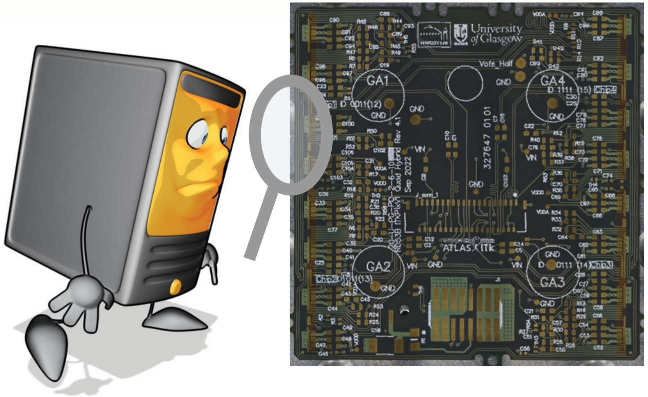
Development of Anomaly Detection techniques applied to the building and Quality Control of ATLAS new silicon tracking detector



Presenter :

VASLIN Louis (KEK / QUP)
ヴァラン　ルイ

KEK

wpi World Premier International Research Center Initiative

QUP International Center for Quantum-field Measurement System for Studies of the Universe and Particles

ATLAS EXPERIMENT

Context

- Towards High-Luminosity LHC

  More luminosity, *more statistics*, **more challenges**

  Upgrade of the **ATLAS detector**

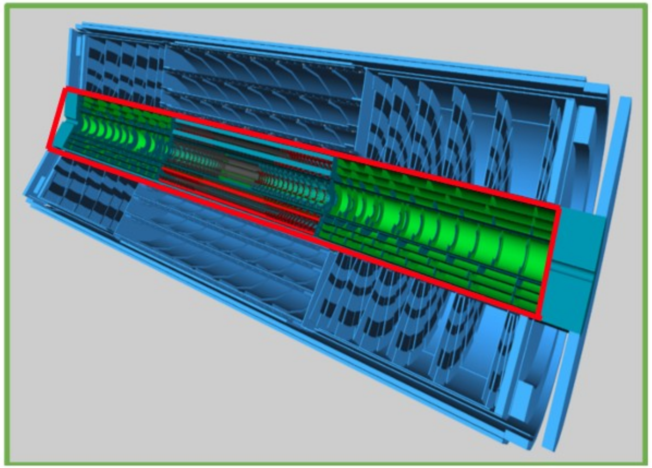
- New ATLAS central detector

  Full silicon tracking detector (ITk)

  ~2800 pixel modules produced in Japan

  Mass production will start in 2024

  => All modules must be checked before commissioning

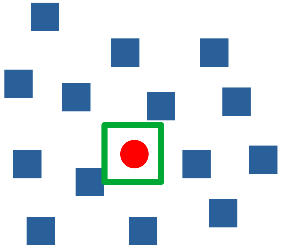


Anomaly Detection

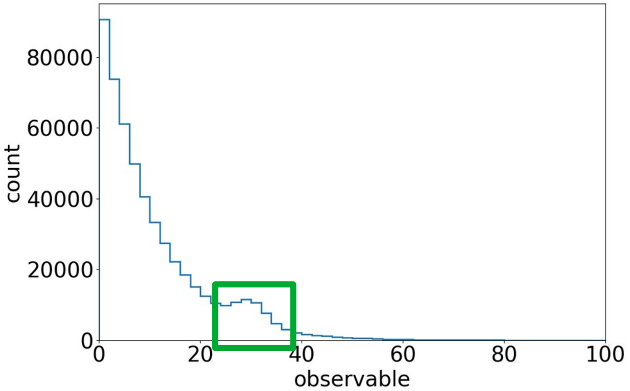
- Principle

Look for things that differ from the norm

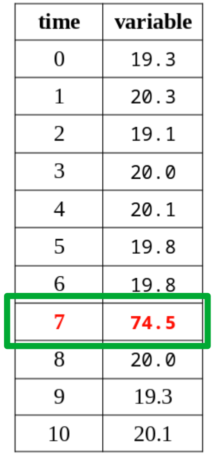
Many applications



feature outlier



overdensity



time	variable
0	19.3
1	20.3
2	19.1
3	20.0
4	20.1
5	19.8
6	19.8
7	74.5
8	20.0
9	19.3
10	20.1

monitoring anomaly

Commonly used in *many fields* …

Including HEP :

Generic BSM searches

Application to detector building and Quality Control ?

Visual Inspection

- Principle

  Look for **visible defects** on detector components

  Major part of Quality Control procedures

- Pixel production in Japan

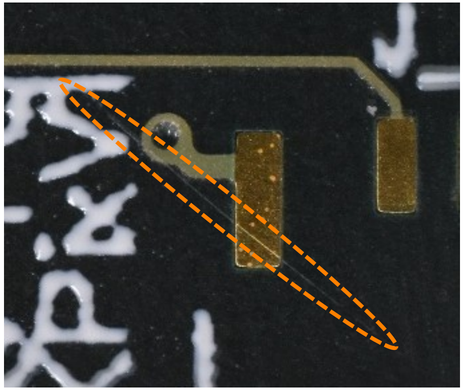
  Visual Inspection usually performed *“by eye”*

  => Slow and error prone

  Our Objective :

  Develop a method for ML assisted Visual Inspection

  => Fast and reliable



example of defect

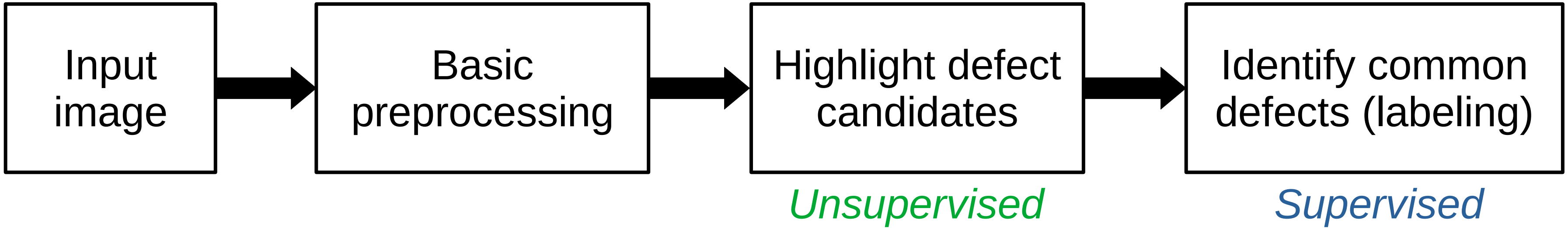
Visual Inspection

- Strategy

Requirements :

- Fast inference (< 1 min)
- Easy to integrate in production workflow
- Generalizable

Proposed workflow



Input image → Basic preprocessing → Highlight defect candidates → Identify common defects (labeling)

Unsupervised

Supervised

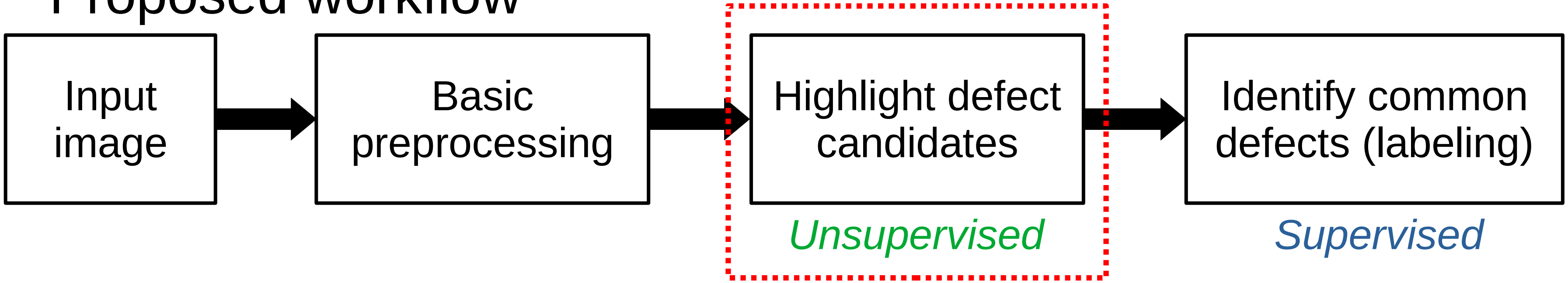
Visual Inspection

- Global strategy

Requirements :

- Fast inference (< 1 min)
- Easy to integrate in production workflow
- Generalizable

Proposed workflow


Input image
Basic preprocessing
Highlight defect candidates
Identify common defects (labeling)
Unsupervised
Supervised

Unsupervised defect detection

- Objective

Detect any **rare defect candidates**

=> Model Independent

=> No labels needed (require less statistics)

- Strategy

Computer Vision model ➡ Select anomalous area ➡ Isolate defect candidates

Computer Vision model:

- Deep CNN with Auto-Encoder structure
- Evaluate reconstruction error per pixels

Select anomalous area:

- Define a selection threshold
- Select pixels above threshold

Isolate defect candidates:

- Cluster anomalous area using DBSCAN
- Filter noisy and small area

Deep Auto-Encoder Model

- Implementation

Custom model with pytorch

Encoding block:

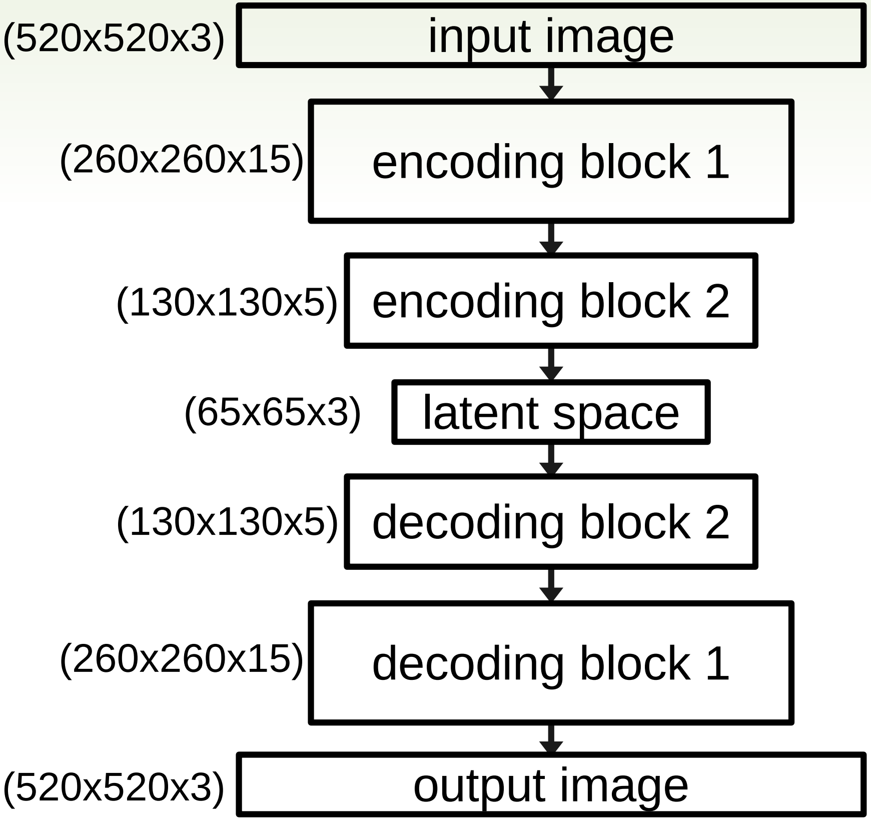
2 convolution layers with Leaky ReLU
10% dropout (second layer)

Decoding block:

Mirroring encoding

Loss function

Average MSE as reconstruction error


(520x520x3) input image

(260x260x15) encoding block 1

(130x130x5) encoding block 2

(65x65x3) latent space

(130x130x5) decoding block 2

(260x260x15) decoding block 1

(520x520x3) output image

Data and training

- Data

  Image taken with microscope

  Use FLEX PCB as test component

  Data augmentation

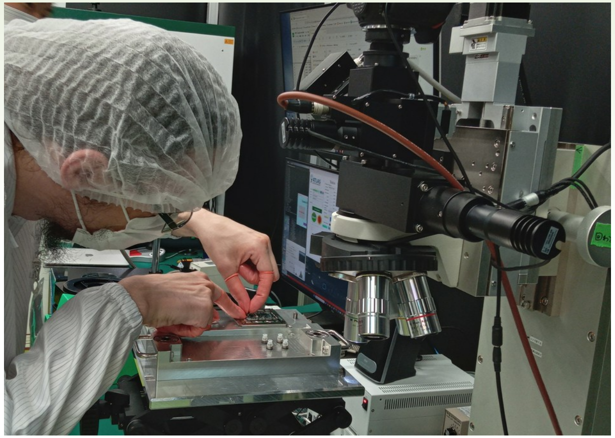
  luminosity, contrast and position variation

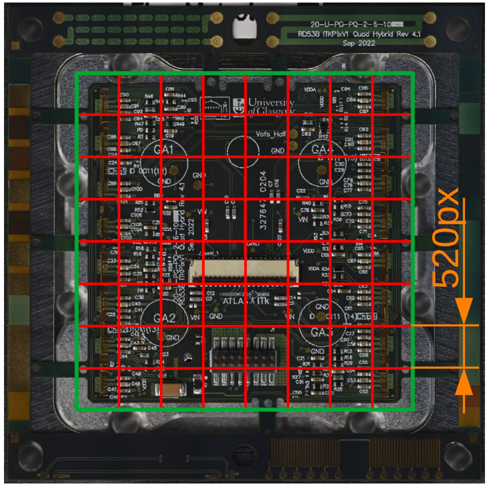
  Split image in 8x8 tiles (520x520px)

- Training

  150 epochs

  ~24000 image tiles



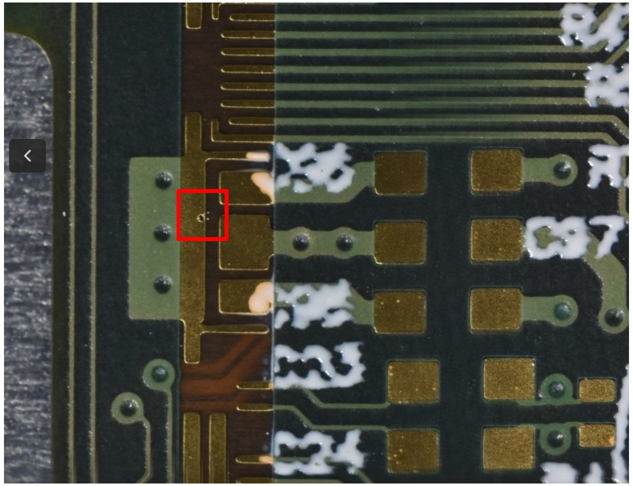


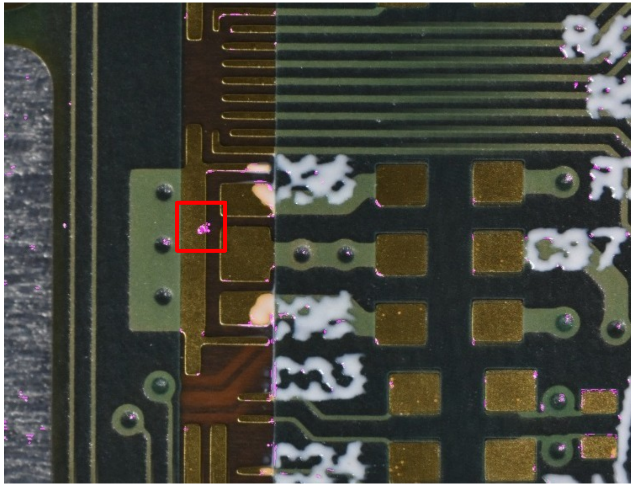
Results

- Rare defect detection example

Choose image where a rare defect was found

=> Never seen in training images





Pink clusters represents defect candidates

New defect is identified

Other defects also identified (dust, ink leak, …)

Inference time : **< 10s**
(on CPU only)

Supervised defect classification

- Model

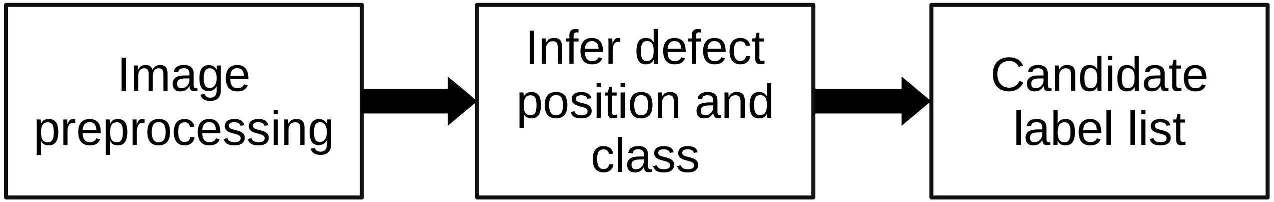
Based on **Detectron2 algorithm** (link)

=> Object segmentation and classification

Give labels for **common defects**

Supervised **=>** Needs more statistics for each class

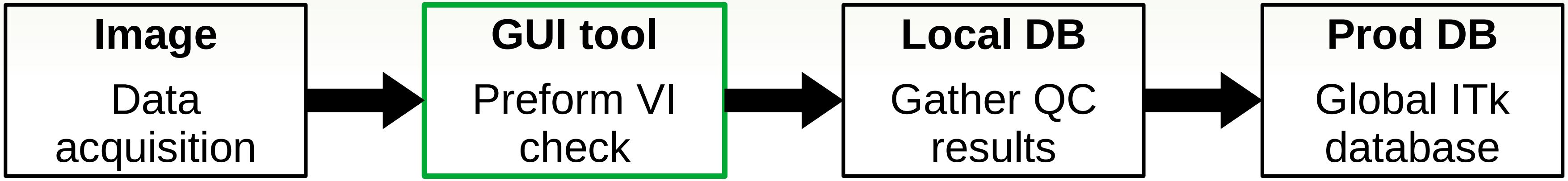
- Strategy



Same as used for unsupervised method

Integration

- Quality Control workflow



Image

Data acquisition

GUI tool

Preform VI check

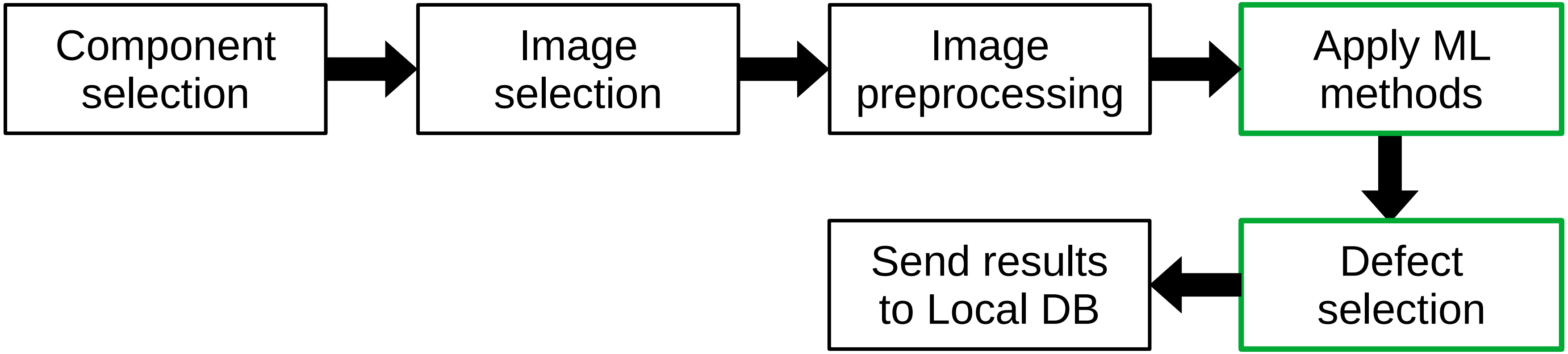
Local DB

Gather QC results

Prod DB

Global ITk database

- GUI tool workflow



Component selection

Image selection

Image preprocessing

Apply ML methods

Defect selection

Send results to Local DB

Integration

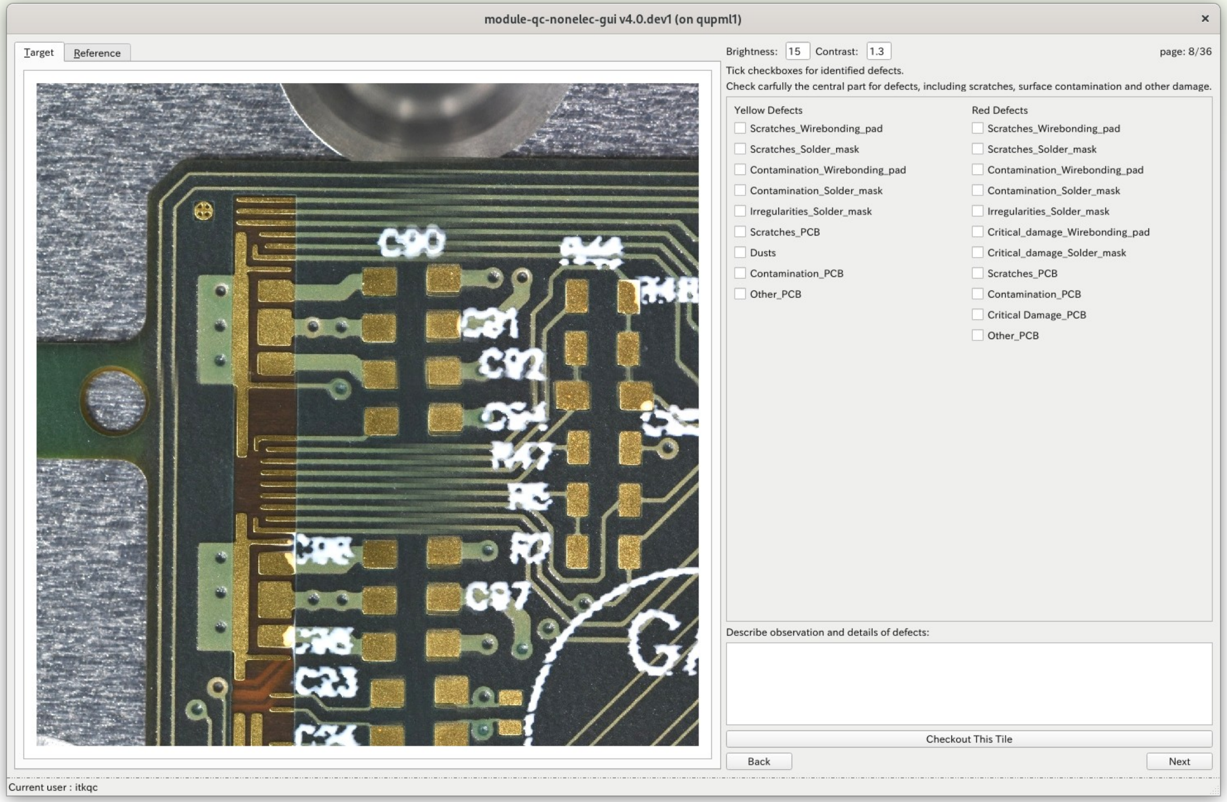
- Screen example

Checkbox list corresponds to possible defects

Visual Inspection performed by **selecting checkboxes** for each image tile

Unsupervised defect detection
=> Add cluster on the image

Supervised defect classification
=> Preselect some checkboxes



Brightness: 15 Contrast: 1.3 page: 8/36

Tick checkboxes for identified defects.

Check carfully the central part for defects, including scratches, surface contamination and other damage.

Yellow Defects

- □ Scratches_Wirebonding_pad
- □ Scratches_Solder_mask
- □ Contamination_Wirebonding_pad
- □ Contamination_Solder_mask
- □ Irregularities_Solder_mask
- □ Scratches_PCB
- □ Dusts
- □ Contamination_PCB
- □ Other_PCB

Red Defects

- □ Scratches_Wirebonding_pad
- □ Scratches_Solder_mask
- □ Contamination_Wirebonding_pad
- □ Contamination_Solder_mask
- □ Irregularities_Solder_mask
- □ Critical_damage_Wirebonding_pad
- □ Critical_damage_Solder_mask
- □ Scratches_PCB
- □ Contamination_PCB
- □ Critical Damage_PCB
- □ Other_PCB

Describe observation and details of defects:

Current user : itkqc

Conclusion

- ML-based Visual Inspection

  Use ML to assist Visual Inspection of detector components
  Improve **efficiency** and **reliability** of Quality Control procedure
  Application to mass production of ATLAS ITk module

- Unsupervised defect detection is working

  Good efficiency for rare/new defects
  Fast inference and **fully integrated** into GUI tool workflow

- Next steps

  Optimize and integrate the **supervised model**

Thank you !

ありがとうございます！