

DeepFakes Detection Lab - Report

Luis Antonio Ortega Andrés
Antonio Coín Castro

March 21, 2021

Task 1: intra-database analysis

The goal of this task is to develop and evaluate DeepFake detection systems over the same database. In this task, you should use only the UADFV database, which is divided into development and evaluation datasets.

- a) Provide all details (including links or references if needed) of your proposed DeepFake detection system.*
- b) Provide all details of the development/training procedure followed and the results achieved using the development dataset of the UADFV database. Show the results achieved in terms of Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC).*
- 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.*

First of all, we are reviewing the followed training procedure along with the acquired results at test evaluations. Secondly, all the approaches that have being considered will be summarized with their reasons for being discarded.

Preprocessing

Let us begin with the preprocessing procedure, for both training and test datasets, the given images from UADFV have been preprocessed following these steps:

1. We use MTCNN face detector ([Zhang et al. \(2016\)](#)) in order to crop the section of the image that contains the face. The aim of this phase is to erase those parts of the image that are not relevant for our classification task.
2. We used `dlib` ([King \(2009\)](#)) in order to retrieve landmarks from face images. With this method, the (x, y) position of 68 landmarks is generated per sample image.
3. We add random noise to the image, more precisely, we randomly decide whether the image is slightly blurred using a GaussianBlur filter from OpenCV ([Bradski \(2000\)](#)) or it is perturbed using Gaussian noise. Each of the procedures is equally probable and every image goes through one or the other but never both. The aim is this procedure is to add some regularization to the learning method, as a result, this is only applied to training samples, that is, validation and test are not perturbed.

4. The face image is transformed into a gray-scale image and the intensity of such gray is retrieved from the landmarks retrieved by `dlib`.
5. The feature vector of each image is composed by each landmark intensity, this results in 68 features for each image.

From the procedure described above, the training dataset is split into actual training and validation using Kera's `flow_from_directory` (Chollet *et al.* (2015)), with 607 and 151 samples respectively.

Training

The training procedure is the following, we considered a parameter-optimization Grid search using Sklearn's API (Buitinck *et al.* (2013)). The considered families of models are

- SVM with RBF and linear kernel.
- Multi-layer perceptron.
- Logistic regression.

Each of these families is tested with a wide range of possible parameters and the usage of principal components analysis for feature reduction.

For example, the number of considered hidden layers for the NLP are $\{(50), (100), (50, 50), (100, 100)\}$ and the linear kernel SVM regularization term C varies from 10^{-3} to 10^3 with 40 equidistant values.

Each of these models is trained to minimize the AUC metric, and not to maximize the classification accuracy.

Evaluation results

From each family, the best model in training is chosen and evaluated over the **evaluation set**, the results are the following:

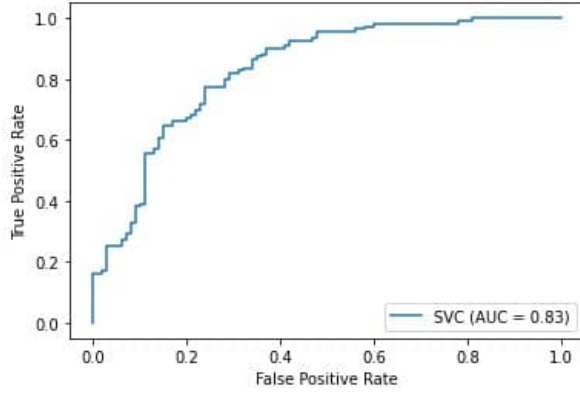
Model	PCA	Parameters	AUC scoring
SVM RBF	Yes	$C = 51.79, \gamma = 0.002$	0.9707
SVM Linear	Yes	$C = 0.04923$	0.9432
MLP	No	N° hidden layers (100,)	0.9826
Logistic regression	Yes	$C = 0.788$	0.9826

Given this results, the considered model for the test evaluation is the multilayer perceptron. However, we are showing the results obtained by the four models in order to highlight is this decision is the correct one.

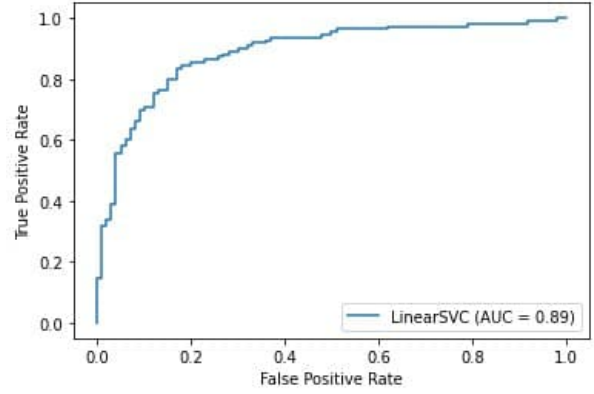
Test results

The results for each model in the test partition are the following

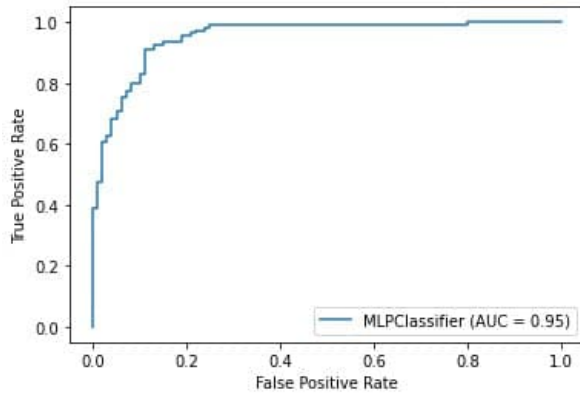
- **SVM RBF:** Accuracy = 0.7619 and AUC = 0.8315
- **SVM Lineal:** Accuracy = 0.8286 and AUC = 0.8883



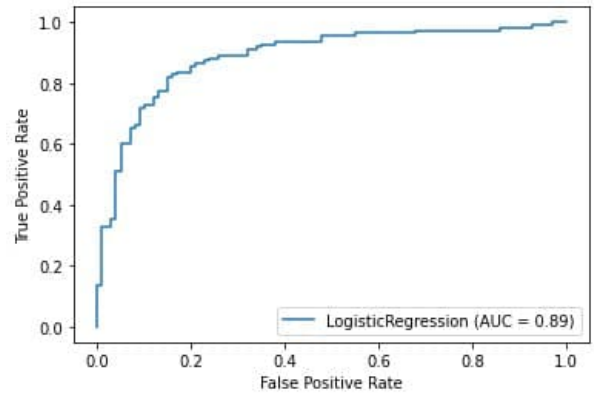
(a) SVM with kernel RBF



(b) SVM with linear kernel



(c) Multilayer Perceptron



(d) Logistic regression

Figure 1: ROC graph of the best model of each family

- **MLP:** Accuracy = 0.8762 and AUC = 0.9507
- **LR:** Accuracy = 0.8238 and AUC = 0.8880

Where we can see that the multilayer perceptron (our selected model) achieves a 0.9363 AUC, slightly ahead of the rest. We can see the ROC curve achieved by each one of the models in Figure 1.

Other considered approaches.

1. Along with the landmark intensity, we made attempts where the landmark locations were used, this resulted in a clear overfitting. As a regularization approach, we centered those locations in order to make them image size-invariant. As a result the performance improved in Task 1 but lowered in Task 2.
2. At one point, MTCNN and dlib did not return the face and landmarks for every image, at training we decided to skip that image if those features were not detected. However, this cannot be applied to test cases. We decided that, if given a test case we could not detect its features, we would assume it has the same features of a previous image of the same class. In the end, we needn't this but we are leaving all the code checks in case it is needed in the

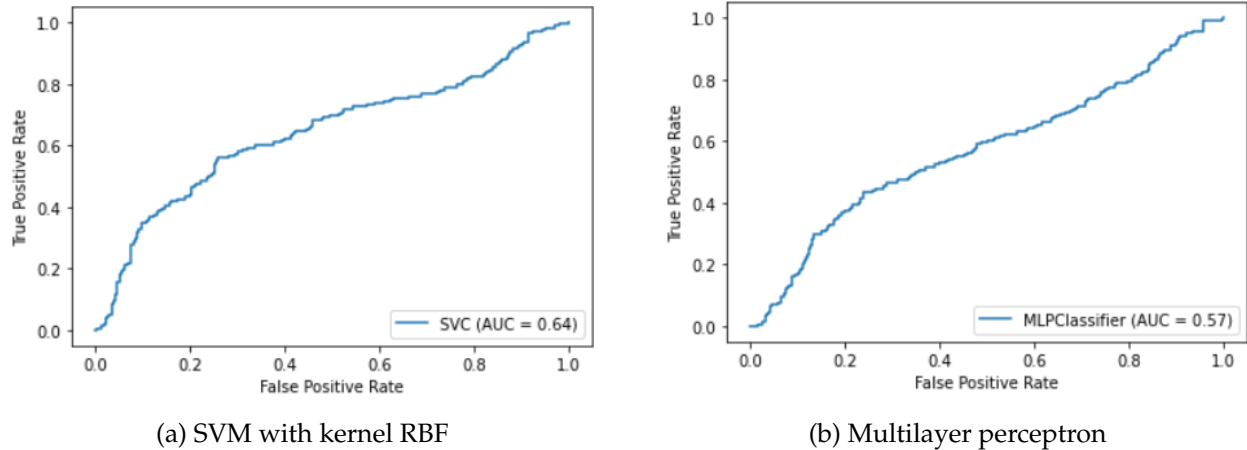


Figure 2: ROC graph over Celeb-DF database

future.

3. More models have been tested in the grid search, such as Random forests and other boosting techniques, however, they performed so poorly compared to the other models that we decided to not include them (decreasing the computational cost of the grid search).

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, you should use only the Celeb-DF. You only need to evaluate your fake detector developed in Task 1 over the evaluation dataset of Celeb-DF, not train again with them.

a) Describe the results achieved by your DeepFake detection system developed in Task 1 using the evaluation dataset of the Celeb-DF database. 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.

As we said in the previous section, our considered model is the multilayer perceptron with the parameters achieved during the grid search.

The model evaluation in the new dataset raises an AUC of 0.57166, this value is considerably lower than the ones achieved during the training phase, which makes sense considering that the first dataset from AUDFV was easier than this one from Celeb-DF in the sense that the deep fake was more easily spotted by an human.

In the training dataset there are usually vertical and horizontal markings surrounding the face. In this new dataset, deepfakes have a higher quality and do not show such markings. In short, the selected model scales poorly to this dataset because the training model is not **representative** enough.

It is worth mentioning that even though we selected the perceptron as our testing model, we checked the performance of the other three. In this results, we saw that the SVM with RBF kernel achieved a 0.601 AUC. This means that the SVM model is the one that did the best at learning *what a deep fake is* and did not overfit as much as the others.

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. You must consider the same evaluation dataset as in Task 2 (i.e. the evaluation dataset of the Celeb-DF database).

- a) Describe the improvements carried out in your proposed DeepFake detection system in comparison with Task 1.
- b) Describe the results achieved by your enhanced DeepFake detection system over the final evaluation dataset. 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.
- c) Indicate the conclusions and possible future improvements.

References

- Bradski, G. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Buitinck, Lars, Louppe, Gilles, Blondel, Mathieu, Pedregosa, Fabian, Mueller, Andreas, Grisel, Olivier, Niculae, Vlad, Prettenhofer, Peter, Gramfort, Alexandre, Grobler, Jaques, Layton, Robert, VanderPlas, Jake, Joly, Arnaud, Holt, Brian, & Varoquaux, Gaël. 2013. API design for machine learning software: experiences from the scikit-learn project. *Pages 108–122 of: ECML PKDD Workshop: Languages for Data Mining and Machine Learning*.
- Chollet, François, et al. 2015. *Keras*.
- King, Davis E. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, **10**, 1755–1758.
- Zhang, Kaipeng, Zhang, Zhanpeng, Li, Zhifeng, & Qiao, Yu. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, **23**(10), 1499–1503.