Robust Underwater Image Classification using Image Segmentation, CNN, and dynamic ROI Approximation

Towards Learning Technical Systems

Stefan Bosse [1,*]

Parth Kasundra [2]

[1]University of Bremen, Dept. Mathematics & Computer Science, Bremen, Germany

[2]marinom GmbH, Bremen

[*]Presenting author

Overview

Let's talk about self-adaptive and intelligent technical systems applied to sensor data.

Overview

Let's talk about self-adaptive and intelligent technical systems applied to sensor data.

Object and Region-of-Interest detection in underwater images is a challenge, even for humans and experts!

Domain-specific Automated Region-of-Interest detection is addressed in this work.

Challenges

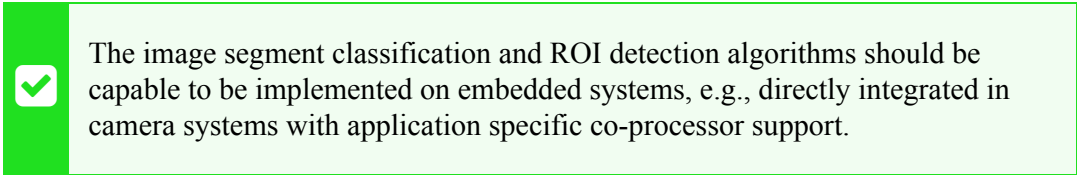
- Underwater images pose low quality with respect to illumination conditions, sharpness, and noise.
- Finding ROIs automatically can help to identify relevant regions in the image quickly by humans, or they can be used as an input for automated inspection and structural health monitoring (SHM).

Underwater inspection of technical structures, e.g., piles of sea mill energy harvester, typically aims to find material changes of the construction, e.g., rust or coverage with pocks, to make decisions about repair and to assess the operational safety.

Challenges

- Currently, for the inspection of piles of sea windmill energy harvester, divers have to go under water.
- But even if humans inspect the underwater surfaces (underwater by the diver or remotely), the scenes are cluttered and the identification of surface coverage is a challenge.
- Automated visual inspection is desired to reduce maintenance and service times.

Goals



The image segment classification and ROI detection algorithms should be capable to be implemented on embedded systems, e.g., directly integrated in camera systems with application specific co-processor support.

Goals

- The image segment classification and ROI detection algorithms should be capable to be implemented on embedded systems, e.g., directly integrated in camera systems with application specific co-processor support.

- The aim is to achieve an accuracy of at least 85-90% for the predicted images, with a high degree of generalization and independence from various image and environmental parameters such as lighting conditions and background colouration, as well as relevant classification features.

Summary and Highlights

- We propose and evaluate a hybrid approach with segmented classification using small-scaled CNN classifiers (with less than 20000 hyper parameters and less than 3 Million unity vector operations)
- A reconstruction of labelled ROIs is provided by using an iterative mean and expandable bounding box algorithm.
  - The iterative bounding box algorithm combined with bounding box overlap checking suppress spurious wrong segment classifications and represent the best and most accurate matching ROI for a specific classification label, e.g., surfaces with pocks coverage.

The overall classification accuracy (true-positive classification) with respect to a single segments is about 70%, but with respect to the iteratively expanded ROI bounding boxes it is about 90%.

Images and Classes

The underwater inspection of technical structures, e.g., construction parts of off-shore wind turbines like piles, involves the identification of various parts in the underwater images:

1. Background with water, bubbles, and fishes, summarized as feature class *B*;
2. Technical structure, e.g., a mono pile of a wind turbine, summarized as feature class *P*;
3. Formation of coverage with marine vegetation or organisms on the surface of the structure, summarized as feature class *C*.

Images and Classes

- The images set consists of different RGB underwater images posing a
  - high variance in illumination conditions,
  - spatial orientation,
  - noise (bubbles, blurring), and
  - colour palettes.
- The images are snapshots taken from videos recorded by a human diver with an underwater camera.

Methods and Architecture

Methods

1. Manual image labelling with labelled polygon regions
2. Segmentation of raw input images in smaller segment images (segment size $n \times m$ pixels, e.g., 64×64)
3. Convolutional Neural Networks (CNN)
   - Input: Image Segment
   - Output: Class probabilities (B, P, C) and maximum likelihood selection of best candidate (or none)
4. Iterative rectangular bounding box approximation using density-based clustering (DBSCAN) and centre-of-mass (COM) computations; COM determines the centre of the bounding box.
   - Input: All labelled segments of input images
   - Output: Set of labelled ROI coordinates
5. Post correction of overlapping conflicting labelled ROIs

Architecture

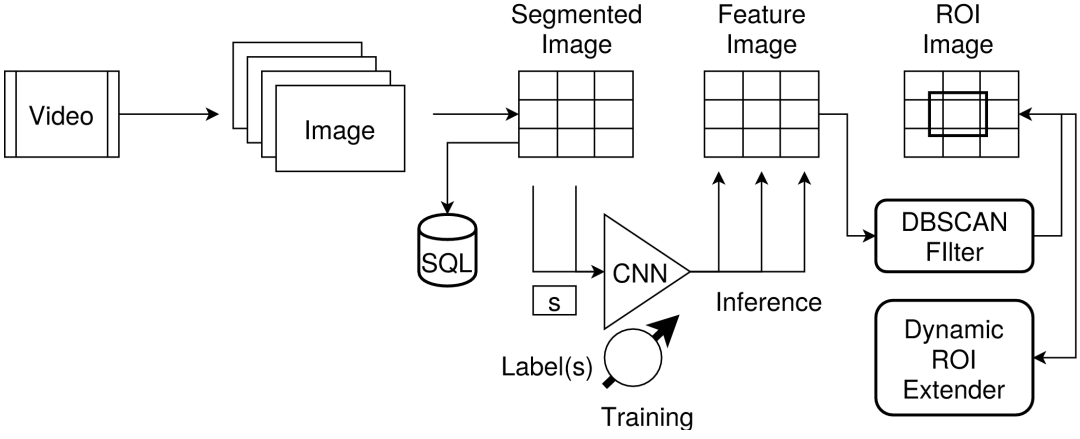


Fig. 1. Overview of the data flow architecture and the used algorithms

Software

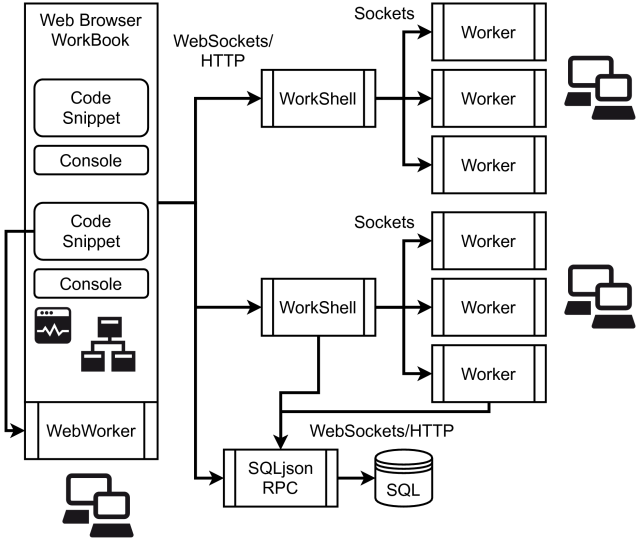


Fig. 2. Web browser-based software architecture with remote shell worker processes (Bosse, Appl. Sciences, 2022)

CNN Parameters

- Four different CNN model architectures were evaluetd:

Arch.	Layer	Filter	Activation	Output	Parameter	VecOps
A (8/16)	Conv	[5×5]×8,s=1	-	64×64×8	608	4915200
	Relu	-	relu	64×64×8	32768	32768
	Pool	[2×2]×8,s=2	-	32×32×8	0	8192
	Conv	[5×5]×16,s=1	-	32×32×16	3216	6553600
	Relu	-	relu	32×32×16	16384	16384
	Pool	[3×3]×16,s=3	-	10×10×16	0	1600
	Fc	-	relu	1×1×3	4803	9600
	SoftMax	-	-	3	3	3
					Σ57782	**Σ11537347**
B (4/8)	Conv	[5×5]×4,s=1	-	64×64×4	304	2457600
	Relu	-	relu	64×64×4	16384	16384
	Pool	[2×2]×4,s=2	-	32×32×4	0	4096
	Conv	[5×5]×8,s=1	-	32×32×8	808	1628400
	Relu	-	relu	32×32×8	8192	8192
	Pool	[3×3]×8,s=3	-	10×10×8	0	800
	Fc	-	relu	1×1×3	2403	4800
	SoftMax	-	-	3	3	3
					Σ28094	**Σ4127878**

CNN Parameters

Arch.	Layer	Filter	Activation	Output	Parameter	VecOps
C (8/8)	Conv	[5×5]×8,s=1	--	64×64×8	608	4915200
	Relu	-	relu	64×64×8	32768	32768
	Pool	[2×2]×8,s=2	-	32×32×8	0	8192
	Conv	[5×5]×8,s=1	-	32×32×8	1608	3276800
	Relu	-	relu	32×32×8	8192	8192
	Pool	[3×3]×16,s=3	-	10×10×8	0	800
	Fc	-	relu	1×1×3	2403	4800
	SoftMax	-	-	3	3	3
					Σ45582	**Σ8246755**
D (4/4)	Conv	[5×5]×4,s=1	-	64×64×4	304	2457600
	Relu	-	relu	64×64×4	16384	16384
	Pool	[2×2]×4,s=2	-	32×32×4	0	4096
	Conv	[5×5]×4,s=1	-	32×32×4	404	819200
	Relu	-	relu	32×32×4	4096	4096
	Pool	[3×3]×4,s=3	-	10×10×4	0	400
	Fc	-	relu	1×1×3	1203	2400
	SoftMax	-	-	3	3	3
					Σ22394	**Σ3304179**

Mean Bounding Box Algorithm (MBB)

- There is a set of class symbols Σ and a class matrix ***M*** consisting of elements labelling an image segment with a class, so that:

$$\Sigma = \{B, P, C, U\}$$
$$\sigma \in \Sigma$$
$$\hat{M} = \begin{pmatrix} \sigma_{1,1} & .. & \sigma_{1,j} \\ \sigma_{2,1} & .. & \sigma_{2,j} \\ .. & .. & .. \\ \sigma_{i,1} & .. & \sigma_{i,j} \end{pmatrix}$$

- The matrix ***M*** is flattened to a point cloud list set $P=\{p_\sigma\}_{\sigma\in\Sigma}$.
- Each class set *p* contains the matrix positions of the respective elements, i.e., $p_\sigma=\{\langle i,j\rangle\}$, with all points classified by the CNN to the same label class $\sigma\in\Sigma$.

Mean Bounding Box Algorithm (MBB)

- The DBSCAN clustering will return a group list of points that satisfy the clustering conditions, one point group list for each label class:

$$DBSCAN : P \rightarrow \left\{ \{p_j\}_j, \{p_k\}_k, \{p_l\}_l, .. \right\}, j \neq k \neq l$$
$$P : \{p_i\}_i, i = \{1, 2, 3, .., n\}$$
$$p_i =< i, j > \in R^2$$

- It is assumed that a cluster will contain a majority of correctly classified points (segments), and a minority of scattered wrong classified points.

Mean Bounding Box Algorithm (MBB)

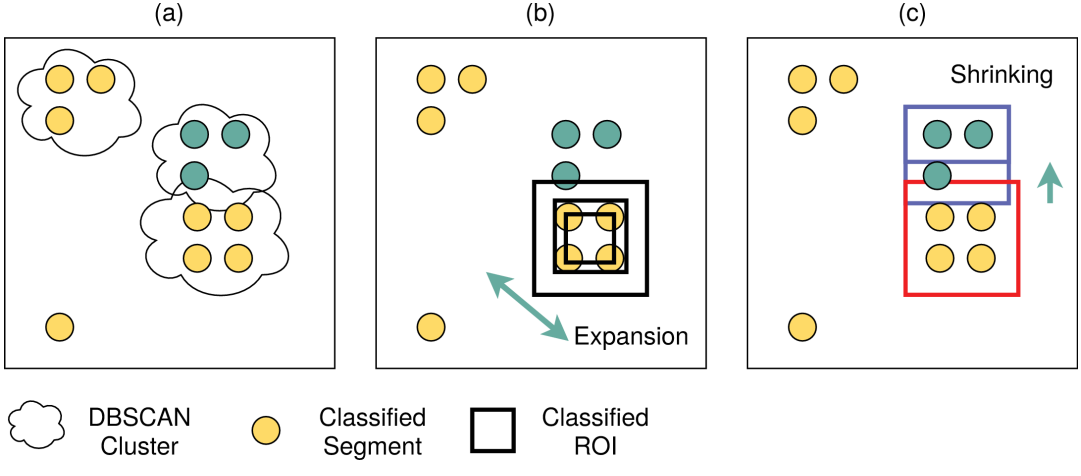


Fig. 3. Iterative bounding box expansion with final conflict overlapping shrinking

Mean Bounding Box Algorithm (MBB)

- The MBB algorithm computes points $\langle x_1, y_1, x_2, y_2 \rangle$ of a bounding box that is centred at the mass-of-centre point *c* of all points of a cluster and with outer sides given by the vectorial mean centred position of all points above or below, and left or right form the *c* point.
- The expansion of a previously computed bounding box is done by all points outside of the current bounding box, performing the next extension iteration.
  - Again, a spatial position averaging is performed, extending the boundary of the bound box.
  - The expansion is performed iteratively.
  - Each step includes more points, but increases the probability that the bound box is over-sized with respect to spurious outlier points that result from wrong CNN classifications.

Mean Bounding Box Algorithm (MBB)

- In case of high iteration loop values, bounding boxes from different classes can overlap.
- To reduce overlapping conflicts, a class priority is introduced layering the class regions by relevance.
- After the ROI expansion is done, overlapping bounding boxes with lower priority are shrink until all overlap conflicts are resolved.
  - Commonly, more than one side of the bounding box can be shrunken to reduce the overlap conflict.
  - The possible candidates are evaluated and sorted with respect to the amount of shrinkage at each side.
  - The lowest shrinkage is applied first. If the conflict is not reduced by the selected side shrinking, the next side is shrink until the conflict (with one or more higher priority bounding boxes) is reduced.

Experiments and Results

Training and test data: Randomly chosen sub-set of about 10000 image segments taken from about 300 snap shot images

Experiments and Results

Training and test data: Randomly chosen sub-set of about 10000 image segments taken from about 300 snap shot images

Four different identical models were trained and applied in parallel (different random initialisation and sub-set of images)

Results

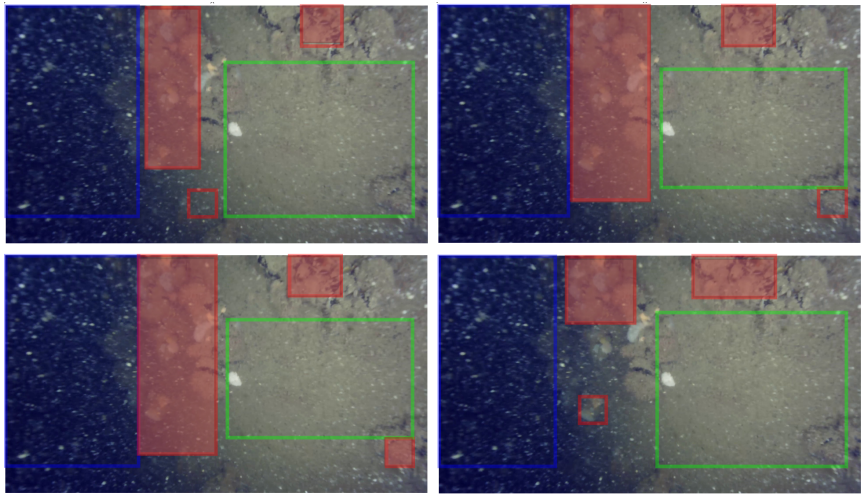


Fig. 4. Classified bounding boxes for one image using four models trained in parallel (same parameters) but with different random initialisation and training data sub-set (Blue: class background, red: class coverage, green: class free construction surface)

Results

Data Set	Total error ($\neg TP_C$) %	Error (¬TP)/class (B,P,C) %	Prediction accuracy/Class C (TP,FP,TN,FN) %
Training	10.6±1.5	5.0±3.4 6.0±2.8 21.0±7.1	79.0±7 4.8±2 94.7±6.6 10.5±3.1
Test	11.1±1.8	5.3±2.6 5.8±3.2 22.0±8.3	78.0±4.3 5.1±2.2 95.1±2.1 11.0±4.4
All	10.9±1.6	4.2±2.8 5.9±3.4 21.7±8	78.4±8 5.0±2.2 95.0±2.2 10.8±4

Tab. 1. Accumulated prediction results for training, test, entire data set union with statistical features of the model ensemble trained in parallel (using different data sub-sets and random initialisation). All errors with 2σ standard deviation interval, and N=9000 samples, n=3000 for each class, and using CNN architecture A.

Results

CNN Architecture	Parameters	Forward Time	Backward Time
A (8/16)	122587	18 ms [1], 0.5 ms [2]	26 ms [1], 1 ms [2]
B (4/8)	66639	8 ms [1]	10 ms [1]
C (8/8)	104603	12 ms [1]	18 ms [1]
D (4/4)	58047	6 ms [1]	8 ms [1]

Tab. 2. Forward and backward (training) times for one 64 × 64 × 3 segment and different CNN architectures using the JavaScript ConvNet.js classifier [1] and TensorFlow (CPU) [2]

Conclusion

Although the overall classification accuracy is about 90%, the high variance of the segment prediction results across differently trained models (model ensemble all having the same architecture) limits the output quality of the labelled ROI detector, typically resulting in an underestimation of the classified regions and a lacking of generalisation.

Although the overall classification accuracy is about 90%, the high variance of the segment prediction results across differently trained models (model ensemble all having the same architecture) limits the output quality of the labelled ROI detector, typically resulting in an underestimation of the classified regions and a lacking of generalisation.

But the presented static segment prediction with point clustering and iterative selective bounding box approximation with final overlap conflict reduction is still reliable. Similar to random forest trees, a multi-model prediction with model fusion (e.g., major coverage estimation) is proposed to get the best matching bonding boxes for the relevant classes.

The reduction of the CNN complexity with respect to the number of filters and dynamic parameters does not lower the classification accuracy significantly.

- Although, CNN are less suitable for low-resource embedded systems, the CNN architecture D (4/4) could be implemented in an embedded camera systems, expecting overall ROI extraction times for one image frame about 5 seconds, not suitable for real-time operation (maximal latency 100 ms). Using control-path parallelisation performing the image segment classifications in parallel, the ROI extraction could be reduced to 1 second using generic multi-core CPUs, or 100 ms using FPGA-based co-processors.

Robust Underwater Image Classification using Image Segmentation, CNN, and dynamic ROI Approximation

Towards Learning Technical Systems

Stefan Bosse [1,*]

Parth Kasundra [2]

[1]University of Bremen, Dept. Mathematics & Computer Science, Bremen, Germany

[2]marinom GmbH, Bremen

[*]Presenting author