program of the bot consists of DataPreprocess and DetectorMTCN components to pre-process the

uploaded video and Ensemble of Neural Networks component for deepfake recognition of the submitted and pre-processed video (Yadryshnikova, 2022).

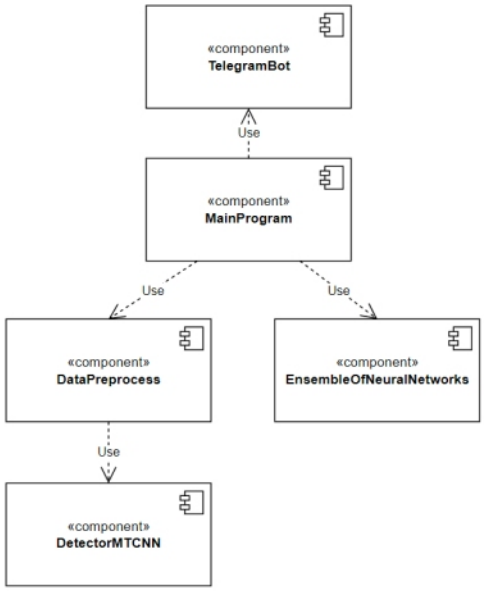


**Figure 4:** Component diagram of telegram-bot implementing ensemble approach.

## CONCLUSION

From 2020 GANs are getting more and more advanced and so that deepfakes are becoming more realistic videos. Therefore, to be effective, recognition systems must be better than adversarial networks discriminators. One of solutions is to increase the training sample and use the latest examples of deepfakes to improve quality of classification. The problem is that existing open datasets don't include most recent high-quality deepfakes because there is always a time gap between acquiring and publishing data, and it is obvious that commercial companies, which are successful at developing deepfakes, do not tend to share their data.

To solve the problem of insufficient test sample, you can create your own high-quality deepfakes for model training. Also, it is necessary to consider different architectures of neural network ensembles, which will improve recognition efficiency. In addition, the temporal nature of the video must be taken into account, not just the individual image frames (Choi et al., 2024).

## REFERENCES

Afchar, D., Nozick, V., Yamagishi, J., Echizen, I. (2018) "MesoNet: a Compact Facial Video Forgery Detection Network," proceedings of 2018 IEEE International Workshop on Information Forensics and Security, Hong Kong, pp. 1–7, DOI:10.1109/WIFS.2018.8630761.

Chauhan, N. (February 01, 2023) Model Evaluation Metrics in Machine Learning. Web-site: https://www.kdnuggets.com/2020/05/model-evaluation-metrics-machine-learning.html.

Choi, J., Kim, T., Jeong, Y., Baek, S., Choi, J. (March 11, 2024) Exploiting Style Latent Flows for Generalizing Deepfake Video Detection. Web-site: https://arxiv.org/pdf/2403.06592v1.pdf.

Coccomini, D. A., Messina, N., Gennaro, C., Falchi, F. (2022) "Combining EfficientNet and Vision Transformers for Video Deepfake Detection," proceedings of Image Analysis and Processing–ICIAP 2022, Lecture Notes in Computer Science, Volume 13233, DOI:10.1007/978-3-031-06433-3_19.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., Ferrer, M. (February 01, 2023) Deepfake Detection Kaggle Dataset. Website: https://www.kaggle.com/c/deepfake-detection-challenge.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., Ferrer, M. (February 01, 2023) The DeepFake Detection Challenge Dataset. Website: https://arxiv.org/abs/2006.07397.

Korshunov, P., Marcel, S. (February 01, 2023) DeepFake TIMIT Dataset. Website: https://www.idiap.ch/en/dataset/deepfaketimit.

Korshunov, P., Marcel, S. (February 01, 2023) DeepFakes: a New Threat to Face Recognition? Assessment and Detection. Web-site: https://arxiv.org/abs/1812.08685

Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S. (2020) "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics," in Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, pp. 3204–3213, DOI:10.1109/CVPR42600.2020.00327.

Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S. (February 01, 2023) DeepFake Celeb-DF Dataset. Web-site: https://github.com/yuezunli/celeb-deepfakeforensics.

Passosa, L., Jodasa, D., Costaa, K., Souza J'unior, L., Rodrigues, D., Del Serb, J., Camacho, D., Papa, P. A Review of Deep Learning-based Approaches for Deepfake Content Detection. Web-site: https://arxiv.org/pdf/2202.06095.pdf.

Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M. (2019) Deepfake Detection Google Dataset. Web-site: https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.

Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M. (2019) "FaceForensics++: Learning to Detect Manipulated Facial Images," proceedings of International Conference on Computer Vision (ICCV), Seoul, Korea (South).

Yadryshnikova, M., Latipova, A. (May 17, 2023) "Application of Ensembles of Neural Networks for Deepfake Recognition," proceedings of 2023 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russian Federation, 2023, pp. 244–246, doi: 10.1109/USBEREIT58508.2023.10158894.

Yadryshnikova, M. (2022) Development of a telegram bot for recognition of deepfakes in video recordings from using neural networks. Website: https://omega.sp.susu.ac.ru/publications/bachelorthesis/2022_403_yadryshnikovamv_slides.pdf.

Zhang, K., Zhang, Z., Li, Z., Qiao, Y. (2016) Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Processing Letters, 2016, Volume 23(10), pp. 1499–1503.

Zhang, K., Zhang, Z., Li, Z., Qiao, Y. (February 01, 2023) MTCNN. Web-site: https://github.com/ipazc/mtcnn.

Zhou, P., Han, X., Morariu, V., Davis, L. (2017) "Two-Stream Neural Networks for Tampered Face Detection," proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, pp. 1831–1839, DOI:10.1109/CVPRW.2017.229.