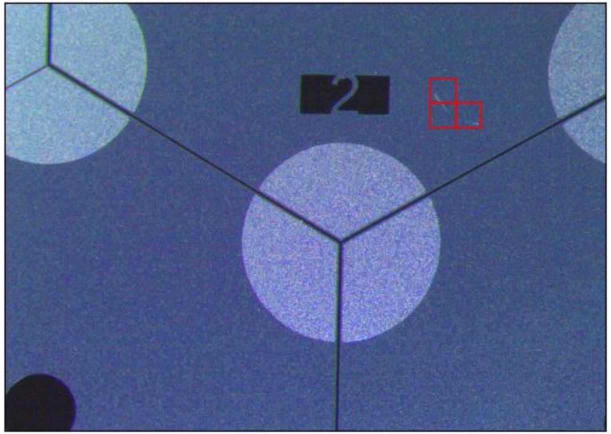


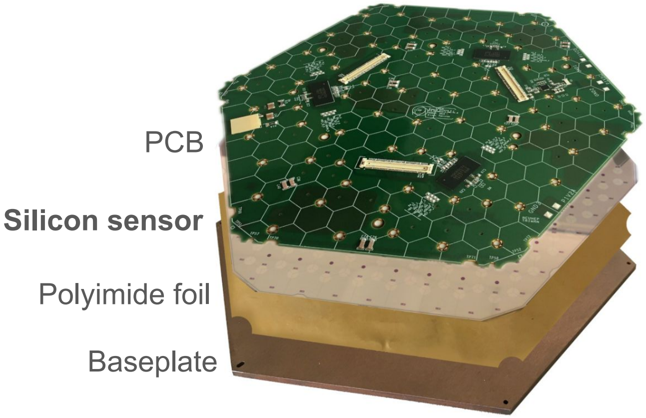
Anomaly detection for the quality control of silicon sensor wafers for the CMS HGCAL upgrade

Sonja Grönroos, Thorben Quast

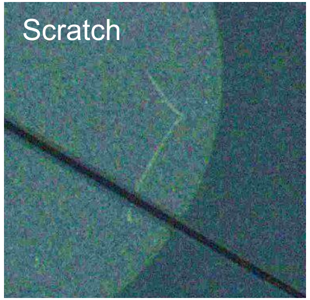
11/05/2022

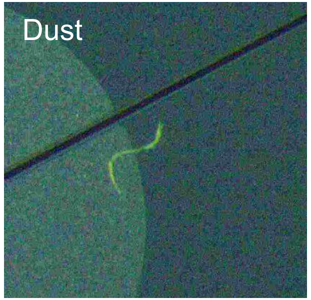
Silicon sensors will cover a large area

- Endcap calorimeters of the CMS must be upgraded for operation at HL-LHC (LS3)
  - High Granularity Calorimeter (HGCAL)
- HGCAL will consist of more than 25,000 hexagonal silicon pad sensor wafers
  - Wafer diameter 20 cm
  - Total sensitive area **620 m^2**
- Electrical breakdowns have been observed during prototype testing
  - Could sometimes be attributed to anomalies such as **scratches** or **dust** on sensor surface



The whole module, silver layer = sensor

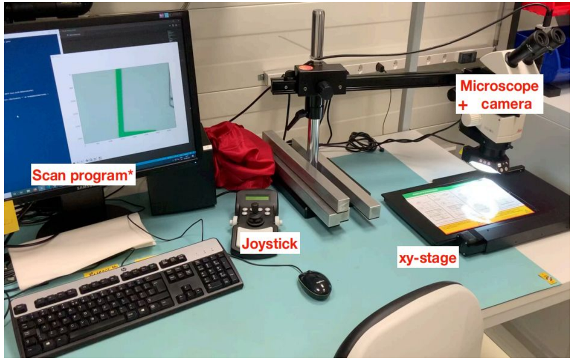




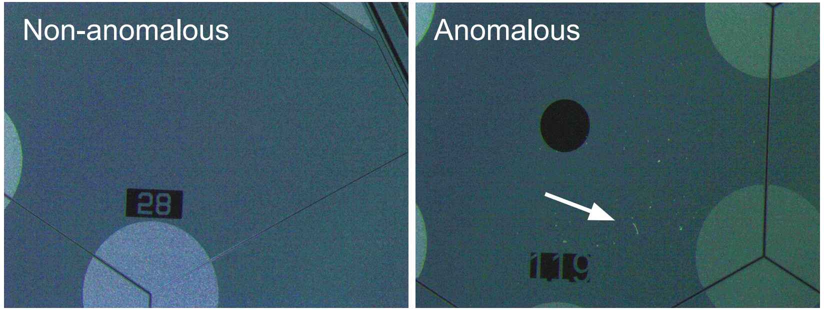
Examples of anomalies seen on sensor surface

Visual inspection as part of sensor quality control

- Wafer is moved underneath a microscope by an automatised xy-table
    - **~500 images** taken per wafer
- Human then inspects images on computer
    - Laborious
    - From experience: procedure **biased** by the subjectivity of the inspector
- Inspections of prototypes have accumulated plenty of image data
    - 26,607 images (size 3840 x 2748 px) taken from 53 sensors
    - 986 images prelabeled by inspectors to contain anomalies



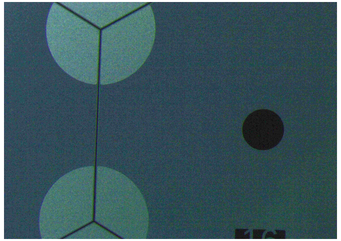
Set-up in the lab with which images are taken



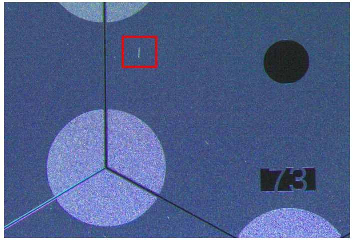
Per anomalous wafer, on average 10 out of the 500 images are anomalous

Deep learning to make inspection more efficient

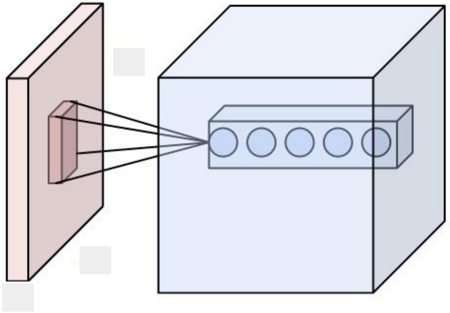
- Preselection algorithm that
  1. Goes through the images,
  2. indicates images with possible anomalies in a loose fashion,
  3. which are then verified by a human
- Image data = use convolutional neural networks (CNNs)
  - Must be **position** and **lighting** invariant
- Reduction of the number of images requiring human inspection by **1-2 orders of magnitude** per wafer



OK!
Ignore



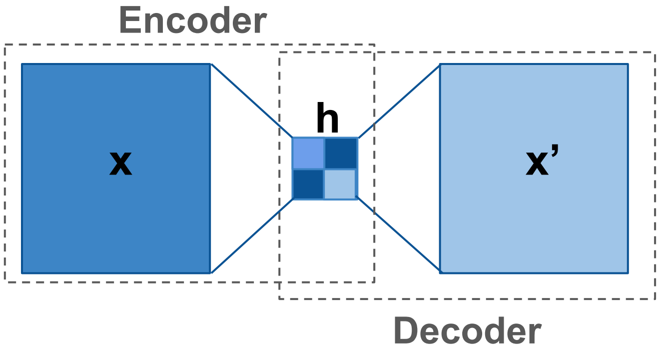
NOT OK!
Show to human



(*http://cs231n.github.io/convolutional-networks/*)

Autoencoders (AEs) in anomaly detection

- AEs have already been proposed for anomaly detection at the LHC, e.g. [1, 2]
  - Deep neural networks used for unsupervised learning
  - Task is to learn a compressed representation of input data
- AE consist of two networks:
  - Encoder (Image data = CNN)
  - Decoder (deconvolutional NN)
- Minimize reconstruction error
  - If trained on non-anomalous images, **error is expected to increase** in case of anomalous input



[1] J. Collins et al., Anomaly Detection for Resonant New physics with Machine Learning, Phys. Rev. Lett. 121. doi:10.1103/PhysRevLett.121.241803.
[2] M. Farina et al., Searching for new physics with deep autoencoders, Phys. Rev. D 101. doi:10.1103/PhysRevD.101.075021N.

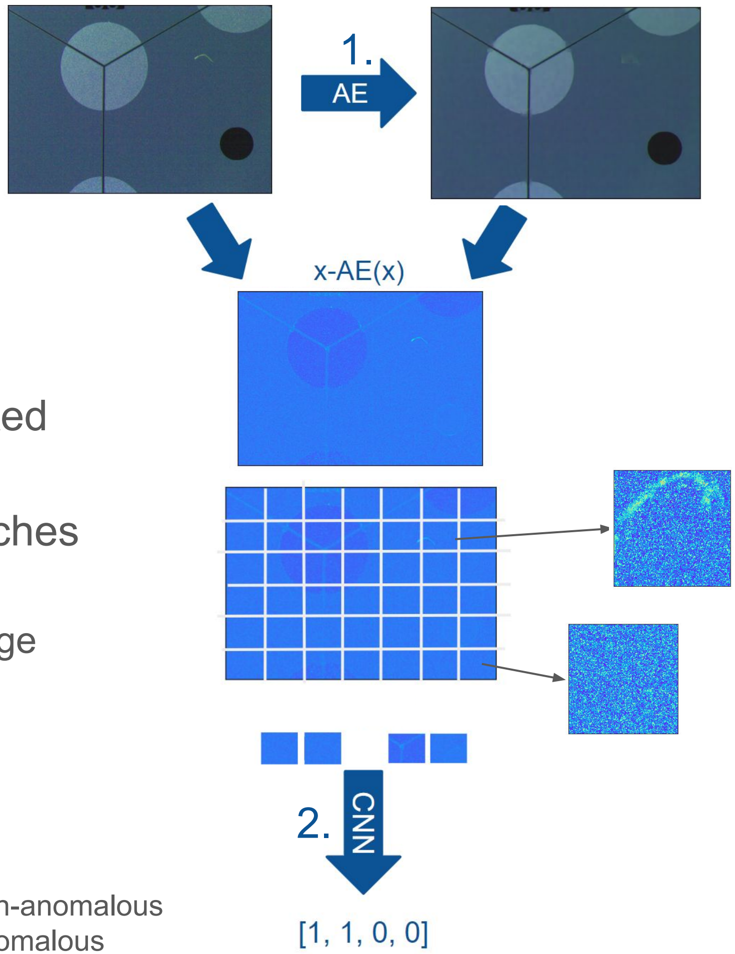
Current approach is an ensemble of two independent networks

1. Autoencoder
    - Trained on **non-anomalous whole** images

⇒ Difference between AE input and output is calculated

⇒ Artificial grid is used to split whole images into patches
  - Increases anomalous area relative to image size
  - Allows general localization of anomaly in whole image

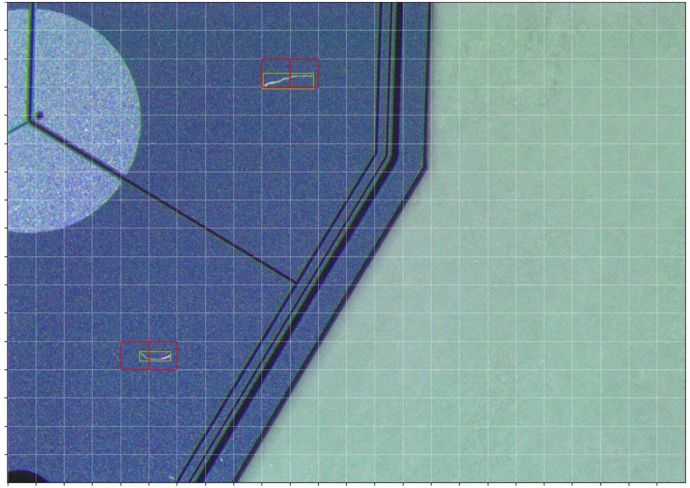
2. Classifier applied on patches
    - CNN for binary classification★



★0 = non-anomalous
1 = anomalous

Classifier requires labels

- Labels have been created for the 986 whole images prelabeled to be anomalous
  - Each whole image is split into 24 x 17 patches
  - Each patch is 160 x 160 px
- Selected patches contain anomaly (labeled as 1)
- Bounding boxes were created for potential future use
  - Free boxes - do not follow a grid
  - Single Shot Detection based approach [3]

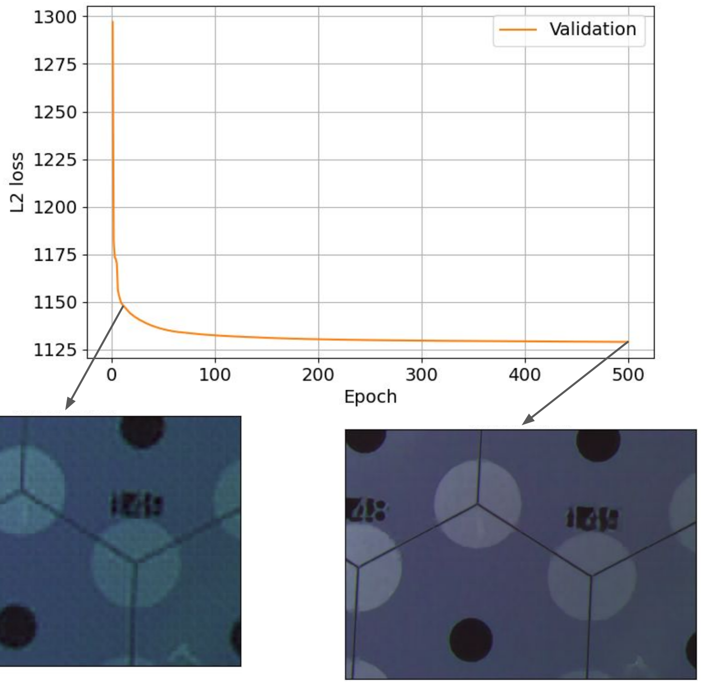


Whole image with both the selected patches (red) and bounding boxes (yellow)

[3] Liu, W. et al. (2016). SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2.

Autoencoder trained with non-anomalous images

- 8000 non-anomalous whole images were used to train an AE
  - 1000 images for testing/validation
- AE consists of a CNN and a DNN with 126,353 free parameters in total
- Trained for 500 epochs, 25 min each, ~9 days
  - 1x NVIDIA GTX 1080 GPU



Example of how AE improves during training

Adam(lr = 1e-4)
Batch size = 1

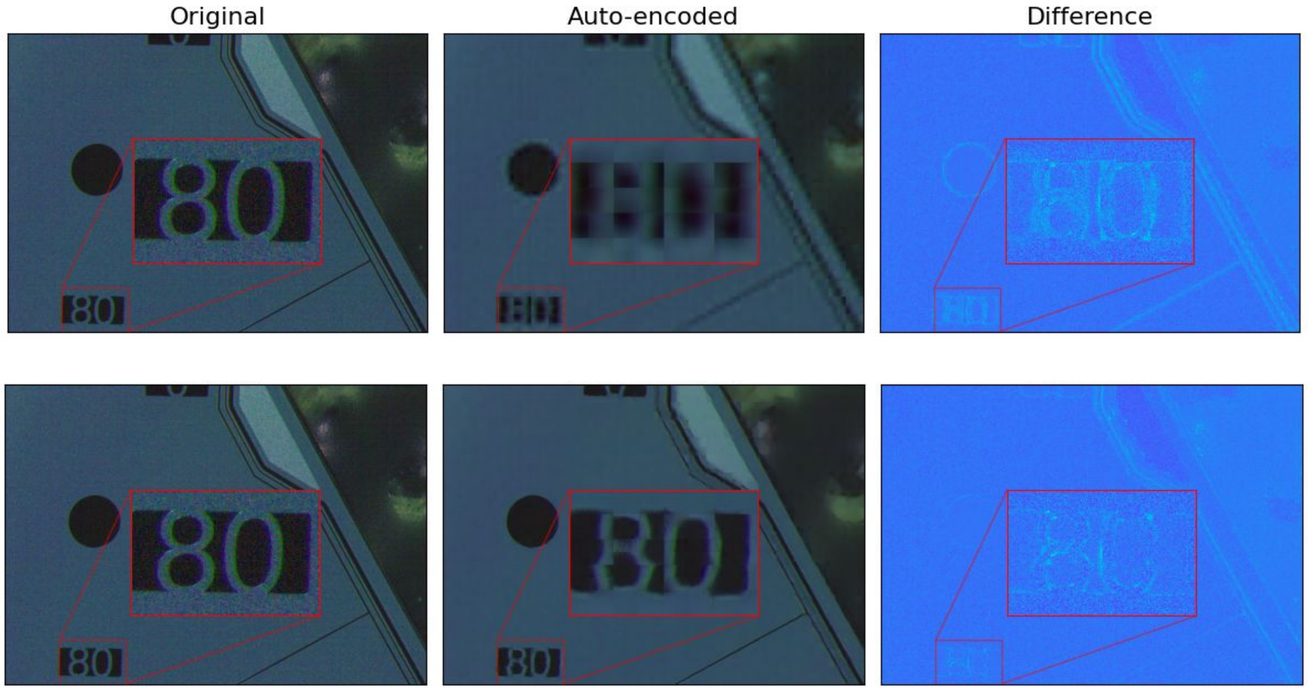
Encoding

Layer	Kernel size = stride	Activation	Output shape
Input			(2720, 3840, 1)
Conv1	(10, 8)	elu	(272, 480, 64)
Conv2	(4, 2)	elu	(68, 240, 64)
Conv3	(2, 5)	elu	(34, 48, 32)
Conv4	(2, 2)	elu	(17, 24, 32)

Decoding

Shape of latent space = (17, 24, 16)
Compression factor = 6400

Autoencoder after 5 epochs vs. 500 epochs

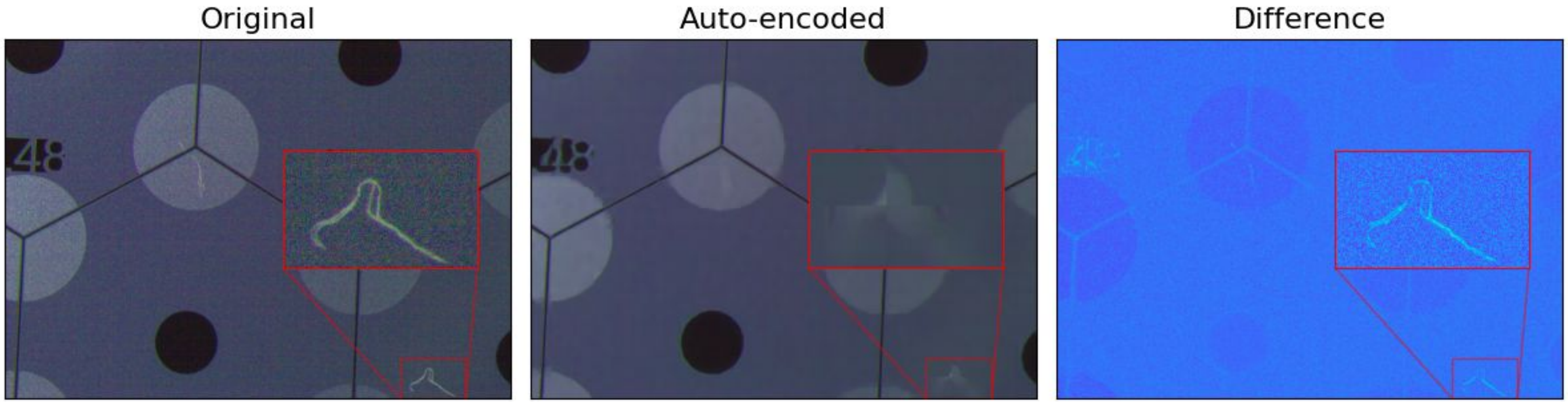


After 5 epochs

After 500 epochs

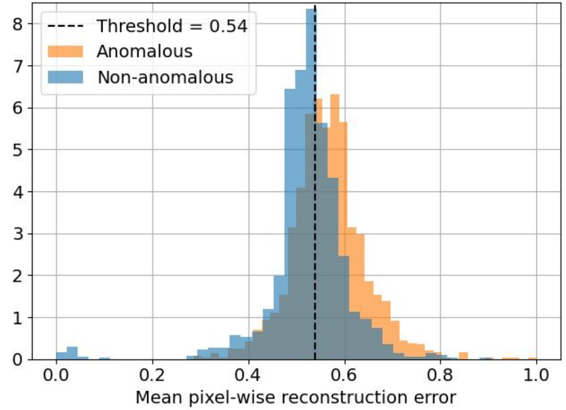
Autoencoder fails to reconstruct anomalies by design

- Anomalies are **enhanced** when the pixel-wise absolute difference between the original and auto-encoded images is calculated
- Reduction of the effects of environmental changes



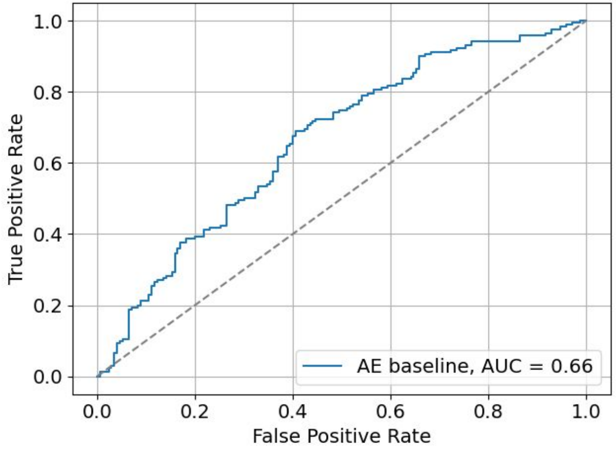
Autoencoder as an anomaly detector

- AE **reconstruction error** acts as **indicator of anomalies**
  - Mean pixel-wise reconstruction error was calculated for patches
- Training data 788 whole images, 1603 patches with anomalies
  - 50 % anomalous, 50 % non-anomalous patches = 3206 training images in total
  - Threshold selected based on validation score
- Some discrimination with test set (N=340)
  - **Baseline result**



False Positive Rate = 0.43
False Negative Rate = 0.31

		Predicted	
		0	1
True	0	97	73
	1	52	118

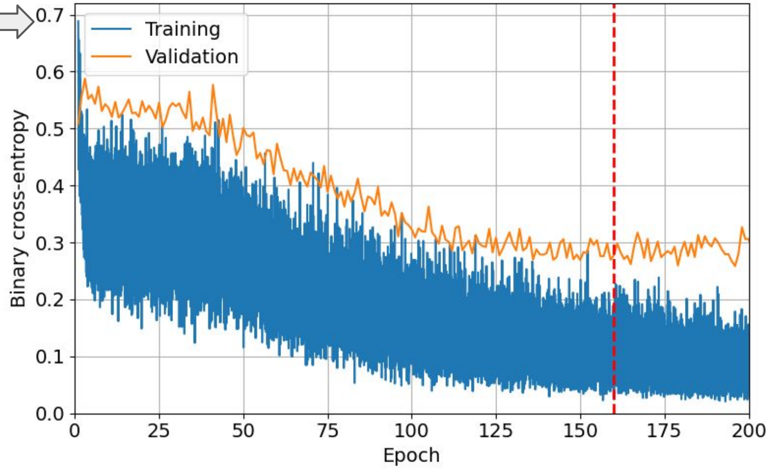


Classifier trained with patches

Layer	Kernel size = stride	Activation	Output shape
Input			(160, 160, 1)
Conv1	(2, 2)	relu	(80, 80, 16)
Dropout 0.4			
Conv2	(4, 4)	relu	(20, 20, 16)
Dropout 0.2			
Conv3	(4, 4)	relu	(5, 5, 32)
Dropout 0.2			
Conv4	(5, 5)	relu	(1, 1, 32)
Dropout 0.2			
Conv5	(1, 1)	sigmoid	(1, 1, 1)

Adam(lr = 1e-4)
Batch size = 128

Random guessing = 0.69



- A **classifier** CNN **is required** on top of the AE
- CNN with 38,081 free parameters trained for 160 epochs
- Training data consisted of 20 % anomalous, 80 % non-anomalous patches = 16,030 training patches
  - Data augmentation (random rotation) was used to double # of anomalous patches

Classifier trained with patches

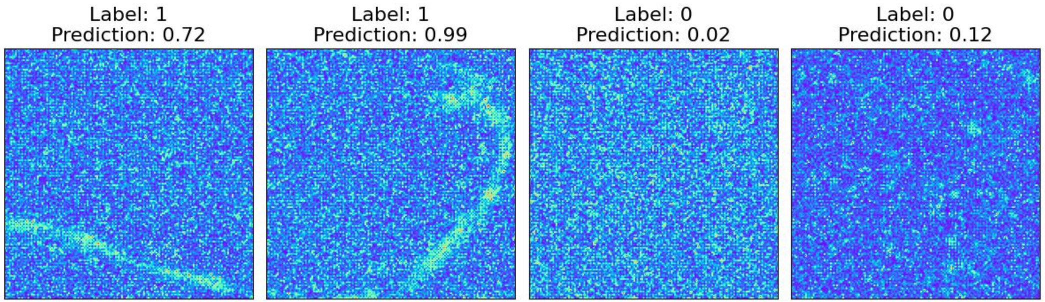
- 850 patches for testing (170 anomalous, 680 non-anomalous)
- Significant **improvement** to the AE baseline result
- Note: different data sets were used to train AE and CNN

Predicted

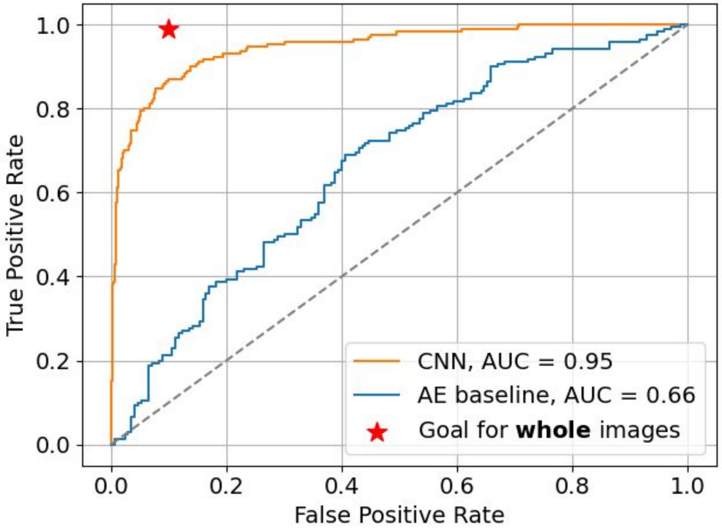
True		0	1
	0	586	94
	1	17	153

Threshold = 0.05

False Positive Rate = 0.14
False Negative Rate = 0.10



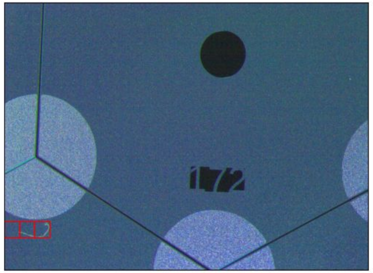
Examples of predictions on test set patches



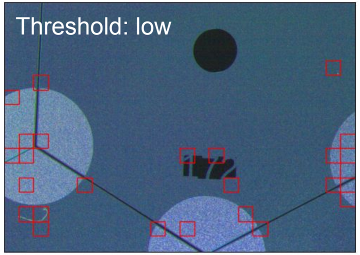
Project ongoing - summary

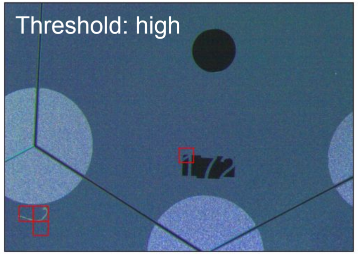
- Target: DL-based preselection algorithm to **accelerate the visual inspection** of silicon sensor surfaces for the HGCAL [4]
  - False negative rate as small as possible: our goal is 0.01
  - Anomaly detection must be fast to allow live inspection
  - Updatability of model with new inspection campaigns
  - Integration to lab environment

Food for thought: face-recognition style video processing



What we would like to see…



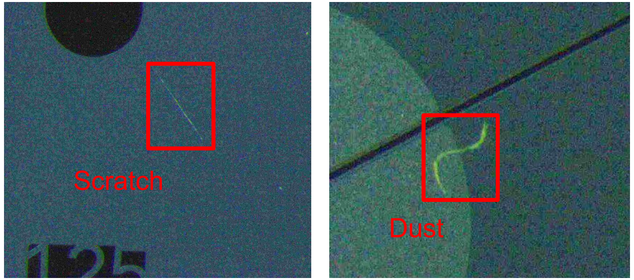


…vs. what can currently be seen

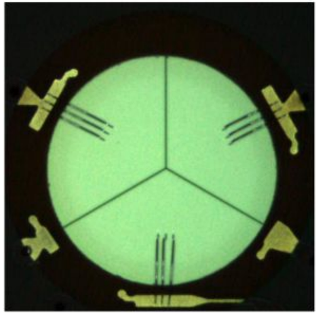
[4] N. Akchurin et al., “Deep learning applications for quality control in particle detector construction”, arXiv:2203.08969 [hep-ex], 2022.

Project ongoing - future

- Next steps include
  - More data augmentation to increase number of anomalous images
  - Considering a different approach: no cropping into patches
    - Anomaly detection + localization instead of binary classification
  - Model extension
    - Identify different kind of anomalies
- **Preliminary results are promising**
  - Potential for other applications, e.g. wire bonding quality control in module production



Anomaly identifier



Zoomed in image of a wire bond hole [4]

[4] N. Akchurin et al., "Deep learning applications for quality control in particle detector construction", arXiv:2203.08969 [hep-ex], 2022.

Thank you!

Questions?

Backup: Goal is to accelerate inspection process

- Model evaluation time must allow live inspection
  - If e.g. dust particle is spotted, it is manually removed while wafer still on the table
- Scan of one wafer (i.e. photo taking of 500 images) takes **~10 min**
  - Images could be evaluated in batches (of e.g. 32) while scan program is running
- Autoencoding step is the current time-consumer

If a human goes through images at the speed of 1 image / 1 s, 500 s = **8.3 min**

	32 images	16 * 32 = 512 images
Compute difference	11 s	176 s
Split	1 s	16 s
Predict	2 s	32 s
Total	14 s	224 s **= 3.7 min**

Current model performance on a GPU