POLITECNICO
MILANO 1863

Network Measurement and Data Analysis Lab

Professor: Francesco Musumeci

Sajad Hamzenejadi
Kiarash Rezaei
Usevalad Milasheuski

Contact: Sajad Hamzenejadi
sajad.hamzenejadi@mail.polimi.it
sajadhamzenejadi76@gmail.com

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)

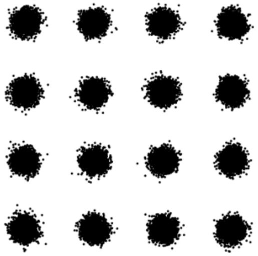
Feature Extraction Methods: 1- Flattening and resizing 2- statistical feature extraction and covariance matrices

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

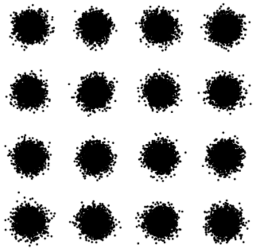
Advanced approaches: 1- Transfer Learning 2- Grad-CAM (briefly)

2. Datasets

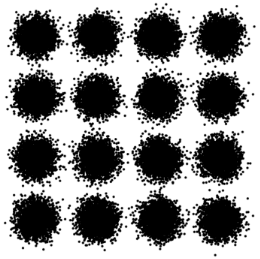
2.1 Full images



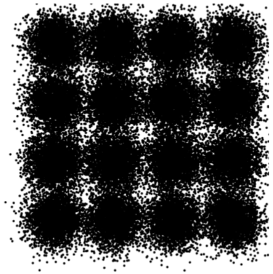
Normal 1



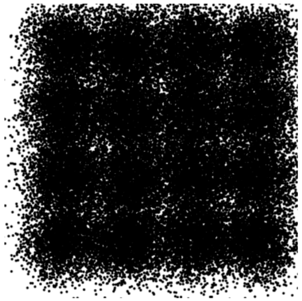
Normal 2



Fault 1



Fault 2



Fault 3

Dataset size = 500 (100 per each image)

image dimension = 614 by 614

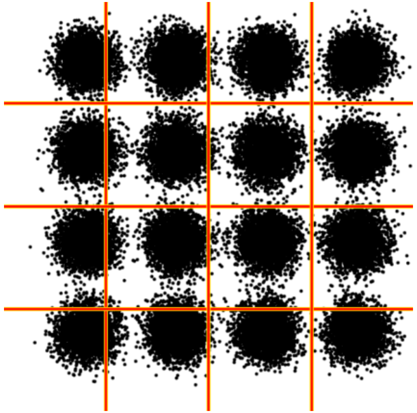
Classes = {Normal : 1, Faulty:-1}

2. Datasets

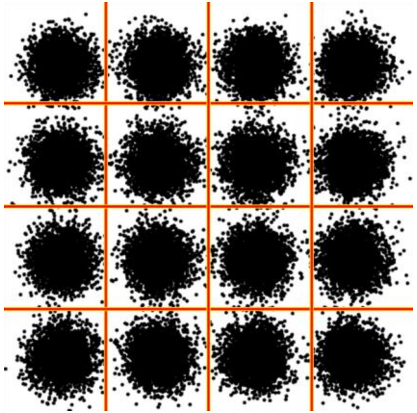
2.2. Creating one-symbol dataset

- Cropped 44 pixels from the left and bottom of the original 614x614 images to centre the sub-images for improved separation.

- Extracted 16 sub-images from each processed image.



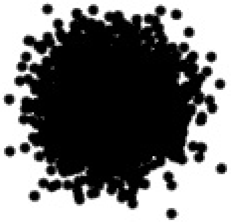
614*614: Before cropping

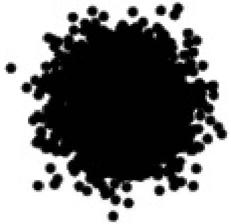


570*570: After cropping

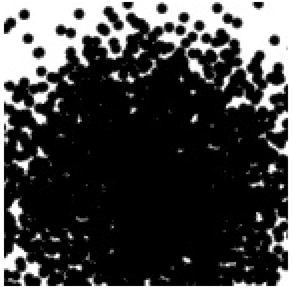
2. Datasets

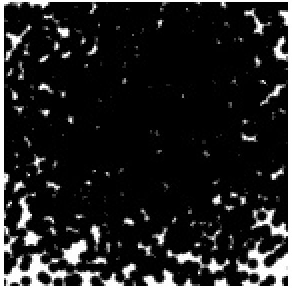
2.2 one-symbol images


Normal 1


Normal 2


Fault 1


Fault 2

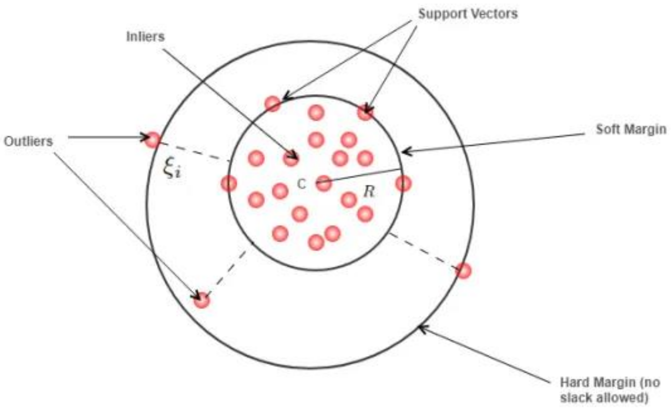

Fault 3

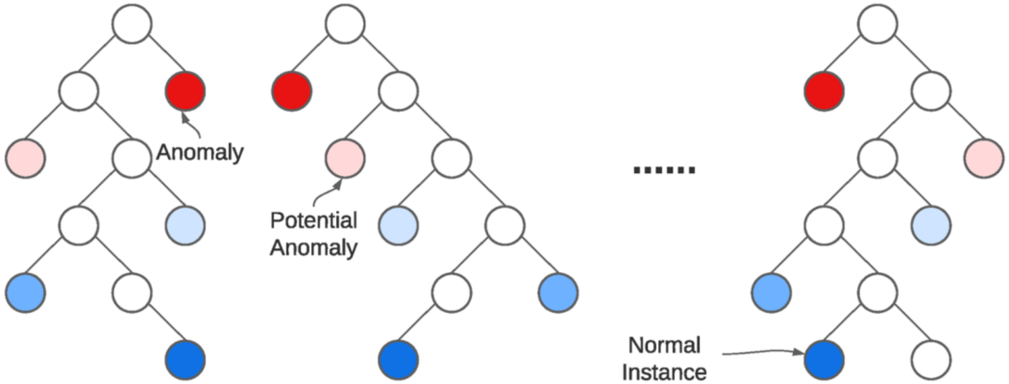
Dataset size = 500 (100 per each image, we use only one symbol per constellation!)

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 hyperparameter)

- **Definition:** In Isolation Forest, the "contamination" hyperparameter estimates the proportion of outliers or anomalies in the dataset.
- **Purpose:** Utilized during model fitting to establish the threshold for anomaly detection.
- **Cross-Validation (CV) Significance:**
  - Cross-Validation is employed to choose the optimal "contamination" value.
  - Given that folds include a combination of normal and faulty data, CV aids in selecting the most suitable hyperparameter.
- **Generalization:** The same principle is applicable to other hyperparameters within the Isolation Forest model.

4. Data Splitting

Methodology: Data was divided into training, validation, and test subsets, employing a fixed random state value for consistency.

Training Set Composition:

- Exclusively comprises normal samples.

Dataset Statistics:

- Training Dataset: 160 samples
- Validation Dataset: 272 samples (Faulty=240, Normal=32)
- Testing Dataset: 68 samples (Faulty=60, Normal=8)

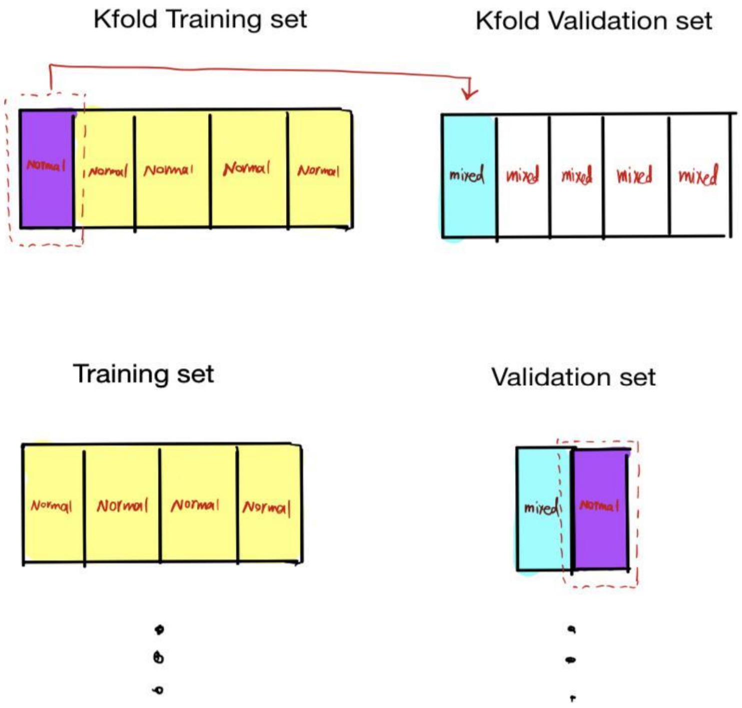
5. Cross-validation

Hyperparameter Space Definition: Established a hyperparameter space grid for each model independently.

Cross-Validation Procedure: Executed a 5-fold cross-validation, as depicted in the accompanying image.

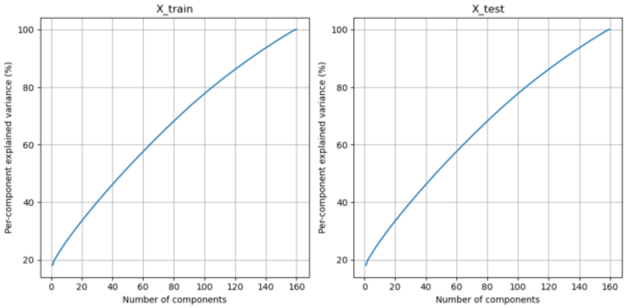
Objective: Fine-tuned hyperparameters to derive the most generalizable model.

Hyperparameter Space Configuration: The hyperparameter space for both models encompasses a grid with dimensions 3*3*2, totalling 18 elements.

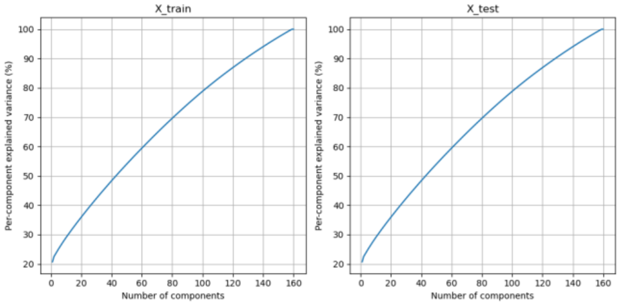


6. Principle Component Analysis (PCA)

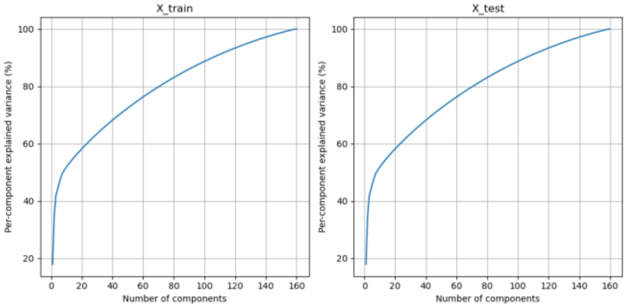
6.1. Original Images + PCA



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

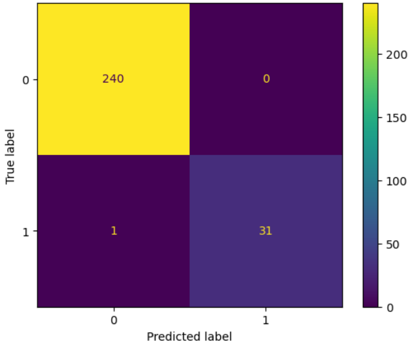
Best hyperparams during crossval: {'n_estimators': 100, 'contamination': 0.1, 'max_samples': 50}

Accuracy of the final model on validation: 0.996

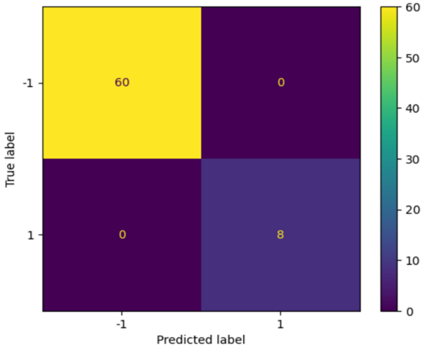
Average K-fold accuracy: 0.984 +- 0.009

Test Accuracy: 1.0

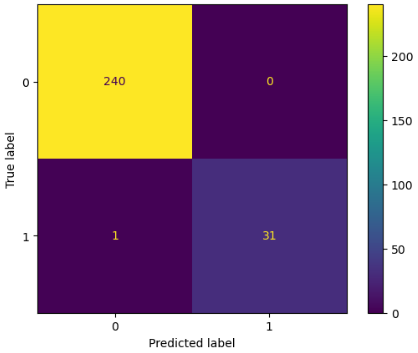
Cross Validation



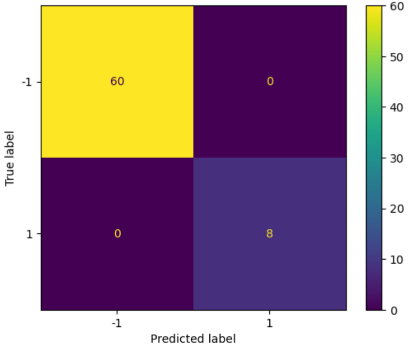
Testing



Cross Validation



Testing



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

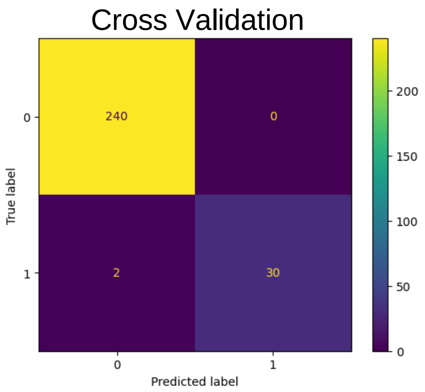
Crossval time 00:00:16

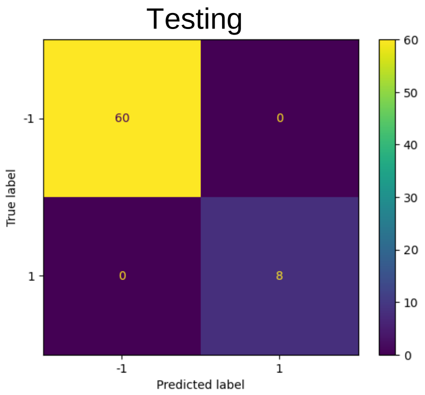
Best hyperparams during crossval: {'n_estimators': 50, 'contamination': 0.1, 'max_samples': 100}

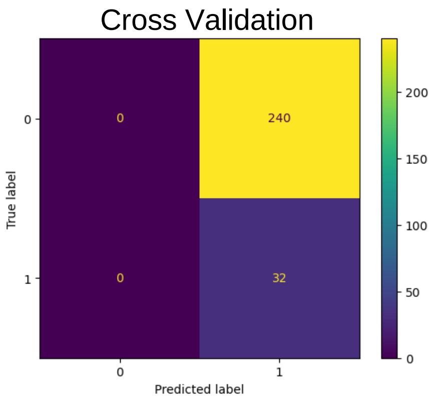
Accuracy of the final model on validation: 0.118

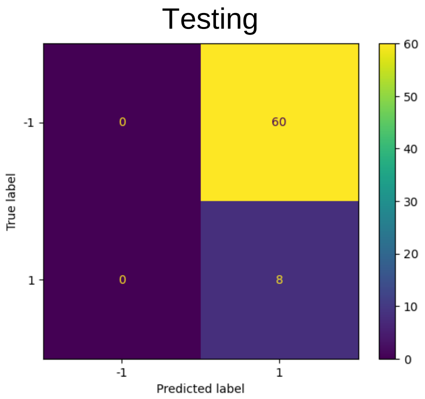
Average K-fold accuracy: 0.403 +- 0.028

Test Accuracy: 0.118









6250 features

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

Crossval time 00:02:53

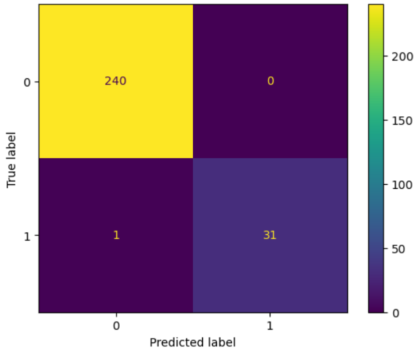
Best hyperparams during crossval: {'n_estimators': 75, 'contamination': 0.1, 'max_samples': 50}

Accuracy of the final model on validation: 0.993

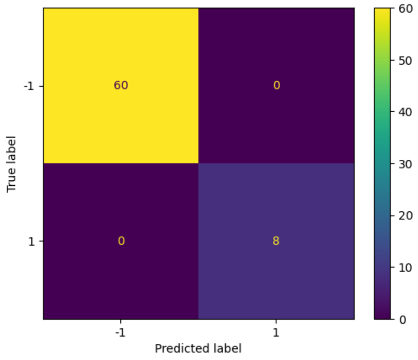
Average K-fold accuracy: 0.986 +- 0.009

Test Accuracy: 1.0

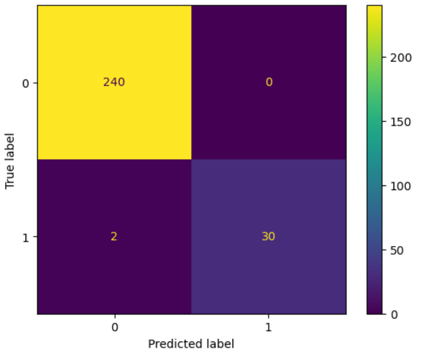
Cross Validation



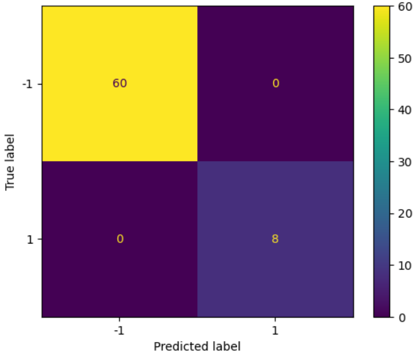
Testing



Cross Validation



Testing



7. Results

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

Crossval time < 1sec

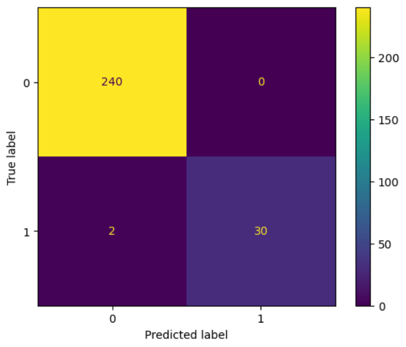
Best hyperparams during crossval: {'nu': 0.01, 'kernel': 'sigmoid', 'gamma': 'auto'}

Accuracy of the final model on validation: 0.993

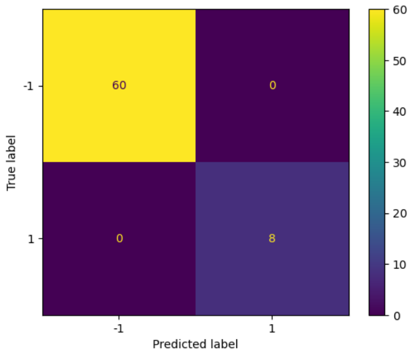
Average K-fold accuracy: 0.851+- 0.621

Test Accuracy: 1.0

Cross Validation



Testing



Crossval time 00:00:16

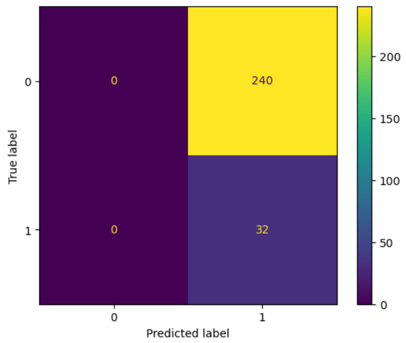
Best hyperparams during crossval: {'n_estimators': 75,, 'contamination': 0.1, 'max_samples': 50}

Accuracy of the final model on validation: 0.118

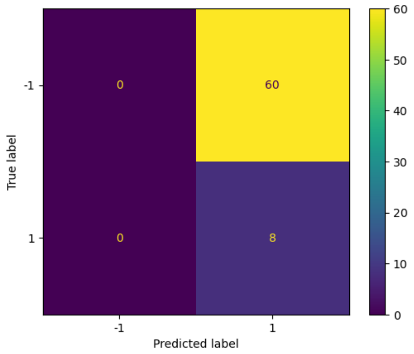
Average K-fold accuracy: 0.412 +- 0.041

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

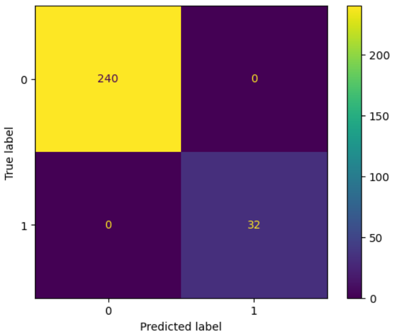
Best hyperparams during crossval: {'nu': 0.01, 'kernel': 'sigmoid', 'gamma': 'auto'}

Accuracy of the final model on validation: 1.0

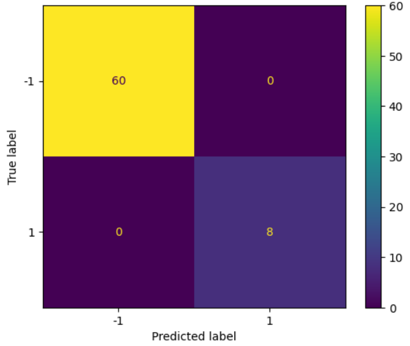
Average K-fold accuracy: 0.986 +- 0.011

Test Accuracy: 1.0

Cross Validation



Testing



Crossval time 00:00:57

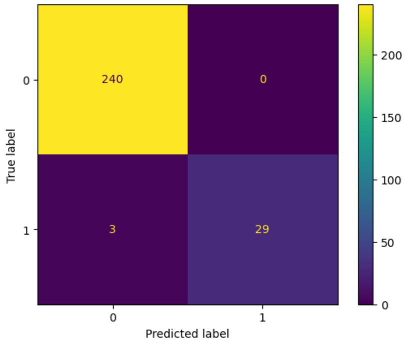
Best hyperparams during crossval: {'n_estimators': 75, 'contamination': 0.1, 'max_samples': 10}

Accuracy of the final model on validation: 0.989

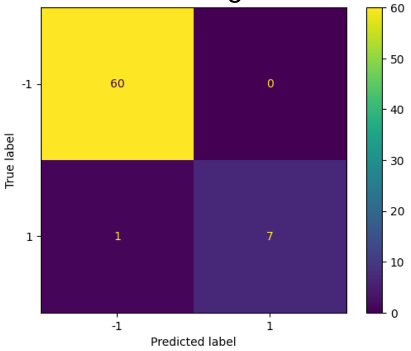
Average K-fold accuracy: 0.986 +- 0.011

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

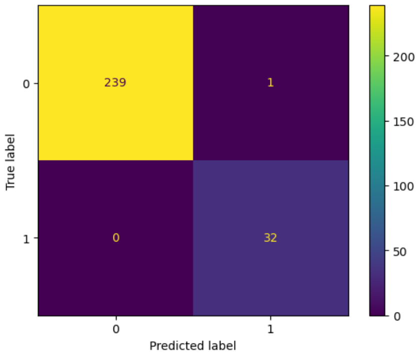
Best hyperparams during crossval: {'n_estimators': 100, 'contamination': 0.1, 'max_samples': 10}

Accuracy of the final model on validation: 0.206

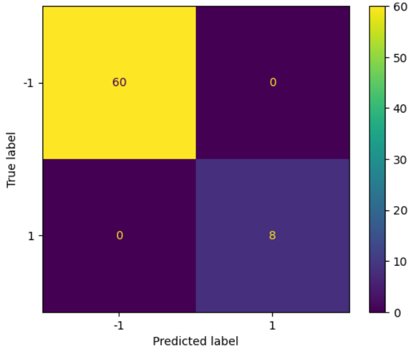
Average K-fold accuracy: `0.412 +- 0.019`

Test Accuracy: 0.22

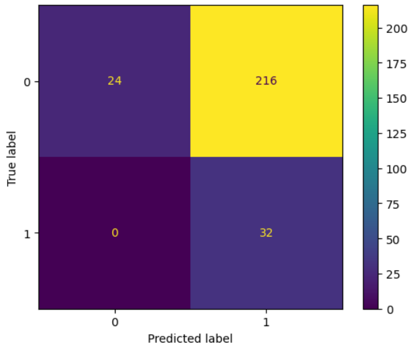
Cross Validation



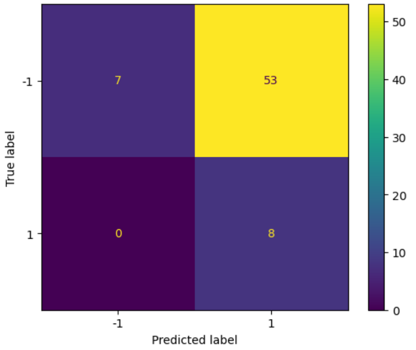
Testing



Cross Validation

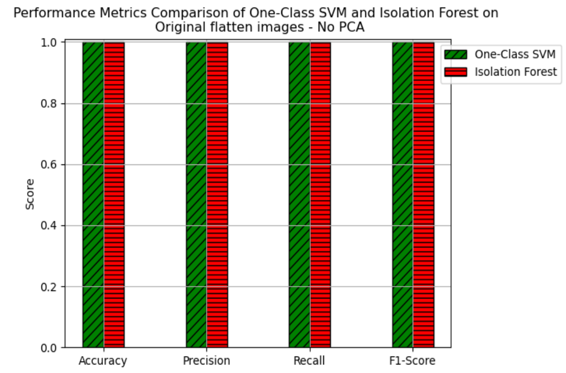


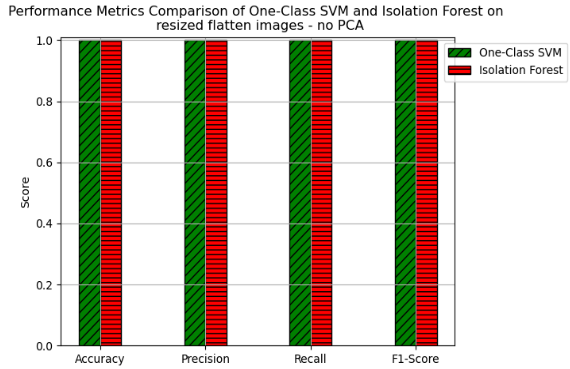
Testing

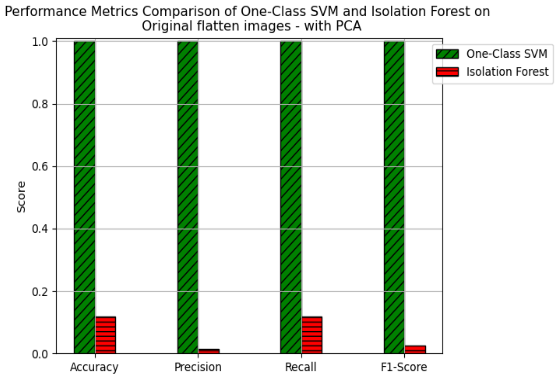


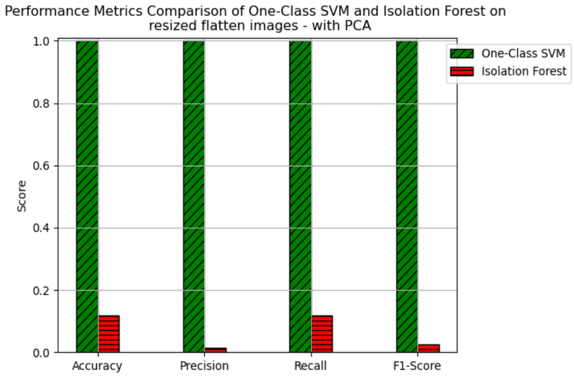
8. Comparison

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









Observation Using PCA

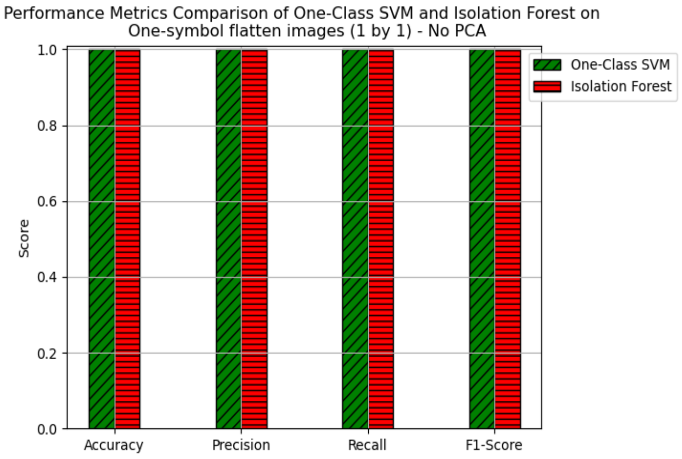
Isolation Forest:
Performance drop observed.

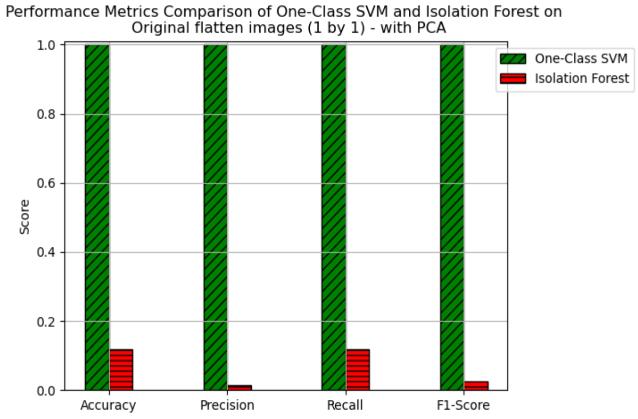
One-Class SVM:
Performance remains consistently high.

Note: Applies to both original and resized images.

8. Comparison

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





Observation using PCA

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

9. Transfer Learning

- **Datasets**

  - Domain A: OSNR = 25 dB
  - Domain B: OSNR = 40 dB (original and one-symbol images)

- **Approaches**

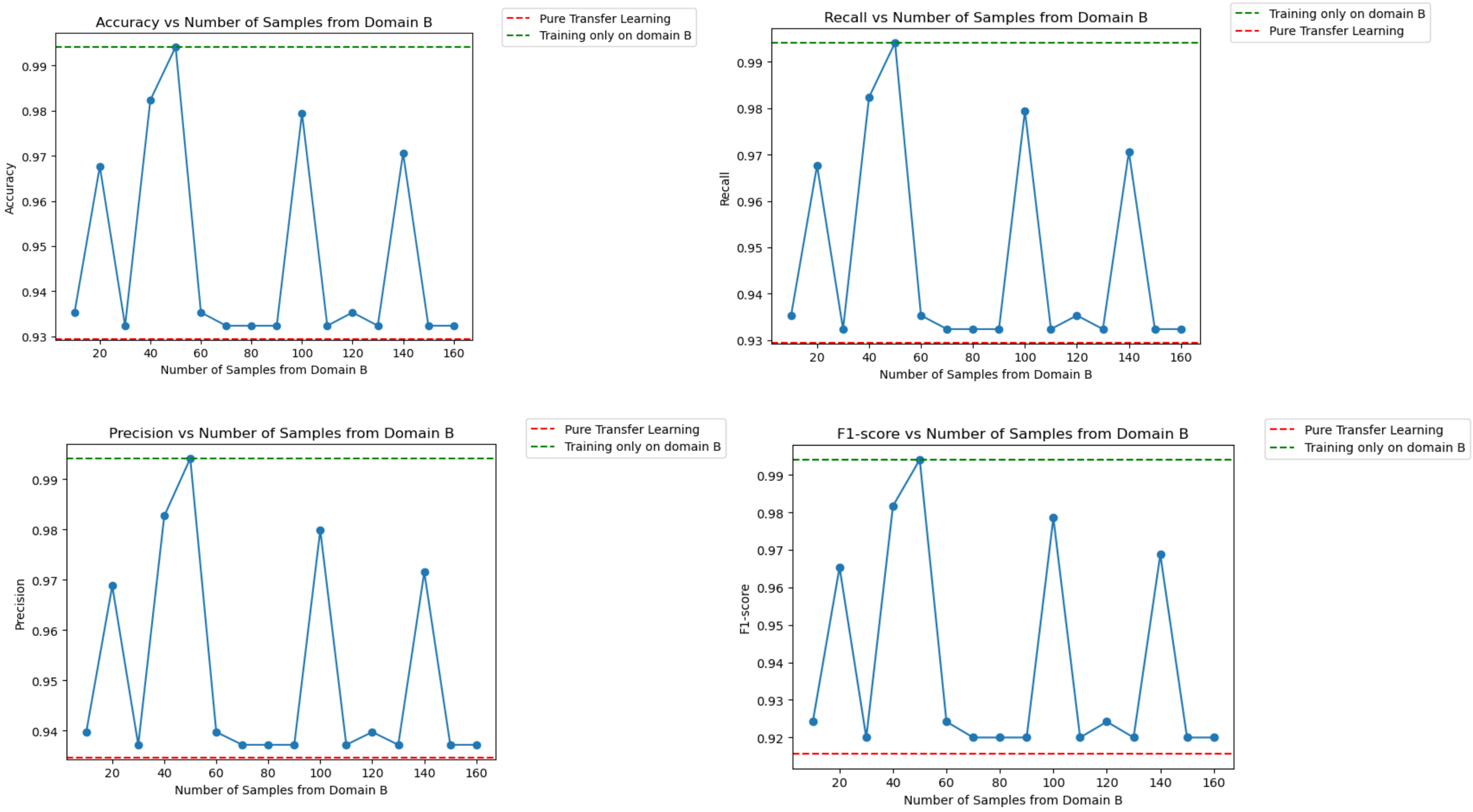
1. Pure Transfer Learning (Pure TL):
   - Train and tune on domain A.
   - Evaluate on domain B.

2. Domain Adaptation:
   - Initial training and tuning on domain A.
   - Sequential re-training on batches from domain B (16 steps).
   - Evaluate on a test set from domain B.

Note: OSNR denotes Optical Signal-to-Noise Ratio.

9. Transfer Learning

9.1. Original Image: One Class SVM with Transfer Learning



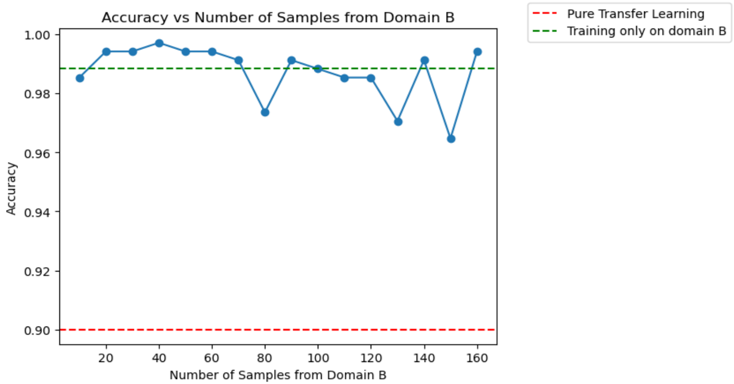
Observation

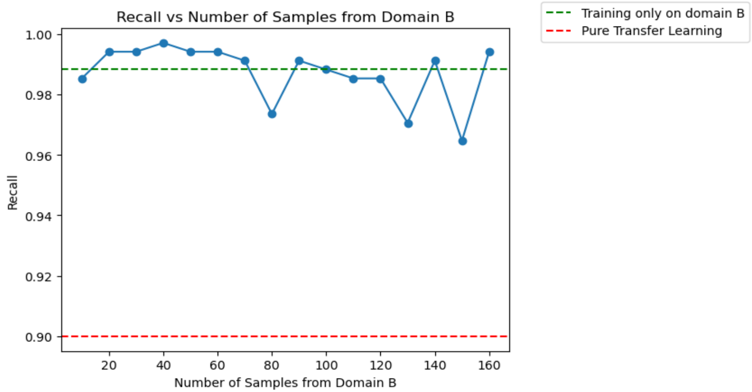
Domain Adaptation: Fluctuating behavior observed.

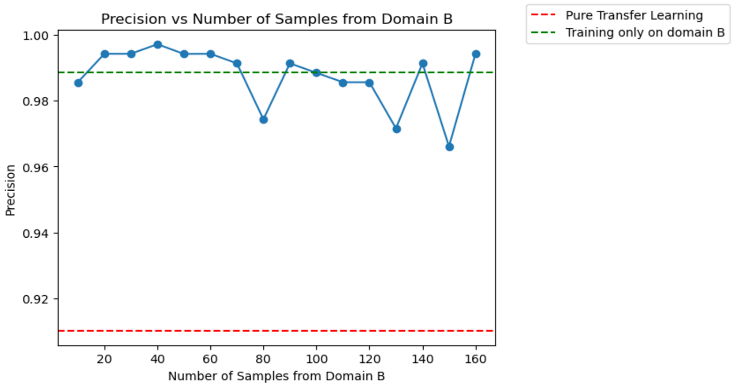
Optimal Result: Maximum performance achieved by adding 50 images from domain B to the training set.

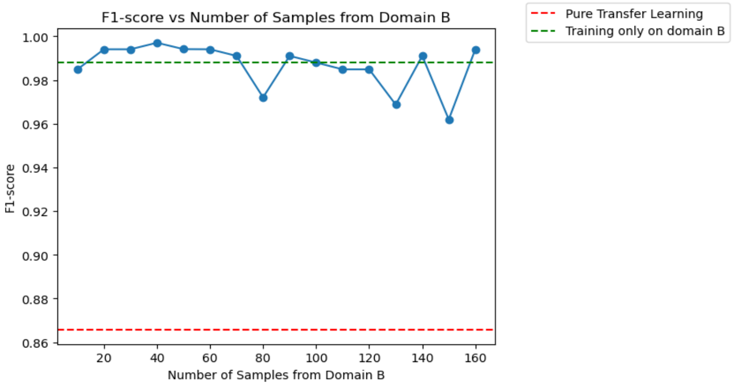
9. Transfer Learning

9.2. Original Image: Isolation Forest with Transfer Learning









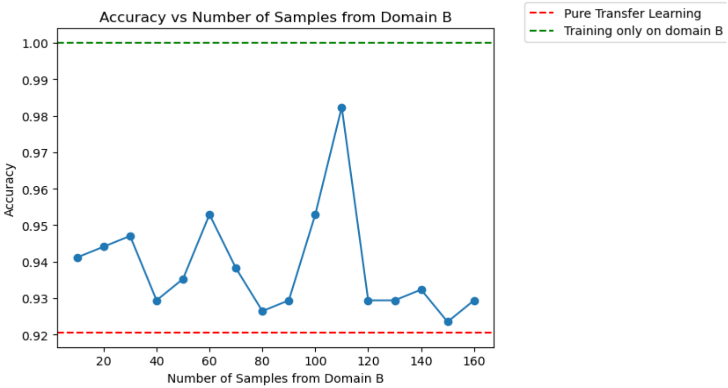
Observation

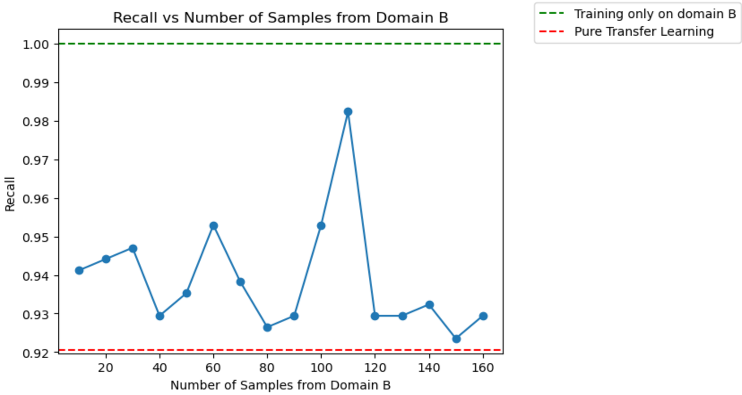
Domain Adaptation: Less fluctuation observed compared to OC-SVM in the adaptation process.

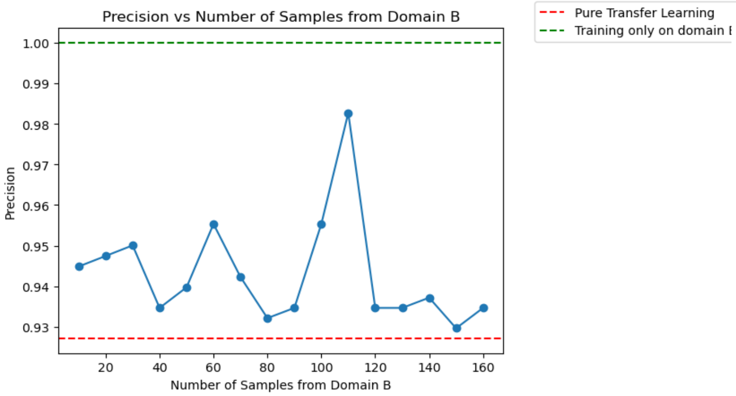
Optimal Result: Best performance achieved by utilizing all images from domain B, surpassing the performance of a model trained solely on domain B.

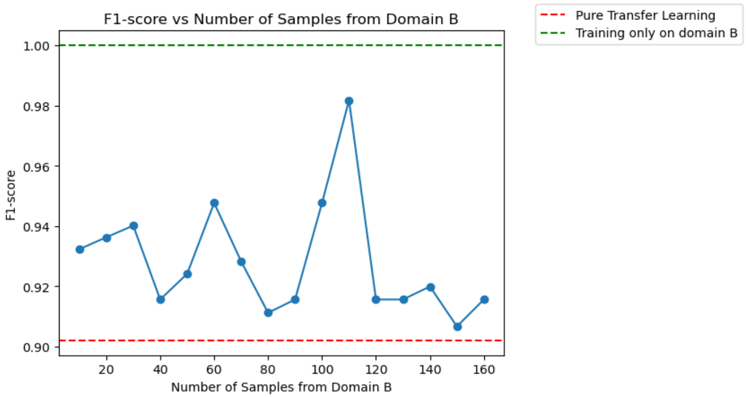
9. Transfer Learning

9.3. One Symbol: One Class SVM with Transfer Learning









Observation

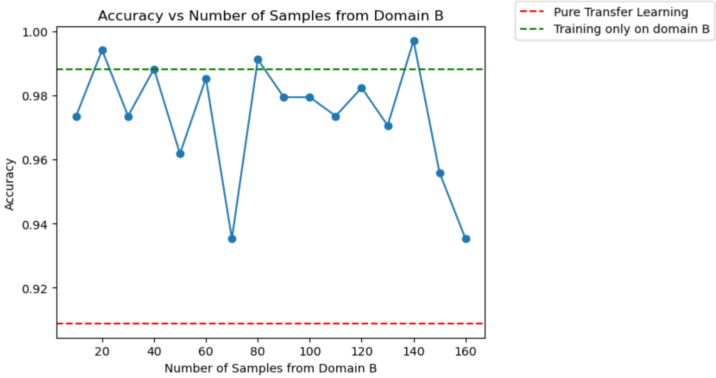
Domain Adaptation: Noticed fluctuating behavior during the domain adaptation process.

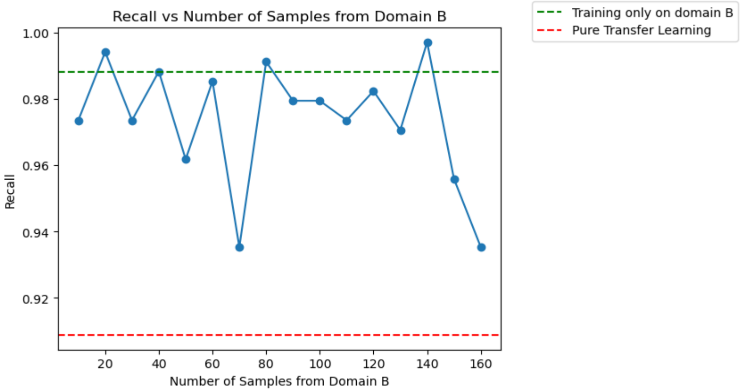
Optimal Result: Maximum performance achieved by adding 110 images from domain B to the training set.

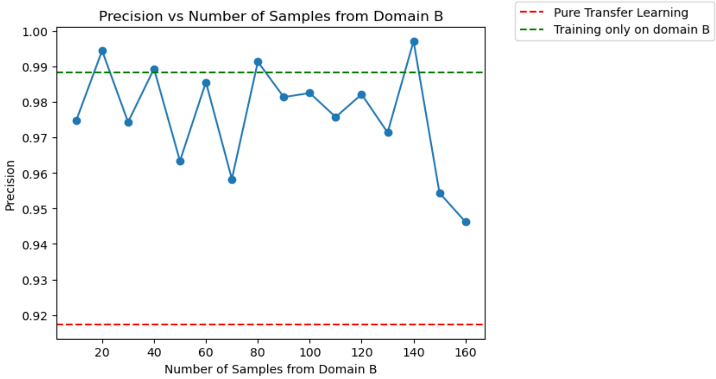
P.S. Despite augmentation, performance remained below that of a model trained exclusively on domain B.

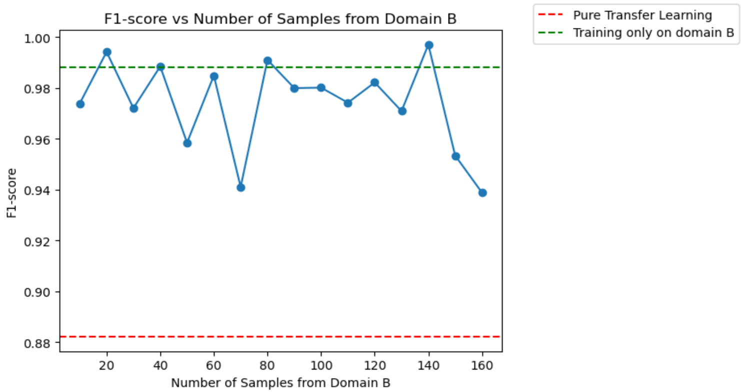
9. Transfer Learning

9.4. One Symbol: Isolation Forest with Transfer Learning









Observation

Domain Adaptation: Observed fluctuating behavior during domain adaptation.

Optimal Results: Achieved improved performance by adding 140 images from domain B to the training set.

P.S. Beyond 140 images, additional augmentation led to a decrease in model performance.

9. Transfer Learning

Scenario Variations:

- Whether [OSNR for domain A: 25dB and OSNR for domain B: 40dB] or [OSNR for domain A: 40dB and OSNR for domain B: 25dB].

Transfer Learning Discrepancy:

- Results observed in Transfer Learning do not align with theoretical expectations.

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

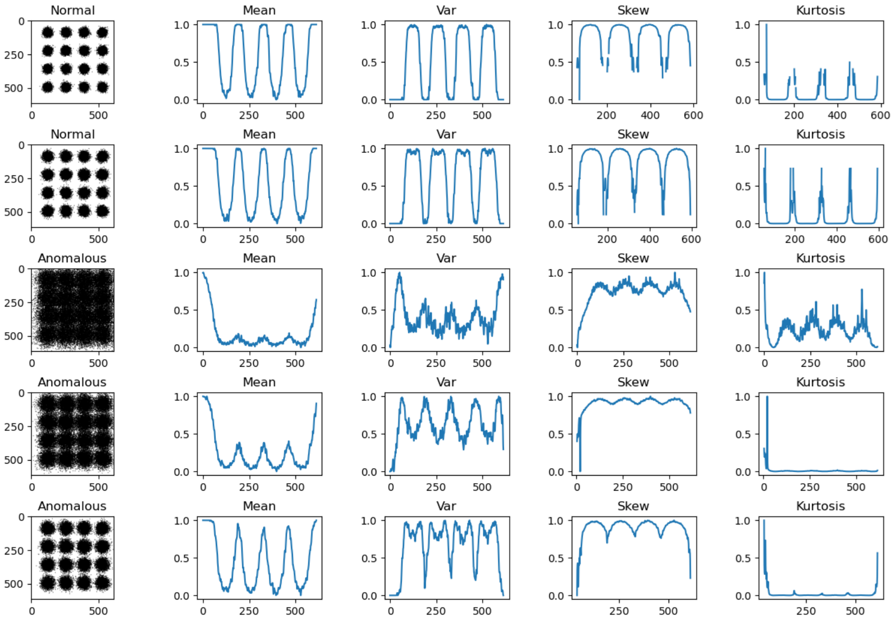
3. PCA

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

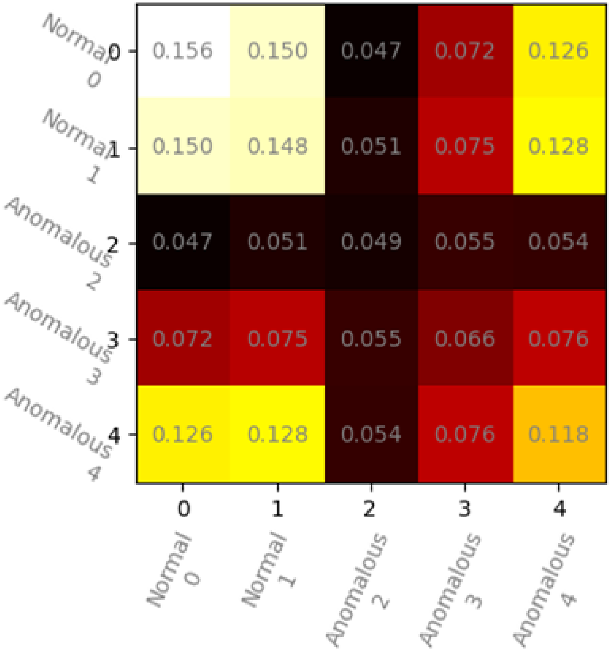
P.S. Experiments were performed on the dataset with **OSNR 25 dB**

10. Alternative: Covariance feature space

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



Covariance matrix of spatial mean feature (5 by 5)



10. Alternative: Covariance feature space

10.1. Separability visualization

Method: Visualization of the covariance matrix corresponding to the spatial mean feature of the entire dataset (500 images).

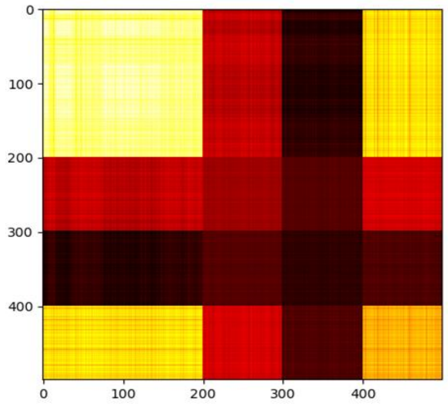
Observation: Noteworthy distinction in the covariance values: first 200 values (normal images) exhibit closer proximity compared to the remaining values (201 to 500).

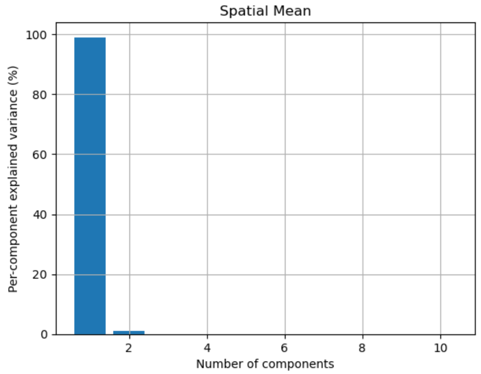
Note: Normal images intentionally placed as the first 200 for enhanced visualization clarity.

10.2. PCA

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

Covariance matrix of spatial mean feature (500 by 500)





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

Crossval time : 21.481 [s]

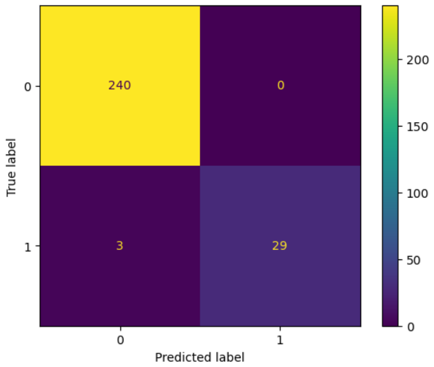
Best hyperparams during crossval: {'n_estimators': 50, 'contamination': 0.1, 'max_samples': 10}

Accuracy of the final model on validation: 0.99

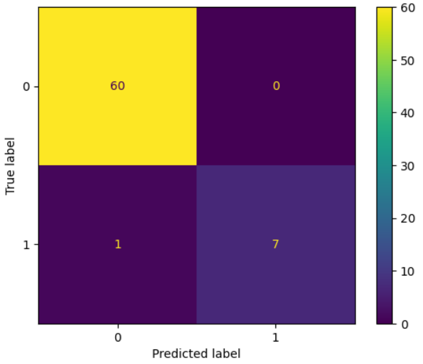
Average K-fold accuracy: 0.94 +- 0.03

Test Accuracy: 1.0

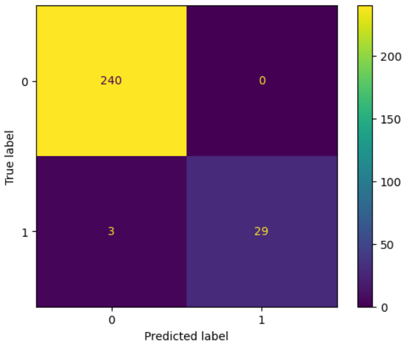
Cross Validation



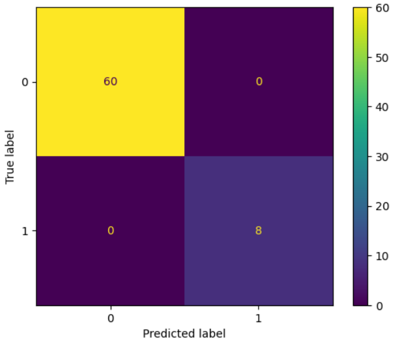
Testing



Cross Validation



Testing

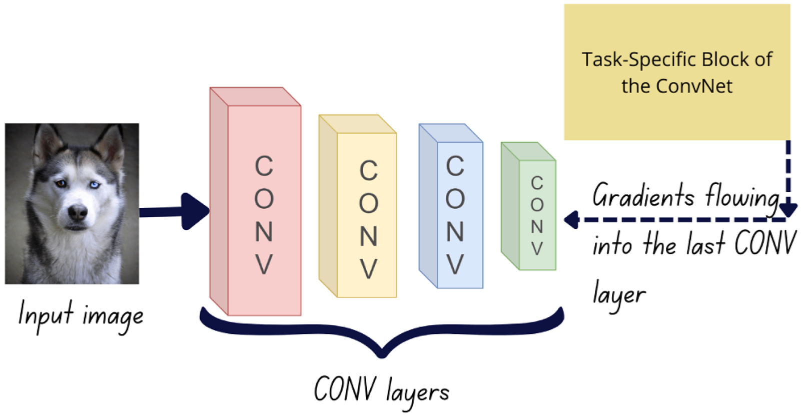


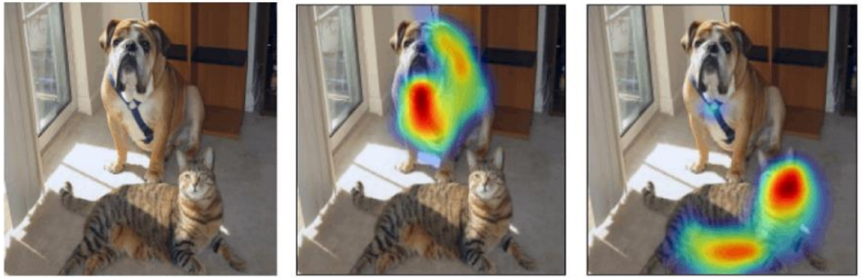
11. XAI: Gradient-weighted Class Activation Mapping (GradCAM)

Definition: GradCAM, employed in computer vision, visualizes significant regions in an image crucial for a deep learning model's prediction.

Methodology: Utilizes a gradient-based localization approach to highlight influential areas in the image.

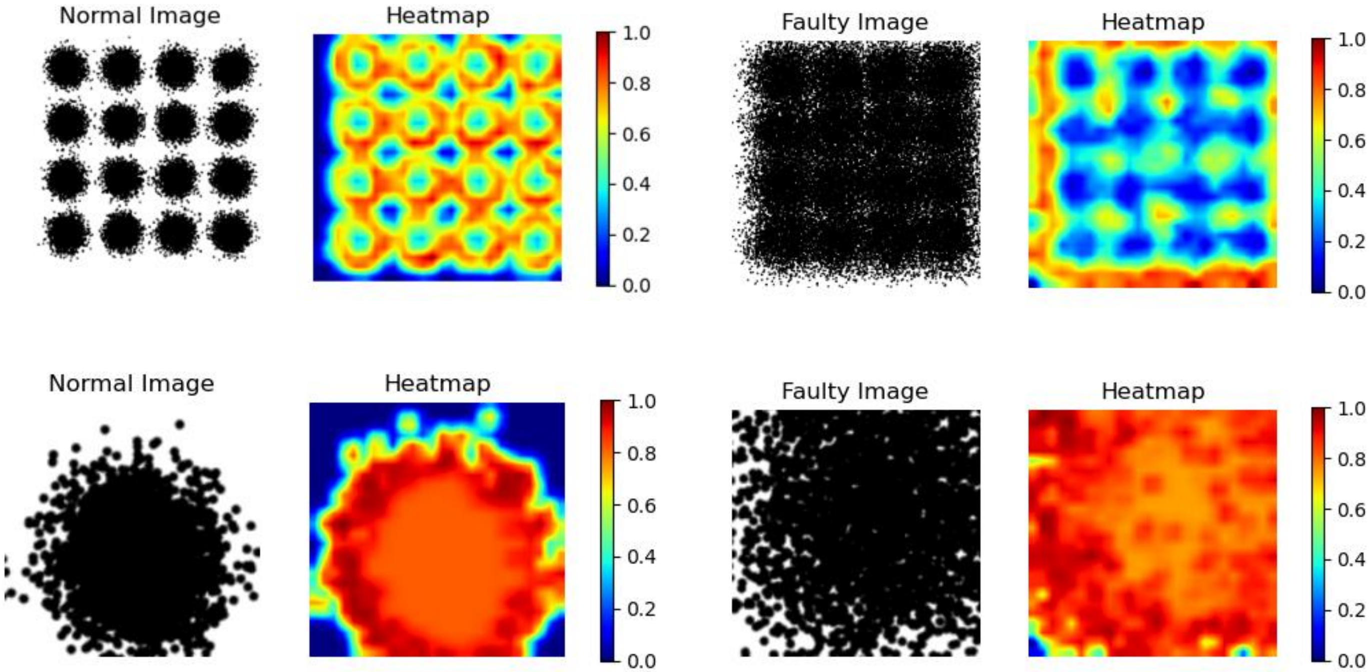
Purpose: Offers insights into the decision-making process of the deep learning model.

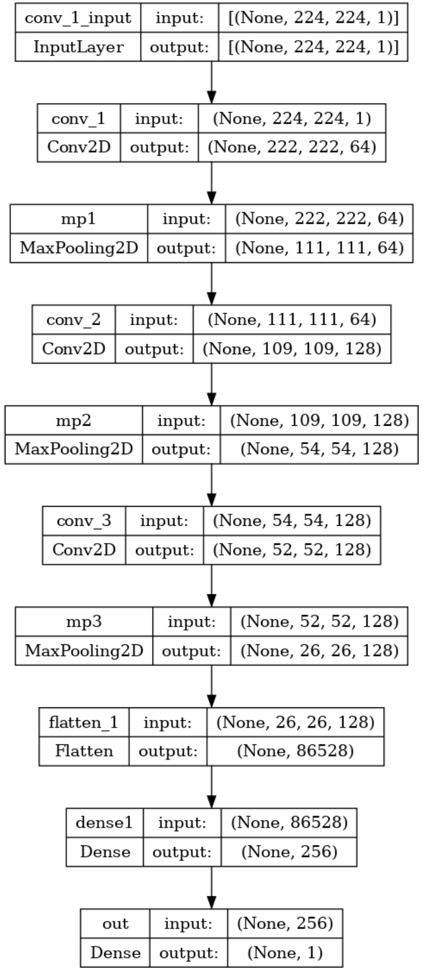




(a) Original Image (b) Cat Counterfactual exp (c) Dog Counterfactual exp

11. XAI: GradCam





12. Conclusion

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).