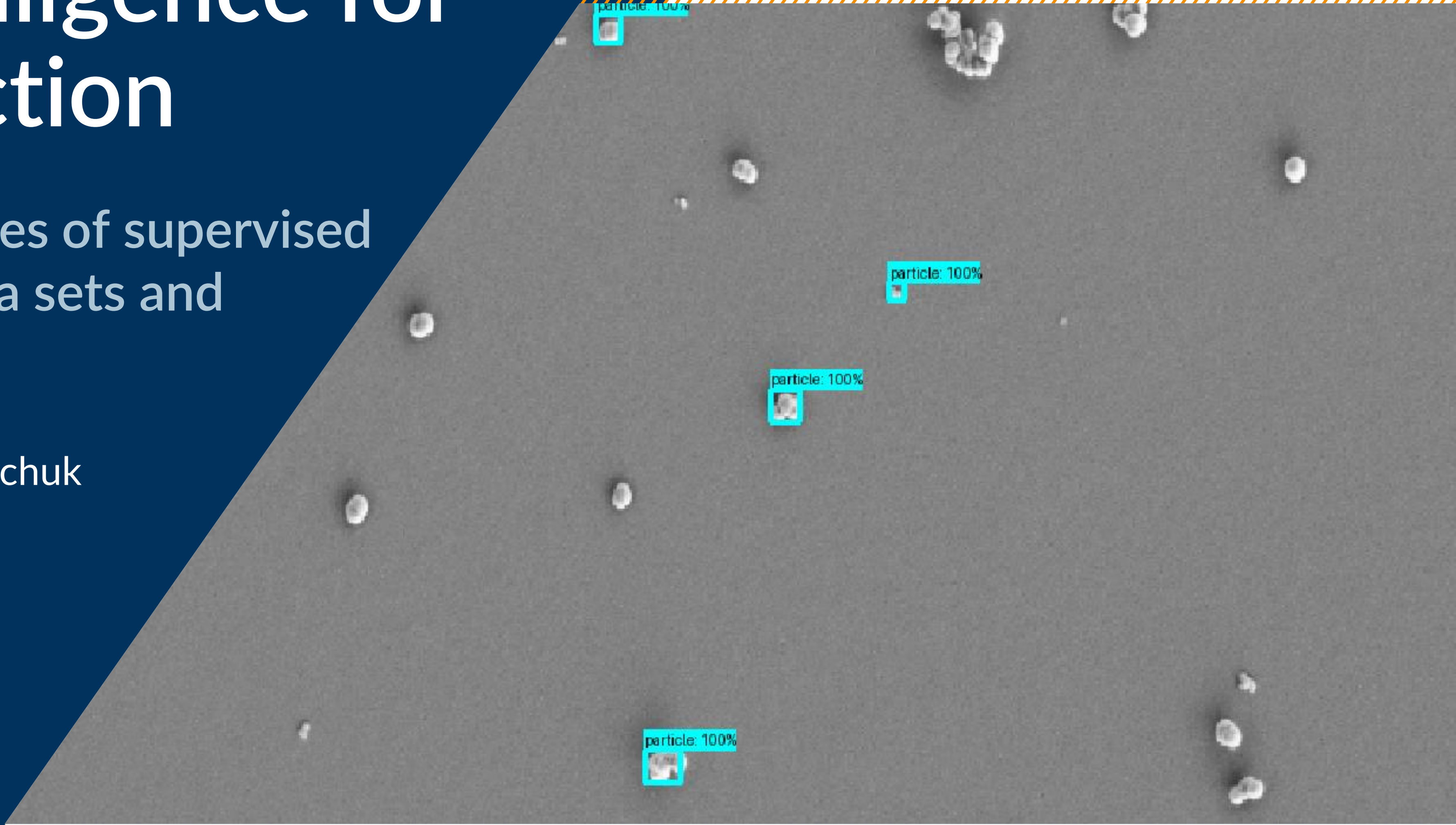
Bachelor thesis presentation
September 2024

IOM
Leibniz Institute of
Surface Engineering

Implementation of Artificial Intelligence for Defect Detection

Assessment of capabilities of supervised techniques for small data sets and objects

Natasha Alexandra Hrycan Robachuk



Leibniz Institute of
Surface Engineering

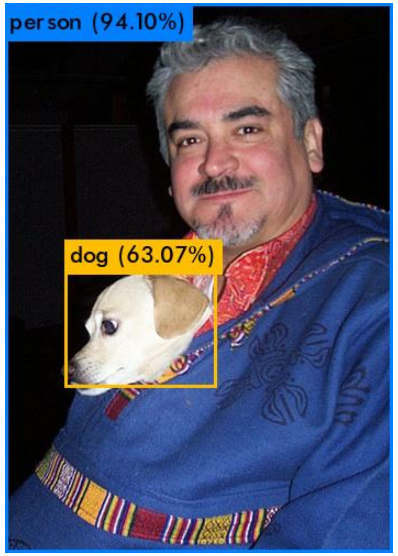
Outline

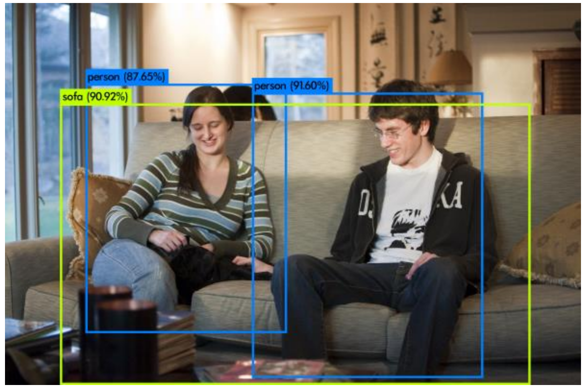
Introduction

- Image Analysis with Machine Learning
- Motivation
- Challenges

Image Analysis with Machine Learning

- Machine learning can be used not only for detecting objects in images, but also to classify them.
- It recognizes characteristic features of an object and determines its location and the confidence level.
- This can speed up the analysis of image batches used in experiments.
- Depending on the data provided by the detector, we can extract or compute features such as: area of the object, color, density of objects within an area, etc.





Examples of detections performed by YOLO [20] in different datasets (PASCAL VOC and COCO). Source: Padilla, R., Netto, S. L. & da Silva, E. A. B. A Survey on Performance Metrics for Object-Detection Algorithms in 2020 International Conference on Systems, Signals and Image Processing (IWSSIP) (2020), 237–242.

Motivation

- Analysis of defects in flexible transparent oxide films (single layer SiO_2 films obtained by UV photoconversion and thickness of 100 nm).
- The images were obtained with a Scanning Electron Microscope.
- Defects affect the permeability of coating films.
- Aspects to consider when analyzing the correlation of permeability loss and presence of defects: types of present defects and density of defects.

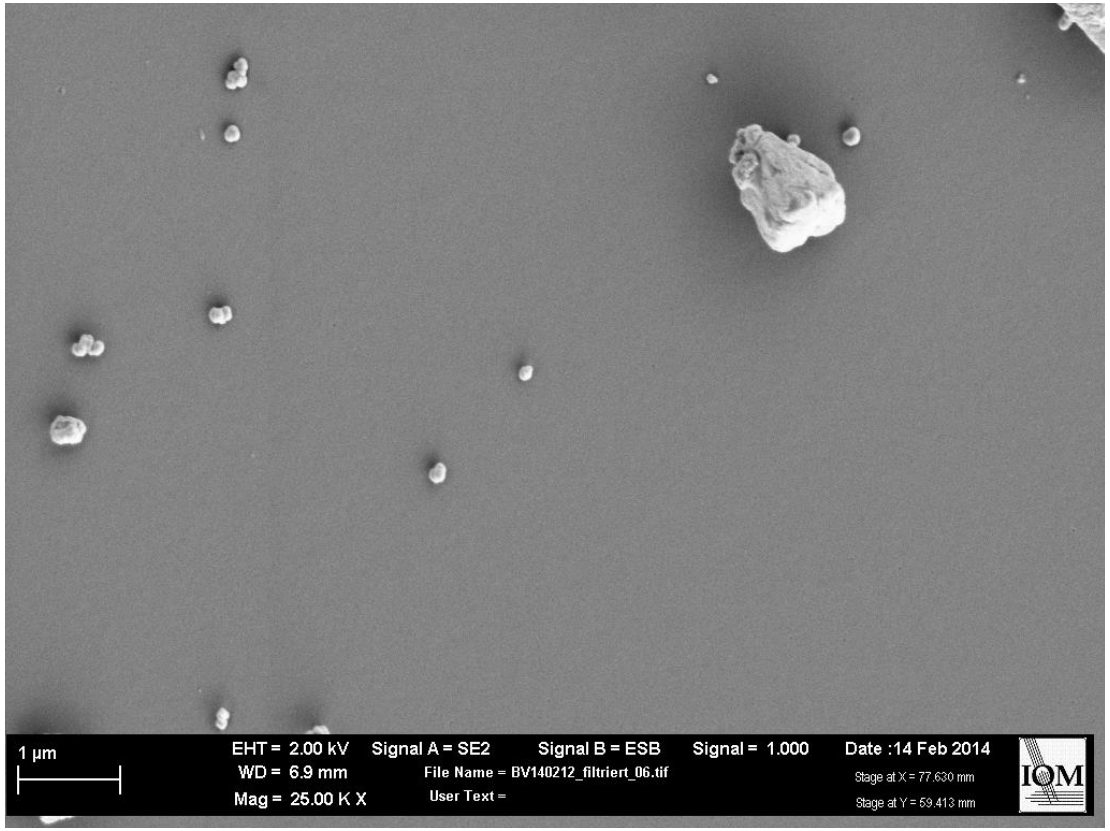
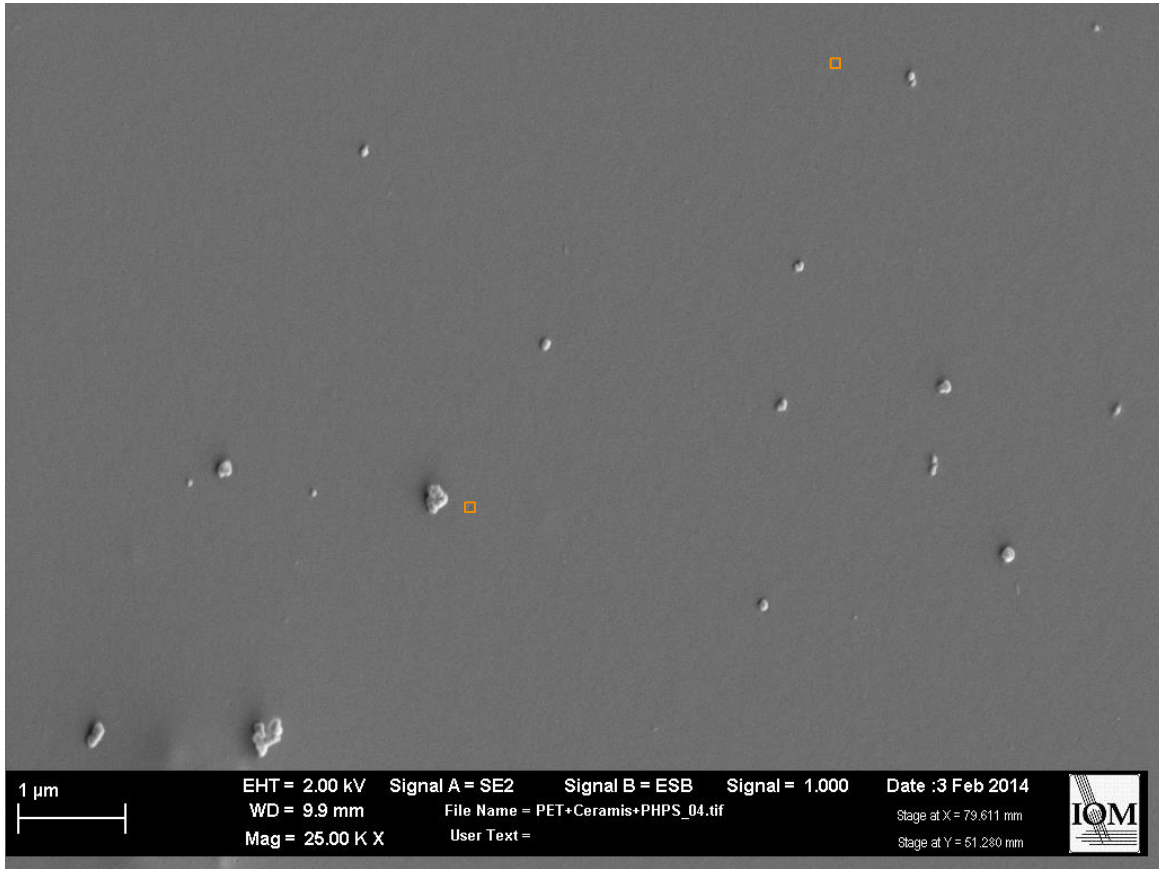
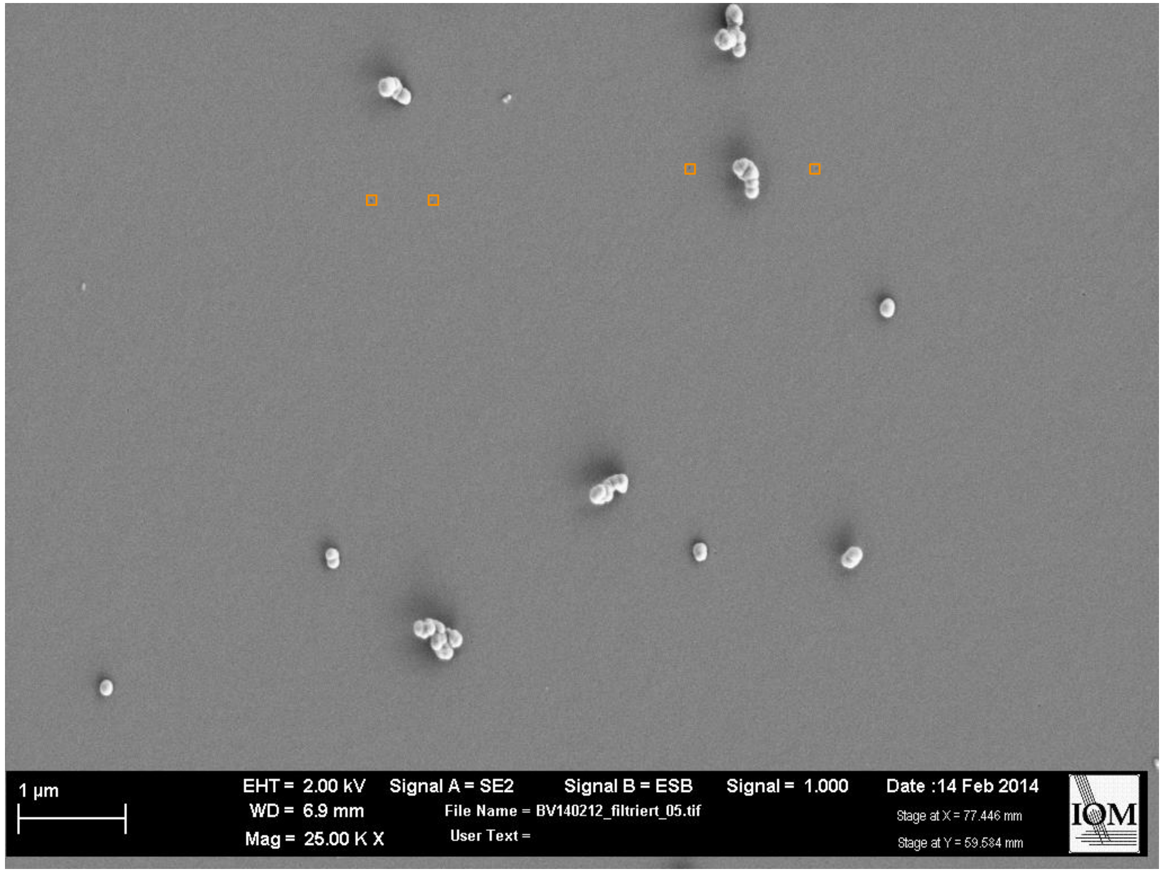


Image from the dataset containing particles laying on the surface of the coating film.

Defects in ceramic coating films





Images from the dataset containing pinholes (highlighted with orange boxes) and particles laying on the surface of the coating film.

Challenges

- Small defects (such as pinholes) located in proximity can have the same effect as a big defect (such as a big hole).
- Images with too many defects, shadows and defects laying on top of each other.
- Using only a small dataset (< 100 images), considering the time it takes to label all defects.
- The “small object problem”: if an object is too small compared to the image where it is, an object detection model may have troubles finding it.

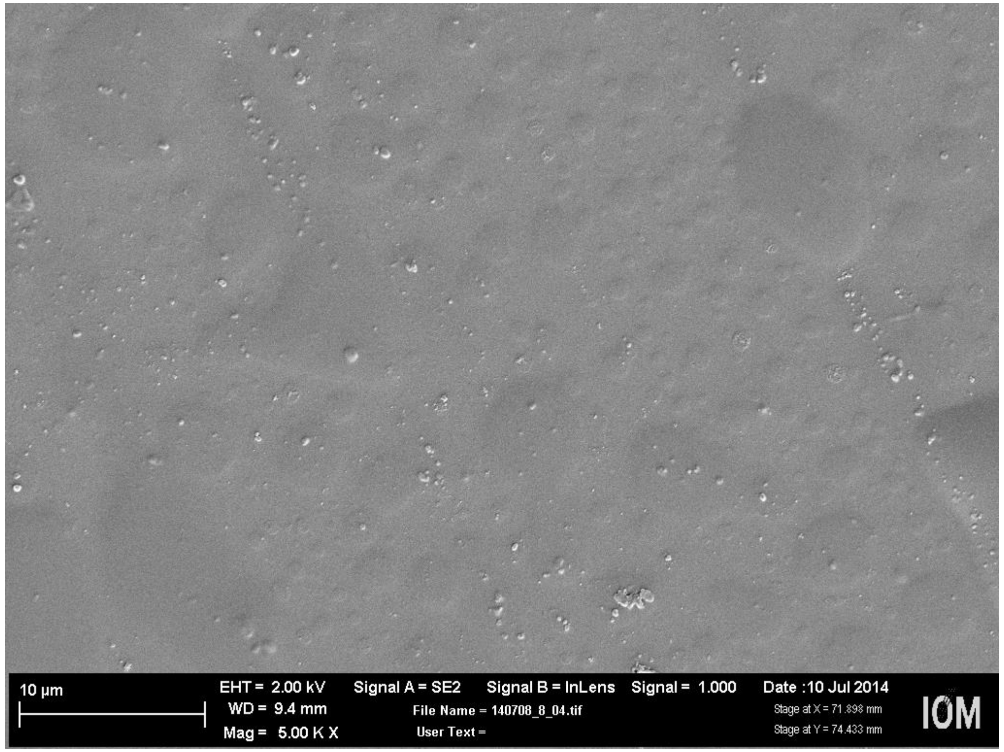
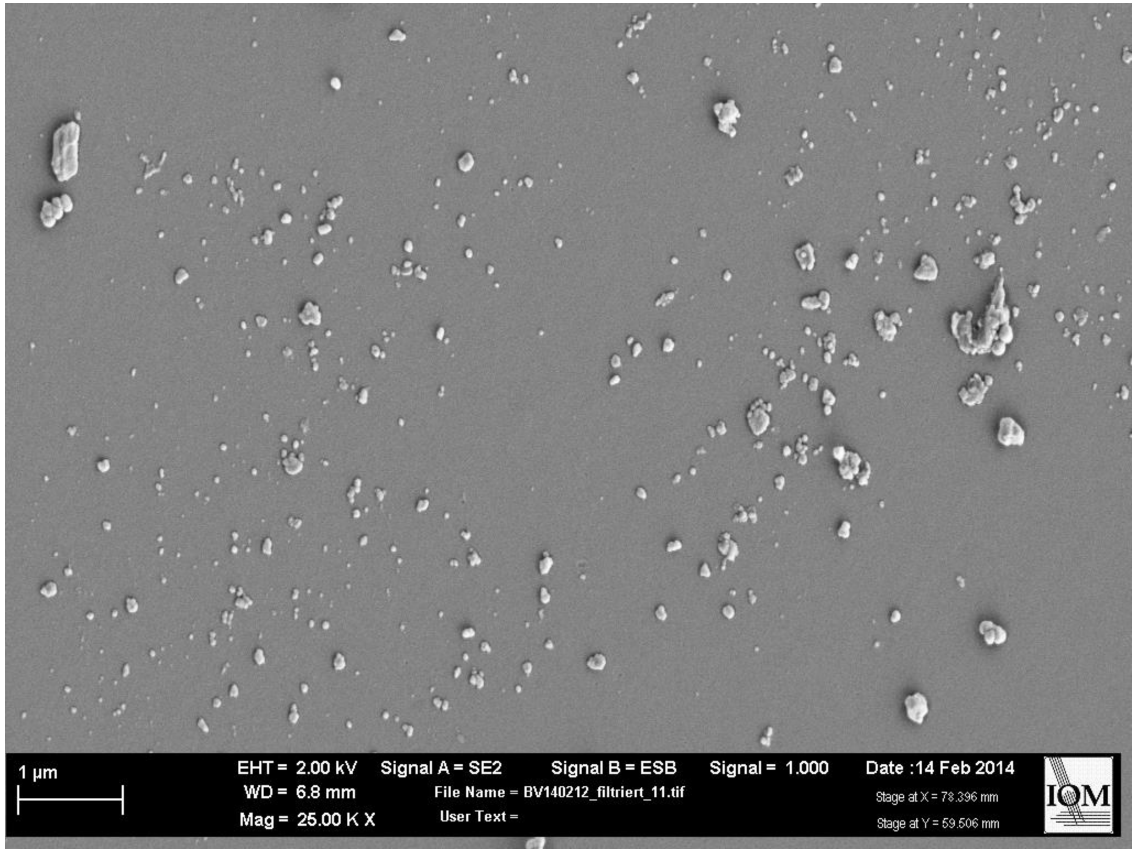
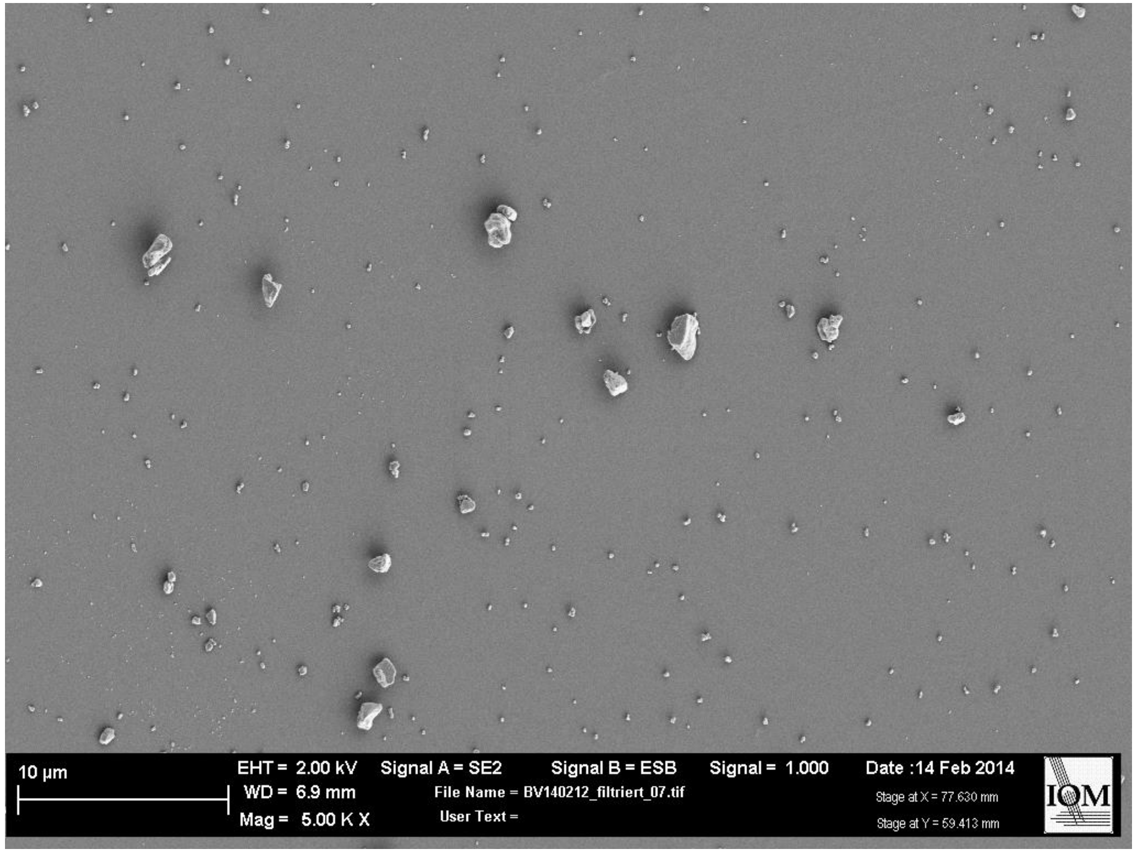


Image from the dataset showing defects difficult to label.

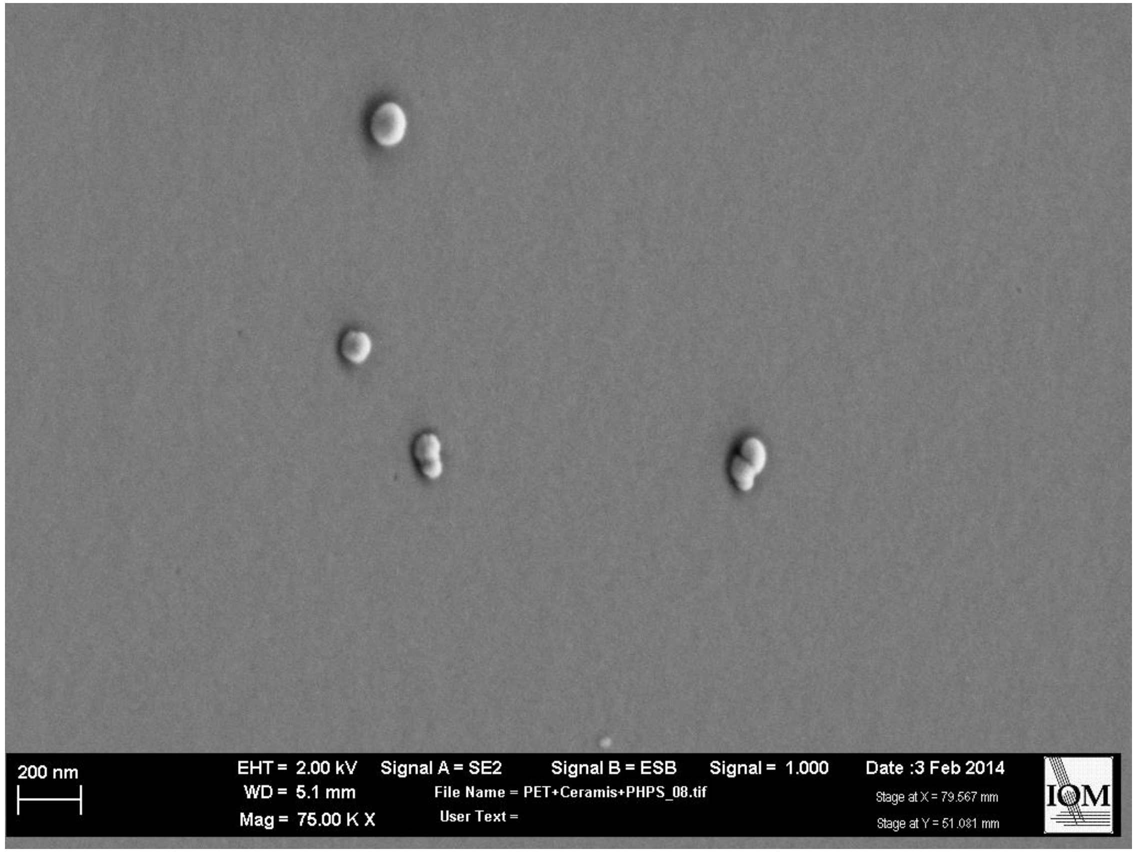
Challenging defects in ceramic coating films

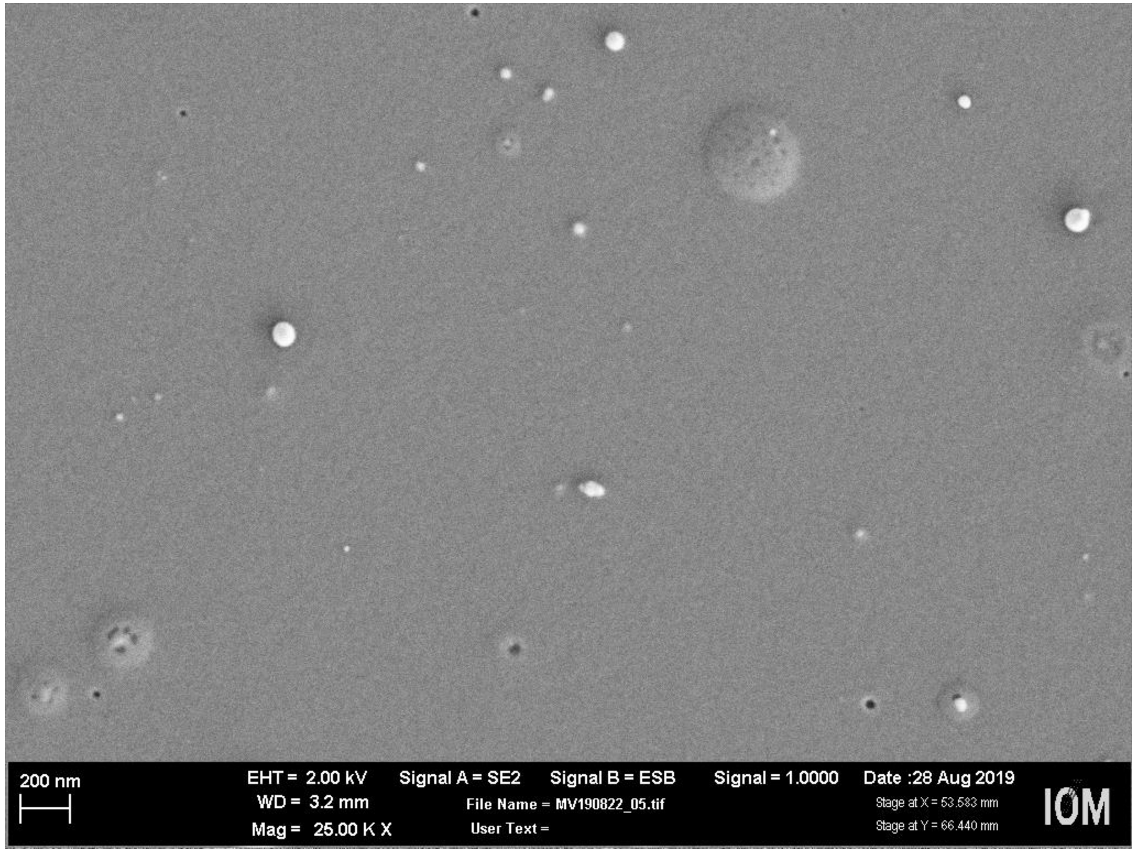




Images from the dataset containing particles laying on the surface of the coating film.

Challenging defects in ceramic coating films





Images from the dataset containing particles laying on the surface of the coating film, bubbles and pinholes.

Results

We tested a dataset of 44 images containing common defects in coating films with 2 methods.

- Method 1
- Method 2

Method 1 - Workflow

Label defects in images
(generate bounding boxes)

→

Provide labels and images to train the model
(the model learns to recognize how these objects look like)

→

Test
(evaluate results and adjust parameters of the model, run again if necessary)

Method 1 - Metrics

The precision **P** of a model is defined as the ability for it to detect correctly relevant objects.

$$P = \frac{TP}{TP + FP}$$

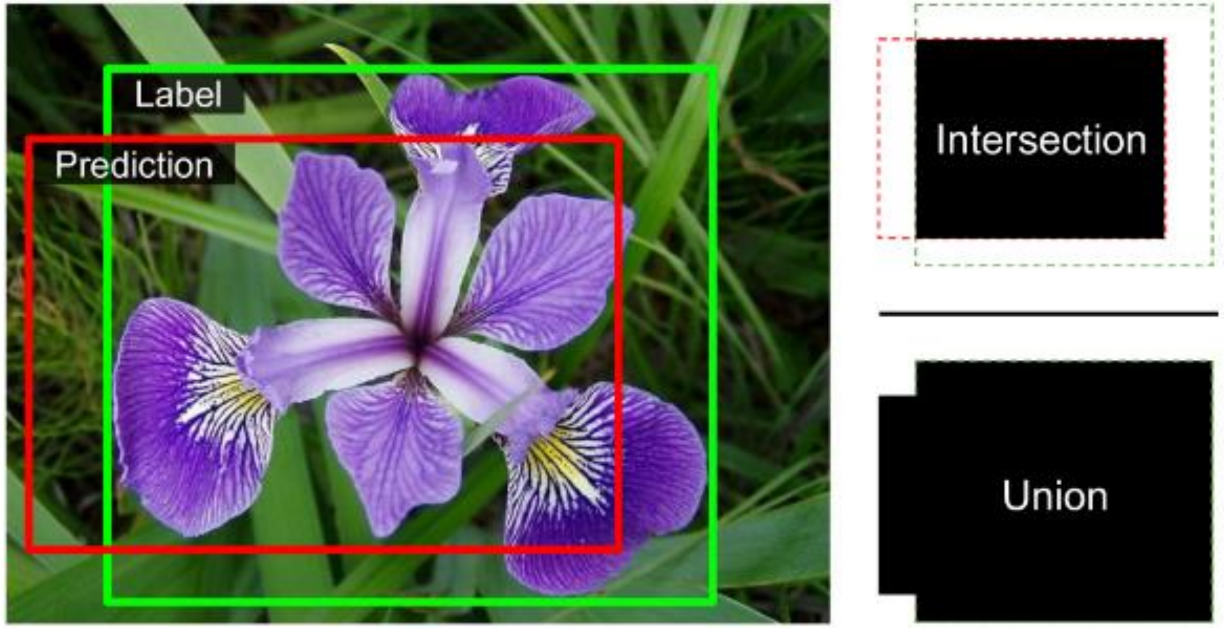
TP – True Positive detection
FP – False Positive detection

Method 1 - Metrics

Model	mAP@0.5
EfficientDet D3	14,4%
Faster R-CNN Resnet50 V1	11%
Roboflow Object Detection 3.0	37%

Results of mean average precision for predictions with an IoU threshold of 50% (mAP@50).

Intersection over Union (IoU)



$$IoU = \frac{A_{gt} \cap A_p}{A_{gt} \cup A_p}$$

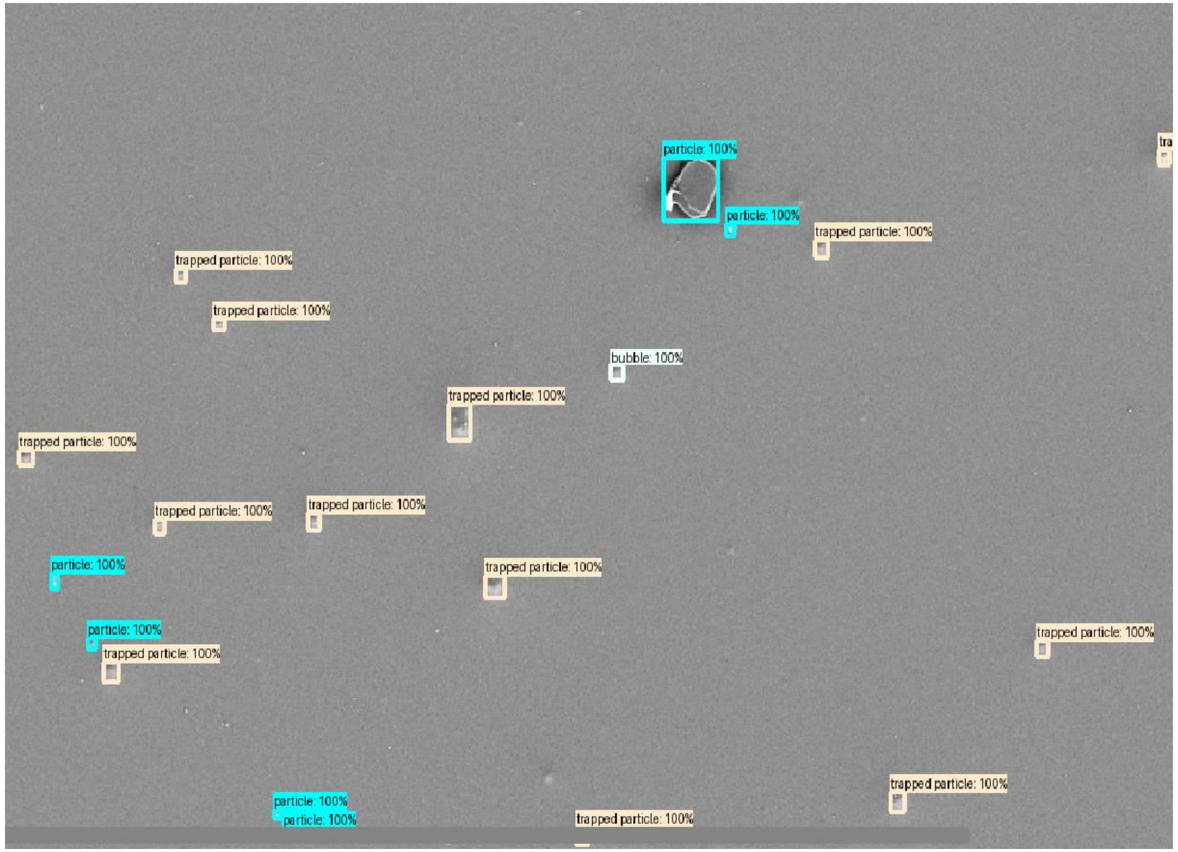
IoU metric explained graphically. Source: Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow ISBN: 978-1-098-12597-4 (O'Reilly Media, Sebastopol, CA, 2023).

Method 1 – EfficientDet D3

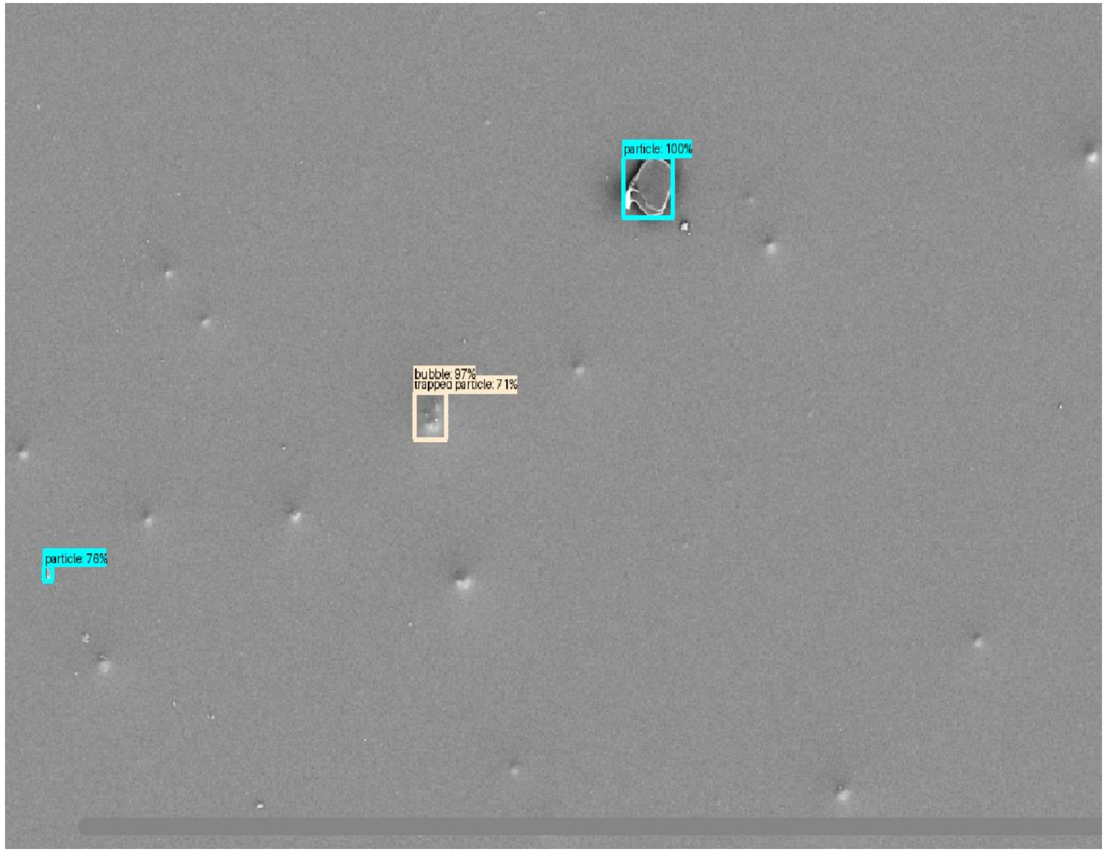
Model	mAP@0.5
EfficientDet D3	14,4%
Faster R-CNN Resnet50 V1	11%
Roboflow Object Detection 3.0	37%

This model presented difficulties mainly with small objects, pinholes were barely detected in any of the images

Ground truth



Detection result

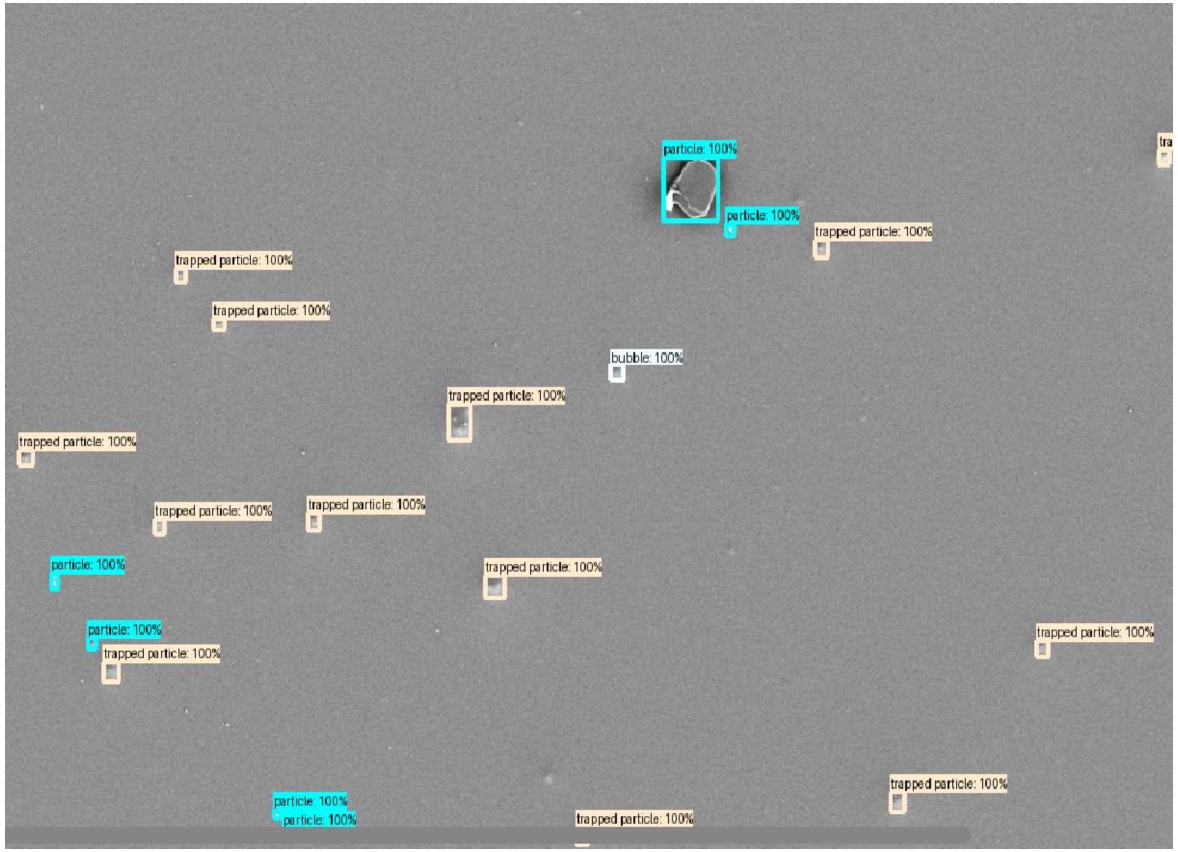


Method 1 – Faster R-CNN Resnet50 V1

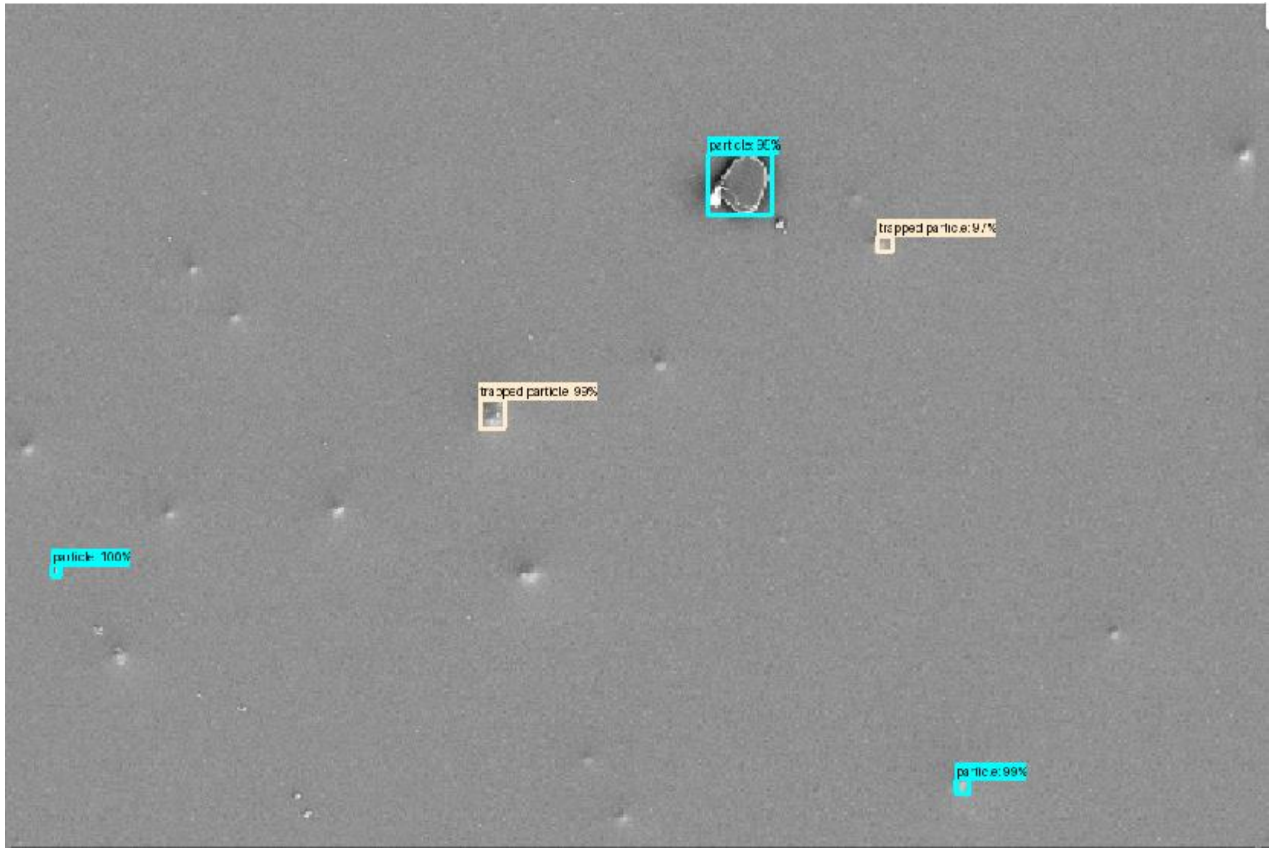
Model	mAP@0.5
EfficientDet D3	14,4%
Faster R-CNN Resnet50 V1	11%
Roboflow Object Detection 3.0	37%

This model presented problems with small and medium size particles; pinholes were barely detected.

Ground truth



Detection result

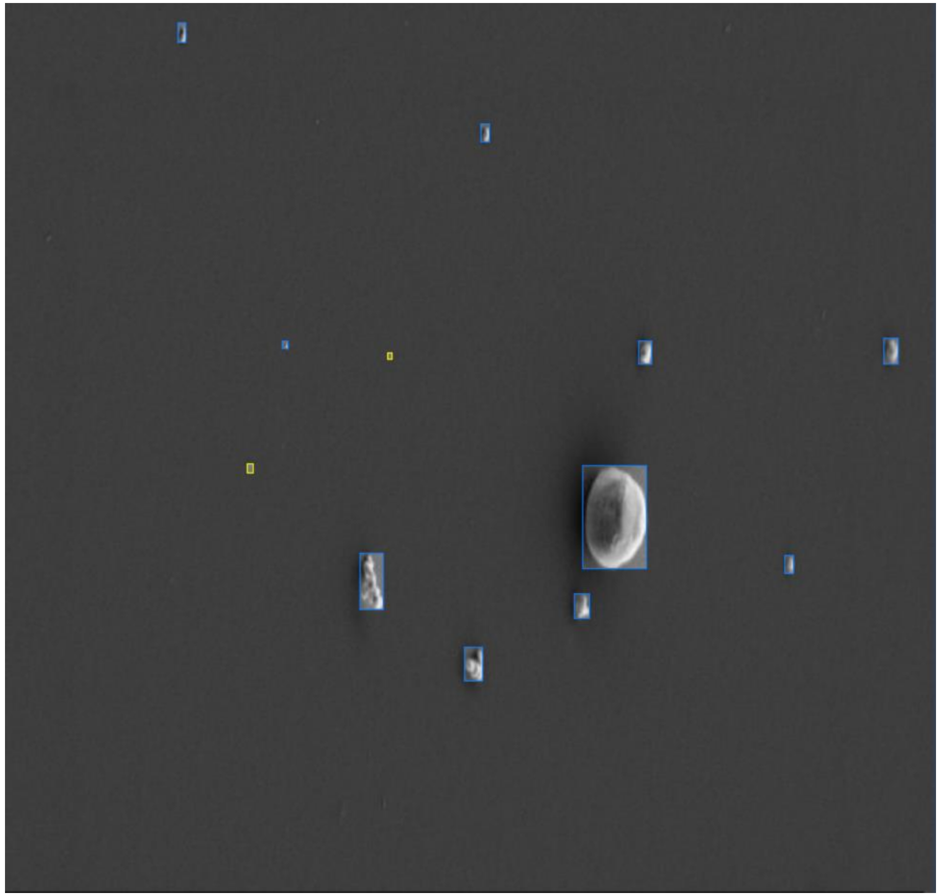


Method 1 – Roboflow Object Detection 3.0

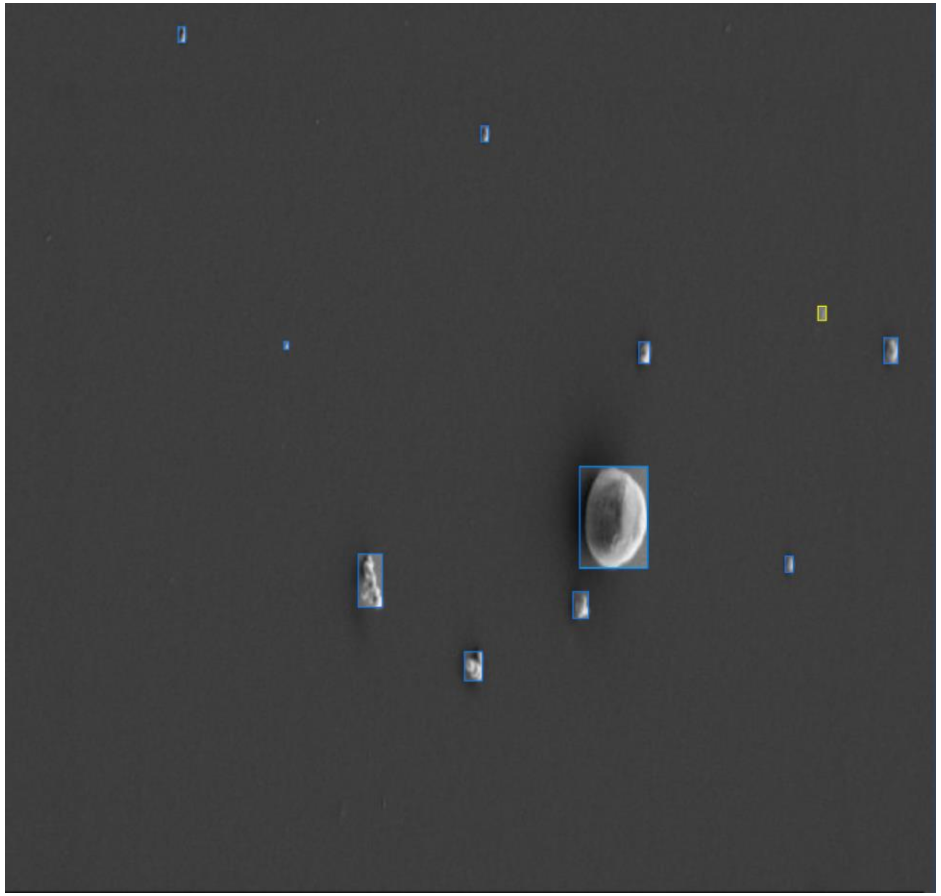
Model	mAP@0.5
EfficientDet D3	14,4%
Faster R-CNN Resnet50 V1	11%
Roboflow Object Detection 3.0	37%

This model detected better small, medium and large particles. There were problems with pinholes, such as the generation of false positives (see Detection result)

Ground truth



Detection result

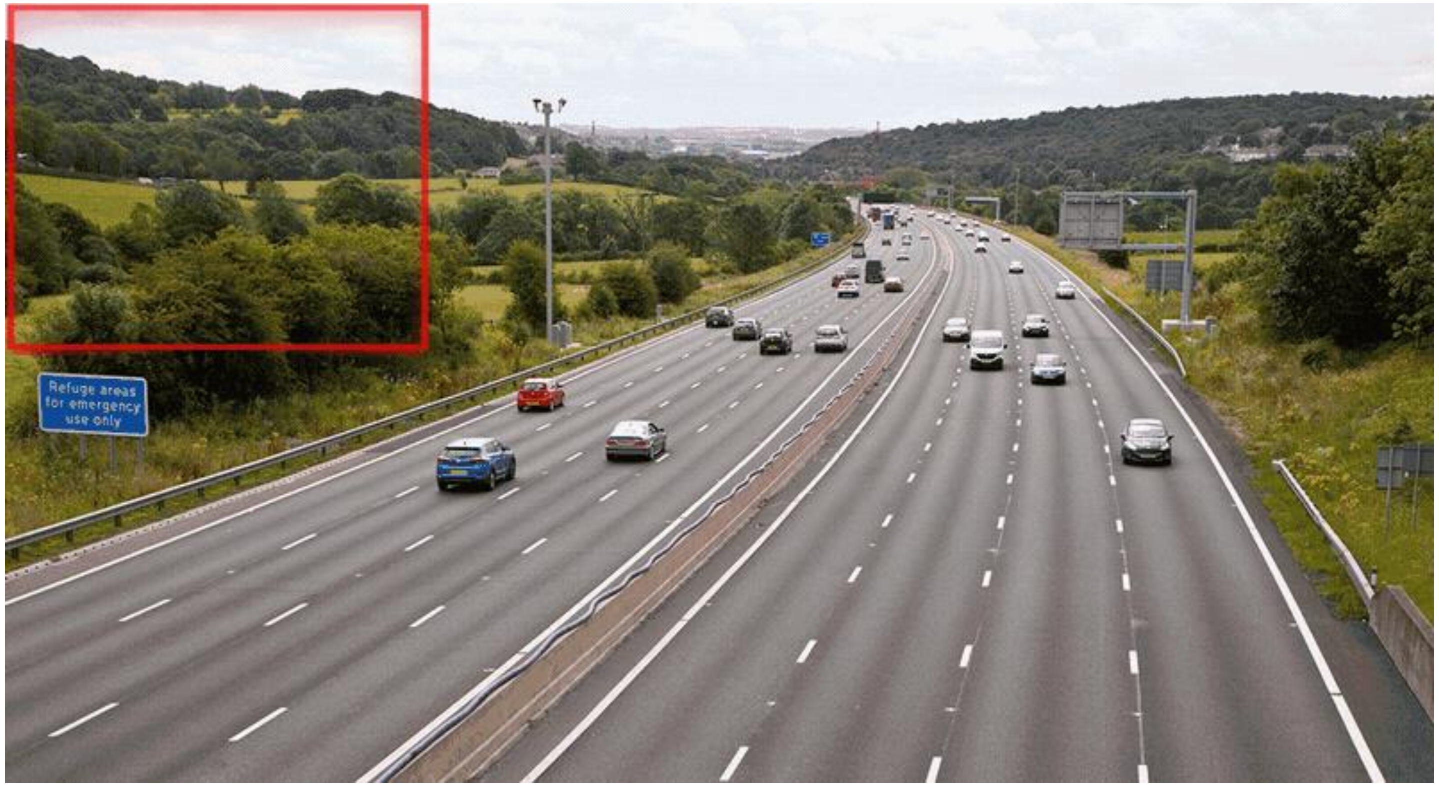


Method 1 – Summary of challenges

- This approach uses models that require a lot of memory, we had limits on the available memory in the cluster.
- Labeling takes time and can be a tedious task, especially with images with many defects.
- Some defects were underrepresented, and the models did not learn enough about how they look.

Method 2 - Approach

- As seen in the detections of Method 1, the models struggled to find small objects.
- To solve this issue, we proposed to use the library SAHI (Slicing Aided Hyper Inference).



Source: Akyon, F. C., Altinuc, S. O. & Temizel, A. Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection. URL: https://github.com/obss/sahi

Method 2 – Workflow

No labels provided

Use SAHI to improve the detection of small objects.

Use YOLOv8 (a type of architecture) **for localizing the objects.**

Perform detections

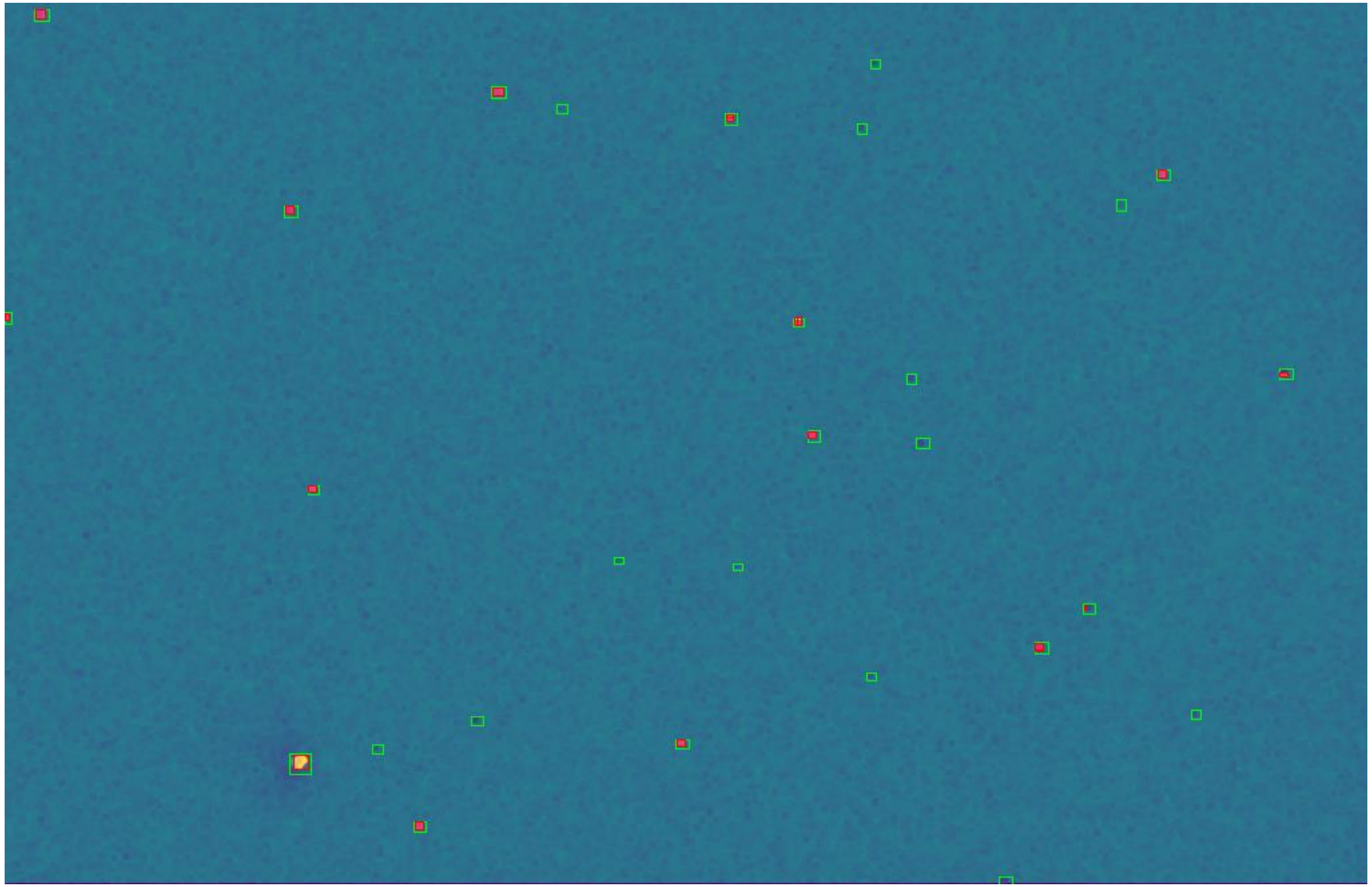
Extract information from the detections of YOLOv8 (i.e. area and colour of the detected object).

Use k-means clustering to classify the detected objects.

Each cluster would correspond to one type of defect, so we would only need to label one random object from each cluster.

Method 2 - Results

Detection results by YOLOv8 + SAHI
Green boxes: ground truth (i.e. labels provided to the model).
Red boxes: detection results.

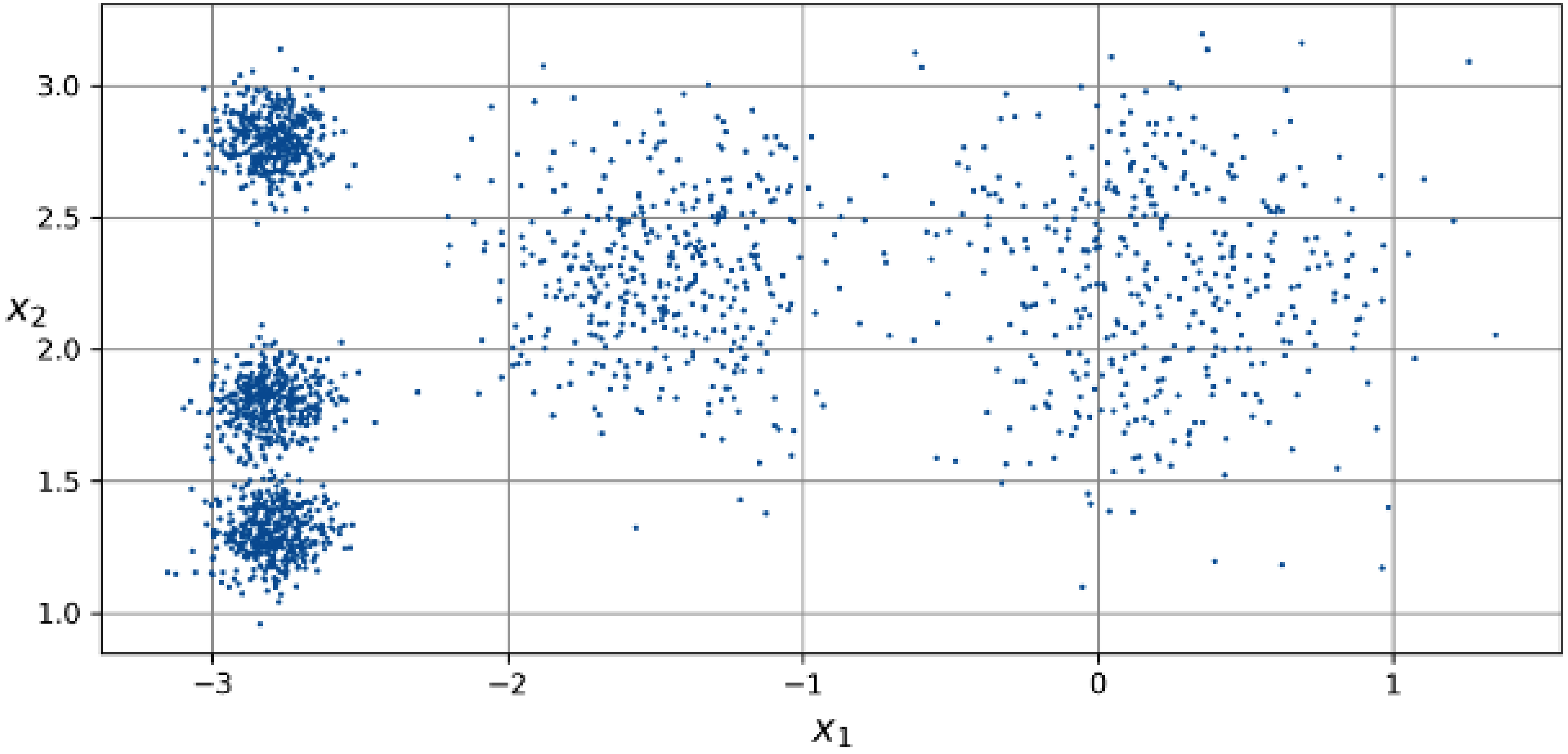


Method 2 - Metrics

Model	Precision of bounding boxes		
	Small	Medium	Large
SAHI+YOLOv8 Big Slices	11%	15%	12%
SAHI+YOLOv8 Small Slices	10%	19%	-*

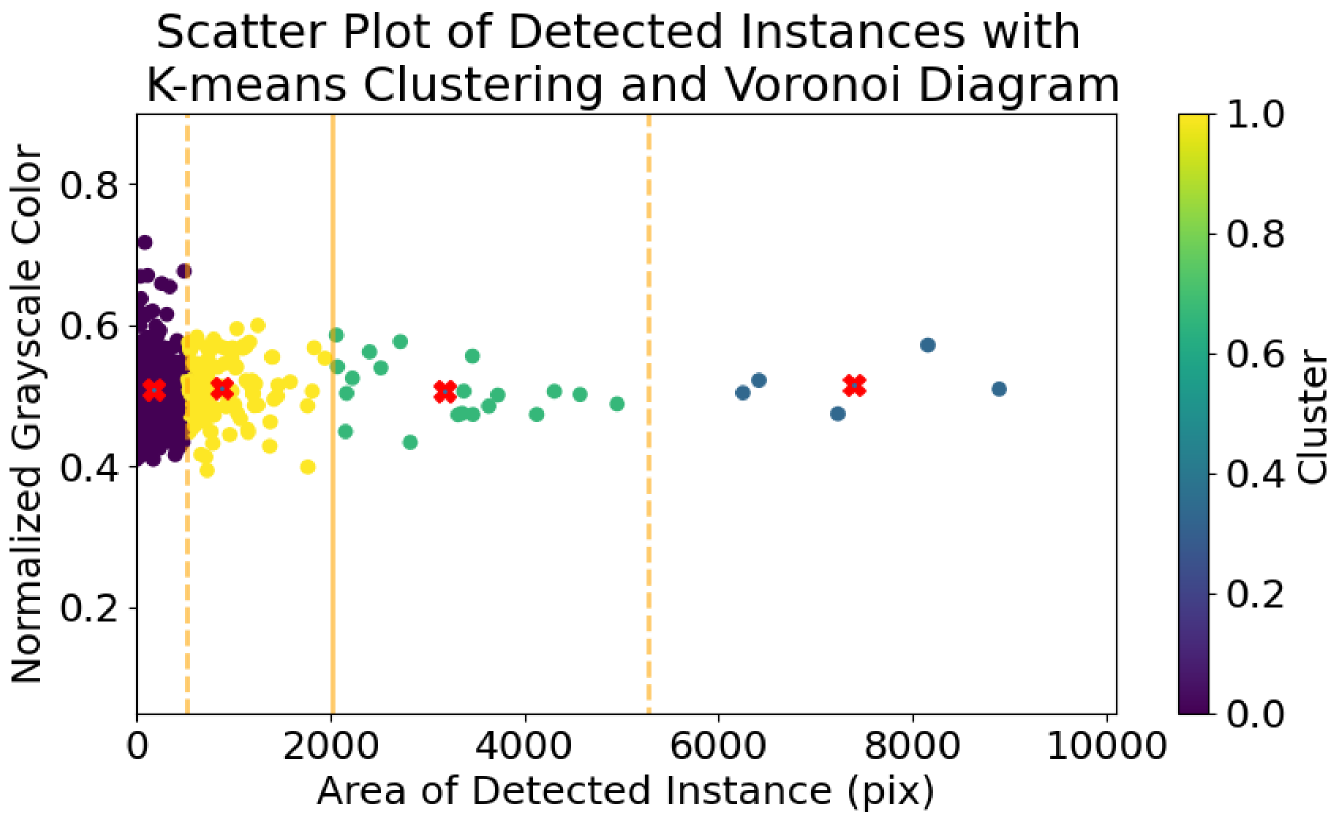
Results of the Precision for predictions with an IoU threshold of 50%.

K-means clustering – expectation



An unlabelled dataset composed of five blobs of instances. Source: Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow ISBN: 978-1-098-12597-4 (O'Reilly Media, Sebastopol, CA, 2023).

K-means clustering - reality



Scatter plot obtained for the detection of the model YOLOv8+SAHI

Conclusions

- Conclusion
- Outlook

Conclusion

- A small dataset (< 100 images) may have enough data for reaching a sufficiently good precision.
- For solving problems with the detection of small objects, libraries like SAHI offer a good solution.
- Object detection models can adapt to detect objects they have not seen before, but the quality is limited. This can be solved with custom training.

Outlook

- Custom training our model on our dataset together with SAHI might be a good solution for better detection and labelling of pinholes (or small objects in general).
- In case labeling is not considered to be used: better parameters for clustering, improvement of the generation of masks or selection of a more appropriate clustering algorithm.
- Use of CutLER (Cut and Learn) with SAHI for completely unsupervised object detection and subsequent classification by the clustering approach.

Appendix: ImageJ Particle Analysis

Particle Analysis

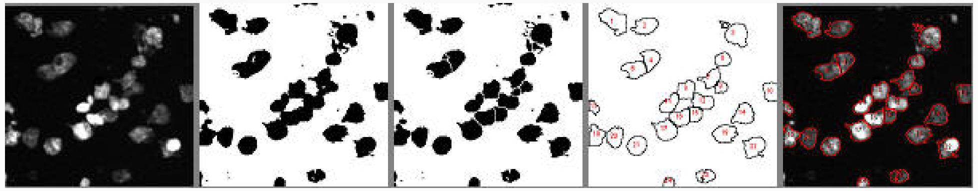
Overview | 3D Rendering | Annotating Images | Chromatic Shift | Colocalization Acquisition | Colocalization Analysis | Color Image Processing | Deconvolution | Information Loss | Image Intensity Processing | Particle Analysis | Principles
Registration | Segmentation | Spatial Calibration | Stack-slice Manipulations | T-functions | Thresholding | Tracking | Visualization | Voxelization | Watershed Separation | Z-functions

The content of this page has not been vetted since shifting away from MediaWiki. If you'd like to help, check out the how to help guide!

Automatic Particle counting

Automatic particle counting can be done if the image does not have too many individual particles touching. Manual particle counting can be done using the Multi-point Tool.

Segmentation, or the ability to distinguish an object from its background, can be a difficult issue to deal with. Once this has been done, however, the object can then be analyzed.

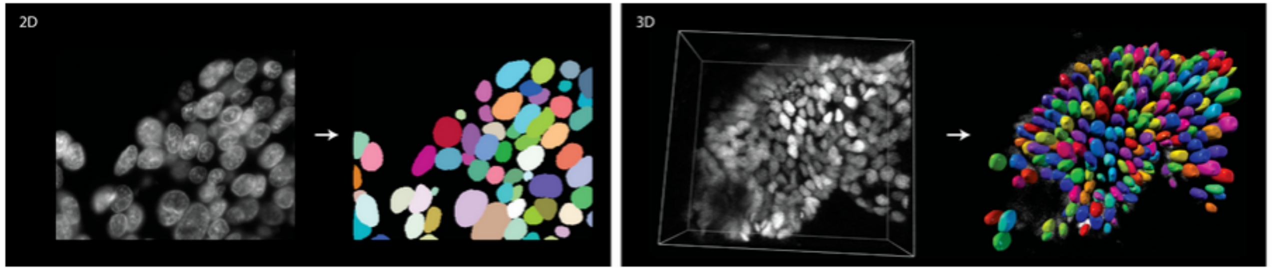


RAW Threshold Watershed “AnalyzeParticles”

ImageJ Particle Analysis website. URL: https://imagej.net/imaging/particle-analysis

Appendix: Stardist

StarDist - Object Detection with Star-convex Shapes



This repository contains the Python implementation of star-convex object detection for 2D and 3D images, as described in the papers:

- Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers.
  Cell Detection with Star-convex Polygons.
  International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Granada, Spain, September 2018.
- Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers.
  Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy.
  The IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, Colorado, March 2020.

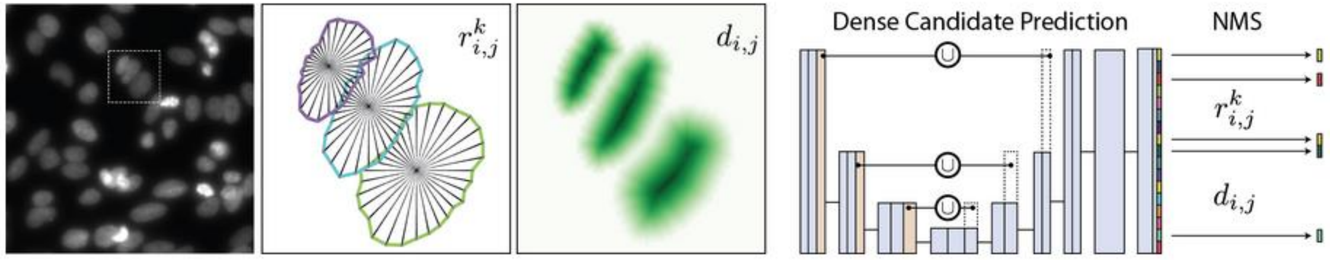
StarDist

Stardist Github page. URL: https://github.com/stardist/stardist

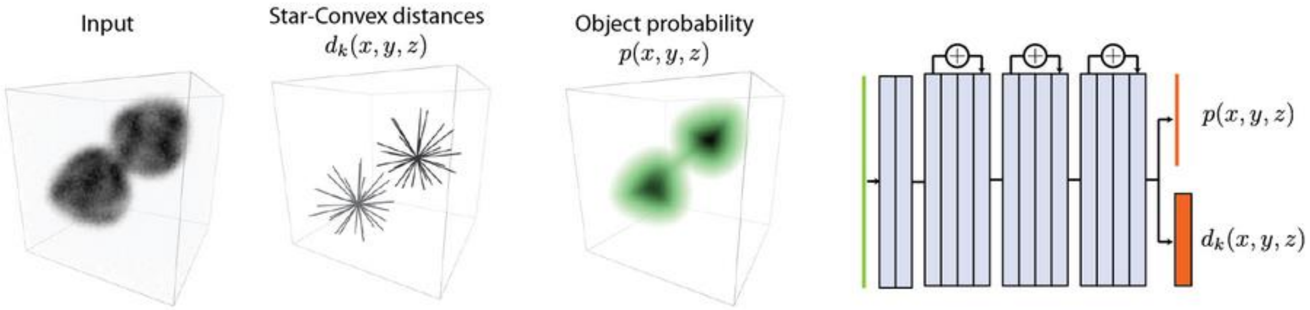
Appendix: Stardist

Overview

The following figure illustrates the general approach for 2D images. The training data consists of corresponding pairs of input (i.e. raw) images and fully annotated label images (i.e. every pixel is labeled with a unique object id or 0 for background). A model is trained to densely predict the distances (r) to the object boundary along a fixed set of rays and object probabilities (d), which together produce an overcomplete set of candidate polygons for a given input image. The final result is obtained via non-maximum suppression (NMS) of these candidates.



The approach for 3D volumes is similar to the one described for 2D, using pairs of input and fully annotated label volumes as training data.



Stardist Github page. URL: https://github.com/stardist/stardist

Thank you for your attention

Acknowledgments

Prof. Dr. André Anders
Dr. Stefan Zahn

IOM

www.iom-leipzig.de

Contact

Natasha Hrycan
Modellierung und Simulation

natashaalexandra.hrycanrobachuk@iom-leipzig.de
natashahrycan.github.io

Leibniz Institute of Surface Engineering
Permoserstr. 15 / 04318 Leipzig / Germany