UDT
Undersea Defence Technology
13-15 May 2019
Stockholmsmässan, Sweden

False Alarm Reduction for Active Sonars using Deep Learning Architectures

Matthias Buß

University of Wuppertal

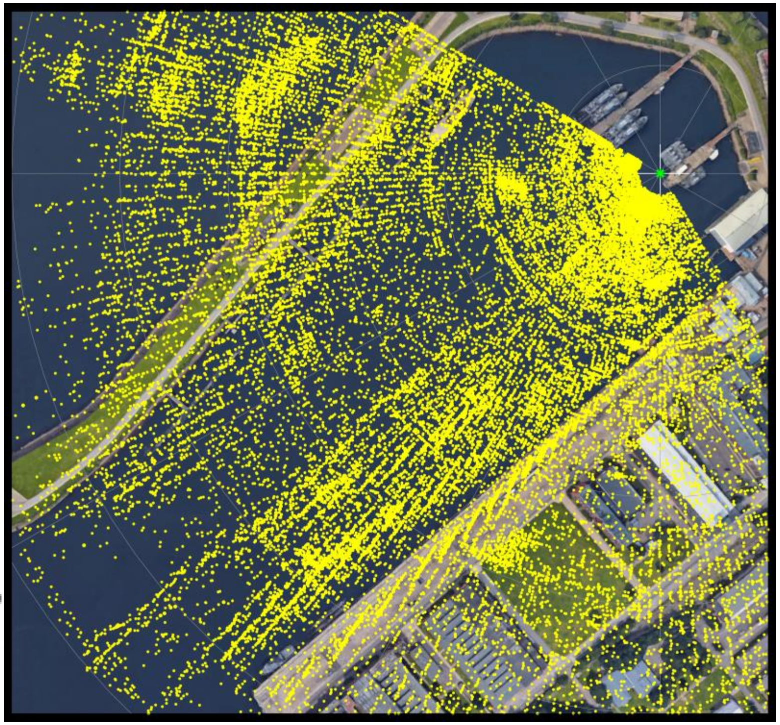
ATLAS ELEKTRONIK WTD 71 HSB Hochschule Bremen City University of Applied Sciences BERGISCHE UNIVERSITÄT WUPPERTAL

Agenda

- Motivation and Application
- Proposed Solution for False Alarm Reduction
- Feature Extraction and Classification
- Data Labelling
- Classification Results
- Summary and Future Work

Motivation

- The false alarm rate (FAR) represents a crucial aspect in all active sonar applications.
- Every contact is represented in the detection display.
- Under different circumstances it results in an enormous number of false contacts.



→ Tracking algorithms might be unable to deal with the large number of contacts.

→ An operator is not able to identify true target contacts.

Aim: Reduce number of false contacts without losing target contacts.

Application

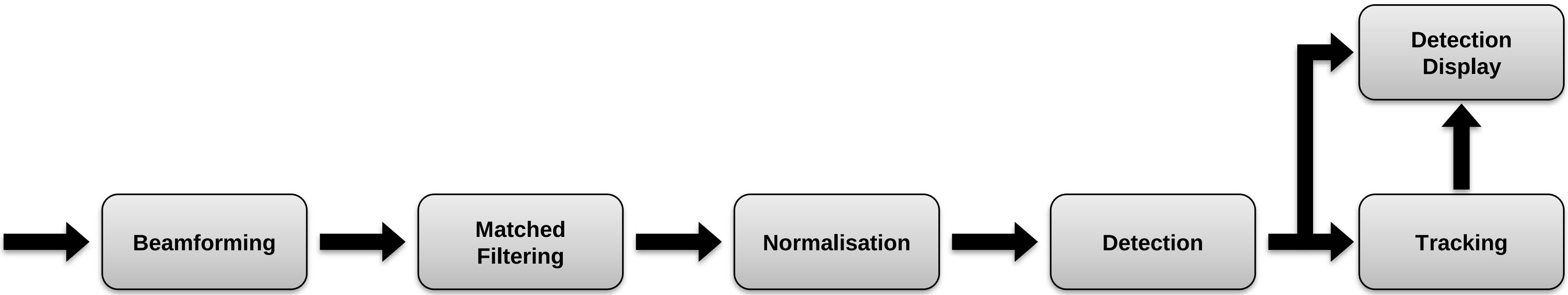
- The False Alarm Reduction is investigated for Active Diver Detection Sonar Data.
- Several Datasets recorded with a Cerberus DDS are provided by the WTD 71.
- Raw Data is processed with experimental active signal processing in MATLAB.
- All results are based on the transmission of Frequency Modulated (FM) Pulses.

Cerberus Diver Detection Sonars (left Mod1, right Mod2)

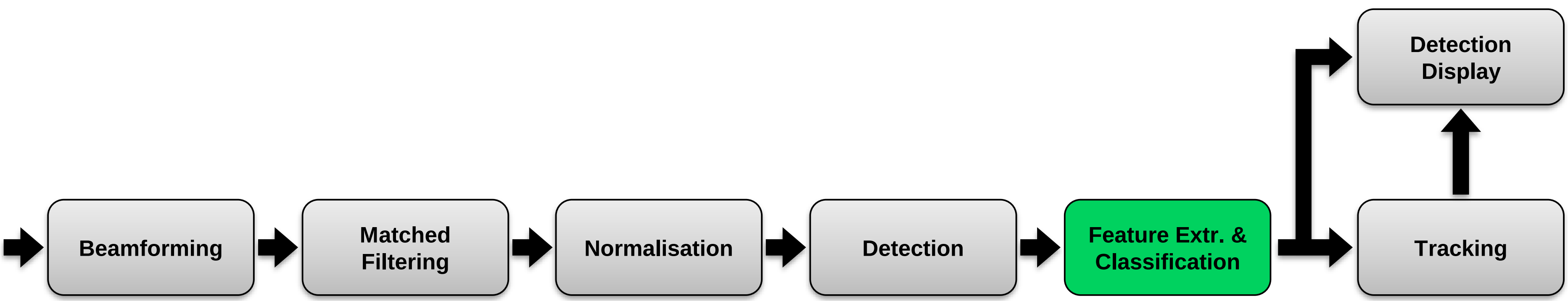
PROPOSED SOLUTION FOR FALSE ALARM REDUCTION

Modification of the Signal Processing

- Standard Active Signal Processing Chain:



- Modified Active Signal Processing Chain for False Alarm Reduction:



FEATURE EXTRACTION AND CLASSIFICATION

Feature Extraction and Classification

- Two different machine learning techniques are considered:

1. **Classical Machine Learning:**

   → Machine Learning based on hand-crafted extracted features.

2. Convolutional Neural Networks:

   → Machine Learning techniques that automatically extract features for input signals/images. No feature engineering required.

Classification with Feed Forward Neural Network (FNN)

Inputs:

Feature Vector for Contact n:

$$\mathbf{x}_n \in \mathbb{R}^{53\times 1}$$

One Hidden Layer:

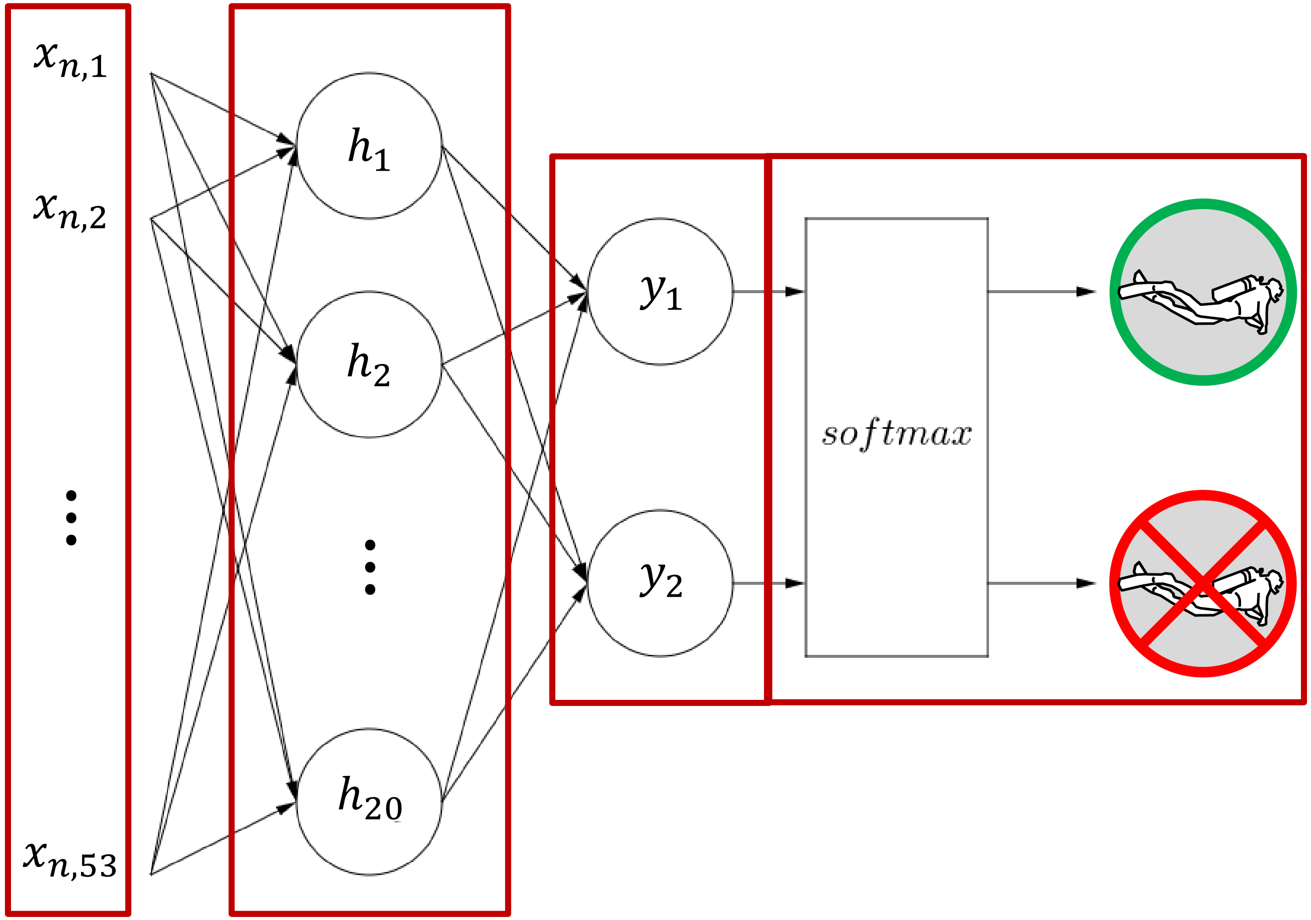
20 Neurons
Activation: hyperbolic tangent

Output Layer:

Binary Classification
→ 2 Neurons

Softmax Function:

Probability for belonging to class
→ Diver Contact
→ False Alarm



Feature Extraction and Classification

- Two different machine learning techniques are considered:

1. Classical Machine Learning:

   → Machine Learning based on hand-crafted extracted features.

2. **Convolutional Neural Networks:**

   → **Machine Learning techniques that automatically extract features for input signals/images. No feature engineering required.**

Two different types of Networks are considered

1. **Shallow Convolutional Neural Network trained from scratch.**

2. Pre-trained deep networks that are originally trained for distinguishing objects in R-G-B images.

Convolutional Neural Networks
Structure of Shallow CNN trained from scratch

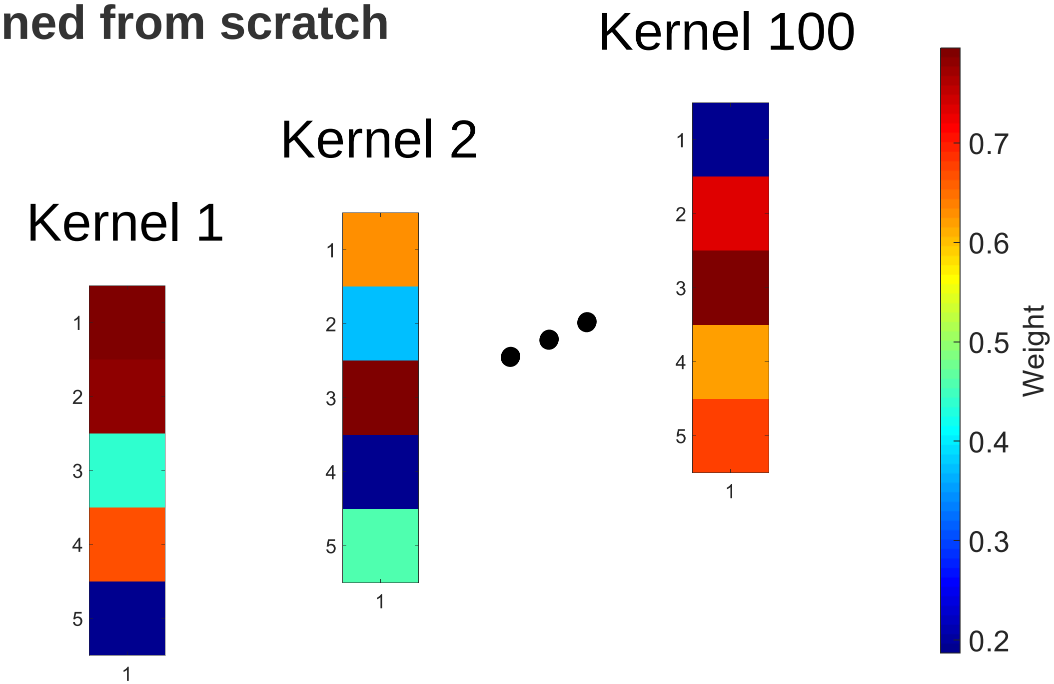
$142 \times 11 \times 1$

$1 \times 5 \times 1;\ 100$

$2 \times 1;\ 2:1$

$4096;\ 2$

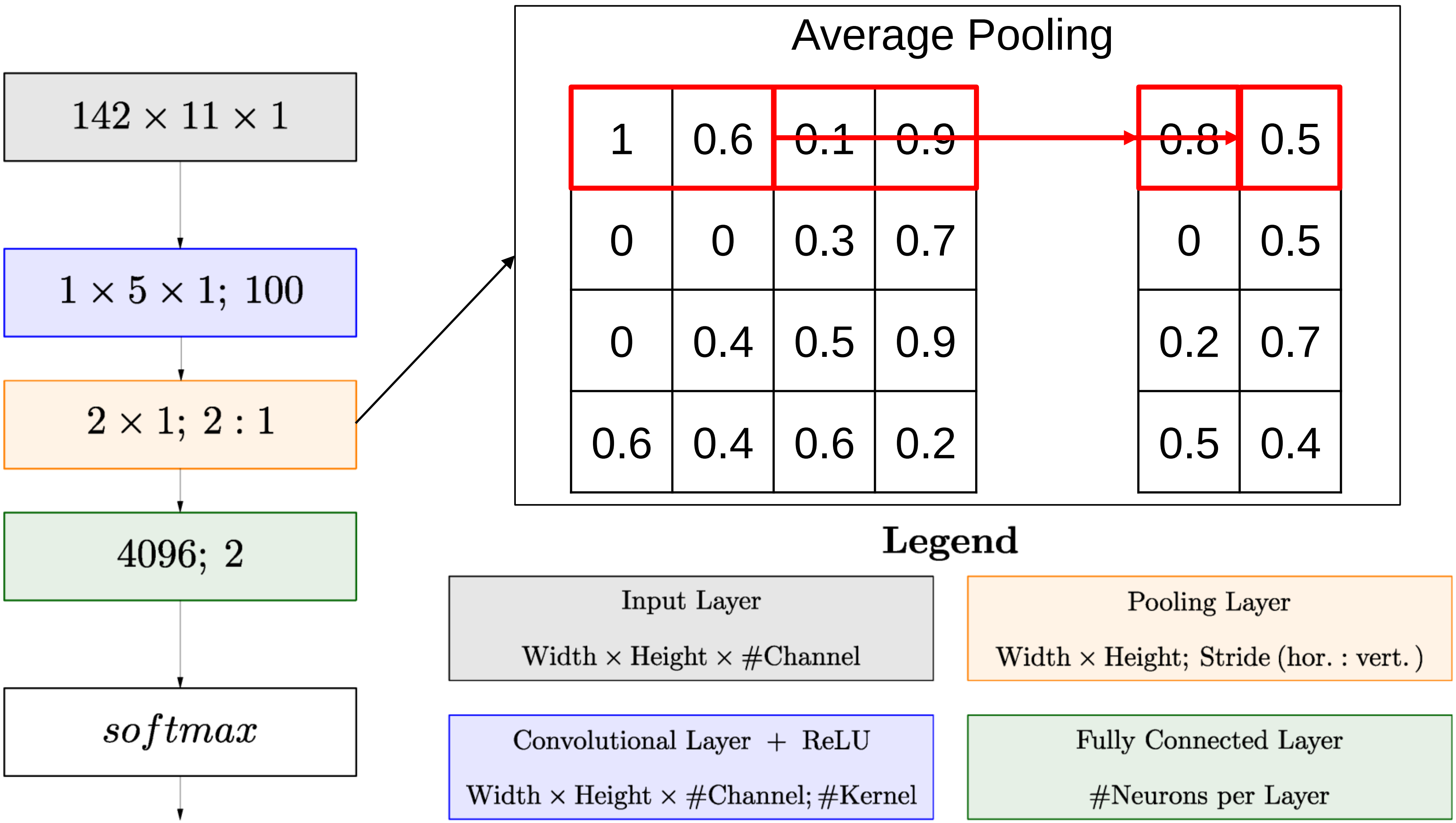
$softmax$



Legend

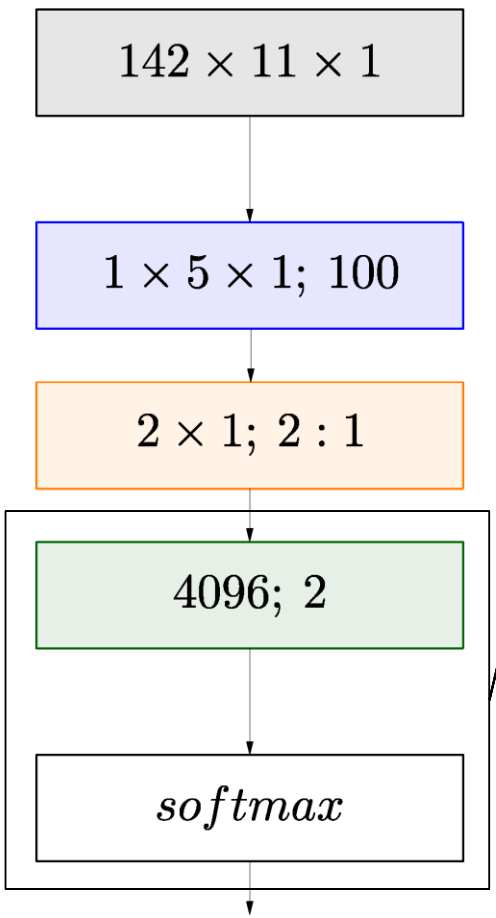
Input Layer Width $\times$ Height $\times$ #Channel	Pooling Layer Width $\times$ Height; Stride (hor. : vert.)
Convolutional Layer + ReLU Width $\times$ Height $\times$ #Channel; #Kernel	Fully Connected Layer #Neurons per Layer

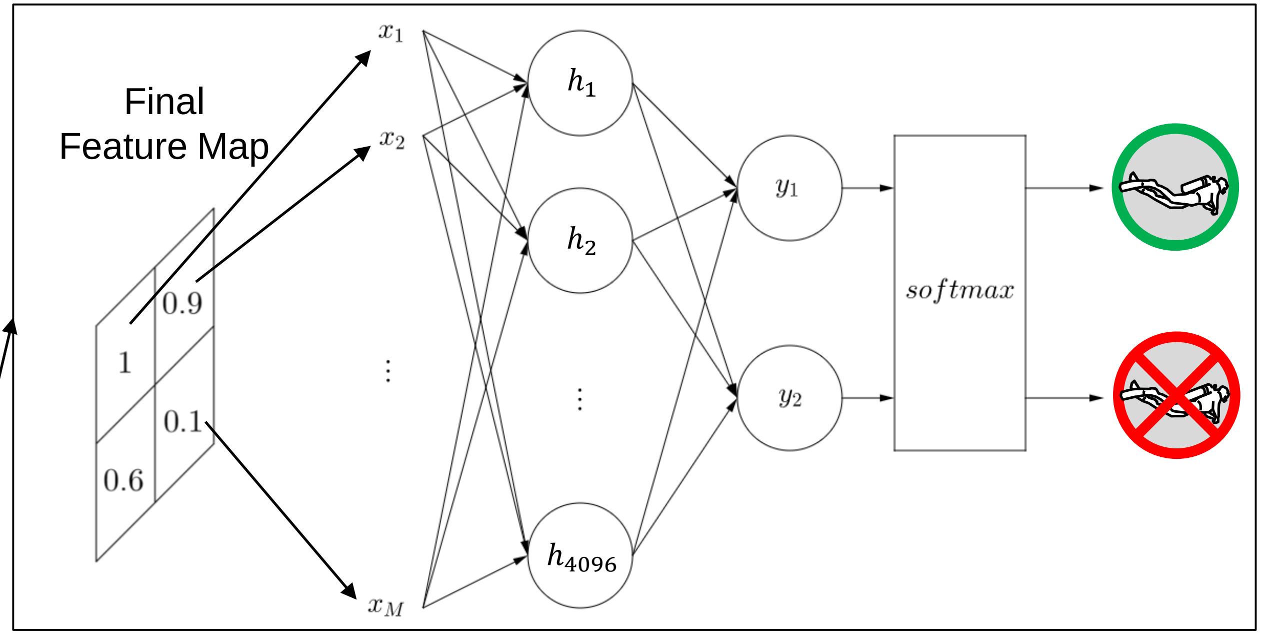
Convolutional Neural Networks
Structure of Shallow CNN trained from scratch



ATLAS ELEKTRONIK | WTD 71 | HSB Hochschule Bremen City University of Applied Sciences | BERGISCHE UNIVERSITÄT WUPPERTAL

Convolutional Neural Networks
Structure of Shallow CNN trained from scratch





Legend

Input Layer	Width × Height × #Channel
Pooling Layer	Width × Height; Stride (hor. : vert.)
Convolutional Layer + ReLU	Width × Height × #Channel; #Kernel
Fully Connected Layer	#Neurons per Layer

Two different types of Networks are considered

1. Shallow Convolutional Neural Network trained from scratch.

2. **Pre-trained deep networks that are originally trained for classifying objects in R-G-B images.**

Convolutional Neural Networks
Transfer Learning of pre-trained Deep Networks

- Many different pre-trained Networks are available in MATLAB / Python / etc.

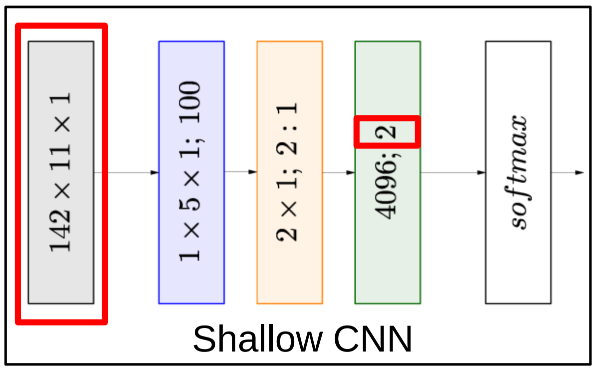
- These are originally trained for distinguishing 1000 different objects in R-G-B images.

Reference: Krizhevsky, A. et al; ImageNet Classification with Deep Convolutional Neural Networks

- Nine networks that are firstly introduced in the ImageNet Large Scale Visual Recognition Challenges are considered:
  - AlexNet (5 Convolutional Layers)
  - GoogLeNet (57 Convolutional Layers)
  - Inception v3 (94 Convolutional Layers)
  - ResNet-18, ResNet-50 and ResNet-101 (20, 53 and 104 Convolutional Layers)
  - SqueezeNet (26 Convolutional Layers)
  - VGG-16 and VGG-19 (13 and 16 Convolutional Layers)

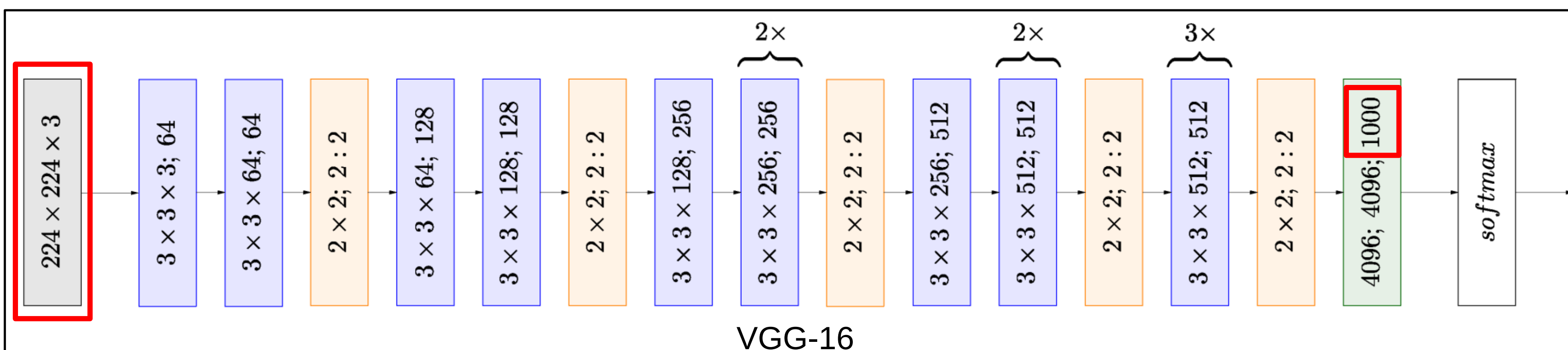
Convolutional Neural Networks
Transfer Learning of pre-trained Deep Networks

- Comparison of Shallow CNN and VGG-16.



Legend

Input Layer Width × Height × #Channel	Pooling Layer Width × Height; Stride (hor. : vert.)
Convolutional Layer + ReLU Width × Height × #Channel; #Kernel	Fully Connected Layer #Neurons per Layer

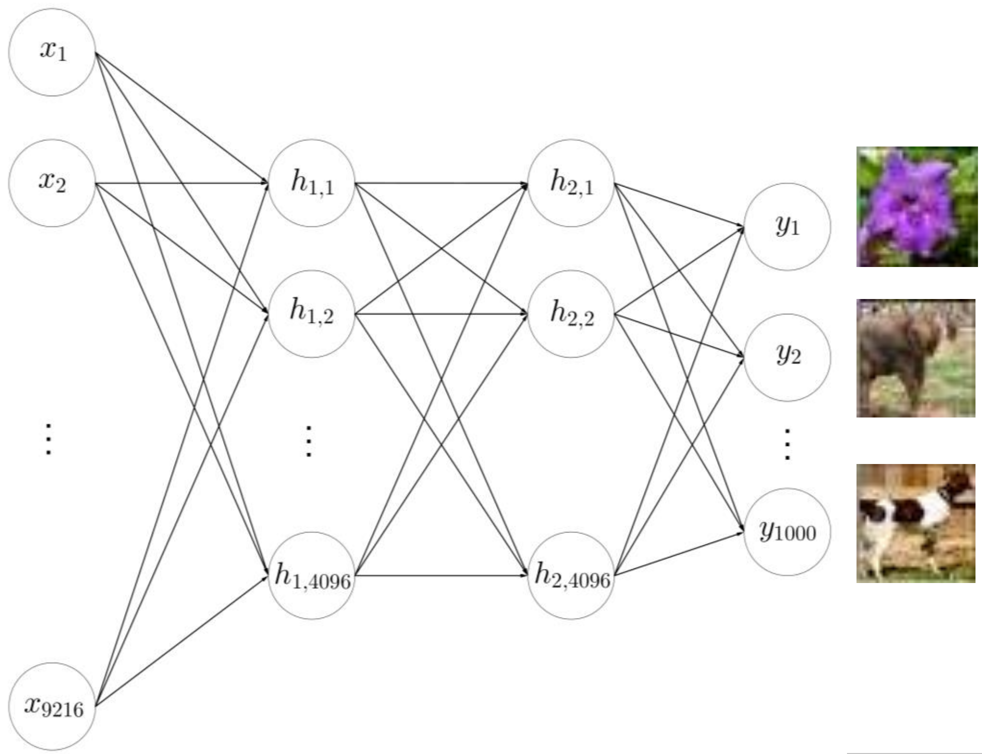


Convolutional Neural Networks
Transfer Learning of pre-trained Deep Networks

- Two steps are required for transfer learning:

1. Resample input images from 142×11×1 → 224×224×3 for GoogLeNet, ResNet, VGG
   227×227×3 for AlexNet, SqueezeNet
   299×299×3 for Inception v3
2. Replace Output Layer of Fully Connected Layer.

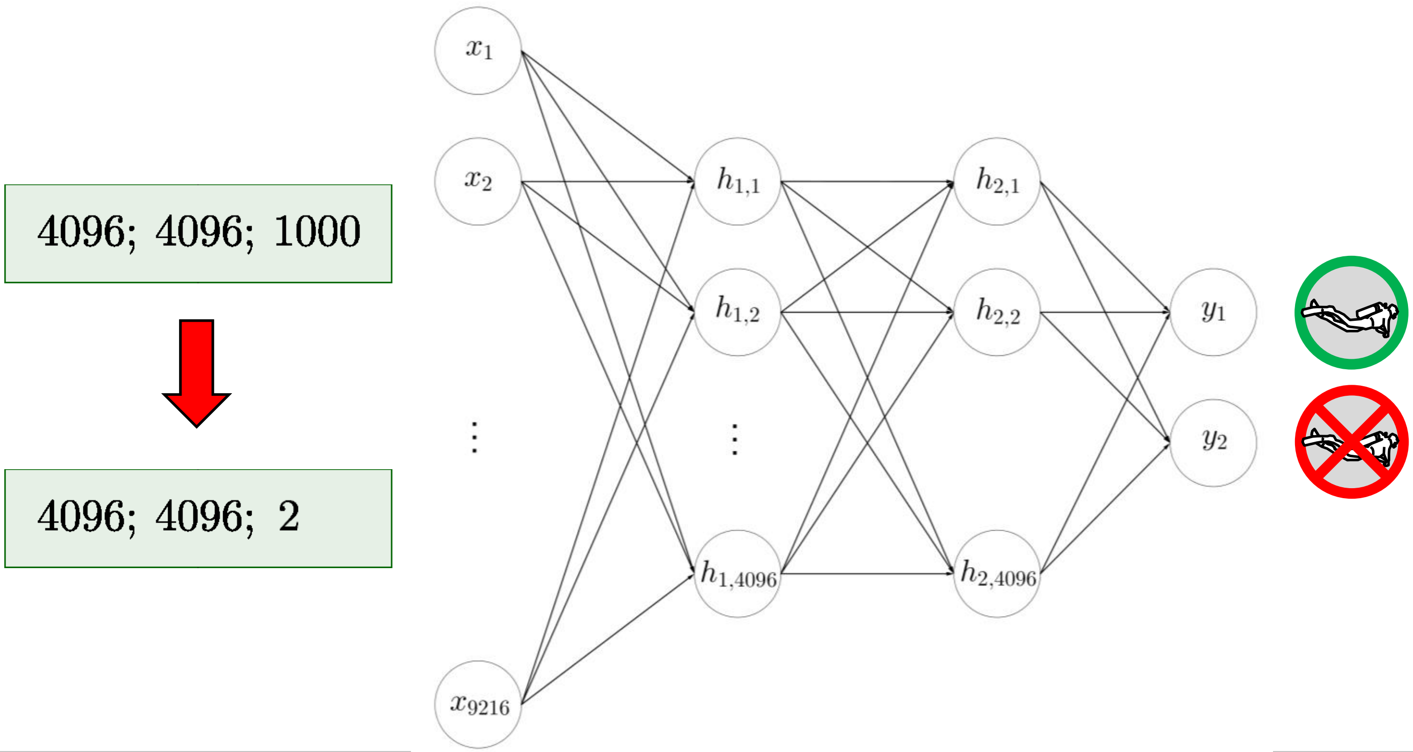
$4096;\ 4096;\ 1000$



Convolutional Neural Networks
Transfer Learning of pre-trained Deep Networks

- Two steps are required for transfer learning:

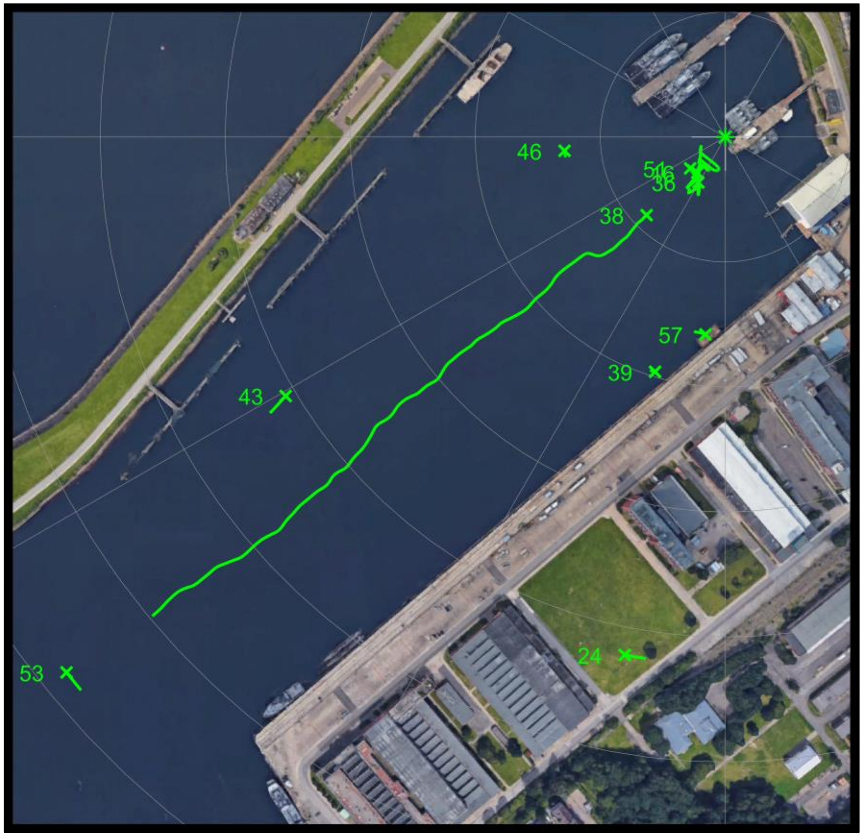
1. Resample input images from 142×11×1 → 224×224×3 for GoogLeNet, ResNet, VGG; 227×227×3 for AlexNet, SqueezeNet; 299×299×3 for Inception v3
2. Replace Output Layer of Fully Connected Layer.



DATA LABELLING

Data Labelling

- Contacts belonging to Track of the diver are labelled as “Diver Contact”.

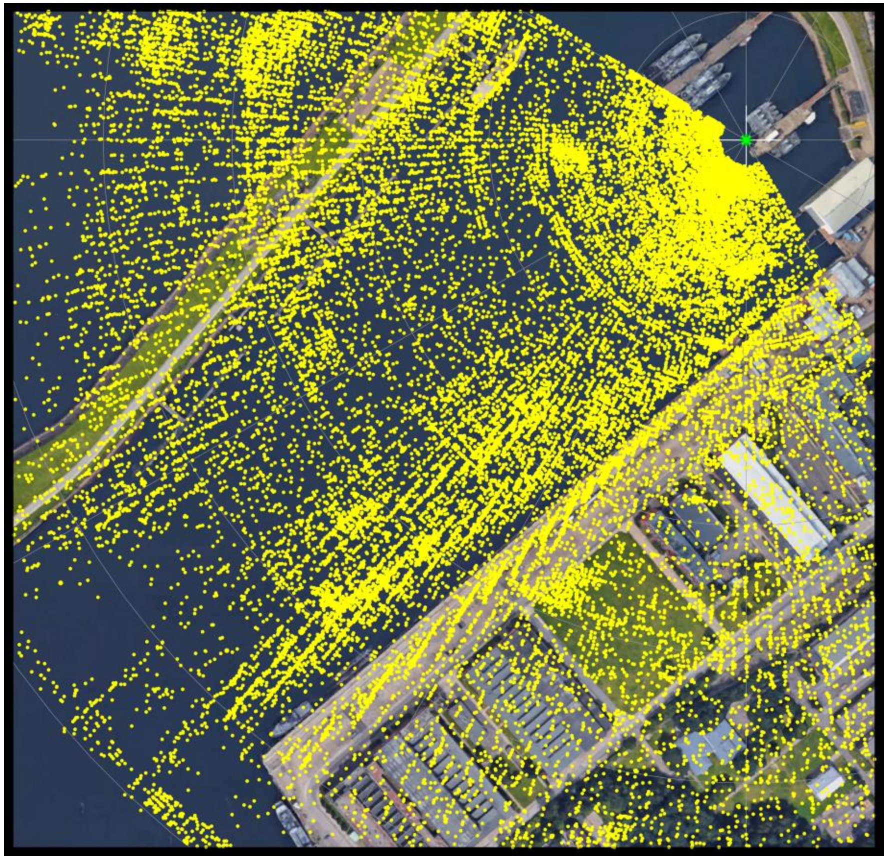


Tracking Results

Positions of Diver Contacts

Data Labelling

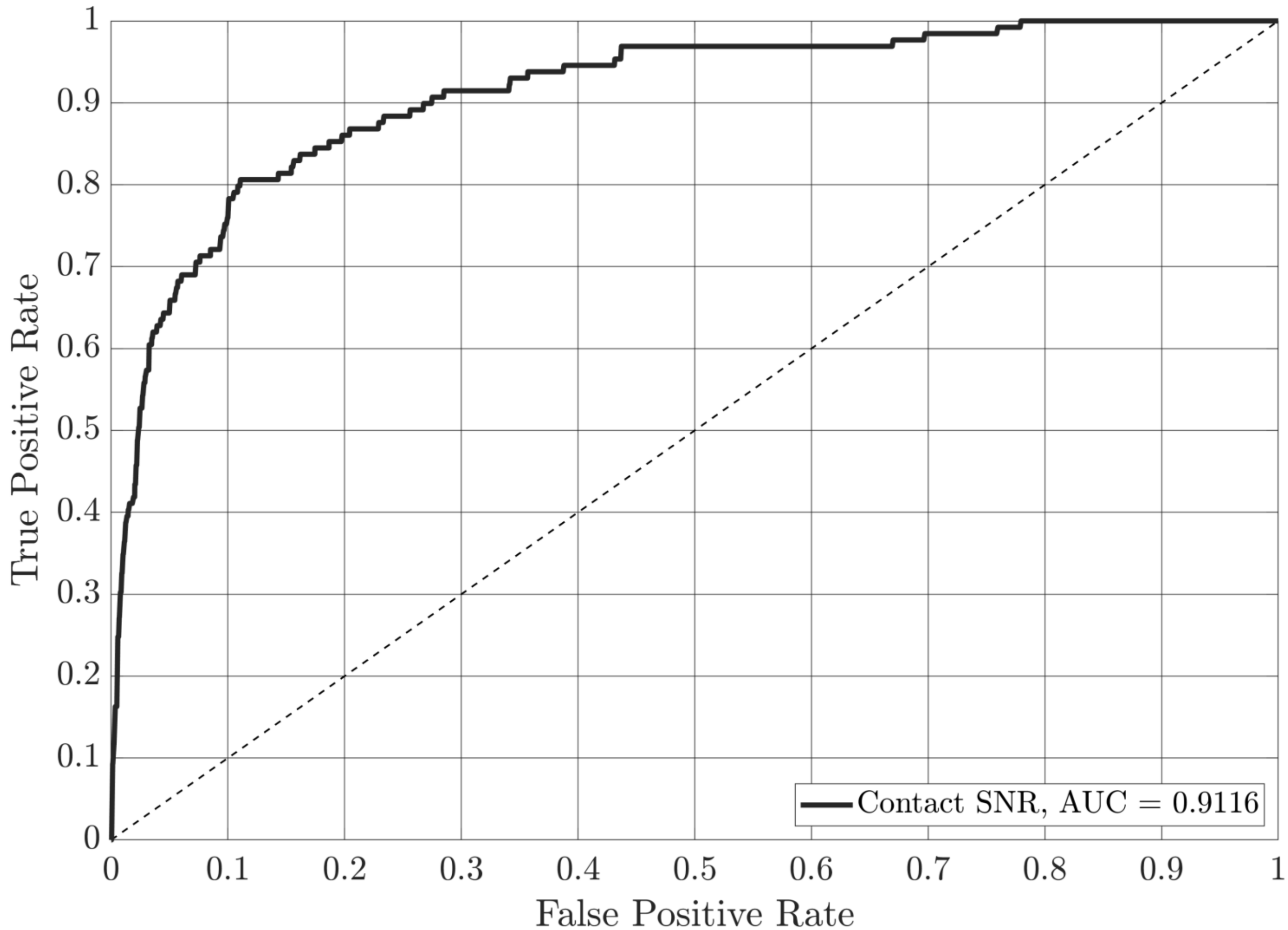
- All reamaining contacts are labelled as “False Alarm”.



Positions of False Alarms

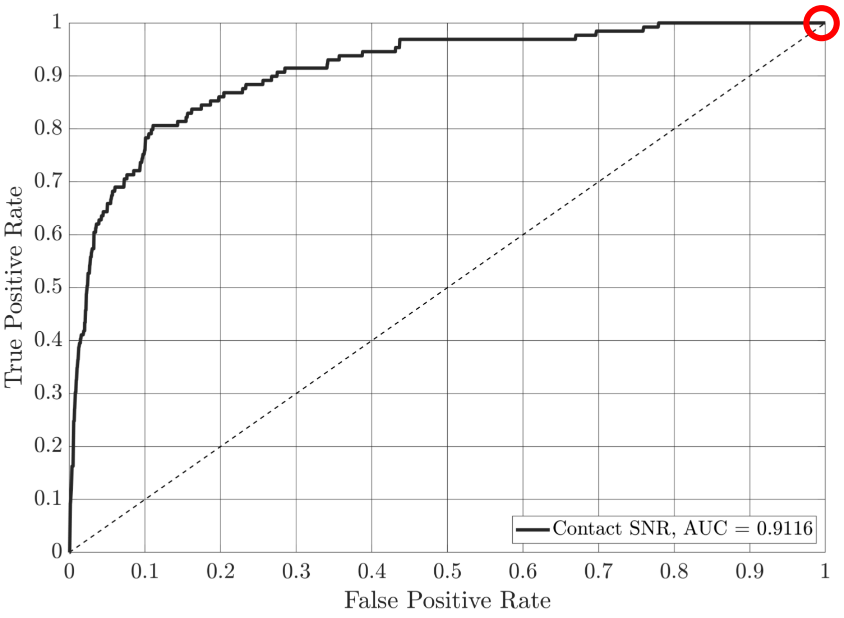
PERFORMANCE CRITERION

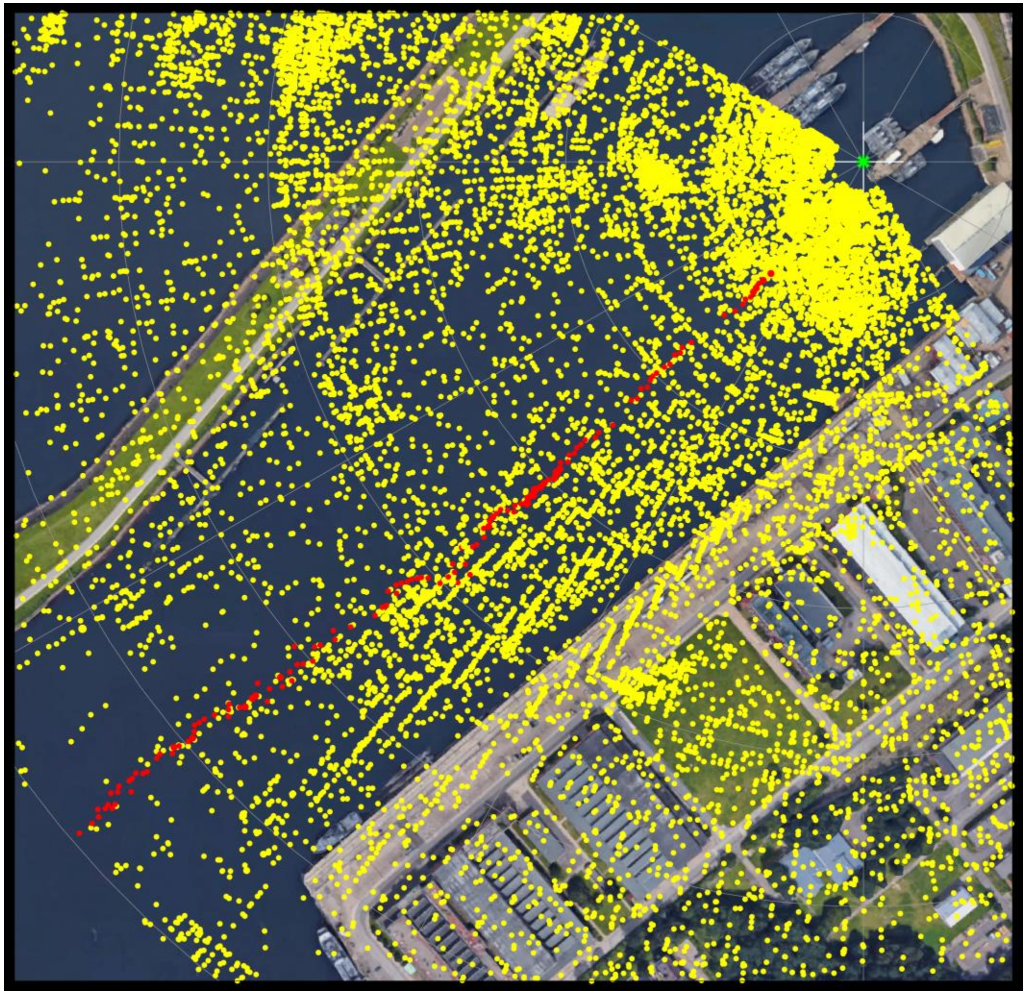
Performance Criterion
Receiver-Operating-Characteristic (ROC) Curves



Performance Criterion
Receiver-Operating-Characteristic (ROC) Curves

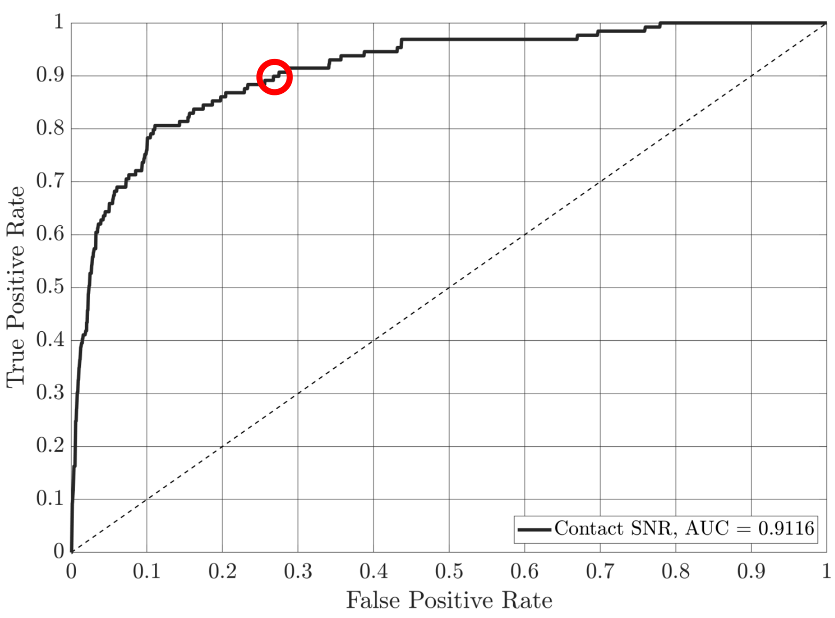
TPR = 1.00

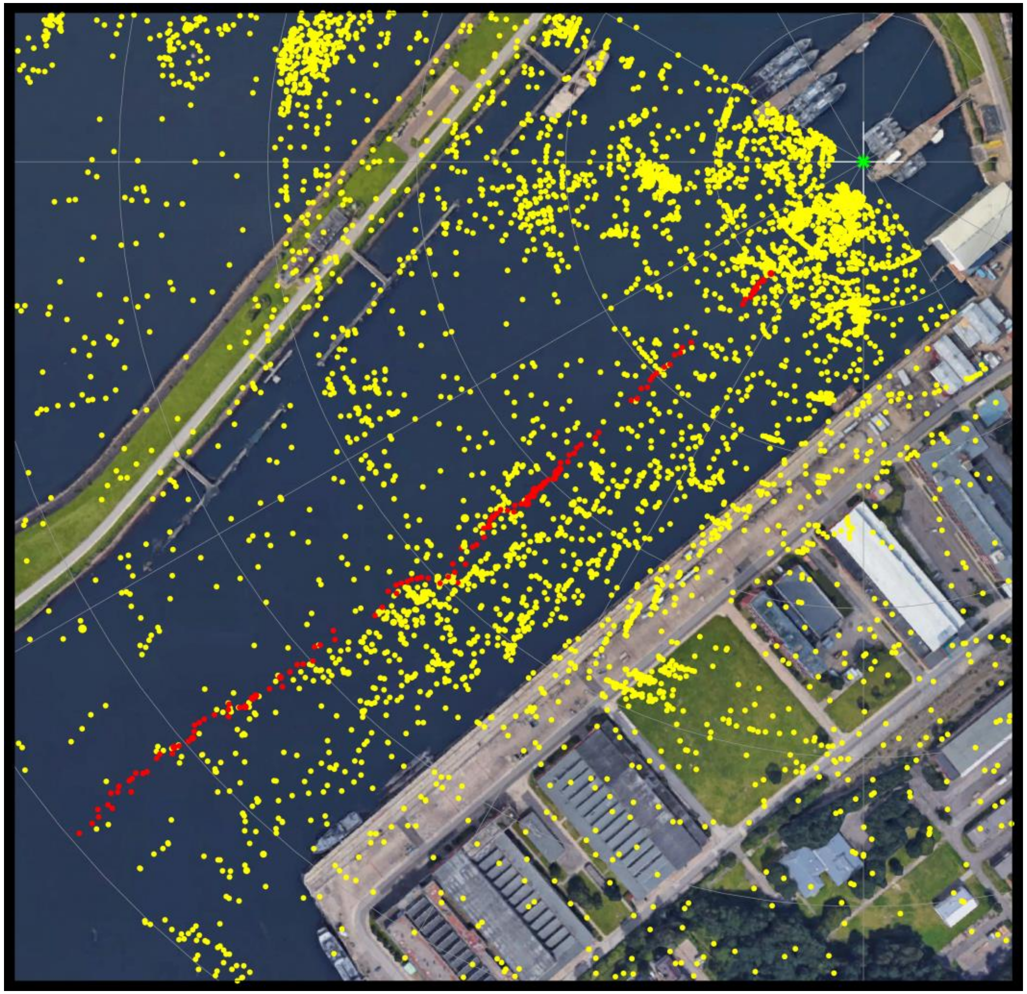




Performance Criterion
Receiver-Operating-Characteristic (ROC) Curves

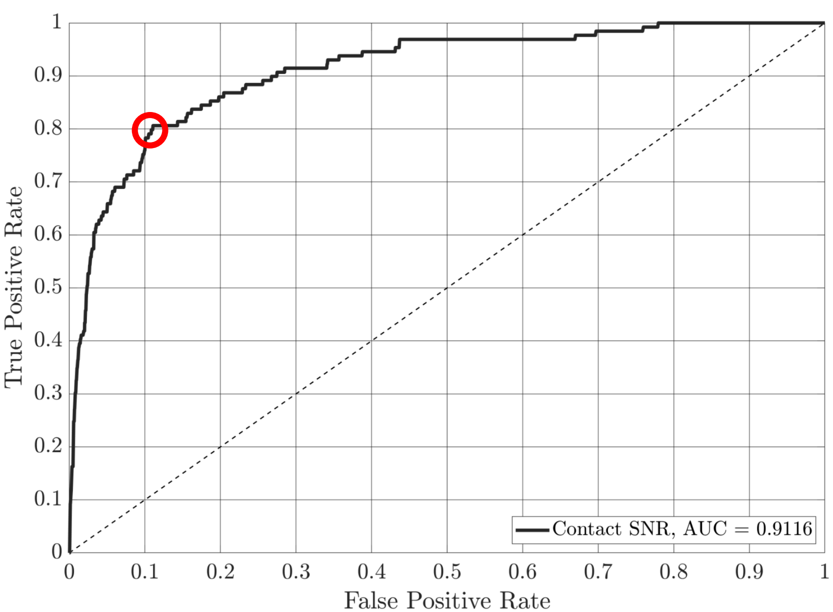
TPR = 0.90

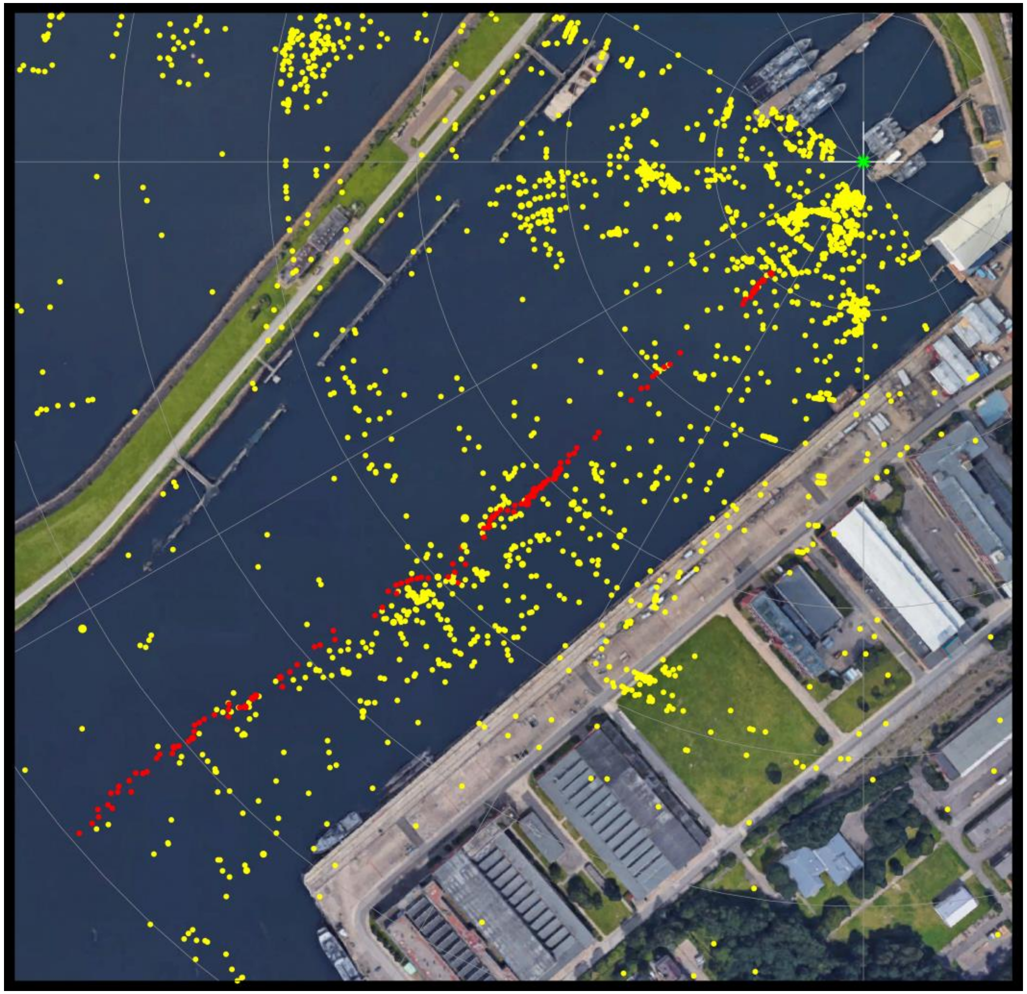




Performance Criterion
Receiver-Operating-Characteristic (ROC) Curves

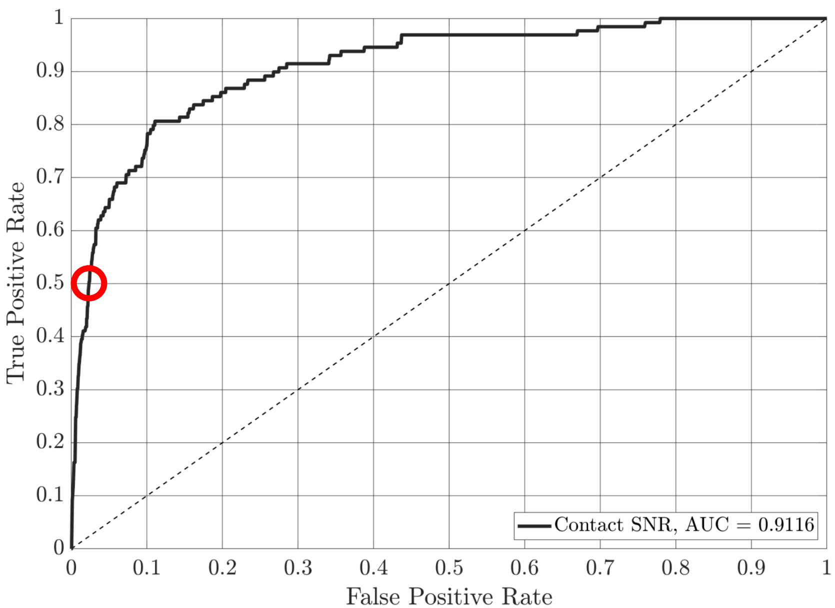
TPR = 0.80

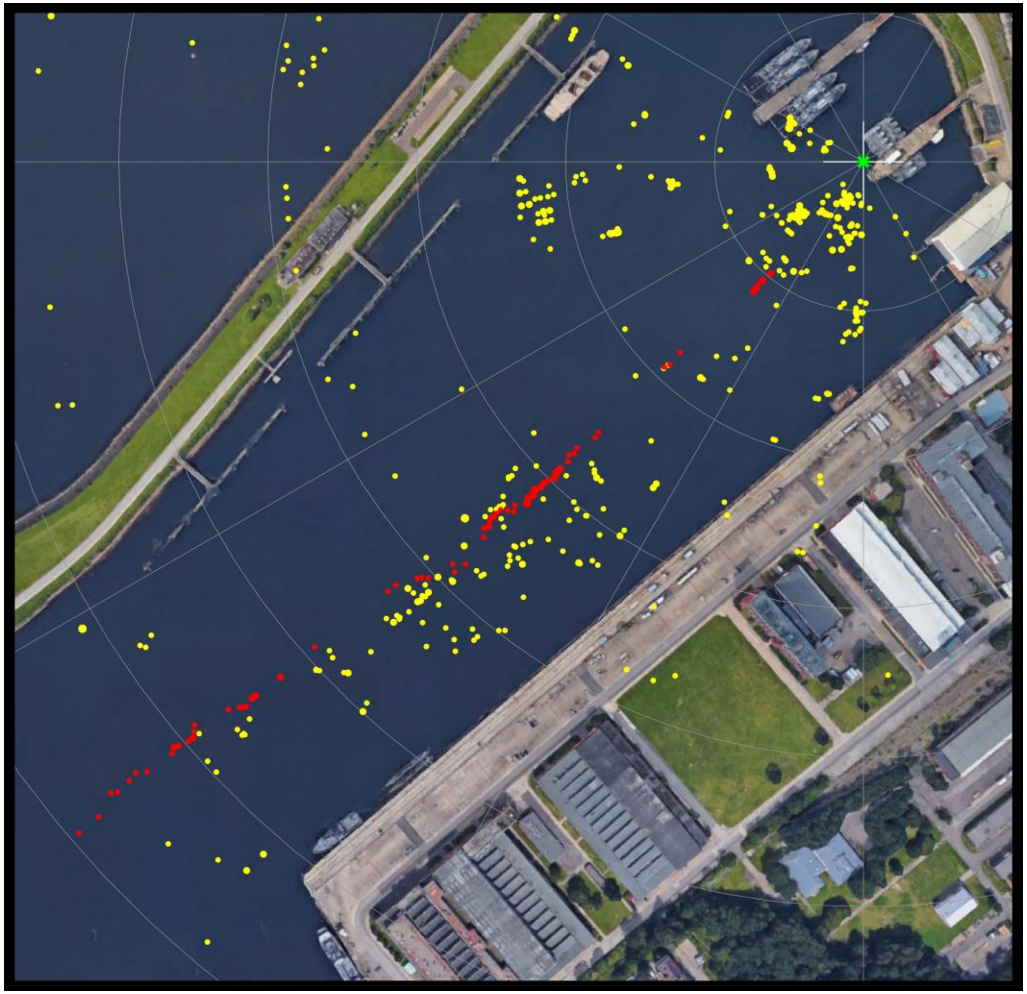




Performance Criterion
Receiver-Operating-Characteristic (ROC) Curves

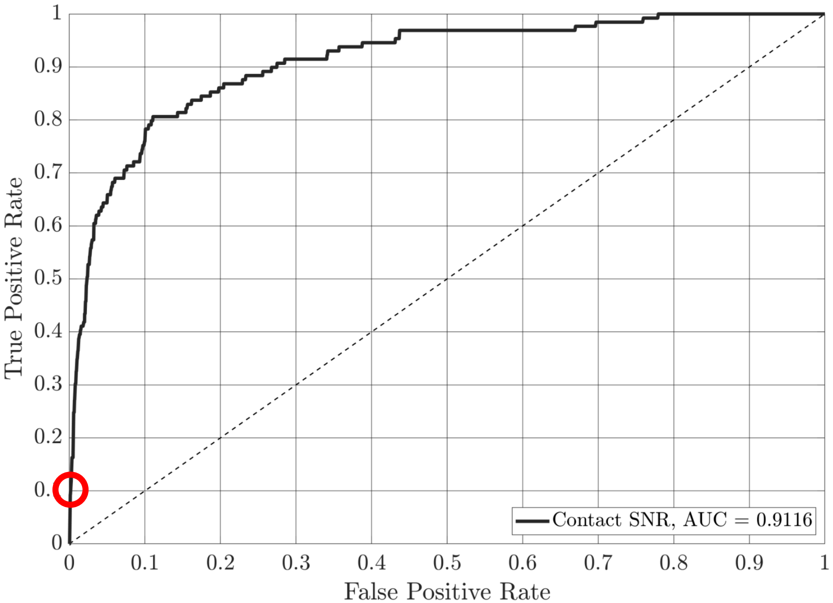
TPR = 0.50

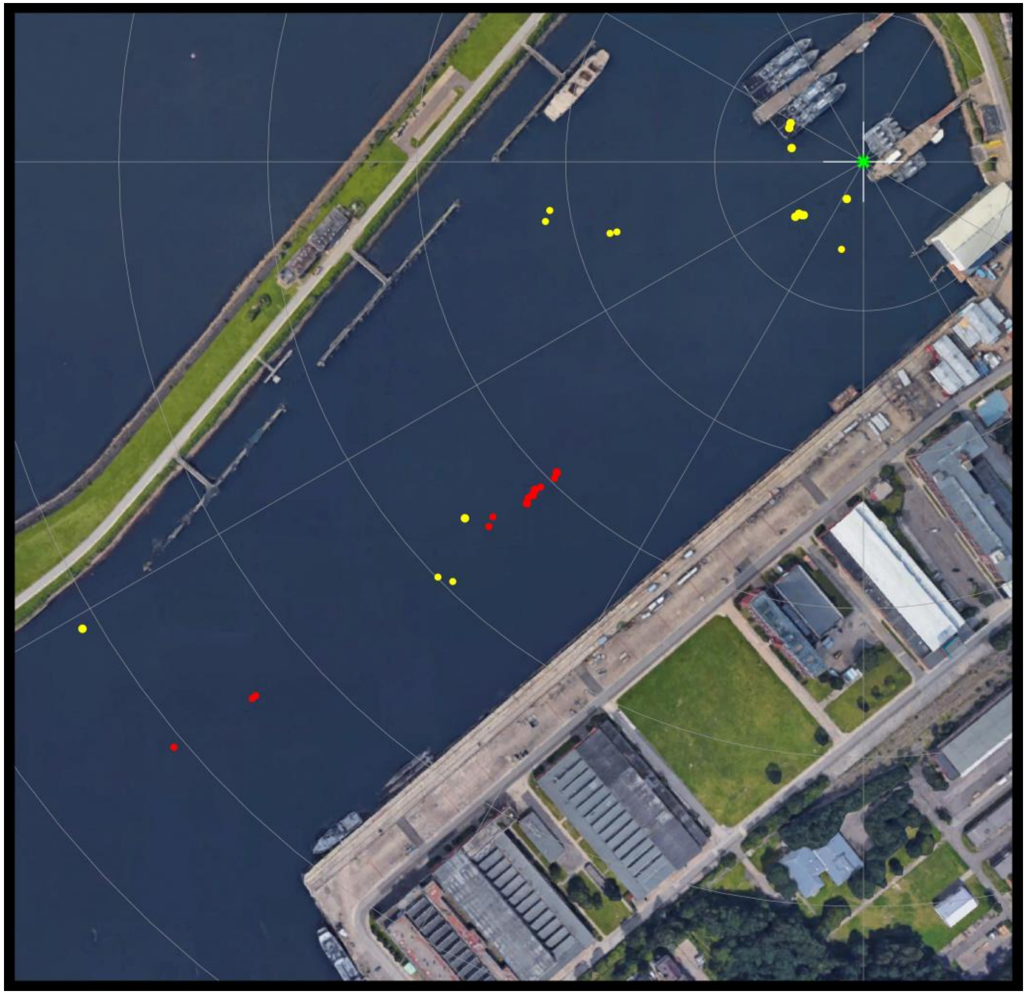




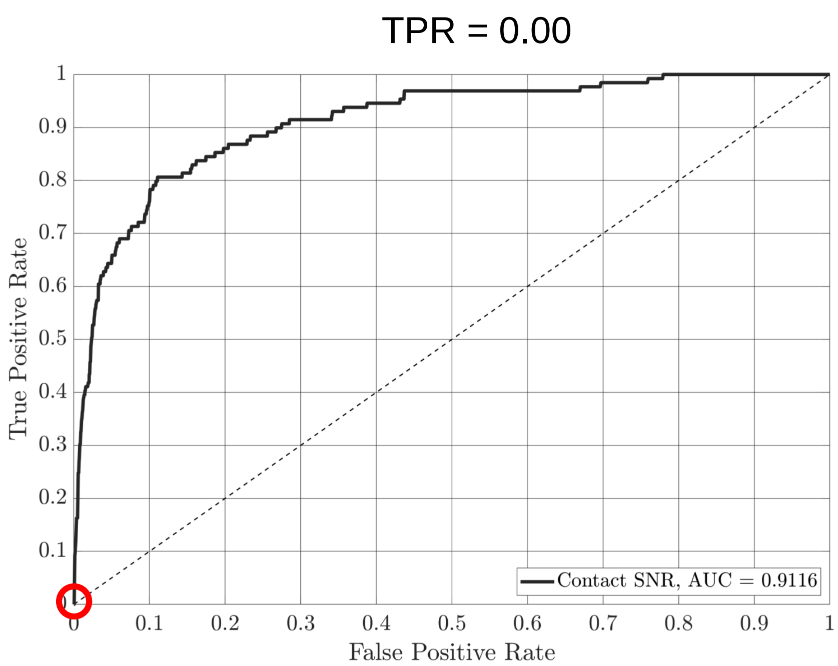
Performance Criterion
Receiver-Operating-Characteristic (ROC) Curves

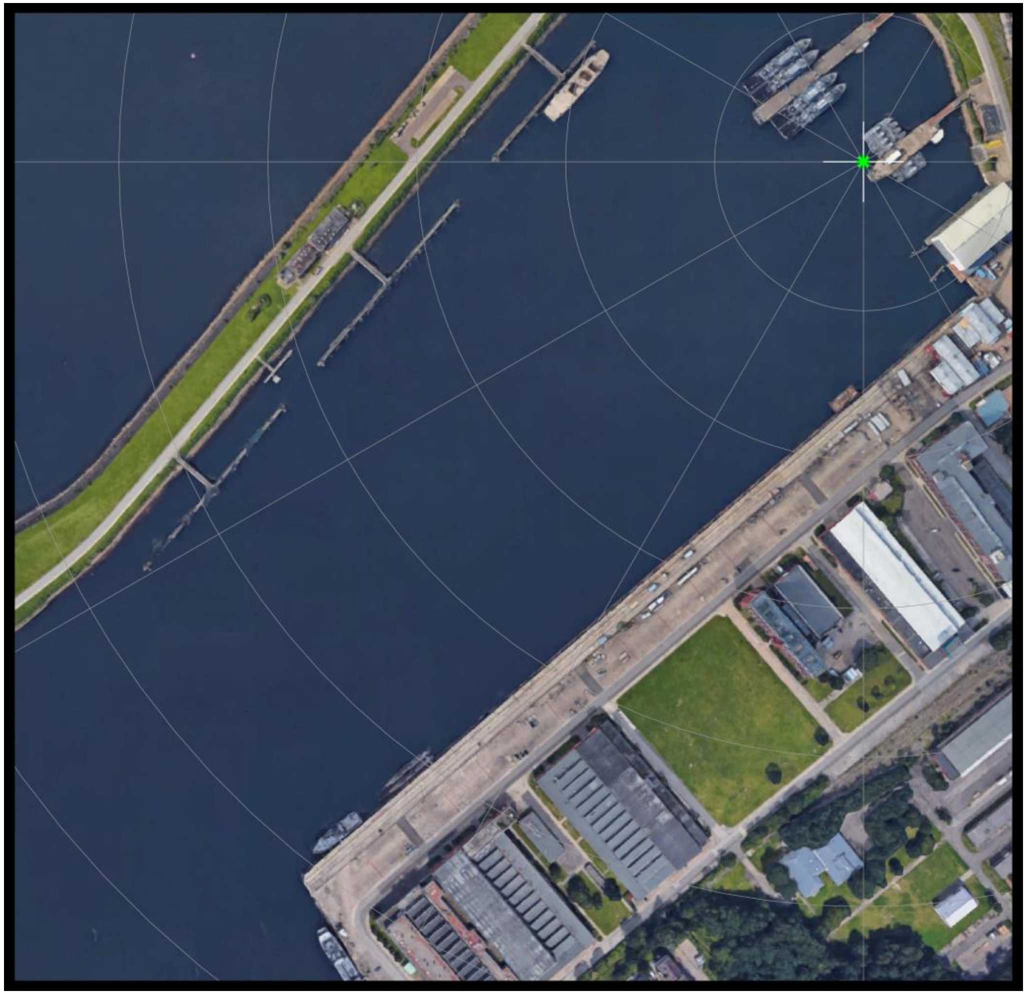
TPR = 0.10





Performance Criterion
Receiver-Operating-Characteristic (ROC) Curves

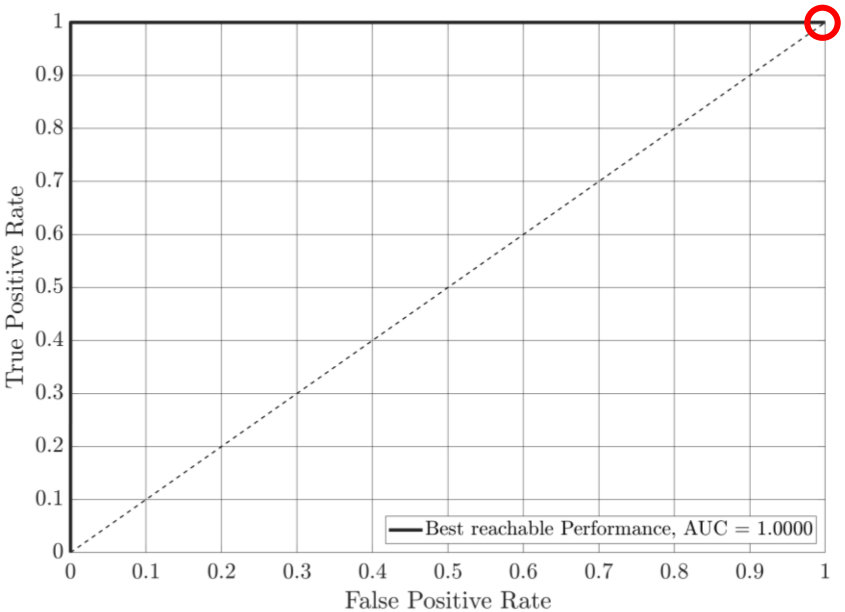


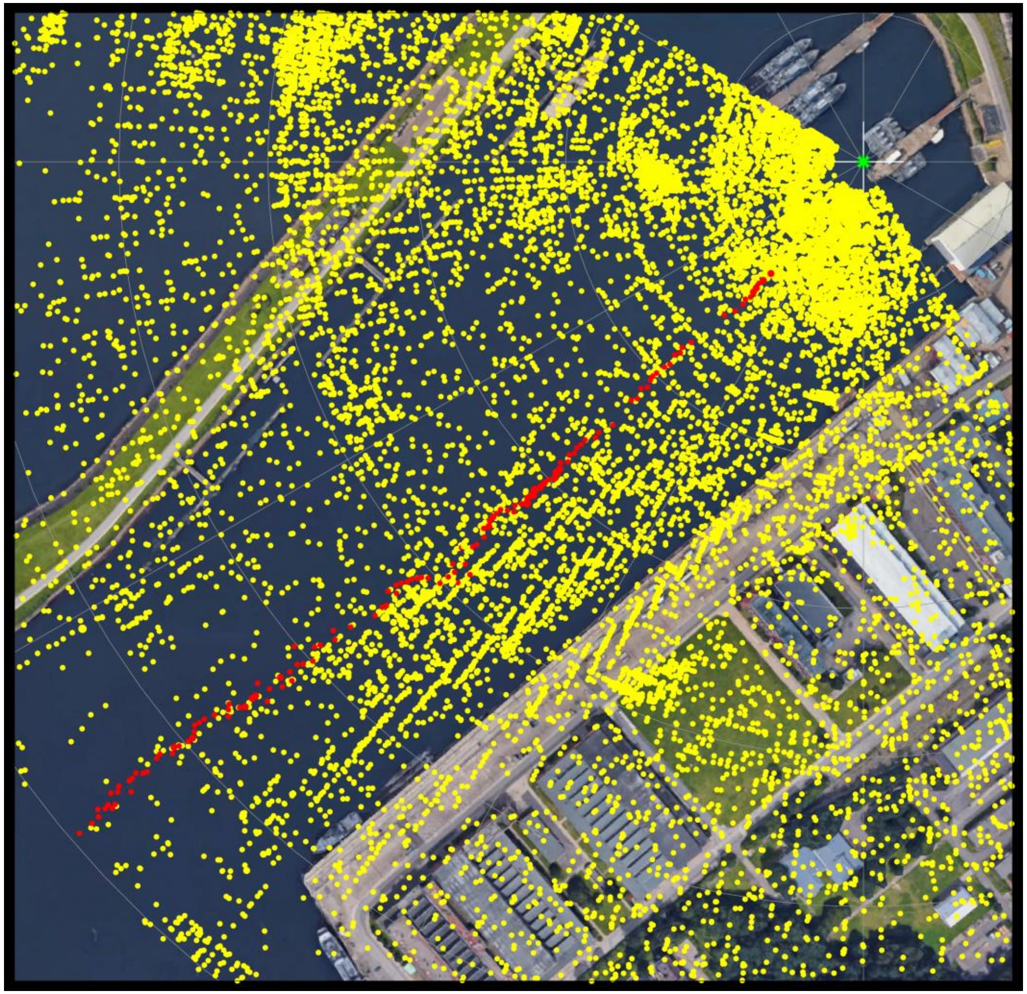


IDEAL CASE

Ideal ROC Curve

TPR = 1.00, FPR = 1.00



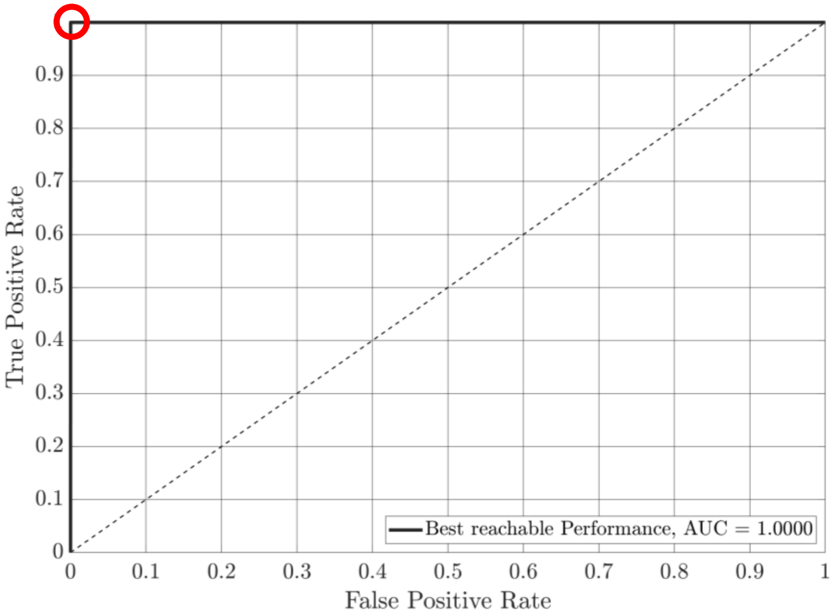


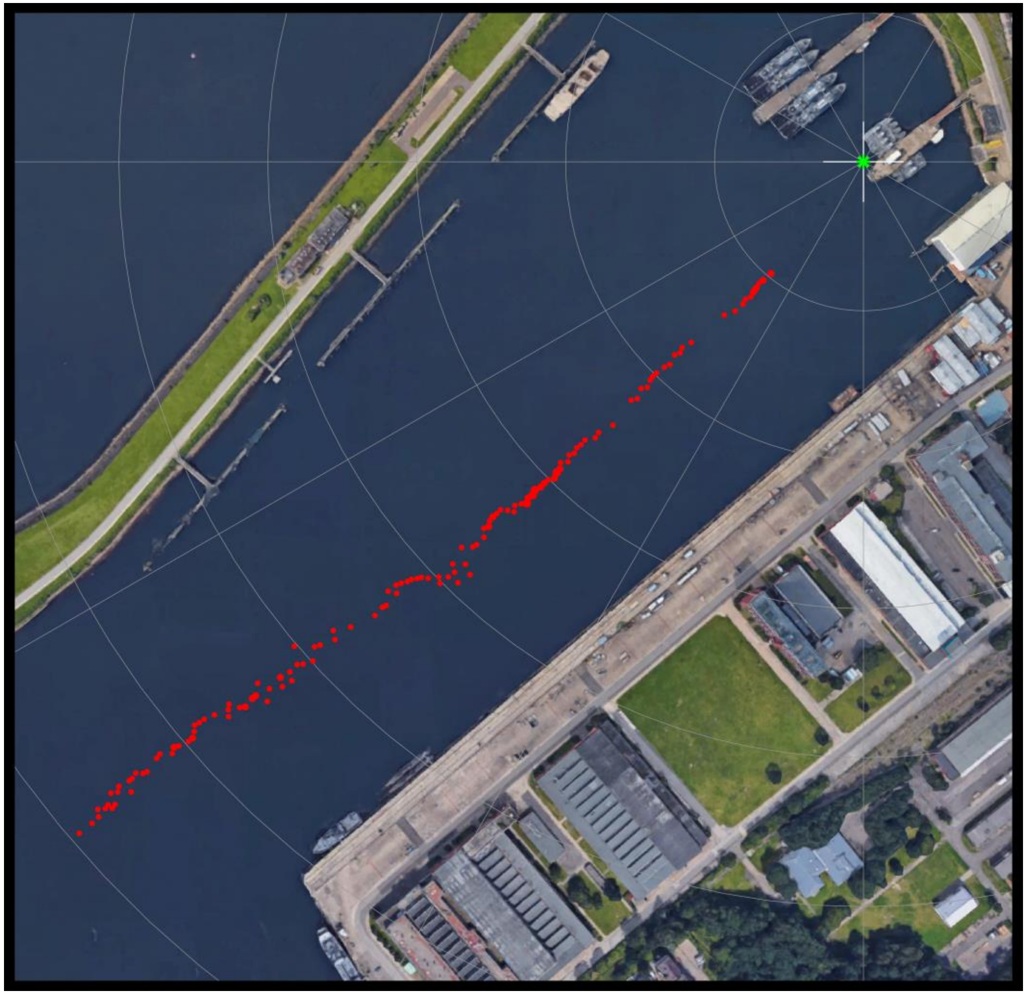
Ideal ROC Curve

- All Diver Contacts and
  No False Alarms Remain.

- **Ideal Case!**
- **Almost impossible to achieve!**

TPR = 1.00, FPR = 0.00





CLASSIFICATION RESULTS

Considered Datasets

- Three datasets recorded in different environments are merged to a big training dataset.

	$D_{\text{Train}}E_1$	$D_{\text{Train}}E_2$	$D_{\text{Train}}E_3$
Diver Contacts	255	136	320
False Alarms	21831	21141	3761

D_{Train}
711
46733

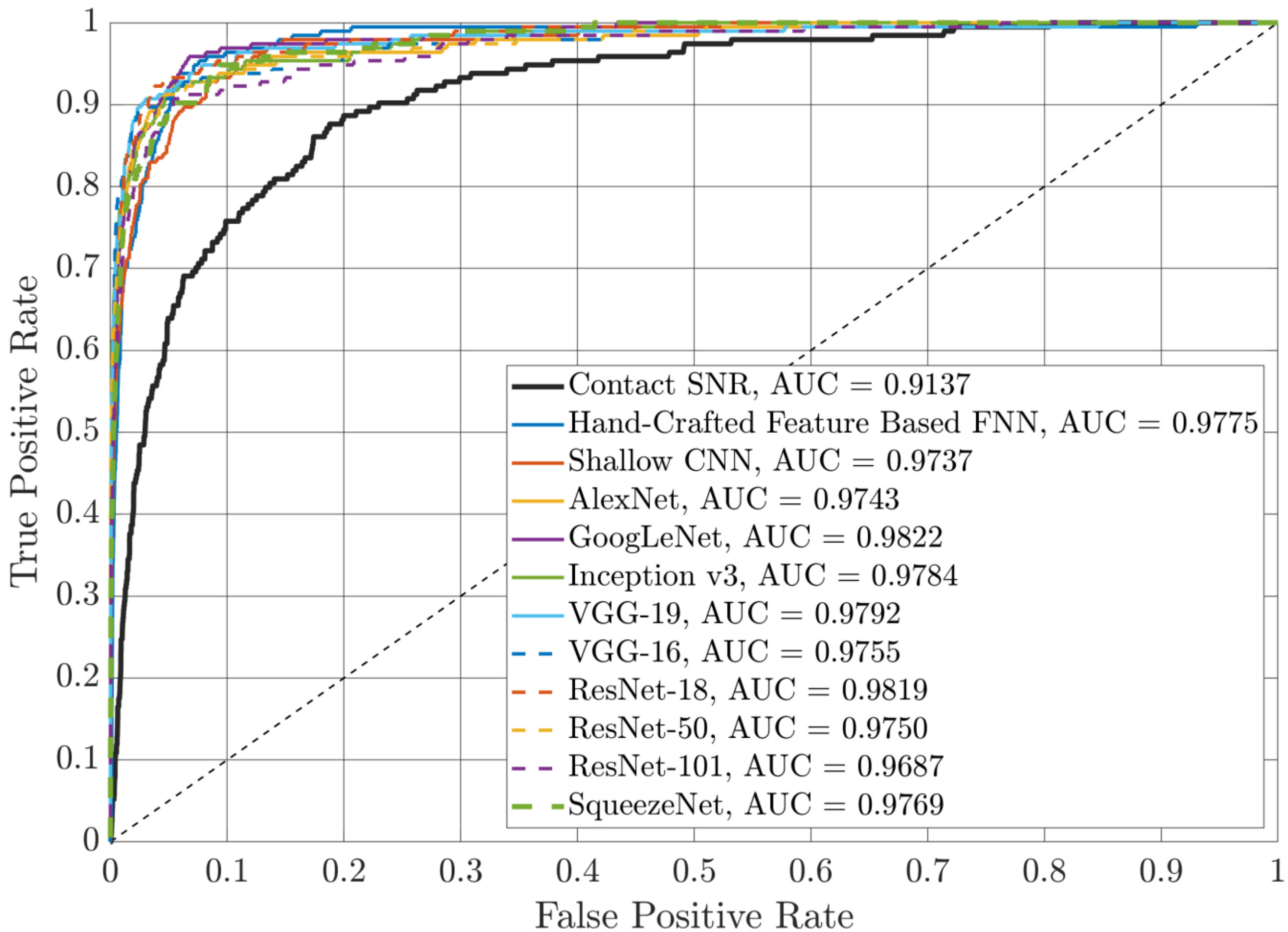
- Three similar datasets are used as test datasets.

	$D_{\text{Test}}E_1$	$D_{\text{Test}}E_2$	$D_{\text{Test}}E_3$
Diver Contacts	356	194	187
False Alarms	37843	22484	2484

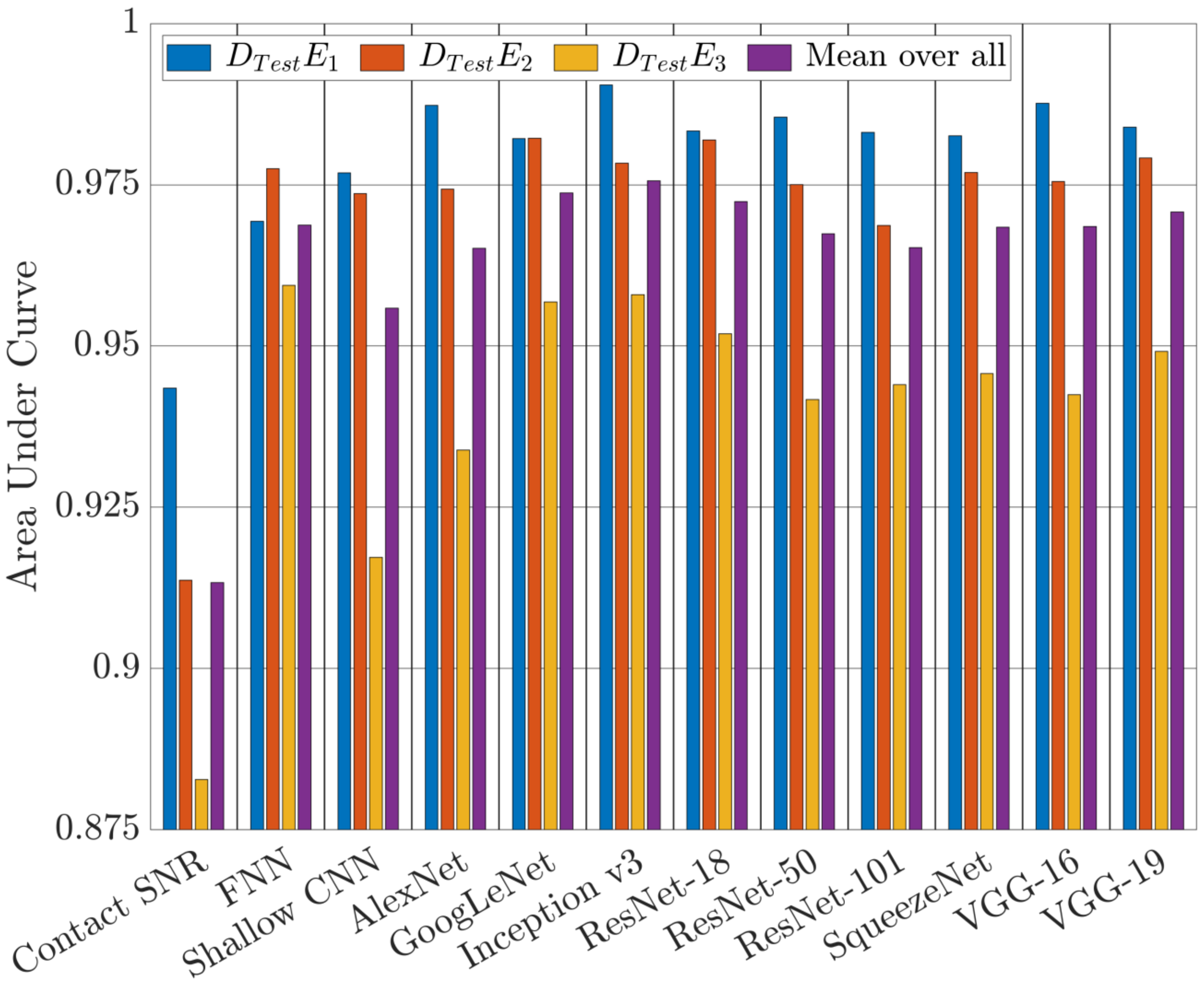
- **All Datasets are highly unbalanced!**

Classification Results
ROC Curves

- Algorithms tested with dataset $D_{\text{Test}}E_2$



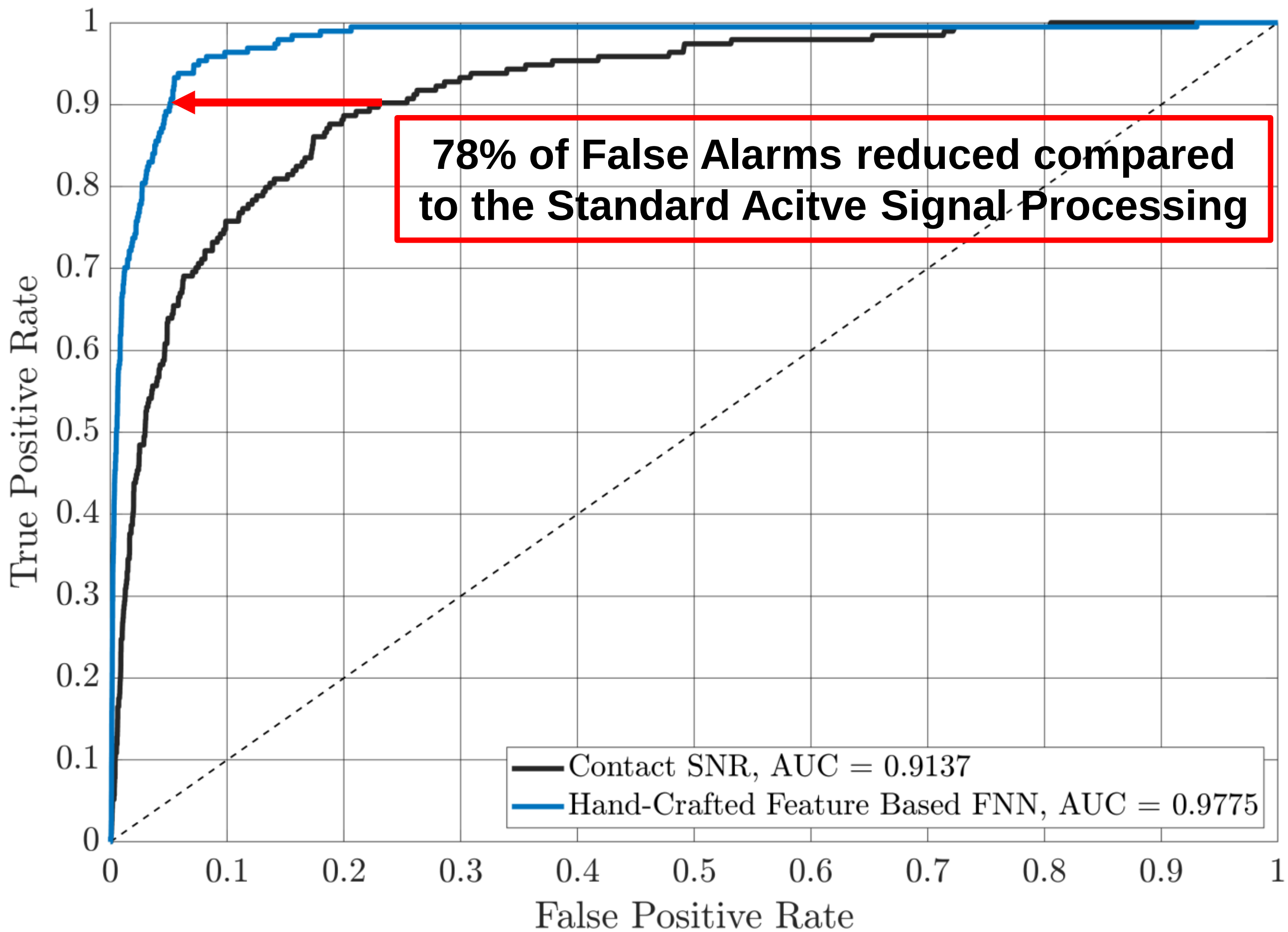
Classification Results
Performance for all Test Datasets



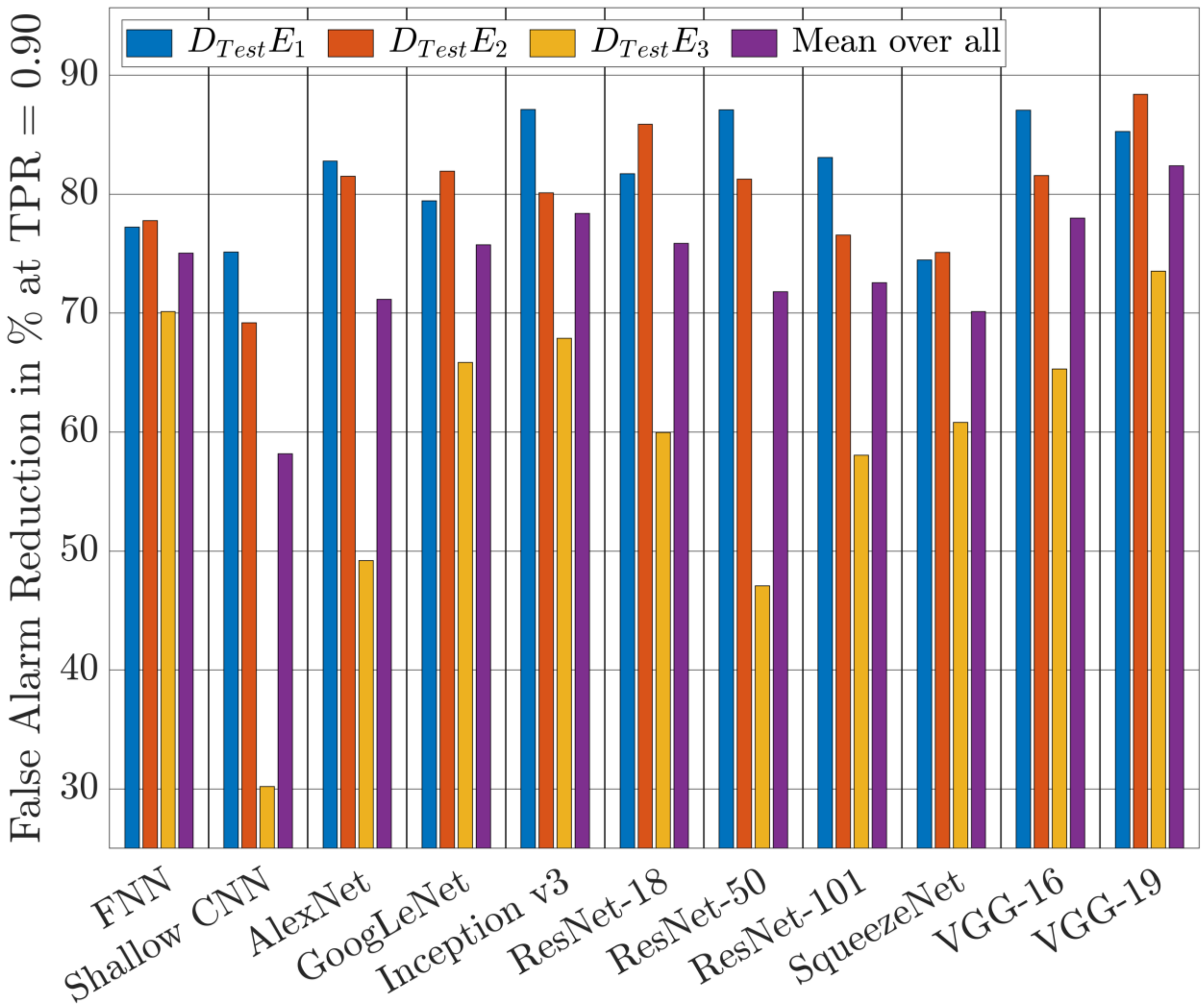
Classification Results

Performance Criteria for False Alarm Reduction

- ROC Curve for testing the FNN with dataset $D_{\text{Test}}E_2$



Classification Results
Performance Criteria for False Alarm Reduction

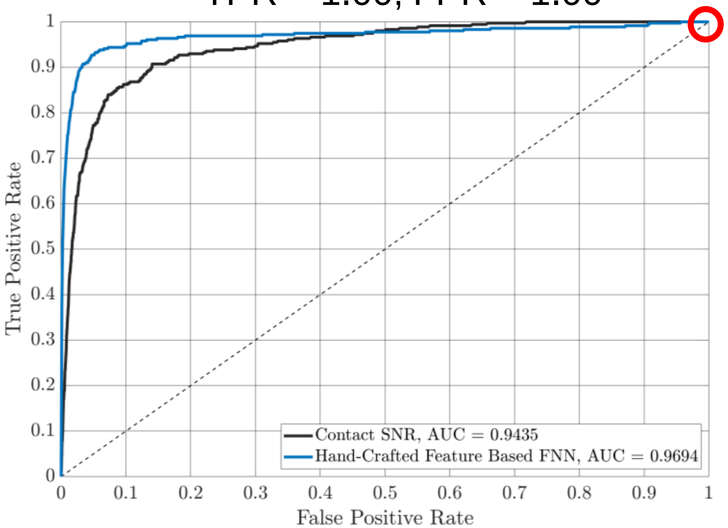


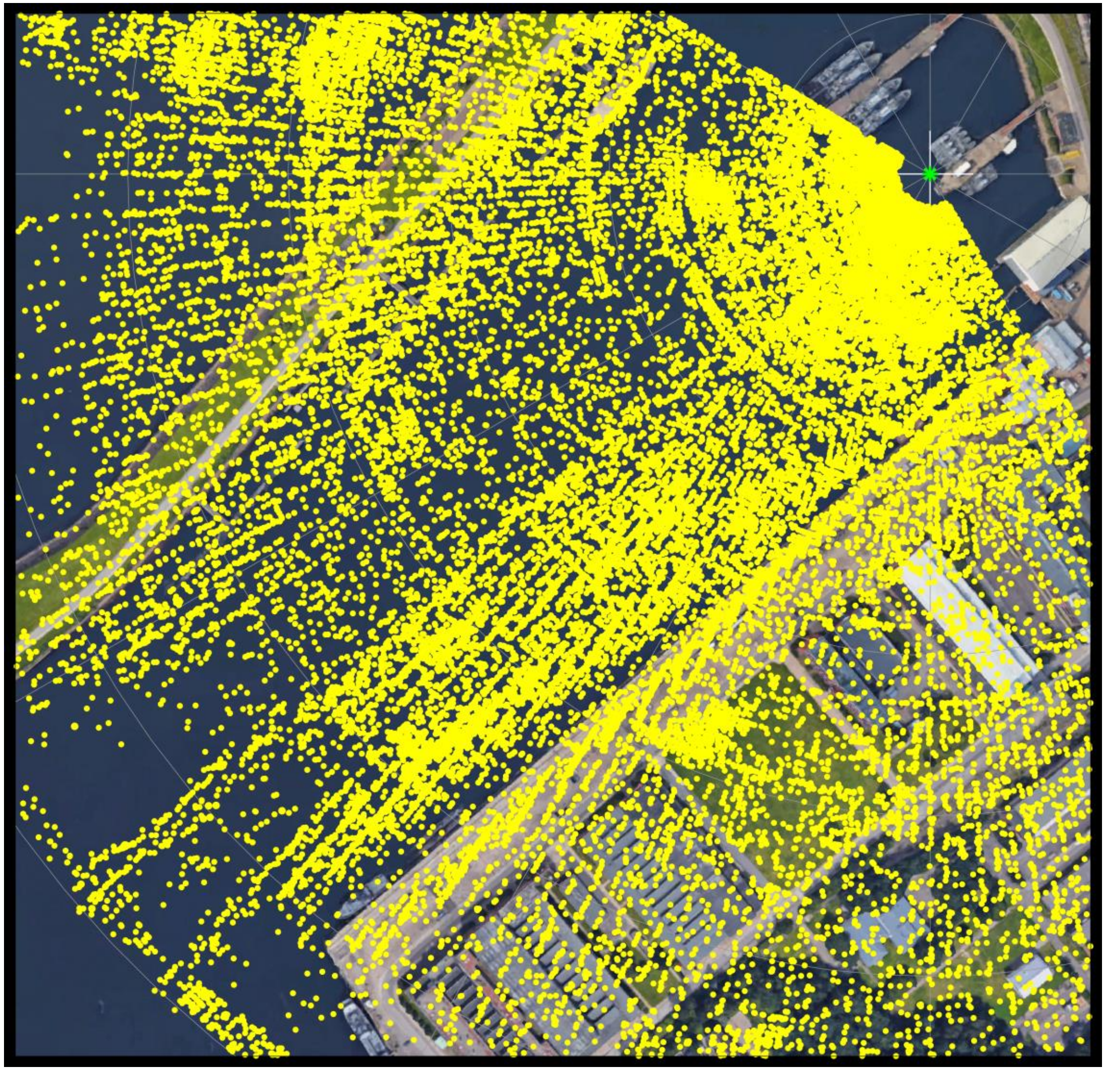
PPI BEFORE AND AFTER FALSE ALARM REDUCTION

Classification Results

- Test Dataset $D_{\text{Test}}E_1$
  - Detection with low Threshold
  - 356 Diver Contacts
  - 37843 False Alarms

TPR = 1.00, FPR = 1.00

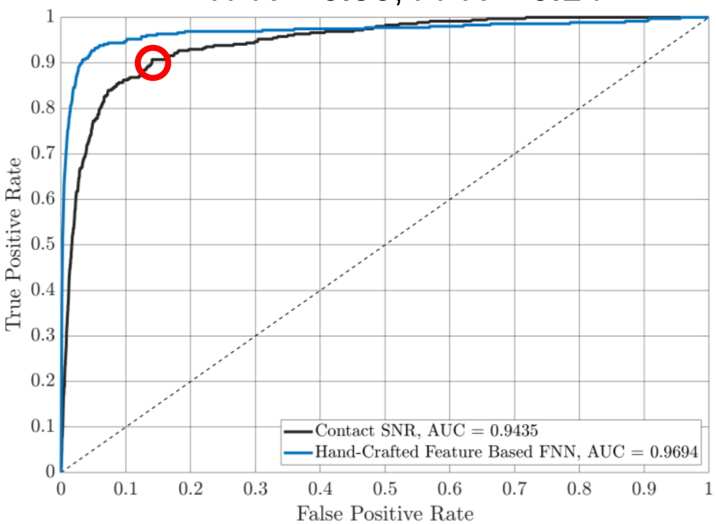


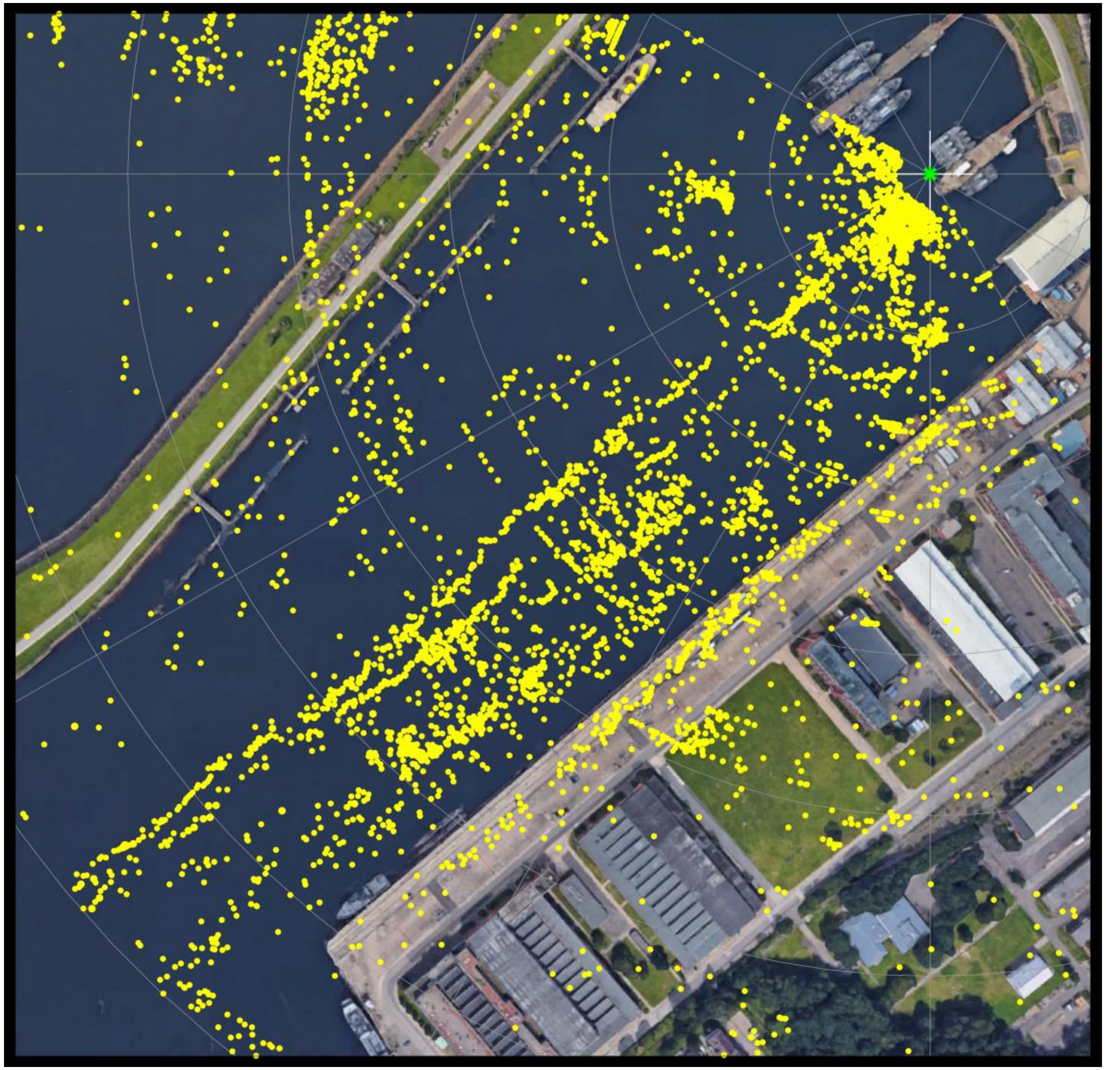


Classification Results

- Test Dataset $D_{\text{Test}}E_1$
  - Detection with higher Threshold
  - 320 Diver Contacts
  - 5301 False Alarms

TPR = 0.90, FPR = 0.14

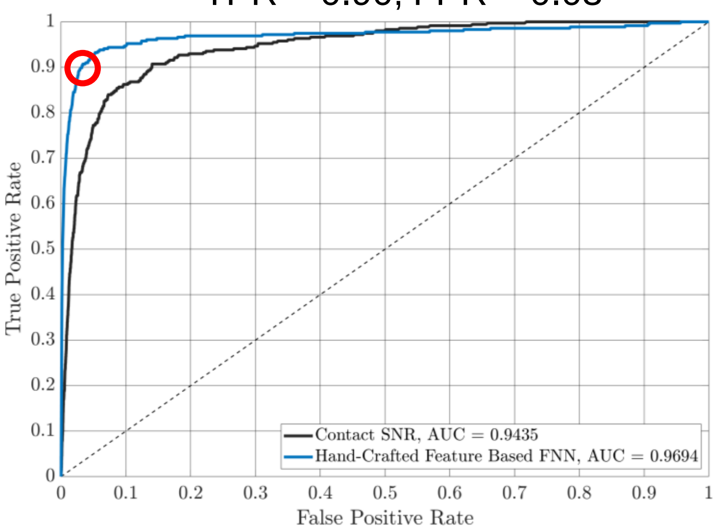


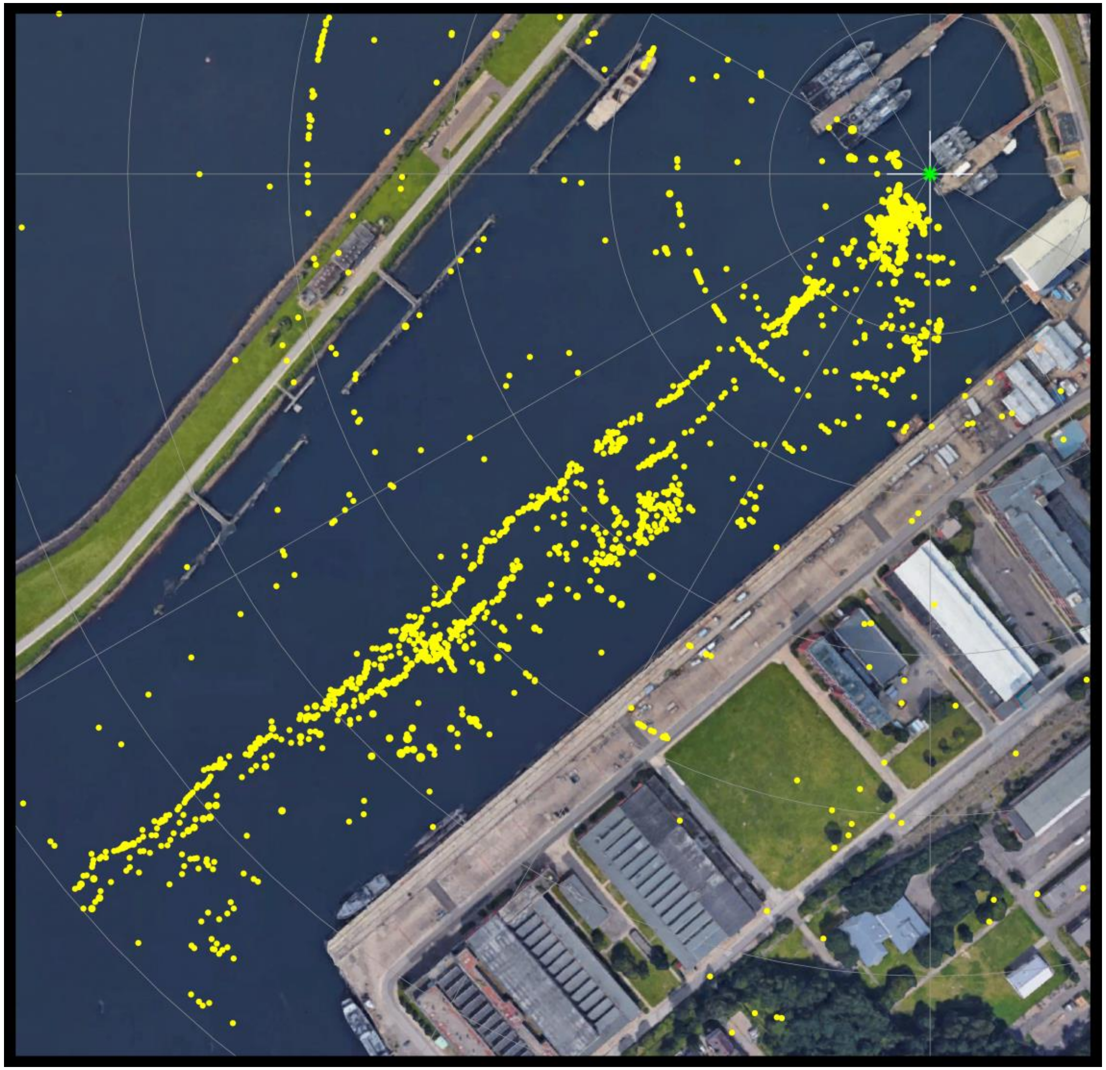


Classification Results

- Test Dataset $D_{\text{Test}}E_1$
  - Contacts after Classification
  - 320 Diver Contacts
  - 1211 False Alarms

TPR = 0.90, FPR = 0.03

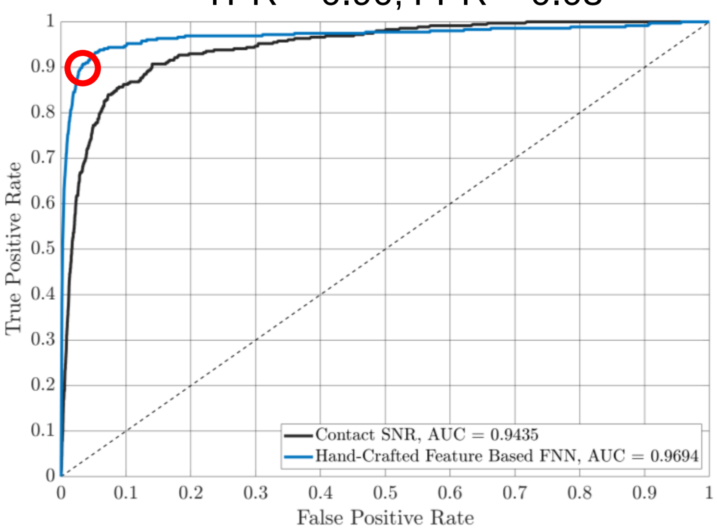


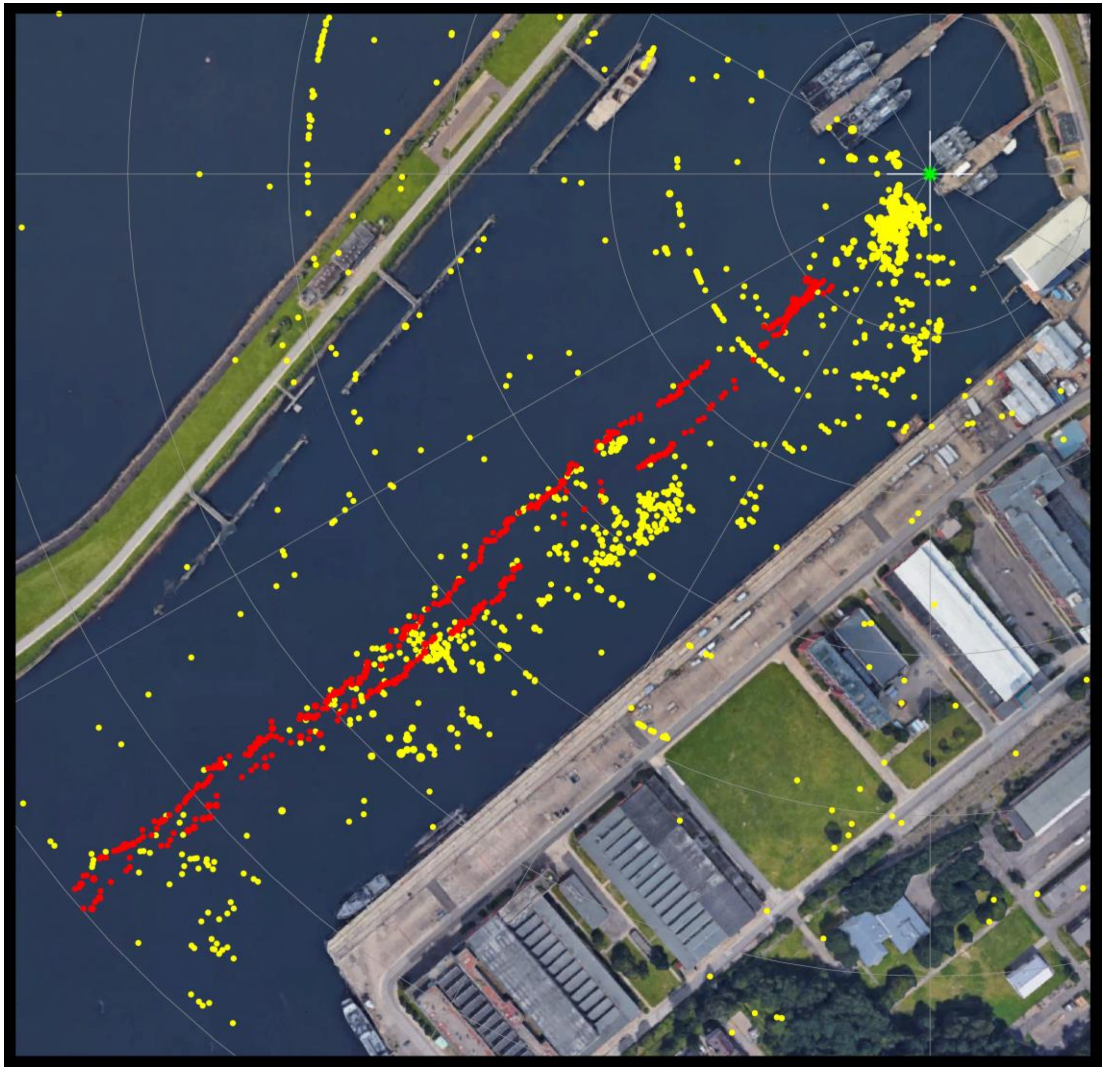


Classification Results

- Test Dataset $D_{\text{Test}}E_1$
  - Contacts after Classification
  - 320 Diver Contacts
  - 1211 False Alarms

TPR = 0.90, FPR = 0.03





SUMMARY AND FUTURE WORK

Summary

- Active signal processing is extended by feature extraction and classification.
- Two different machine learning techniques are considered.
- With both methods the number of false alarms can significantly be reduced.
- Deep CNNs perform better than considered Shallow CNN.
- Performance achieved with FNN is similar to that achieved with CNNs.

Future Work

- Use hand-crafted features in combination with features of CNNs.
- Combine different classification algorithms.
- Additional use of kinematic features estimated in tracking.
- Apply method to other active sonar applications (e.g. ASW).

THANK YOU FOR YOUR ATTENTION

Contact:

Matthias Buß
University of Wuppertal
Rainer-Gruenter-Str. 21
42119 Wuppertal, Germany
E-mail: matthias.buss@uni-wuppertal.de