

# Biometric systems

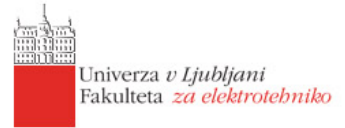
Report on the second laboratory exercise

Timotej Petrovčič

Mentor: prof. dr. Vitomir Štruc

Subject: Biometric systems

Date: November 2023



## Contents

1	Workflow	1
2	Questions and Answers	1
2.1	Question 1	1
2.2	Question 2	1
2.3	Question 3	2
2.3.1	HOG descriptors	2
2.3.2	Training the SVM	2
2.4	Question 4	2
2.5	Question 5	5
2.6	Question 6	6
2.7	Question 7	7
3	References	7

**Key words:** OpenBR, SVM, SVC, ROC curves, LFW/LFW-a dataset

## 1 Workflow

The first step of the exercise was creating a valid Docker image that would have all the required tools and packages installed. After that we first tested the OpenBR toolkit from the command line and then created an automatic script that does this for us. As for the SVM classifier we started with OpenCV based face detector whose cropped image output we used in the SVM. After we got this done we implemented additional features like manual cropping of the images and the option of calculating absolute differences between HOG descriptors. Finally we tested all of the possible options using our custom SVM and came relatively close to the 4SF algorithm implemented in the OpenBR toolkit. The SVM was modified with different kernel choices and a variety of hyperparameters, which we optimised using the parallelized version of the built-in approach using grid search for the defined parameter space. More implementation details and documentation with more examples is available at the following link

## 2 Questions and Answers

### 2.1 Question 1

What is the equal error rate (EER) for the 4SF algorithm on this database and in the selected test? EER is the operating point on the ROC curve where the false acceptance rate (FAR) and false rejection rate (FRR) have equal values.

The equal error rate using the 4SF algorithm can be seen on the following roc graph ( $EER = 0.2660$ ).

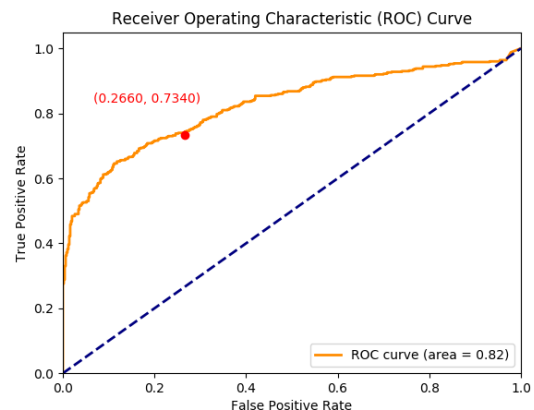


Fig. 1: ROC of OpenBR 4SF algorithm

### 2.2 Question 2

What errors does the 4FS algorithm make with pairs that actually match and with pairs that should not match? Provide some visual examples. Based on the information obtained from the article describing the 4SF algorithm, assess (or provide your opinion) on the reasons for these errors.

The algorithm 4SF operates based on the use of the LBP descriptor and the projection method into a subspace. When comparing pairs of images that actually match, errors leading to incorrect classification can occur. These errors may be due to various factors such as the diversity of expressions on faces, lighting conditions, as well as possible uneven alignment of faces. To gain a better understanding of the nature of these errors, we can examine some visual examples.

Visual examples of errors in pairs that actually match:

- A case where faces have different expressions or lighting conditions.
- A case where faces are turned at different angles.
- A case where faces are partially obscured (e.g., with a mouth or nose shield).

Visual examples of errors in pairs that should not match:

- A case where faces have identical expressions but belong to different individuals.
- A case where faces are very similar but do not belong to the same person.

Based on information from the article describing the 4SF algorithm, it can be inferred that errors in matching may be due to limitations of the LBP descriptor in capturing complex facial features. Additionally, the subspace projection method may not always be robust enough to handle various scenarios, leading to misclassification in certain pairs of images.

Here are a few examples of false negative and false positive recognitions:



Fig. 2: False negative examples



Fig. 3: False positive examples

## 2.3 Question 3

Present the procedure for calculating HOG descriptors that you used (parameter values, use of overlap, normalization, method of facial region cropping, ...) and the process of training the SVM classifier (what were the input data for training, what type of normalization was used, ...).

### 2.3.1 HOG descriptors

In crafting our HOG descriptors, we took a methodical two-step approach:

1. First off, we carved up the image into neat little cells, each measuring 20x20 pixels. These cells served as the foundation for calculating the Histogram of Oriented Gradients (HOG).
2. Following that, we organized these cells into blocks during the HOG computation, applying normalization to each block. Our preference was for blocks that encompassed 2x2 cells, ensuring a systematic normalization process.

With our HOG descriptors at the ready, we united them for each image pair. This amalgamation involved either adding or taking the absolute difference of the HOG descriptors, followed by another round of normalization.

### 2.3.2 Training the SVM

Shifting gears to training our Support Vector Machine (SVM) classifier, we opted for the SVC version within the scikit-learn module. Adhering to the default training approach, we fine-tuned the SVC parameters through a grid parameter search. With this we tried to improve the accuracy of the SVM which proved to be quite successful.

## 2.4 Question 4

What is the equal error rate (EER) for the tested algorithm on this database and in the selected test?

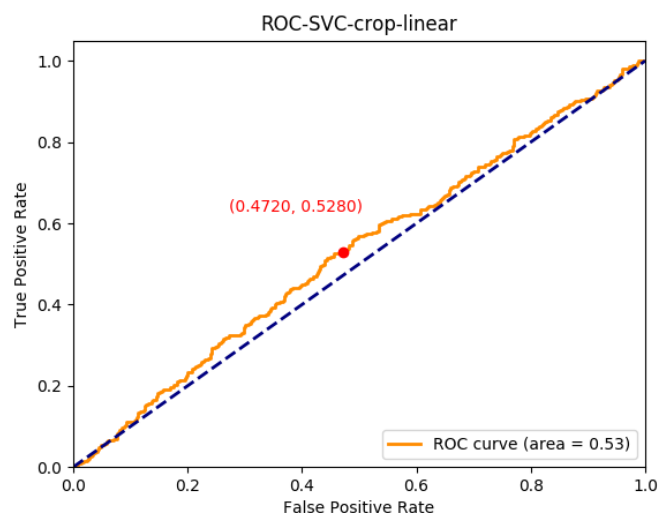


Fig. 4: SVM ROC using manual cropping and linear kernel

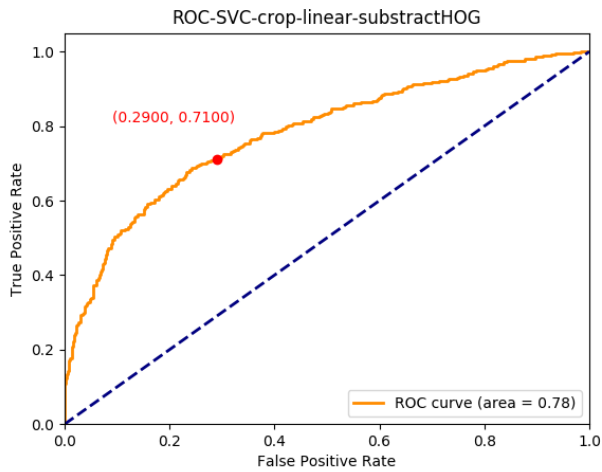


Fig. 5: SVM ROC using manual cropping, linear kernel and absolute difference of HOG descriptors

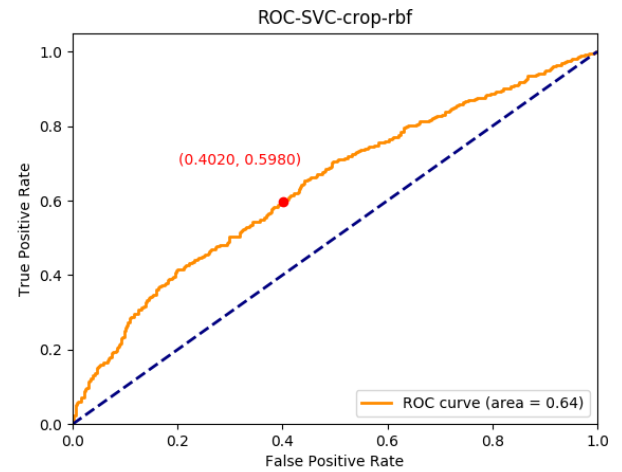


Fig. 8: SVM ROC using manual cropping and radial basis function kernel

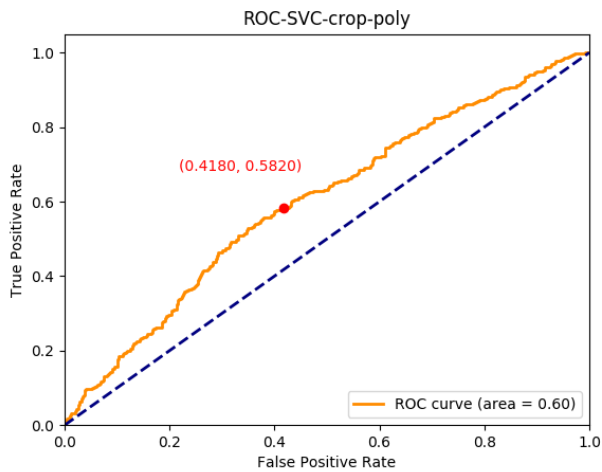


Fig. 6: SVM ROC using manual cropping and polynomial kernel

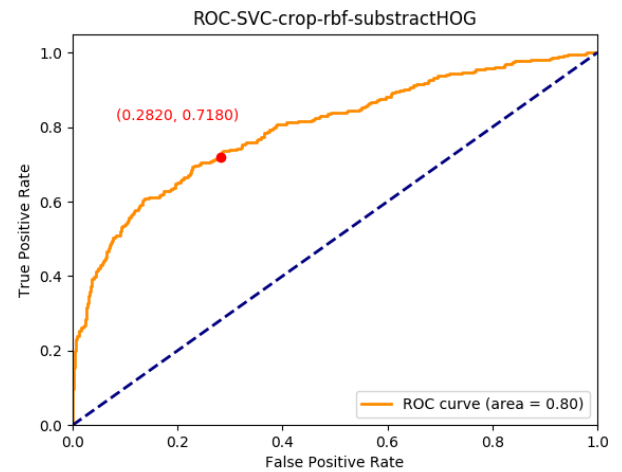


Fig. 9: SVM ROC using manual cropping, radial basis function kernel and absolute difference of HOG descriptors

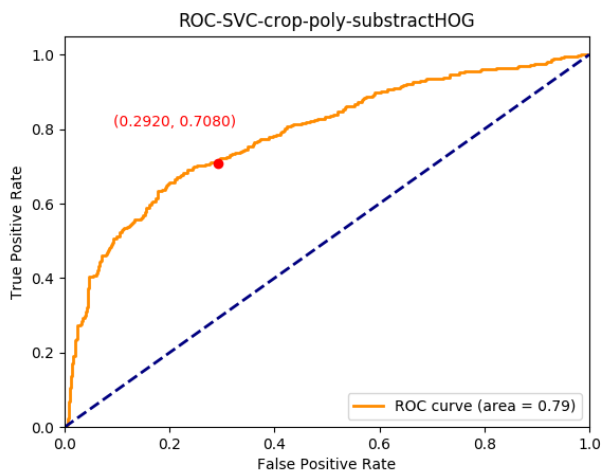


Fig. 7: SVM ROC using manual cropping, polynomial kernel and absolute difference of HOG descriptors

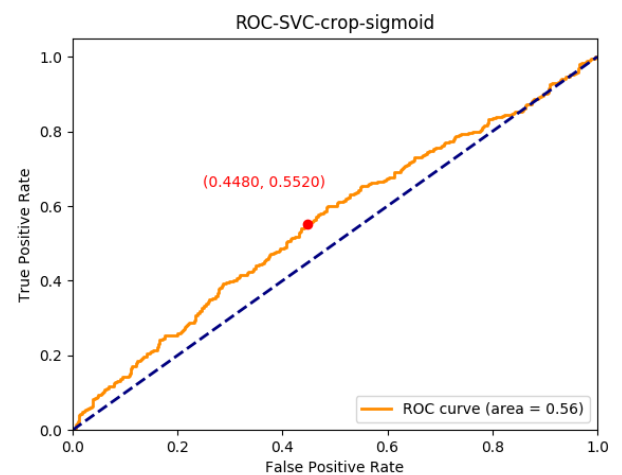


Fig. 10: SVM ROC using manual cropping and sigmoid kernel

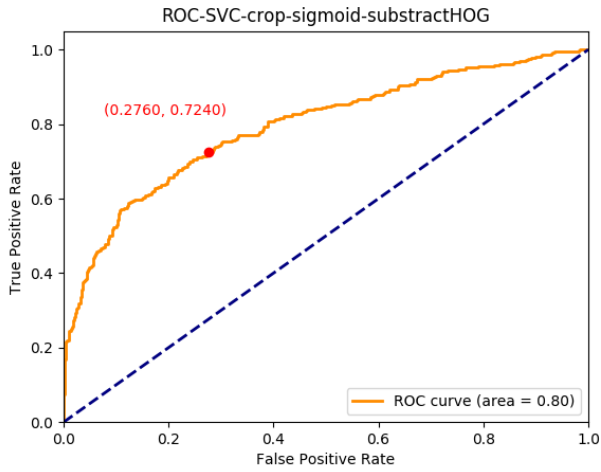


Fig. 11: SVM ROC using manual cropping, sigmoid kernel and absolute difference of HOG descriptors

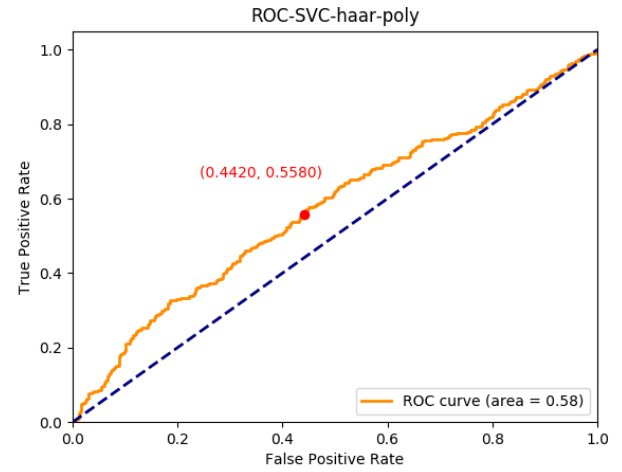


Fig. 14: SVM ROC using haar-cascade and polynomial kernel

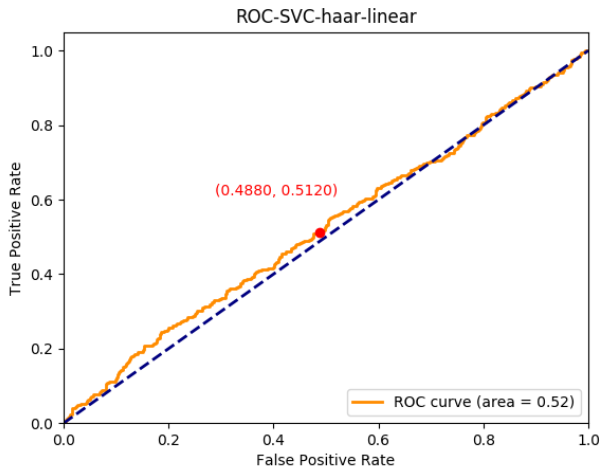


Fig. 12: SVM ROC using haar-cascade and linear kernel

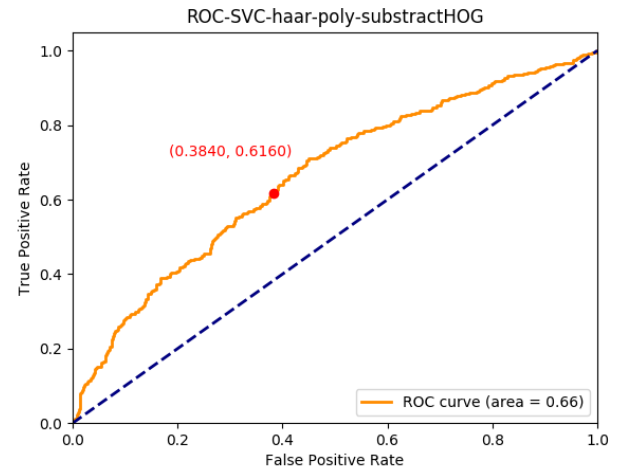


Fig. 15: SVM ROC using haar-cascade, polynomial kernel and absolute difference of HOG descriptors

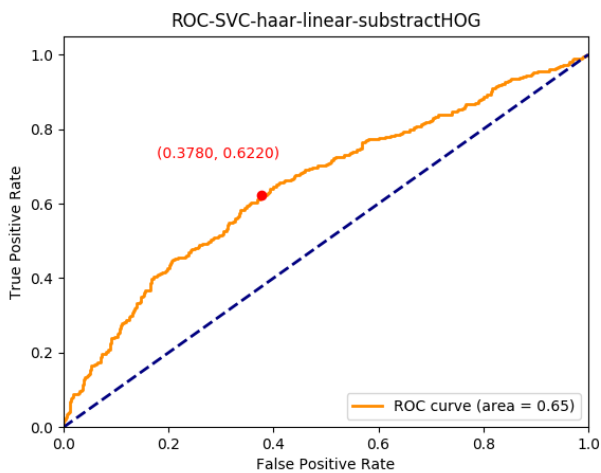


Fig. 13: SVM ROC using haar-cascade, linear kernel and absolute difference of HOG descriptors

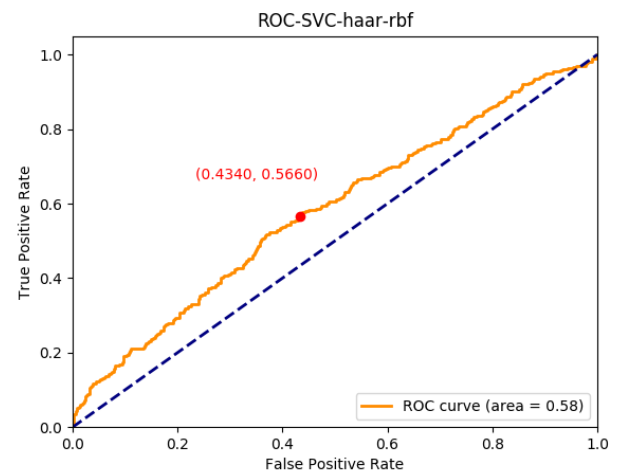


Fig. 16: SVM ROC using haar-cascade and radial basis function kernel

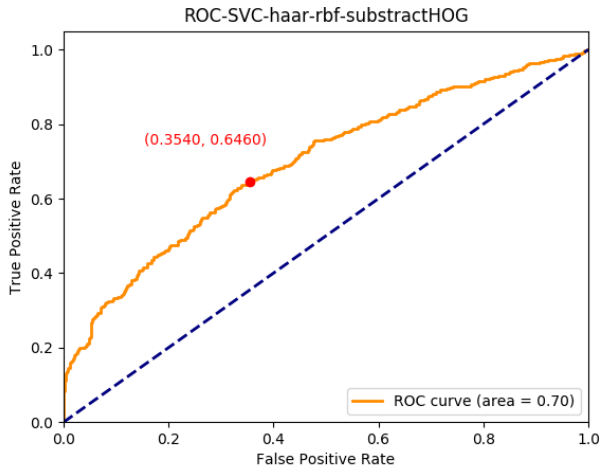


Fig. 17: SVM ROC using haar-cascade, radial basis function kernel and absolute difference of HOG descriptors

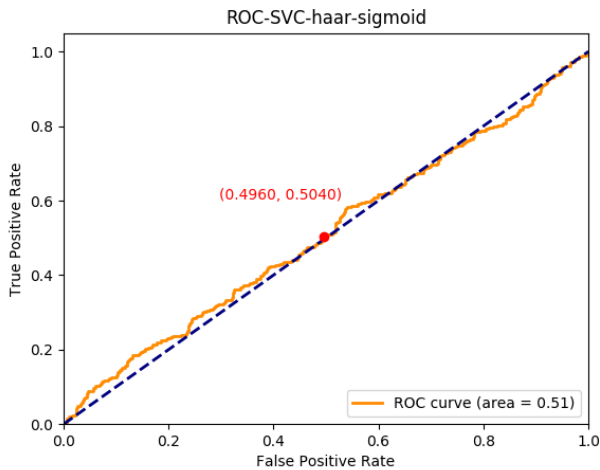


Fig. 18: SVM ROC using haar-cascade and sigmoid kernel

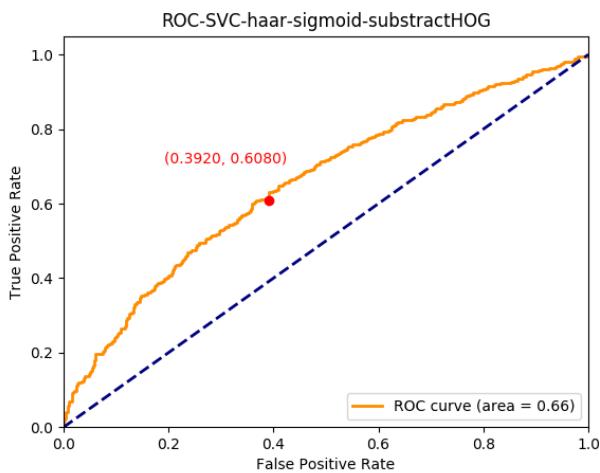


Fig. 19: SVM ROC using haar-cascade, sigmoid kernel and absolute difference of HOG descriptors

## 2.5 Question 5

What errors does the algorithm make with pairs that actually match and with pairs that should not match? Provide some visual examples. Based on knowledge of HOG descriptors and SVM classifiers, give your opinion on the reasons for these errors.

The success of the SVC classifier in the above case is dependent on many factors which might contribute to the overall result. Error contributions can be split among the following causes:

- HOG descriptor limitations
- SVC hyperparameter tuning
- SVC overfitting or underfitting

If we look at the HOG descriptors from the problem of using them for face recognition, the following problems might occur:

### 1. Sensitivity to Image Quality:

HOG descriptors are sensitive to image quality, and variations in lighting conditions can significantly impact their performance. Changes in illumination may cause the descriptor to emphasize or de-emphasize certain features, leading to recognition errors.

### 2. Limited Handling of Facial Expressions:

HOG descriptors are less effective in capturing subtle changes in facial expressions. Since they focus on local gradients and edge orientations, they may not adequately represent the dynamic aspects of facial expressions crucial for recognizing individuals in different emotional states.

### 3. Pose Sensitivity:

HOG descriptors are sensitive to variations in facial pose. Large head rotations or tilts can result in different gradient patterns, making it challenging for HOG-based systems to recognize faces across a wide range of poses.

### 4. Geometric Deformations:

HOG descriptors assume that local shape information can be represented by local intensity gradients. However, they may not handle well the geometric deformations that can occur in faces due to factors such as aging or plastic surgery.

### 5. Uniform Texture Representation:

HOG descriptors may not effectively capture facial areas with relatively uniform textures, such as the forehead or cheeks. In regions lacking distinctive edge information, the descriptor might not provide discriminative features.

### 6. Computationally Intensive:

Computing HOG descriptors for large images or in real-time applications can be computationally intensive. This may limit the speed and efficiency of face recognition systems, especially in scenarios where quick responses are crucial.



### 7. Dependency on Parameter Settings:

The performance of HOG descriptors is influenced by parameter settings such as the cell size, block size, and the number of histogram bins. Selecting appropriate parameters is essential for achieving optimal performance, and suboptimal choices can lead to reduced accuracy.

If we then look at the problems regarding the SVC:

- Using SVC hyperparameter tuning we probably wouldn't find the best parameters in our limited scope of testing; therefore, this approach would have to be improved.
- The SVM classifier may either overfit the training data, capturing noise rather than genuine patterns, or underfit the training data, failing to capture the complexities of facial variations.

Here are some of the practical examples in which pairs were not recognized correctly for each SVC variation:



Fig. 20: Error examples using the linear kernel



Fig. 21: Error examples using the polynomial kernel



Fig. 22: Error examples using the radial basis function kernel



Fig. 23: Error examples using the sigmoid kernel

## 2.6 Question 6

Which of the two tested algorithms proved to be more effective on this dataset?

When we first experimented comparing the 4SF algorithm with SVC we thought that the 4SF would in the end be a lot better, but after picking the correct kernel for the SVC and the use of hyperparameter optimization the algorithms came pretty close considering their effectiveness. Below are the receiver operating curves (ROCs) of the original 4SF algorithm and the best results we received from the SVC.

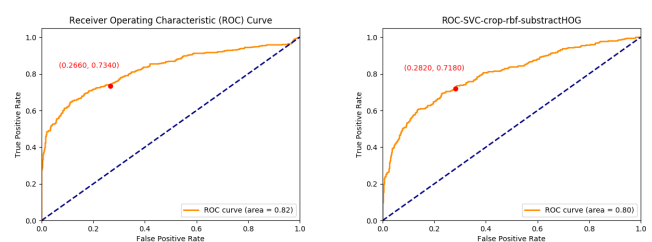


Fig. 24: ROC of the OpenBR 4SF algorithm and the best ROC of our SVM

## 2.7 Question 7

If you were able to develop a better face recognition system, what challenges of recognition would you focus on?

In the context of the OpenBR and SVM-based face verification problem, if I were to develop a better face recognition system, I would primarily address the following challenges:

### 1. Handling Variability in Pose and Expression

- **Approach:** Enhance the algorithm's capability to recognize faces under various poses and expressions.
- **Insights:** Investigate advanced pose-invariant features in HOG descriptors and fine-tune SVM configurations to improve performance under diverse facial orientations.

### 2. Illumination Robustness

- **Approach:** Improve recognition under varying lighting conditions.
- **Insights:** Investigate preprocessing techniques within OpenBR to normalize illumination variations. Optimize SVM kernels to handle changes in lighting effectively.

### 3. Partial Occlusion

- **Approach:** Develop strategies to recognize partially occluded faces.
- **Insights:** Leverage OpenBR's feature extraction methods that are less sensitive to occlusion. Explore SVM configurations within OpenBR for better handling of partially obscured facial features.

### 4. Aging and Temporal Changes

- **Approach:** Enhance recognition performance considering facial changes due to aging.
- **Insights:** Utilize OpenBR tools for age progression/regression in HOG descriptors. Optimize SVM parameters to account for temporal variations in facial features.

### 5. Adverse Environmental Conditions

- **Approach:** Strengthen the system to perform well in challenging environments.
- **Insights:** Explore robust preprocessing techniques in OpenBR to mitigate adverse conditions. Fine-tune SVM configurations for optimal performance in diverse scenarios.

### 6. Ethnic and Gender Diversity

- **Approach:** Improve system inclusivity across diverse ethnicities and genders.
- **Insights:** Train the model on diverse datasets using OpenBR functionalities. Optimize SVM parameters for unbiased and inclusive recognition.

### 7. Security and Privacy Considerations

- **Approach:** Implement measures to address security and privacy concerns.
- **Insights:** Enhance secure storage and transmission protocols in OpenBR. Consider SVM configurations that prioritize security and privacy without compromising recognition accuracy.

### 8. Real-Time Processing

- **Approach:** Optimize algorithms for real-time face recognition.
- **Insights:** Utilize OpenBR's efficiency features and explore SVM configurations for rapid and reliable real-time processing.

### 9. Adversarial Attack Resilience

- **Approach:** Strengthen the system against adversarial attacks.
- **Insights:** Incorporate OpenBR mechanisms to handle adversarial attacks. Explore SVM configurations with adversarial training to enhance system resilience.

### 10. Addressing Data Bias and Ensuring Fairness

- **Approach:** Address biases in training data and ensure fairness in recognition.
- **Insights:** Scrutinize training datasets for biases using OpenBR's dataset analysis tools. Implement strategies for balanced representation. Continually assess the fairness of the system and SVM configurations.

By focusing on these challenges and integrating advanced features and optimizations within OpenBR and SVM, the face recognition system can be significantly enhanced to deliver superior performance across various scenarios and demographics.

## 3 References

- V. Štruc: Biometric systems, lectures on the subject, FE 2023
- Lior Wolf, Tal Hassner, and Yaniv Taigman, Effective Face Recognition by Combining Multiple Descriptors and Learned Background Statistics, IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI), 33(10), Oct. 2011 (PDF)
- Documentation on the following datasets and software:
  - LFW dataset and technical report: [Link](#)
  - LFW-a dataset and technical report [Link](#)
  - OpenCV: [Link](#)
  - OpenBR: [Link](#)
  - Python libraries used:
    - \* *scikit-learn*, *scikit-image*, *opencv*, *matplotlib*, *numpy*, *subprocess*, *multiprocessing*