

# DeepFakes Detection Lab - Report

Martín Bernardi

Mary Chris Go

Noor Abdelhamed

**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:

We worked in two main approaches, the first is block based and the second is based in Siamese Networks

## **Block based:**

It is based on the assumption that there are low level artifacts in the image that are invisible to the eye but that can be detected with statistical information. When resizing images changing the brightness and when smoothing the boundary between the fake face and the rest of the image, the statistical characteristics of the image should be different than in the surroundings.

For example when rescaling the image, interpolation is done, which should be visible in the fourier domain. When changing the brightness, it is possible that in the result some pixel values are not present, or that the distribution of values has particularities, which is visible in the histogram.

[[A survey on image tampering and its detection in real-world photos](#), Lilei Zheng Ying Zhang, and Vrizlynn L.L. Thing]

[Statistical Tools for Digital Forensics, Alin C. Popescu and Hany Farid]

Additionally, if we assume that the deep fake generator started with an image of a face in JPEG format, it is possible that the portion of the face was compressed two times while the outside only once. Also it can happen that the initial images had different compression ratios. These lossy compressions should leave more compression artifacts in the fake version. One way to analyze this is to use ELA (Error Level Analysis) which consists in compressing the images again and measuring how much the image changes. Another way is to look at the DCT coefficients, which are quantized in the compression.

[[A Picture's Worth... Digital Image Analysis and Forensics](#), Neal Krawetz]

## **Siamese network:**

By viewing the development dataset, the real and fake datasets are associated. For every real image, there is a synthetic version of it that is fake. Moreover, the number of training data is

small to train 760 images in total; that's why learning a deep learning model using this small amount of data would not yield good result processing images on pixel level. Therefore, reducing the number of features representing the images would be highly effective. We make use of images associations and the siamese network to perform feature engineering of the problem we have, especially in case of using small datasets. For the siamese input, we paired images of the real and its synthetic one as not similar. Also, we made the similar pairs from real/fake images and any image from the same category but not from the same video (avoid choosing frames of the same video). The resultant network should learn the features that yield good similarity classification accuracy. From this point, we can fine tune the network to the main objective of classifying real or fake whether by using the siamese as an embedding vector extractor (freeze the network and feed forward to extract 48 embeddings per image) and feed it to a neural network to train more, or by training the siamese backbone and change the input and the output to adapt to the binary classifier.

We referred to this blog site "[Siamese networks with Keras, TensorFlow, and Deep Learning](#)" for implementing the siamese network. (DeepFake\_Siamese.ipynb)

#### **Other approaches:**

We tried to find shortcuts to solve the problem, for example in the metadata of the images there is no discriminative information. It could happen that the images were generated with different software, with different quantization tables, EXIF, etc.

We think that the best possible way to get high scores in Task 1 and 2 is to do a Nicholas Cage detector, because all the fake frames have his face. We propose doing correlations of a sample of the face of Nicholas Cage, with different rotations and scales.

Also, we applied standard machine learning algorithms to train the ELA version of the dataset in order to assess the performance of the ELA transformations. (DeepFake\_ELA.ipynb)

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).

#### **Block based approach**

Our approach was to divide the image in blocks of 32x32 pixels, and try to detect if a given block is real or fake by looking at some low level features. The best results were obtained by using the DCT coefficients of the block, and the DCT of the histogram of the block.

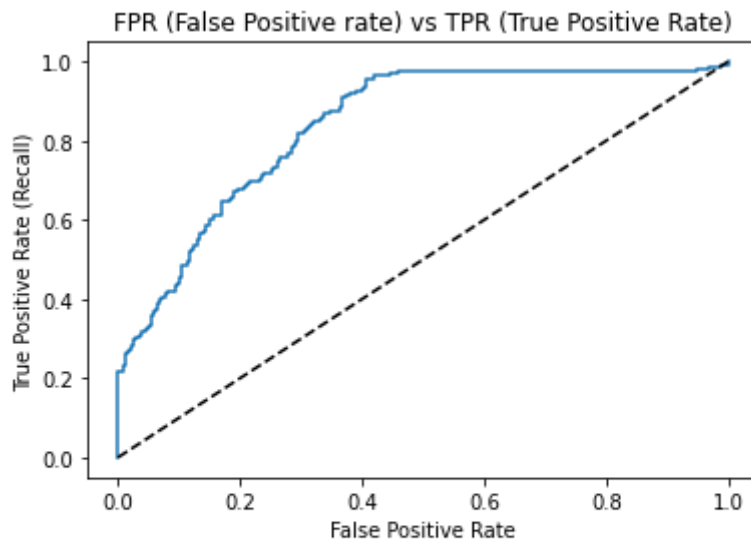
As the image is divided in blocks, in the case of evaluation, the scores for each block are averaged to get a final score of the whole image. For a small improvement, especially in training time, we are only training with the blocks that have a predominant skin color, although we are doing evaluation with all the blocks.

The best results were obtained for 32x32 blocks, where the features are the DCT of the image and the DCT of the histogram of the image.

Block-level training accuracy: 0.9308

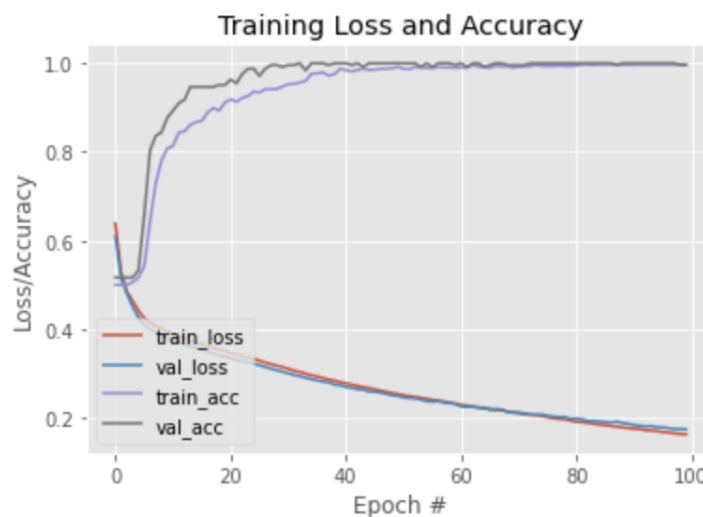
Block-level validation accuracy: 0.8165

When evaluating over full images, the results are consistent with the accuracy observed during training:



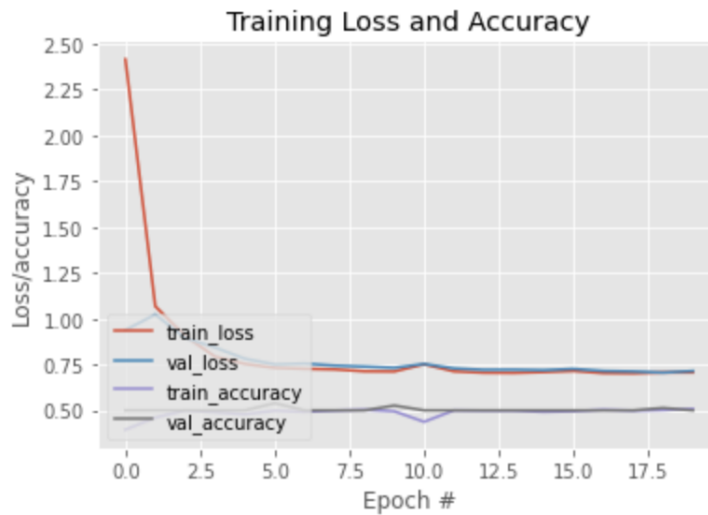
AUC: 0.8372576177285319  
EER threshold: 0.74856347  
EER: 0.2631578947368421  
EER: 0.26315789473684215

By applying **the siamese network (DeepFake\_Siamese.ipynb)**, the network learnt will to differentiate between the pairs. The plot below shows the training performance of the siamese.

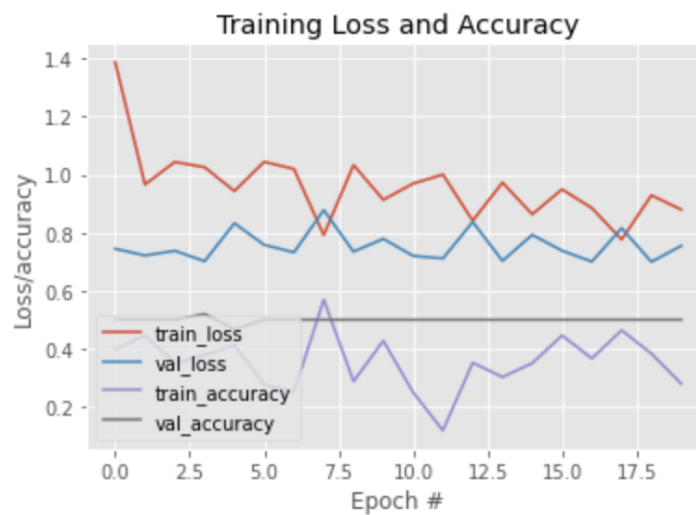


By applying similarity classification on the “evaluation” set, we get a testing accuracy of 93% percent which indicates that the model learns good features from the dataset to differentiate similarities of real and fake.

We then applied fine tuning of the network to build the binary classifier; however, the network did not learn as shown in the plot below.



By applying transfer learning of the siamese for embedding extraction and training an MLP network, the network did not learn well as shown below.

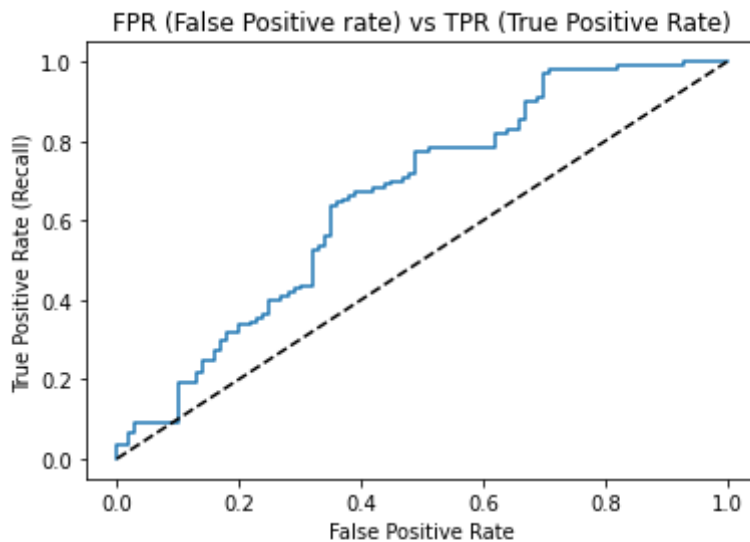


1.c) Describe the final evaluation of your proposed DeepFake detection system and the results achieved using the "evaluation" dataset (not used for training). Show the results achieved in terms of ROC curve and AUC. Provide an explanation of your results.

#### Block based approach

Test AUC: 0.651

Test EER: 0.38



The results are much worse than when considering evaluation against the training dataset, but this is expected due that there is a high amount of features and a small dataset, producing memorization instead of generalizing correctly.

By using the histogram as an additional feature, the AUC decreased to 0.607 without seeing changes in the training or validation accuracy

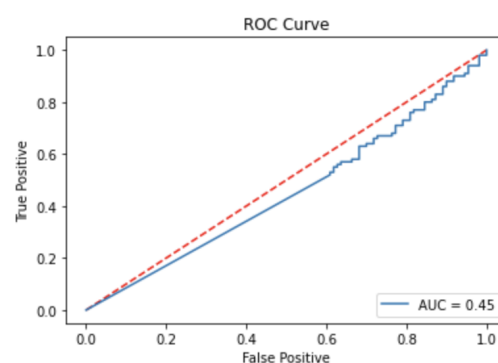
By using a stride of 16 pixels when generating the blocks, we reached an AUC of 0.644 even if the training and validation accuracy were the same.

By adding the DCT of the ELA image as a feature, the AUC decreased to 0.634.

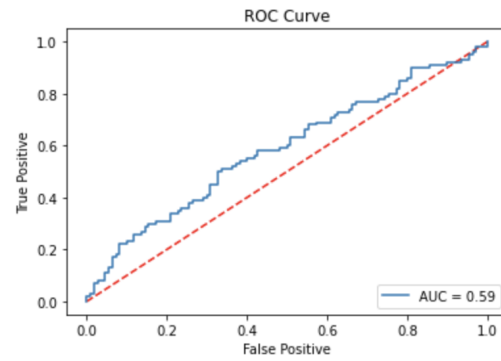
Using ELA, we tried several standard ML algorithms to assess the effectiveness of this technique. We tried the standard classifiers from sklearn library, SVM+PCA, Logistic regression, random forests, gradient boosting and Adaboosting.

Below are the plots of the ROC curves of those classifiers respectively.

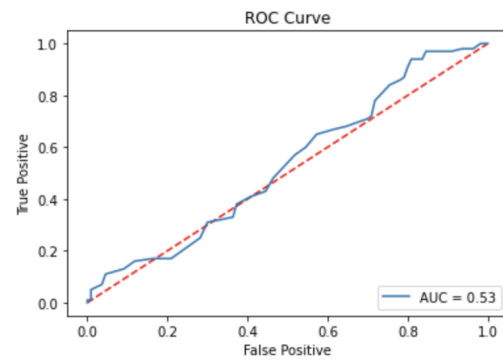
PCA+SVM



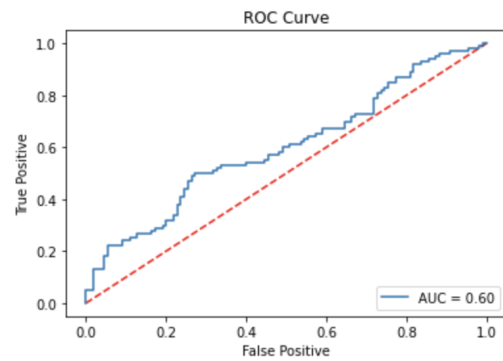
Logistic Regression



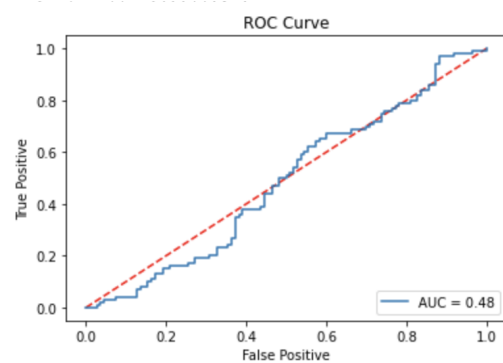
Random Forest



Gradient Boosting



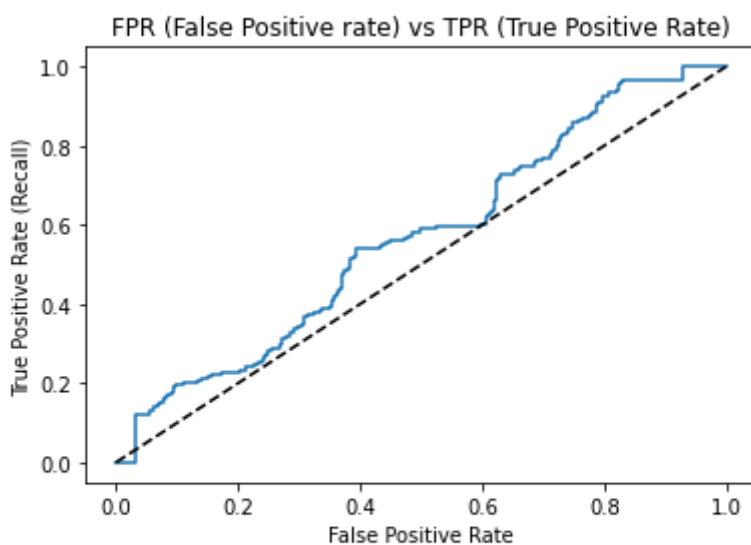
AdaBoost



**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.

We only evaluated the block-based approach because it had the best performance in the evaluation dataset of Task 1



AUC: 0.5668888888888889  
EER threshold: 0.6758491  
EER: 0.44333333333333336  
EER: 0.44333333333333336

As expected, the results for the unseen dataset are worse than those in the previous task. This is because it is very likely that the second dataset was generated using a different program, and the network has learnt to discriminate low level artifacts that are specific to the software used to generate the training 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:**

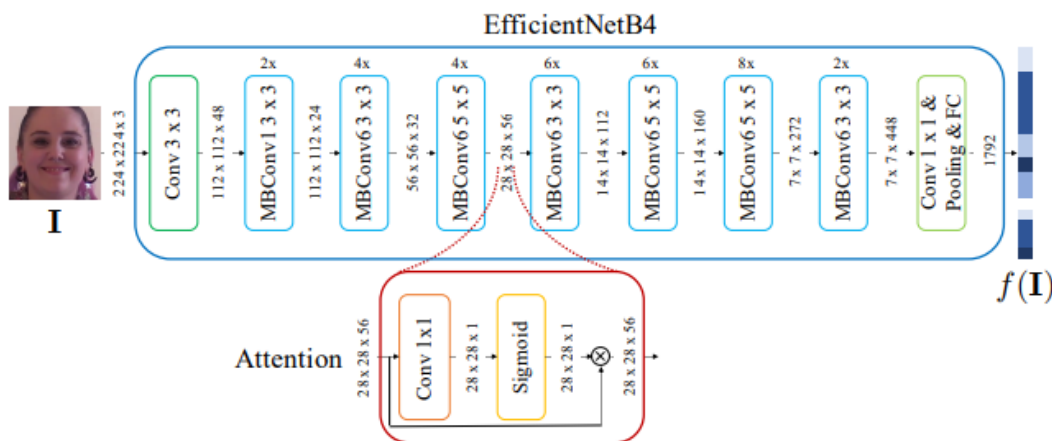
- Development: no restrictions, you can use public software and databases if you like.

- 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.

Having no limits for this task, we searched for [state of the art implementation](#) of deep fake detections. It was mentioned in this paper: [DeepFakes Evolution: Analysis of Facial Regions and Fake Detection Performance](#), that the present study is the fusion of CNN with a deep learning method with 90.61% AUC. We came across this [paper](#) that uses an ensemble of CNN models which we know that this method may lead to better prediction performance. We used their pre-trained model against our own evaluation dataset.

In its implementation, they obtained different models starting from a base network, EfficientNet. From this architecture, two paths were proposed to make the ensembling possible. First, they used an attention mechanism. It provides a method to focus which portion of the input video is informative. The attention map generated from this step can be mapped to the input sample that gives emphasis which elements have been prioritized by the network. With this method, we increase the learning capability of the network. The result from this block will then be processed by the remaining layers. The second path is what we developed from the previous task: triplet siamese training strategy. It is responsible for extracting features from the data for a better performance of the model. As mentioned above, it is beneficial for extrapolating more information during the learning process. There are various types of EfficientNet models available from the implementation: EfficientNetB4, EfficientNetB4ST, EfficientNetAutoAttB4, EfficientNetAutoAttB4ST, and Xception.

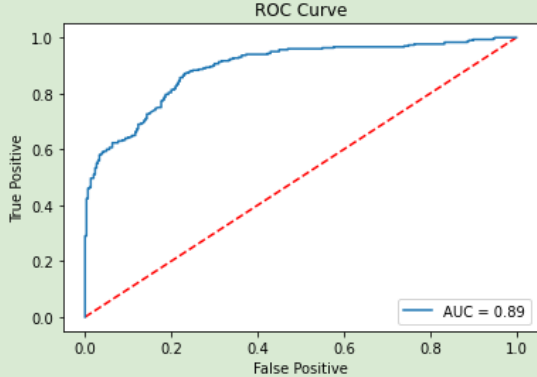
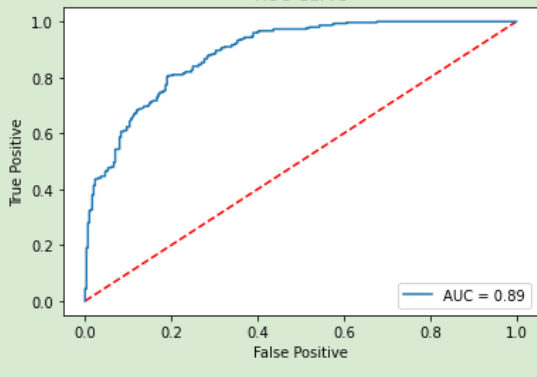
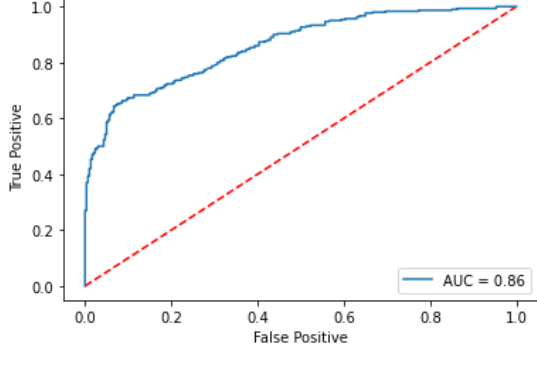


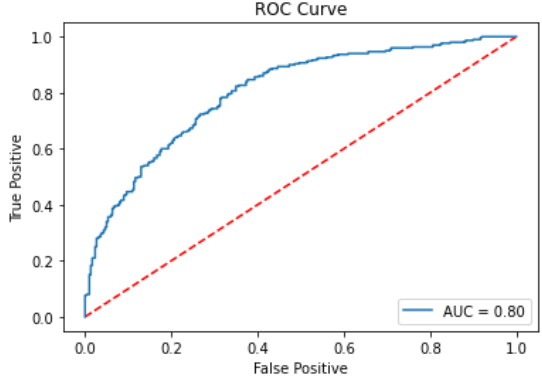
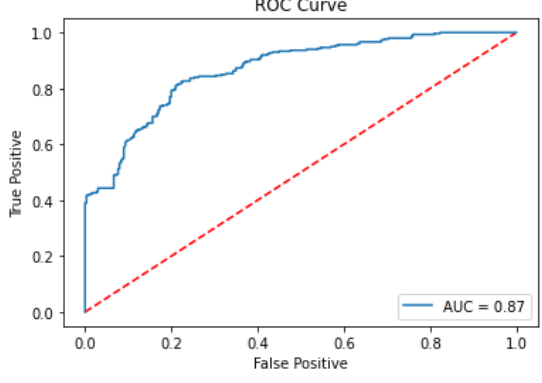
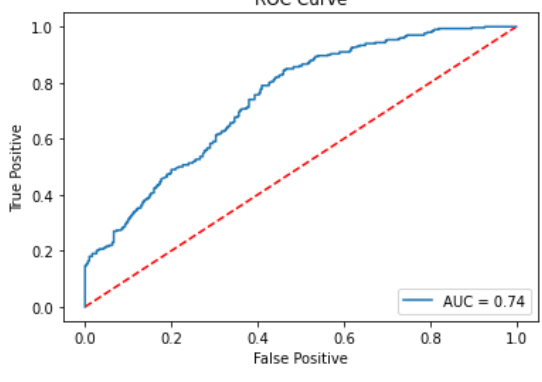
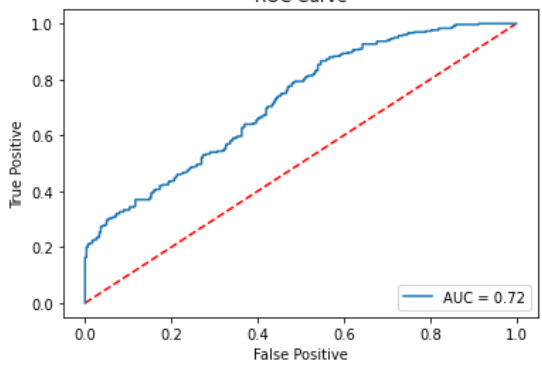
The training was done in two different paradigms: siamese and end-to-end. The latter is the basic training strategy that makes use of the DFDC dataset. The goal for this part is to have a representation in the encoding space of the layers that separates real and fake classes. This step is then finalized by fine tuning a classifier on top of the network.

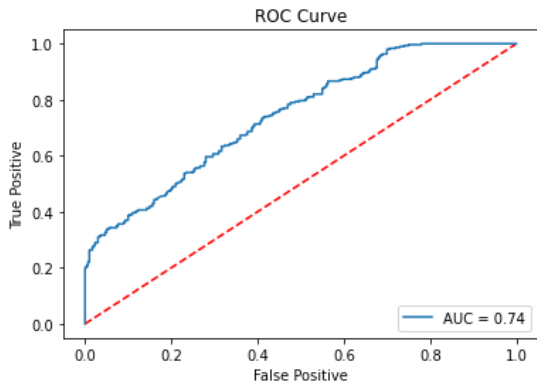
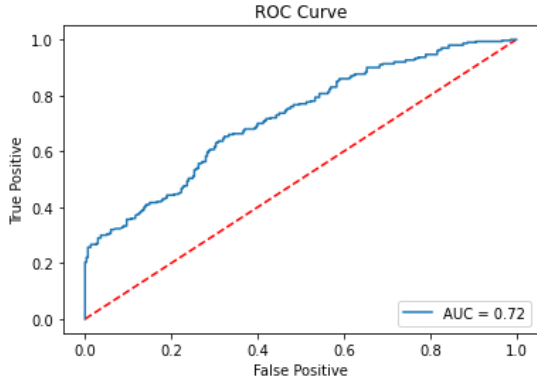
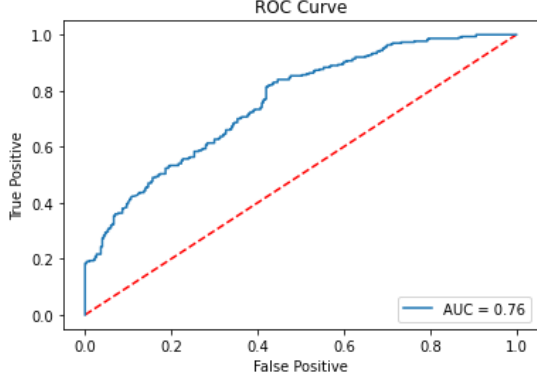


3.b) Describe the results achieved by your enhanced DeepFake detection system over the final evaluation dataset ("Task\_2\_3" folder). Show the results achieved in terms of ROC curve and AUC. Provide an explanation of your results in comparison with the results of Task 2.

There are two training datasets available: DFDC and FFPP. There are five architectures that can be chosen. Since we wanted to get the best result, we trained the model in all possible combinations. The following table showed each result.

Arc	Dataset	AUC
EfficientNetB4	DFDC	
EfficientNetB4ST	DFDC	
EfficientNetAutoAttB4	DFDC	

EfficientNetAutoAttB4ST	DFDC	 <p>ROC Curve</p> <p>True Positive</p> <p>False Positive</p> <p>AUC = 0.80</p>
Xception	DFDC	 <p>ROC Curve</p> <p>True Positive</p> <p>False Positive</p> <p>AUC = 0.87</p>
EfficientNetB4	FFPP	 <p>ROC Curve</p> <p>True Positive</p> <p>False Positive</p> <p>AUC = 0.74</p>
EfficientNetB4ST	FFPP	 <p>ROC Curve</p> <p>True Positive</p> <p>False Positive</p> <p>AUC = 0.72</p>

EfficientNetAutoAttB4	FFPP	 <p>ROC Curve</p> <p>AUC = 0.74</p>
EfficientNetAutoAttB4ST	FFPP	 <p>ROC Curve</p> <p>AUC = 0.72</p>
Xception	FFPP	 <p>ROC Curve</p> <p>AUC = 0.76</p>

As seen from the table above, the best combination is the EfficientNetB4/EfficientNetB4ST and DFDC. It generated an AUC of 89% for both models.

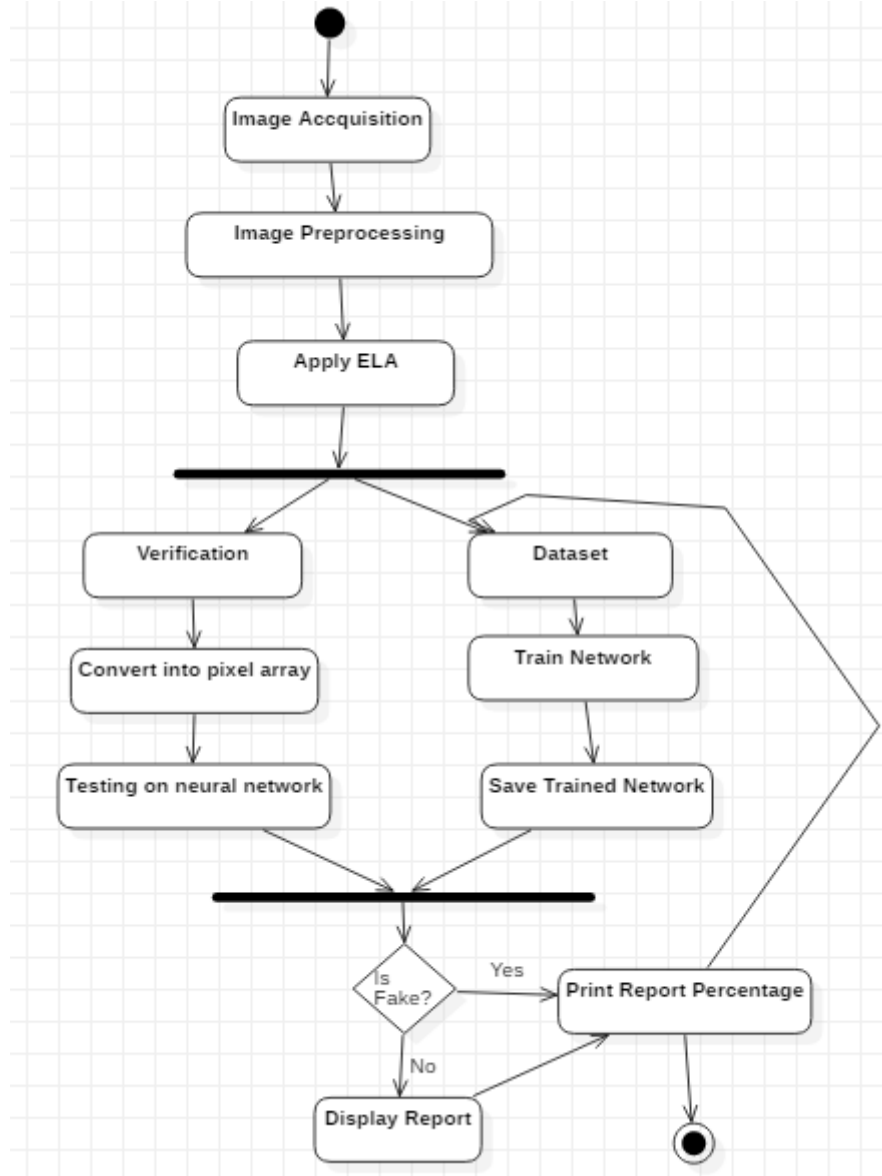
Comparing this to task 2, a 33% increase in AUC is a statistically significant improvement. Comparing these obtained results, we see the efficiency of ensembling of CNNs for deepfake detection. We expect this improvement as we are using simpler methods for the previous tasks that may not be able to handle feature-specific dataset. With the fusion of CNN models, we see that it can handle more generalization than a single CNN model.

3.c) Include additional information if needed.

There have been a number of trials of different fake image detection systems for attaining the best results.

## 1. Error-level Analysis and Deep Learning

The first analysis we tried uses an artificial neural network with the help of error-level analysis (ELA). This method is a forensic method for identifying regions of an image with a different level of compression. This is a good way of checking if the photo has been digitally tampered. It specializes in images with JPEG format because of the presence of lossy compression. The following figure was followed in implementing ML with ELA.



The network was trained using CASIA dataset: 5,122 fake photos and 7,491 real photos. The images were random and not specifically just faces. The accuracy using the test set of CASIA dataset is 98%. Although when using the given evaluation dataset for this task, the model obtained 53.17% of accuracy. The model is not robust enough to be adaptive to any image you test it. The evaluation dataset is not for this model as it was trained with a dataset with images focusing on landscape images. Most of the training images were also faceless. It might be an effective deep fake detector but only for a specific evaluation set.

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

Deep fake images have gone to another level of imitation through the use of artificial intelligence. Recently, there has been a prevalence of [deepfake satellite images](#). As it imposes threats on the general public to be misinformed, computer scientists find ways to detect fake images. In this project, we explored various methods such as block-based method, siamese architecture, and ensemble of CNNs, in detecting fake images.

Regarding the development of deep face classifiers for the Task 1 dataset, we consider that the number of samples is too low to approach the problem directly with deep neural network features from pixel data. We tried to find a generic solution for detecting fake images from image artifacts without taking into consideration high-level features and we did not get good results. We tried to do a siamese network to take advantage of the pairs of images present in the dataset but also without good results.

Most architecture published were not specific for face images. Most were trained for random images from CASIA dataset which is one famous dataset for deepfake detection. Because of this, it was a struggle to obtain good results with our evaluation set that focuses solely on faces. Out of the many architectures studied for task 3, the ensemble of CNNs performed the best with the given evaluation set. The combination of two concepts, attention mechanism and siamese training, is beneficial for obtaining good results.

There are two paths which computer scientists should take note of in deep fake: creation and detection. There are still many ways in improving not just the fake image detector, but also as to how we produce images. A solution of having a digital signature straight from the camera for each image captured can be a great way to ensure that images are real. Although, this would also create another problem of having 'signatures' forged. In terms of artificial intelligence, there are still many techniques to explore. [Some researchers](#) still prefer the traditional methods such as SURF descriptor for feature extraction which will be used for classification trained by SVM. Although, this is only applicable for static images. Furthermore, the concept of ELA can be improved since it is not perfect because as we save it a lot of times, the grid square can reach the minimum error level making it hard to detect if it's fake or real. Also, we can explore more from the noise of the image as we expect fake images to have an entirely different noise map than the real ones. As of now, deep fakes evolved from static images to videos.