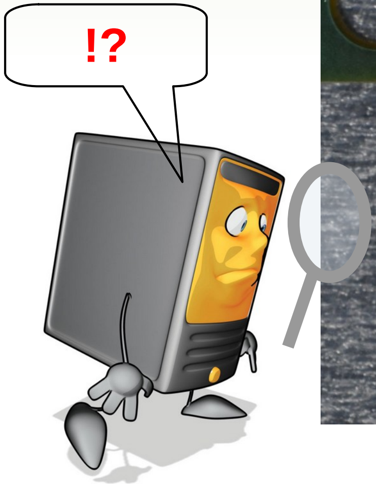
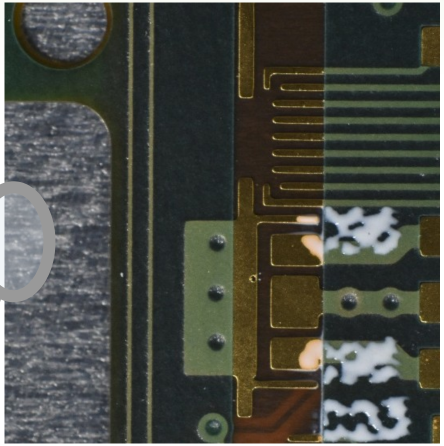
Improving the Visual Inspection of new detector components with Anomaly Detection algorithms





Presenter :

Louis VASLIN

KEK

wpi World Premier International Research Center Initiative

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

Quality Control

- New Detectors for New Physics

  Good detector quality is required to reach our goal

  => Quality Control (QC) during detector production

  Need bigger and more complex detectors

- Visual Inspection of detector components

  Look for visible defects on detector components

  => Prevent future failures

  => Improve fabrication process

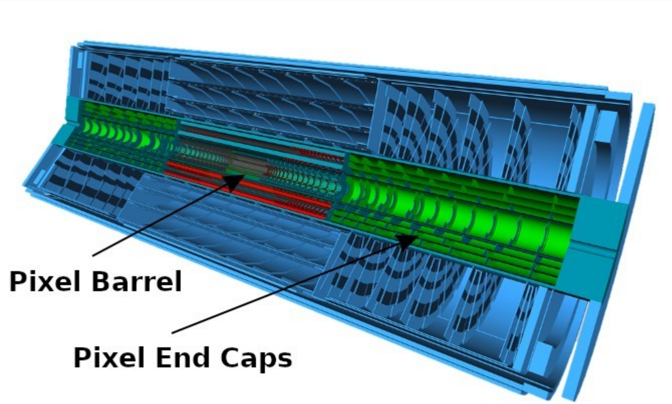
  Time consuming and error prone process

ITk pixel modules

- New Inner Tracker (ITk) for ATLAS detector

  Part of the High Luminosity upgrade of **ATLAS experiment**

  8372 Pixel modules



- Pixel module production in Japan

  2800 pixel modules to be assembled and delivered

  => Main production as already started

  Major challenge for QC

  => New tools are needed to improve QC procedures

ITk pixel modules

- Assembly and QC stages

Sensor tile | **FE chip (x4)**

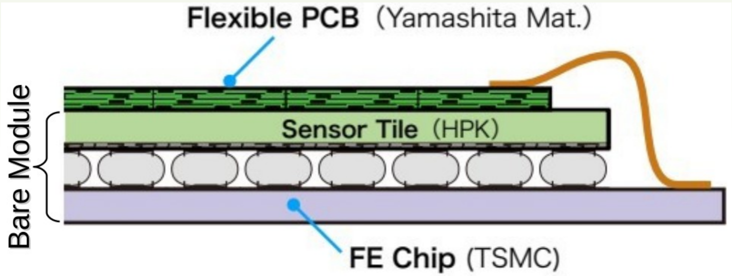
Bump bonding ↓

Bare module

Flex PCB
- Reception
- Population

Gluing ↓

Module
- Assembly
- Wire-bonding
- Parylene coating
- Thermal cycle



Schematic of a pixel module

Several assembly stages

=> QC (including Visual Inspection) at each step

Two main QC sites in Japan

PCB and Bare Module QC @KEK

Assembled module QC @HR

New Visual Inspection step added *before bump bonding*

=> Looking for critical defects

AI for Visual Inspection

- Objective

Improve the efficiency and reliability of Visual Inspection

=> Use deep Learning techniques

- Two categories of defects

AI for Visual Inspection

- Objective

Improve the efficiency and reliability of Visual Inspection

=> Use deep Learning techniques

- Two categories of defects

Statistical anomalies

Anomalies that appear in a minority of images

=> O(1%) occurrence rate

Very few examples available

=> Cannot make labeled dataset

Unsupervised defect detection

AI for Visual Inspection

- Objective

  Improve the efficiency and reliability of Visual Inspection

  => Use deep Learning techniques

- Two categories of defects

Statistical anomalies	**Expert anomalies**
Anomalies that appear in a minority of images **=> O(1%) occurrence rate**	More recurrent/common anomalies **=> Recognized as defects by experts**
Very few examples available **=> Cannot make labeled dataset**	Many examples available **=> Can make labeled dataset**
Unsupervised defect detection	**Supervised defect classification**

Unsupervised defect detection

- Denoising Auto-Encoder

  Reconstruct main input fetures **and** remove defect-like pattern

  => Enhance sensitivity for defect detection

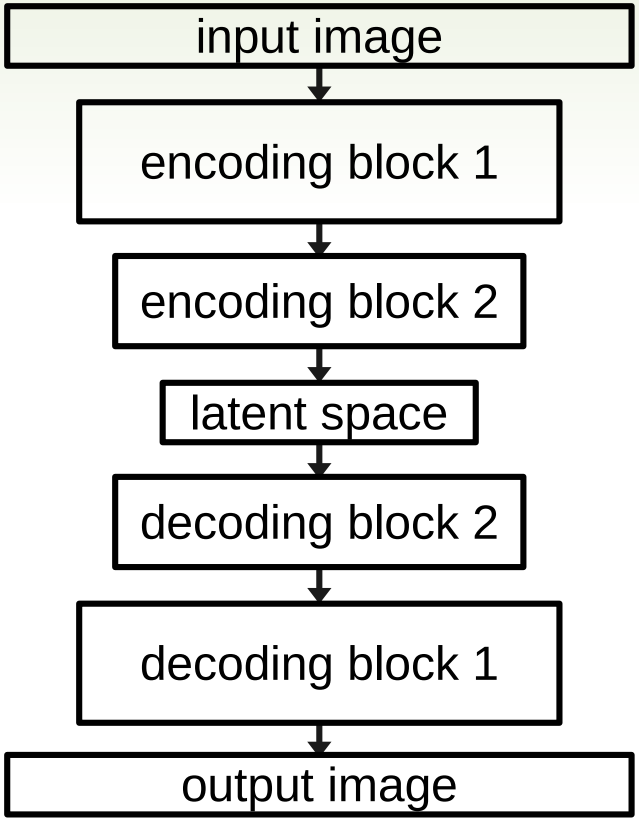
  Select pixel areas with **high reco error**

- Clustering and filtering

  Apply **clustering** to selected pixels

  Keep *only* major clusters

  => Defect candidates



Unsupervised defect detection

- Custom noise patterns

  Add noise patterns on input images during training

  => Randomize size, rotation and color

  Compute **MSE loss** between <u>output</u> and <u>*original* input</u>

  <u>Noise patterns</u> are made “by hand” to *ressemble* defect

  => Expert knowlege

Pattern examples

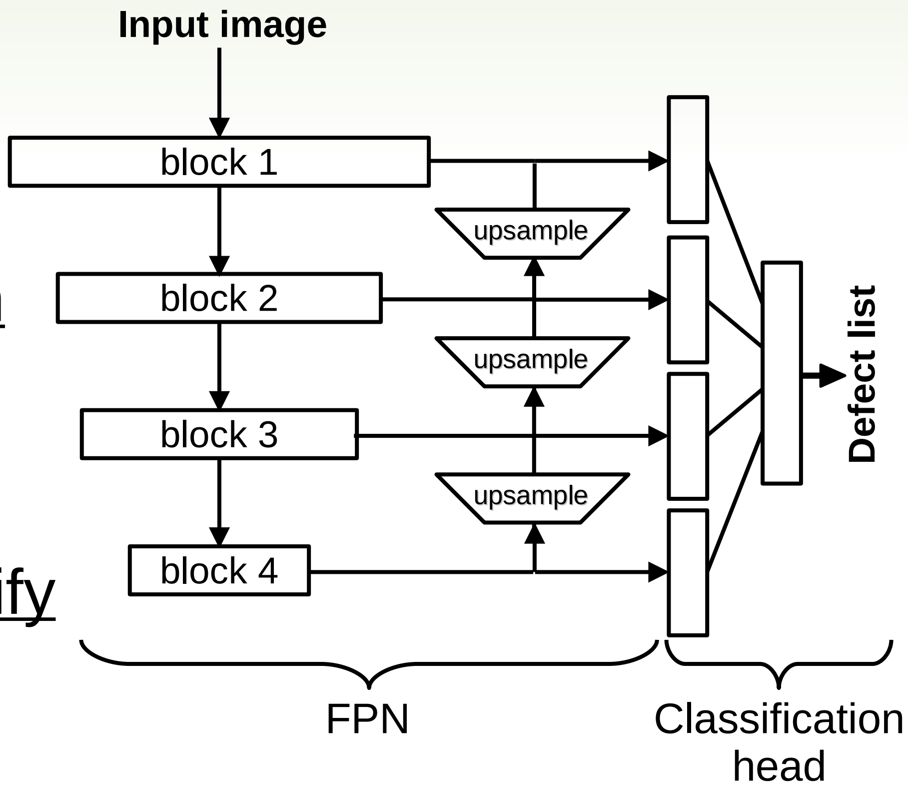
- Selection threshold

  Use *clean* test images and <u>compute reconstruction error</u>

  => Define threshold based on image without major defects

Supervised defect classification

- Feature Pyramid Network
  - Extract high level features from input image
  - Successive <u>dimension reduction</u>
    - **=> Pattern of different sizes**
- Classification head network
  - Use FPN feature space to <u>classify specific defects</u>
    - **=> Multiple classification objective**



Supervised defect classification

- FPN pretraining

  Generic feature extraction with *Auto-Encoder-like* objective

  Training with images of various components and stages

  => Common training of each specific components

- Main classifier training

  Use the **same pretrained FPN** for each component

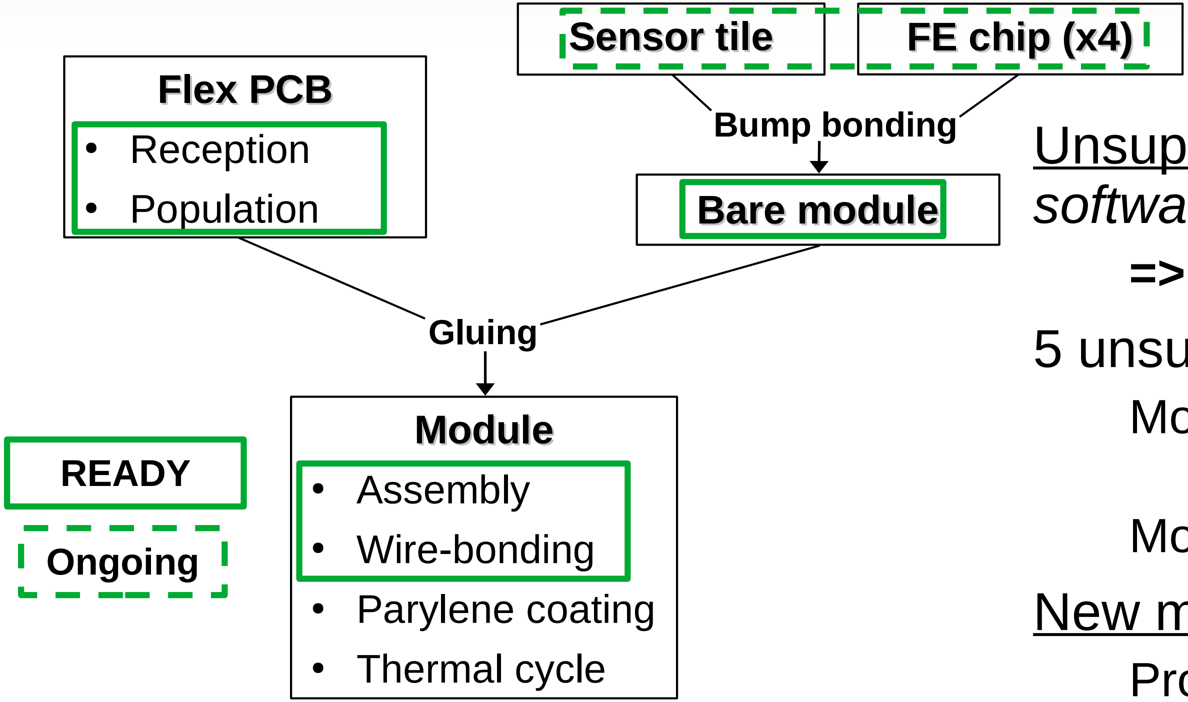
  Use one binary cross-entropy term per **target defect class**

$$\mathrm{loss}(x,y)=\sum_{C} y_C \log(x_C)+(1-y_C)\log(1-x_C)$$ ← **one score per category**

Improved training loop under testing

Deployment status

- Available models



<u>Unsupervised model</u> *integrated to QC software*

=> Ready to use

5 unsupervised models already trained

Models for *PCB* and *bare module* <u>deployed</u>

=> Visual Inspection @KEK

Models for *module* <u>ready for deployment</u>

<u>New model</u> for ASIC before bump bonding

Procedure under discution

Data preparation

- Data Acquisition

  <u>Camera and microscope</u> with moving stage

- Data augmentation

  Duplicate source images with random modifications

  => Cropping, scaling, brightness

  Split augmented images in 8x8 tiles (512x512 each)

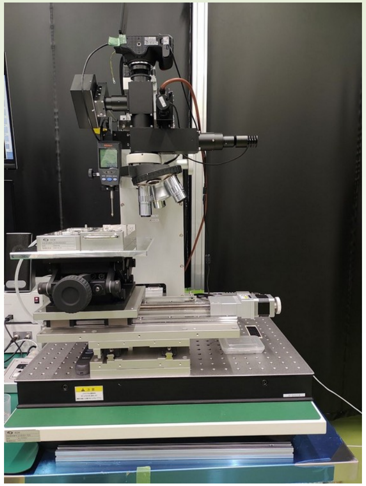
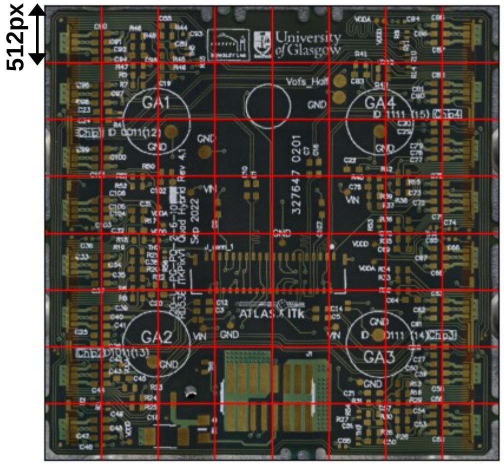


Image acquisition system @KEK



Some results

- Unsupervised defect detection

    Visual Inspection before Bare Module assembly

    => *Latest* model

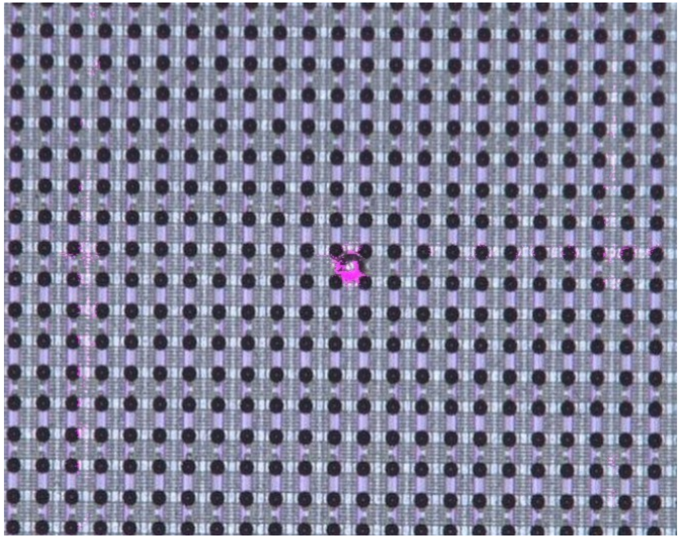
    Investigating <u>critical defects</u>

    => Might cause failure in later stages

    Selected area is colored in **pink**

    => Successful defect detection

    Still need to *reduce fake rates*

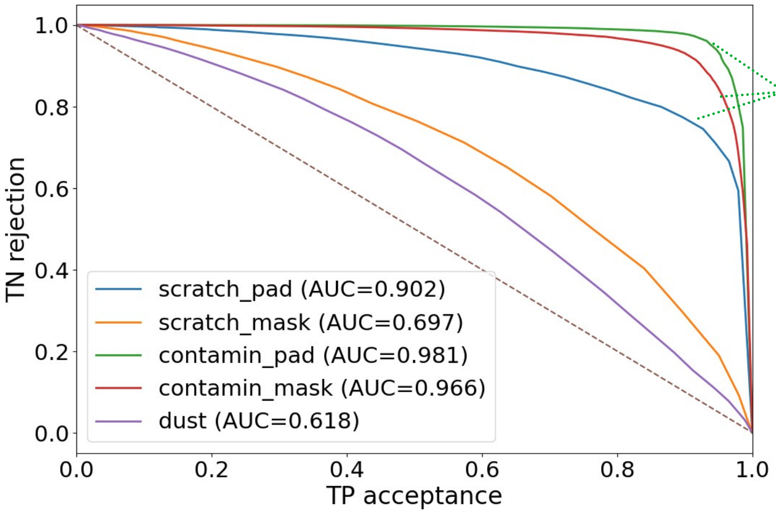


Some results

- Supervised defect classification

ROC curves showing performances for
5 defect categories

Good performances for 3 out of 5



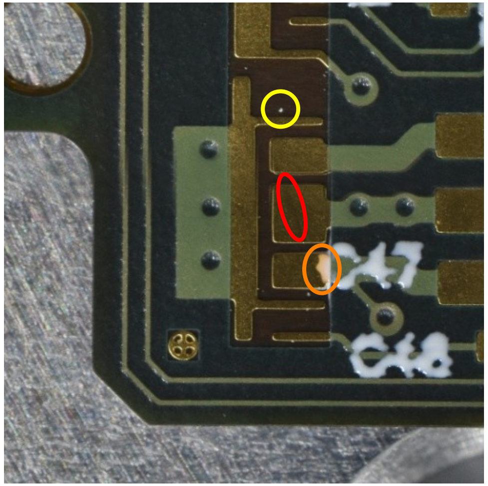
> 90% detection
< 20% fake

Application to a flex PCB image

3 defect categories detected

Dust, contamination and scratch

All 3 defects were properly identified



Conclusion

- Improving QC for ITk pixel modules

  Use Deep Learning techniques to improve Visual Inspection

  => Find more defect faster

  Successful integration of unsupervised model

  => Part of ITk QC software

- Generic API for AI-based Visual Inspection

  Available on GitHub and PyPI

  => Facilitate integration into existing QC framework

  Documentation and development features will *be implemented*

Thank you !

ありがとうございます！