DeepFakes Detection Lab - Report

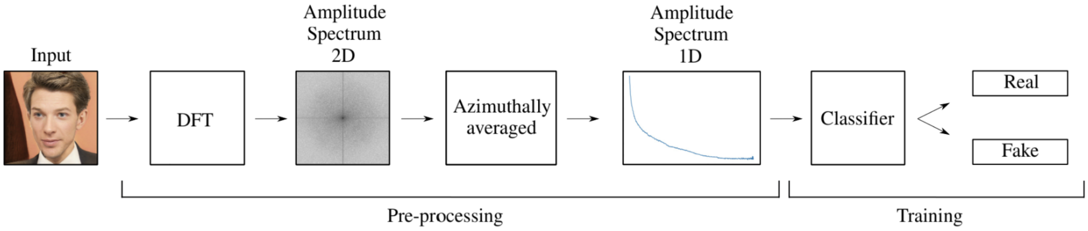
Peter Kun, Sanjay Kumar, Baglan Aitu

Task 1 – Intra-database analysis: The goal of this task is to develop and evaluate DeepFake detection systems over the same database (intra-database analysis). In this Task 1, you should use only the UADFV database included in the folder named **"Task_1".** This database is divided into "development" and "evaluation" datasets.

Important information: you should train your system using only the development dataset. The evaluation dataset must be considered only for the final evaluation of the system (after training).

1.a) Provide all details (including links or references if needed) of your proposed DeepFake detection system:

In task 1, we are using the approach of ***Classical frequency domain analysis*** followed by a basic classifier.



As it is shown in the pipeline above, the method contains two main blocks. They are:

1. Feature extraction
2. Training

In the feature extraction part, an image is converted from spatial to frequency domain by using **Discrete Fourier Transform.** The reason for that is in this domain, each frequency of the signal (image) carries information about its corresponding amplitude and phase which are considered as major features in this work. At the output, we obtain a 2D amplitude spectrum of the image.

After that, we use **azimuthal averaging** to compress the number of features. It gathers an average similar frequency feature into a vector of features, **1D amplitude spectrum.** It is a robust feature vector which preserves essential information of features for classification tasks.

Once we obtain all necessary features, we can use basic classifiers (logistic regression, support vector machine, k-means, etc.) in order to distinguish whether the image is real or fake.

References:

- Paper: https://arxiv.org/pdf/1911.00686.pdf
- Code: https://github.com/cc-hpc-itwm/DeepFakeDetection

1.b) Provide all details of the development/training procedure followed and the results achieved using the “development” dataset. Show the results achieved in terms of Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC).

In this task, we were provided with 2 different image datasets:

1. Development (for training)
2. Evaluation (for testing).

Before training the model with the given images, it performed face detection by cropping the part of image surrounding the face. The face detection was executed by MTCNN by facenet-pytorch. In order to avoid the false detections, only the biggest blob was considered as the final detection. In the case of the development dataset, the images with undetected faces were excluded (2-3). After that we implemented the steps discussed above in 1(a).

There are several hyperparameters of feature extraction part:

- epsilon = 1e-8 (slack variable).
- N = 300 (the length of the feature vector)

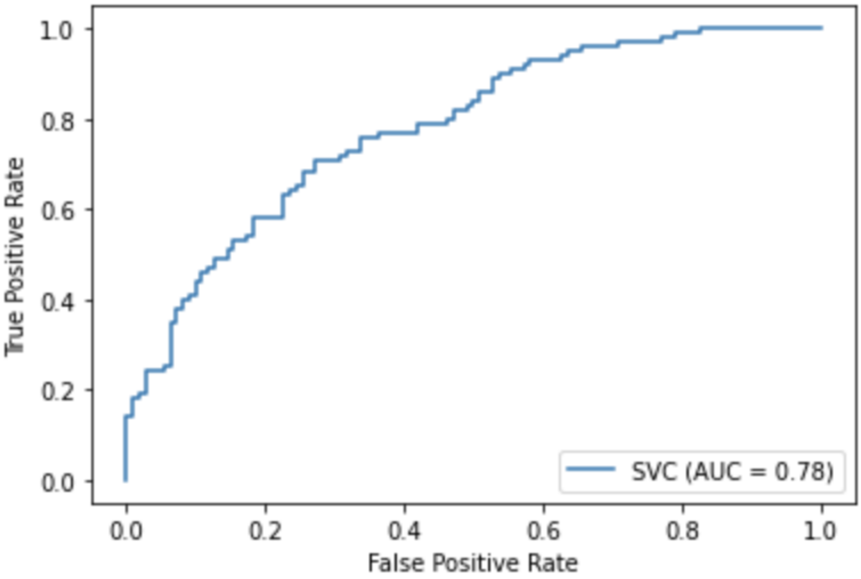
We tried tuning it in and did not get any improvements, and left as it was suggested by the authors of the paper.

In the classification part, based on the discriminatory problem of the given task, we decided to stick to the **Support Vector Machine.** One of the major reasons was the target of the SVM as its formulation is to produce a model (based on the training data) which will identify an optimal separating hyperplane, maximizing the margin between different classes.

To implement it in python, “scikit-learn” framework was used where we had an option to opt for the type of kernel. We implemented all of them which AUC results is shown below:

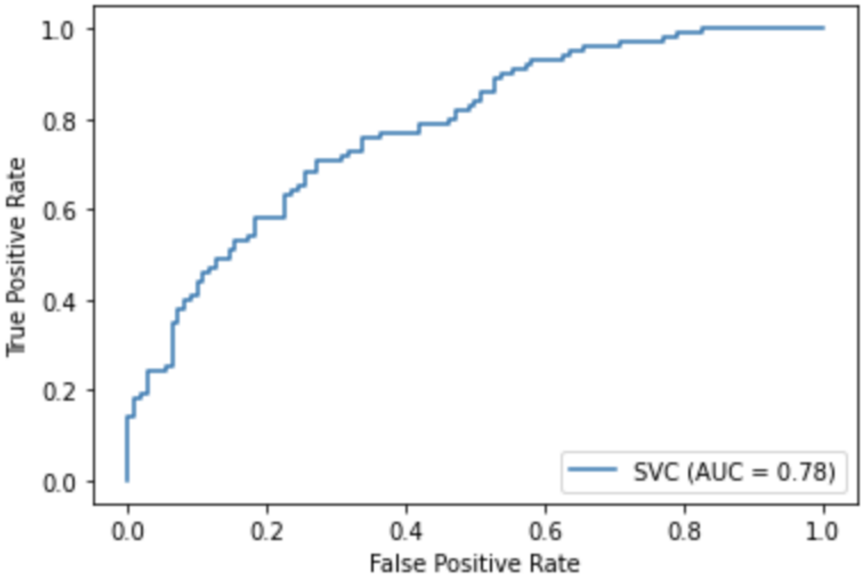
- poly = 0.78
- linear = 0.71
- rbf = 0.73
- sigmoid = 0.42

By setting above mentioned parameters, we achieved the **AUC = 0.78.**



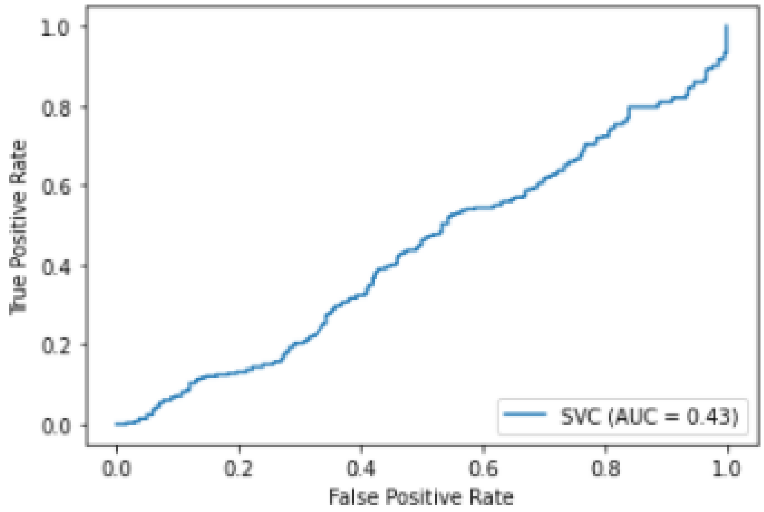
1.c) Describe the final evaluation of your proposed DeepFake detection system and the results achieved using the “evaluation” dataset (not used for training). Show the results achieved in terms of ROC curve and AUC. Provide an explanation of your results.

Our final obtained score is **0.78**. It was achieved by a SVM classifier with a **polynomial** kernel. Face detection was applied as a preliminary step in order to extract the important information of the images. The length of the feature vectors from each image was **300**, in which the frequency domain information was stored.



Task 2 – Inter-database analysis: The goal of this task is to evaluate the DeepFake detection system developed in Task 1 with a new database (not seen during the development/training of the detector). In this Task 2, you should use only the Celeb-DF database included in the folder named **"Task_2_3".** You only need to evaluate your fake detector developed in Task 1 over the evaluation dataset of Celeb-DF, not training again with them.

2.a) Describe the results achieved by your DeepFake detection system developed in Task 1 using the "evaluation" dataset of the "Task_2_3" folder. Show the results achieved in terms of ROC curve and AUC. Provide an explanation of your results in comparison with the results of Task 1.



The result is **0.43** AUC score for the Task 2_3 dataset (by changing the labels, it is 0.57 AUC in the case of binary classification), thus there can be seen a significant degradation in the performance. It can be explained with the large inter-class variability between the two dataset. In other words, Celeb-DF dataset's fake images are closer to real (high quality) comparatively with the fake images of task 1 dataset.

Task 3 – Inter-database proposal: The goal of this task is to improve the DeepFake detection system originally developed in Task 1 in order to achieve better inter-database results.

Important information:

- Development: no restrictions, you can use public software and databases if you like.
- Evaluation: you must consider the same evaluation dataset of Task 2 (i.e., the evaluation dataset included in "Task_2_3" folder).

3.a) Describe the improvements carried out in your proposed DeepFake detection system in comparison with Task 1.

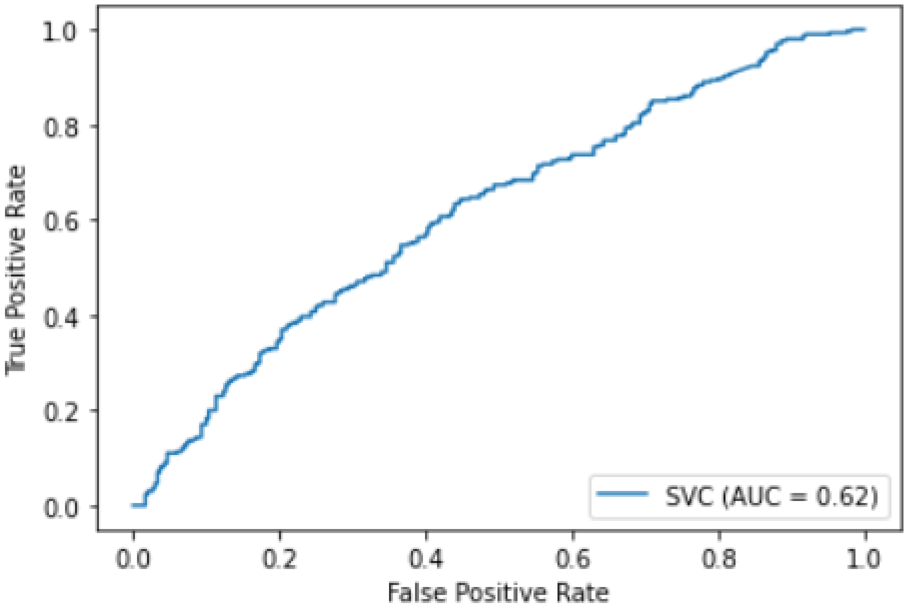
To improve the performance of our model we implemented several approaches:

1. Data augmentation
2. Introducing new dataset
3. Using pretrained model
4. Using pretrained model + frequency domain features

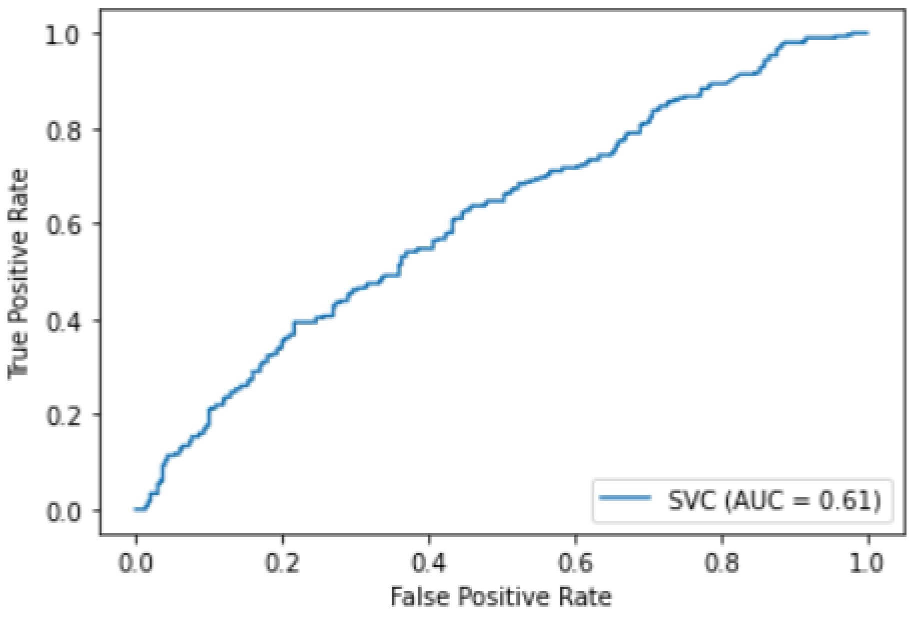
1. The original dataset provided for task 1 was not efficient to model well the task 3 dataset, thus we proposed to utilize a new dataset. The dataset from the kaggle competition **"Real and Fake Face Detection"** was exploited. It includes **960** fake and **1081** real images.
2. In the data augmentation approach, we produced 3 different images (horizontally flipped, rotated to 45 and -45 degrees) from the dataset of task 1 to check how it affects the result for task 1. It didn't give us any improvements since the number of extracted features were the same before adding augmented images. The reason for that is in the nature of the feature extraction approach of our model. As it was discussed in the beginning of the report, we are representing the input signal in the frequency domain for which features of original and augmented images are the same. Then we tried with a new dataset. The dataset size was increased for 4 times with augmented images. Unfortunately, it added only 10 extra detected faces, and we obtained 2 times smaller AUC from what it was before augmentation.
3. The feature extraction was realized by a pre-trained **Resnet 50** network. This way we extracted a **2048** length vector for each image. After that, the Support vector machine method was performed for classification with a linear kernel.
4. Mixing the task 1 and Resnet 50 features was also tested. This way the input to the SVM classifier was a **2048+300** length feature vector.

3.b) Describe the results achieved by your enhanced DeepFake detection system over the final evaluation dataset (“Task_2_3” folder). Show the results achieved in terms of ROC curve and AUC. Provide an explanation of your results in comparison with the results of Task 2.

1. There was no detection in 6 images out of the 2041 training images. (kaggle dataset) The model was entirely built from these images.
2. The obtained AUC score was **0.62,** so by the pre-trained Resnet 50 features improvements could be achieved.



3. The concatenating of the two feature vectors did not result in improving the performance. It is likely due to the **poor robustness of the frequency domain features.**



.c) Include additional information if needed.

Links:

- Task 1:

https://colab.research.google.com/drive/1WShWhuV0JDUW34WvHpHt1NiBBvgPzZ7y?usp=sharing

- Task 2 and 3:

https://colab.research.google.com/drive/1CZRsLk3i3_7TThHEuZ3VGXNbLODQxvFg?usp=sharing

- The dataset for task 3

https://www.kaggle.com/ciplab/real-and-fake-face-detection

- Data augmentation code:

https://drive.google.com/file/d/1TG89zRfCOmfuTUnOmFK_ZtJa5Vn6Ogxq/view?usp=sharing

3.d) Indicate the conclusions and possible future improvements.

According to the original paper the *Classical frequency domain analysis* is mainly efficient in the case of high quality face images. In our case we dealt with cropped images, thus lower resolution images, which can be one of the reasons for this method's poor performance. Moreover, it is not suitable for augmented data since it will give the same feature as the original one.

Features by Resnet 50 caused improvements. However, further improvements can be also achieved by gaining more data, which are efficient to model the corruptions executed in the test data. In addition we suggest using more advanced augmented data strategies to improve the result. For example, adversarial training, generative adversarial networks, neural style transfer, etc.