POLITECNICO
MILANO 1863

Network Measurement and Data Analysis Lab

Professor: Francesco Musumeci

Team members:
Usevalad Milasheuski 10816982
Kiarash Rezaei 10809307
Sajad Hamzenejadi 10818436

Project #4 Anomaly Detection

Outline:

1. Problem definition
2. Datasets
3. Algorithms
4. Data splitting
5. Cross-validation
6. Principal Component Analysis (PCA)
7. Results
8. Comparison
9. Transfer learning
10. Alternative: Covariance feature space
11. XAI: GradCAM
12. Conclusion

1. Problem Definition

Anomaly detection in an optical network based on images of 16 QAM constellation diagrams at the receiver of Channel Under Test (CUT).

Problem type: Unsupervised

Datasets: 1- Full Image 2- one-symbol image (from 2 different Domains)

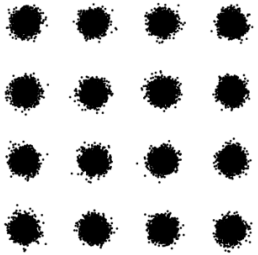
Feature Extraction Methods: 1- Flattening and resizing 2- statistical feature extraction and covariance matrices

Proposed Algorithms : 1- one class support vector machine 2- Isolation Forest

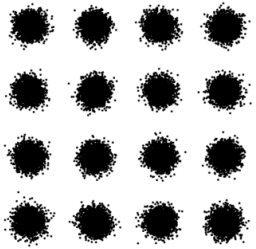
Advanced approaches: 1- Transfer Learning 2- Grad-CAM (briefly)

2. Datasets

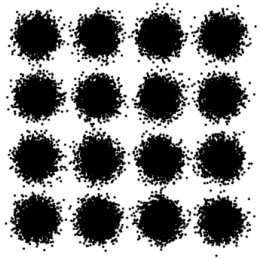
2.1 Full images



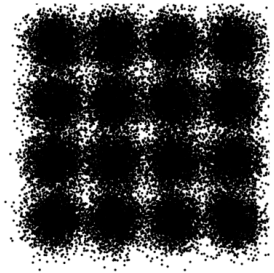
Normal 1



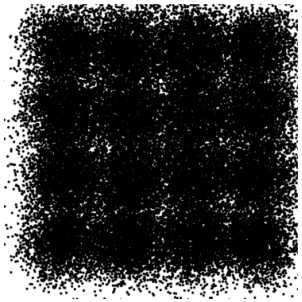
Normal 2



Fault 1



Fault 2



Fault 3

Dataset size = 500 (100 per each image)

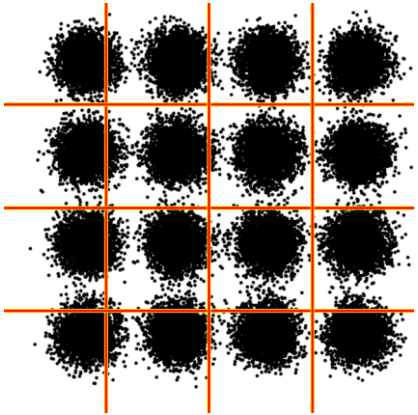
image dimension = 614 by 614

Classes = {Normal : 1, Faulty:-1}

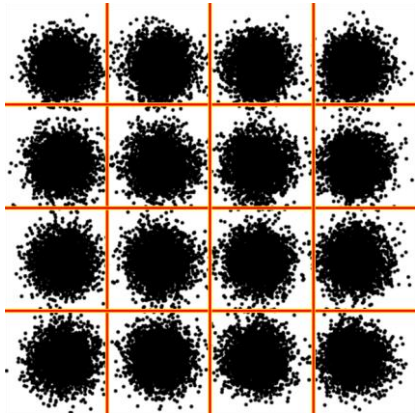
2. Datasets

2.2. Creating one-symbol dataset

Converting Images into Sub-images: We cropped 44 pixels from the left and bottom of the original 614x614 images to shift the sub-images to the center for better separation. In the end, we extracted 16 sub-images from each image.



614*614: Before cropping



570*570: After cropping

2. Datasets

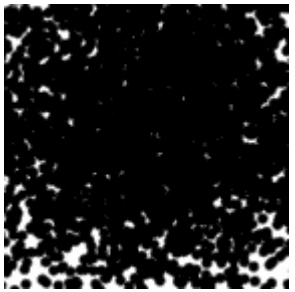
2.2 one-symbol images

Normal 1

Normal 2

Fault 1

Fault 2



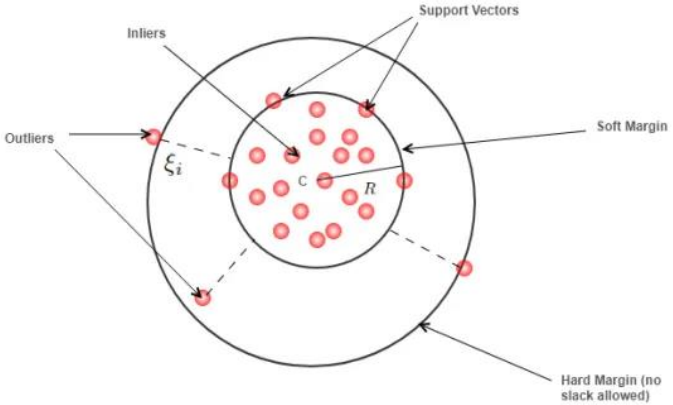
Fault 3

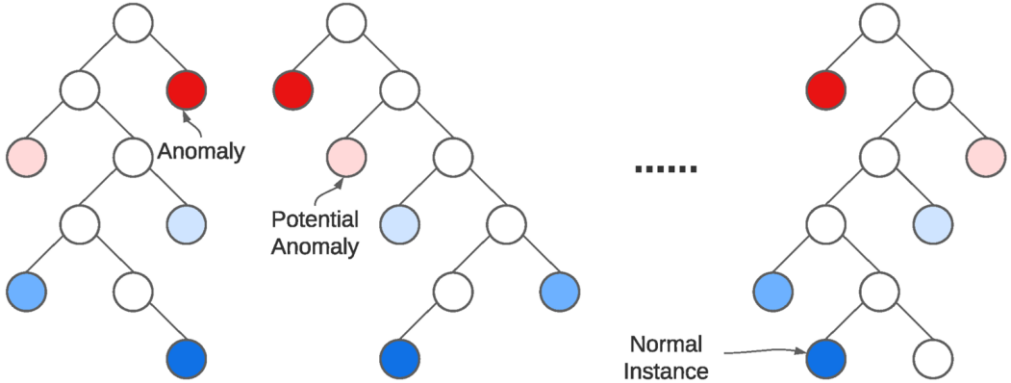
Dataset size = 500 (100 per each image, we use only one symbol per constellation!)

image dimension = 142*142

Classes = {Normal : 1, Faulty:-1}

3. Algorithms: OC-SVM and Isolation Forest





OC-SVM

Idea: map all the data points in higher dimensional feature space, ϕ, and then try to use a hyper-sphere such that most of the data lie inside it.

Isolation Forest

Idea: create a set of random decision trees where each tree isolates instances in a dataset. Anomalies are expected to require fewer partitions to be isolated compared to normal instances. By measuring the average path length of each instance in the trees, anomalies can be identified.

3. Algorithms: Isolation Forest (contamination hyperpar.)

In Isolation Forest, the "contamination" hyperparameter represents the estimated proportion of outliers or anomalies in the dataset. It is used during model fitting to determine the threshold for anomaly detection. Since out folds contain a mix of normal and faulty data, we can use CV to select the best value for the problem (same principle holds for the other hyperparameters).

4. Data Splitting

We splitted data to training, validation and test subsets with a fixed random state value.

It should be noted the **training set** contains **only normal samples.**

The number of samples in the training dataset is 160

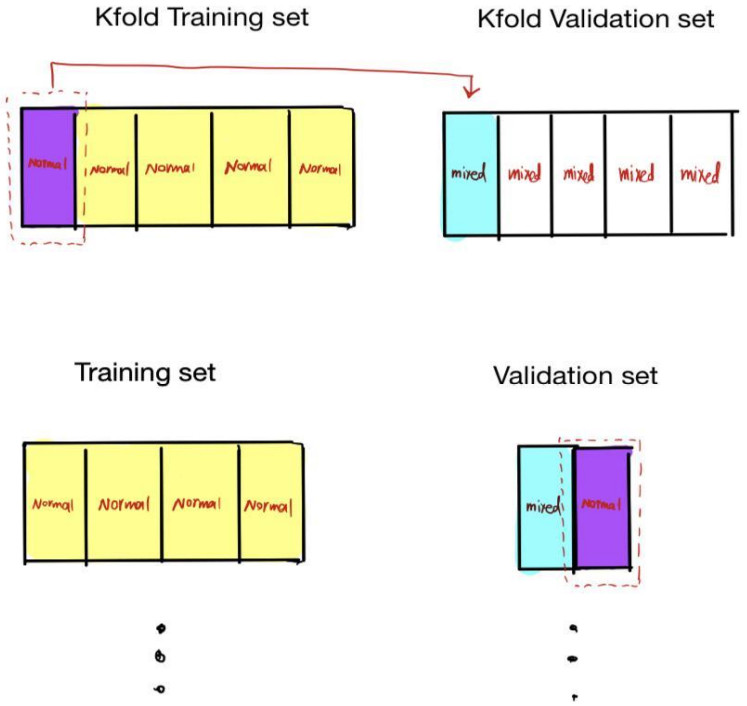
The number of samples in the validation dataset is 272 (Faulty= 240, Normal=32)

The number of samples in the testing dataset is 68 (Faulty= 60, Normal=8)

5. Cross-validation

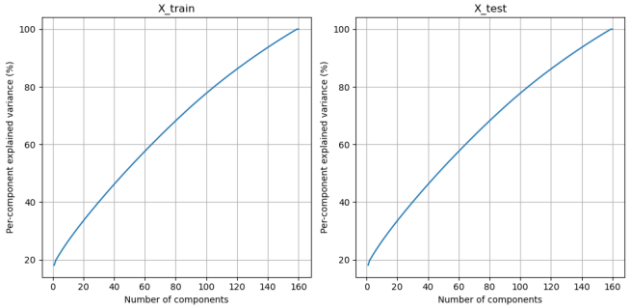
We defined the hyper parameter space as grid. Separately for each model then performed a **5 fold cross-validation** (as it is shown in the image) to tune the hyper parameters in order to obtain the most generalizable model.

The hyperparameter space for both models consists of a grid of 3*3*2 = **18 elements**.

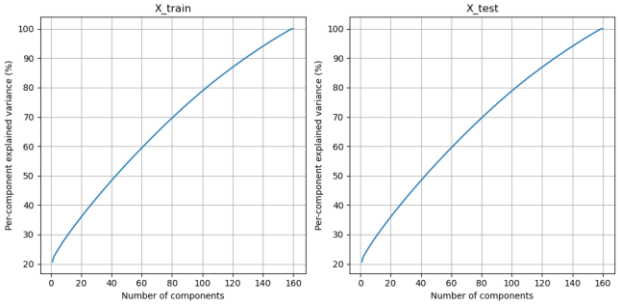


6. Principle Component Analysis (PCA)

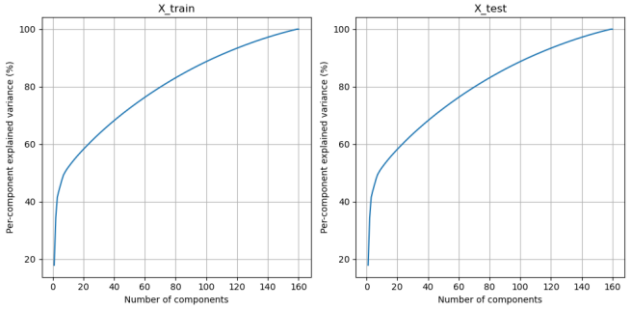
6.1. Original Images + PCA



6.2. Resized image + PCA (250*250)



6.3. One-symbol Images + PCA



As it can be seen, with 160 principle components we can obtain 100% of explained variance ratio in all cases. 160 is the number of samples in training set

7. Results

7.1. Original Images: One class SVM vs. Isolation Forest

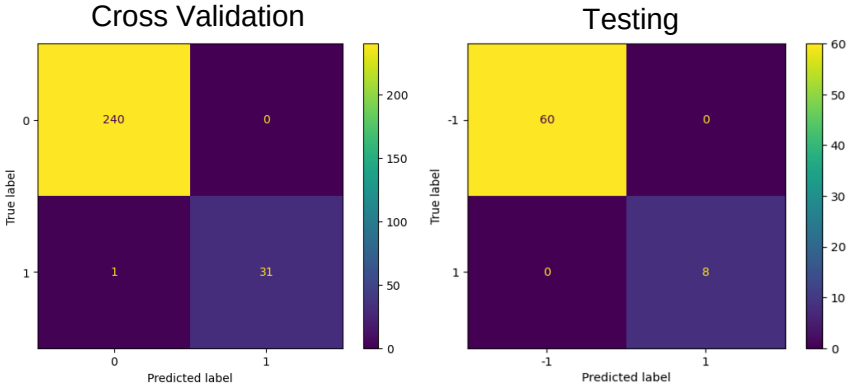
Crossval time 00:05:29

Best hyperparams during crossval: {'nu': 0.01, 'kernel': 'sigmoid', 'gamma': 'auto',

Accuracy of the final model on validation: 0.996

Average K-fold accuracy: 0.984 +- 0.009

Test Accuracy: 1.0



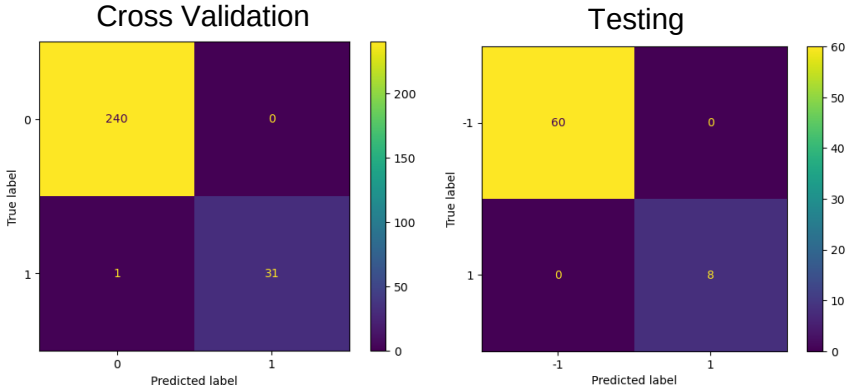
Crossval time 00:20:08

Best hyperparams during crossval: {'n_estimators': 100, 'contamination': 0.1, 'max_samples': 50}

Accuracy of the final model on validation: 0.996

Average K-fold accuracy: 0.984 +- 0.009

Test Accuracy: 1.0



7. Results

7.2. Original Images+PCA: One class SVM vs. Isolation Forest

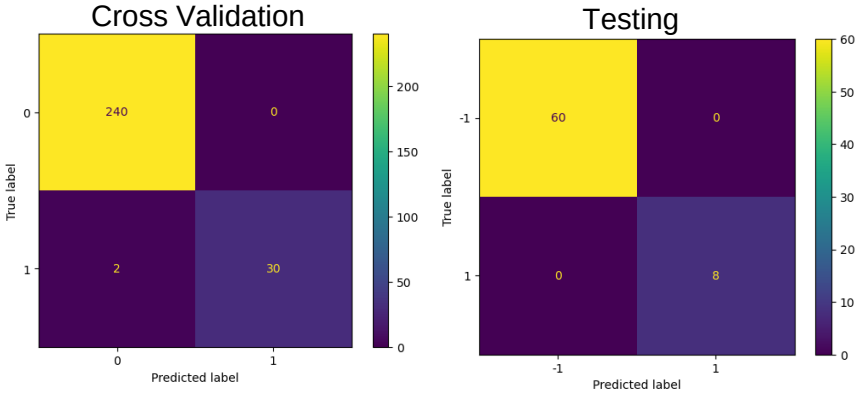
Crossval time < 1s

Best hyperparams during crossval: {'nu': 0.2, 'kernel': 'sigmoid', 'gamma': 'auto'}

Accuracy of the final model on validation: 0.9926470588235294

Average K-fold accuracy: 0.926 +- 0.027

Test Accuracy: 1.0



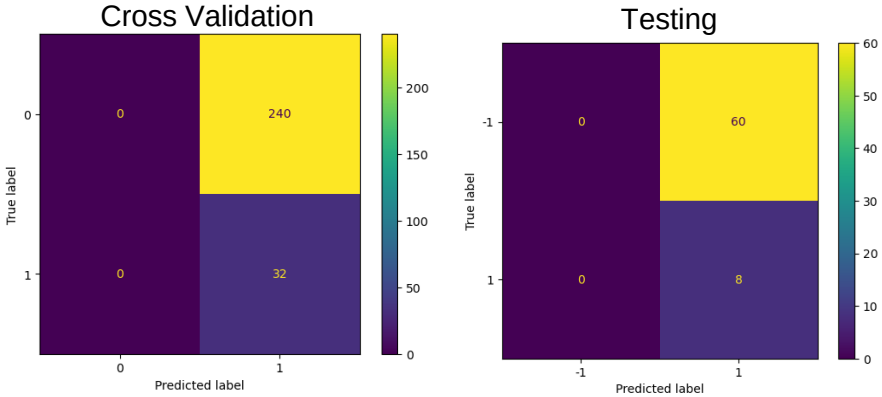
Crossval time 00:00:16

Best hyperparams during crossval: {'n_estimators': 50, 'contamination': 0.1, 'max_samples': 100}

Accuracy of the final model on validation: 0.118

Average K-fold accuracy: 0.403 +- 0.028

Test Accuracy: 0.118



7. Results

7.3. Resized image (250*250): One class SVM vs. Isolation Forest

Crossval time 00:00:47

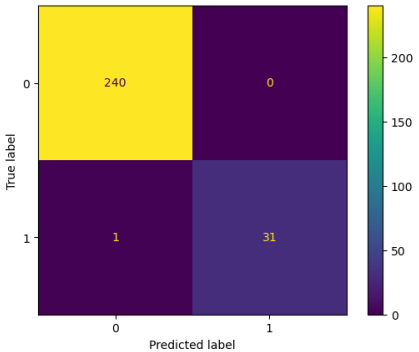
Best hyperparams during crossval: {'nu': 0.01, 'kernel': 'sigmoid', 'gamma': 'auto'}

Accuracy of the final model on validation: 0.996

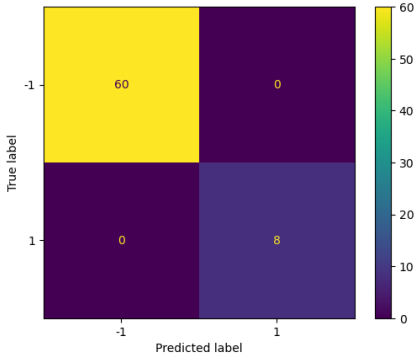
Average K-fold accuracy: 0.986 +- 0.008

Test Accuracy: 1.0

Cross Validation



Testing



Crossval time 00:02:53

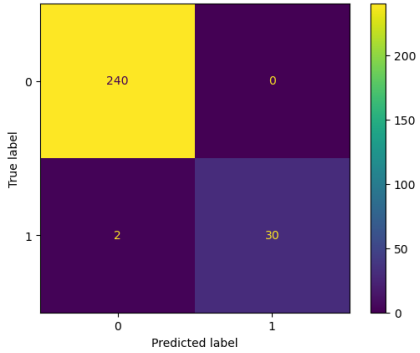
Best hyperparams during crossval: {'n_estimators': 75, 'contamination': 0.1, 'max_samples': 50}

Accuracy of the final model on validation: 0.993

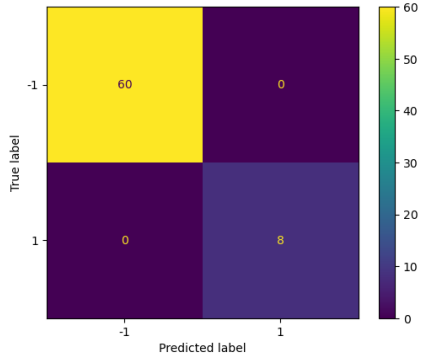
Average K-fold accuracy: 0.986 +- 0.009

Test Accuracy: 1.0

Cross Validation



Testing



7. Results

7.4. Resized image + PCA: One class SVM vs. Isolation Forest

Crossval time < 1sec

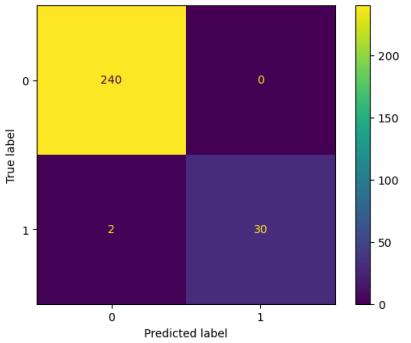
Best hyperparams during crossval: {'nu': 0.01, 'kernel': 'sigmoid', 'gamma': 'auto'}

Accuracy of the final model on validation: 0.993

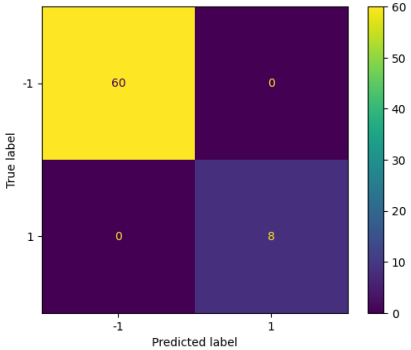
Average K-fold accuracy: 0.851+- 0.621

Test Accuracy: 1.0

Cross Validation



Testing



Crossval time 00:00:16

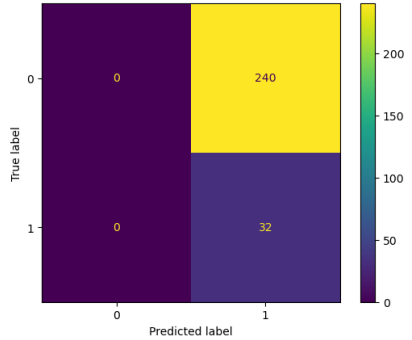
Best hyperparams during crossval: {'n_estimators': 75,, 'contamination': 0.1, 'max_samples': 50}

Accuracy of the final model on validation: 0.118

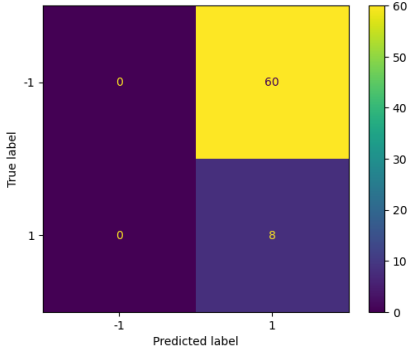
Average K-fold accuracy: `0.412 +- 0.041`

Test Accuracy: 0.118

Cross Validation



Testing



7. Results

7.5. One-symbol Images: One class SVM vs. Isolation Forest

Crossval time 00:00:13

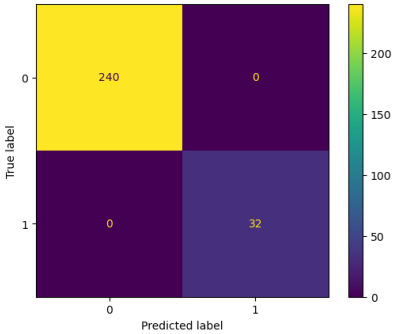
Best hyperparams during crossval: {'nu': 0.01, 'kernel': 'sigmoid', 'gamma': 'auto'}

Accuracy of the final model on validation: 1.0

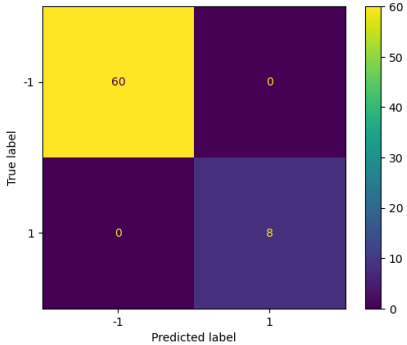
Average K-fold accuracy: 0.986 +- 0.011

Test Accuracy: 1.0

Cross Validation



Testing



Crossval time 00:00:57

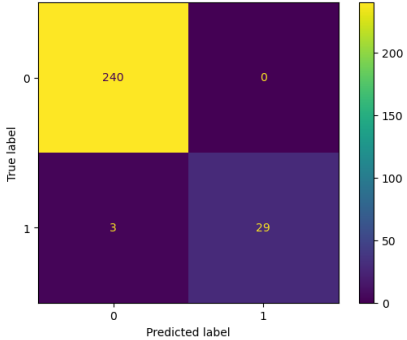
Best hyperparams during crossval: {'n_estimators': 75, 'contamination': 0.1, 'max_samples': 10}

Accuracy of the final model on validation: 0.989

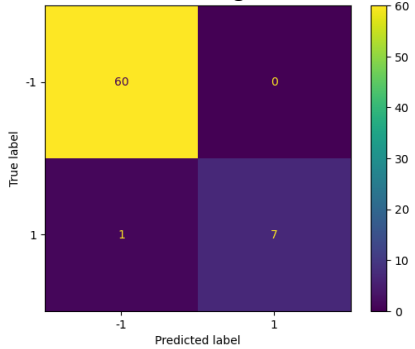
Average K-fold accuracy: 0.986 +- 0.011

Test Accuracy: 0.985

Cross Validation



Testing



7. Results

7.6. One-symbol Images + PCA: One class SVM vs. Isolation Forest

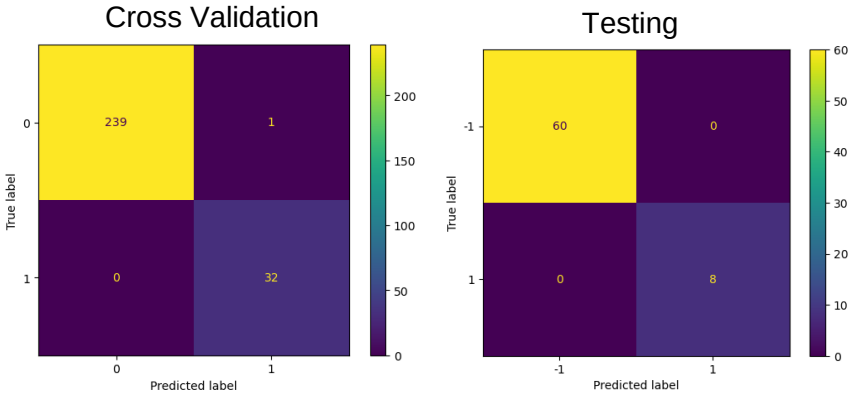
Crossval time < 1sec

Best hyperparams during crossval: {'nu': 0.2, 'kernel': 'rbf', 'gamma': 'scale'}

Accuracy of the final model on validation: 0.996

Average K-fold accuracy: 0.812 +- 0.039

Test Accuracy: 1.0



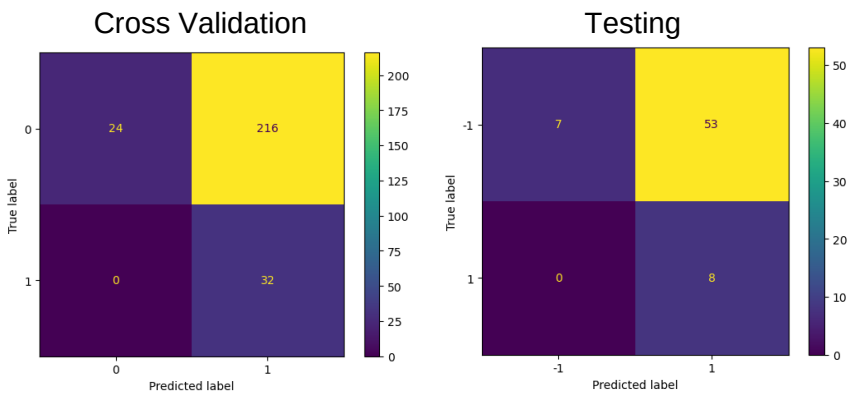
Crossval time 00:00:16

Best hyperparams during crossval: {'n_estimators': 100, 'contamination': 0.1, 'max_samples': 10}

Accuracy of the final model on validation: 0.206

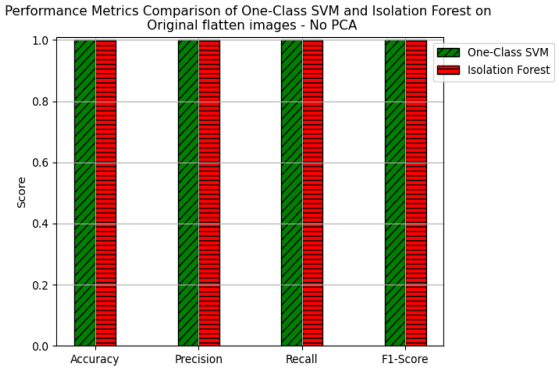
Average K-fold accuracy: `0.412 +- 0.019`

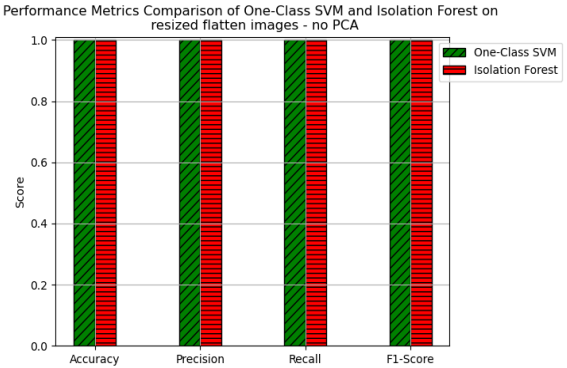
Test Accuracy: 0.22

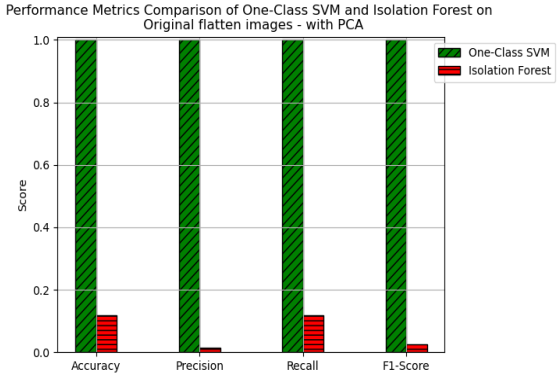


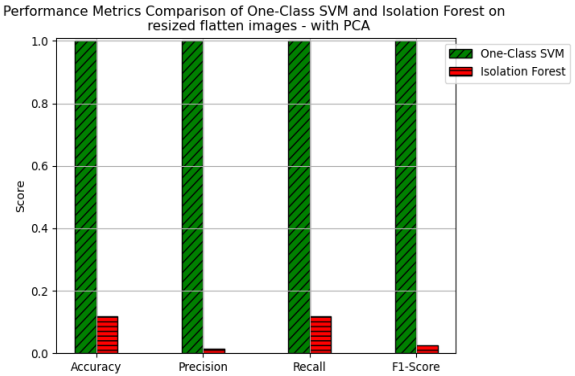
8. Comparison

8.1. Original and resized images, OC-SVM and IF, with/without PCA







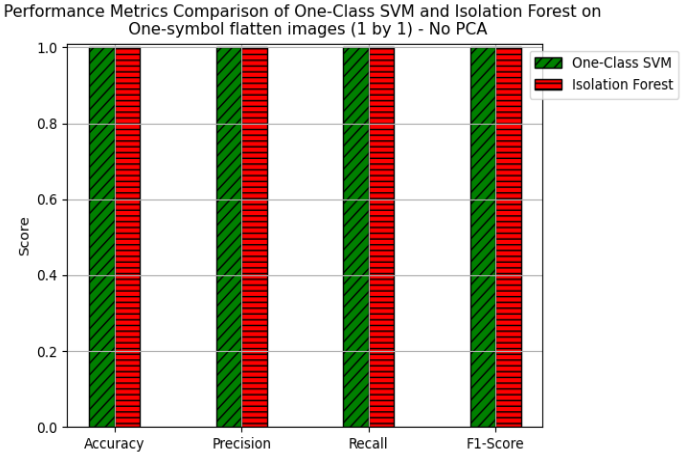


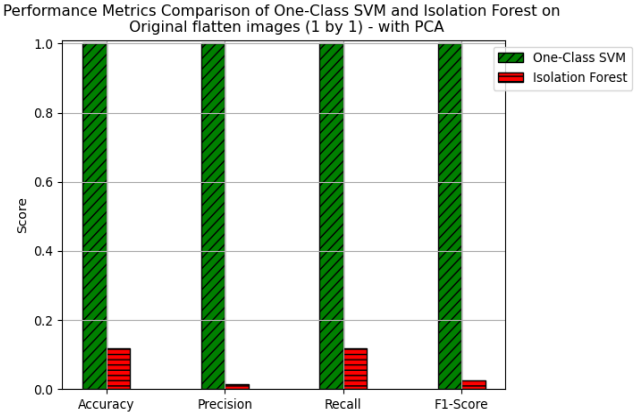
Observation

We have observed that for both original and resized images, using the principle components only **decrease** the performance of the isolation forest while for the one-class SVM, the performance **remains high.**

8. Comparison

8.2. One symbol, OC-SVM and IF, with/without PCA





Observation

The **same** situation happened for one symbol images

9. Transfer Learning

For the following results we used dataset with **OSNR = 25 dB** as the **domain A** and the other dataset with **OSNR = 40** dB as the **domain B** (original images and one-symbol images have been used).

Approach:

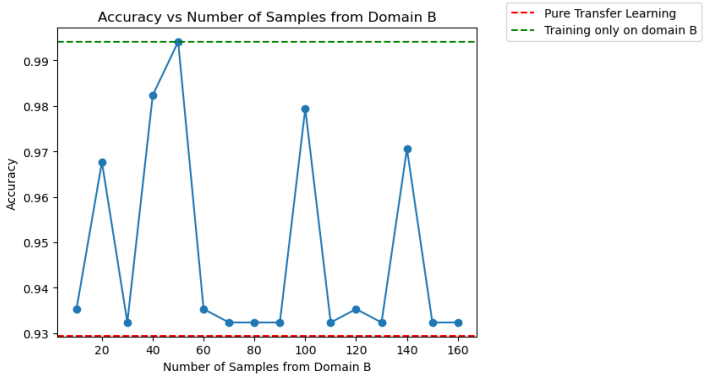
1- Pure TL: We performed training and hyper parameter tuning on the domain A, evaluated the model on a testset containing images from domain B

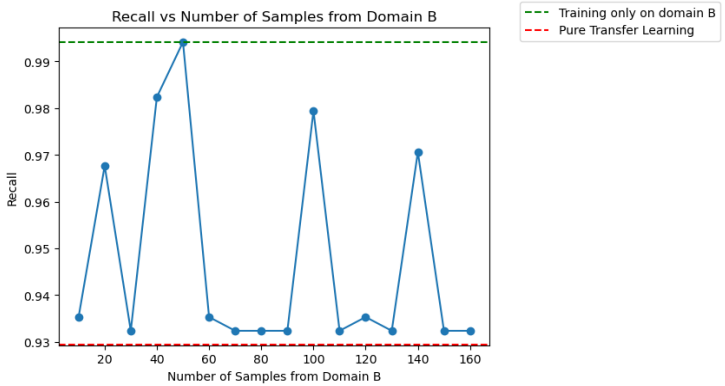
2- Domain adaptation:

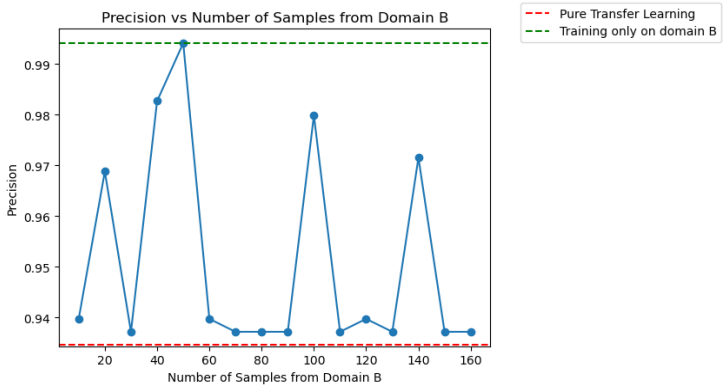
a) We performed training and hyper parameter tuning on the domain A
b) In each step, the model is re-trained using batches of 10 images from the domain B (16 steps)
c) We evaluate the resulting models on a test set of unseen images belong to domain B

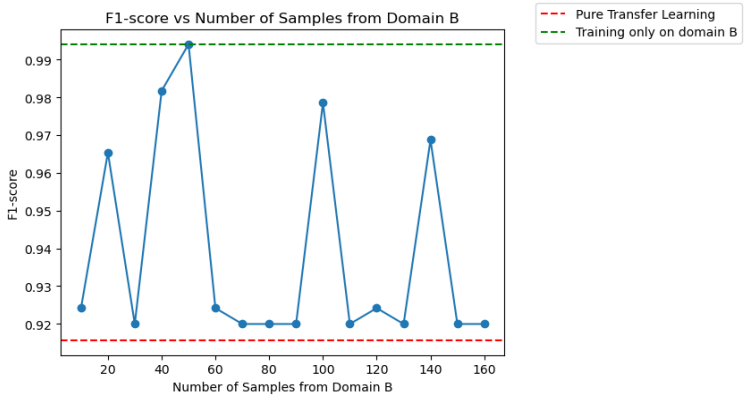
9. Transfer Learning

9.1. Original Image: One Class SVM with Transfer Learning







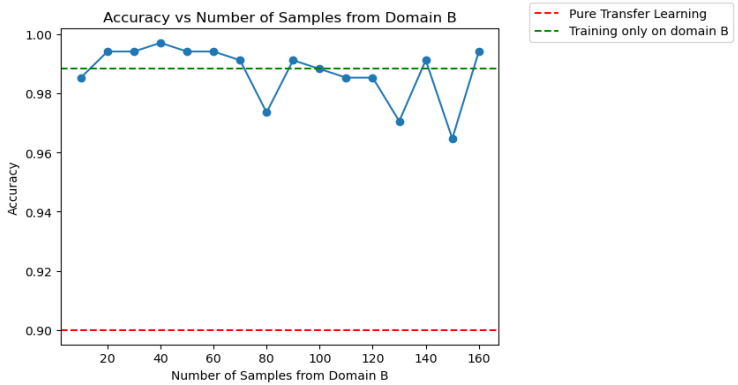


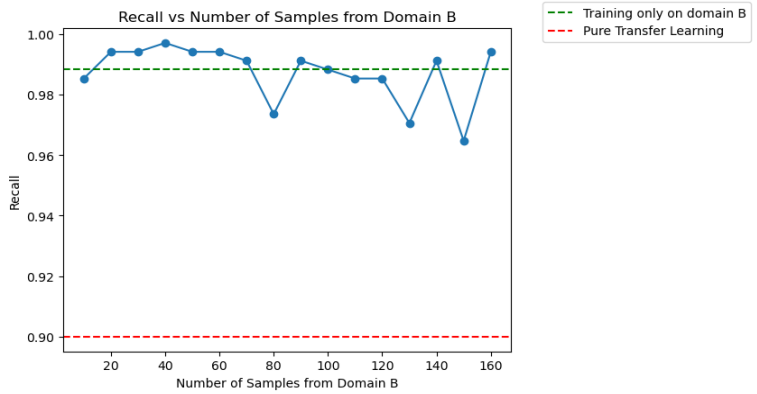
Observation

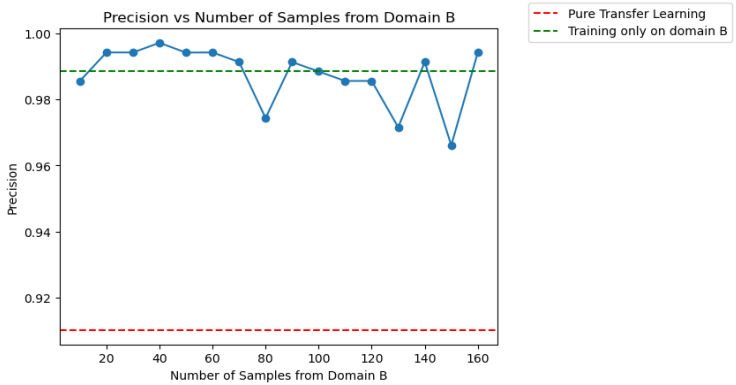
We have observed a fluctuated behavior while performing domain adaptation.
The maximum performance has been obtained by adding **50** images of domain B to the training set.

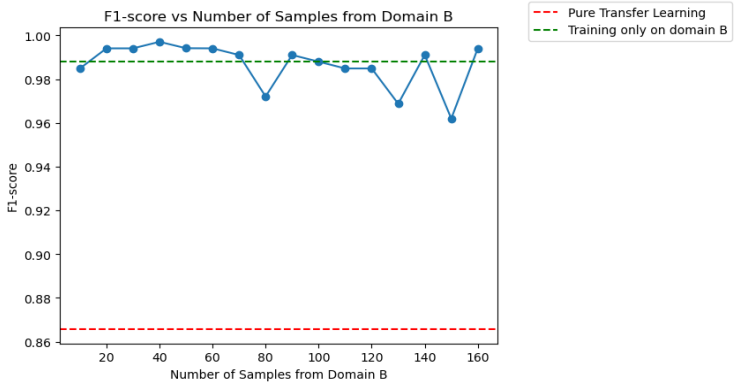
9. Transfer Learning

9.2. Original Image: Isolation Forest with Transfer Learning









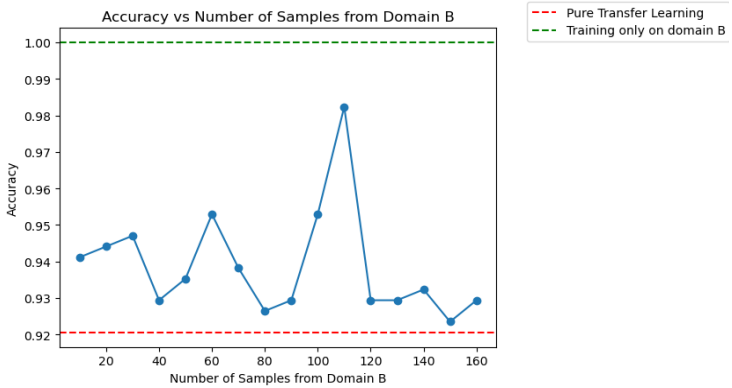
Observation

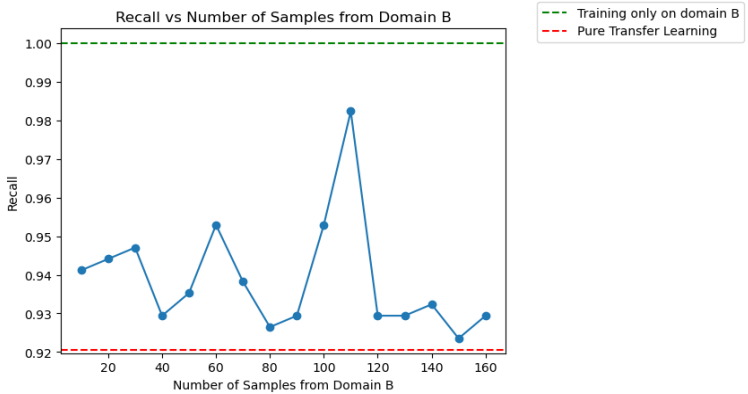
We have observed an less fluctuated behavior compared to OC-SVM in domain adaptation process.

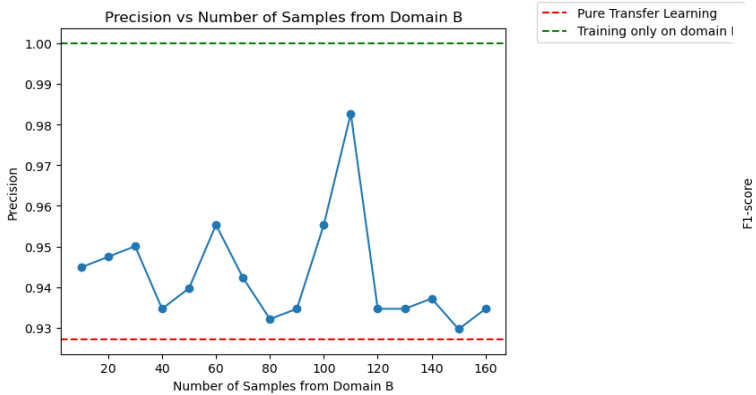
The best result gained by using **all** the images from the domain B and outperformed the performance obtained by the model trained only on the domain B.

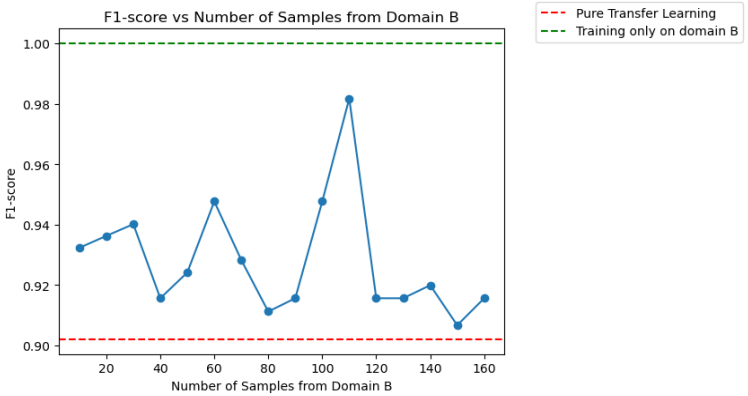
9. Transfer Learning

9.3. One Symbol: One Class SVM with Transfer Learning







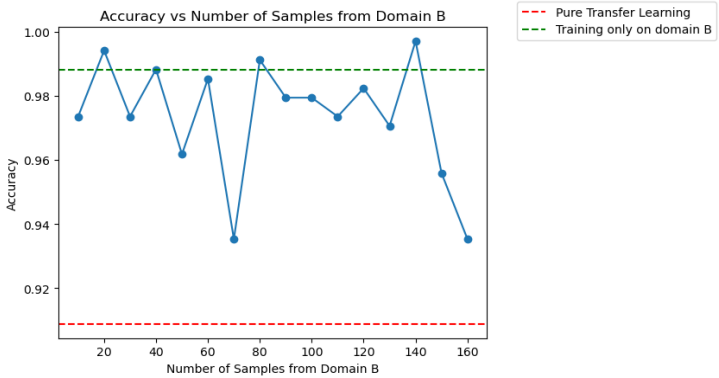


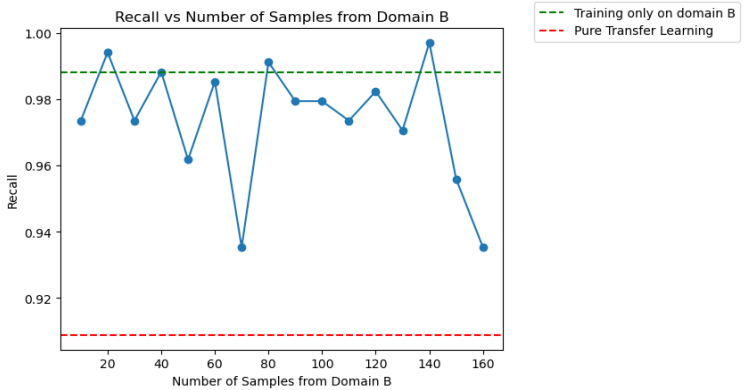
Observation

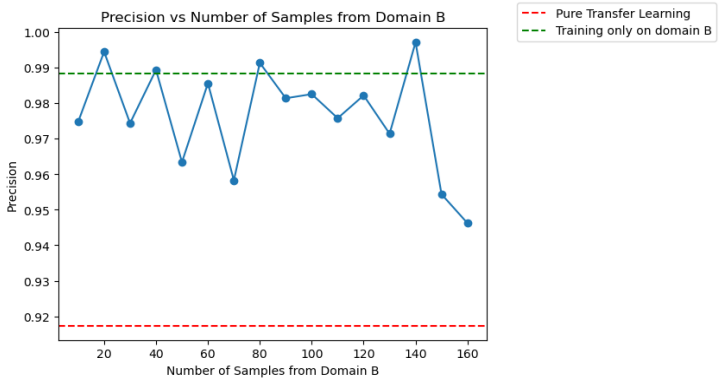
We have observed a fluctuated behavior while performing domain adaptation. The maximum performance has been obtained by adding **110** images of domain B to the training set. However, the performance of models were still below the model that had been trained only on the domain B.

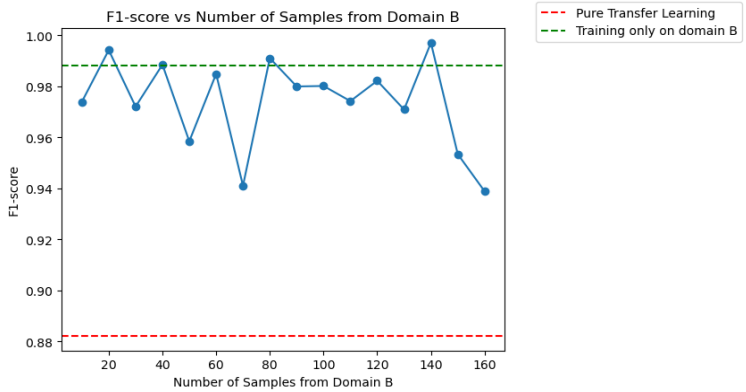
9. Transfer Learning

9.4. One Symbol: Isolation Forest with Transfer Learning









Observation

We have observed a fluctuated behavior while performing domain adaptation. By adding **140** images of domain B to the training set, we could outperform the performance of the model trained only on domain B. However, adding more images resulted in decrease in the performances of models.

9. Transfer Learning

Remark: Whether [OSNR for domain A: 25dB and OSNR for domain B: 40dB] or [OSNR for domain A: 40dB and OSNR for domain B: 25dB], the results for Transfer Learning do not coincide with the theory.

10. Alternative: Covariance feature space

Approach:

1- extract statistical features (Mean, Variance, skewness and Kurtosis) of images along the column, in both spatial and Fourier domain.

2- Create a new feature space based on the covariance matrix of each feature vector of each image and check the sparebility of this new feature space

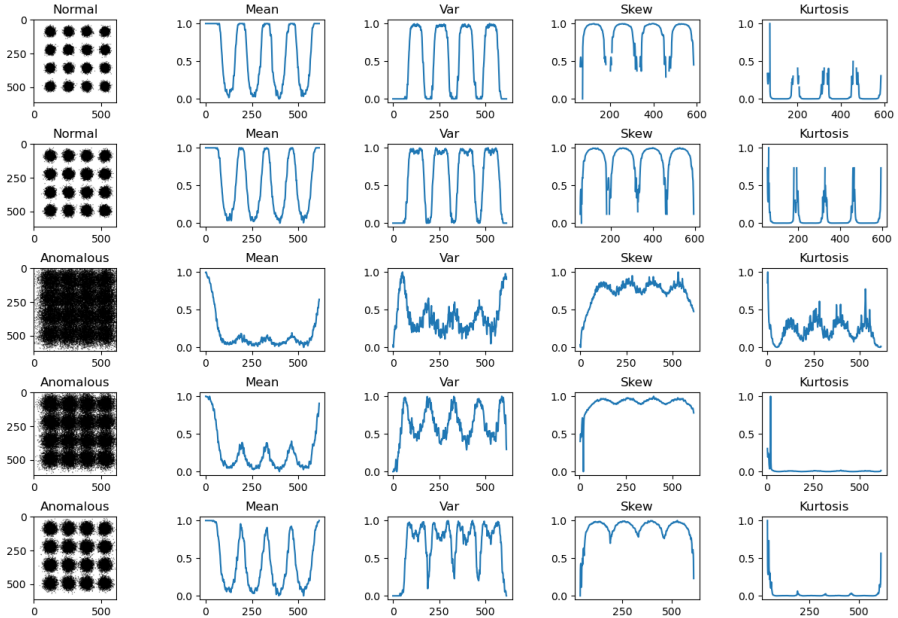
3- PCA

4- Training both models on the covariance feature space (with and without PCA)

- Experiments were performed on the dataset with **OSNR 25 dB**

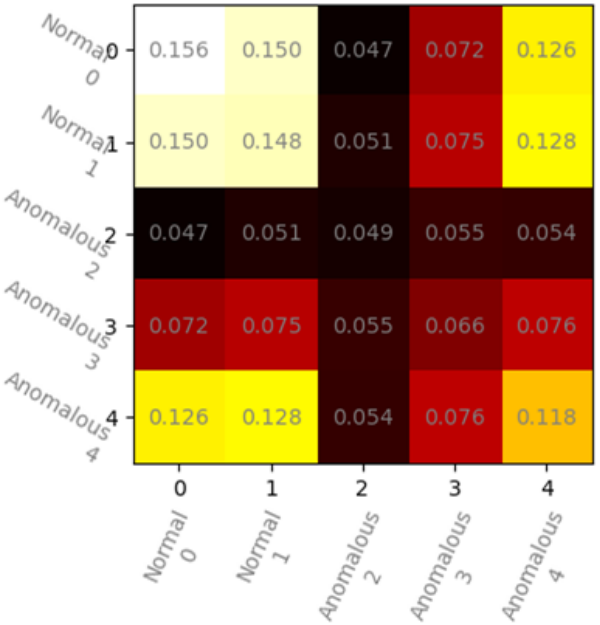
10. Alternative: Covariance feature space

Example of extracted features from 5 images with different level of anomaly:



the corresponding mean feature covariance matrices:

Covariance matrix of spatial mean feature (5 by 5)



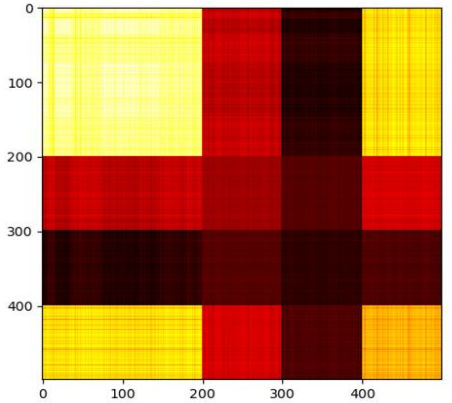
10. Alternative: Covariance feature space

10.1. Separability visualization

We visualized the covariance matrix corresponding to the **spatial mean feature** of all the images (500) of the dataset. It can be seen that the **first 200** covariance values shows less distances w.r.t each other compared to the other (201 to 500) values.

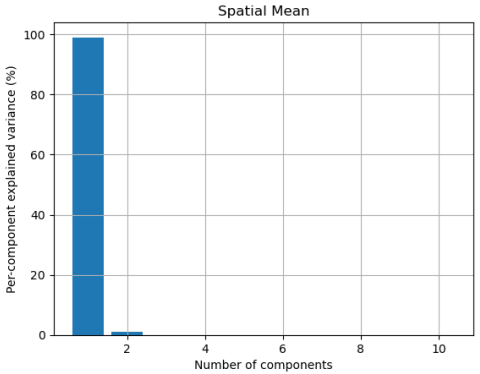
* we put normal images as the first 200 images for the sake of visualization

Covariance matrix of spatial mean feature (500 by 500)



10.2. PCA

The first principle component has approx. 100% explained variance ratio



10. Alternative: Covariance feature space

10.3. Mean covariance feature + PCA: One class SVM vs. Isolation Forest

Crossval time: 0.135 [s]

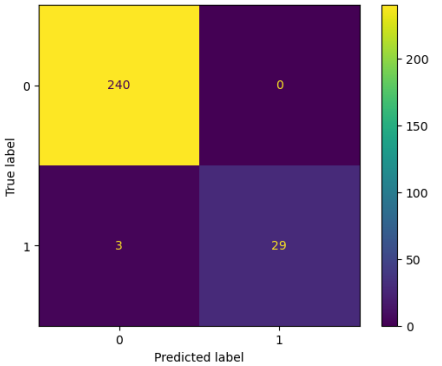
Best hyperparams during crossval: {'nu': 0.01, 'kernel': 'sigmoid', 'gamma': 'auto'}

Accuracy of the final model on validation: 0.99

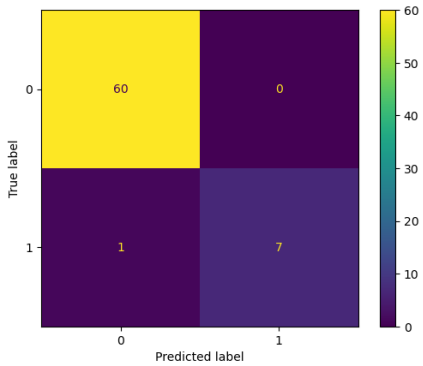
Average K-fold accuracy: 0.94+- 0.03

Test Accuracy: 0.99

Cross Validation



Testing



Crossval time : 21.481 [s]

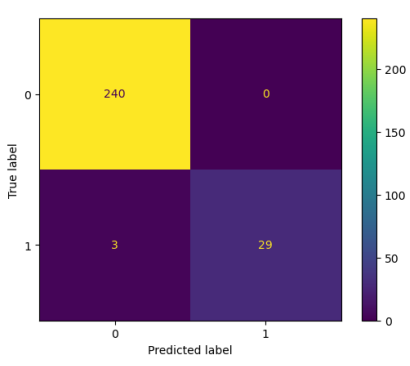
Best hyperparams during crossval: {'n_estimators': 50, 'contamination': 0.1, 'max_samples': 10}

Accuracy of the final model on validation: 0.99

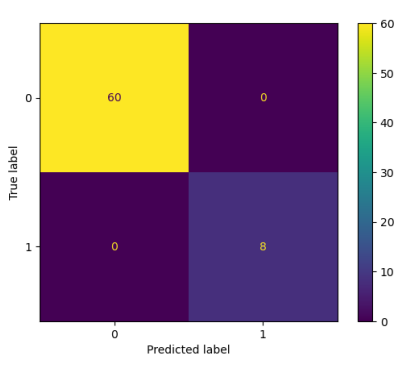
Average K-fold accuracy: 0.94 +- 0.03

Test Accuracy: 1.0

Cross Validation

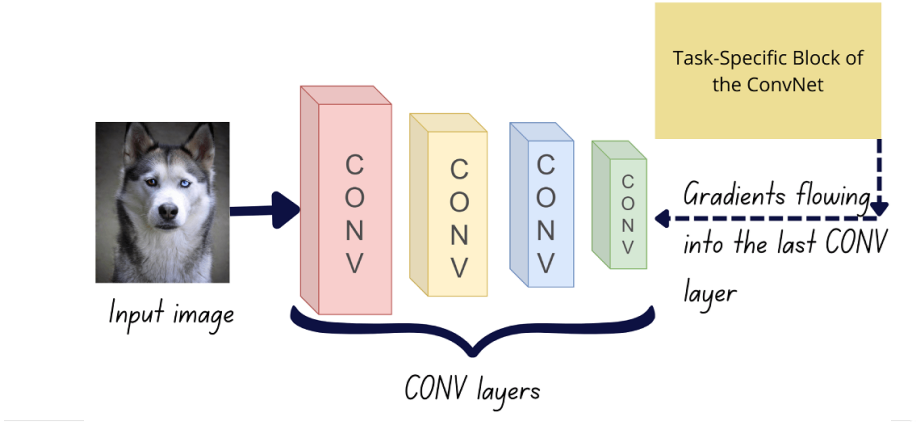


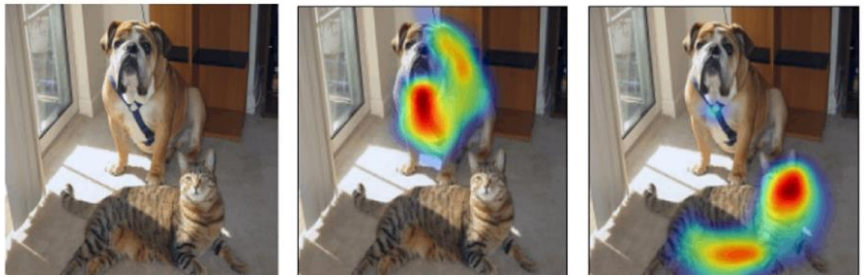
Testing



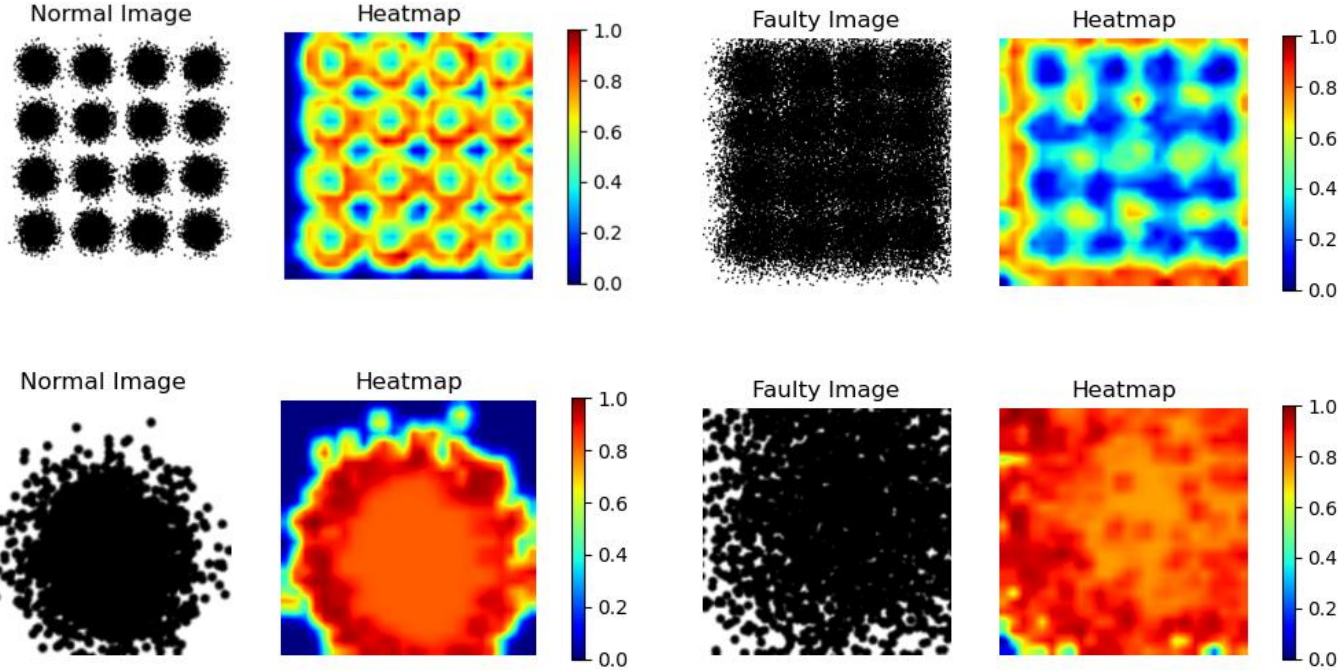
11. XAI: Gradient-weighted Class Activation Mapping (GradCAM)

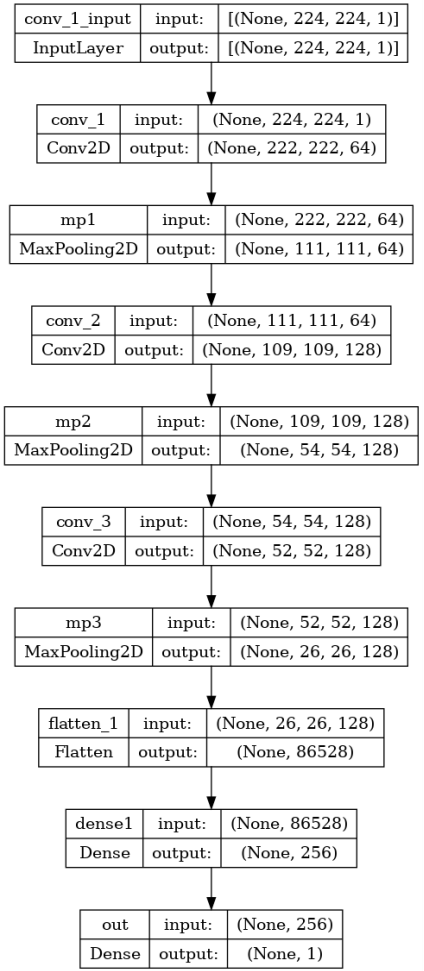
GradCAM is a technique used in computer vision to visualize the regions of an image that are important for a deep learning model's prediction. It achieves this by highlighting the most influential areas of the image through a gradient-based localization approach, providing insights into the model's decision-making process.





11. XAI: GradCam





12. Conclusions

1) In terms of the **final performance** of algorithms on images (full, resized, one-symbol), there is insignificant advantage of one over the other. In terms of CV time, OC-SVM significantly outperforms IF (at least 4 times faster).
2) **PCA** decreases the number of features, thus decreases the training time, although the IF gets a significantly worse performance than OC-SVM (doesn't manage to identify faulty samples). However, in the alternative approach, using only the first component of covariance feature space led us to obtain sufficient results for both OC-SVM and IF in the fastest time.
3) Likely, due to high dimensionality of the problem, overally, **TL** implementation fails to meet the expected results.
4) In **classification scenario**, the separation between the classes heavily relies on the most distant points (ex. An outlier is likely to have black dots further from the center wrt. normal class).