IQONIC

Innovative strategies, sensing and process chains for increased Quality, re-configurability, and recyclability of Manufacturing Optoelectronics

This project receives funding in the European Commission's Horizon 2020
Research Programme under Grant Agreement Number 820677

Objectives

- Develop models for predicting defects due to deterioration of material and machine condition.
- Develop multi-parametric models of product instances (single stage).
- Build an AI driven self-adjusting inference engine for integrating real-time inspection and control at the shop-floor and decision support at single-stage and multi-stage manufacturing (process plan).
- Interface with the Manufacturing Enterprise Systems and other higher level management systems.

Tasks

- Online condition monitoring – using precision sensors and actuation technologies to build the causality relationships for accurate modelling and control of production process.
- Event Modelling for fast forward circularity, cost functions and validation of KPFs (tracking & tracing products as they evolve through the production line).

This project receives funding in the European Commission's Horizon 2020
Research Programme under Grant Agreement Number 820677

iQonic

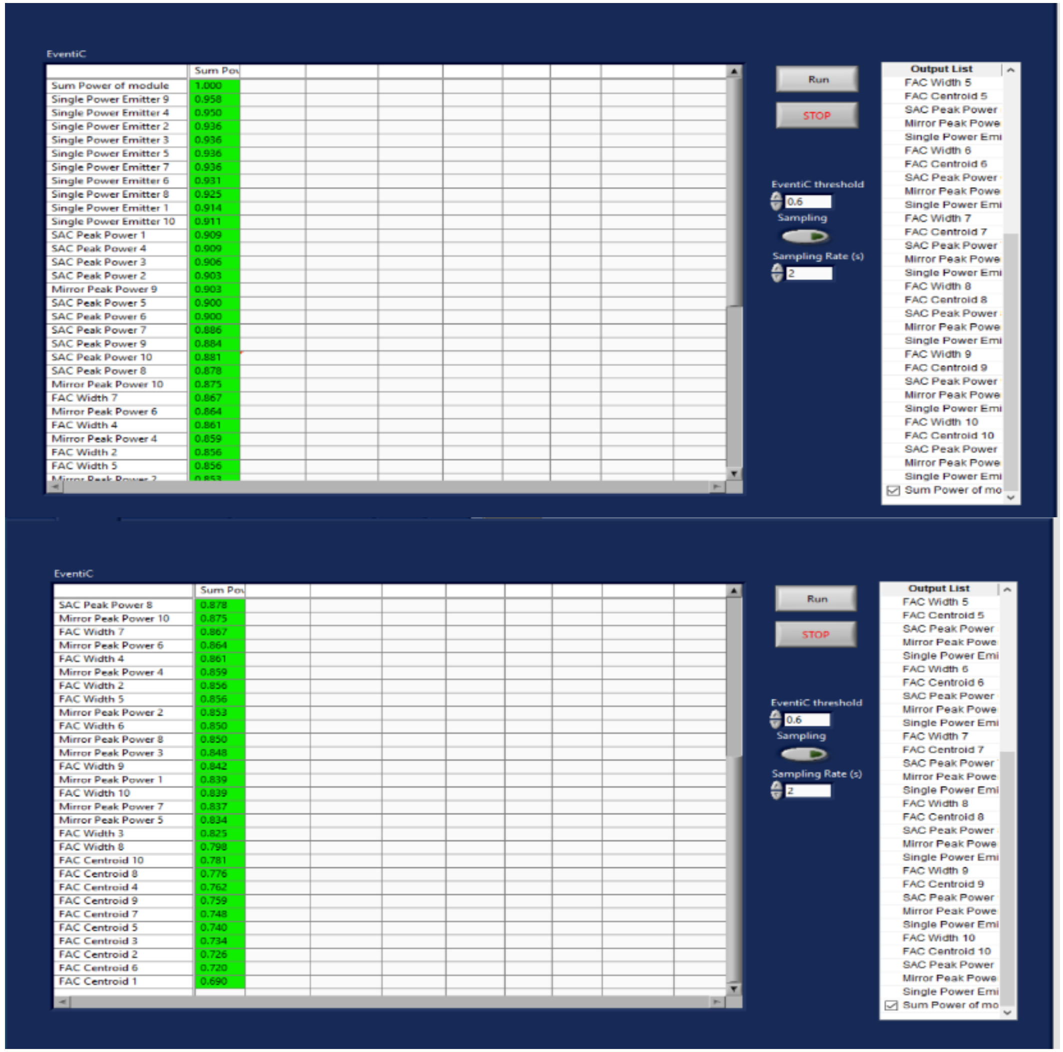
Prediction Approach

Correlation analysis of the data

1st step → Knowledge from the experts on the field regarding the important input variables that affect the outcome.

2nd step→Use of EventiC to validate their correlation.

3rd step →Continuous cross-validation of the results with the experts in order to acquire a better tuning on the thresholds and the accepted ranges.



Correlation analysis with EventiC

Explanation of Algorithm

- The assembly process is divided in 31 steps. 10 FAC placements, 10 SAC placements, 10 Mirror placements and 1 Lens placement.

- Each step has several data inputs. To a current step , apart from the respective data inputs, a cumulative information from previous data inputs is added e.g. Step 15 (Placement of SAC in front of Emitter 5) has as inputs the inputs from the previous steps (Steps 1-14) and the inputs from step 15.

- Due to the complexity that might rise, the NNs currently investigated are focusing on eliminating any unnecessary information and transfer only the needed information to the next step. That way the model will be fast and computationally efficient.

Assembly process steps

FAC Placement 1-10 → SAC Placement 1-10 → Mirror Placement 1-10 → Lens placement

Explanation of Algorithm: Continued

- Each step will have its own implemented NN (many steps might have similar NNs). This will augment the model's ability to work in real-time. E.g. in the first steps that have few inputs and low complexity, a NN that is highly sophisticated will produce the same results with a simpler one or even with a simple regression model.
- Computational time and computational resources are of the essence here and should be treated carefully.
- The final model will be the combination of the NNs and other methods that were used in each step.

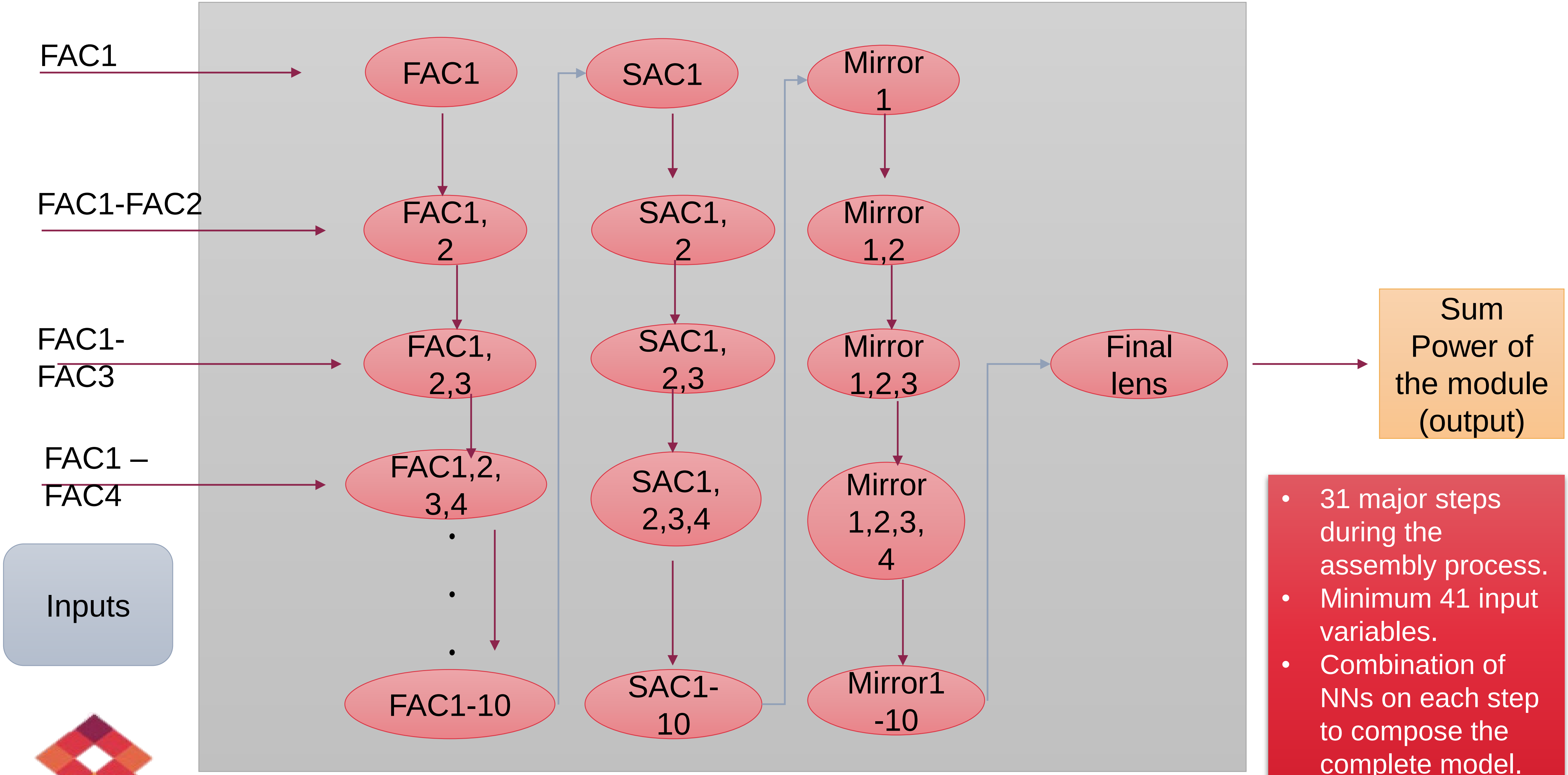
User-friendly interface and an example of the desired outcome

Step 1 is complete.
Probability of failure: 7%.
Recyclability of material: 85%.
Cost: 22.56$.
Suggestion: Proceed normally.

Step 7 is complete.
Probability of failure: 35%.
Recyclability of material: 40%.
Cost: 250.42$.
Suggestion: Risk of defect. Proceed after manual examination.

Step 15 is complete.
Probability of failure: 65%.
Recyclability of material: 15%.
Cost: 600.31$.
Suggestion: High risk of defect. Abort assembly process.

Algorithm Flowchart



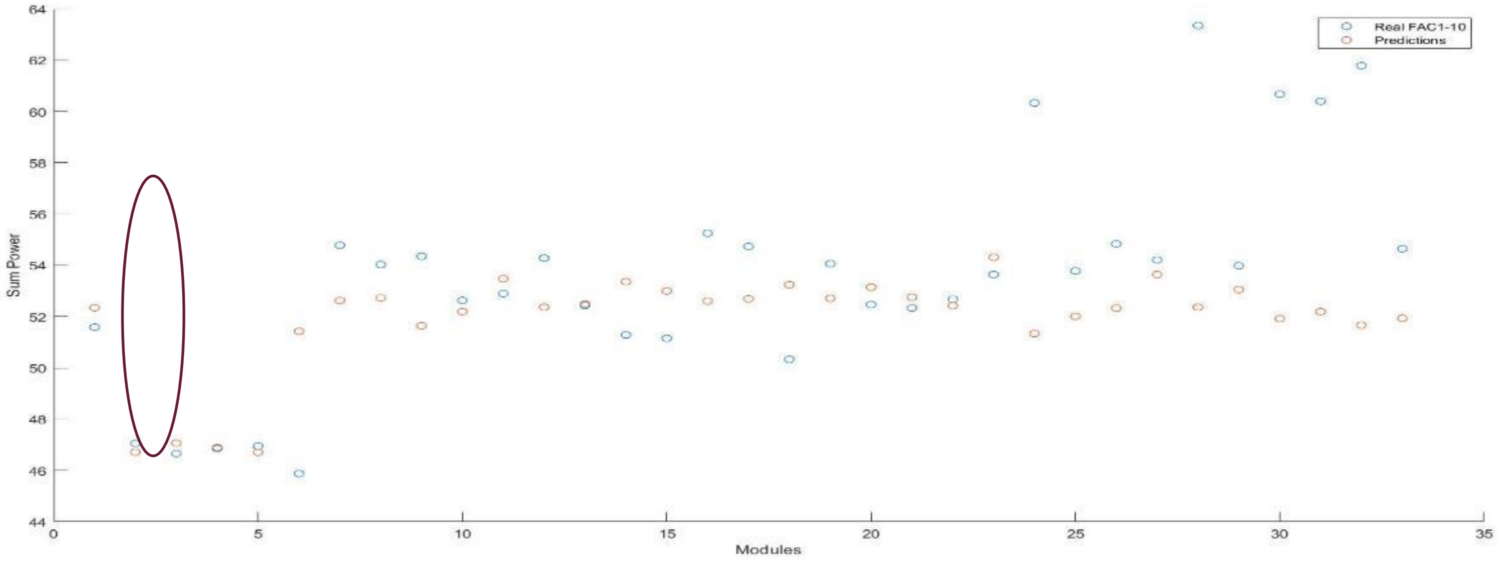
- 31 major steps during the assembly process.
- Minimum 41 input variables.
- Combination of NNs on each step to compose the complete model.

Example of Early Stage Prediction

- Step 10 (placement of FAC in front of Emitter 10).

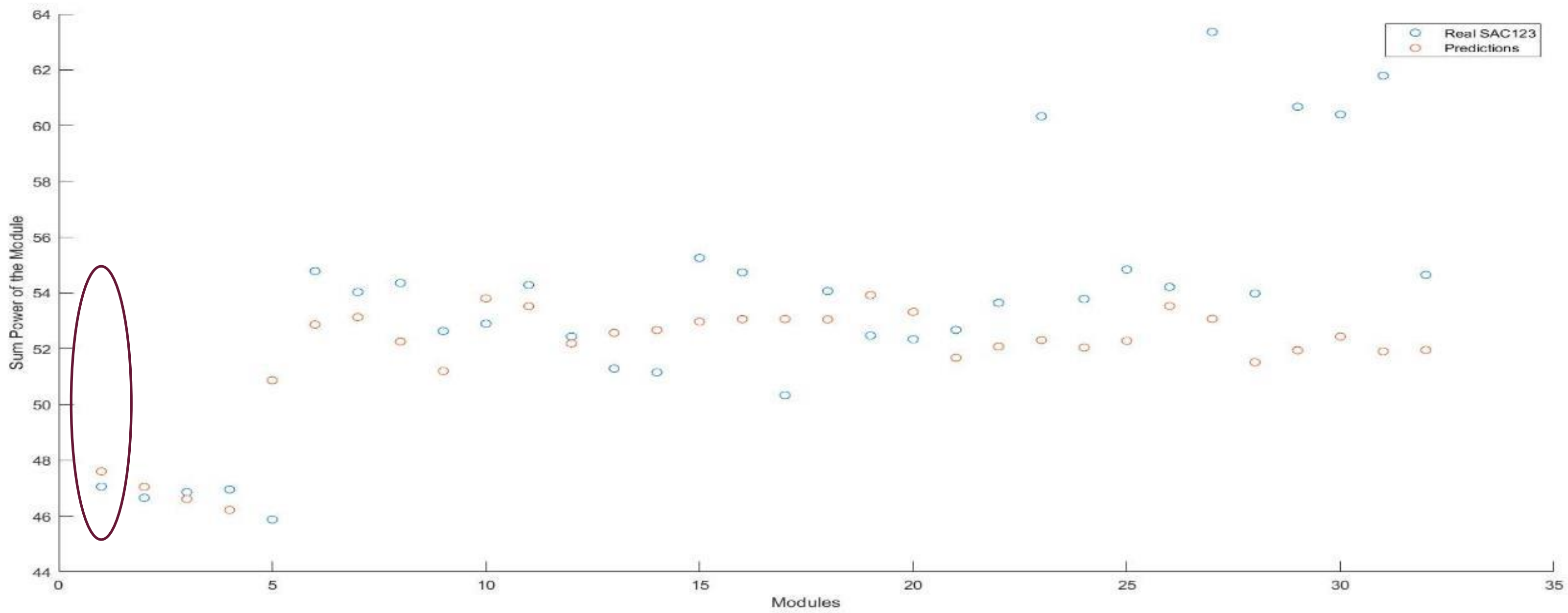
- Squared Exponential Gaussian Process Regression.

- 20 Inputs (**FAC Width**, **FAC Residual**) , 1 output (**Sum Power of the Module**).
- 330 modules were used for training and 32 modules were used for testing.
- From the 32 predictions only 1 was wrong (categorical predictions of **pass** or **fail**).



Example of Early Stage Prediction: Continued

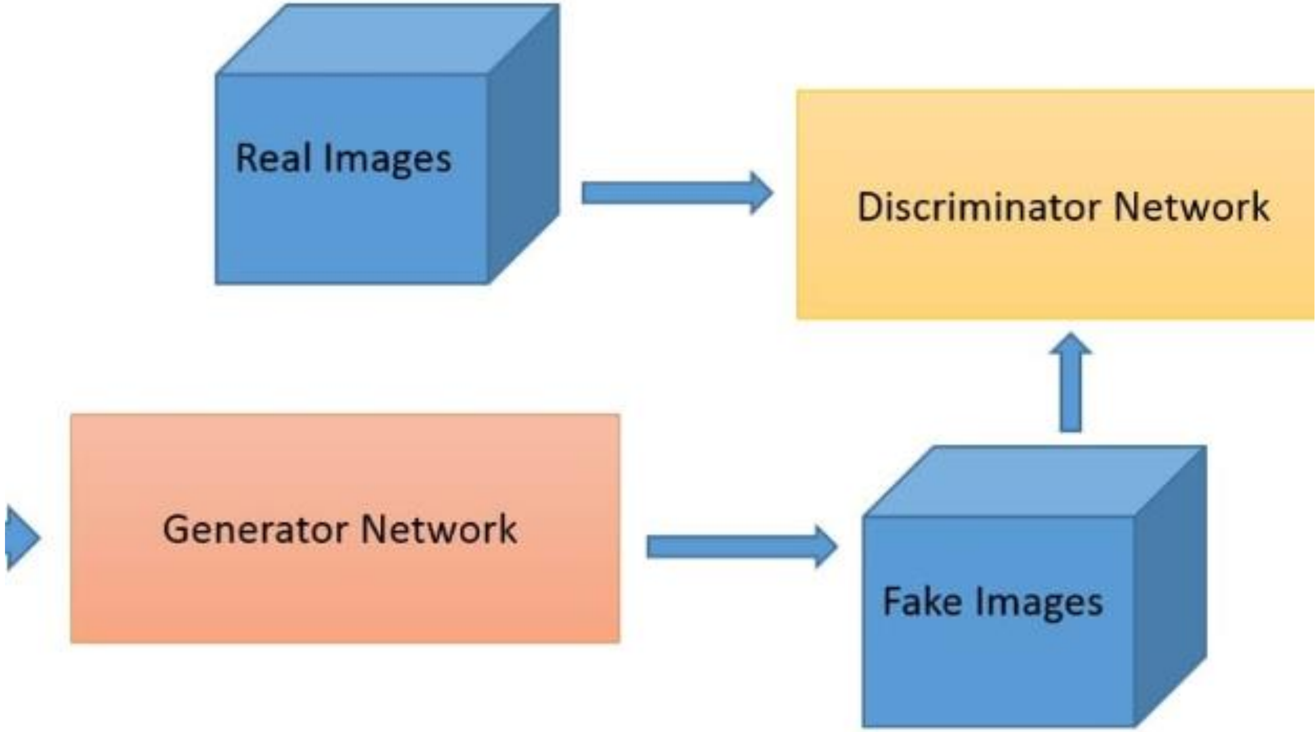
- Medium Gaussian SVM model for Step 13 (placement of SAC in front of Emitter 3). Input variables are **FAC Width**, **FAC Residual and SAC Peak Power**.
- Output is: **Sum Power of the Module**. In total 23 inputs (FAC Width and Residual from Emitters 1 to 10, SAC Peak Power from Emitters 1 to 3) and 1 output.
- In later stages the complexity will not allow us probably to deploy such simple methods.
- From the 32 predictions only 1 was wrong (categorical predictions of **pass** or **fail**).

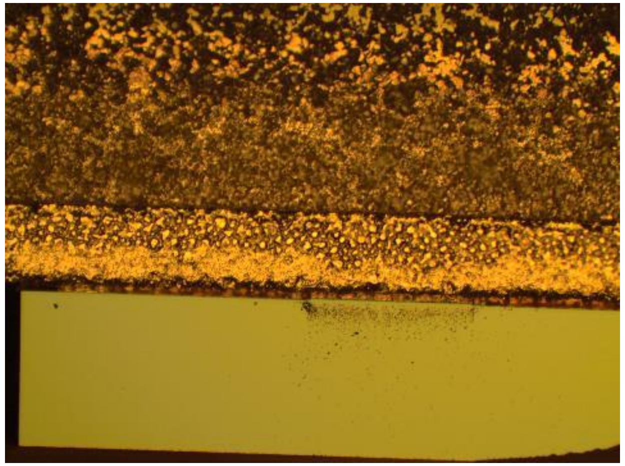


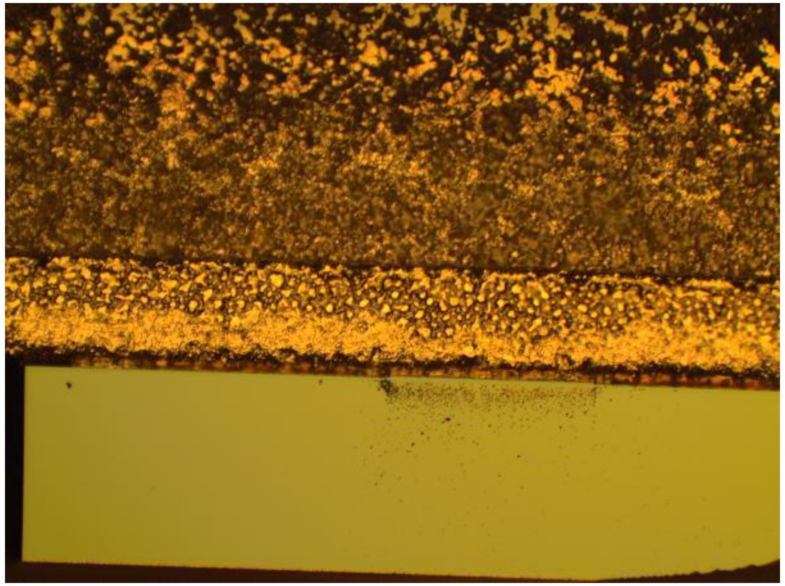
Detection Approach

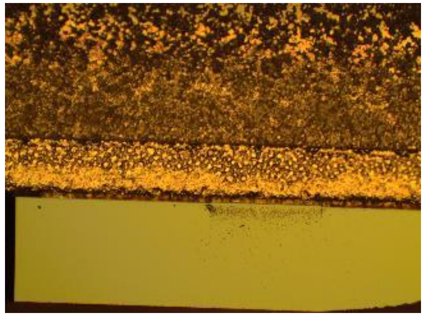
GAN for Prima Image Processing

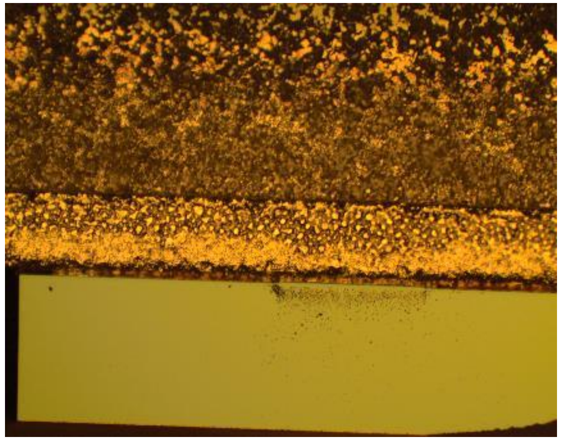
- **Generative Adversarial Network (GAN)**
  - Given a training set (e.g. images), GAN learns to generate new data (images) with the same statistics as the training set
  - Generator: Generate candidates
  - Discriminator: evaluates candidates





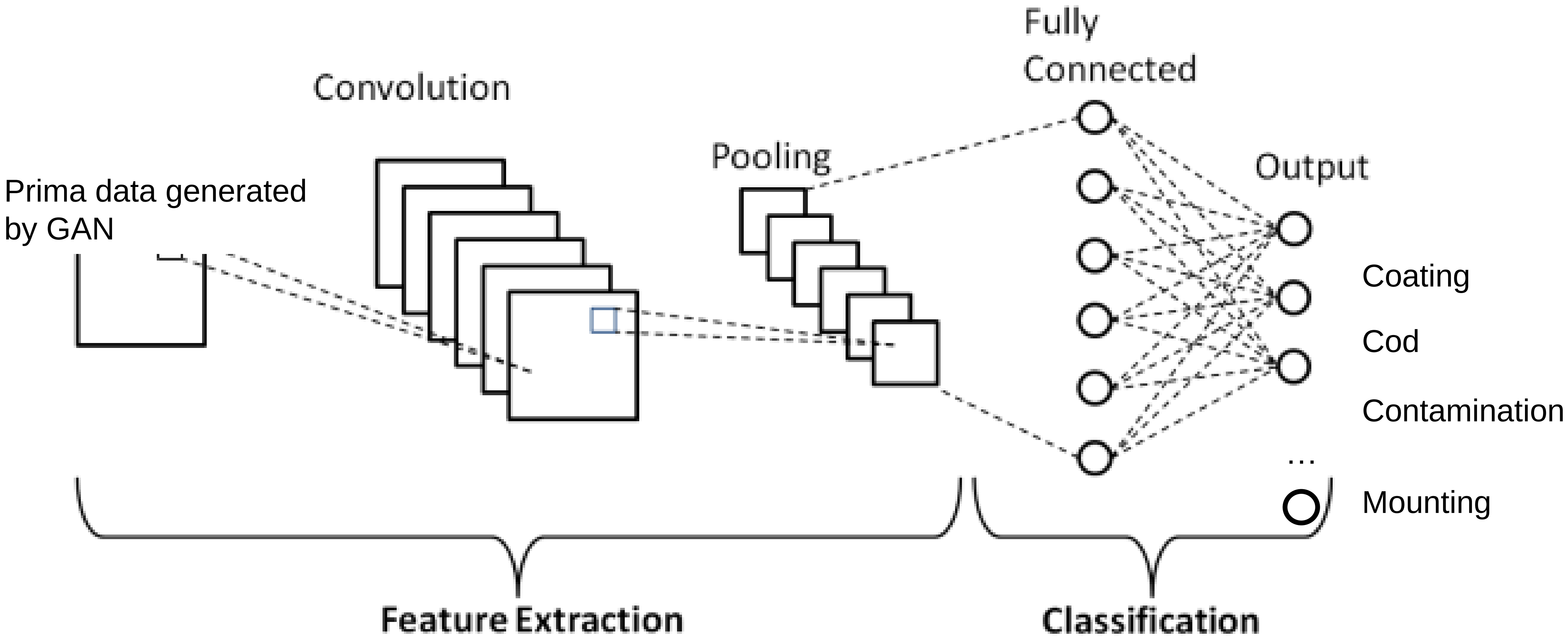






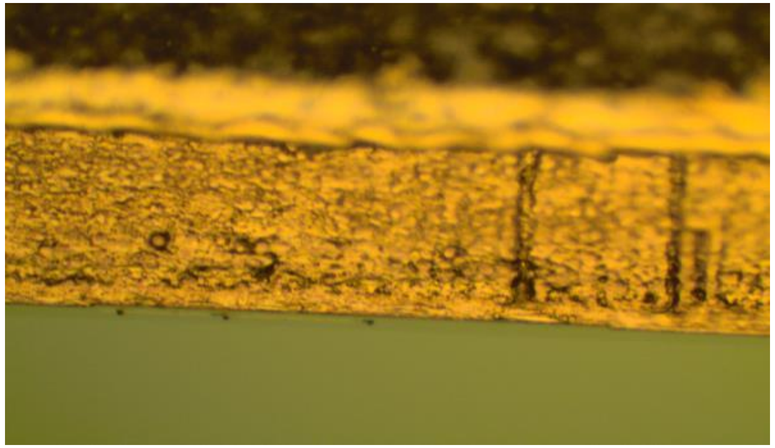
```
train_x = permute(train_x,[3,2,4,1]);
batch_size = 100;
% ----------- model
generator.layers = {
    struct('type', 'input', 'output_shape', [100, batch_size])
    struct('type', 'fully_connect', 'output_shape', [7*7*32, batch_size], 'activation', 'leaky_relu')
    struct('type', 'reshape', 'output_shape', [7,7,32, batch_size])
    struct('type', 'conv2d_transpose', 'output_shape', [14, 14, 16, batch_size], 'kernel_size', 5, 'stride', 2, 'p
    struct('type', 'conv2d_transpose', 'output_shape', [28, 28, 1, batch_size], 'kernel_size', 5, 'stride', 2, 'pa
};
discriminator.layers = {
    struct('type', 'input', 'output_shape', [28, 28, 1, batch_size])
    struct('type', 'conv2d', 'output_maps', 16, 'kernel_size', 5, 'padding', 'same', 'activation', 'leaky_relu')
    struct('type', 'sub_sampling', 'scale', 2)
    struct('type', 'conv2d', 'output_maps', 32, 'kernel_size', 5, 'padding', 'same', 'activation', 'leaky_relu')
```

Using Deep Learning approach (CNN)



CNN Implementation

- Note on improving accuracy:
  - If Prima can provide more image samples that would help to optimize model’
  - s accuracy instead of using GAN. The range of good image samples would be more that 1000 image/defect.
  - If we can get images from the same angle for the same defect category.



- Explanation:
  - Two images belong to the same defect category
  - but algorithm cannot be trained with such variations.

Future Work

- Final selection of the best NNs for the job.
- Optimization of the thresholds used in EventiC.
- Communication with Prima for some questions that might rise in the next few weeks.
- Presentation to the next meetings of more results of the later assembly steps and the detection part.

Thank you !
Any questions?

Brunel University London

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 820677