https://ai4mi.epotentia.com

AI FOR MATERIALS INDUSTRY

ARTIFICIAL INTELLIGENCE

A MASSIVE OPEN ONLINE COURSE

Hands-on session 2

Steel defect classification with deep learning

ePotentia Data·Science·Hosting

VLAAMS SUPERCOMPUTER CENTRUM

Vlaanderen is supercomputing

fwo Fonds Wetenschappelijk Onderzoek Vlaanderen Opening new horizons

INVESTEERT IN JOUW TOEKOMST

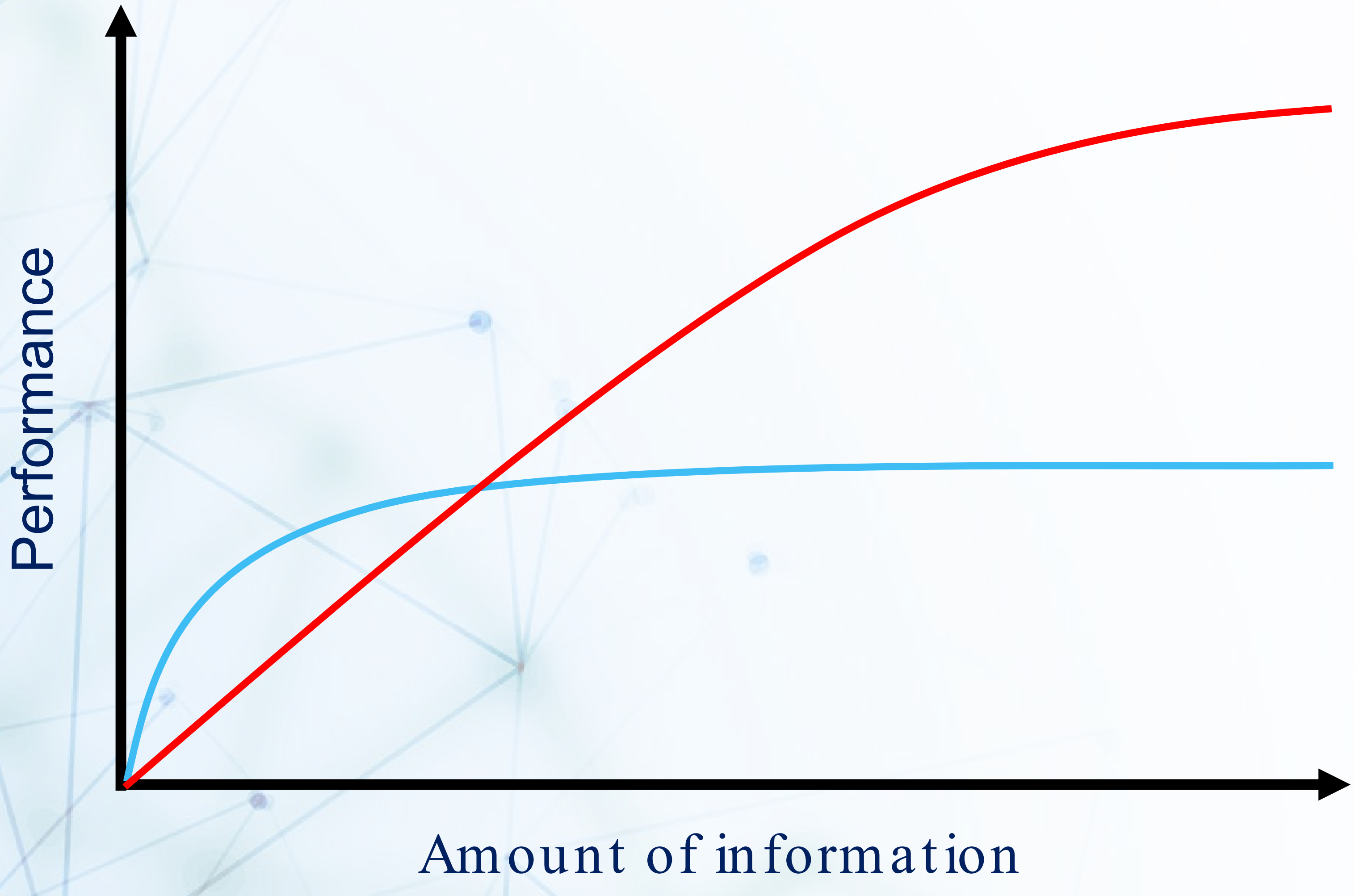
ESF Europese Unie Vlaanderen is werk

AI FOR MATERIALS INDUSTRY

What is AI?

Series of techniques to extract **information** from data

Types of AI



The more data we have, the more we trade insight for predictive power

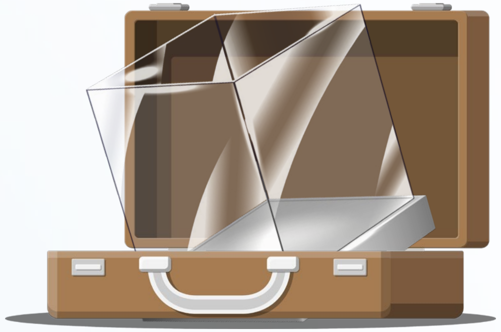
Case studies

Spreadsheet data

Steel

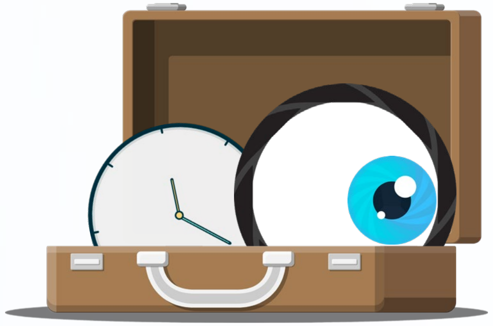
Computer vision

Glass



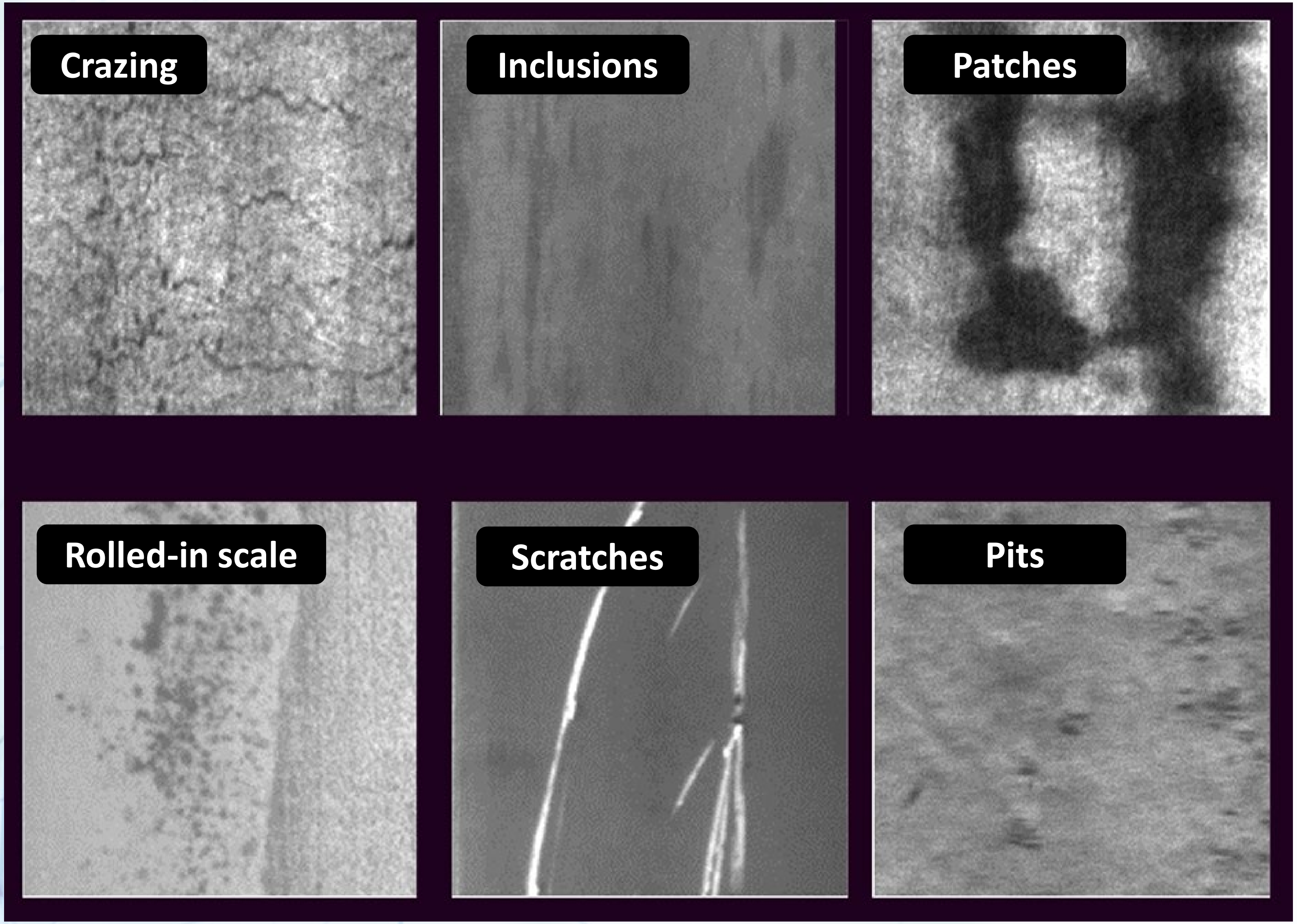
Materials discovery

Manufacturing

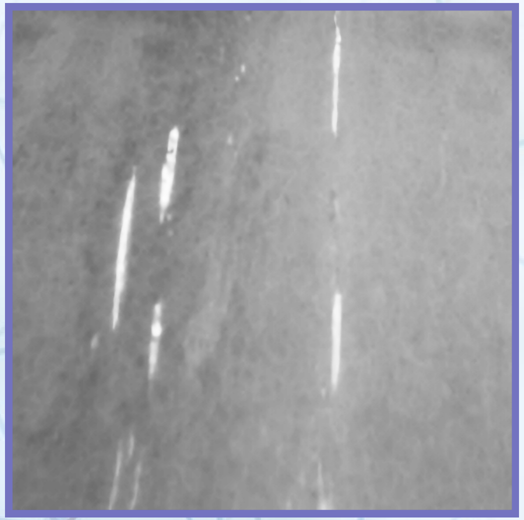


Sensor data

Steelplate defects



Multi-step pipeline

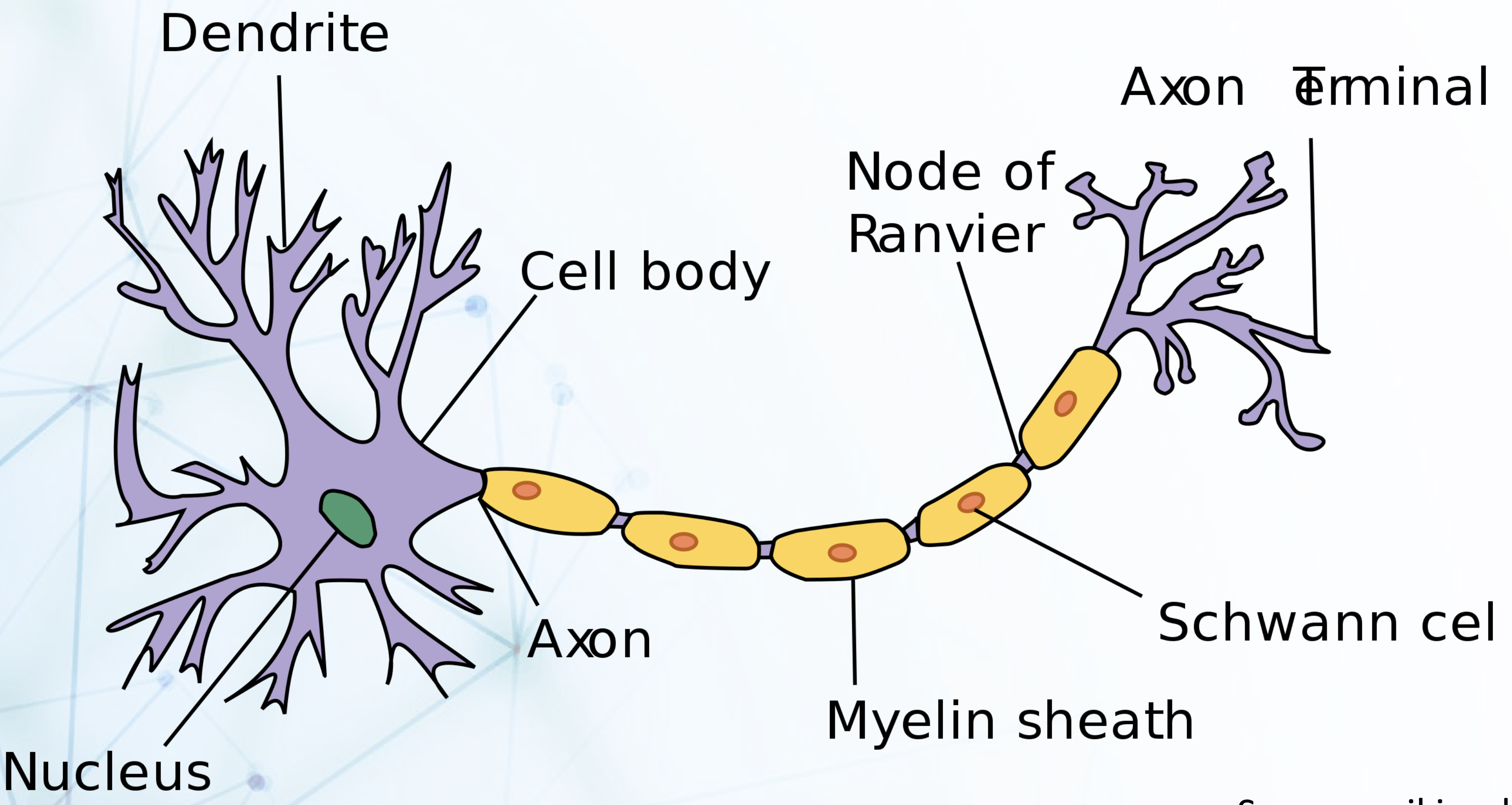


→ Detector → Featurizer → Classifier

Can we make a model that classifies the raw image directly?

AI FOR MATERIALS INDUSTRY

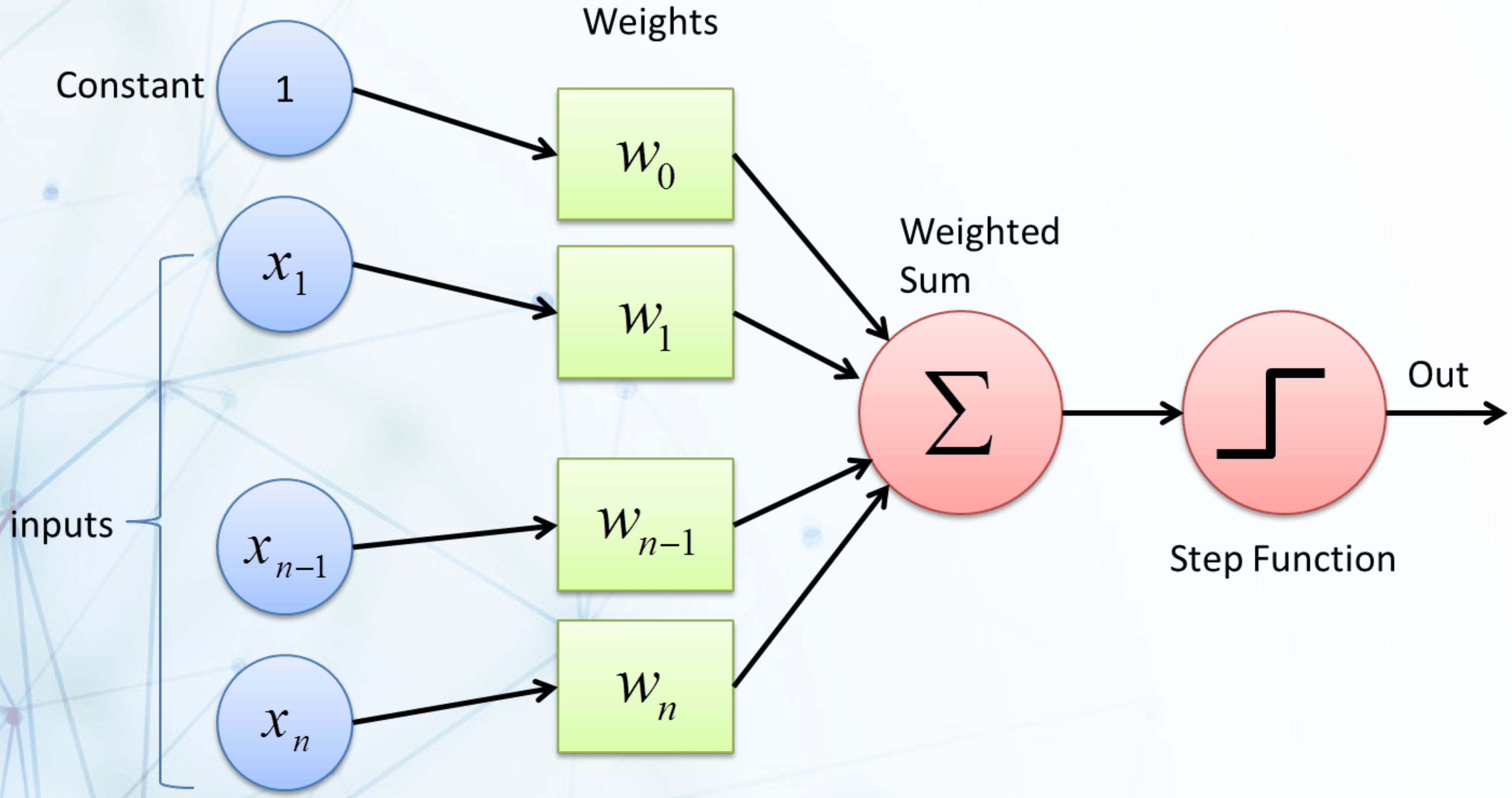
Neural networks



Source: wikipedia

Inspired by nature

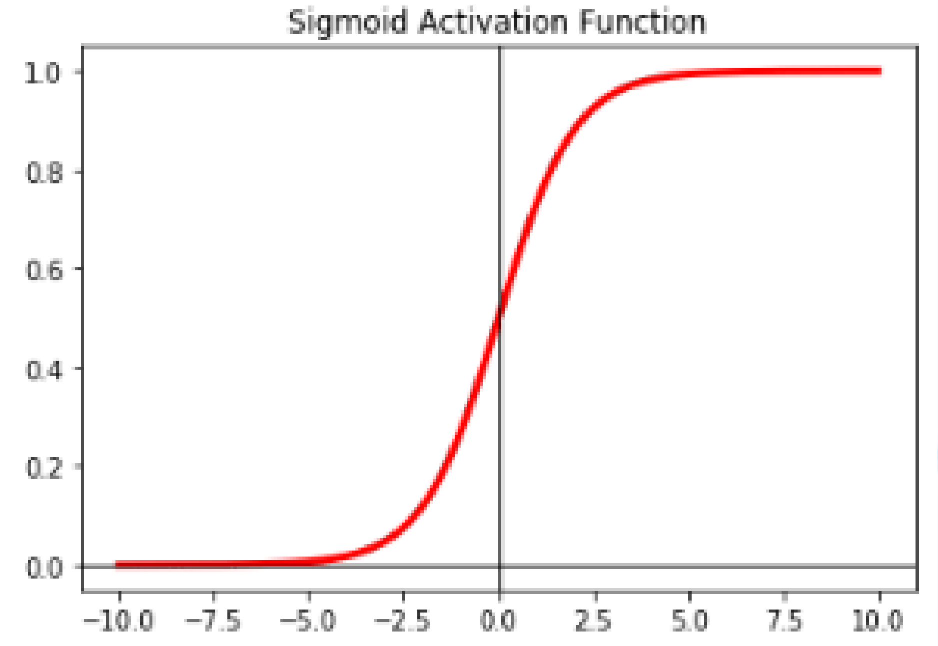
Artificial neurons or perceptrons



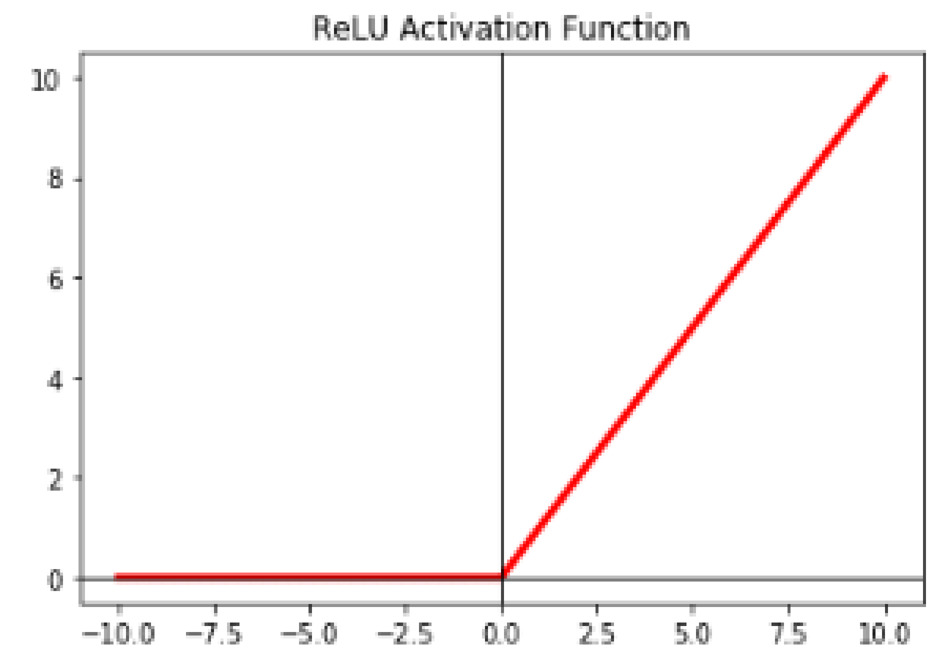
A perceptron is equal to linear or logistic regression

Activation functions

Sigmoid

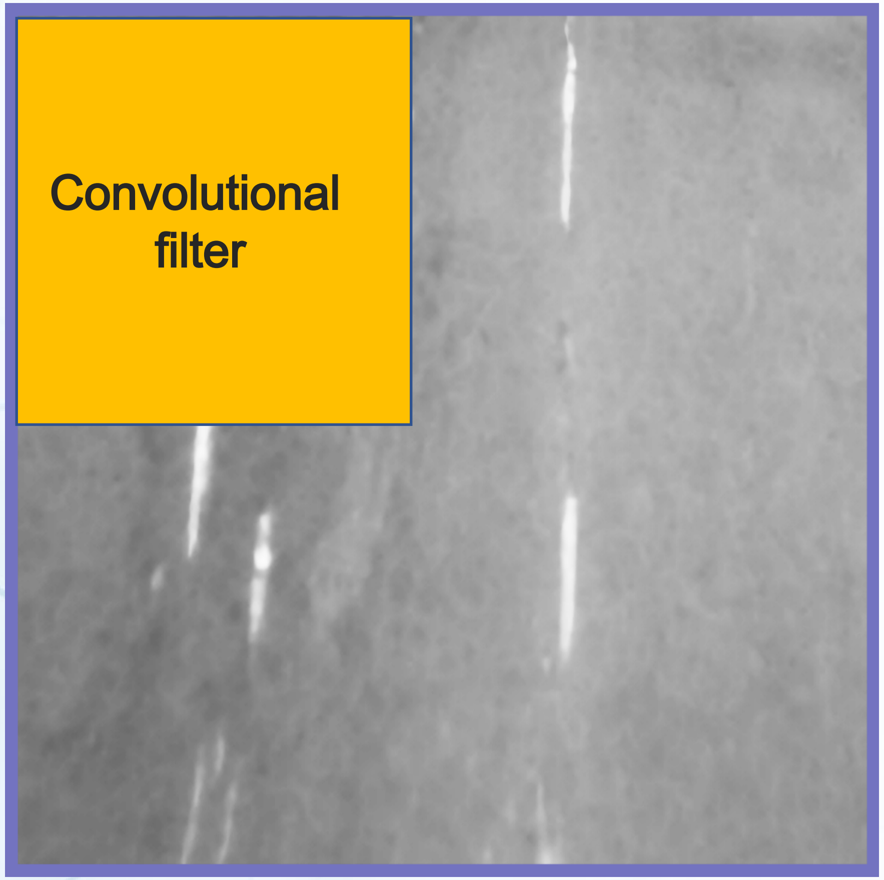


Relu



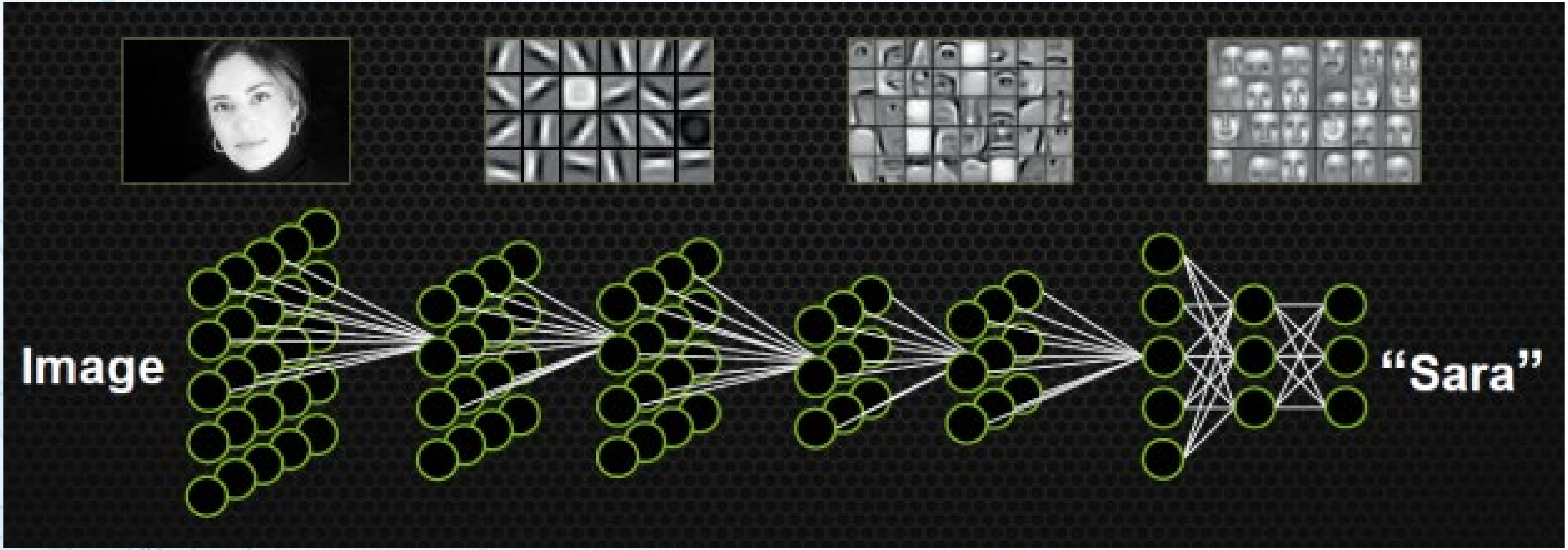
Using non-linear activations any function can be approximated

Convolutional layers



Convolutions allow us to filter using less parameters and have symmetry!

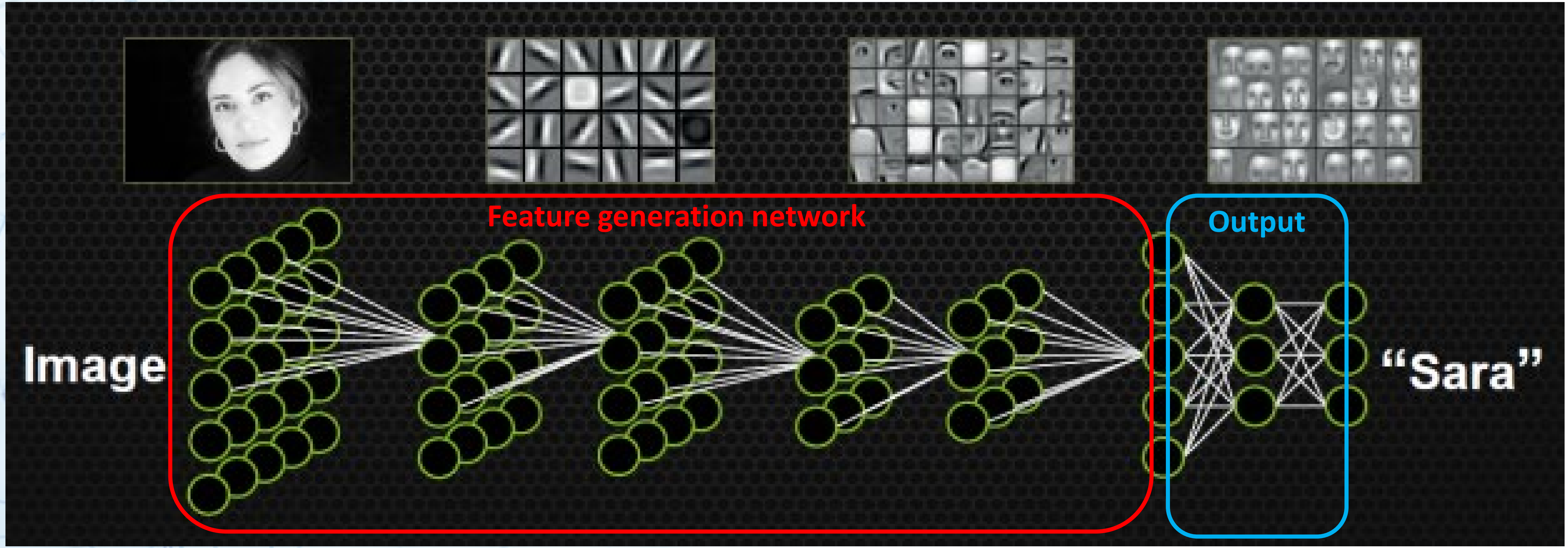
Deep learning



Source: NVIDIA Deep learning training

Features are engineered for you! but need lots of data… or do you?

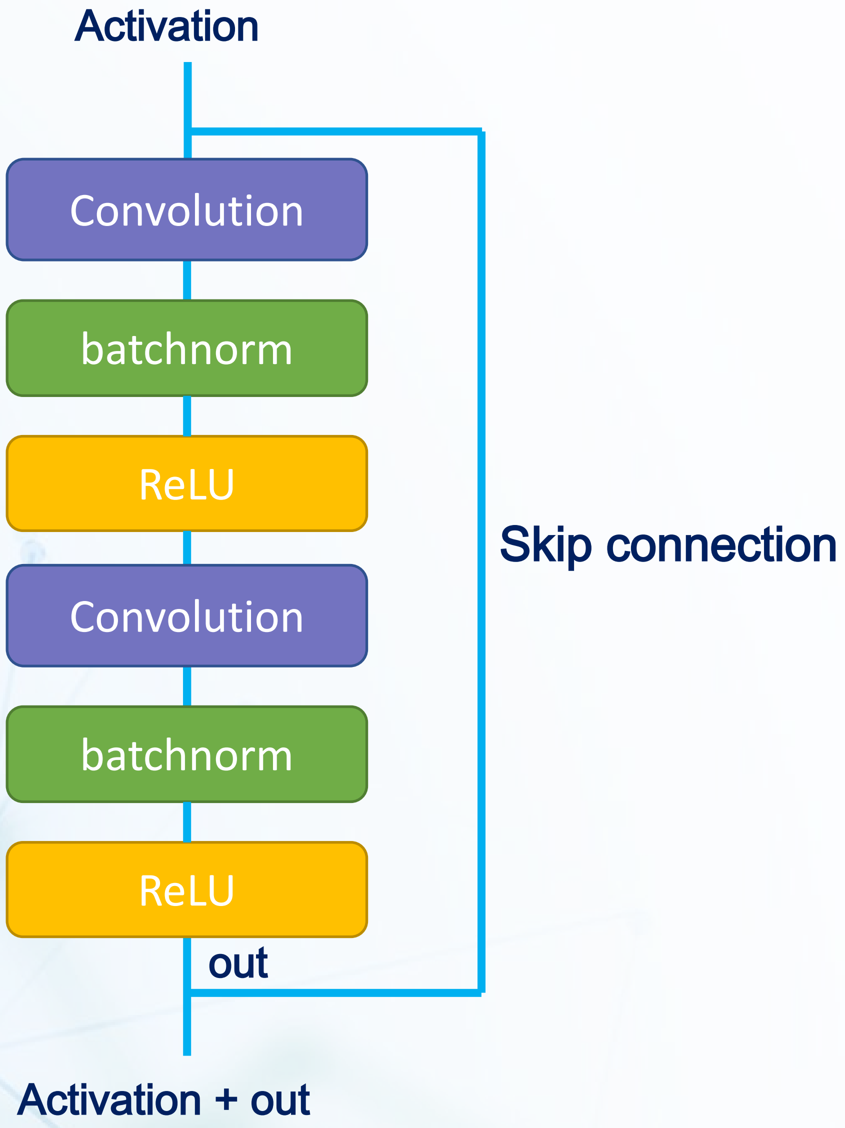
Transfer learning



Source: NVIDIA Deep learning training

With transfer learning we train on large datasets and finetune on small ones

Residual networks



Normalization and skip connections stabilize training

This session

- Analyze the images
- Choose a model
- Optimize the model
- Evaluate the results
- Interpret with explainable AI

Part of the pipeline needs to run in production

Requesting a job

Jupyter Lab

This app will launch a Jupyter Lab server on one or more nodes.

Cluster

dodrio cpu_rome

Time (hours)

12

Number of nodes

1

Number of cores per node

32 (quarter)

Mode

JupyterLab version

3.1.6 GCCcore 11.2.0

Custom code

```
source /dodrio/scratch/projects/explor_2022_008/init.sh
```

This code is executed before the JupyterLab is started. Primarily used for extra modules you want to load or swap.

Extra Jupyter Arguments

```
--notebook-dir=$PDIR/$USER
```

Project account

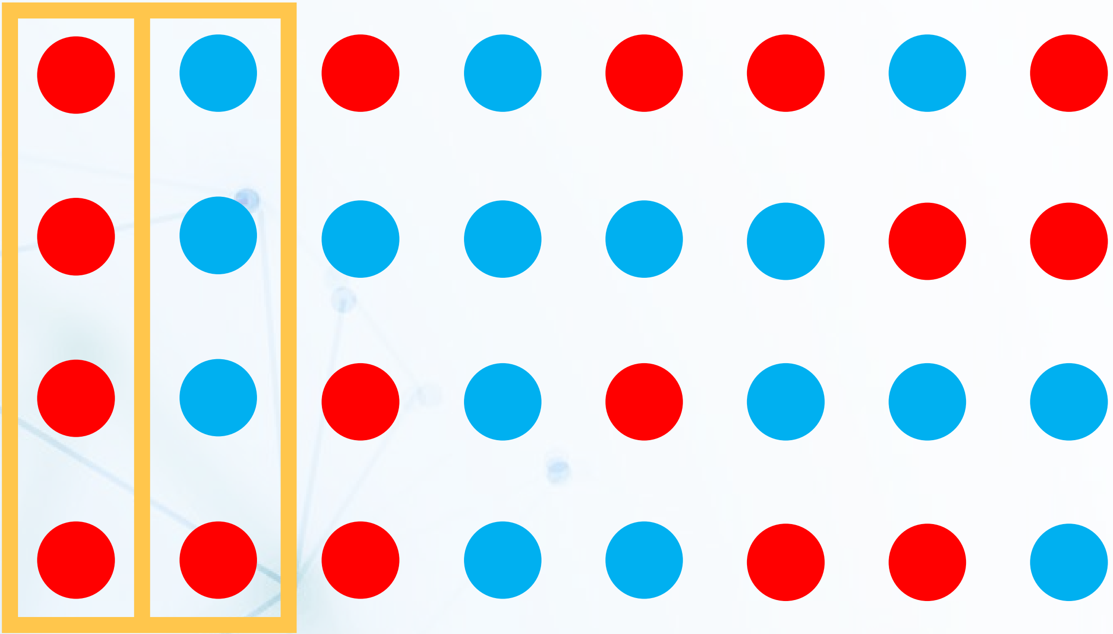
explor_2022_008

Extra sbatch arguments

```
--reservation=MaterialsAI-GPU
```

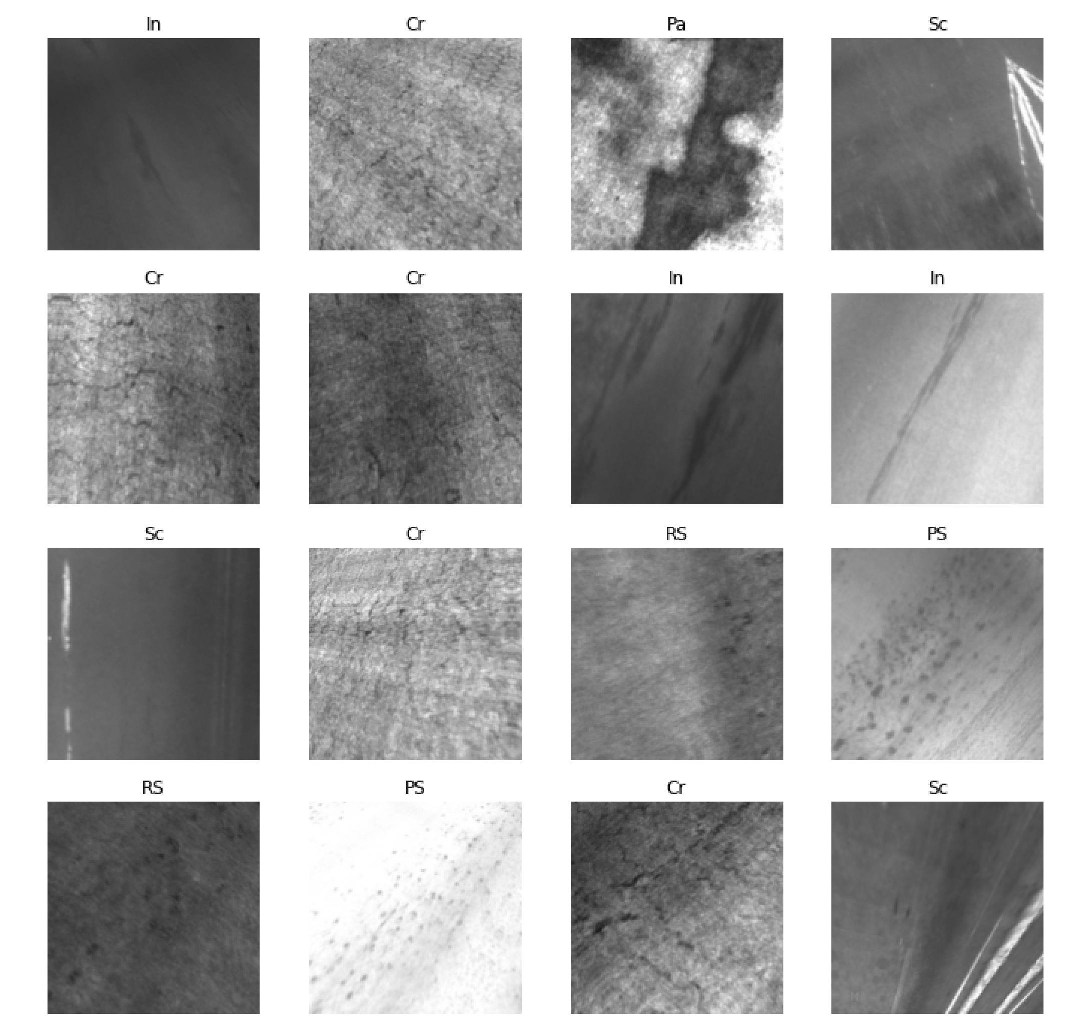
Log in at https://tier1.hpc.ugent.be

Training a neural network



Batches should contain enough variation

Augmentation



Adding some distortion to images improves our model's robustness

Loss functions

Regression

- MSE (L2)

$$\frac{\sum_N (target - pred)^2}{N}$$

- MAE (L1)

$$\frac{\sum_N |target - pred|}{N}$$

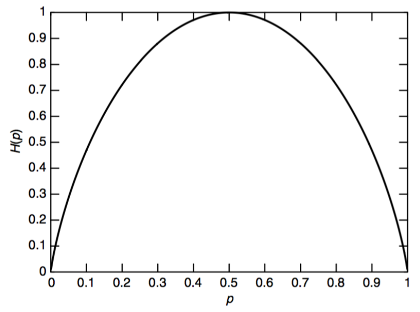
- Custom weights
- …

Classification

- (Binary) Cross entropy,

$$-target * \log(prob\ pred)\ (target=1)$$
$$+(1 - target) * \log(1 - prob\ pred)\ \ (target = 0)$$

Summed over classes if multiclass

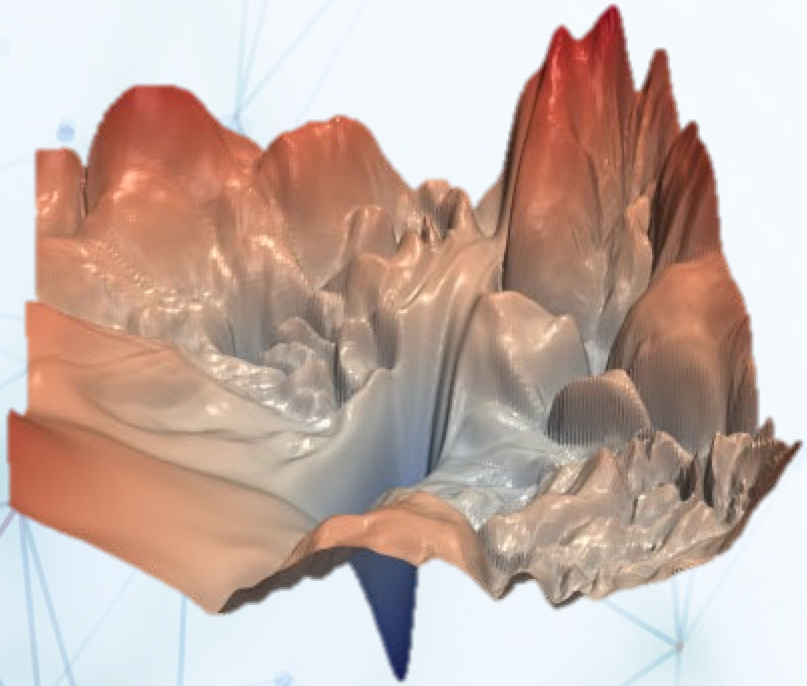


- For segmentation applied per pixel
- …

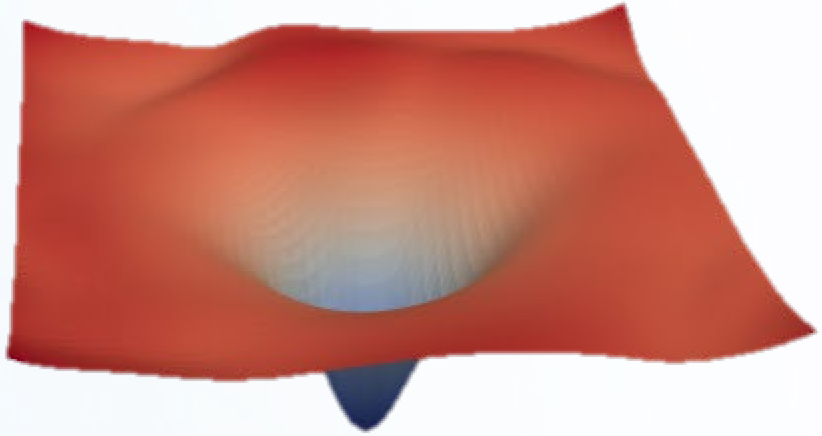
The right metric guides the optimizer to the right goal

Loss surfaces

Normal net



Resnet

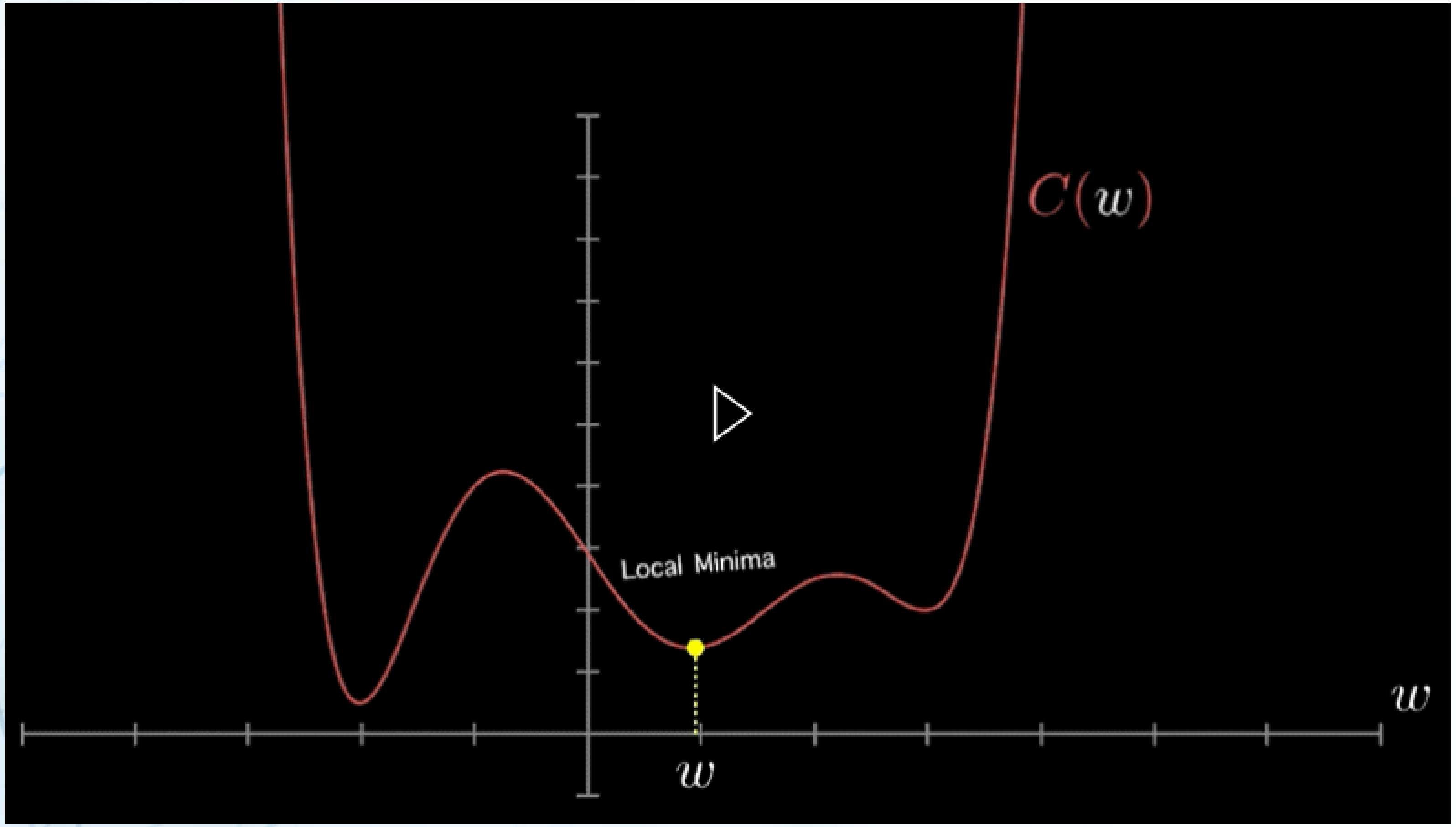


https://proceedings.neurips.cc/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf

https://izmailovpavel.github.io/curves_blogpost/

Loss functions have complex surfaces with millions of parameters

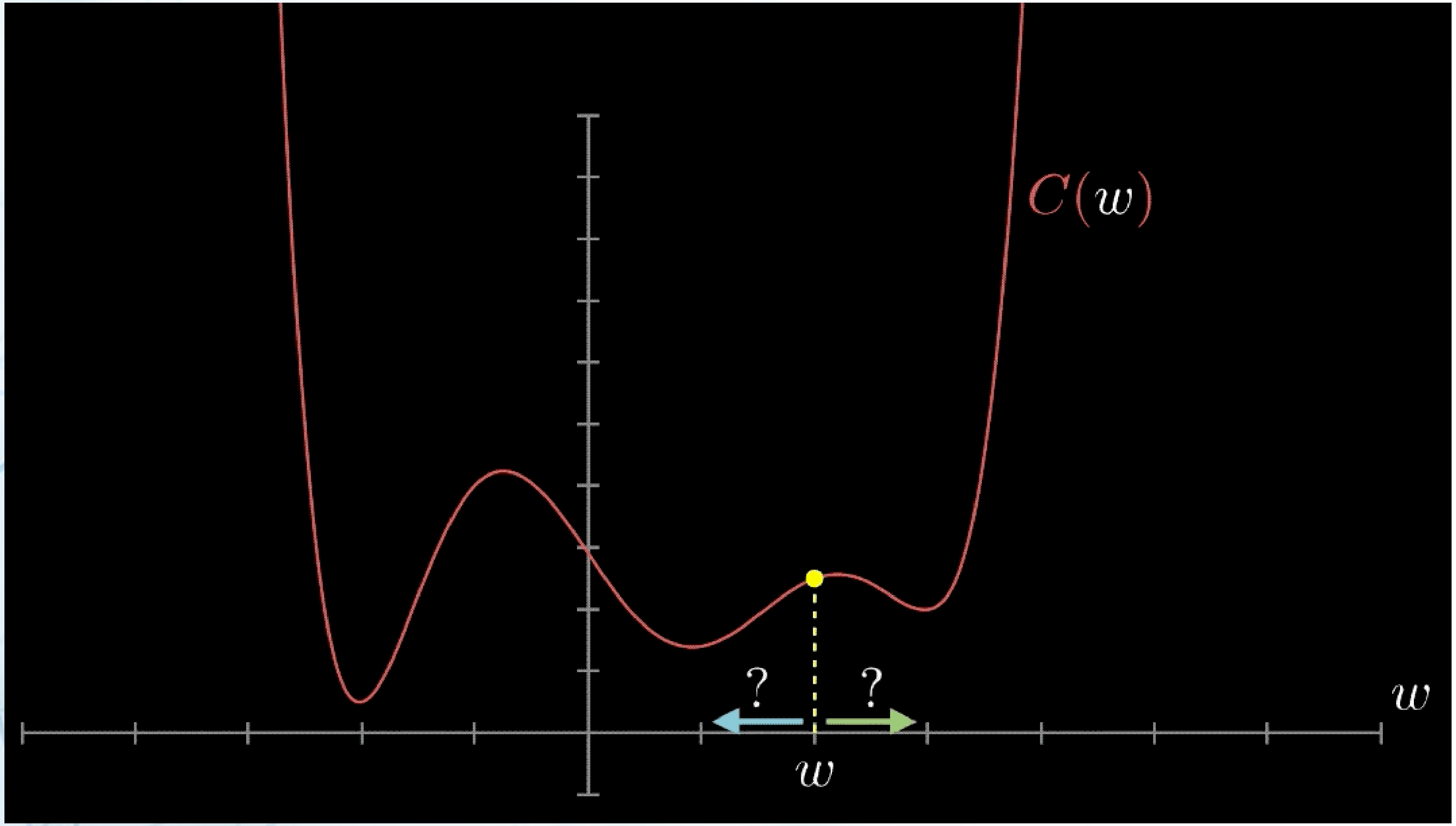
Optimizing



3blue1brown - https://mlfromscratch.com/optimizers-explained/#/

Gradient descent allows us to stepwise optimize our parameters for our loss

Optimizing



3blue1brown - https://mlfromscratch.com/optimizers-explained/#/

Gradient descent allows us to stepwise optimize our parameters for our loss

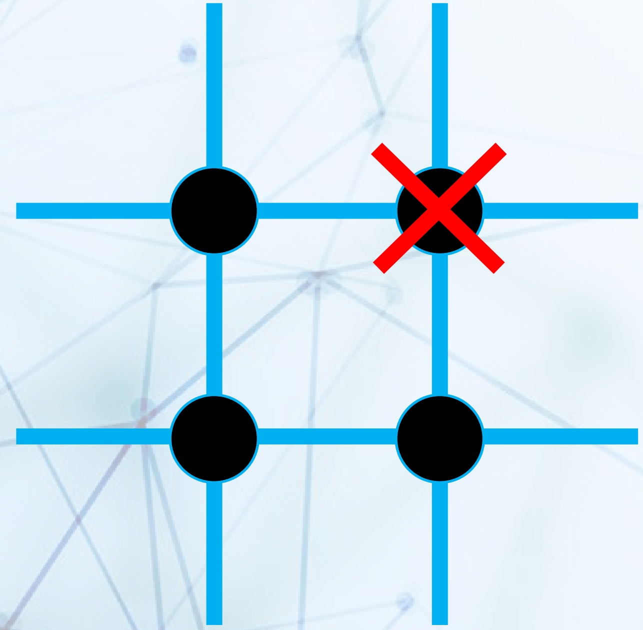
More on optimizers

- https://distill.pub/2017/momentum/ - try this yourself
- https://ruder.io/optimizing-gradient-descent/
- https://towardsdatascience.com/a-visual-explanation-of-gradient-descent-methods-momentum-adagrad-rmsprop-adam-f898b102325c
- https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd

Adding some distortion to images improves our model’s robustness

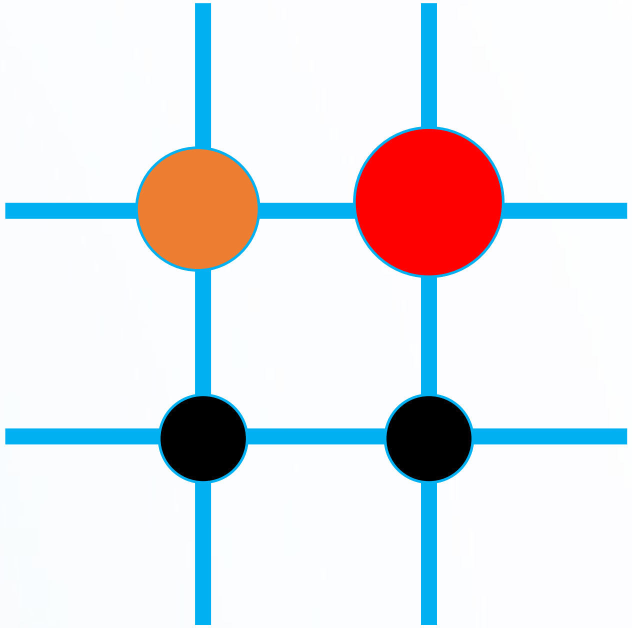
Regularization

Dropout



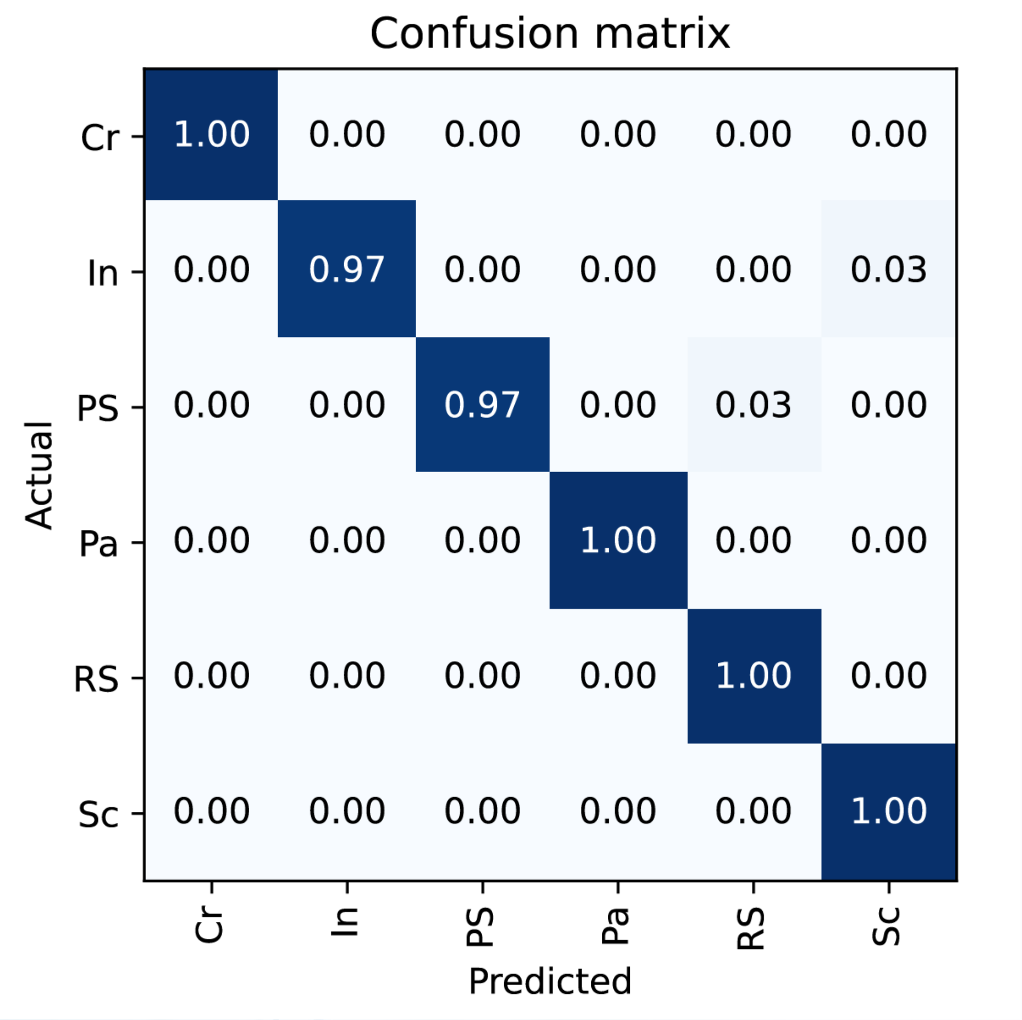
Randomly delete neurons doing training

Weight decay



Add the norm of weights to loss

Metrics

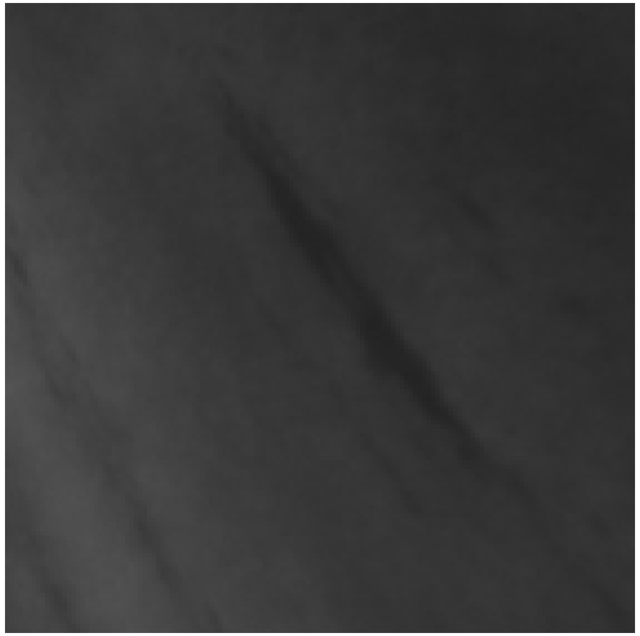


Metrics can, but don't have to be the same as a loss function (no backprop)

Top losses

Predicted Actual Loss Probability

Sc/In / 2.82 / 0.94



The best guess is not necessarily a good guess

Top losses

Softmax

1 1 8

$$softmax(a) = \frac{e^{a_j}}{\sum_j e^{a_j}}$$

0 0 1

Always gives the best prediction

Sigmoid

1 1 8

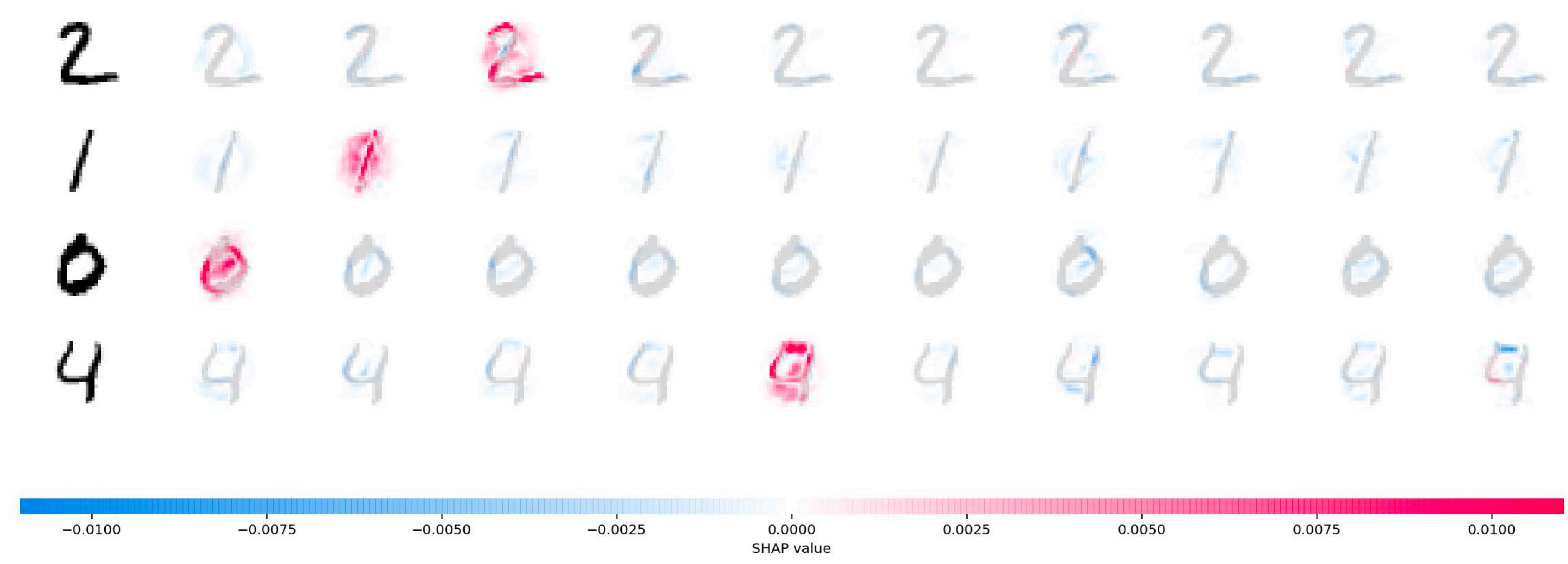
$$sigmoid(a_j) = \frac{1}{1 - e^{-a_j}}$$

0.7 0.7 1

Gives probability per class

SHAP

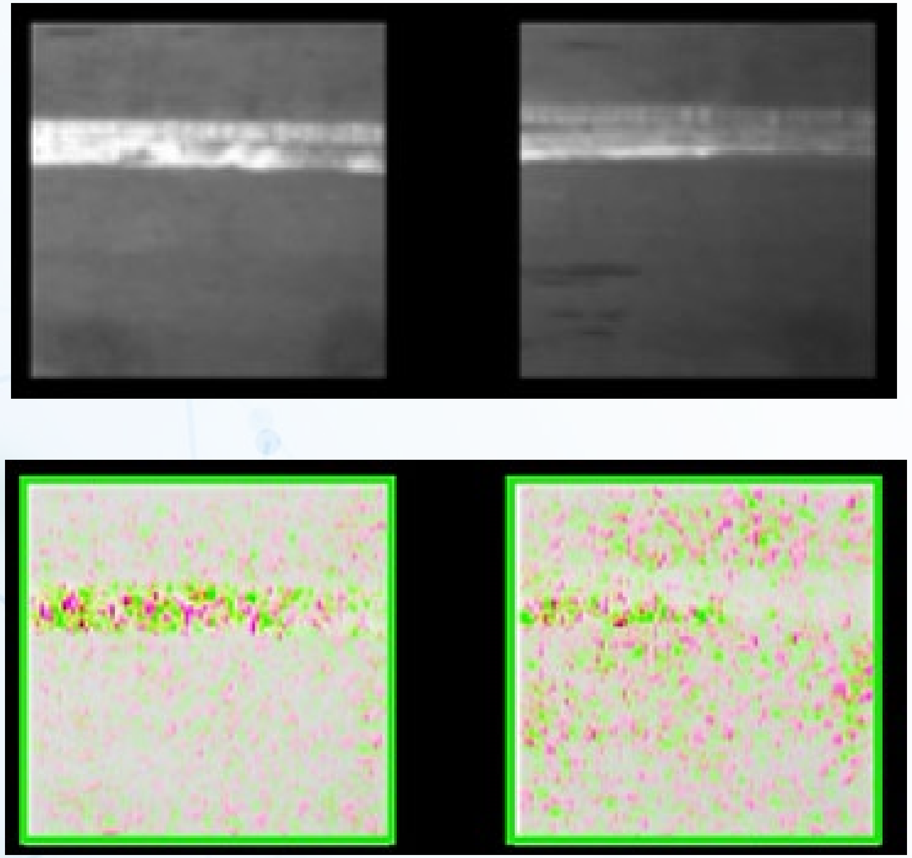
Our input features are pixels, can we trace them to the output?



https://github.com/slundberg/shap

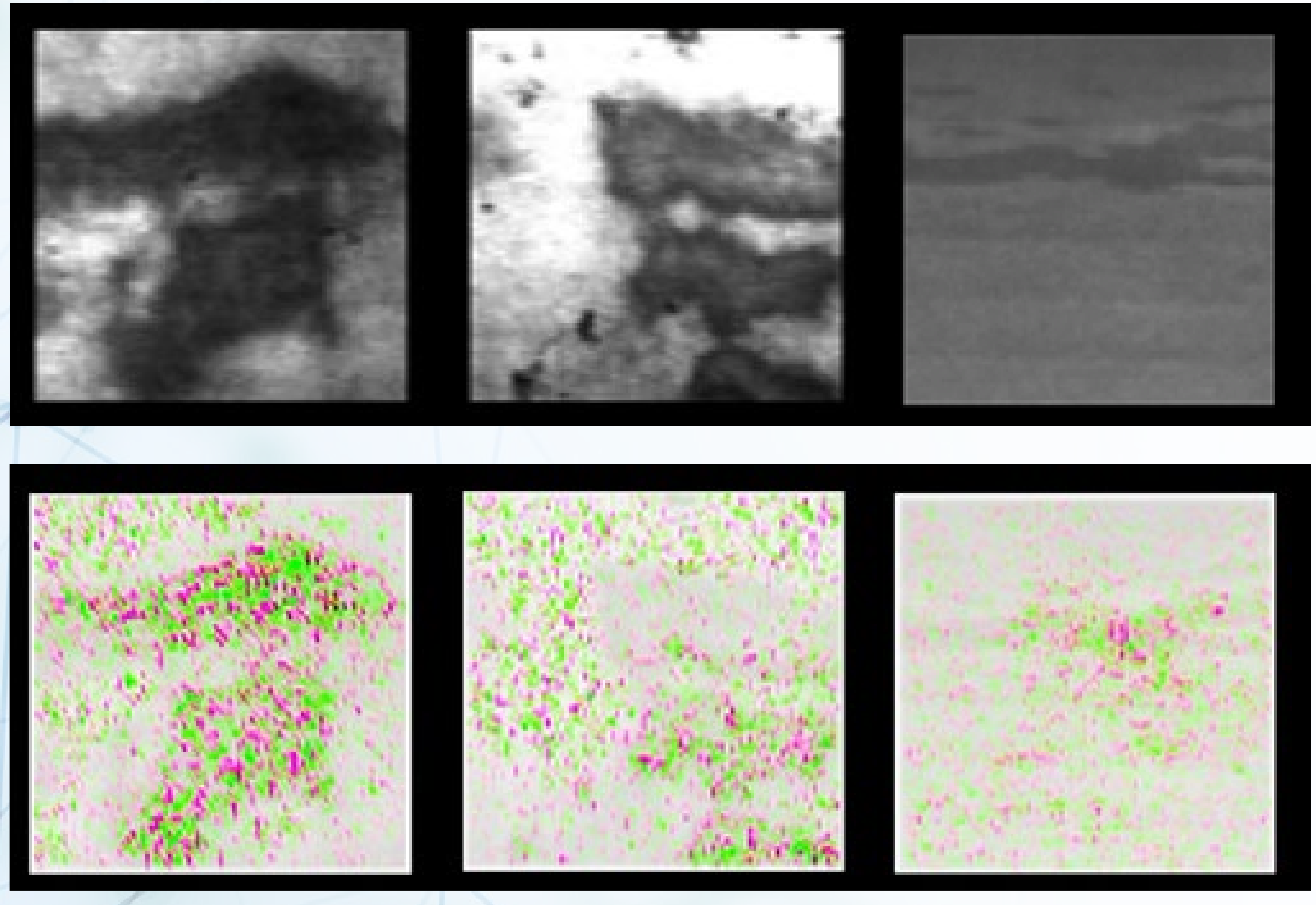
Yes, by comparing to baseline images we can approximate shap values using grads

SHAP: scratches



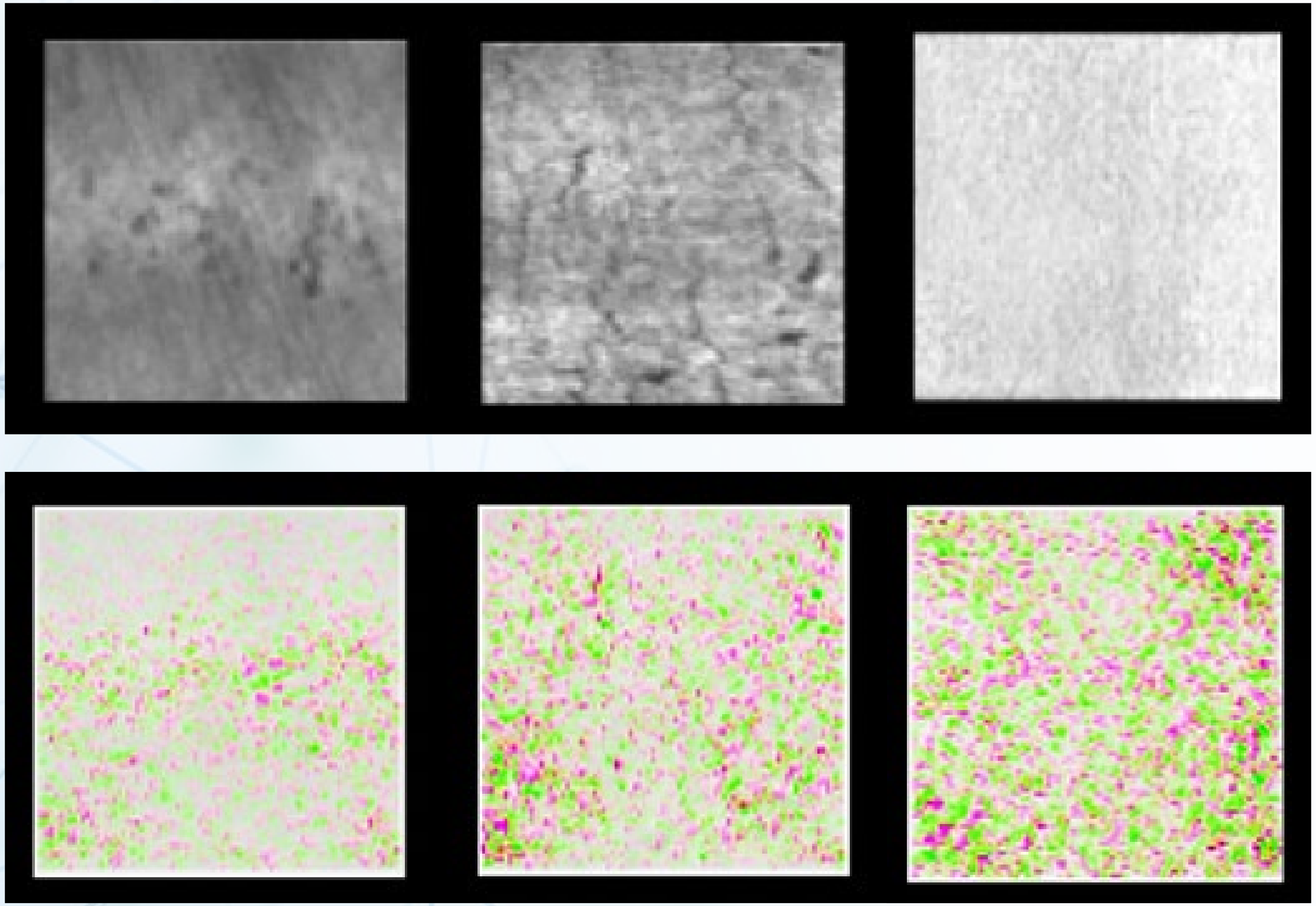
Scratch is clearly highlighted

SHAP: large defects



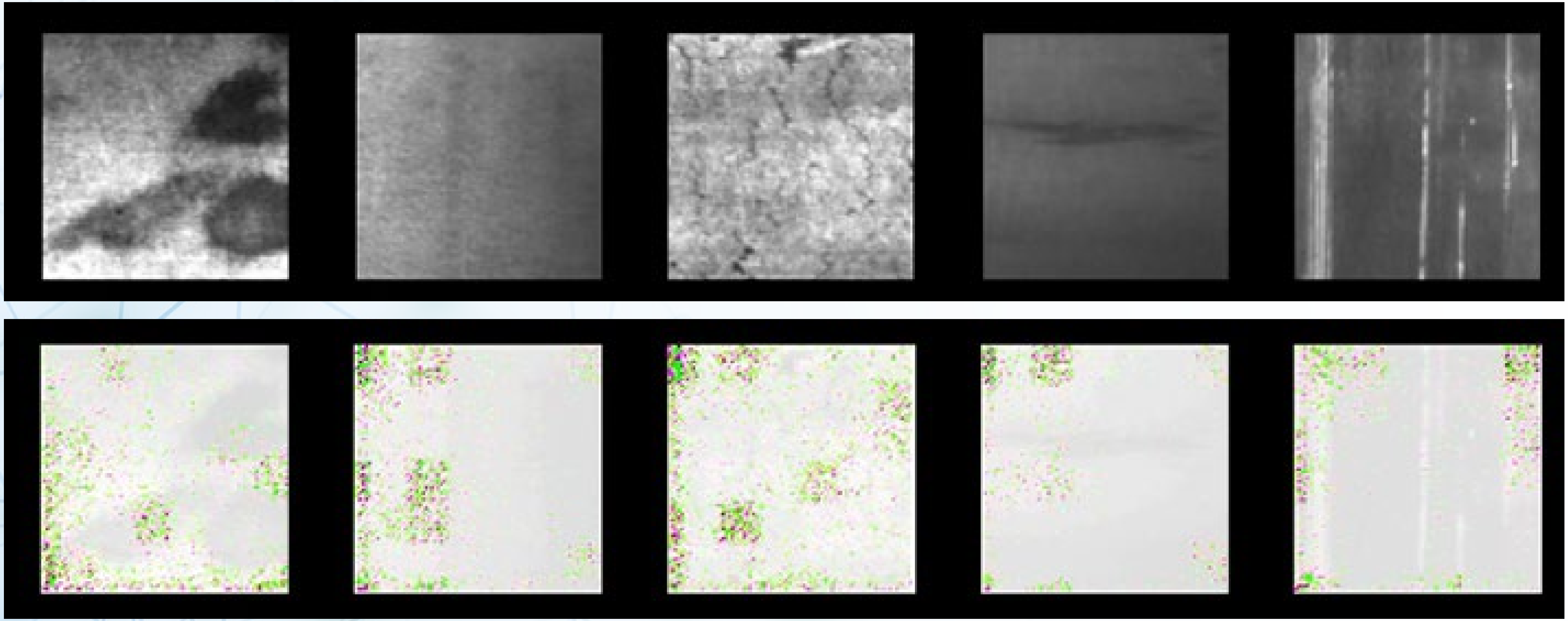
Both dark and light regions used

SHAP: distributed defects



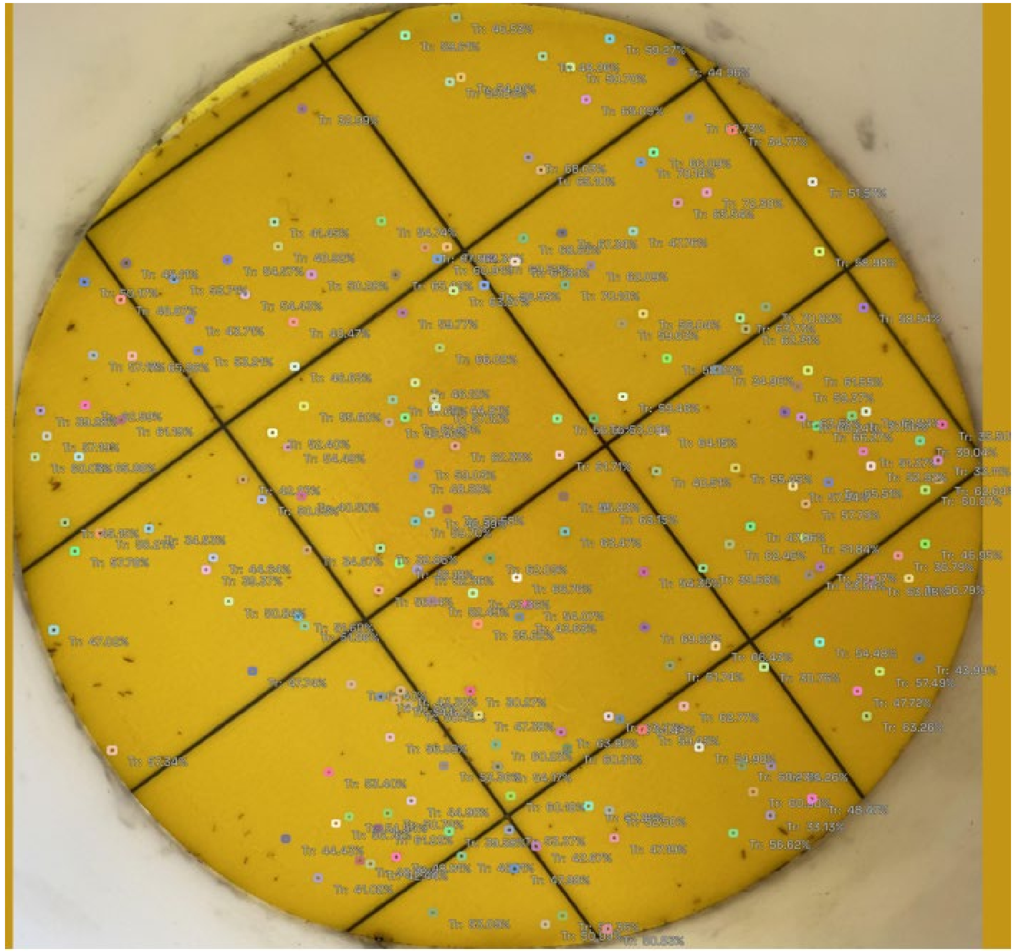
Distributed activation regions

Detecting problems with SHAP



Edges seem more important than they should be

How does this work in production?



AI FOR MATERIALS INDUSTRY

How does this work in production?

API

Queue

(Accelerated) Model

(Accelerated) Model

(Accelerated) Model

API lets users schedule tasks for the model

How does this work in production?

Cloud (VSC, Google, Azure, AWS)

Edge

Cloud allows global deployment, but requires communication

INVESTEERT IN
JOUW TOEKOMST

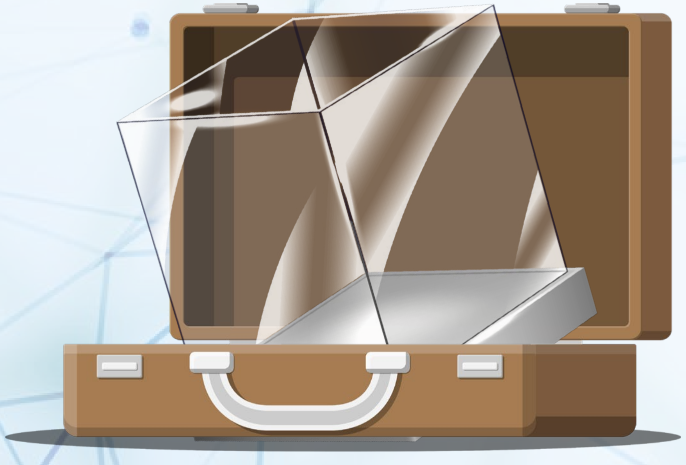
ESF Europese Unie Vlaanderen is werk

AI FOR MATERIALS INDUSTRY

View more online

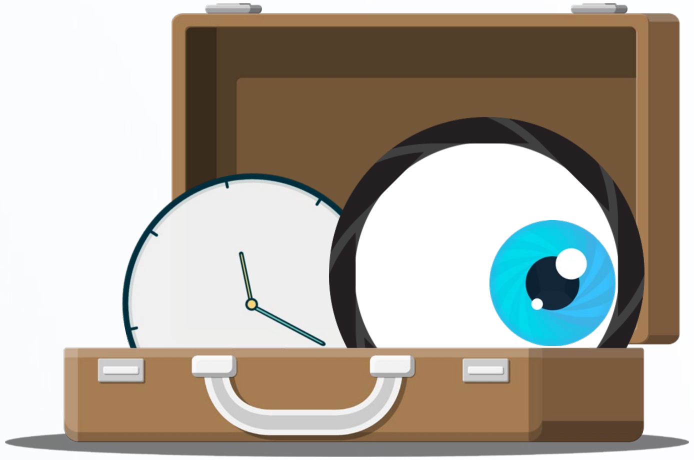
https://ai4mi.epotentia.com

Glass



Materials discovery

Manufacturing



Sensor data

GHENT UNIVERSITY

ePotentia
Data•Science•Hosting

ocas

umicore
materials for a better life

Sensor data

AGFA Agfa