Inspection visuelle automatique des modules pixel d'ITk avant le wire-bonding

Jerome Bobin, Frederic Deliot, Oussama Eladdachi, Julien Giraud, Jonathan Kern, Vera Maiboroda, Matthias Saimpert

ITk pixel modules

- High-Luminosity LHC upgrade:
  current ATLAS inner tracker → all-silicon ATLAS inner tracker (ITk)
- Innermost ITk part: state-of-the-art pixel detector

More about ITk and pixels:
ATLAS ITk Pixel Detector Overview
Module development for the ATLAS ITk pixel detector

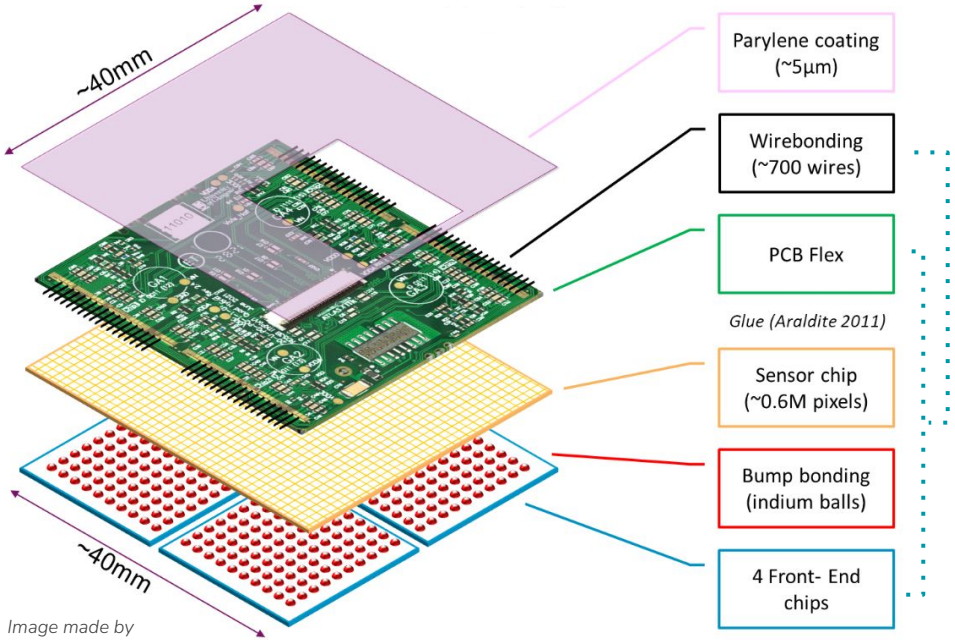


Image made by J. Giraud

Wire-bonding assembling step of ITk pixel modules

During wire-bonding assembling step the copper pads on the flex PCB get connected with front-end chip by ~700 wires.

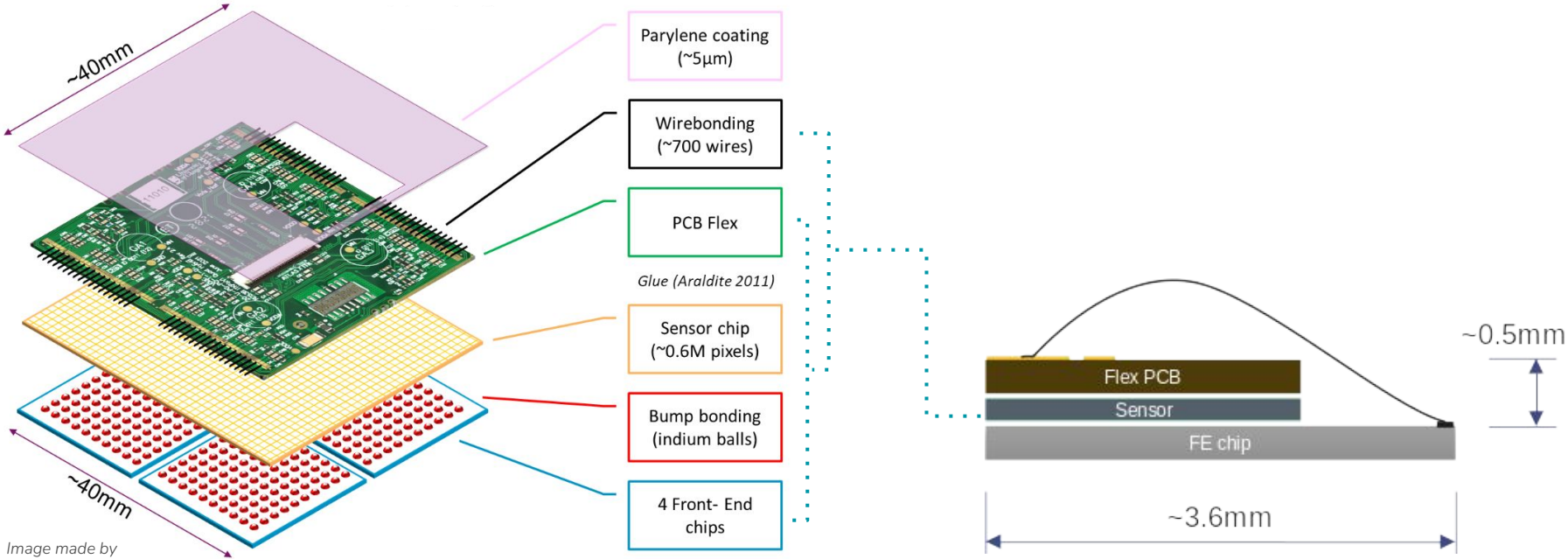


Image made by J. Giraud

Wire-bonding assembling step of ITk pixel modules

During wire-bonding assembling step the copper pads on the flex PCB get connected with front-end chip by ~700 wires.

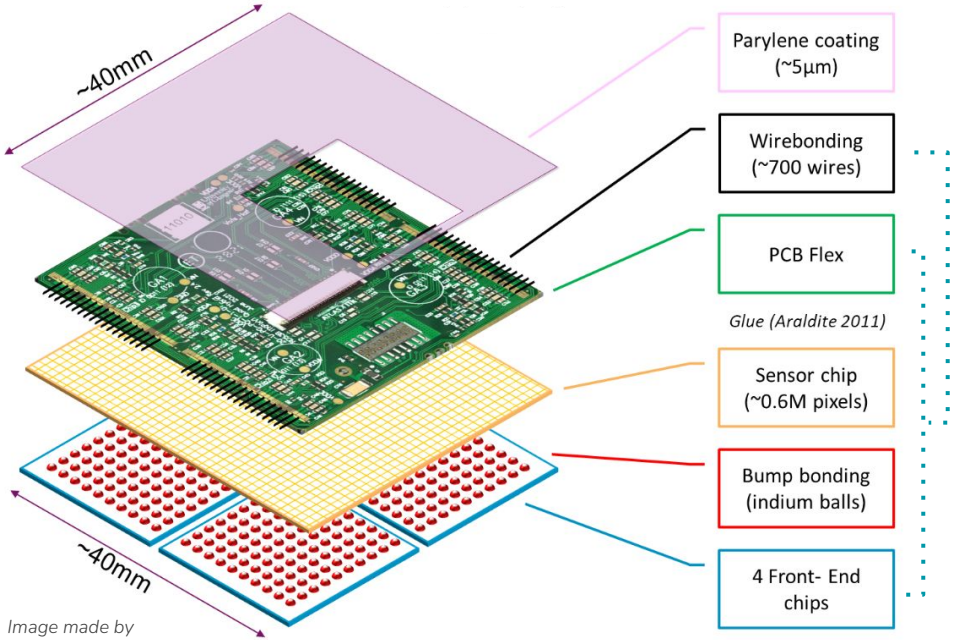
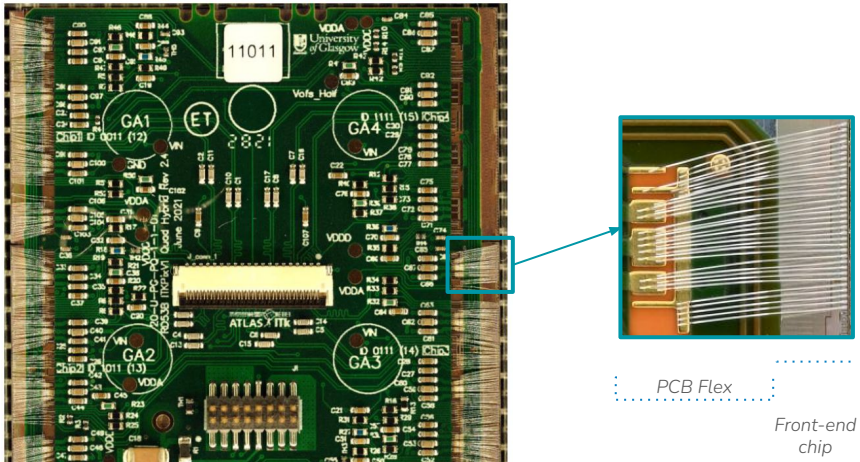
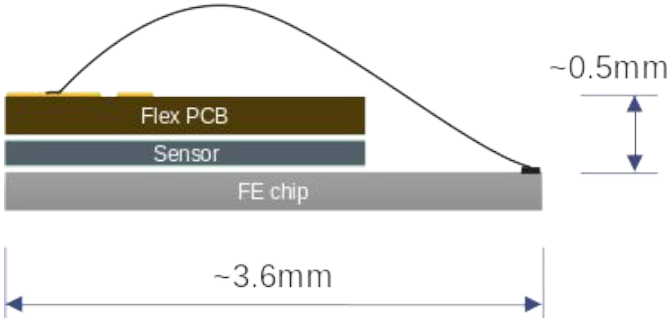


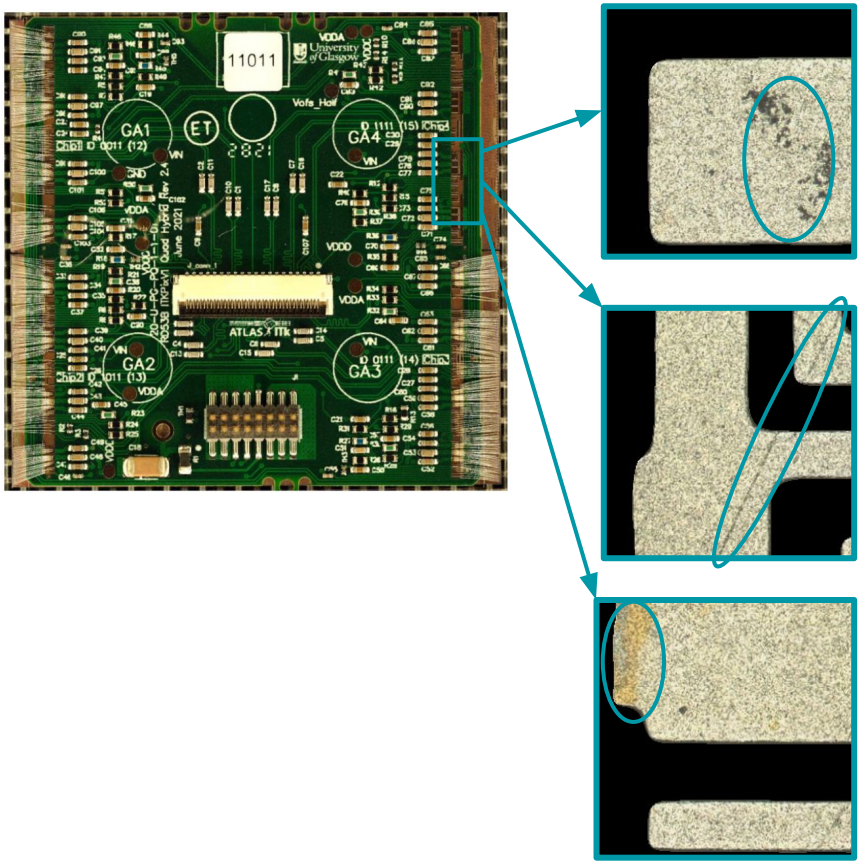
Image made by J. Giraud





Visual inspection before wire-bonding

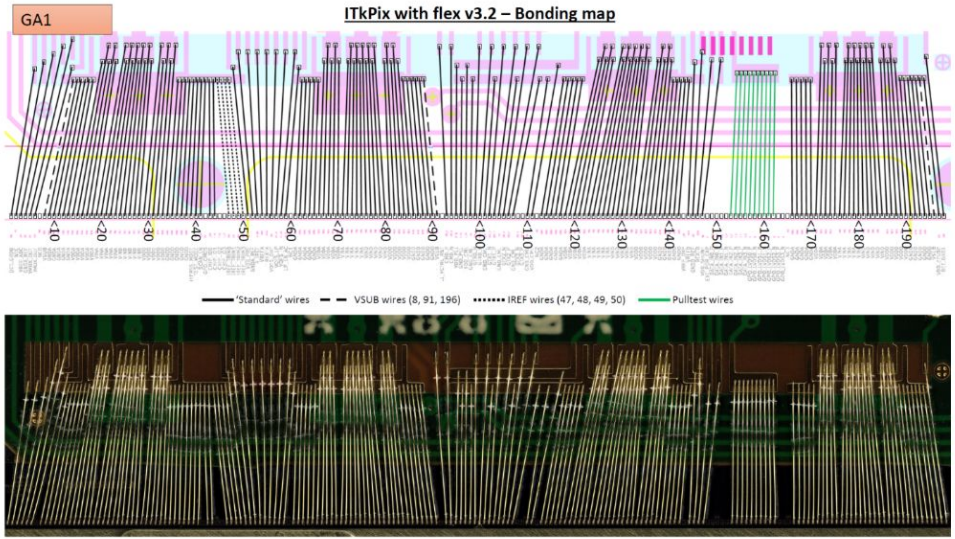
The wire-bonding assembling step requires the knowledge about PCB pads surface quality since certain surface defects could complicate the wires placement.

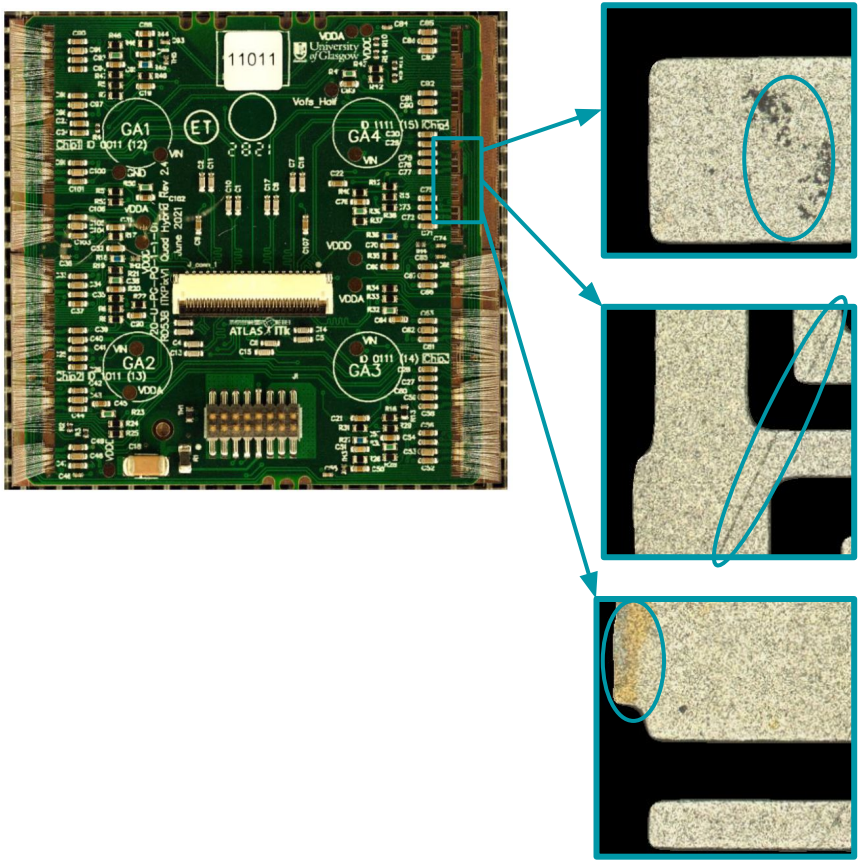


Visual inspection before wire-bonding

The wire-bonding assembling step requires the knowledge about PCB pads surface quality since certain surface defects could complicate the wires placement.

Need to do a visual inspection to provide to the wire-bonding teams an idea about placement of defects on pads surface.





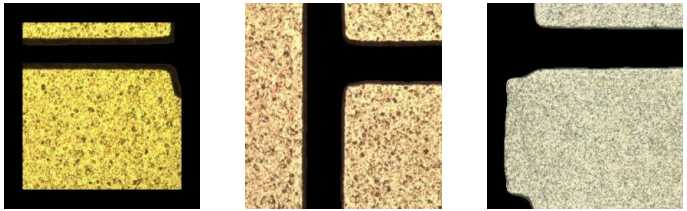
Pixel modules production timeline

- 2021-2022:
  production of about 160 modules based on the RD53A chip
- 2023-2024:
  site qualification & pre-production of about 800 modules
- 2024:
  beginning of module production (~ 12,000 modules)
- 2026:
  module production complete

Pixel modules production timeline

- 2021-2022:
  production of about 160 modules based on the RD53A chip
- 2023-2024:
  site qualification & pre-production of about 800 modules
- 2024:
  beginning of module production (~ 12,000 modules)
- 2026:
  module production complete

Start of the research on
RD53A and pre-production modules
(Vera Maiboroda, Xiang Chen)



few modules, different pad properties

Pixel modules production timeline

- 2021-2022:
  production of about 160 modules based on the RD53A chip
- 2023-2024:
  site qualification & pre-production of about 800 modules
- 2024:
  beginning of module production (~ 12,000 modules)
- 2026:
  module production complete

Research continues on production modules
(Oussama Eladdachi)

many modules, fixed pad properties

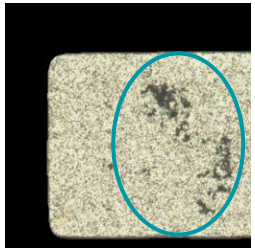
Automatic visual inspection with machine learning

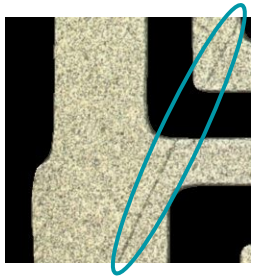
- Segmentation of defects

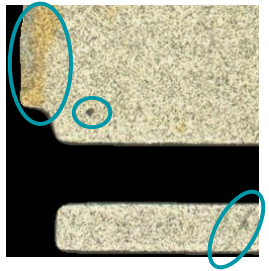
Input: image

Output: mask

wire-bonding map adjustment





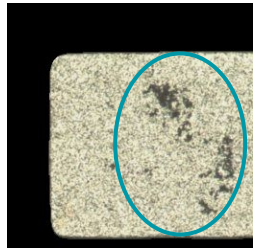


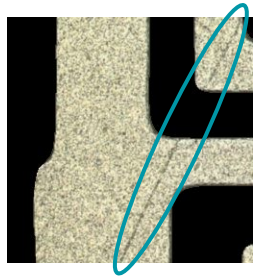
Automatic visual inspection with machine learning

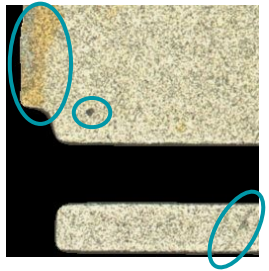
- Segmentation of defects

Input: image *Output: mask*

- Need to have ground truth masks to compare with output masks.

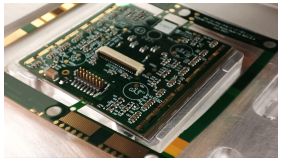


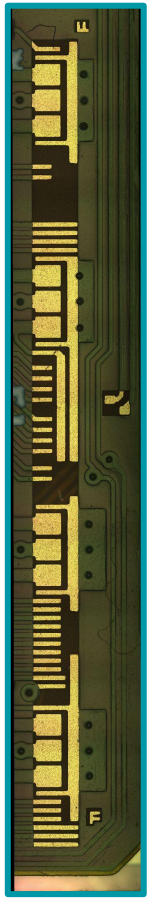


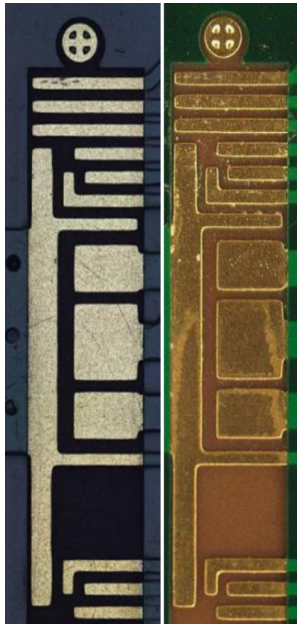


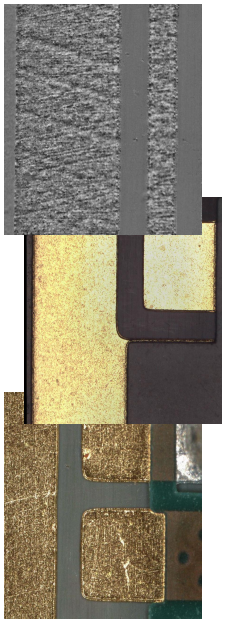
Automatic visual inspection with machine learning: dataset

Microscope Keyence 700-VHX.





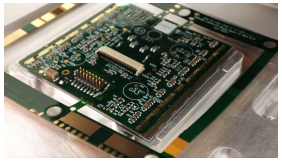


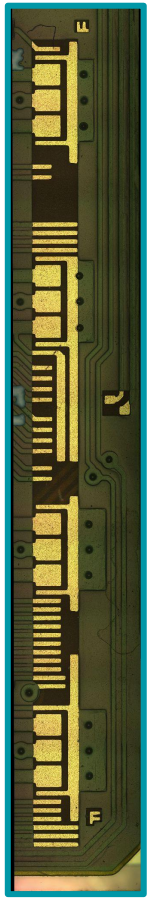


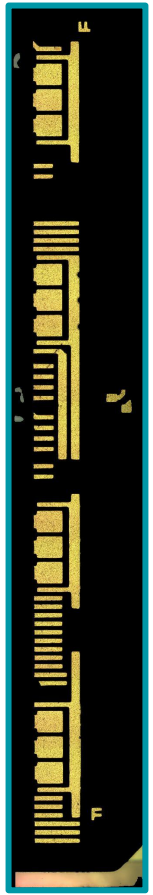
Microscope lightning settings

Automatic visual inspection with machine learning: dataset

Microscope Keyence 700-VHX.



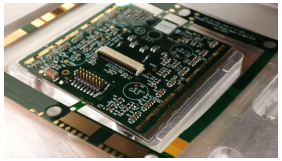


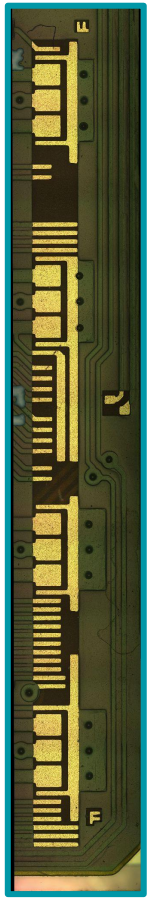


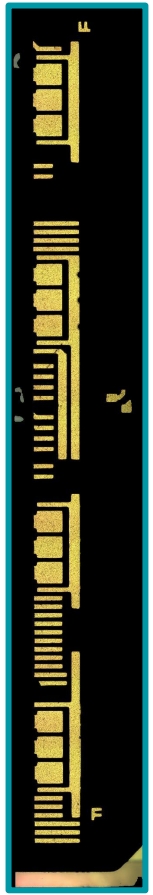
Background removal

Automatic visual inspection with machine learning: dataset

Microscope Keyence 700-VHX.





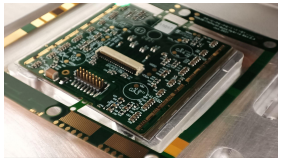


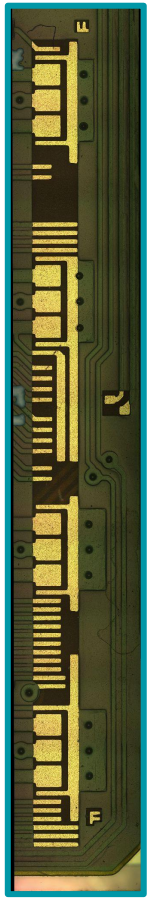
Masks creation:

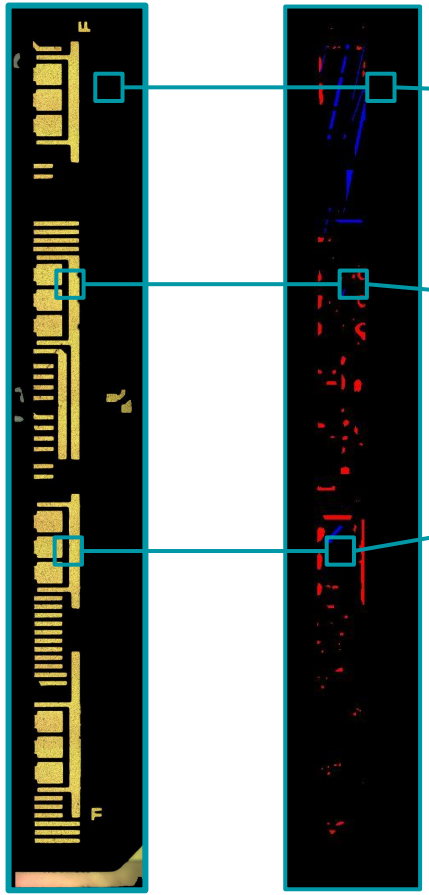
- Manual labelling
- Color coding for defect types

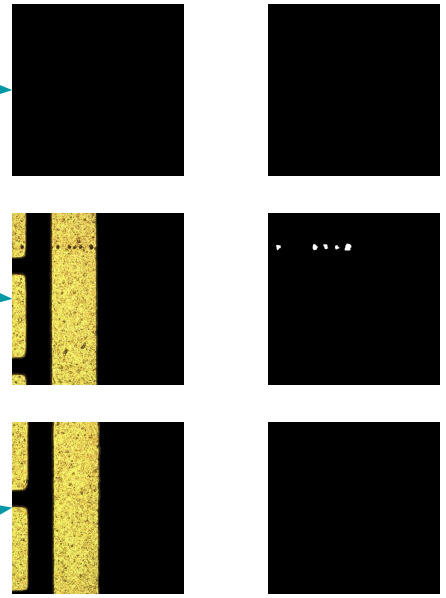
Automatic visual inspection with machine learning: dataset

Microscope Keyence 700-VHX.









Tiles Masks

Dataset

Automatic visual inspection with machine learning: quality metrics

- AUC: general metric
  - Varies from 0.5 to 1
- F1-score: good with imbalance classes
  - Varies from 0 to 1

Automatic visual inspection with machine learning: quality metrics

- AUC: general metric
  Varies from 0.5 to 1
- F1-score: good with imbalance classes
  Varies from 0 to 1

- Image level:

  Checks if image contains a defect

Ground truth

Algorithm output: same high AUC

Algorithm output: low AUC

Automatic visual inspection with machine learning: quality metrics

- AUC: general metric
  Varies from 0.5 to 1
- F1-score: good with imbalance classes
  Varies from 0 to 1

- Image level:

  Checks if image contains a defect

Ground truth

Algorithm output: same high AUC

Algorithm output: low AUC

- Pixel-level:
  Checks if each pixel was correctly classified as anomalous

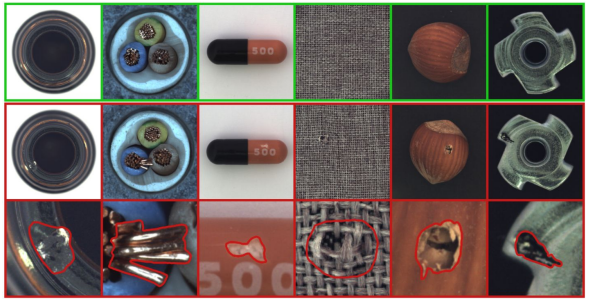
Ground truth

Algorithm output: same high F1-score

Algorithm output: lower F1-score

What performance to expect?

- MVTec is a dataset for anomaly detection benchmarking

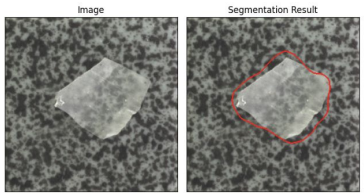


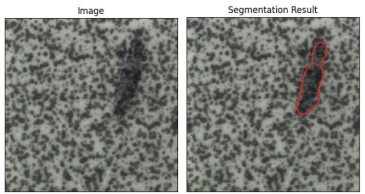
Different object and texture categories

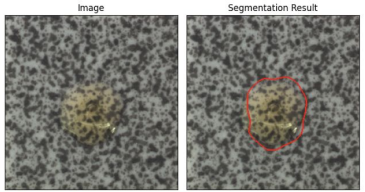
- MVTec contains “tile” subset that looks very similar to pad surface and many models show high metric values on this subset

image_AUROC	1.0
image_F1Score	1.0
pixel_AUROC	0.977867066860199
pixel_F1Score	0.7631763219833374

PatchCore algorithm performance on “tile” MVTec subset

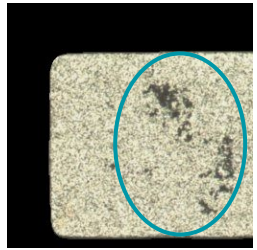


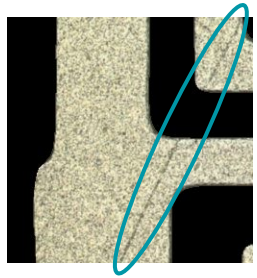


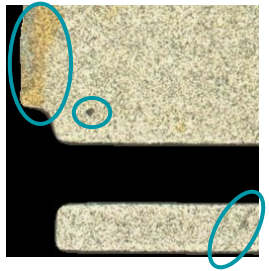


Automatic visual inspection with machine learning: algorithms

- Unsupervised algorithms (defects = anomalies)
  - Learn not-anomalous features, distinguish them from anomalous ones with the help of loss function (CFA, PatchCore)
  - Learn to transform not-anomalous features to normal distribution, anomalous features would be transformed into outliers (CFlow, FastFlow)
  - Learn to reconstruct not-anomalous images as not-anomalous via VAE, so that defects are detected based on difference between input and output images (DRAEM)
- Zero-/few-shot algorithms
  - Segment-any-anomaly (foundation model)
- Supervised algorithms
  - U-net-based

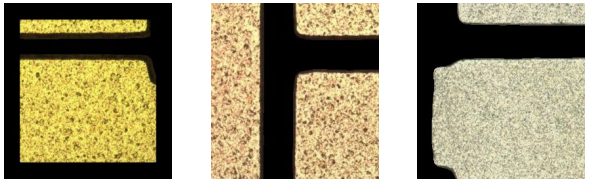




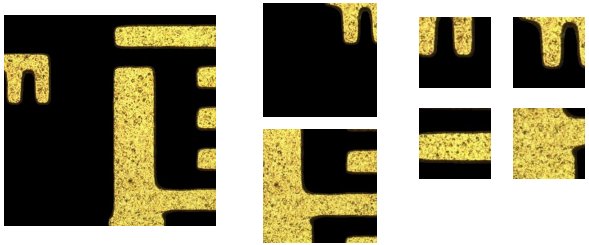


Selecting most promising algorithm

Created different datasets

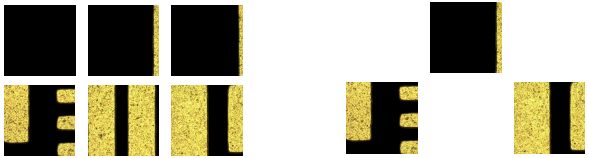


Pads surface variations



500 *256* *128*

Different tile size

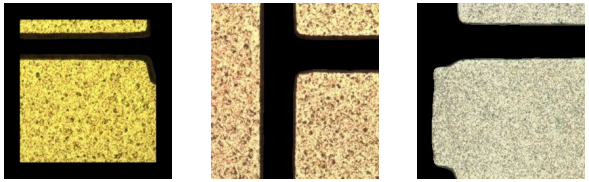


Original *Filtered*

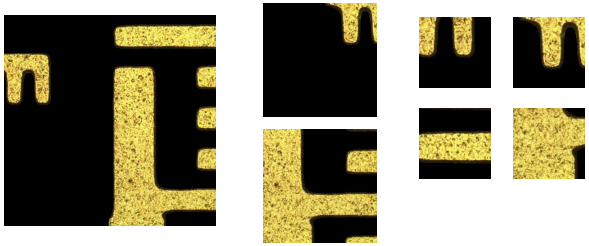
Selectivity level

Selecting most promising algorithm

Created different datasets

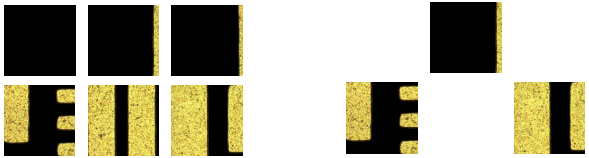


Pads surface variations



500 *256* *128*

Different tile size



Original *Filtered*

Selectivity level

Tested different algorithms on datasets

		CFA	Patchcore	FastFlow	CFlow	DRAEM
		image/pixel	image/pixel	image/pixel	image/pixel	image/pixel
RD53A, 500x500, original	AUROC	0.54/0.77	0.65/0.79		0.68/0.70	0.60/0.72
	F_1	0.90/0.22	0.90/0.13		0.91/0.09	0.90/0.14
RD53A, 500x500, filtered	AUROC	0.52/0.89	0.80/0.89	0.72/0.80	0.68/0.79	0.59/0.72
	F_1	0.78/0.25	0.86/0.24	0.78/0.21	0.77/0.17	0.77/0.10
RD53A, 256x256, original	AUROC	0.54/0.77	0.57/0.76	0.63/0.72	0.53/0.69	0.52/0.79
	F_1	0.90/0.22	0.90/0.21	0.90/0.19	0.90/0.17	0.90/0.26
RD53A, 256x256, filtered	AUROC	0.48/0.79	0.55/0.77	0.67/0.72	0.47/0.72	0.55/0.56
	F_1	0.89/0.22	0.89/0.21	0.89/0.20	0.89/0.19	0.89/0.15
ITkPix_Green, 500x500, filtered	AUROC	0.71/0.88	0.71/0.86	0.55/0.87	0.46/0.82	0.61/0.81
	F_1	0.85/0.11	0.85/0.11	0.84/0.12	0.84/0.08	0.79/0.18
ITkPix_Green, 256x256, filtered	AUROC	0.70/0.91	0.75/0.74	0.45/0.67	0.49/0.63	0.66/0.80
	F_1	0.75/0.39	0.82/0.14	0.78/0.12	0.77/0.10	0.77/0.17
ITkPix_Polar, 256x256, filtered	AUROC	0.83/0.90	0.92/0.86			
	F_1	0.87/0.36	0.92/0.36			

The model best scores at the image-level are highlighted in red and at the pixel-level in blue.

Selecting most promising algorithm: PatchCore

Created different datasets

Pads surface variations

500 256 128

Different tile size

Original Filtered

Selectivity level

Tested different algorithms on datasets

		CFA	Patchcore	FastFlow	CFlow	DRAEM
		image/pixel	image/pixel	image/pixel	image/pixel	image/pixel
RD53A, 500x500, original	AUROC	0.54/0.77	0.65/0.79		0.68/0.70	0.60/0.72
	F_1	0.90/0.22	0.90/0.13		0.91/0.09	0.90/0.14
RD53A, 500x500, filtered	AUROC	0.52/0.89	0.80/0.89	0.72/0.80	0.68/0.79	0.59/0.72
	F_1	0.78/0.25	0.86/0.24	0.78/0.21	0.77/0.17	0.77/0.10
RD53A, 256x256, original	AUROC	0.54/0.77	0.57/0.76	0.63/0.72	0.53/0.69	0.52/0.79
	F_1	0.90/0.22	0.90/0.21	0.90/0.19	0.90/0.17	0.90/0.26
ITkPix_Green, 500x500, filtered	AUROC	0.71/0.88	0.71/0.86	0.55/0.87	0.46/0.82	0.61/0.81
	F_1	0.85/0.11	0.85/0.11	0.84/0.12	0.84/0.08	0.79/0.18
ITkPix_Green, 256x256, filtered	AUROC	0.70/0.91	0.75/0.74	0.45/0.67	0.49/0.63	0.66/0.80
	F_1	0.75/0.39	0.82/0.14	0.78/0.12	0.77/0.10	0.77/0.17
ITkPix_Polar, 256x256, filtered	AUROC	0.83/0.90	0.92/0.86			
	F_1	0.87/0.36	0.92/0.36			

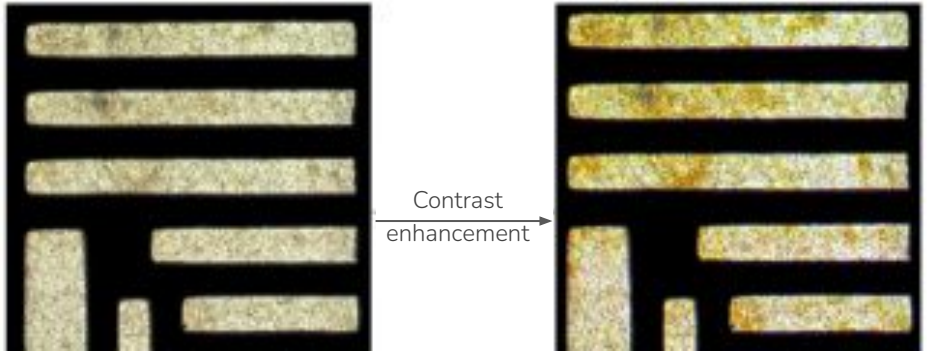
The model best scores at the image-level are highlighted in red and at the pixel-level in blue.

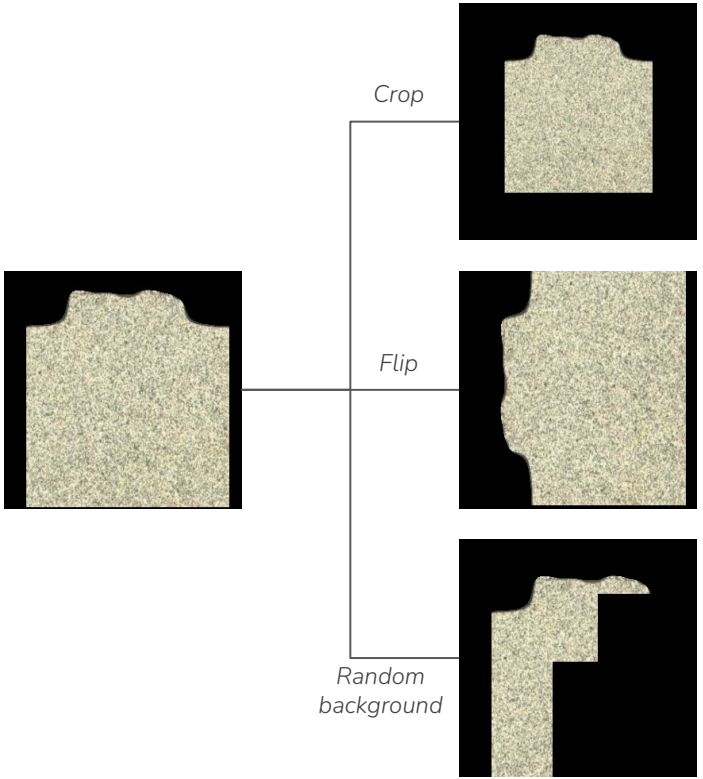
PatchCore consistently showed good performance on different datasets.
This algorithm was chosen for the following studies.

Basic dataset preprocessing for PatchCore

- The tile size was selected to be 256px due to a balance between image-/pixel-level performance and memory consumption.
- Training on filtered images leads to better performance.
- Augmentations such as spatial (crop and flip) and colour (contrast enhancement) can be used to increase the training dataset (up to a certain amount)

The best achieved performance is still not comparable with the one achieved by Partchcore on the MVTec dataset.



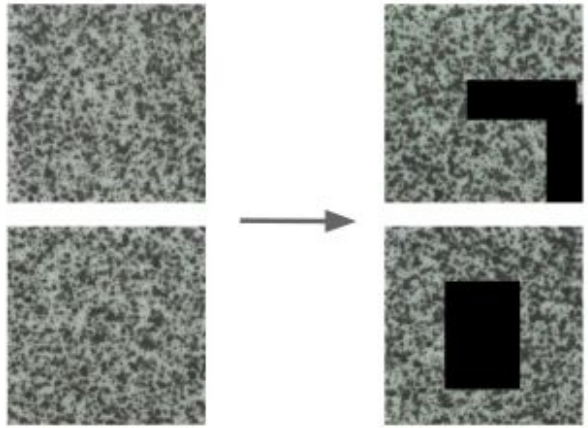


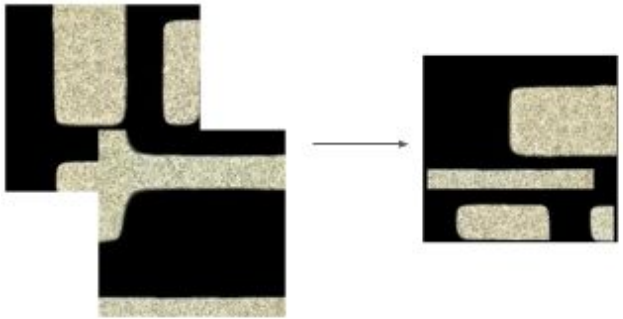
Understanding the difference between MVTec and our datasets

We can artificially introduce complexity coming from pad shapes to the MVTec tile dataset to check if the difference in performance is caused by non-uniform pad shapes.

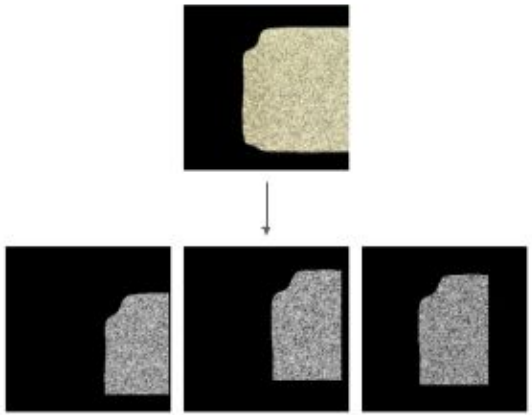
Augmentation by randomly combining pad parts into an image doesn't significantly help. This augmentation doesn't bring new information about pads surface but augments the diversity in pad shapes.

The checks show that the difference in performance doesn't come from the pad shape.





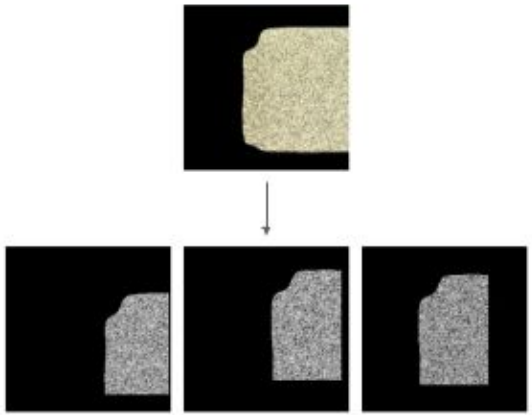
Basic dataset preprocessing for PatchCore



The dataset can be augmented by generating noise based on the wavelet decomposition.

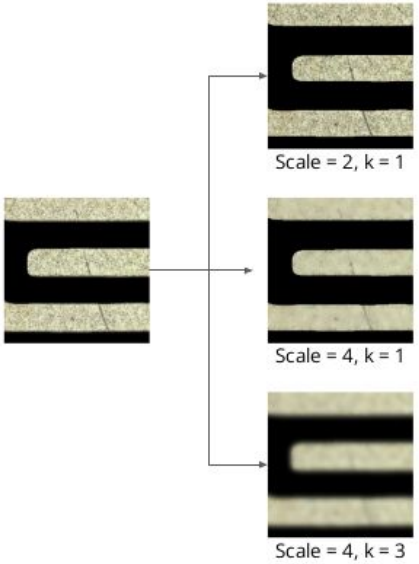
The synthetic texture doesn’t catch all the original texture properties.

Basic dataset preprocessing for PatchCore



The dataset can be augmented by generating noise based on the wavelet decomposition.

The synthetic texture doesn't catch all the original texture properties.



Samples can be denoised via the techniques based on wavelet decomposition.

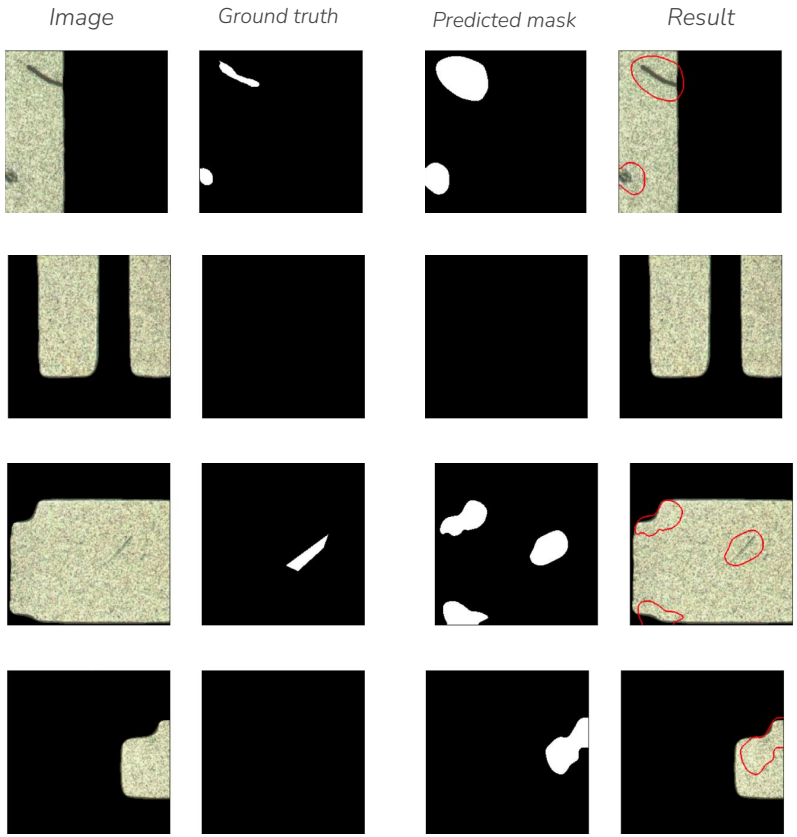
Performance on denoised images is slightly improved (1-2%) in comparison with not denoised ones.

Adjustable solution

The Patchcore model trained on the latest ITK dataset shows the best results with tile size 256px, trained on filtered images, with spatial augmentations. Denoising was dropped due to time-consumption.

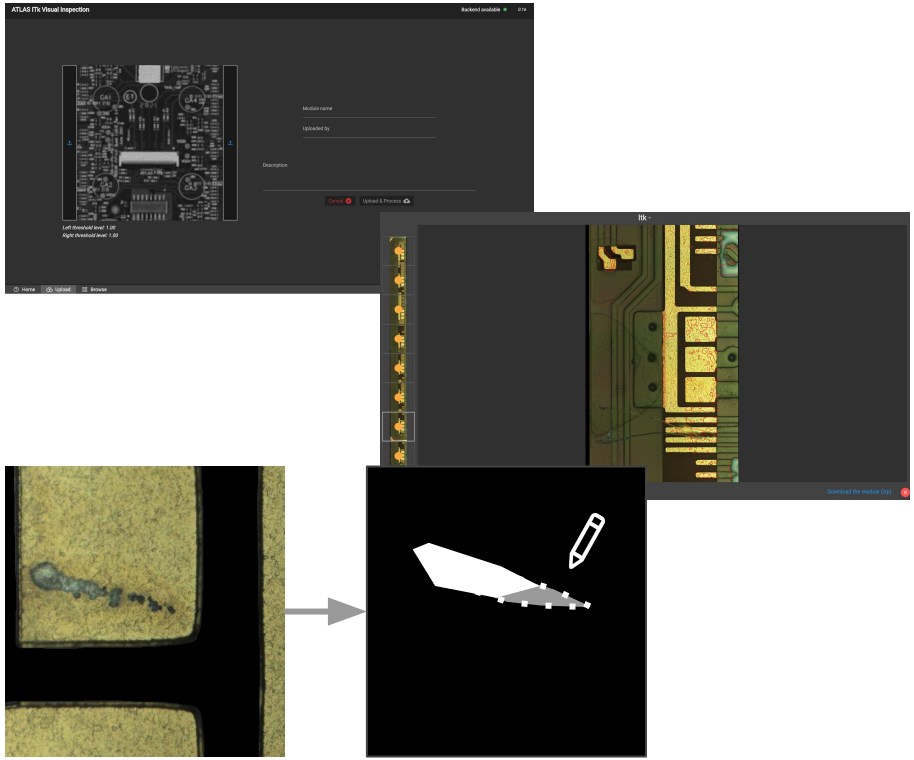
With spatial augmentations we can get similar performance using 10 training tiles to train instead of 167.

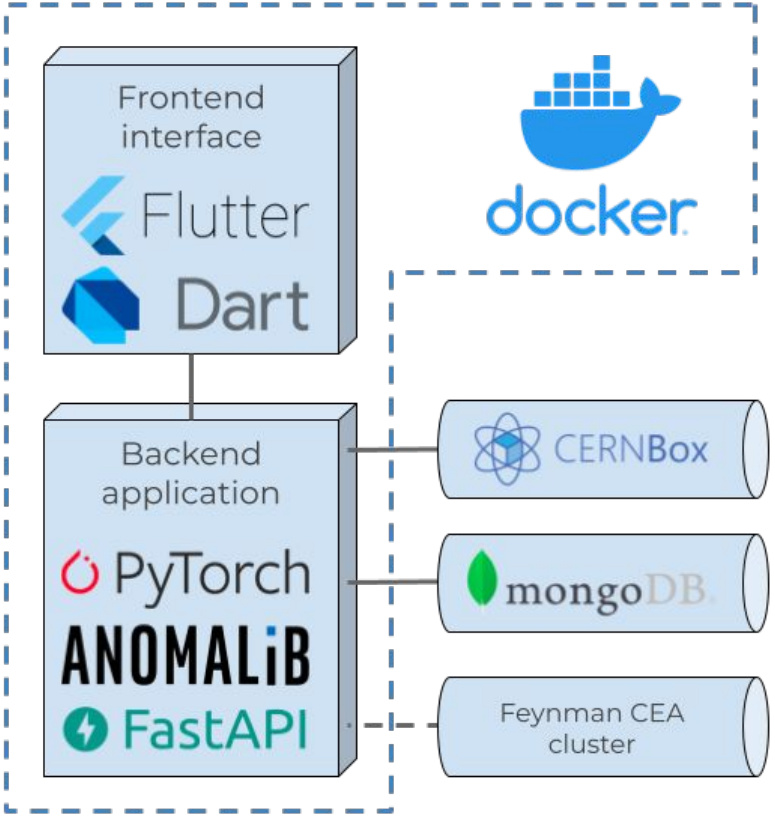
Metric	167 images	10 images + augmentations
AUROC image -level	0.93	0.90
F1 image -level	0.92	0.91
AUROC pixel -level	0.86	0.84
F1 pixel -level	0.37	0.34



Graphical user interface

GUI developed in form of a web application connected with a local database and a cluster to run the anomaly detection algorithm.

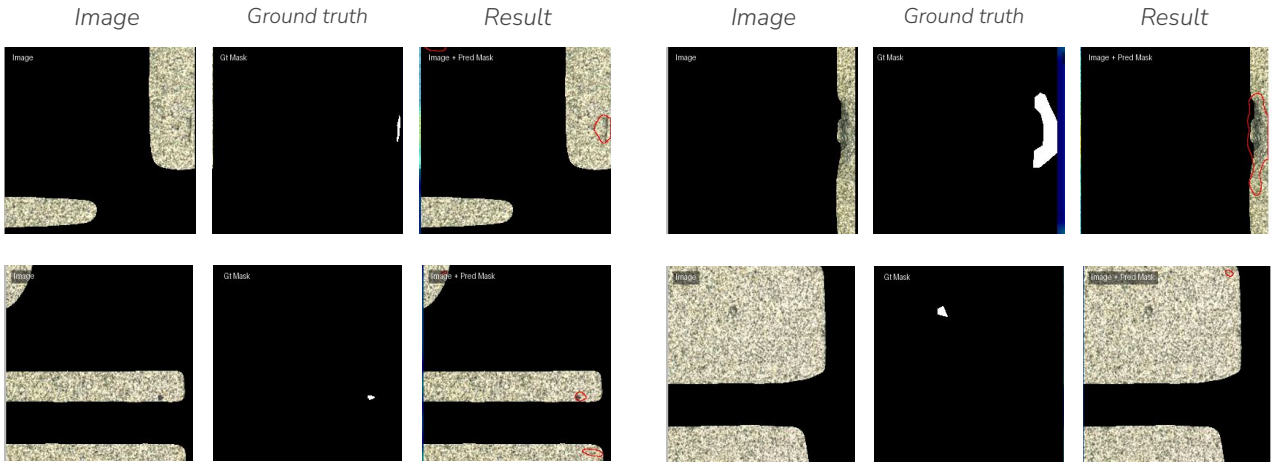




Summary

- Found an algorithm that is able to perform anomaly segmentation (yet needs improvement)
  - Multiple models tested
  - Data preprocessing optimization
- Created a pipeline from data-taking to GUI
- Currently Oussama (6-months internship) uses production modules pictures.
  He continues to optimize PatchCore performance and checks on newest foundation models

AUROC image -level	0.68
F1 image -level	0.71
AUROC pixel -level	0.95
F1 pixel -level	0.28



Thanks! Questions?