Anomaly Detection in Images

Giacomo Boracchi,

Politecnico di Milano, DEIB.

http://home.deib.polimi.it/boracchi/

Diego Carrera,

System Research and Applications, STMicroelectronics, Agrate Brianza

September 9th, 2019

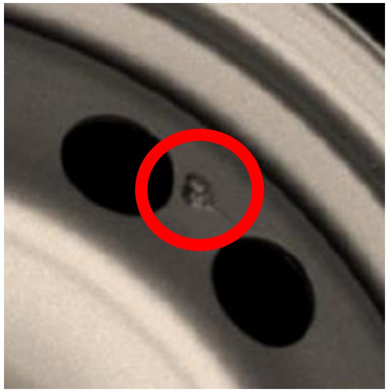
ICIAP 2019, Trento, Italy

ANOMALY DETECTION PROBLEMS

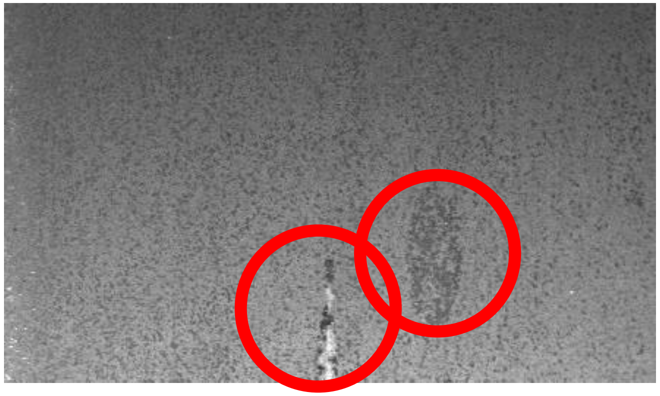
Anomaly detection problems are ubiquitous in imaging applications.

Relevant examples spans from quality inspection and health

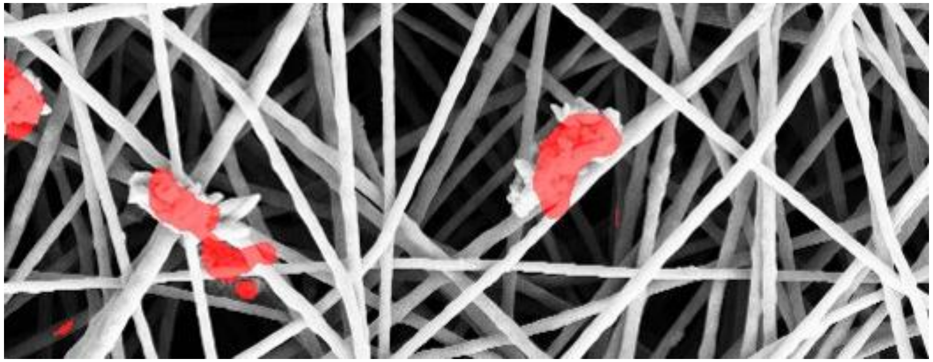
manufacturing



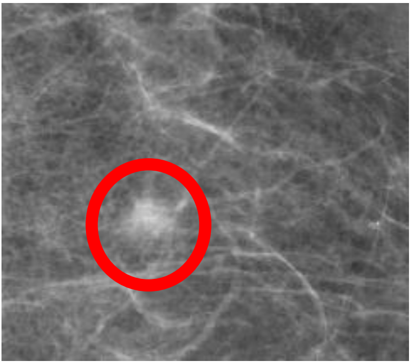
manufacturing

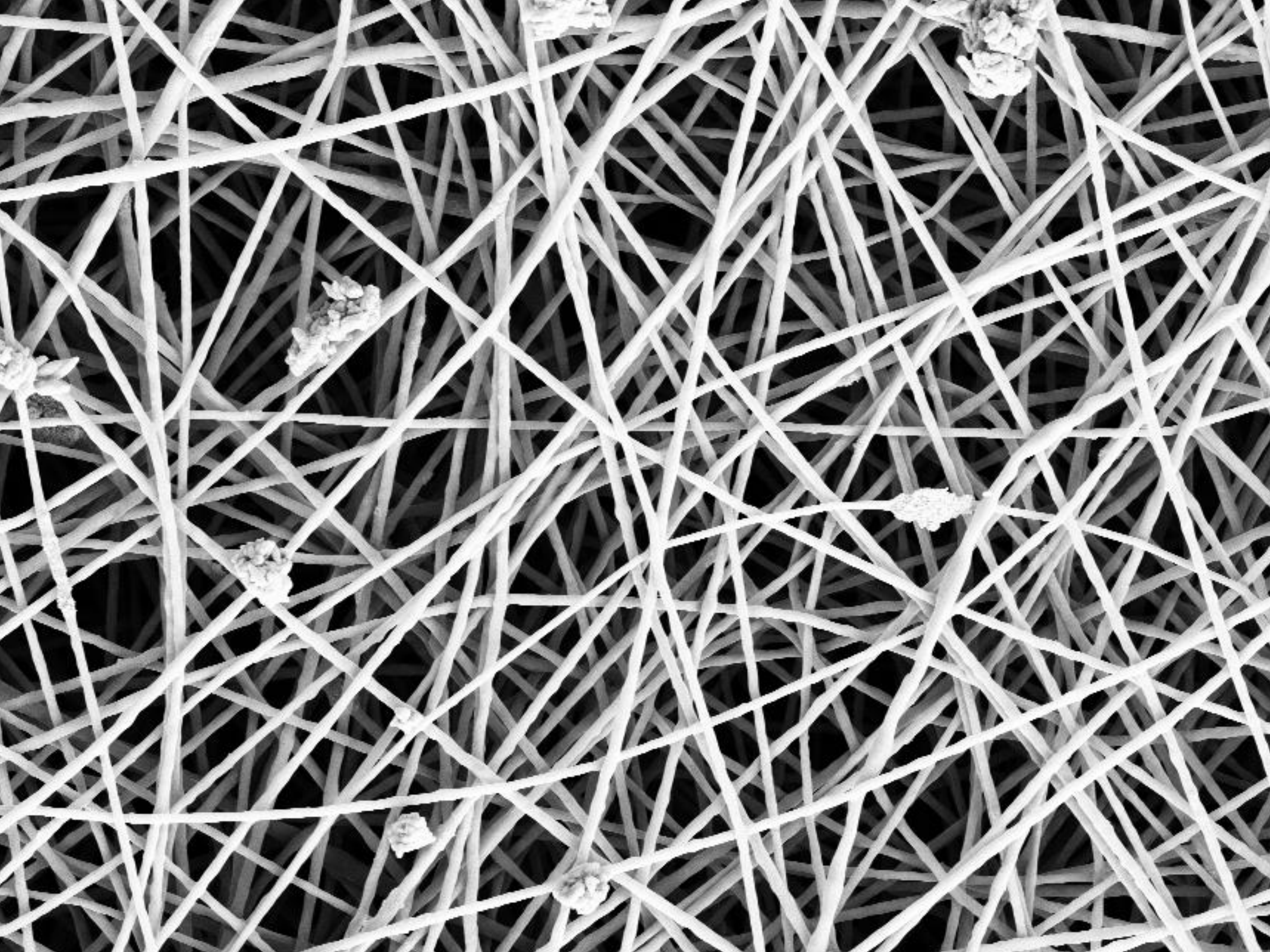


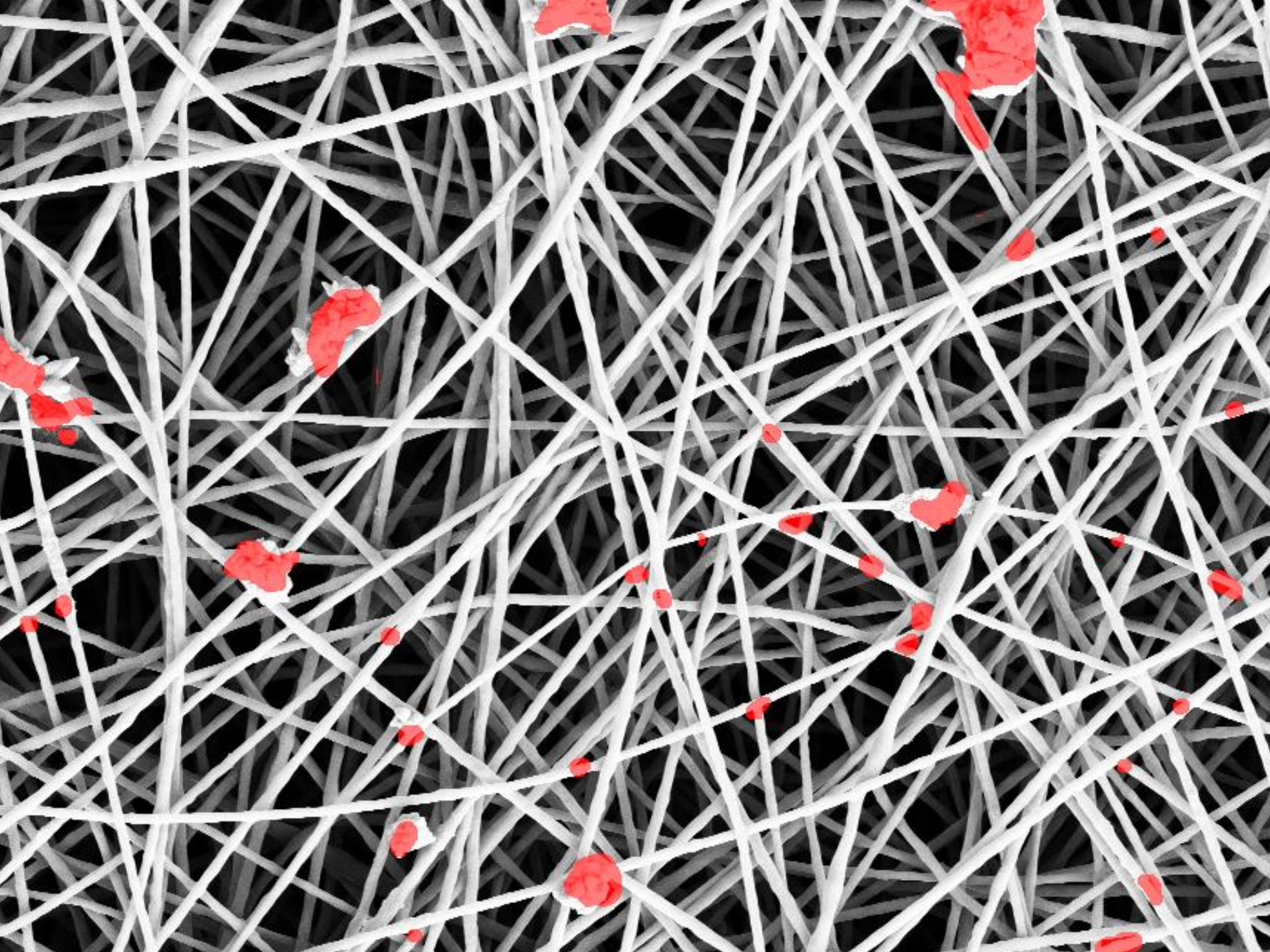
nanofiber production



mammograms

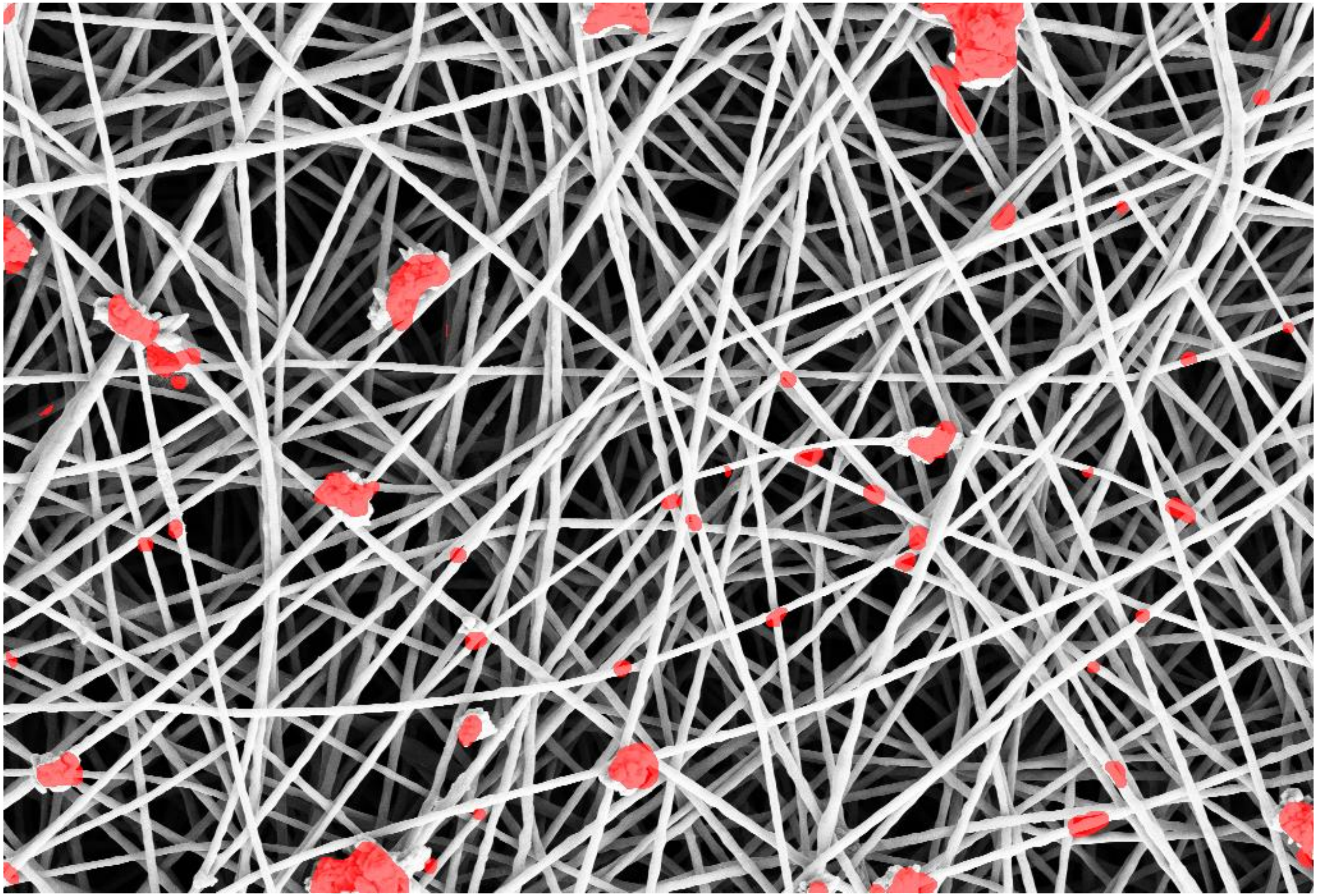






… AN ANOMALY-DETECTION PROBLEM

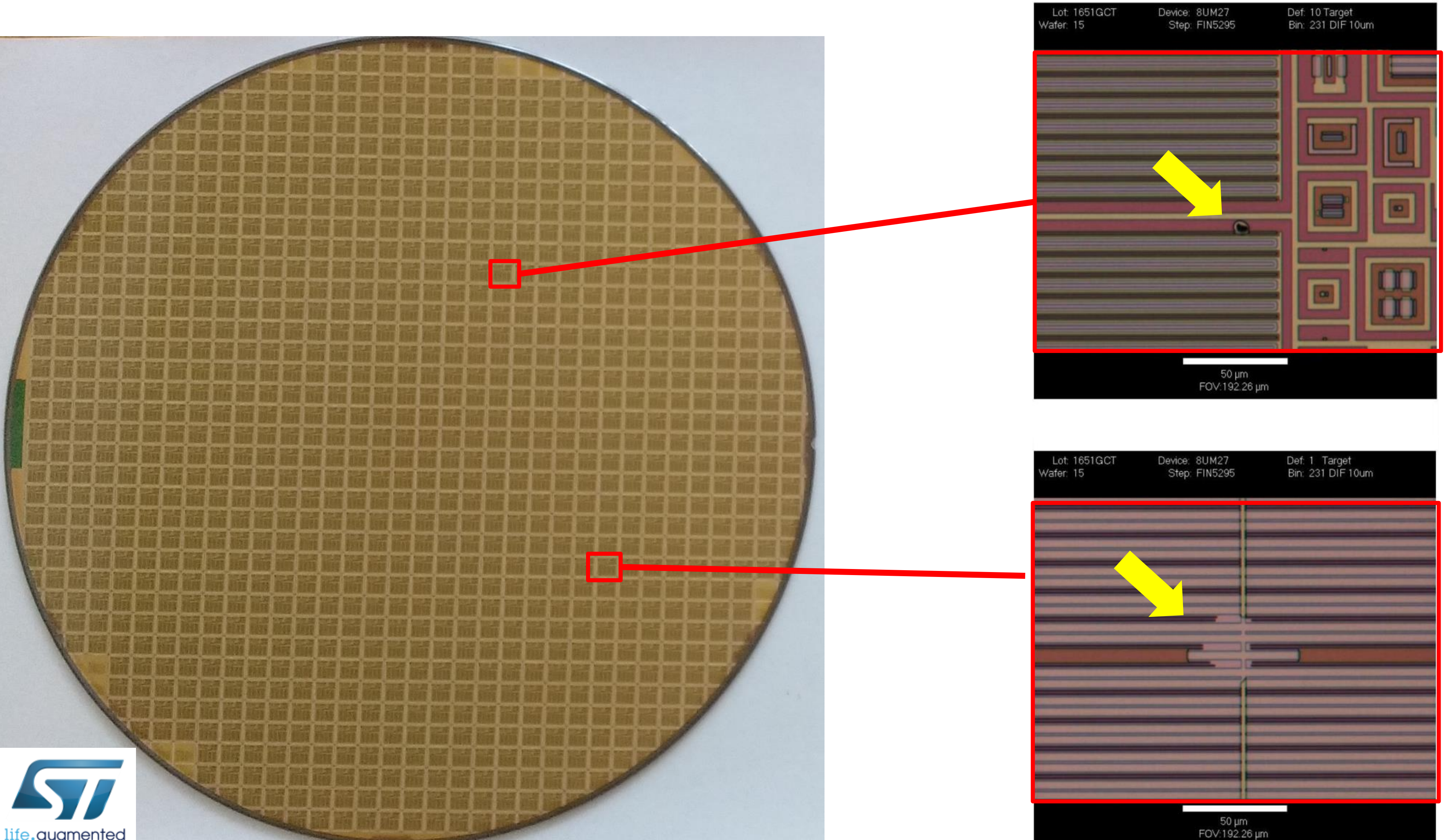
Quality Inspection Systems: monitoring the nanofiber production



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

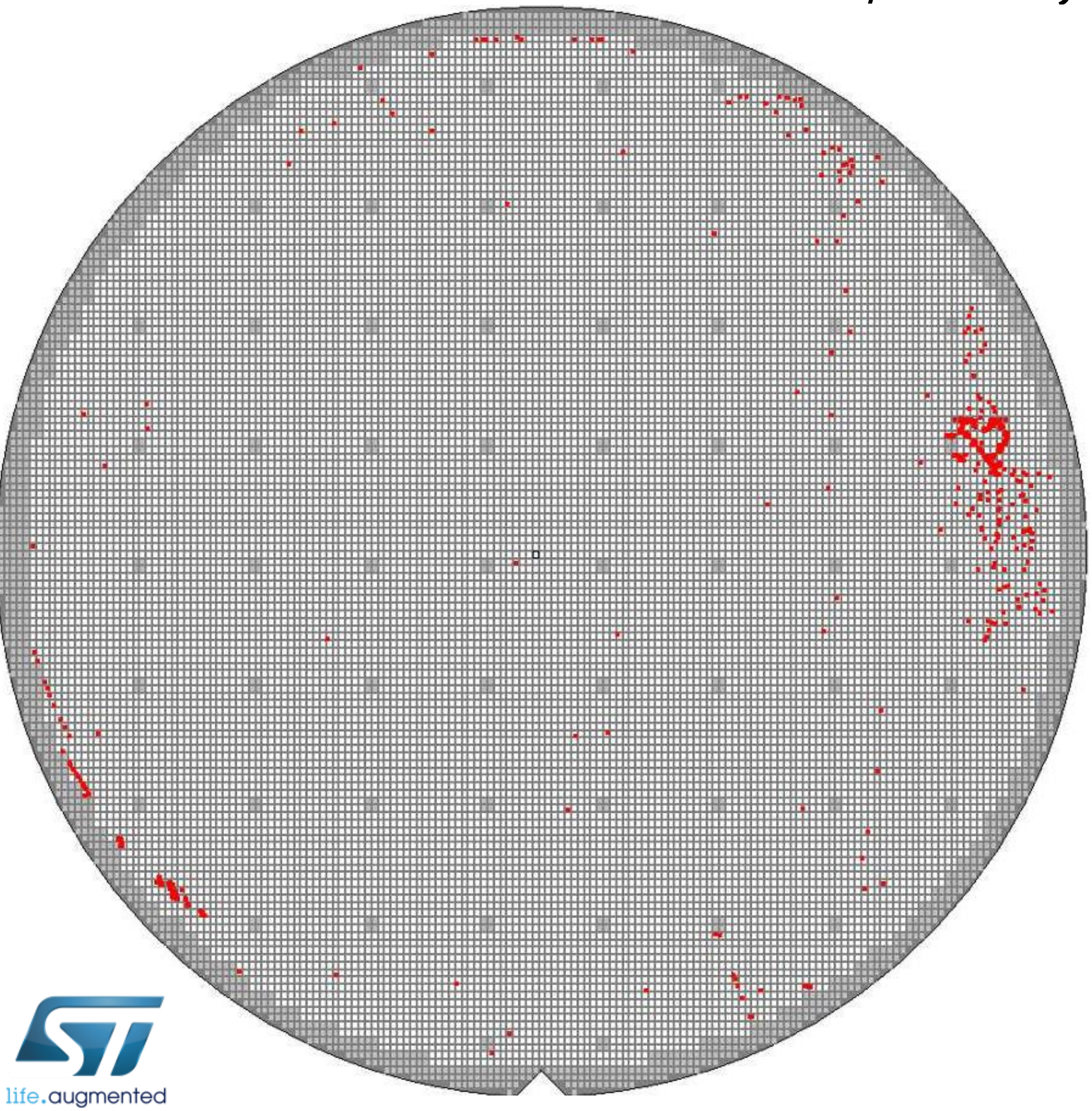
… AN ANOMALY-DETECTION PROBLEM

Detection of anomalies in chip production

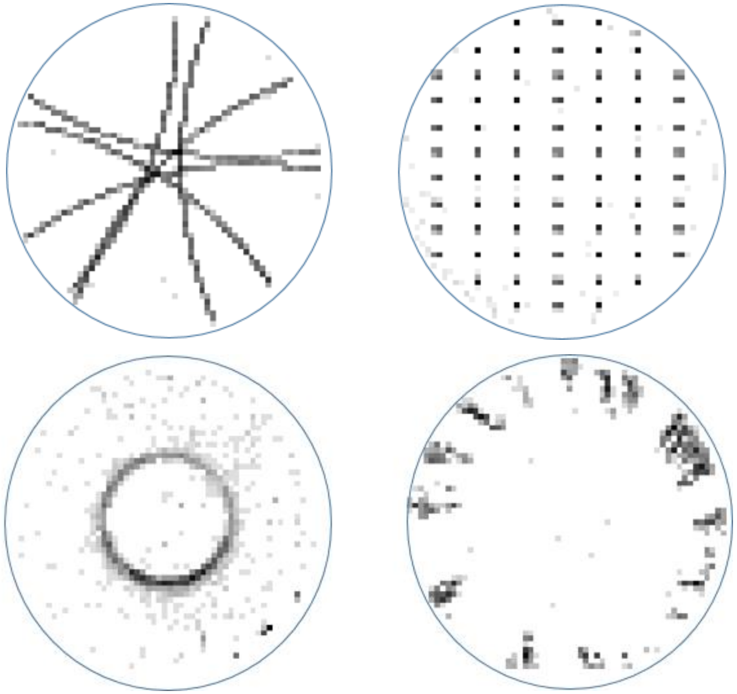


… A DETECTION PROBLEM

Detect/Identify **patterns** in **wafer defect maps**



These might indicate faults, problems or malfunctioning in the chip production.

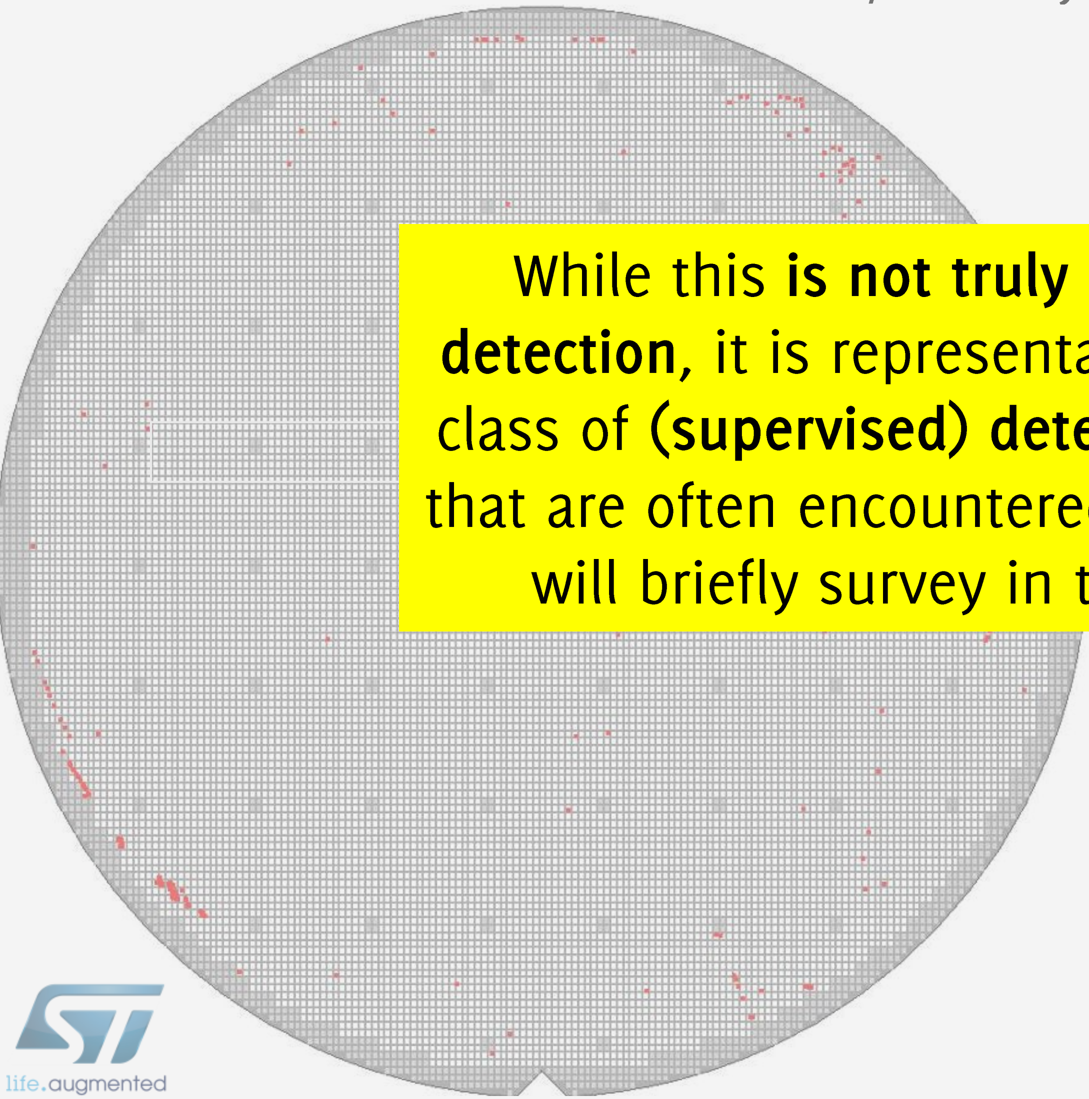


Di Bella, Carrera, Rossi, Fragneto, Boracchi *Wafer Defect Map Classification Using Sparse Convolutional Neural Networks* ICIAP09

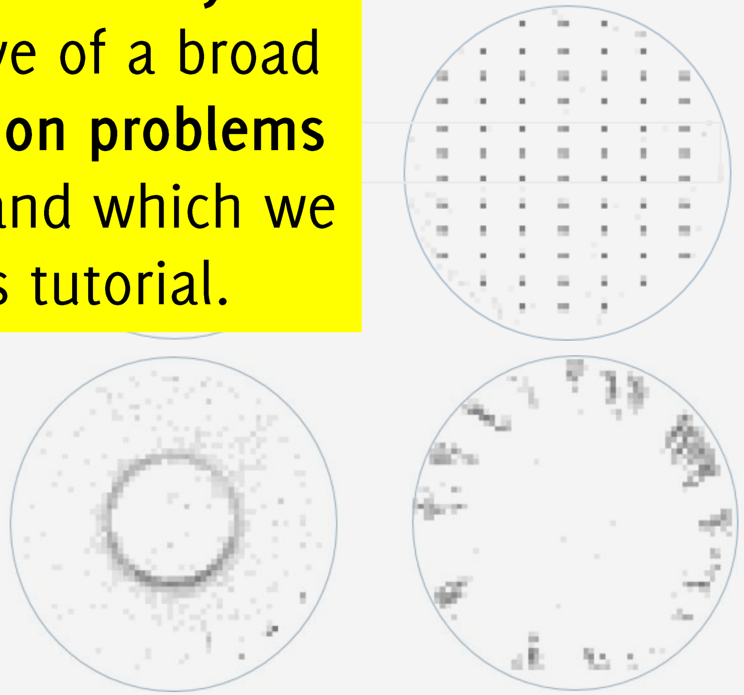
… A DETECTION PROBLEM

Detect/Identify **patterns** in **wafer defect maps**

These might indicate faults, problems or malfunctioning in the chip production.



While this **is not truly an anomaly-detection**, it is representative of a broad class of **(supervised) detection problems** that are often encountered and which we will briefly survey in this tutorial.



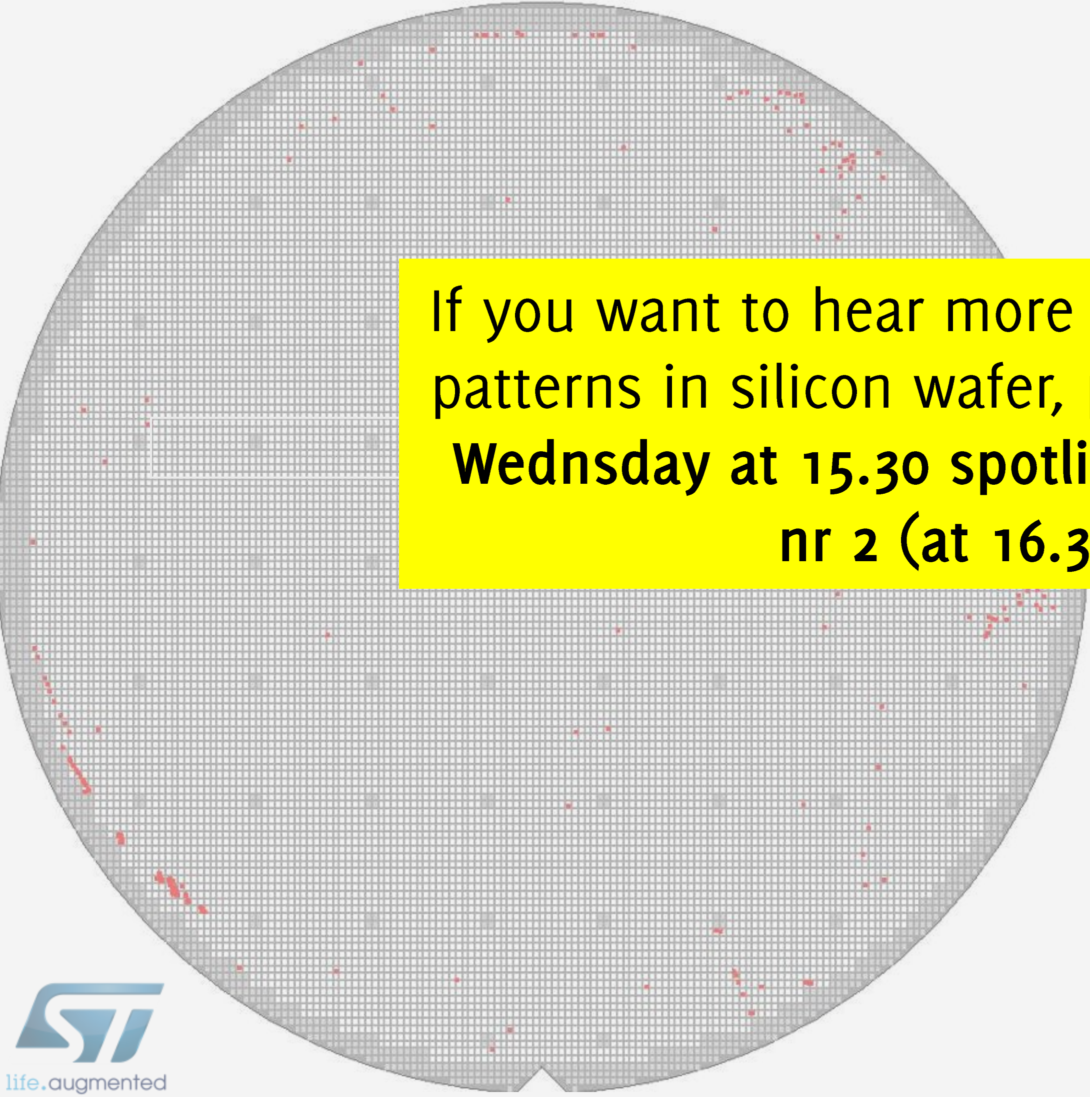
Di Bella, Carrera, Rossi, Fragneto, Boracchi *Wafer Defect Map Classification Using Sparse Convolutional Neural Networks* ICIAP09

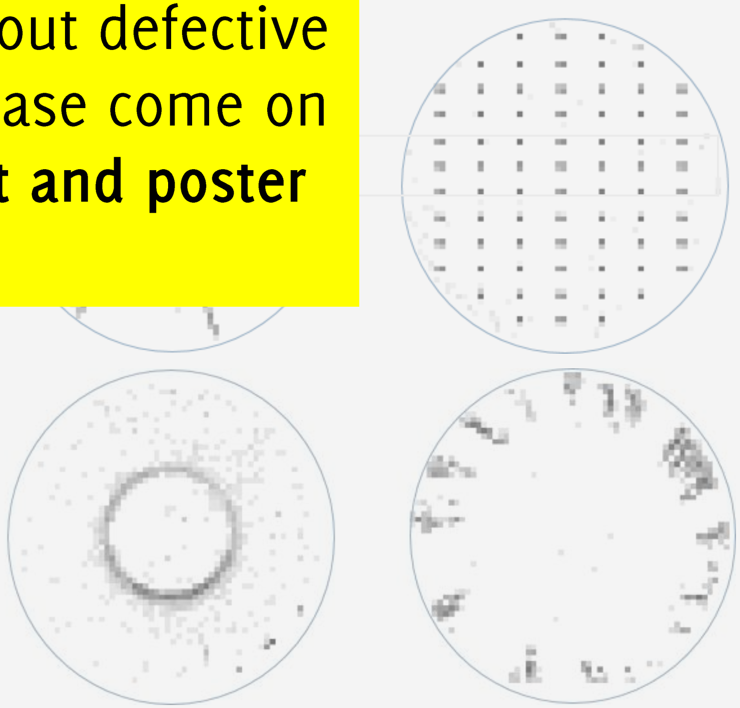
… A DETECTION PROBLEM

Detect/Identify **patterns** in **wafer defect maps**

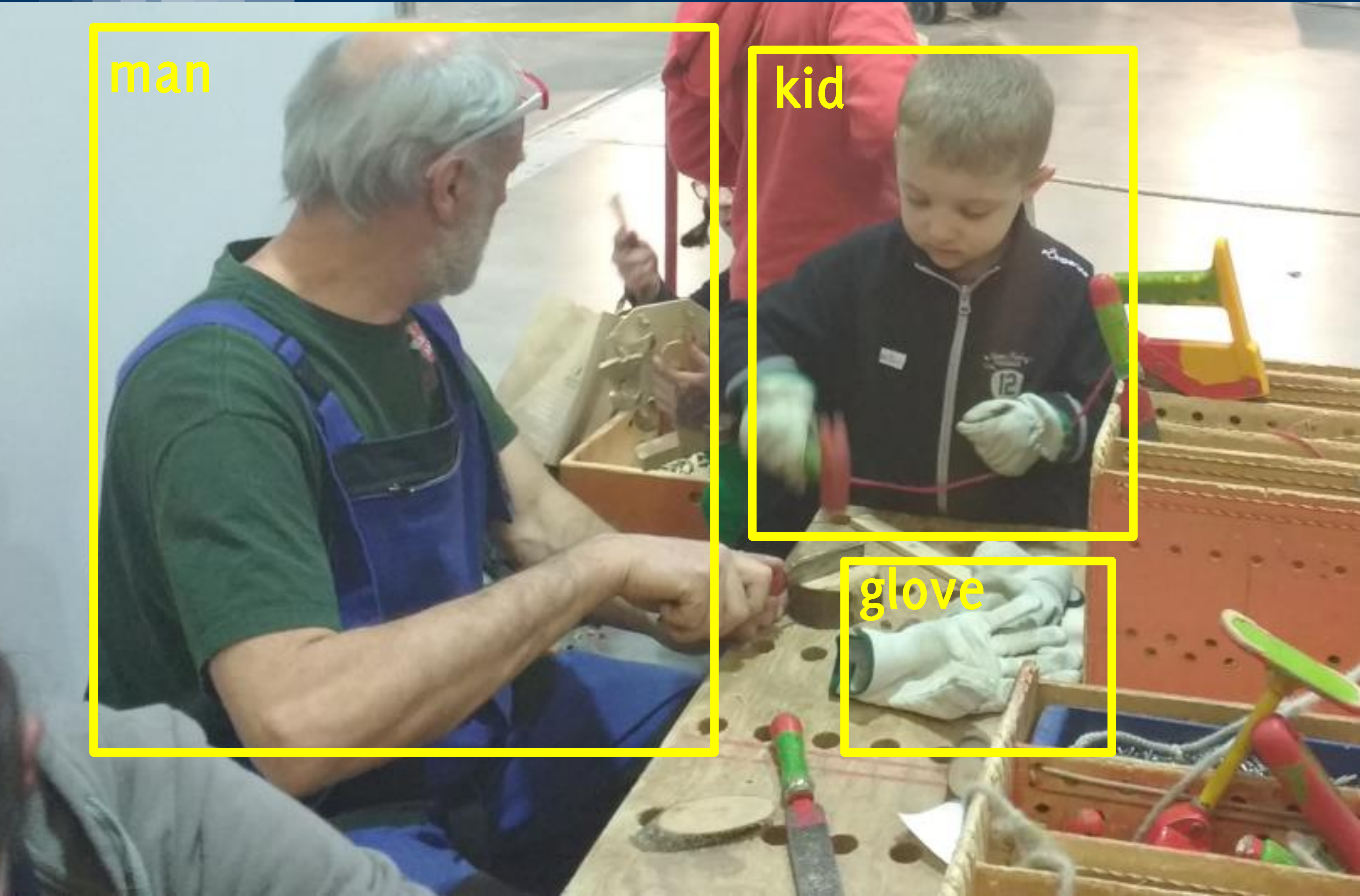
These might indicate faults, problems or malfunctioning in the chip production.

If you want to hear more about defective patterns in silicon wafer, please come on **Wednesday at 15.30 spotlight and poster nr 2 (at 16.30)**





OBJECT DETECTION IN NATURAL IMAGES

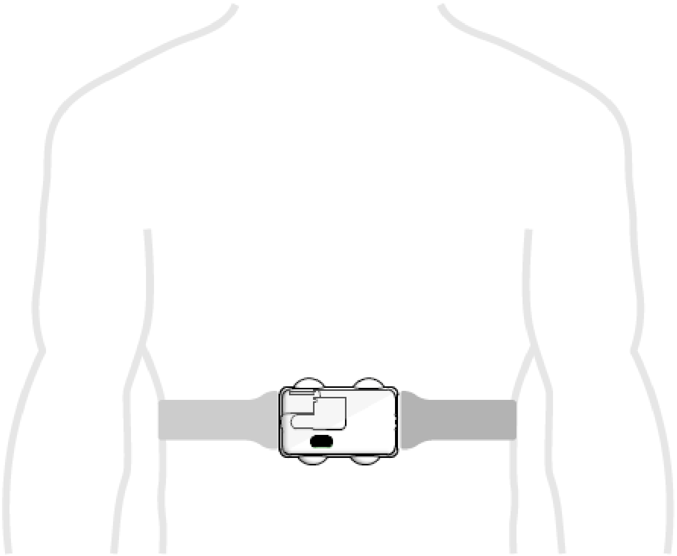


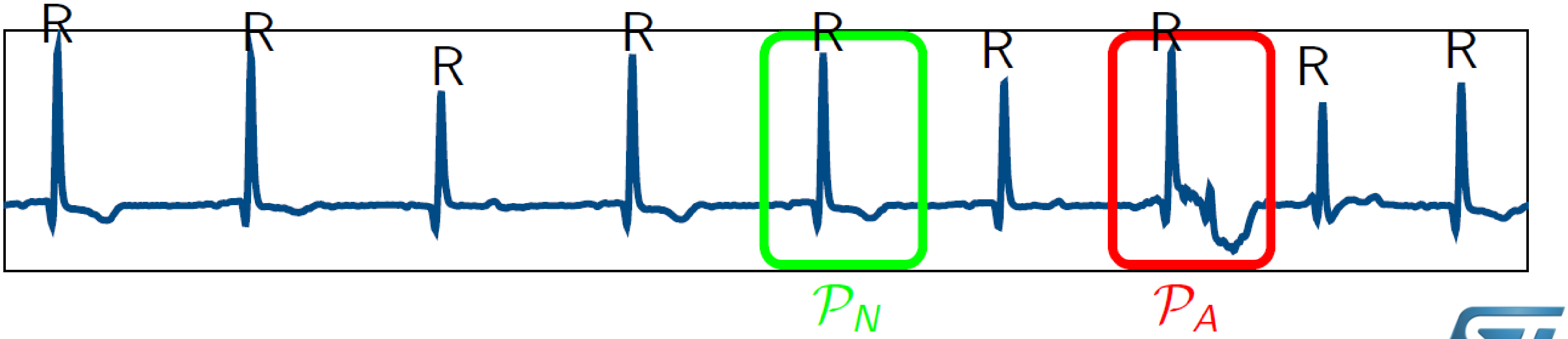
Not only images

… AN ANOMALY-DETECTION PROBLEM

Health monitoring / wearable devices:

Automatically analyze EGC tracings to detect arrhythmias or incorrect device positioning

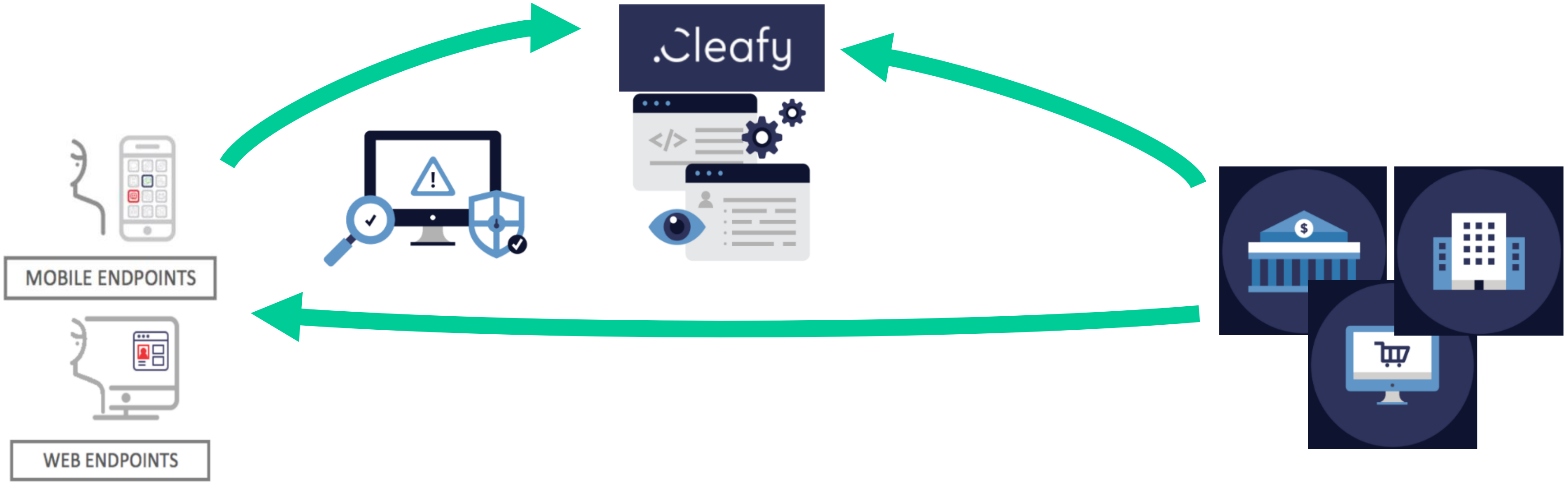




D. Carrera, B. Rossi, D. Zambon, P. Fragneto, and G. Boracchi "*ECG Monitoring in Wearable Devices by Sparse Models*" in Proceedings of ECML-PKDD 2016, 16 pages

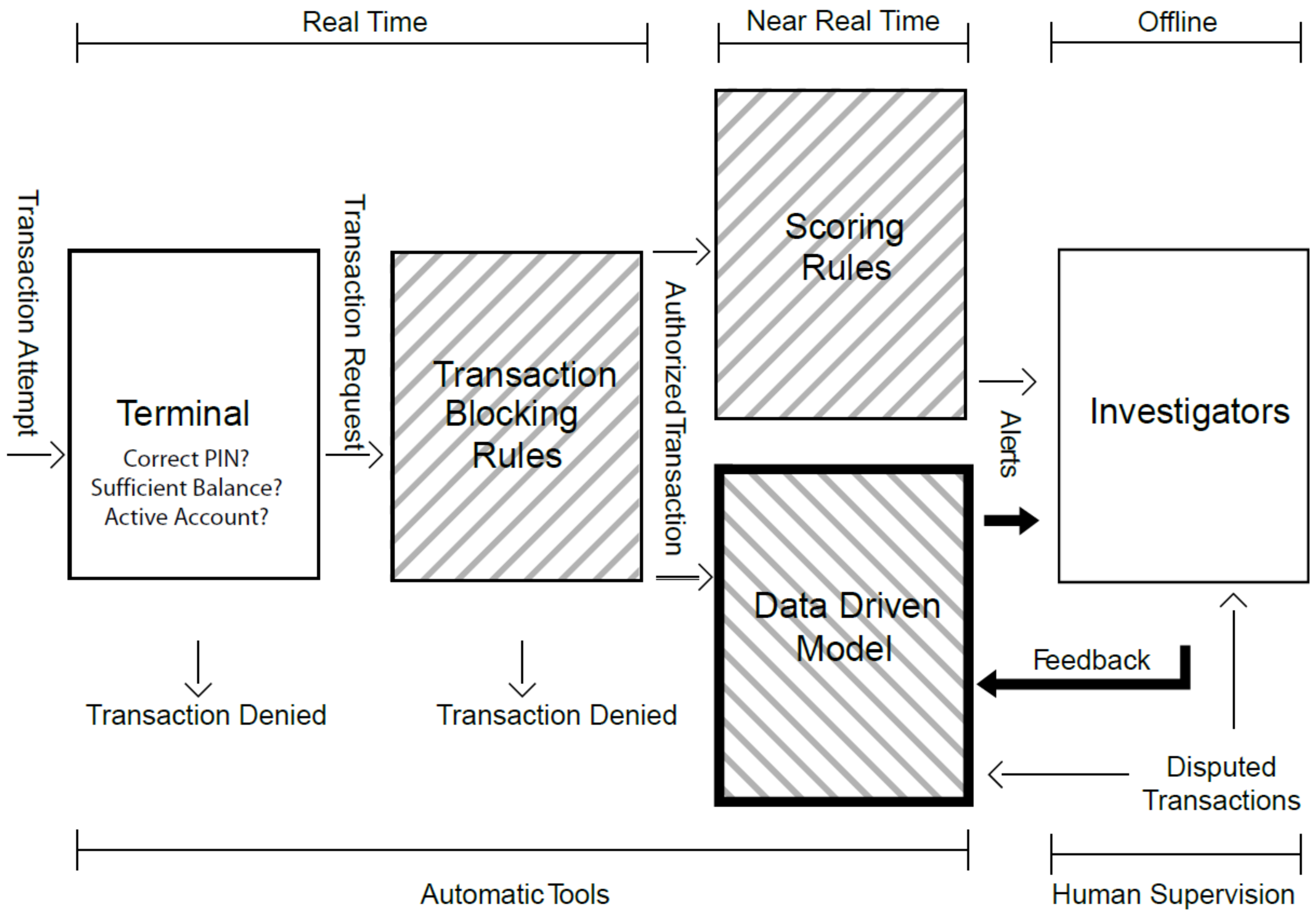
… A DETECTION PROBLEM

Anomaly detection in web sessions in bank e-commerce site



… A DETECTION PROBLEM

Fraud detection in credit card transactions/web sessions



Dal Pozzolo A., Boracchi G., Caelen O., Alippi C. and Bontempi G., *"Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy"* , IEEE TNNL 2017, 14 pages

PRESENTATION OUTLINE

Part1, Problem Formulation and the "Random Variable" world:

- Problem formulation
- Performance measures
- Anomaly detection approaches for random variables (supervised, semi-supervised, unsupervised)

Part2, Anomaly detection in images by learned models:

- Patch-based approaches (semi-supervised, unsupervised)
- Reference-based solutions
- Deep-learning solutions (supervised, semi-supervised, unsupervised)

DISCLAIMERS

I refer to **either changes/anomalies** according to **our personal experience** in the applications we have been addressing.

Anomaly and change detection are different problems, we will briefly summarize the two.

For a **complete overview** on change/anomaly algorithms please refer to surveys below.

T. Ehret, A. Davy, JM Morel, M. Delbracio *"Image Anomalies: A Review and Synthesis of Detection Methods"*, Journal of Mathematical Imaging and Vision, 1-34

V. Chandola, A. Banerjee, V. Kumar. *"Anomaly detection: A survey"*. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages.

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

A. Zimek, E. Schubert, H.P. Kriegel. *"A survey on unsupervised outlier detection in high-dimensional numerical data"* Statistical Analysis and Data Mining: The ASA Data Science Journal, 5(5), 2012.

The Problem Formulation

Anomaly Detection Problem where observations are i.i.d. realizations of a random variable

Forget about images for a while and look for anomalies in **a set of random vectors**

… these techniques will come **very handy also for images**

ANOMALIES

"Anomalies are patterns in data that do not conform to a well defined notion of normal behavior"

Thus:

- **Normal data** are generated from a **stationary process** $\mathcal{P}_N$
- **Anomalies** are from a **different process** $\mathcal{P}_A \neq \mathcal{P}_N$

Examples:

- **Frauds** in the stream of all the credit card transactions
- **Arrhythmias** in ECG tracings
- **Defective regions in an image**, which do not conform a reference pattern

Anomalies might appear as **spurious** elements, and are typically the most **informative** samples in the stream

V. Chandola, A. Banerjee, V. Kumar. *"Anomaly detection: A survey"*. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages.

ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

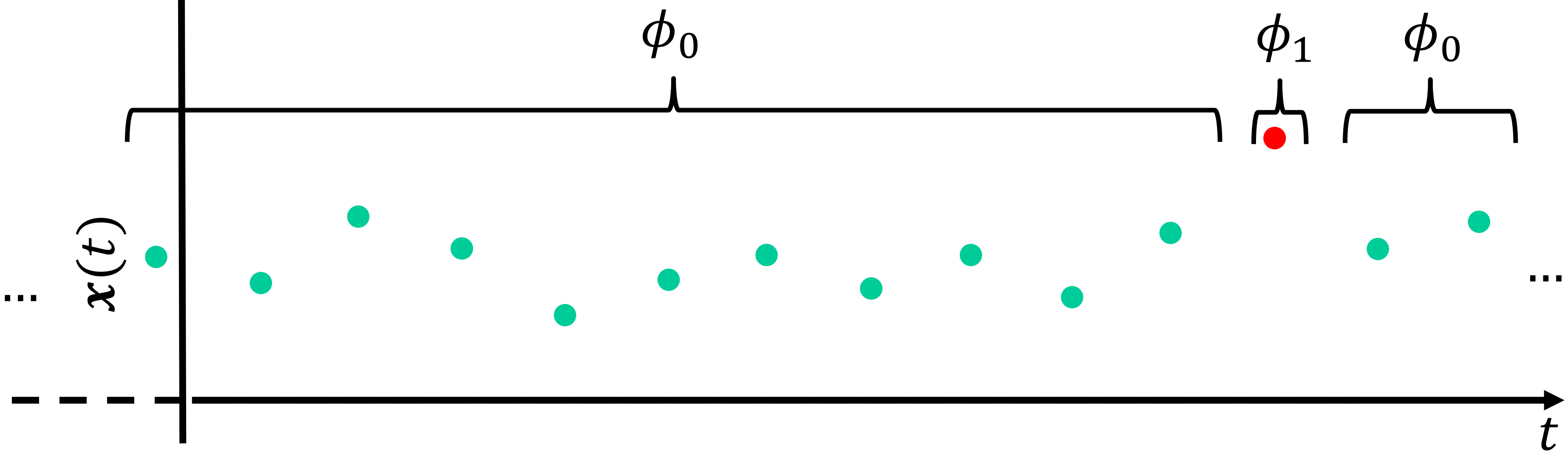
Often, the anomaly-detection problem **boils down to:**

Monitor a set of data (not necessarily a stream)

$$\{\boldsymbol{x}(t),\ t = t_0, \dots\},\ \ \boldsymbol{x}(t) \in \mathbb{R}^d$$

where $x(t)$ are realizations of a **random variable having pdf** ϕ_o, and detect **outliers** i.e., those points that do not conform with ϕ_o

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & \text{normal data} \\ \phi_1 & \text{anomalies} \end{cases},$$



ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

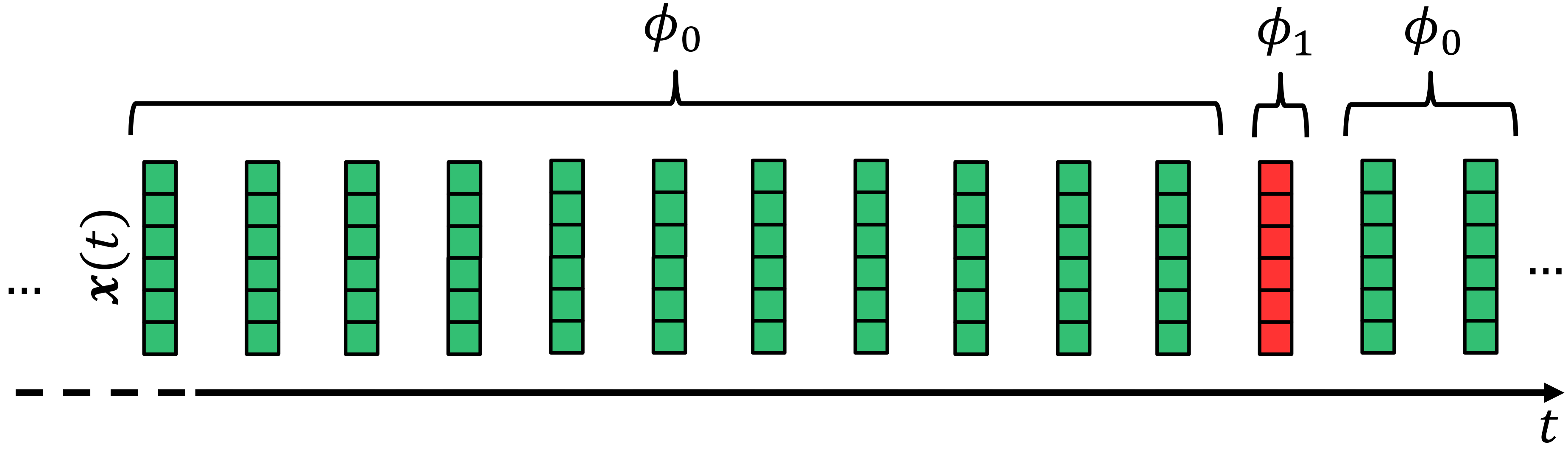
Often, the anomaly-detection problem **boils down to:**

Monitor a set of data (not necessarily a stream)

$$\{\boldsymbol{x}(t),\ t = t_0, \dots\},\ \ \boldsymbol{x}(t) \in \mathbb{R}^d$$

where $x(t)$ are realizations of a **random variable having pdf** ϕ_o, and detect **outliers** i.e., those points that do not conform with ϕ_o

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & \text{normal data} \\ \phi_1 & \text{anomalies} \end{cases},$$



THE LEGAL CASE OF MR HADLUM V. MRS HADLUM (1949)

The sole evidence of adultery consisted of the birth of a child 349 days after Mr Hadlum had left for military service abroad.

Vic Barnett *"The Study of Outliers: Purpose and Model"* Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 27, No. 3 (1978), pp. 242-250

Statistical Approaches

..to detect anomalies

THE ANOMALY / CHANGE DETECTION PROBLEMS

Anomaly-detection problem:

*Locate those **samples that do not conform the normal ones** or a **model explaining normal ones***

Anomalies in data translate to significant information

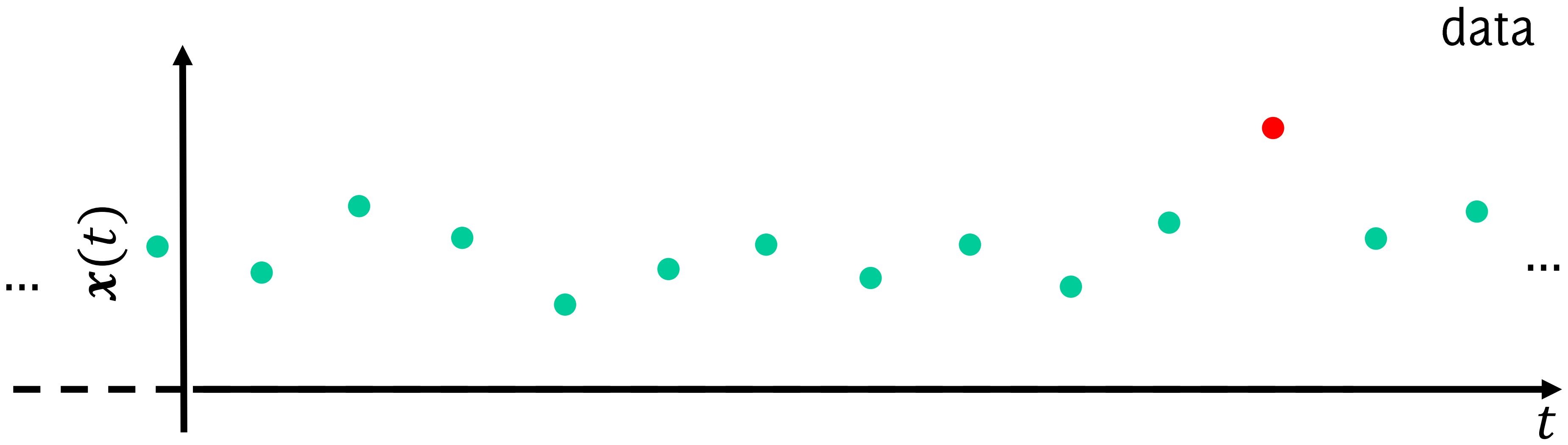
V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

THE TYPICAL SOLUTIONS

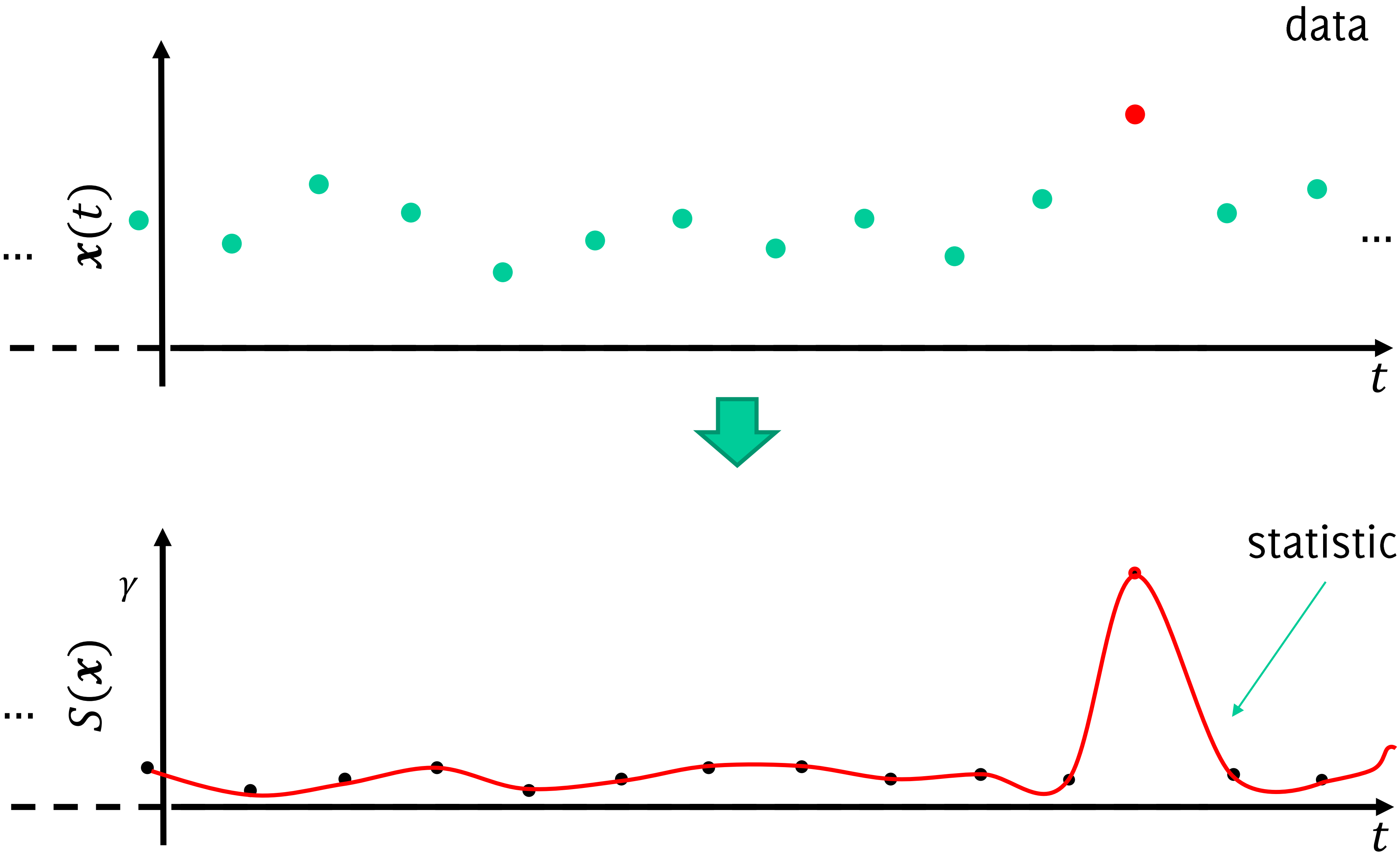
Most algorithms are composed of:

- A **statistic** that has a known response to normal data (e.g., the average, the sample variance, the log-likelihood, the confidence of a classifier, an “anomaly score”…)
- A **decision rule** to analyze the statistic (e.g., an adaptive threshold, a confidence region)

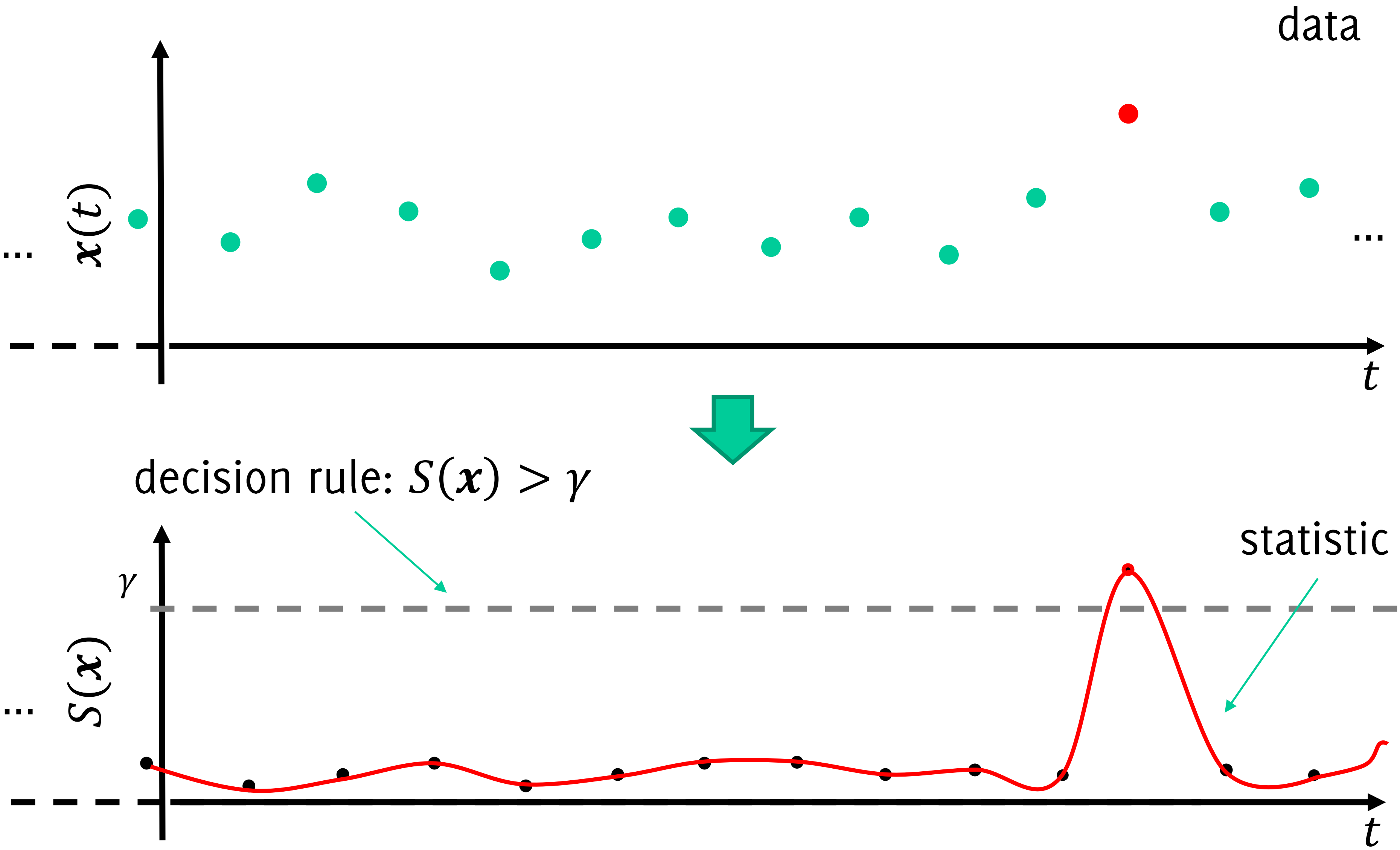
THE TYPICAL SOLUTIONS



THE TYPICAL SOLUTIONS



THE TYPICAL SOLUTIONS



Performance Measures

Assessing performance of anomaly detection algorithms

ANOMALY-DETECTION PERFORMANCE

Anomaly detection performance:

- True positive rate: $TPR = \dfrac{\#\{\text{anomalies detected}\}}{\#\{\text{anomalies}\}}$
- False positive rate: $FPR = \dfrac{\#\{\text{normal samples detected}\}}{\#\{\text{normal samples}\}}$

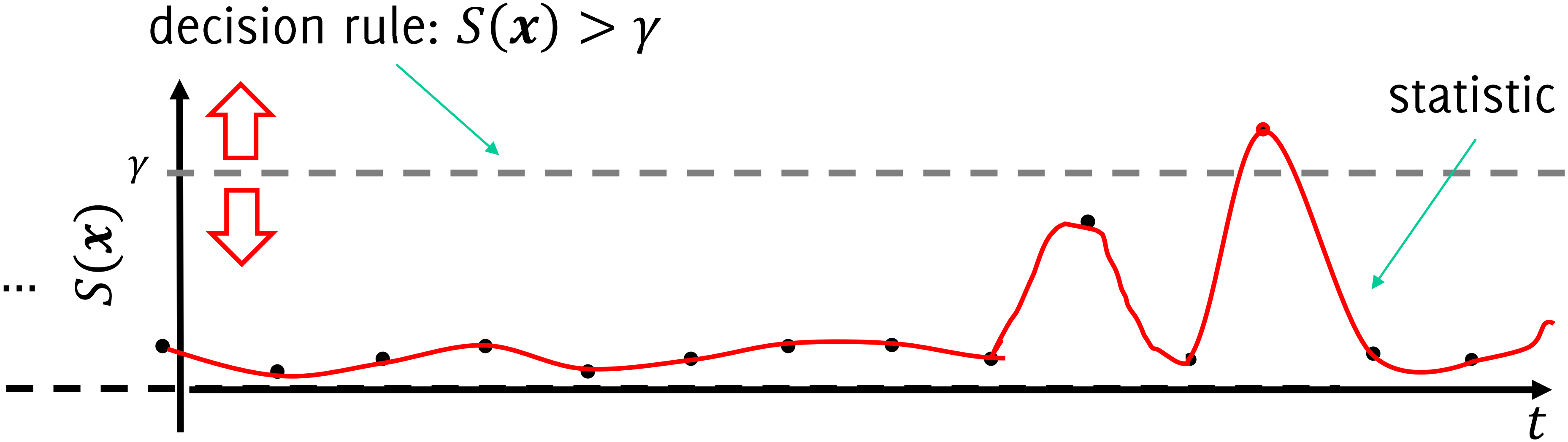
You have probably also heard of

- False negative rate (or miss-rate): $FNR = 1 - TPR$
- True negative rate (or specificity): $TNR = 1 - FPR$
- Precision on anomalies: $\dfrac{\#\{\text{anomalies detected}\}}{\#\{\text{detections}\}}$
- Recall on anomalies (or sensitivity, hit-rate): TPR

THE TYPICAL SOLUTIONS

There is always a **trade-off between $\boldsymbol{TPR}$ and $\boldsymbol{FPR}$** (and similarly for derived quantities), which is ruled by algorithm parameters

By changing γ performance changes (e.g. true positive increases but also false positives do)

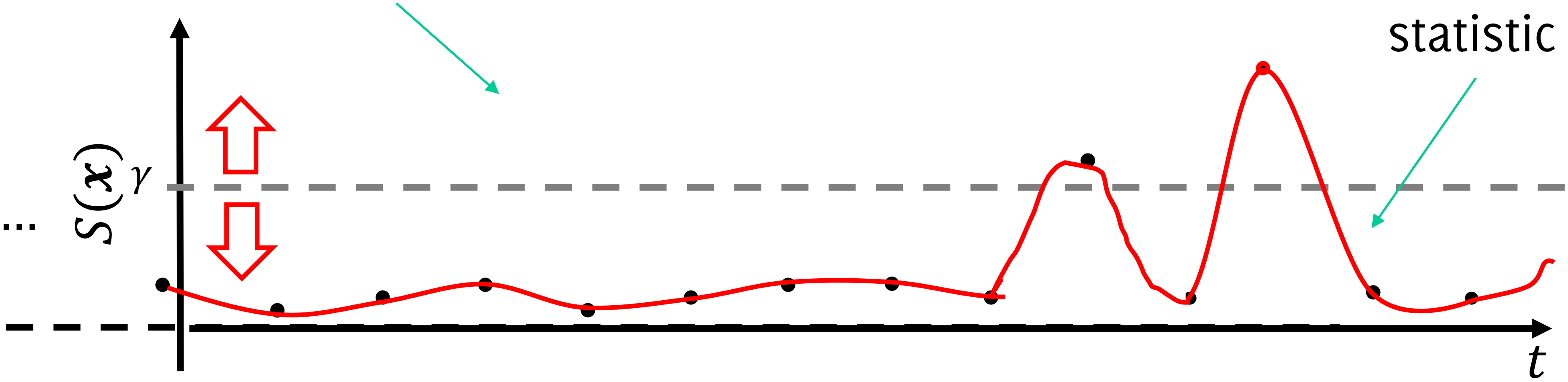


THE TYPICAL SOLUTIONS

There is always a **trade-off between $\boldsymbol{TPR}$ and $\boldsymbol{FPR}$** (and similarly for derived quantities), which is ruled by algorithm parameters

By changing γ performance changes (e.g. true positive increases but also false positives do)

decision rule: $S(\boldsymbol{x}) > \gamma$



statistic

ANOMALY-DETECTION PERFORMANCE

There is always a **trade-off between $\boldsymbol{TPR}$ and $\boldsymbol{FPR}$** (and similarly for derived quantities), which is ruled by algorithm parameters

Thus, to correctly assess performance it is necessary to consider at least **two indicators** (e.g., TPR, FPR)

Indicators combining both TPR and FPR:

$$\text{Accuracy} = \frac{\#\{\text{anomalies detected}\} + \#\{\text{normal samples not detected}\}}{\#\{\text{samples}\}}$$

$$\text{F1 score} = \frac{2\#\{\text{anomalies detected}\}}{\#\{\text{detections}\} + \#\{\text{anomalies}\}}$$

These equal 1 in case of "ideal detector" which detects all the anomalies and has no false positives

ANOMALY-DETECTION PERFORMANCE

Comparing different methods might be tricky since we have to make sure that both have been configured in their best conditions

Testing a large number of parameters lead to the **ROC** (receiver operating characteristic) **curve**

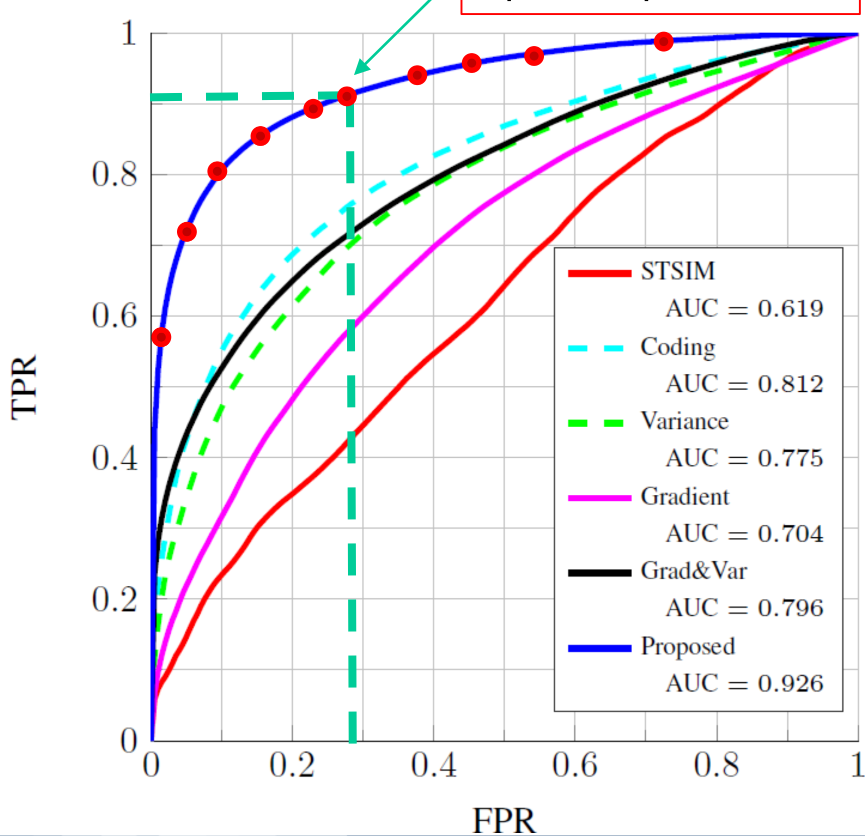
The ideal detector would achieve:

- $FPR = 0\%$,
- $TPR = 100\%$

Thus, the closer to (0,1) the better

The largest the **Area Under the Curve** (AUC), the better

The optimal parameter is the one yielding the point closest to (0,1)



(FPR, TPR) for a specific parameter

Anomaly detection approaches

…when ϕ_0 and ϕ_1 are unknown

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

SUPERVISED ANOMALY DETECTION – DISCLAIMER

Most papers and reviews agree that supervised methods have not to be considered part of anomaly detection, because:

- Anomalies in general lacks of a statistical coherence
- Not (enough) training samples are provided for anomalies

However,

- Some supervised problems are often referred to as «detection», in case of severe class imbalance (e.g. fraud detection)
- Supervised models can be transferred in unsupervised methods, in particular for deep learning

T. Ehret, A. Davy, JM Morel, M. Delbracio *"Image Anomalies: A Review and Synthesis of Detection Methods"*, Journal of Mathematical Imaging and Vision, 1-34

SUPERVISED ANOMALY DETECTION - SOLUTIONS

In **supervised methods** training data are annotated and divided in normal $(+)$ and anomalies $(-)$:

$$TR = \{(\boldsymbol{x}(t), y(t)), \qquad t < t_0, \boldsymbol{x} \in \mathbb{R}^d, y \in \{+,-\}\}$$

Solution:

- Train a two-class classifier to distinguish normal vs anomalous data.

During training:

- Learn a classifier $\mathcal{K}$ from TR.

During testing:

- Compute the classifier output $\mathcal{K}(\boldsymbol{x})$, or
- Set a threshold on the posterior $p_{\mathcal{K}}(-|\boldsymbol{x})$, or
- Select the k $-$most likely anomalies

SUPERVISED ANOMALY DETECTION – CHALLENGES

These **classification problems are challenging** because these anomaly-detection settings typically imply:

- **Class Imbalance:** Normal data far outnumber anomalies
- **Concept Drift:** Anomalies might **evolve** over time, thus the few annotated anomalies might not be representative of anomalies occurring during operations
- **Selection Bias:** Training samples are typically selected through a **closed-loop and biased procedure**. Often only detected anomalies are annotated, and the vast majority of the stream remain unsupervised. This biases the selection of training samples.

Dal Pozzolo A., Boracchi G., Caelen O., Alippi C. and Bontempi G., *"Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy"* , IEEE TNNL 2017, 14 pages

SUPERVISED ANOMALY DETECTION – AN EXAMPLE

This is **what typically happens in fraud detection.**

Class Imbalance:

- Frauds are typically less than 1% of genuine transactions

Concept Drift:

- Fraudster always implement new strategies

Sampling Selection Bias:

- Only alerted / reported transactions are controlled and annotated
- Old transactions that have not been disputed are considered genuine transactions

Dal Pozzolo A., Boracchi G., Caelen O., Alippi C. and Bontempi G., *"Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy"* , IEEE TNNL 2017, 14 pages

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

SEMI-SUPERVISED ANOMALY DETECTION

In semi-supervised methods the TR is composed of normal data

$$TR = \{x(t), t < t_0, x \sim \phi_0\}$$

Very practical assumptions:

- **Normal data** are often **easy to gather**
- **Anomalous data** are **difficult**/costly **to collect**/select and it would be **difficult to gather a representative training set**
- Training examples in TR might not be **representative of all the possible anomalies** that can occur

All in all, it is often **safer** to **detect any data departing from** the **normal conditions**

Semi-supervised anomaly-detection methods are also referred to as **novelty-detection methods**

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

DENSITY-BASED METHODS

Density-Based Methods: ***Normal*** *data occur in* ***high probability regions*** *of a stochastic model, while* ***anomalies*** *occur in the* ***low probability regions*** *of the model*

During training: $\hat{\phi}_0$ can be **estimated** from the training set

$$TR = \{x(t), t < t_0, x \sim \phi_0\}$$

- parametric models (e.g., Gaussian mixture models)
- nonparametric models (e.g. KDE, histograms)

During testing:

- Anomalies are detected as data yielding $\hat{\phi}_0(\boldsymbol{x}) < \eta$

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

DENSITY-BASED METHODS

Advantages:

- $\hat{\phi}_0(\boldsymbol{x})$ indicates how safe a detection is (like a p-value)
- If the density estimation process is robust to outliers, it is possible to tolerate few anomalous samples in TR
- **Histograms are simple to compute** in relatively small dimensions

Challenges:

- **It is challenging to fit models for high-dimensional data**
- Histograms traditionally suffer of **curse of dimensionality** when d increases
- Often the **1D histograms** of the marginals are monitored, **ignoring the correlations** among components

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

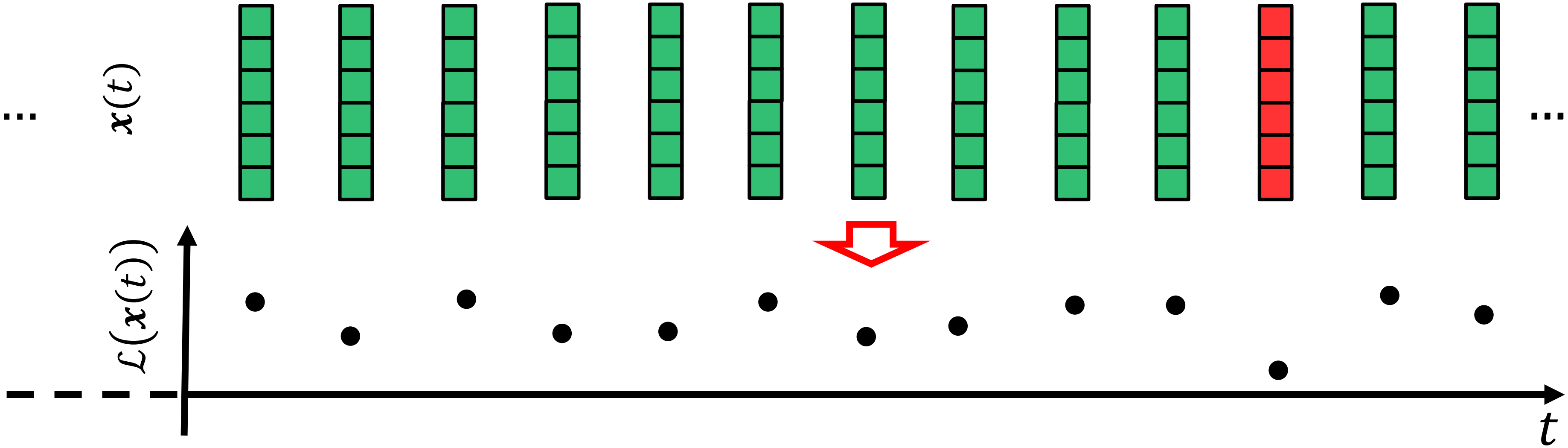
DENSITY-BASED METHODS: MONITORING THE LOG-LIKELIHOOD

Monitoring the log-likelihood of data w.r.t $\hat{\phi}_0$ allow to address anomaly-detection problem in multivariate data

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute

$$\mathcal{L}\big(\boldsymbol{x}(t)\big) = \log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

3. Monitor $\{\mathcal{L}\big(\boldsymbol{x}(t)\big),\ t = 1, \dots\}$



DENSITY-BASED METHODS: MONITORING THE LOG-LIKELIHOOD

Monitoring the log-likelihood of data w.r.t $\hat{\phi}_0$ allow to address anomaly-detection problem in multivariate data

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute
$$\mathcal{L}\big(\boldsymbol{x}(t)\big) = \log(\hat{\phi}_0(\boldsymbol{x}(t)))$$
3. Monitor $\{\mathcal{L}\big(\boldsymbol{x}(t)\big),\ t = 1, \dots\}$

This is quite a popular approach in either anomaly and change detection algorithms

L. I. Kuncheva, *“Change detection in streaming multivariate data using likelihood detectors,"* IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 5, 2013.

X. Song, M. Wu, C. Jermaine, and S. Ranka, *“Statistical change detection for multidimensional data,"* in Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD), 2007.

J. H. Sullivan and W. H. Woodall, *“Change-point detection of mean vector or covariance matrix shifts using multivariate individual observations,"* IIE transactions, vol. 32, no. 6, 2000.

C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016, New York, USA, July 9 - 13

DOMAIN-BASED METHODS

Domain-based methods: ***Estimate a boundary around normal*** *data, rather than the density of normal data.*

A **drawback of density-estimation methods** is that they are meant to be accurate in high-density regions, while anomalies live in low-density ones.

One-Class SVM are domain-based methods defined by **the normal samples at the periphery of the distribution.**

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C. "*Support Vector Method for Novelty Detection*". In NIPS 1999 (Vol. 12, pp. 582-588).

Tax, D. M., Duin, R. P. "*Support vector domain description*". Pattern recognition letters, 20(11), 1191-1199 (1999)

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

ONE-CLASS SVM (SCHÖLKOPF ET AL. 1999)

Idea: define boundaries by estimating a **binary function** f that **captures regions of the input space where density is higher**.

As in support vector methods, f is **defined in the feature space** F and **decision boundaries are defined by a few support vectors** (i.e., a few normal data).

Let $\psi(\boldsymbol{x})$ the feature associated to $\boldsymbol{x}$, f is defined as

$$f(\boldsymbol{x}) = \text{sign}(< w, \psi(\boldsymbol{x}) > -\rho)$$

Where the hyperplane parameters w, ρ are optimized to yield a **function that is positive on most training samples.** Thus in the feature space, normal points can be separated from the origin.

A linear separation in the feature space corresponds to a **variety of nonlinear boundaries in the space of** $\boldsymbol{x}$.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C. "*Support Vector Method for Novelty Detection*". In NIPS 1999 (Vol. 12, pp. 582-588).

ONE-CLASS SVM (TAX AND DUIN 1999)

Boundaries of normal region can be also defined by an **hypersphere that, in the feature space, encloses most of the normal data.**

Similar detection formulas hold, measuring the distance in the feature space between the sphere center and $\psi(\boldsymbol{x})$ for $\boldsymbol{x} \in TR$.

The function is always defined by a few support vectors.

Remarks: In both one-class approaches, the amount of samples that falls within the margin (outliers) is controlled by regularization parameters.

This parameter regulates the number of outliers in the training set and the detector sensitivity.

Tax, D. M., Duin, R. P. "*Support vector domain description*". Pattern recognition letters, 20(11), 1191-1199 (1999)

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

UNSUPERVISED ANOMALY-DETECTION

The training set TR might contain **both normal and anomalous data**. However, **no labels** are provided

$$TR = \{x(t), t < t_0\}$$

Underlying assumption: ***Anomalies are rare*** *w.r.t. normal data* TR

One in principle could use:

- Density/Domain based methods that are robust to outliers can be applied in an unsupervised scenario
- Unsupervised methods can be improved whenever labels are available

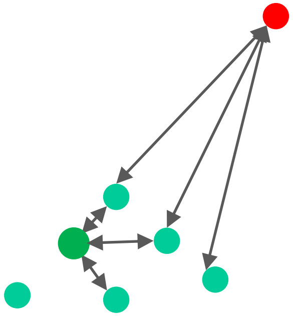
DISTANCE-BASED METHODS

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors.***

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its $\boldsymbol{k}$ **−nearest neighbor**



V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

Zhao, M., Saligrama, V. *"Anomaly detection with score functions based on nearest neighbor graphs"*. NIPS 2009

A. Zimek, E. Schubert, H. Kriegel. *"A survey on unsupervised outlier detection in high-dimensional numerical data"* SADM 2012.

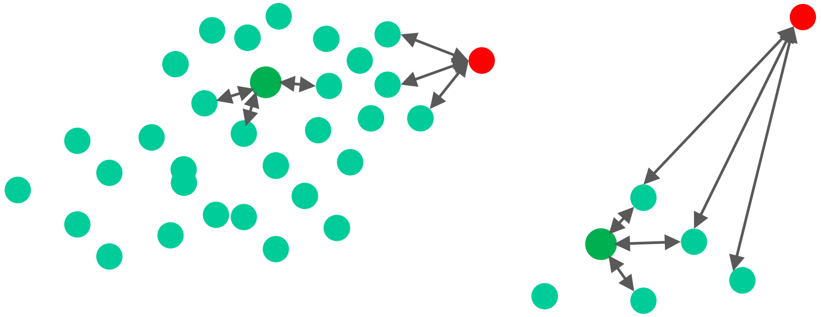
DISTANCE-BASED METHODS

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its $\boldsymbol{k}$ **−nearest neighbor**
- the **above distance** considered **relatively to neighbors**



V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

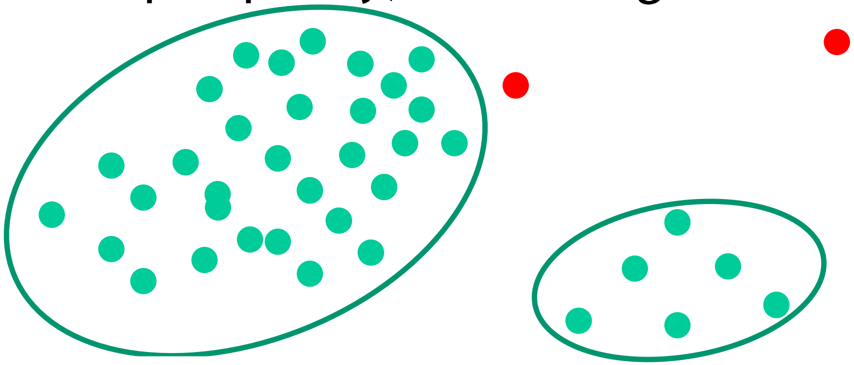
DISTANCE-BASED METHODS

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its $\boldsymbol{k}$ **−nearest neighbor**
- the **above distance** considered **relatively to neighbors**
- whether they do not belong to **clusters**, or are at the cluster periphery, or belong to small and sparse clusters

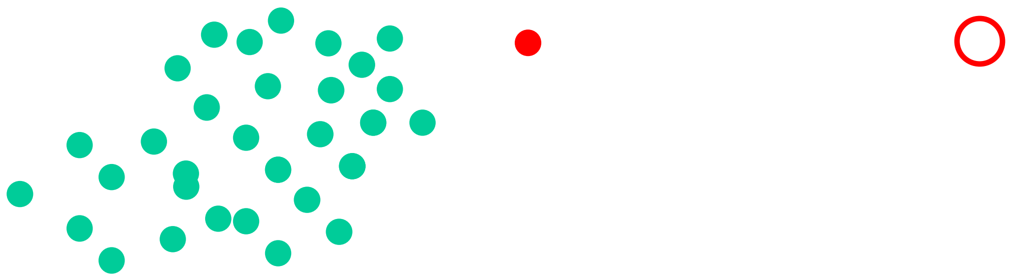


V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure

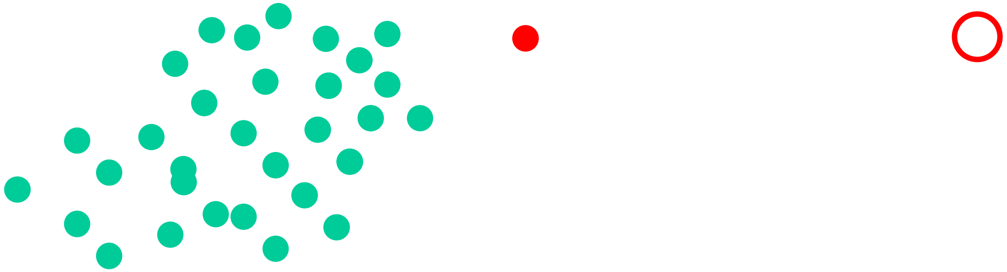


Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure



Randomly choose

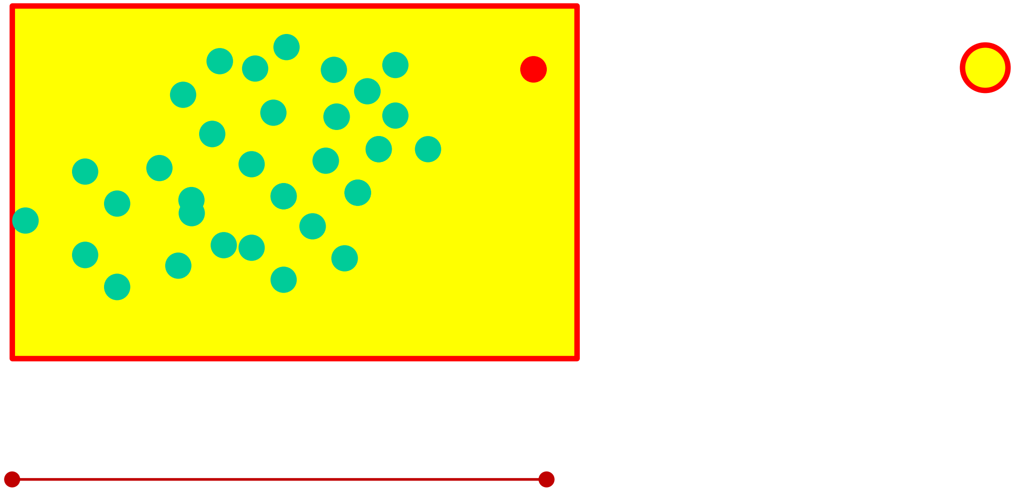
1. a component x_i

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure



Randomly choose

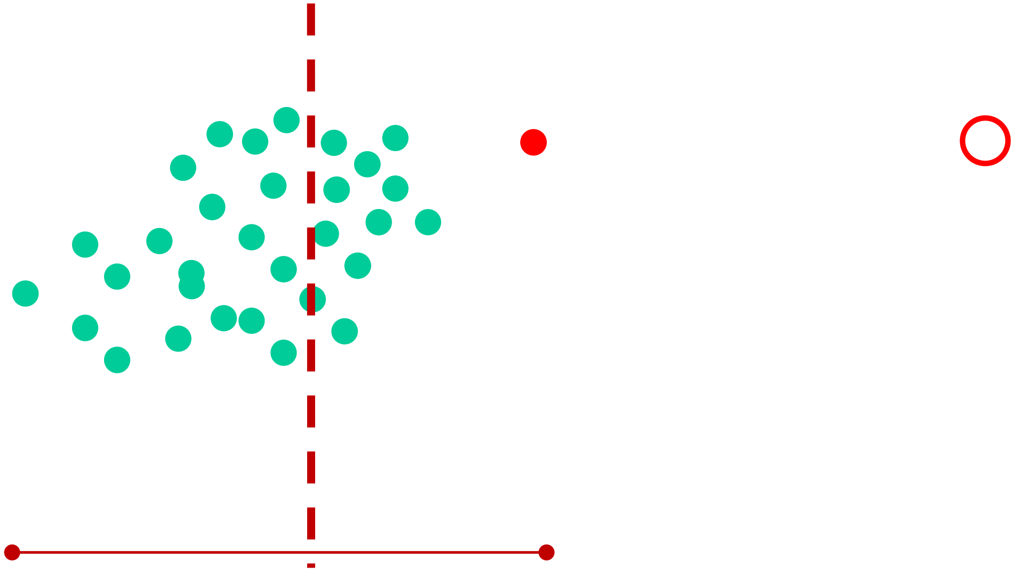
1. a component x_i

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure



Randomly choose

1. a component x_i
2. a value in the range of projections of TR over the i-th component

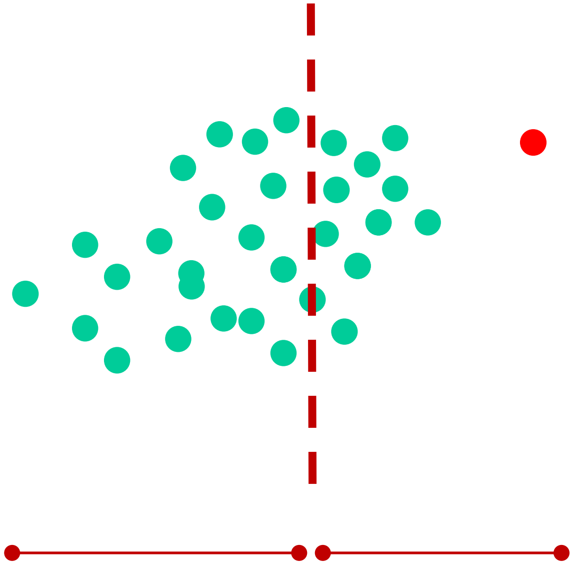
This yields a splitting

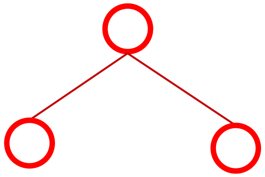
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





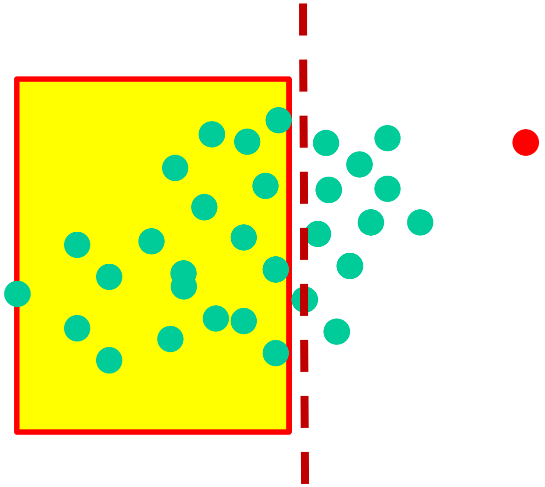
This yields a splitting criteria

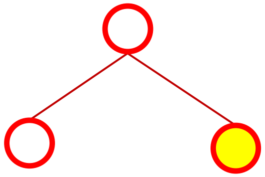
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





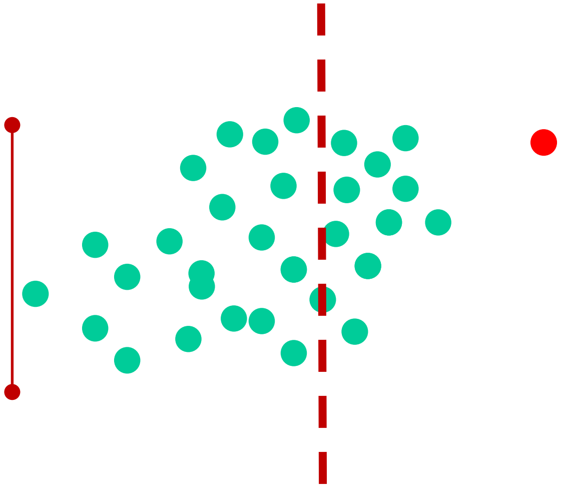
Repeat the procedure on each node:

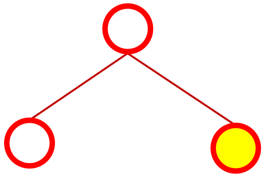
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





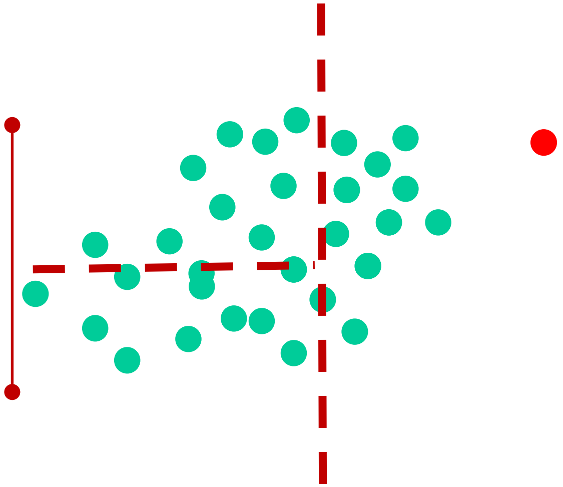
Repeat the procedure on each node: Randomly select a component

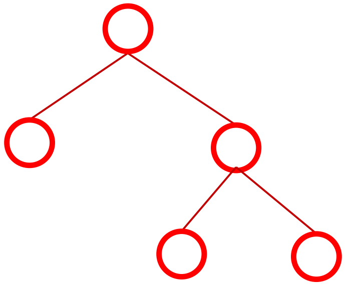
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





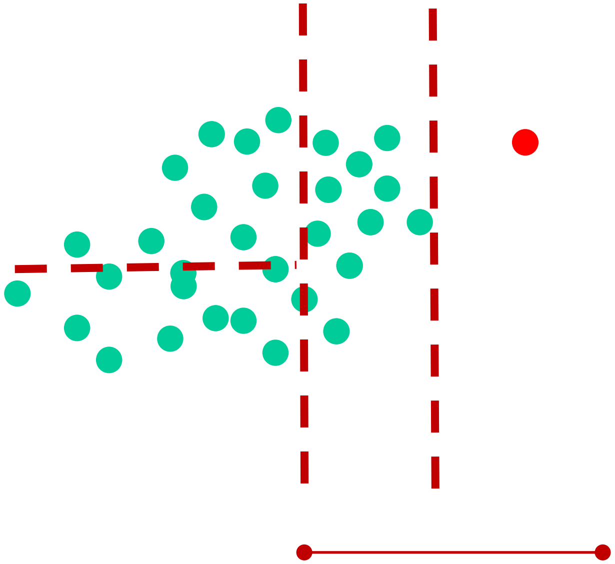
Repeat the procedure on each node: Randomly select a component and a cut point

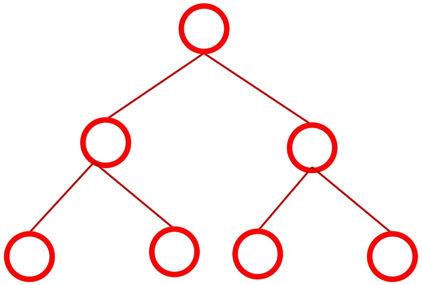
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed through an iterative scheme





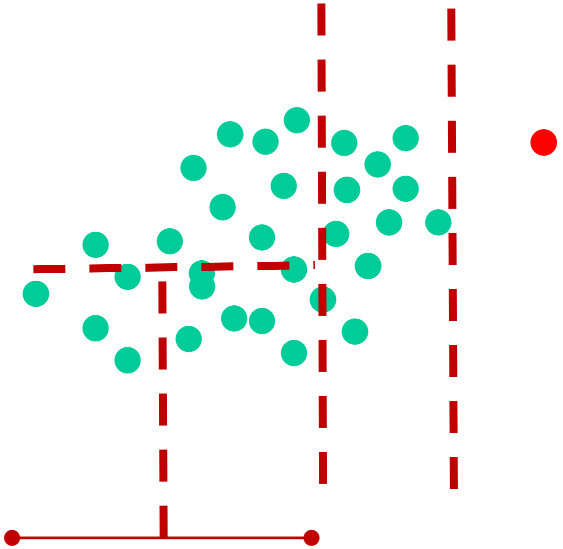
Randomly choose a component and a value within the range and define a splitting criteria

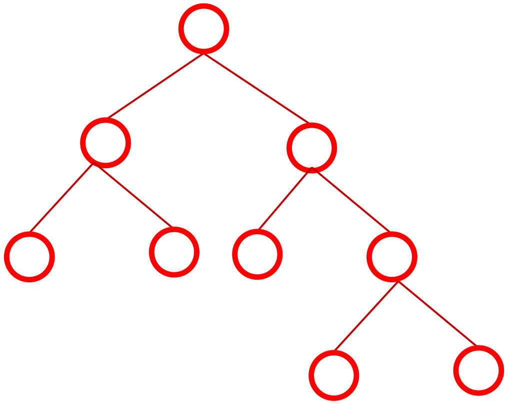
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





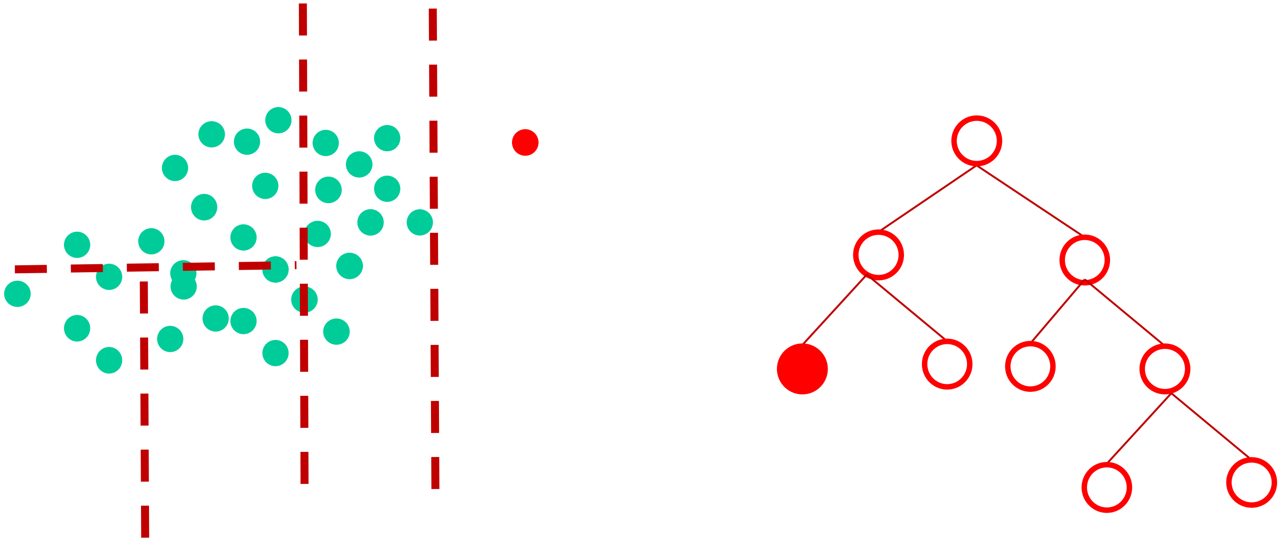
Repeat the procedure on the nodes:
Randomly select a component and a cut point

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure



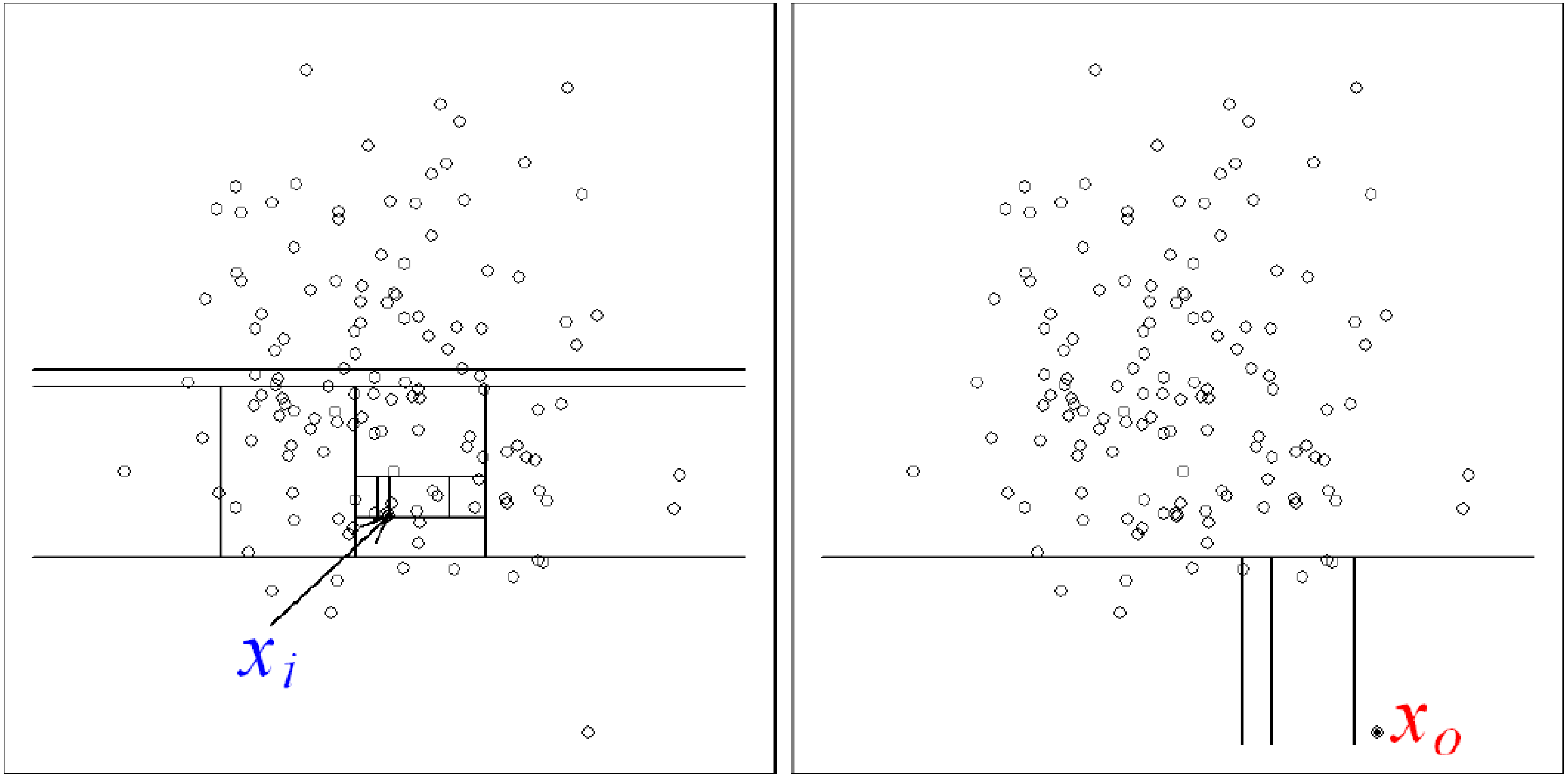
Anomalies lies in leaves close to the root.

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

An anomalous point (x_0) can be easily isolated

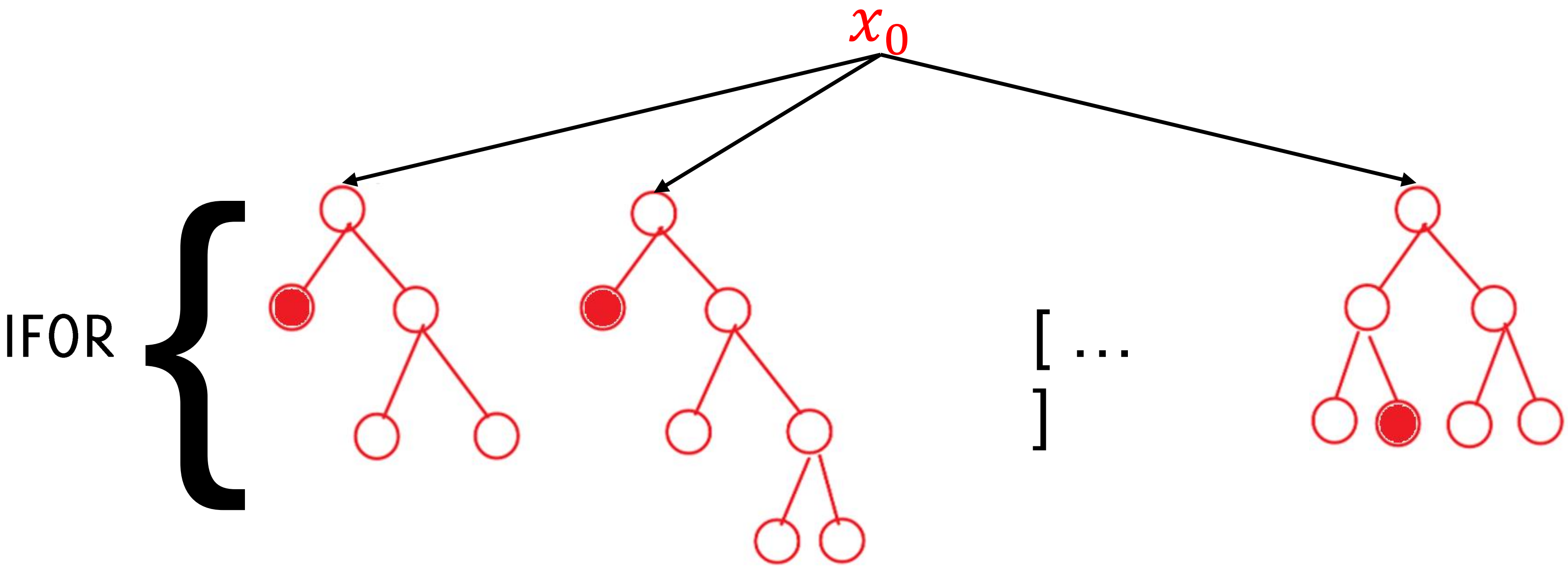
Genuine points (x_i) are instead difficult to isolate.



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

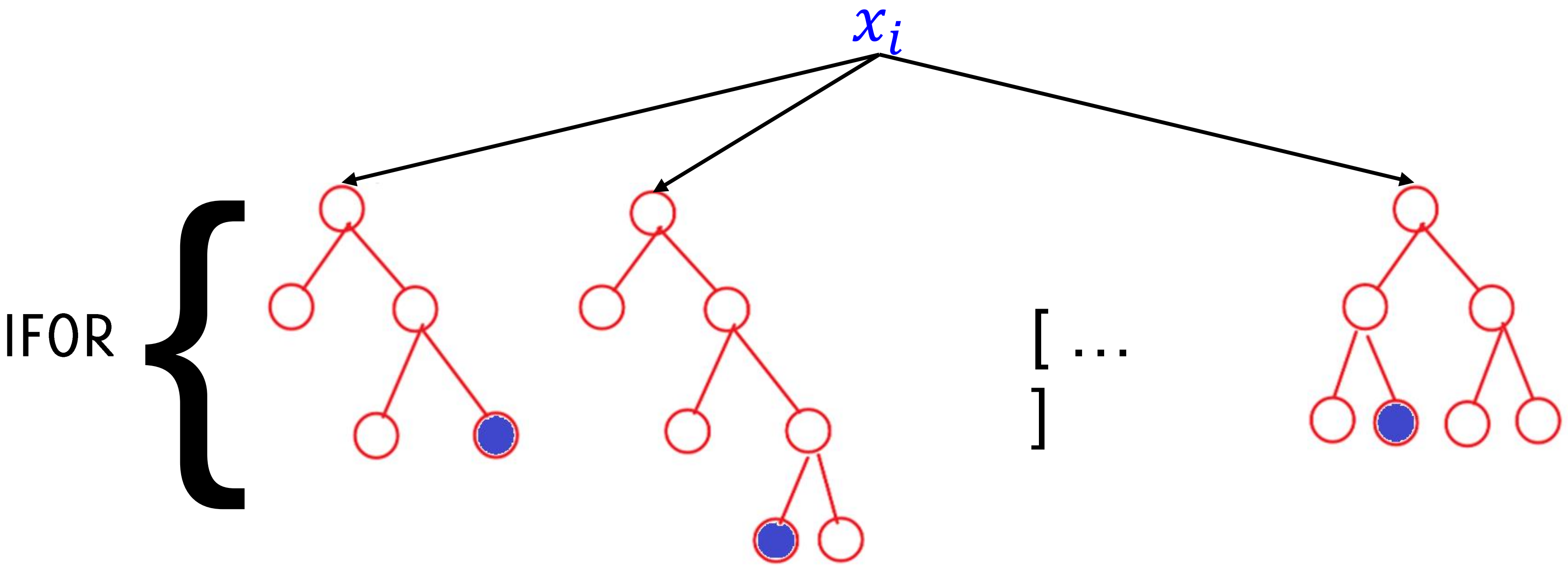
Anomalies



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

Normal data



x_i

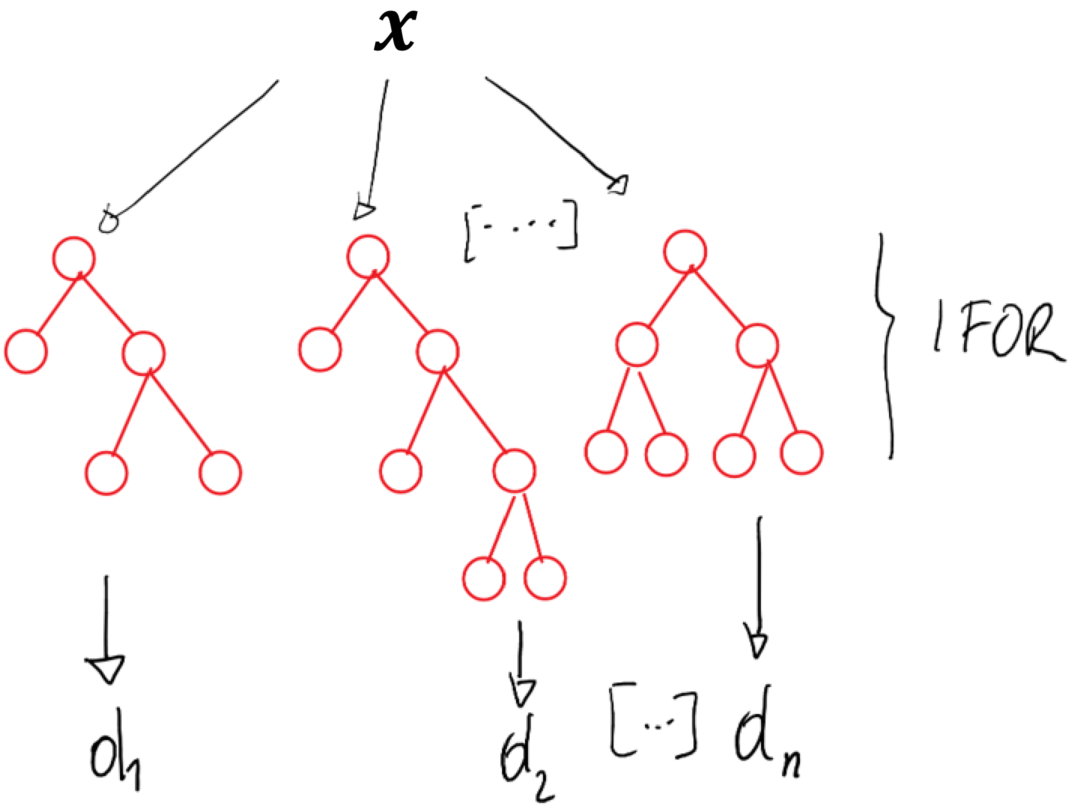
IFOR

[…
]

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST: TESTING

Compute $E(h(\boldsymbol{x}))$, the **average path length** among all the trees in the forest, of a test sample $\boldsymbol{x}$



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST: TESTING

A test sample is identified as **anomalous** when:

$$\mathcal{A}(\boldsymbol{x}) = 2^{-\frac{E\left(h(\boldsymbol{x})\right)}{c(n)}} > \gamma$$

- n : number of sessions in TR
- $c(n)$: average path length of unsuccessful search in Binary

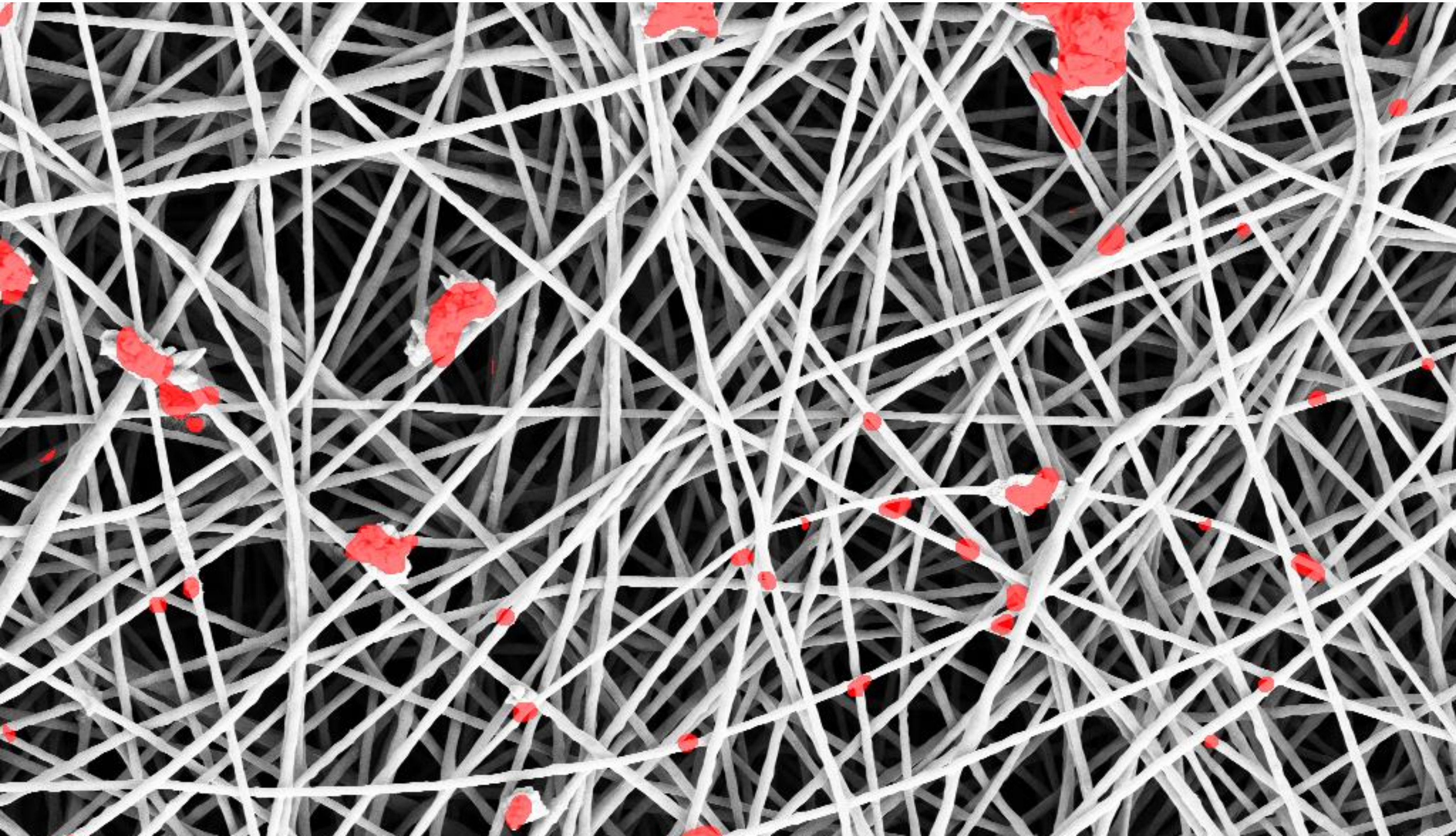
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

Let’s go back to images now…

Applying statistical methods to image patches

OUR RUNNING EXAMPLE

Goal: Automatically measure area covered by defects

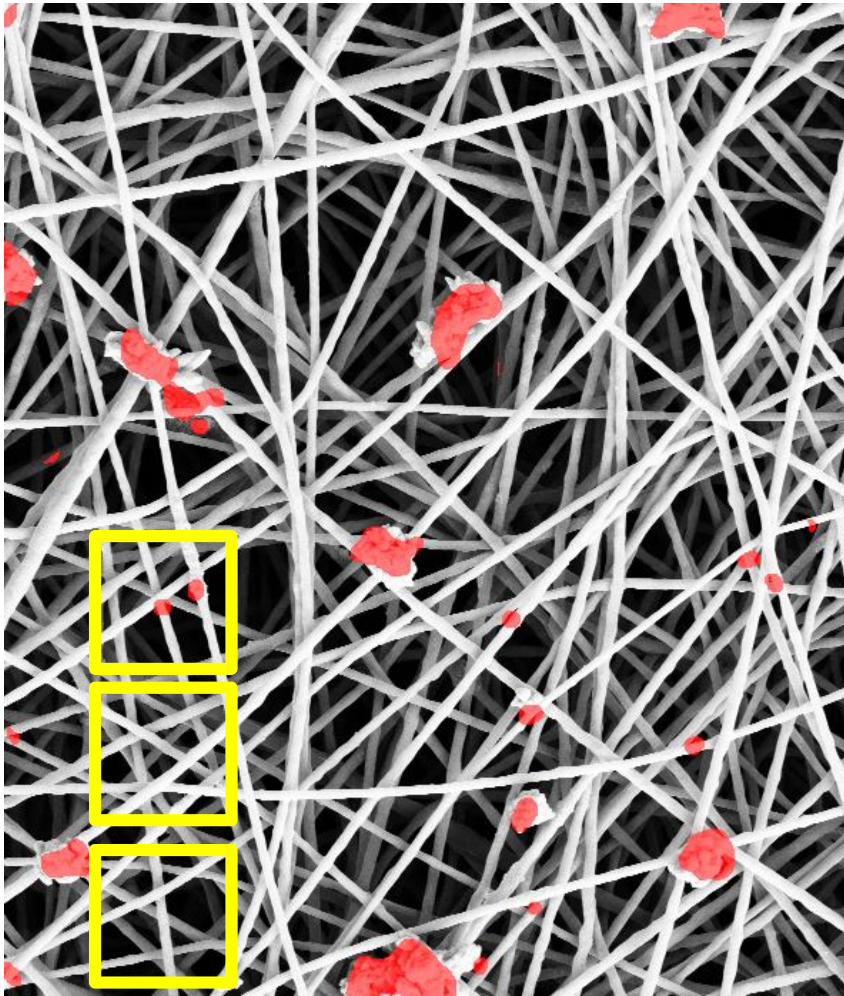


ANOMALY DETECTION IN IMAGES

The goal not determining whether the whole image is normal or anomalous, but **locate/segment possible anomalies**

Therefore, it is convenient to

1. **Analyze the image patch-wise**
2. Isolate regions containing patches that are detected as as anomalies



Can we pursue approaches meant for random variables on image patches?

DENSITY-BASED APPROACH ON IMAGE PATCHES

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ describing normal patches.

Testing

i. Split the test image in patches

ii. Compute $\hat{\phi}_0(\boldsymbol{s})$ the likelihood of each test patch $\boldsymbol{s}$

iii. Detect anomalies by thresholding the likelihood

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing

DENSITY-BASED APPROACH ON IMAGE PATCHES

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\boxed{\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)}$ describing normal patches.

This model is rarely accurate on natural images.
Small patches (e.g. 2 × 2 or 5 × 5) are typically preferred

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing

X Xie, M Mirmehdi *"Texture exemplars for defect detection on random textures"* – ICPR 2005

DENSITY-BASED APPROACH ON IMAGE PATCHES

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\boxed{\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)}$ describing normal patches.

In some cases (textures) a Gaussian Mixture was used as a more general model

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing

X Xie, M Mirmehdi *"Texture exemplars for defect detection on random textures"* – ICPR 2005

DENSITY-BASED APPROACH ON IMAGE PATCHES

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

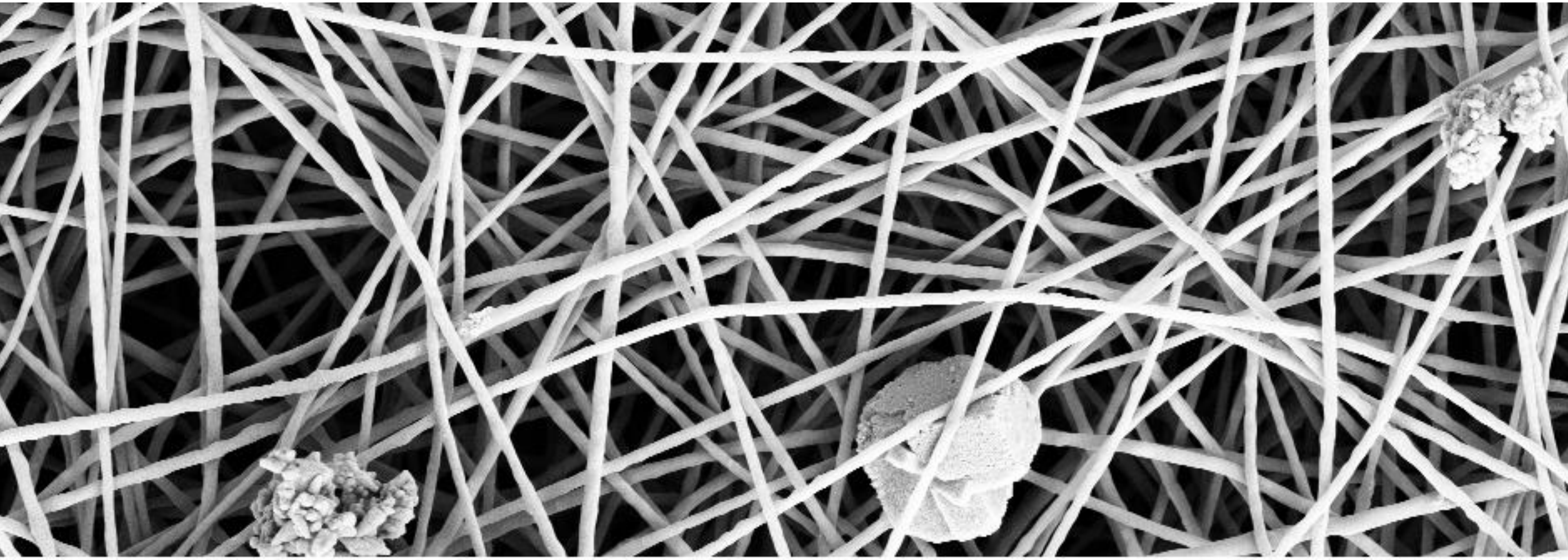
ii. Fit a statistical model $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ describing normal patches.

Random selection procedures can be employed to minimize the risk of including outliers

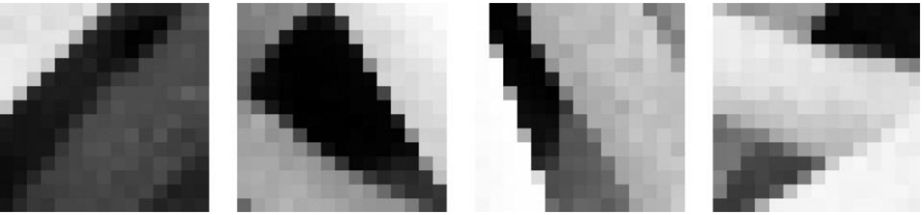
Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

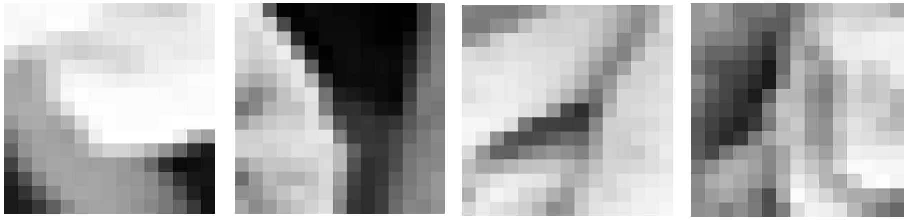
In many anomaly-detection problems in imaging, **normal regions exhibit peculiar structures and spatial correlation**



Normal patches

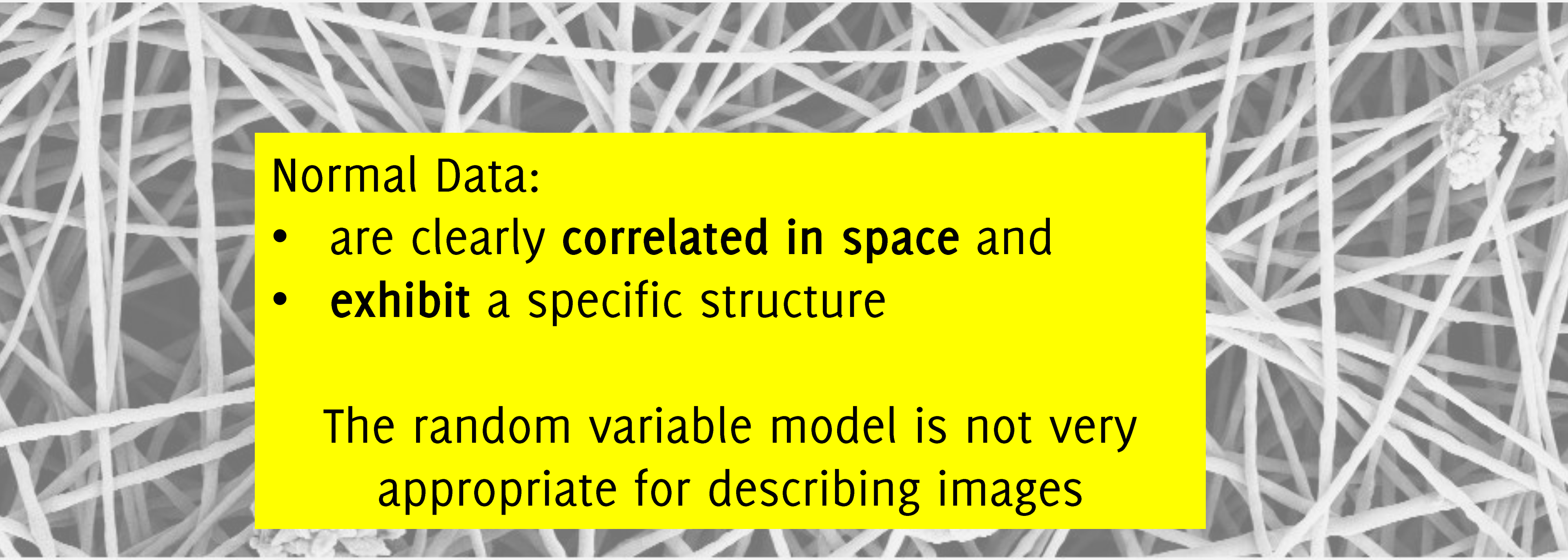


Anomalous patches



THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

In many anomaly-detection problems in imaging, **normal regions exhibit peculiar structures and spatial correlation**

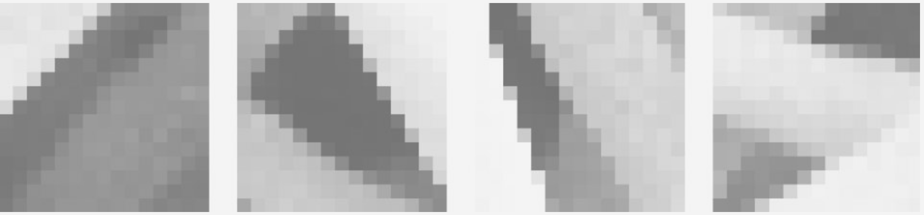


Normal Data:

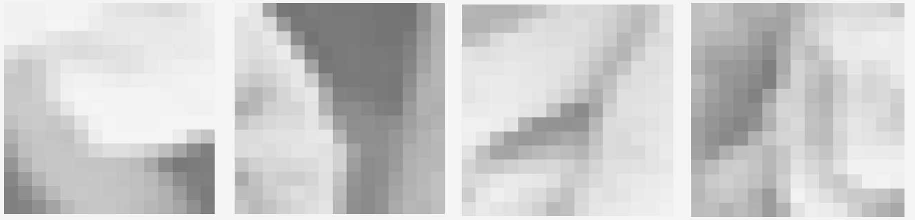
- are clearly **correlated in space** and
- **exhibit** a specific structure

The random variable model is not very appropriate for describing images

Normal regions

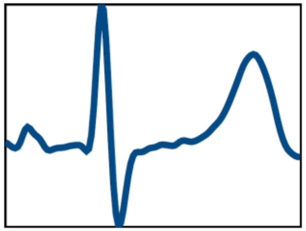


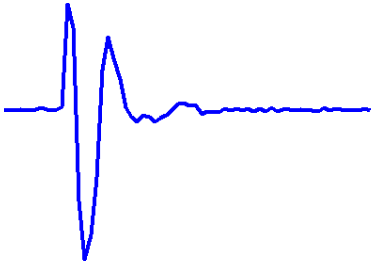
Anomalous regions

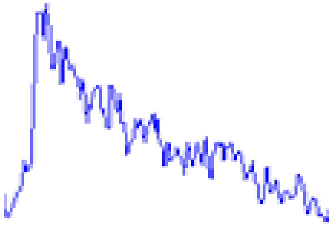


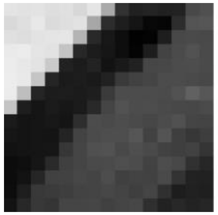
REAL WORLD DETECTION PROBLEMS

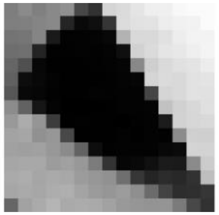
Random variable model **does not successfully apply to signals or images** (not even small portions)





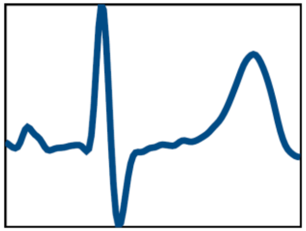


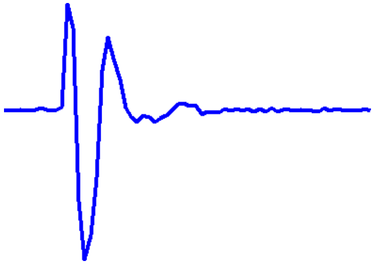


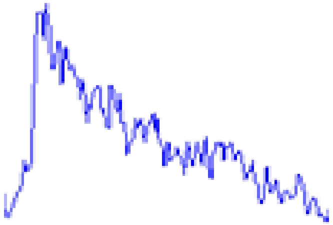


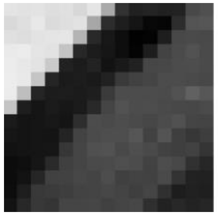
REAL WORLD DETECTION PROBLEMS

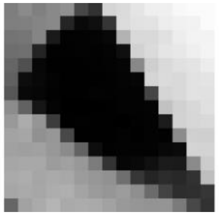
Random variable model **does not successfully apply to signals or images** (not even small portions)









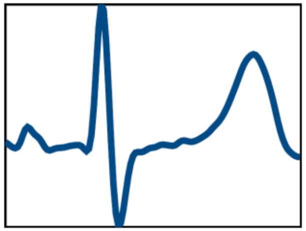
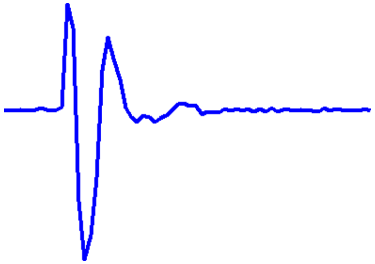
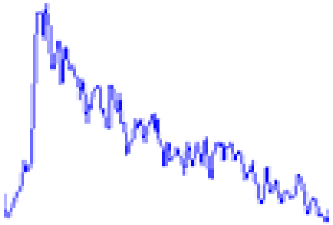
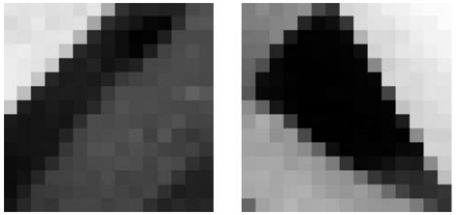


Stacking each signal $\boldsymbol{s} \in \mathbb{R}^d$ in a vector $\boldsymbol{x}$ is not convenient:

- **Data dimension** d can become **huge**
- **Correlation among components is difficult to model**

REAL WORLD DETECTION PROBLEMS

Random variable model **does not successfully apply to signals or images** (not even small portions)

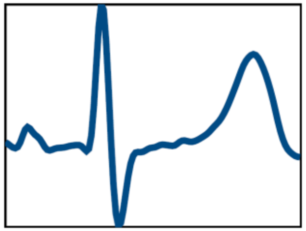
Stacking each signal $\boldsymbol{s} \in \mathbb{R}^d$ in a vector $\boldsymbol{x}$ is not convenient:

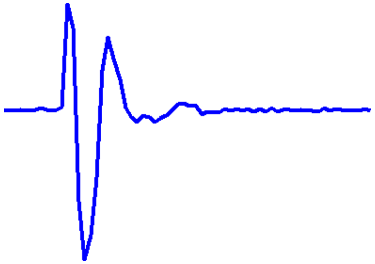
- **Data dimension** d can become **huge**
- **Correlation among components is difficult to model**

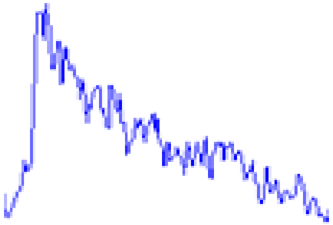
It is not easy to **estimate a density model** or threat these as **realizations of a random variable**

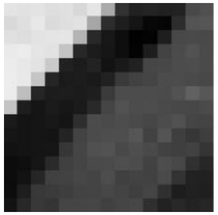
REAL WORLD DETECTION PROBLEMS

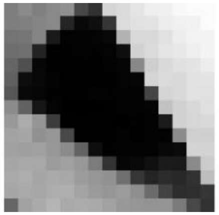
Random variable model **does not successfully apply to signals or images** (not even small portions)











Stacking each signal $\boldsymbol{s} \in \mathbb{R}^d$ in a vector $\boldsymbol{x}$ is not convenient:

- **Data dimension** d can become **huge**
- **Correlation among components is difficult to model**

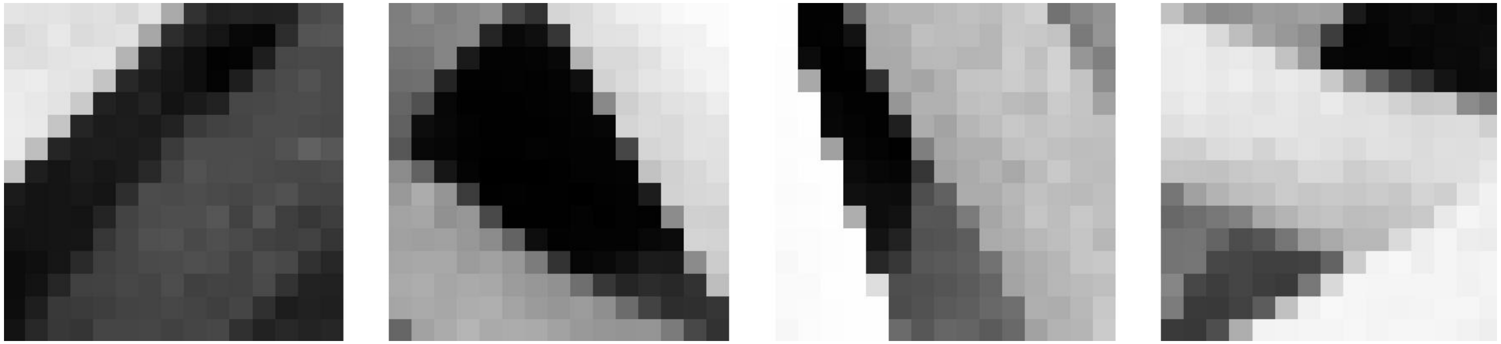
It is not easy to **estimate a density model** or threat these as **realizations of a random variable**

Moreover, when **normal data** exhibit a peculiar **structure**, we are interested in **detecting changes/anomalies affecting that structure**

REAL WORLD DETECTION PROBLEMS

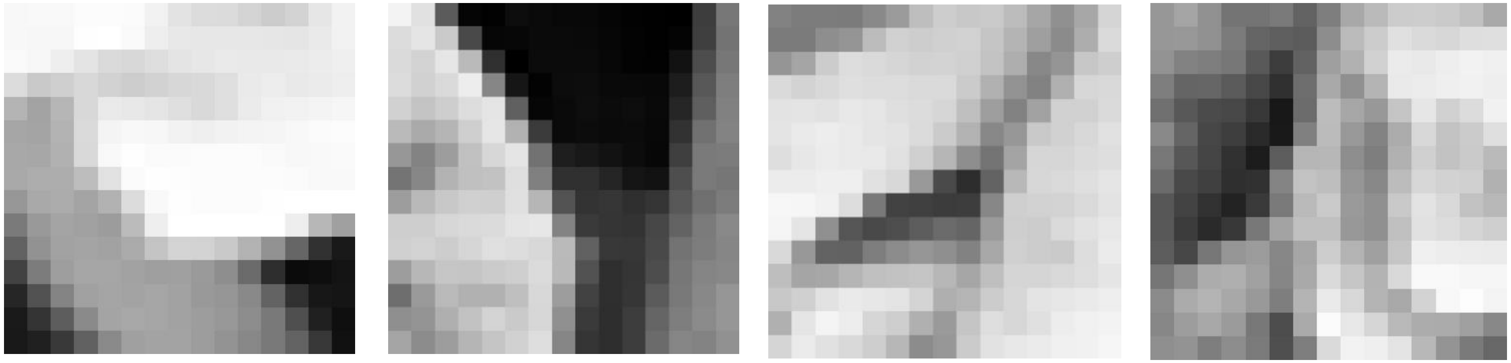
Normal patches -> background

- Exhibit a specific structure (geometry) or intensities



Anomalous patches:

- Are rare elements that do not confrom with the background



Patch-based approaches

Out of the “Random Variable” World:
signal-based models for images

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Provide, for each test sample, an **anomaly score,** or measure of rareness, w.r.t. the learned model
3. Apply a **decision rule** to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighborhoods

Remark: Statistical-based approaches seen before uses as background model the statistical distribution $\widehat{\phi}_0$ and a statistic as anomaly score

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Provide, for each test sample, an **anomaly score,** or measure of rareness, w.r.t. the learned model
3. Apply a **decision rule** to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth at are not consist

Remark: St uses as background nd a statistic as anomaly sc

The background model is used to **bring an image patch into the "random variable world"** (regression, encoding, feature extraction…)

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Provide, for each test sample, an **anomaly score,** or measure of rareness, w.r.t. the learned model
3. Apply a **decision rule** to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smo … e not cons …

Remark: … as backgro … statistic as anomaly …

Once "applied" the background model, one can use **most of anomaly detection methods for the "random variable world".**
This might require **fitting an additional model**

THE TYPICAL APPROACH

Different options to learn the background model

- **semi-supervised approach,** background model is learned exclusively normal data
- **unsupervised approach**, background model is fit to both normal and anomalous but it is robust to outliers

SEMI-SUPERVISED ANOMALY-DETECTION IN IMAGES

Out of the "Random Variable" world

- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

SEMI-SUPERVISED ANOMALY-DETECTION IN IMAGES

Out of the "Random Variable" world

- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

RECONSTRUCTION-BASED METHODS

Fit a statistical model to the observation to ***describe dependence****,* ***apply anomaly detection*** *on the independent* ***residuals****.*

Detection is performed by using a model $\mathcal{M}$ which **represents normal data**:

- **During training:** learn the model $\mathcal{M}$ from training set TR
- **During testing:**
  - Reconstruct each test signal $\boldsymbol{s}$ through $\mathcal{M}$.
  - Assess the **residuals** between $\boldsymbol{s}$ and its reconstruction

The rationale is that $\mathcal{M}$ can reconstruct only normal data, thus anomalies are expected to yield large reconstruction errors.

Bengio, Y., Courville, A., Vincent, P. *"Representation learning: A review and new perspectives"*. IEEE TPAMI 2013

RECONSTRUCTION-BASED METHODS

Popular models are:

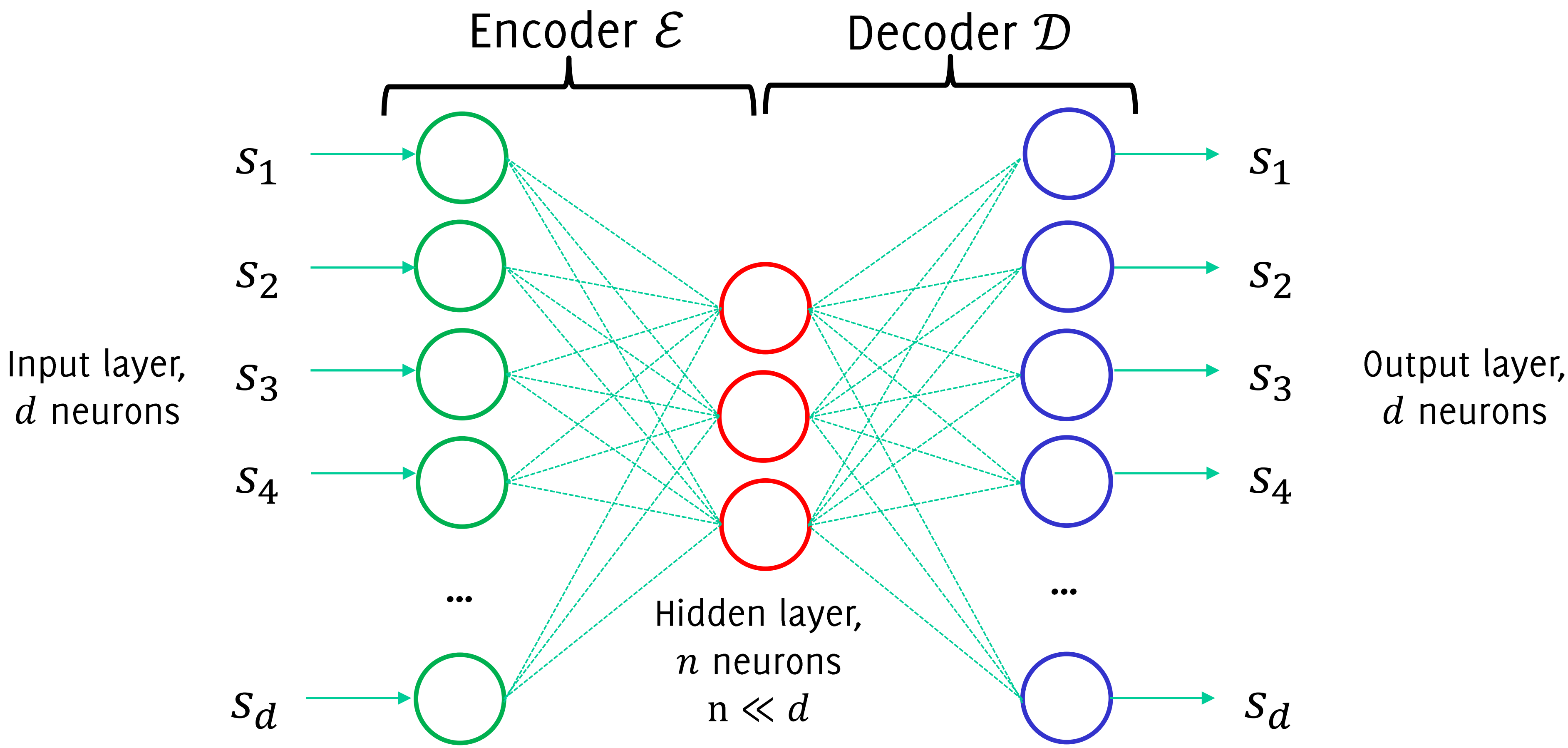
- autoregressive models for time series (ARMA, ARIMA…)
- neural networks, in particular auto-encoders, for higher dimensional data
- projection on subspaces / manifolds
- dictionaries yielding sparse-representations

The two latter can be also interpreted as subspace methods

RECONSTRUCTION-BASED METHODS

Autoencoders are neural networks used for data reconstruction (learn the identity function)

The typical structure of an autoencoder is:



RECONSTRUCTION-BASED METHODS

Autoencoders are non-parametric models that can be trained to reconstruct all the data in a training set. The reconstruction loss is

$$\sum_{\boldsymbol{s}\in S} \left\| \boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big) \right\|_2$$

and training of $\mathcal{D}\big(\mathcal{E}(\cdot)\big)$ is performed through standard backpropagation algorithms (e.g. SGD)

Remark

- AE typically does not provide exact reconstruction since $n \ll d$.
- Additional regularization terms might be included in the loss function

Bengio, Y., Courville, A., Vincent, P. "*Representation learning: A review and new perspectives*". IEEE TPAMI 2013

Mishne, G., Shaham, U., Cloninger, A., & Cohen, I. *Diffusion nets*. Applied and Computational Harmonic Analysis (2017).

MONITORING THE RECONSTRUCTION ERROR

Detection by reconstruction error monitoring (AE notation)

Training (Monitoring the Reconstruction Error):

1. Train the model $\mathcal{D}(\mathcal{E}(\cdot))$ from the training set TR
2. Learn the distribution of reconstruction errors

$$\mathrm{err}(\boldsymbol{s}) = \left\|\boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big)\right\|_2, \qquad \boldsymbol{s} \in V$$

over a validation set $V \neq TR$ and define a suitable threshold γ

Testing (Monitoring the Reconstruction Error):

1. Perform encoding and compute the reconstruction error

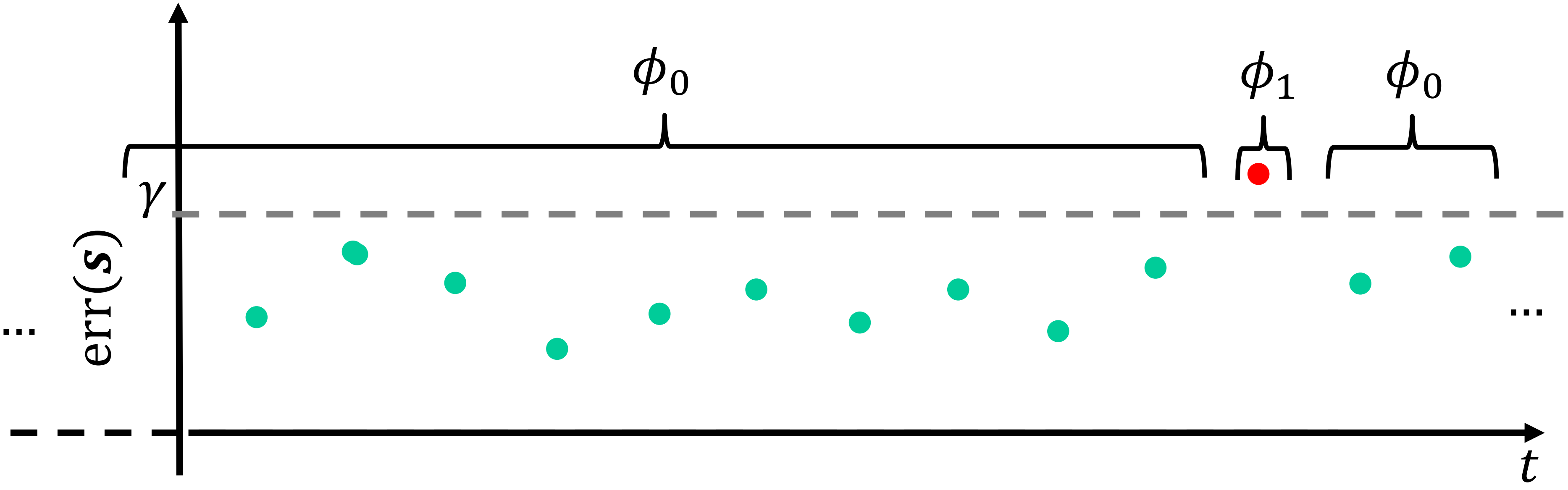
$$\mathrm{err}(\boldsymbol{s}) = \left\|\boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big)\right\|_2$$

2. Consider $\boldsymbol{s}$ anomalous when $\mathrm{err}(\boldsymbol{s}) > \gamma$

MONITORING THE RECONSTRUCTION ERROR

Normal data are expected to yield values of $\text{err}(\boldsymbol{s})$ that **are low**, while anomalies do not. This holds when the model $\mathcal{M}$ was specifically learned to describe normal data

Outliers can be detected by a threshold on $\text{err}(\boldsymbol{s})$



OUTLINE ON SEMI-SUPERVISED APPROACHES

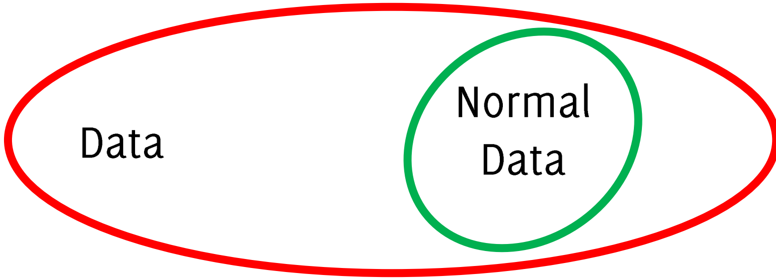
Out of the "Random Variable" world

- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Extended models

SUBSPACE METHODS

The underlying assumption is that

- **normal patches live in a subspace** that can be identified by TR
- **anomalies can be detected by projecting test patches** in such subspace and by monitoring the reconstruction error (distance with the projection)

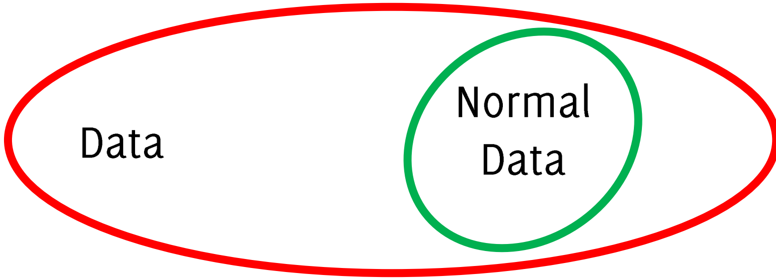


V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

SUBSPACE METHODS

A few example of **models used for describing normal patches**:

- Fourier transform: normal patches can be expressed by a few specific frequencies
- PCA: normal patches live in the linear subspace of the first components.
- Robust PCA: defined on the ℓ^1 distance to be insensitive to outliers in normal data
- Kernel PCA: normal patches live in a non-linear manifold
- Random projections



V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

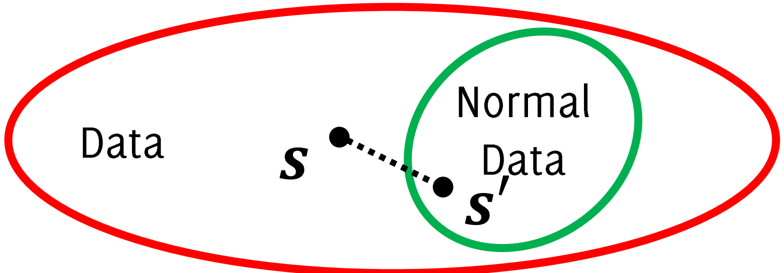
SUBSPACE METHODS: STATISTICS

Examples of **statistics for PCA monitoring** (and similar techniques):

- The **projection on the subspace**,
$$\boldsymbol{s}' = P^T\boldsymbol{s}, \qquad P \in \mathbb{R}^{m\times d}, \qquad m \ll d$$
which is the projection over the first m principal components and a way to reduce data-dimensionality. Statistical techniques can be applied to **monitor projections** $P^T\boldsymbol{s}$

- The **least-principal component**, which resembles an anomaly score: low in normal patches, increases for anomalies.

- The **reconstruction error**:
$$\text{err}(\mathbf{s}) = \ \|\boldsymbol{s} \ - PP^T\boldsymbol{s}\|_2$$
which is the distance between $\boldsymbol{s}$ and its projection $PP^T\boldsymbol{s}$ over the subspace of normal patches



SUBSPACE METHODS: SPARSE REPRESENTATIONS

Basic assumption: normal data live in **a union of low-dimensional subspaces** of the input space

The model learned from S is a matrix: the **dictionary** D.

Each signal is decomposed as **a sum of a few dictionary atoms** (representation is constrained to be **sparse**).

Atoms represent the many **building blocks** that can be used to reconstruct normal signals.

There are typically **more atoms** than the signal dimension.

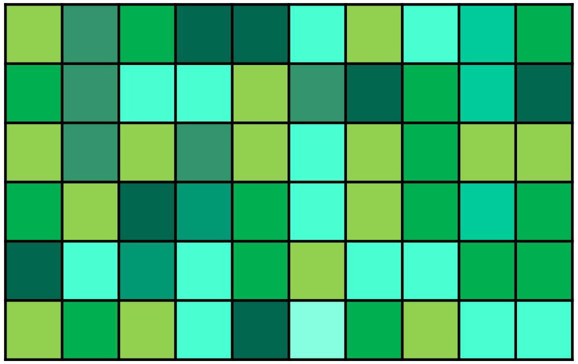
Effective as long as the learned **dictionary** D is **very specific for normal data**

M. Elad *"Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing"*, Springer, 2010

DICTIONARIES YIELDING SPARSE REPRESENTATIONS

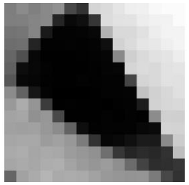
Dictionaries are just matrices! $D \in \mathbb{R}^{d\times m}$

$D =$



DICTIONARIES YIELDING SPARSE REPRESENTATIONS

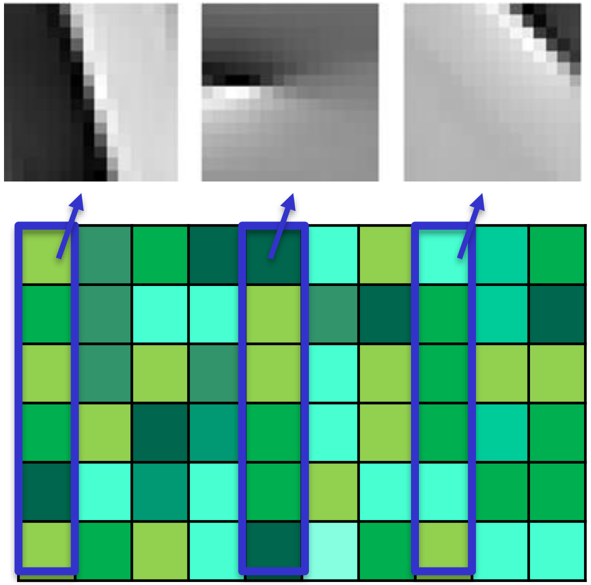
Dictionaries are just matrices! $D \in \mathbb{R}^{d \times m}$



$\boldsymbol{s}$

Each column is an atom:

- lives in the input space
- it is one of the learned building blocks to reconstruct the input signal



$D =$

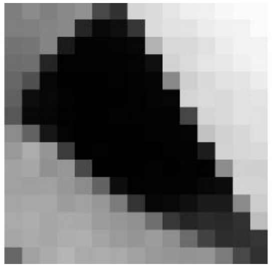
SPARSE REPRESENTATIONS

Let $\boldsymbol{s} \in \mathbb{R}^n$ be the input signal, a sparse representation is

$$\boldsymbol{s} = \sum_{i=1}^{M} \alpha_i \, \boldsymbol{d_i}$$

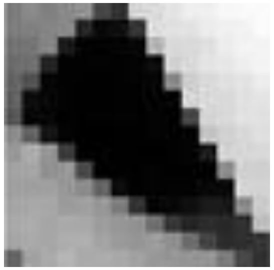
a linear combination of **few dictionary atoms** $\{\boldsymbol{d_i}\}$, i.e., most of coefficients are such that $\alpha_i = 0$

An illustrative example in case of our patches

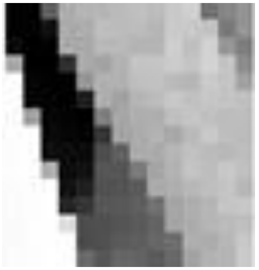


$= 0.7 \; *$

$+0.1 \; *$



$-0.2 \; *$

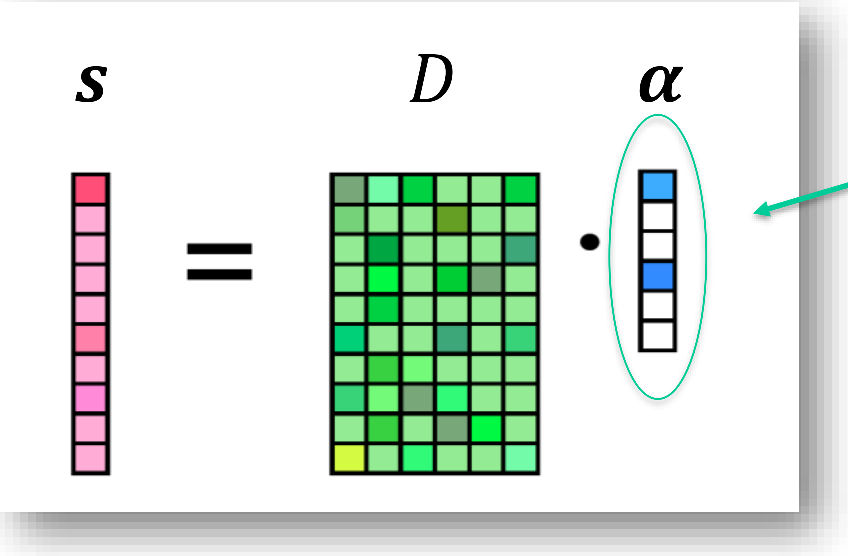


SPARSE REPRESENTATIONS… MATRIX EXPRESSION

Let $\boldsymbol{s} \in \mathbb{R}^n$ be the input signal, a sparse representation is

$$\boldsymbol{s} = \sum_{i=1}^{M} \alpha_i\, \boldsymbol{d_i} = D\boldsymbol{\alpha}$$

a linear combination of **few dictionary atoms** $\{\boldsymbol{d_i}\}$ and $\|\boldsymbol{\alpha}\|_0 < L$, i.e. only a few coefficients are nonzero, i.e. $\boldsymbol{\alpha}$ is sparse.



This vector
$\boldsymbol{\alpha} = [\alpha_1, \quad \dots, \quad \alpha_M]$
is **sparse**

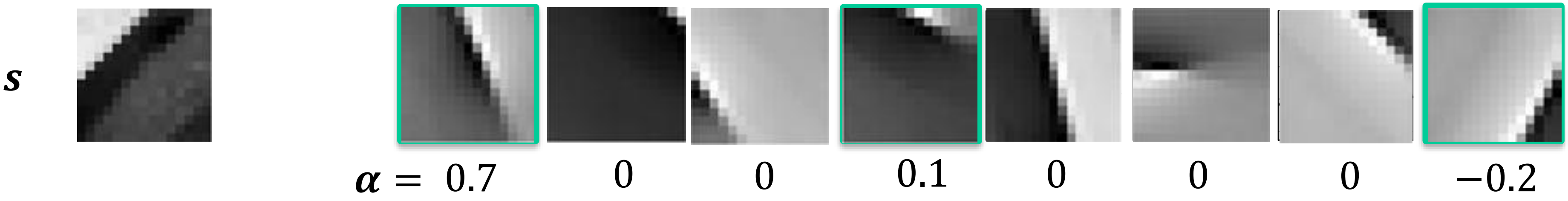
SPARSE CODING…

Sprase Coding: computing the sparse representation for an input signal $\boldsymbol{s}$ w.r.t. D

$$\boldsymbol{s} \in \mathbb{R}^d \quad \longrightarrow \quad \boldsymbol{\alpha} \in \mathbb{R}^n$$

It is solved as the following optimization problem, (e.g. via the Orthogonal Matching Pursuit, OMP)

$$\boldsymbol{\alpha} = \underset{\boldsymbol{a} \in \mathbb{R}^n}{\operatorname{argmin}} \; \| D\boldsymbol{a} - s \|_2 \;\; \text{s.t.} \;\; \|\boldsymbol{a}\|_0 < L$$



In the previous illustration $\boldsymbol{\alpha} = [0.7, 0, 0, 0.1, 0, 0, 0, -0.2]$

Pati, Y.; Rezaiifar, R.; Krishnaprasad, P. *Orthogonal Matching Pursuit: recursive function approximation with application to wavelet decomposition.* Asilomar Conf. on Signals, Systems and Comput. 1993

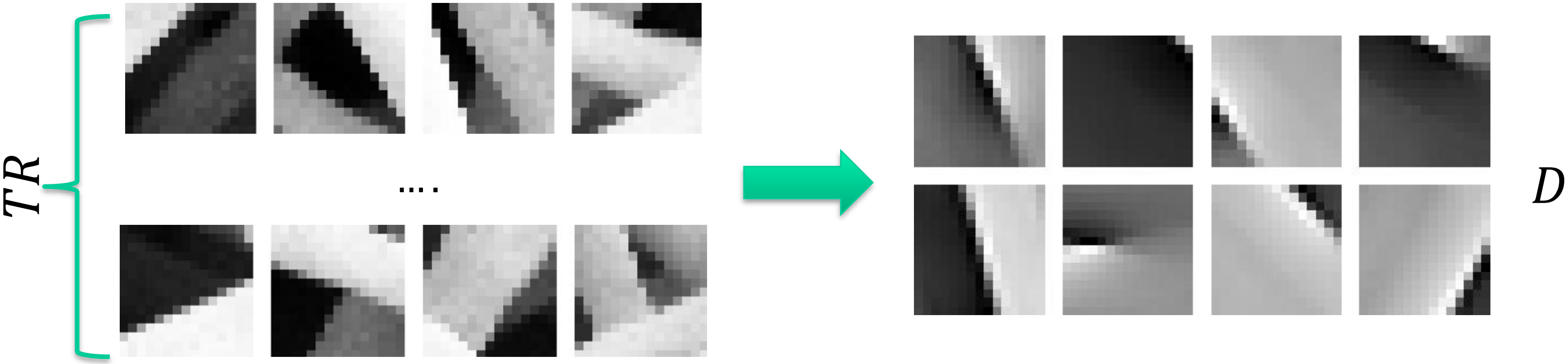
… AND DICTIONARY LEARNING

Dictionary Learning: estimate D from a training set of M

$$S = \{\boldsymbol{s}_1, \dots \boldsymbol{s}_M\} \xrightarrow{\in \ \mathbb{R}^{d\times n}} D \in \mathbb{R}^{d\times n}$$

It is solved as the following optimization problem typically through **block-coordinates descent** (e.g. KSVD algorithm)

$$[D, X] = \underset{A\in \mathbb{R}^{d\times n},\, Y\in \mathbb{R}^{n\times M}}{\operatorname{argmin}} \|AY - S\|_2 \ \text{s.t.} \ \|\boldsymbol{y}_i\|_0 < L, \quad \forall \boldsymbol{y}_i$$



Aharon, M.; Elad, M. Bruckstein, A. *K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation* IEEE TSP, 2006

SPARSE REPRESENTATION MONITORING: STATISTICS

Anomalies can be directly **detected during the sparse coding** stage, by changing the functional being optimized.

A set of test signals is modeled as:

$$S = DX + E + V$$

where X is sparse, V is a noise term, and E is a matrix having most columns set to zero. Columns $\boldsymbol{e}_i \neq \mathbf{0}$ indicate anomalies, as they do not admit a sparse representation w.r.t. D

A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, *"Sparse coding with anomaly detection"* Journal of Signal Processing Systems, vol. 79, no. 2, pp. 179–188, 2015.

SPARSE REPRESENTATION MONITORING: STATISTICS

Anomalies can be detected by solving (through ADMM) the following sparse coding problem

$$\underset{X,E}{\operatorname{argmin}}\left(\frac{1}{2}\|S-DX\ -E\|_F^2+\lambda\|X\|_1+\mu\|E\|_{2,1}\right)$$

Data-fidelity for normal data

Sparsity

Group sparsity regularization, only a few columns can be nonzero

.. and identifying as anomalies the signals corresponding to columns of E that are nonzero.

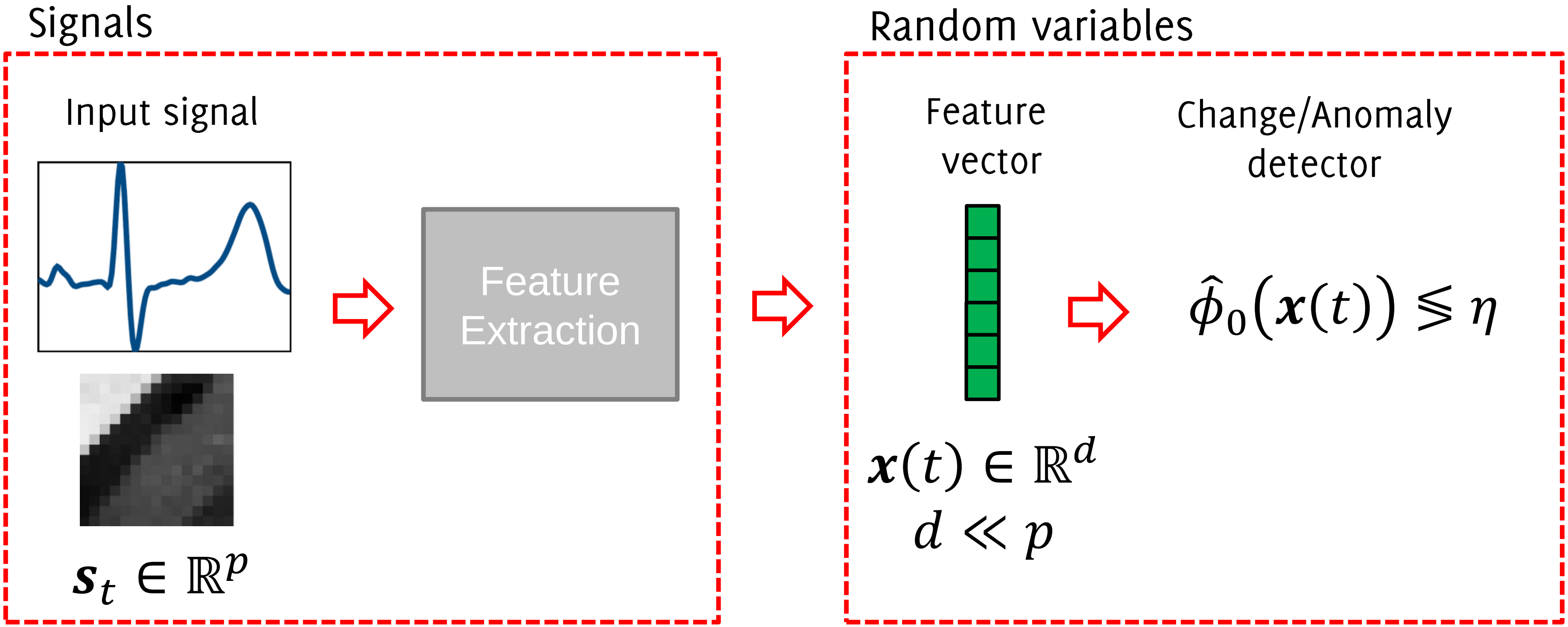
A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, *"Sparse coding with anomaly detection"* Journal of Signal Processing Systems, vol. 79, no. 2, pp. 179–188, 2015.

OUTLINE ON SEMI-SUPERVISED APPROACHES

- Detrending/Filtering for time-series
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Extended models

MONITORING FEATURES

Feature extraction: meaningful indicators to be monitored which have a known / controlled response w.r.t. normal data



Signals

Input signal

$\boldsymbol{s}_t \in \mathbb{R}^p$

Feature Extraction

Random variables

Feature vector

$\boldsymbol{x}(t) \in \mathbb{R}^d$

$d \ll p$

Change/Anomaly detector

$\hat{\phi}_0(\boldsymbol{x}(t)) \lessgtr \eta$

Feature Extraction: signal processing, a priori information, learning methods

The customary framework for change / anomaly detection

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

FEATURE EXTRACTION

The peculiar structures of normal images and signals suggest that **normal data live in a manifold** having **lower dimension** than the input domain

Data dimensionality can be reduced by extracting features

Good features should:

- Yield a **stable response** w.r.t. **normal data**
- Yield **unusual response** on **anomalies** / when data change

Reconstruction error and representation coefficients can be considered features.

Features can be monitored in either one-shot/sequential monitoring schemes.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

FEATURE EXTRACTION APPROACHES

There are two major approaches for extracting features:

Expert-driven (hand-crafted) features: computational expressions that are **manually designed by experts** to distinguish between normal and anomalous data

Data-driven features: **features** characterizing normal data are automatically **learned from training data** TR

OUTLINE ON SEMI-SUPERVISED APPROACHES

- Detrending/Filtering for time-series
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Extended models

EXAMPLES OF EXPERT-DRIVEN FEATURES

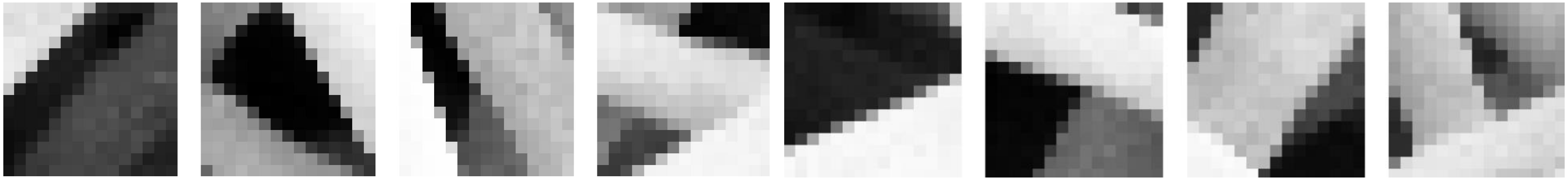
Expert-driven features: each patch of an image s

$$\mathbf{s}_c = \{s(c+u), u \in \mathcal{U}\}$$

Example of features are:

- the average,
- the variance,
- the total variation (the energy of gradients)

These can hopefully **distinguish normal** and **anomalous** patches (since image in anomalous region is expected to be flat or without edges characterizing normal regions)



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

OUTLINE ON SEMI-SUPERVISED APPROACHES

- Detrending/Filtering for time-series
- Reconstruction-based methods
    - Subspace methods
- Feature-based monitoring
    - Expert-driven Features
    - Data-driven Features
    - Extended models

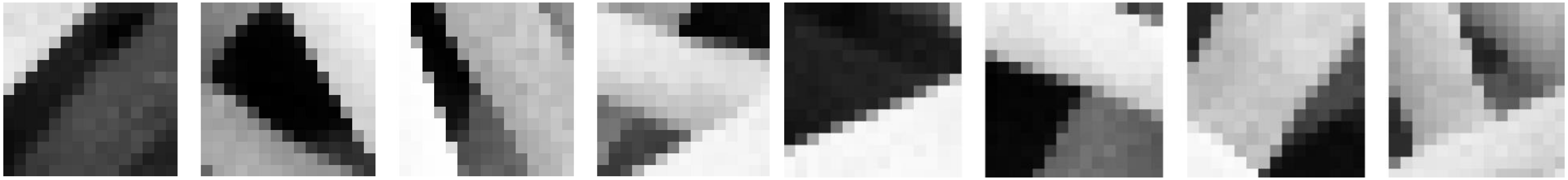
EXAMPLES OF DATA-DRIVEN FEATURES

Analyze each patch of an image s

$$\mathbf{s}_c = \{s(c+u), u \in \mathcal{U}\}$$

and determine whether it is normal or anomalous.

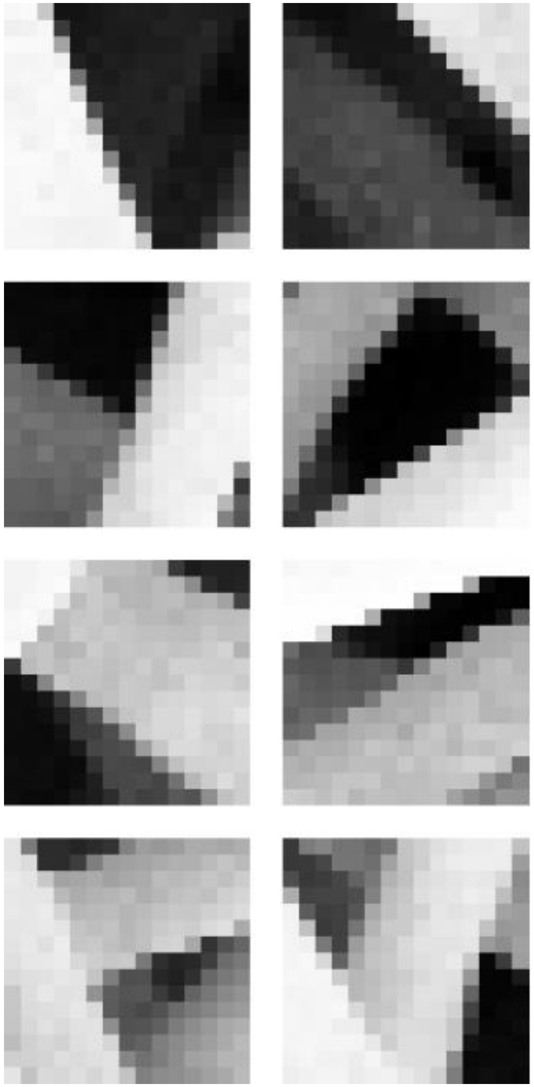
Data driven features: expressions to **quantitatively assess whether test patches conform** or not **with the model**, learned from normal data.



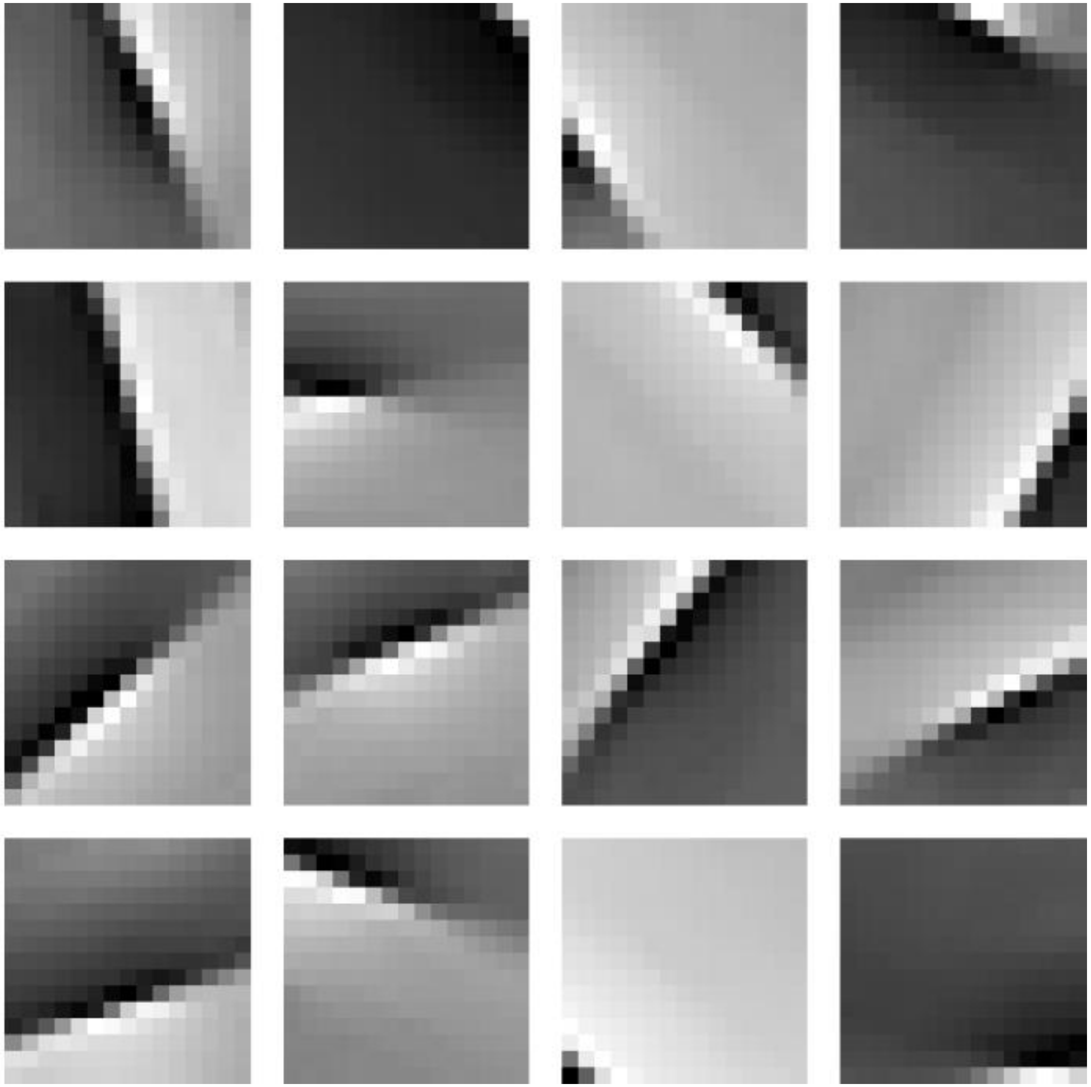
Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

A LEARNED DICTIONARY OF NORMAL PATCHES

Example of training patches



Few learned atoms (BPDN-based learning)



DATA-DRIVEN FEATURES

To assess the conformance of $\boldsymbol{s}_c$ with D we solve the following

Sparse coding:

$$\boldsymbol{\alpha} = \underset{\widetilde{\boldsymbol{\alpha}} \in \mathbb{R}^n}{\operatorname{argmin}} \|D\widetilde{\boldsymbol{\alpha}} - \mathbf{s}\|_2^2 + \lambda \|\widetilde{\boldsymbol{\alpha}}\|_1, \qquad \lambda > 0$$

which is the BPDN formulation and we solve using ADMM.

The penalized ℓ^1 formulation has more degrees of freedom in the reconstruction, **the conformance of $\boldsymbol{s}$ with $\boldsymbol{D}$ have to be assessed monitoring both terms of the functional**

Boyd S., Parikh N., Chu E., Peleato B., Eckstein J., *"Distributed optimization and statistical learning via the alternating direction method of multipliers"* 2011

DATA-DRIVEN FEATURES

Features then include both the **reconstruction error**

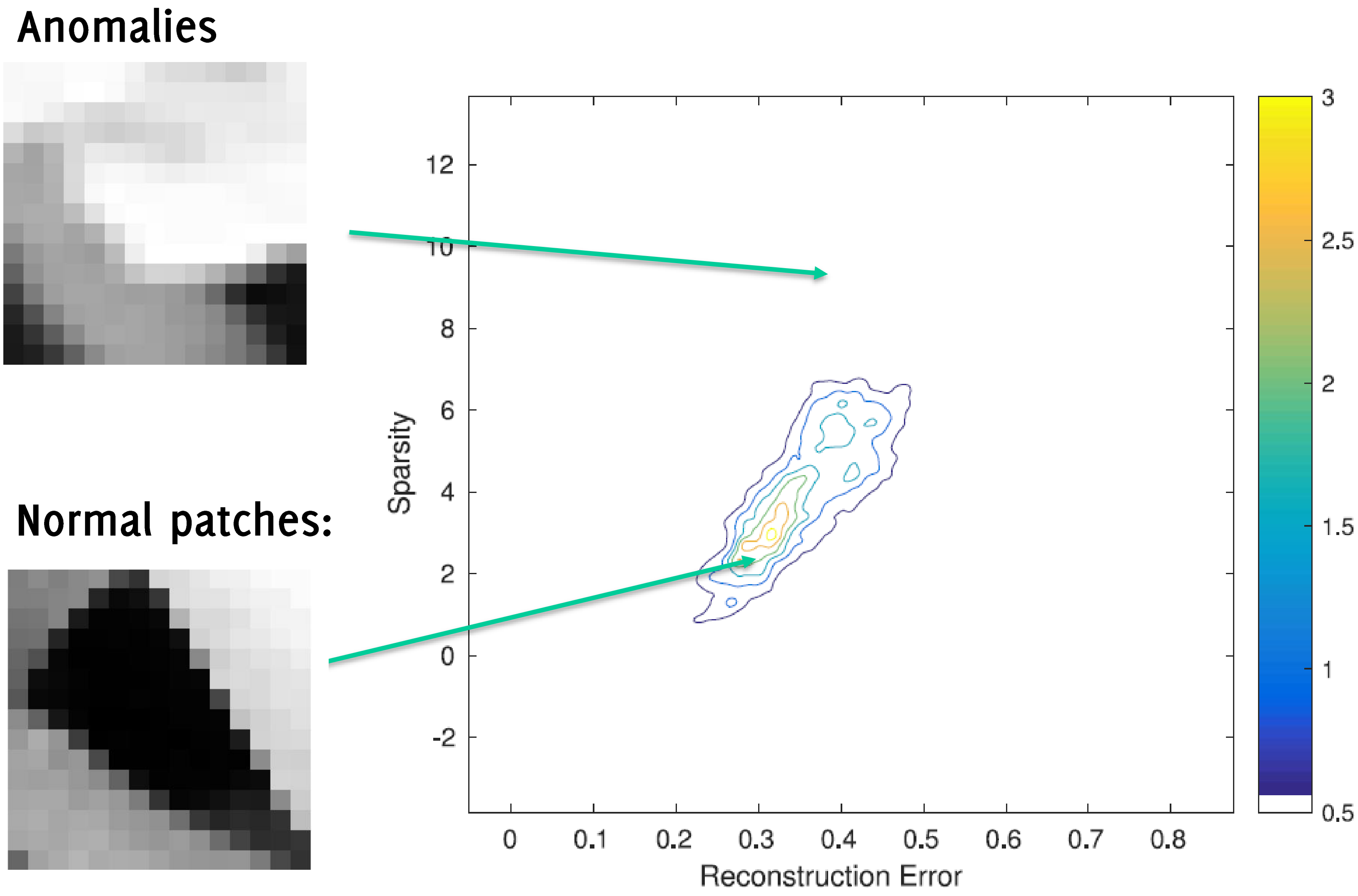
$$\mathrm{err}(\boldsymbol{s}) = \|D\boldsymbol{\alpha} - \mathbf{s}\|_{\mathbf{2}}^{\mathbf{2}}$$

and **the sparsity** of the representation

$$\|\boldsymbol{\alpha}\|_{\mathbf{1}}$$

Thus obtaining **a data-driven feature vector** $\boldsymbol{x} = \begin{bmatrix} \|D\boldsymbol{\alpha} - \mathbf{s}\|_{\mathbf{2}}^{\mathbf{2}} \\ \|\boldsymbol{\alpha}\|_1 \end{bmatrix}$

DENSITY-BASED MONITORING ON DATA-DRIVEN FEATURES



Anomalies

Normal patches:

DATA-DRIVEN FEATURES

Training:

- **Learn** from $TR\setminus V$ the dictionary D
- **Learn** from V, the distribution $\hat{\phi}_0$ of normal features vectors $\boldsymbol{x}$.

Testing:

- Compute feature vectors $\boldsymbol{x}$ via sparse coding
- Detect anomalies when $\hat{\phi}_0(\boldsymbol{x}) < \eta$

Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

DATA-DRIVEN FEATURES

Training:

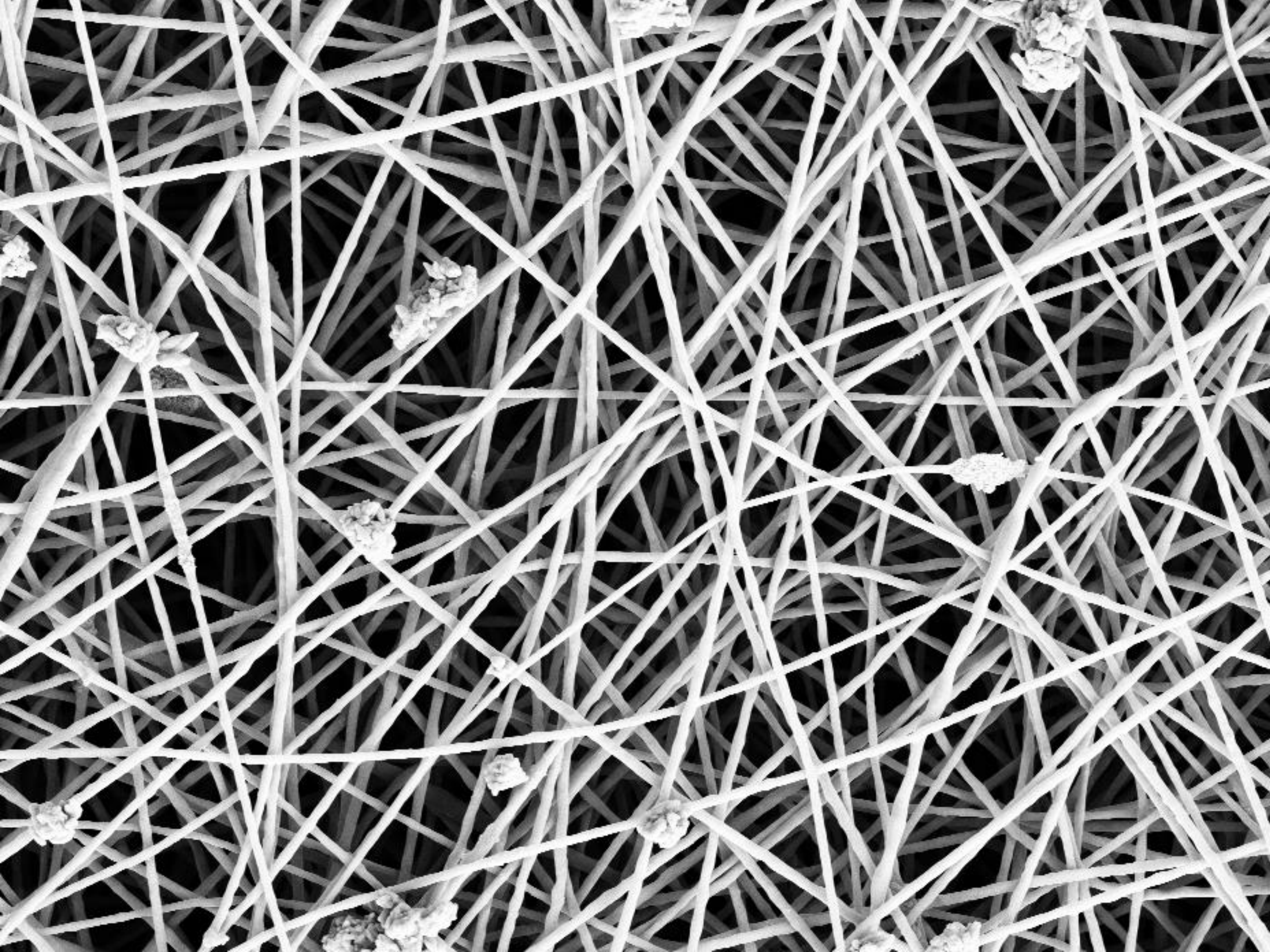
- **Learn** from $TR\backslash V$ the dictionary D
- **Learn** from V, the distribution $\hat{\phi}_0$ of normal features vectors $\boldsymbol{x}$.

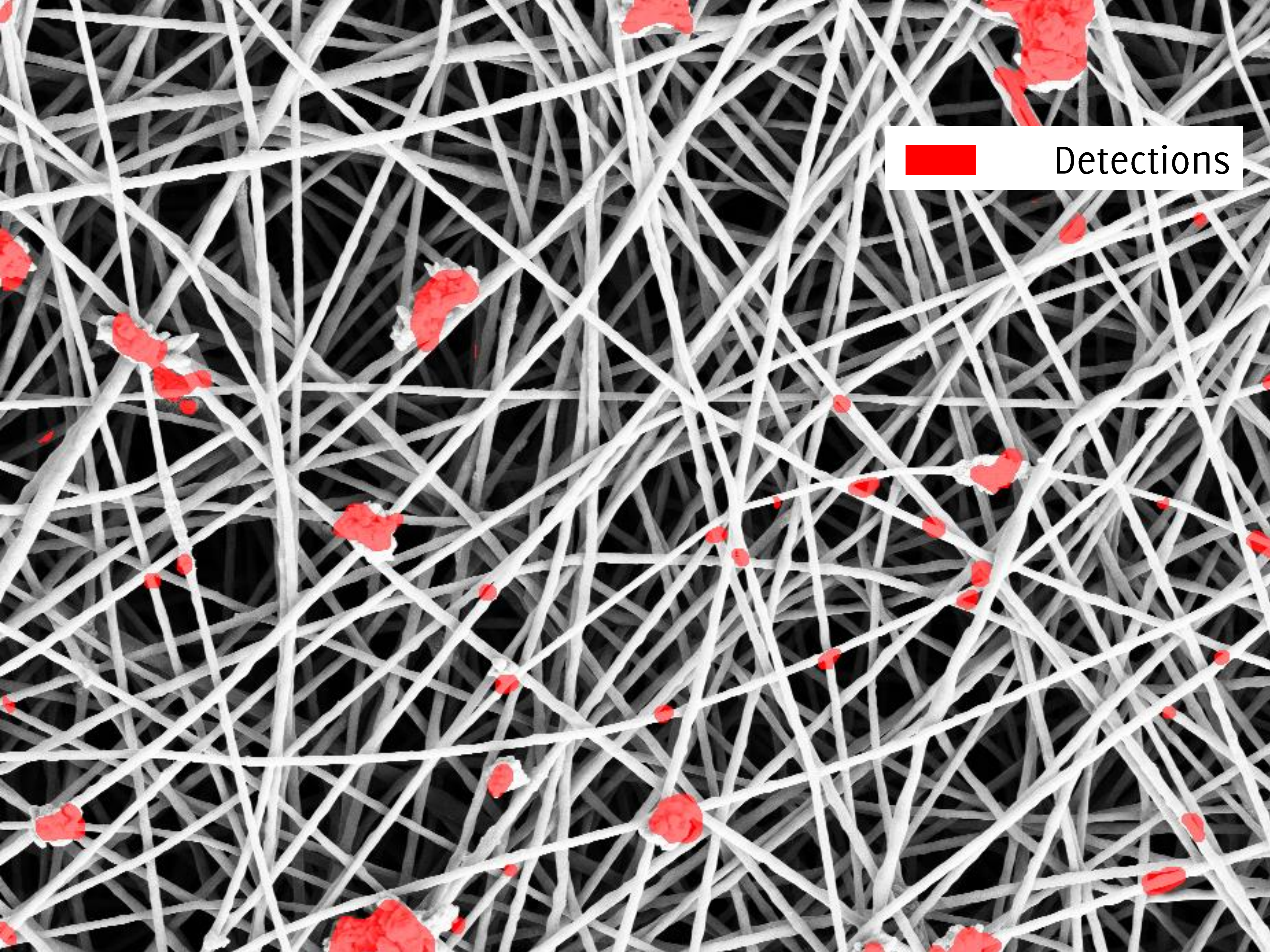
Testing:

- Compute feature vectors $\boldsymbol{x}$ via sparse coding
- Detect anomalies when $\hat{\phi}_0(\boldsymbol{x}) < \eta$

This solution is rather flexible and can be adapted when operating conditions changes (e.g. different zooming level)

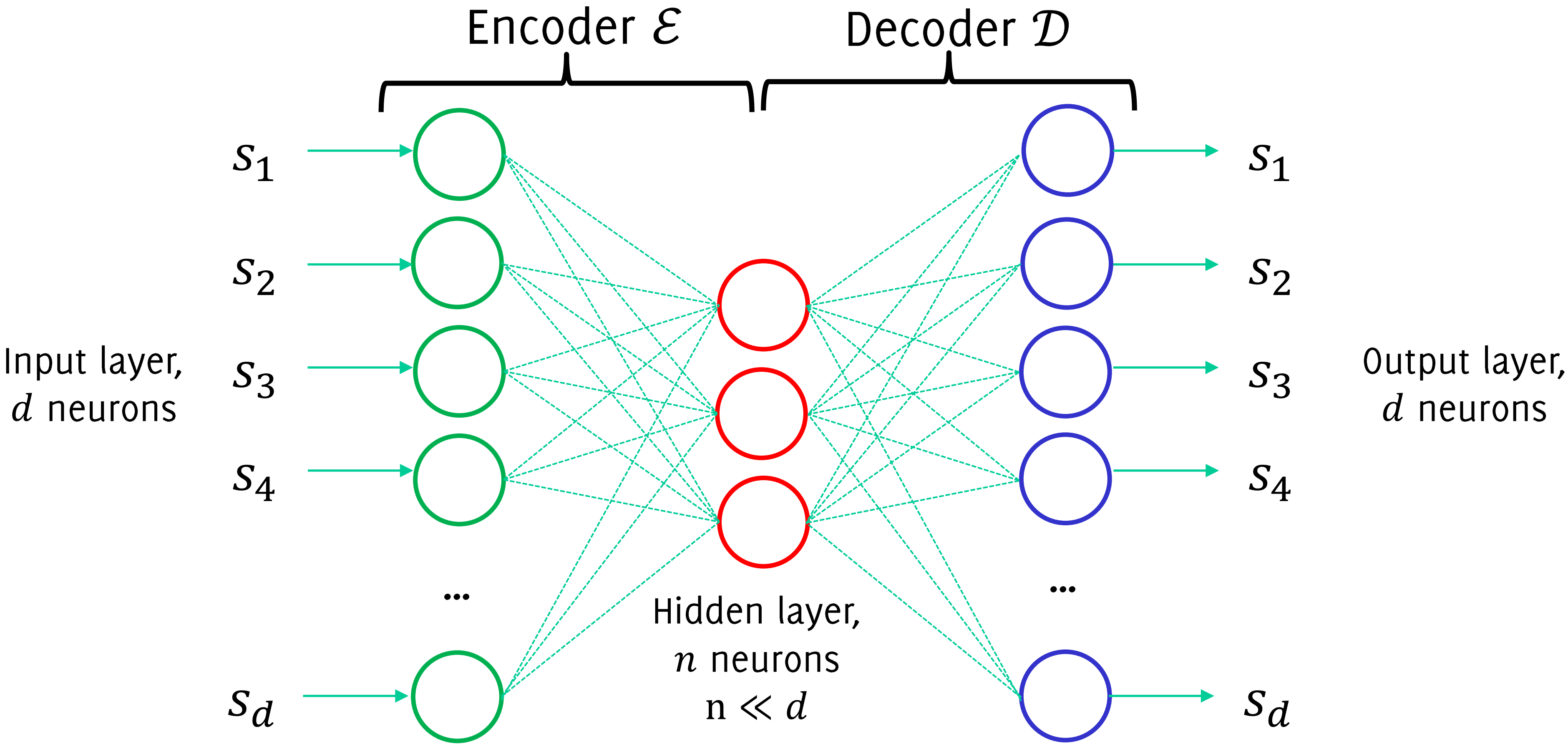
Carrera D., Boracchi G., Foi A., Wohlberg B. *"Scale-invariant Anomaly Detection With multiscale Group-sparse Models"* ICIP 2016




🟥 Detections

FEATURE-BASED METHODS

Autoencoders can be also used in feature-based monitoring schemes, where the hidden representation of the input is the feature being monitored



MONITORING FEATURE DISTRIBUTION

Detection by **feature monitoring** (AE notation)

Training (Monitoring Feature Distribution):

- Learn the autoencoder $\mathcal{D}(\mathcal{E}(\cdot))$ from the training set S
- Fit a density model $\hat{\phi}_0$ to the encoded features
$$\{\mathcal{E}(\boldsymbol{s}), \boldsymbol{s} \in V\}$$
over a validation set $V \neq S$
- Define a suitable threshold γ for $\hat{\phi}_0(\boldsymbol{s})$

Testing (Monitoring Feature Distribution):

- Encode each incoming signal $\boldsymbol{s}$ through $\mathcal{E}$
- Detect anomalies if $\hat{\phi}_0(\mathcal{E}(\boldsymbol{s})) < \gamma$

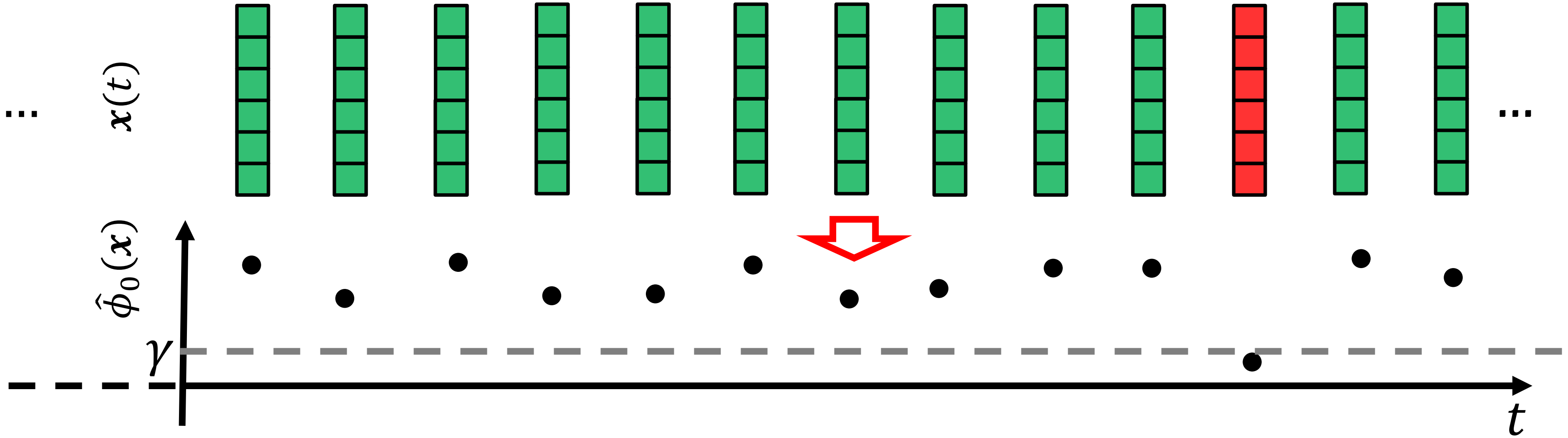
MONITORING FEATURE DISTRIBUTION

Normal data are expected to yield $\mathcal{E}(\boldsymbol{s})$ that are **i.i.d. vectors (or features)** and that **follow an** unknown **distribution** ϕ_0.

Anomalous data do not, as they follow $\phi_1 \neq \phi_0$.

We are **back to our statistical framework** and we can

- learn $\hat{\phi}_0$ from a set features extracted from normal data
- detect anomalous data by computing $\boldsymbol{x} = \mathcal{E}(\boldsymbol{s})$ and then check whether $\hat{\phi}_0(\boldsymbol{x}) < \gamma$



Reference-based approaches

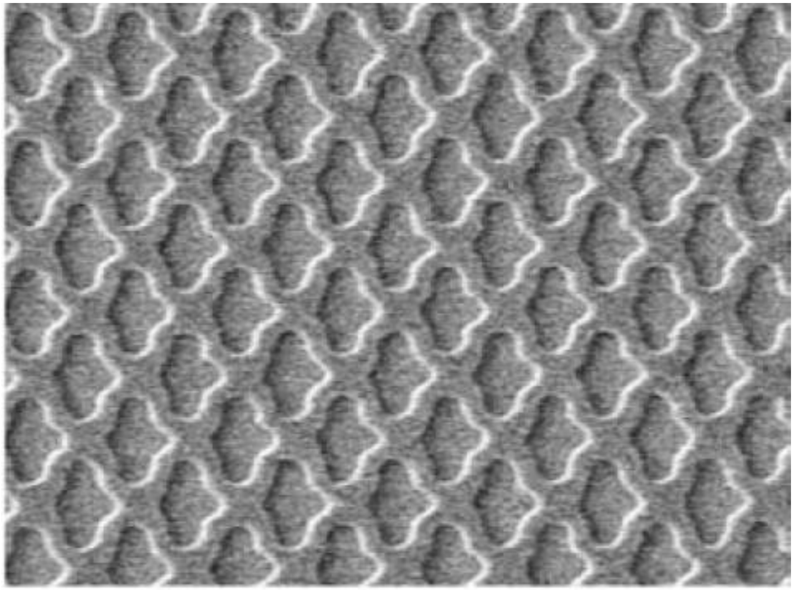
REFERENCE BASED-METHODS

In some cases anomalies can be detected by comparing

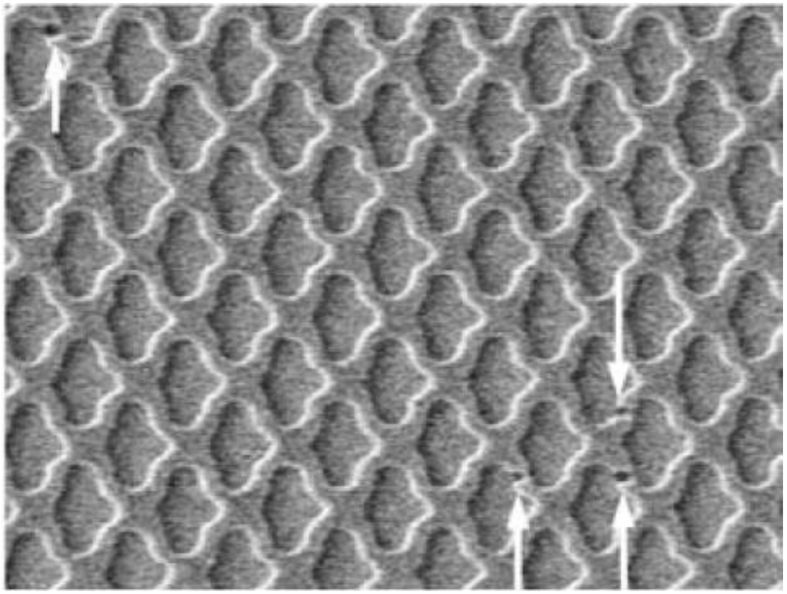
- the **target**, namely the image to be tested
- against a **reference**, namely an anomaly-free image

Examples include: a pair of temporally close images (in SAR and remote sensing) images for quality inspection.

reference



target



Zontak, M., Cohen, I.: *Defect detection in patterned wafers using anisotropic kernels.*" Machine Vision and Applications 21(2), 129{141 (2010)

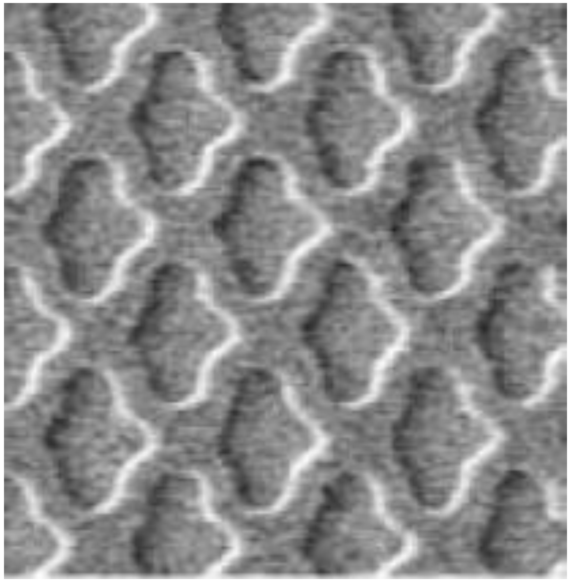
REFERENCE BASED-METHODS

In some cases anomalies can be detected by comparing

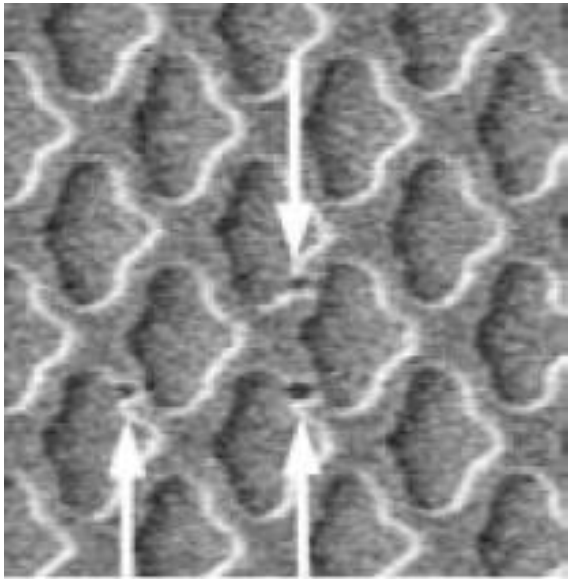
- the **target**, namely the image to be tested
- against a **reference**, namely an anomaly-free image

Examples include: a pair of temporally close images (in SAR and remote sensing) images for quality inspection.

reference



target

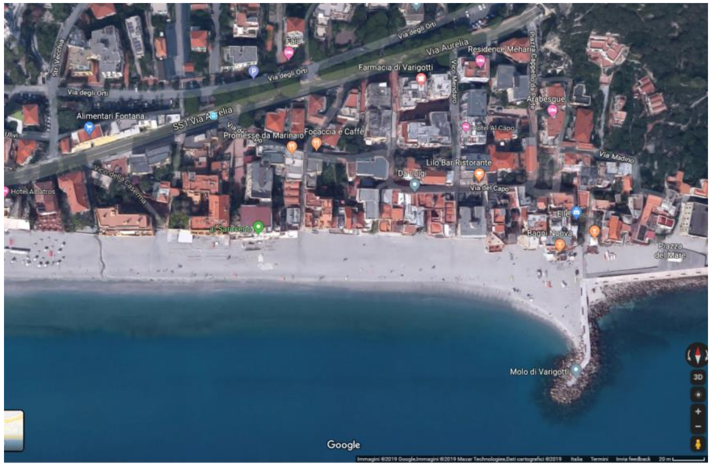


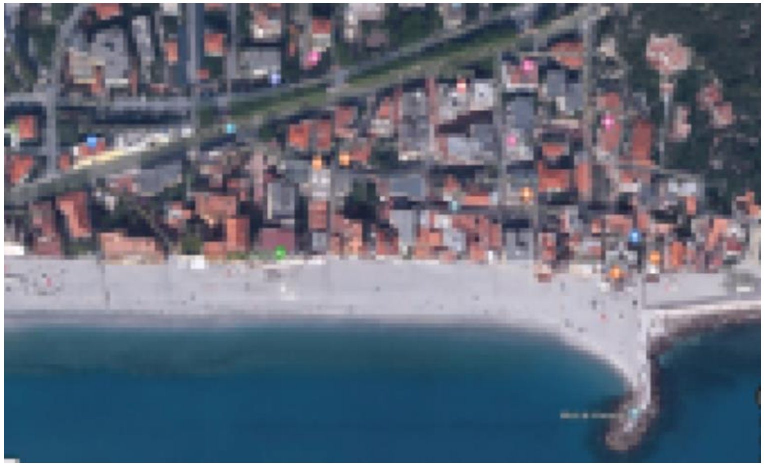
Zontak, M., Cohen, I.: *Defect detection in patterned wafers using anisotropic kernels.*" Machine Vision and Applications 21(2), 129{141 (2010)

REFERENCE BASED-METHODS

Non trivial when direct comparison is prevented:

- Reference and **target might not be aligned nor easy to register with a global transformation**
- Reference and target might be **from different modalities** or resolution (e.g. a SAR image and an optical image)





Zontak, M., Cohen, I.: *Defect detection in patterned wafers using anisotropic kernels."* Machine Vision and Applications 21(2), 129{141 (2010)

L. T. Luppino, F. M. Bianchi, G. Moser, S. N. Anfinsen, *"Unsupervised Image Regression for Heterogeneous Change Detection"* IEEE Transactions on Geoscience and Remote Sensing (2019)

Deep Learning Approaches

OUTLINE

- Supervised approaches
  - Detection by image classification
- Semi-supervised approaches
  - Neural Networks as feature extractors
  - Generative Models

Supervised Approaches

Detection by Image Classification

IMAGE CLASSIFICATION

The problem: assigning to an *input image* $\boldsymbol{s}$ one *label* l from a *fixed set of* L *categories* Λ

$\boldsymbol{s}$ ⇨ “wheel”

$\boldsymbol{s}$ ⇨ “castle”

$$\Lambda = \{\text{"wheel"}, \text{"cars"} \ldots . . \\ \ldots\ldots \text{"castle"}, \text{"baboon"}, \ldots \}$$

IMAGE CLASSIFICATION

The problem: assigning to an *input image* $\boldsymbol{s}$ one *label* l from a *fixed set of* L *categories* Λ

$\boldsymbol{s}$ ⇨ “wheel” 65%, “tyre” 30%..

$\boldsymbol{s}$ ⇨ “castle” 55%, “tower” 43%..

$$\Lambda = \{\text{"wheel"}, \text{"cars"} \ldots\ldots \\ \ldots\ldots\text{"castle"}, \text{"baboon"}, \ldots\}$$

DEEP LEARNING AND IMAGE CLASSIFICATION

Since 2010 ImageNet organizes **ILSVRC** (ImageNet Large Scale Visual Recognition Challenge)

Classification error rate (top 5 accuracy):

- In 2011: 25%

Deep learning

- In 2012: 16% (achieved by a CNN)
- In 2017: < 5% (for 29 of 38 competing teams, deep learning)

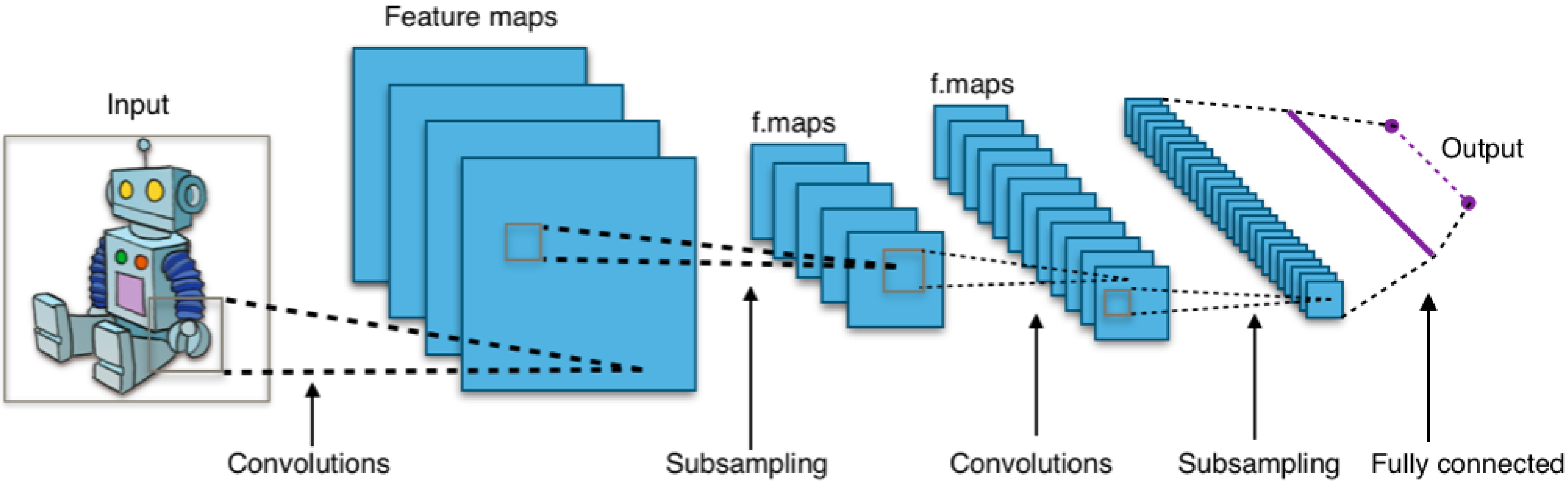
DEEP LEARNING AND IMAGE CLASSIFICATION

Deep Learning boasted image classification performance, thanks to

- Advances in parallel hardware (e.g. GPU)
- Availability of large annotated dataset (e.g. the **ImageNet project** is a large database visual recognition over **14M hand-annotated images** in more than **20K categories**)

CONVOLUTIONAL NEURAL NETWORKS (CNN)

The typical architecture of a convolutional neural network



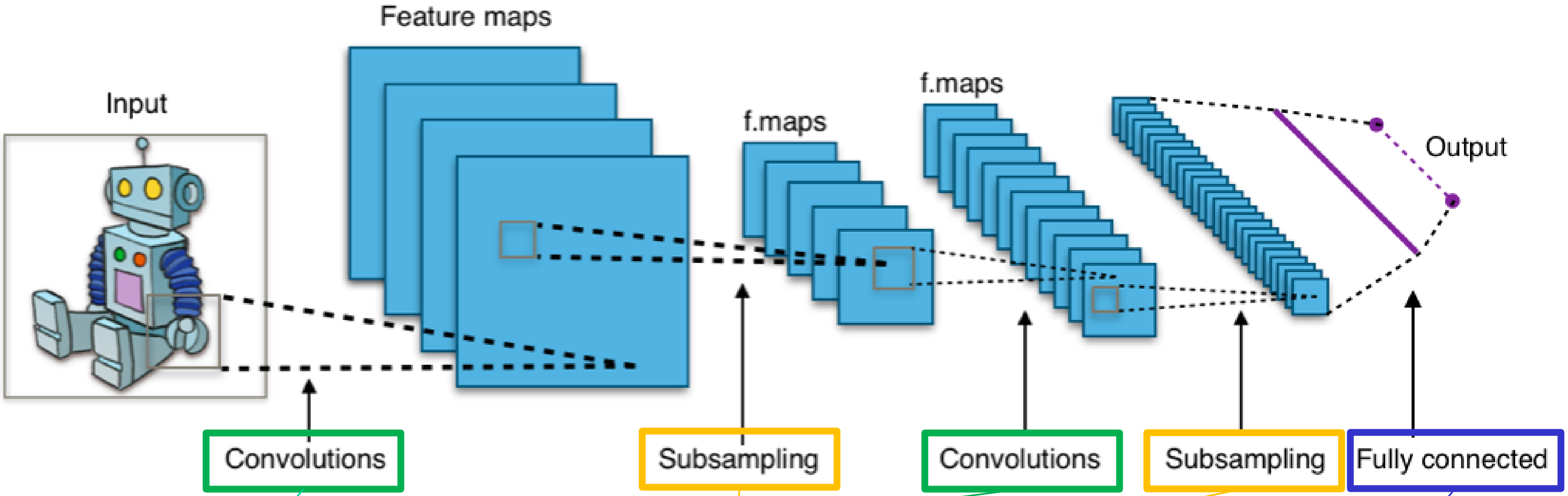
By Aphex34 - Own work, CC BY-SA 4.0,
https://commons.wikimedia.org/w/index.php?curid=45679374

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. *"Gradient-based learning applied to document recognition"*

Proceedings of the IEEE, 1998 86(11), 2278-2324

CONVOLUTIONAL NEURAL NETWORKS (CNN)

The typical architecture of a convolutional neural network



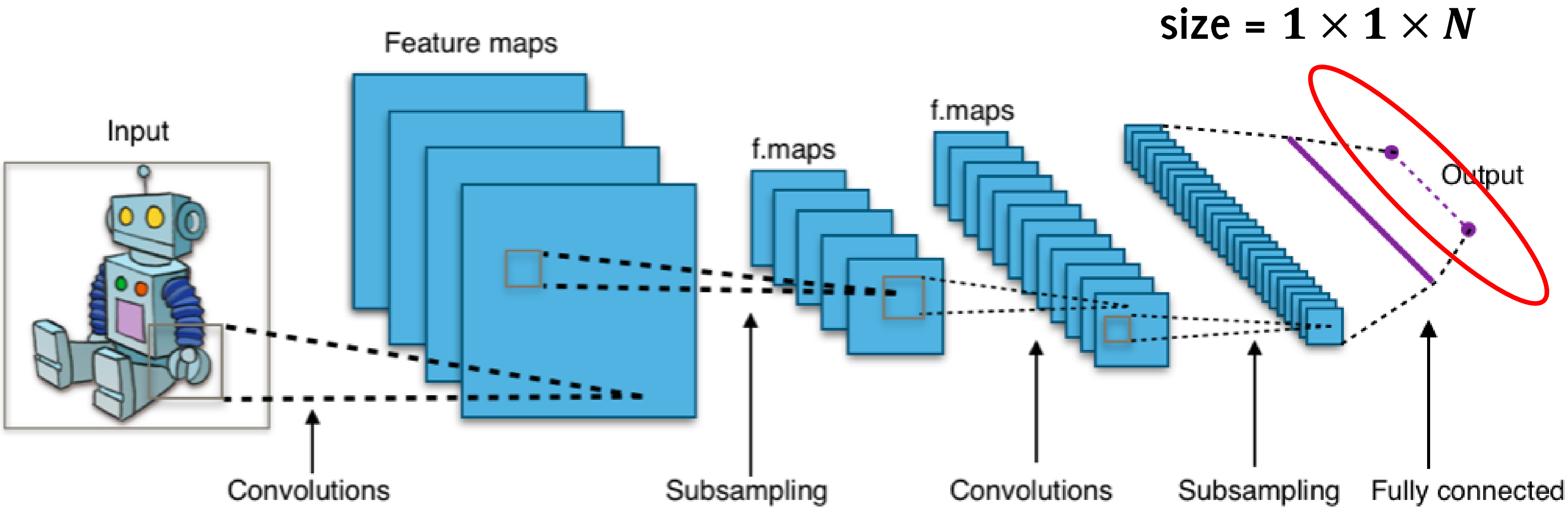
Convolution filters are learned for the classification task at hand

Thresholding + Downsampling (ReLu + Maxpooling)

Fully connected Neural Network providing as output the class scores

CONVOLUTIONAL NEURAL NETWORKS (CNN)

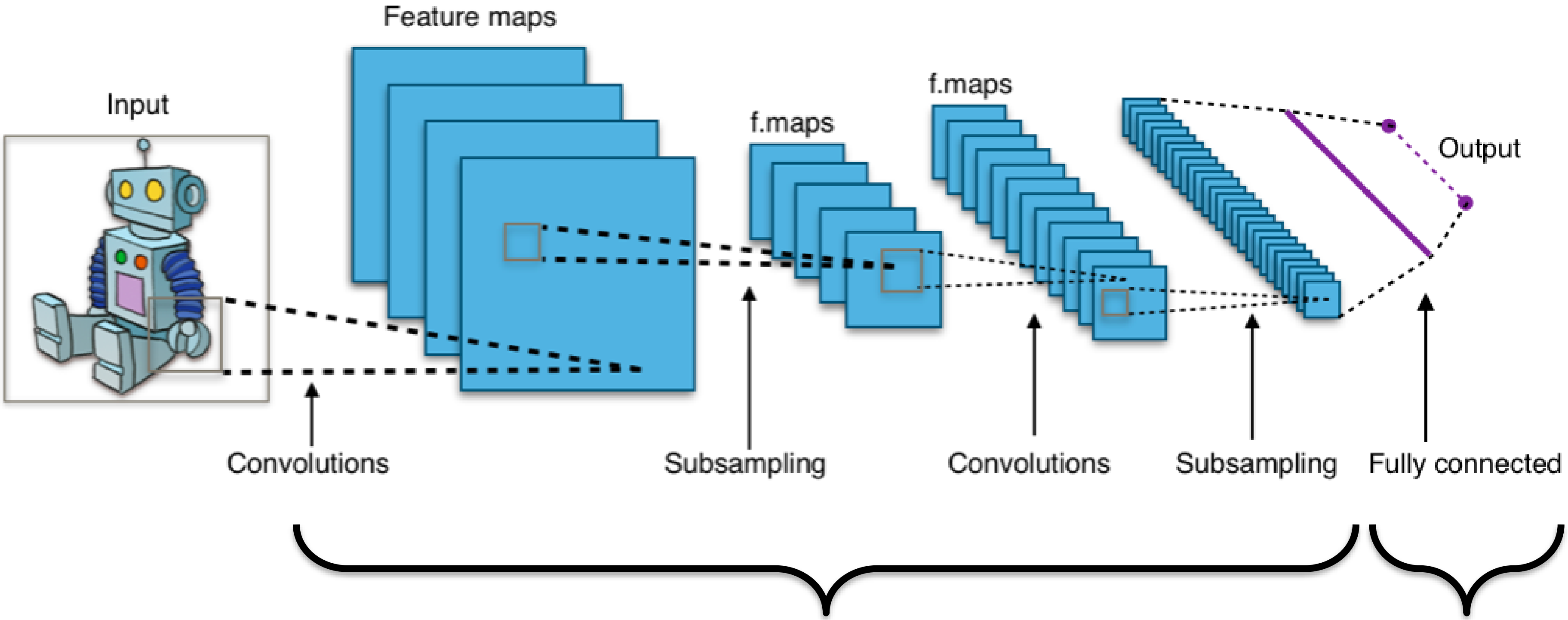
The typical architecture of a convolutional neural network



The output of the fully connected layer has the same size as the number of classes, and each component provide a score for the input image to belong to a specific class

CONVOLUTIONAL NEURAL NETWORKS (CNN)

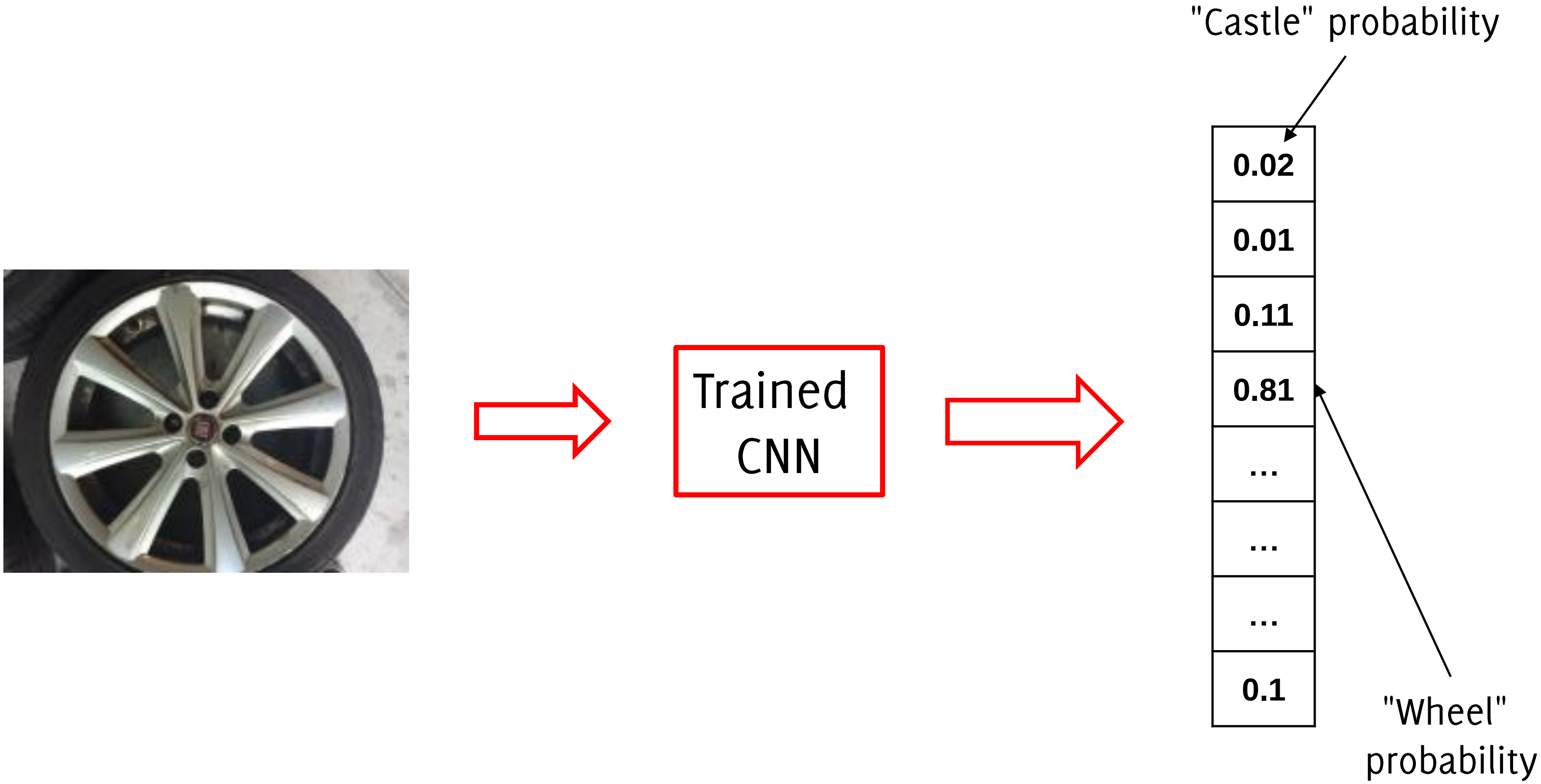
The typical architecture of a convolutional neural network



Extract high-level features from pixel data

Classify

THE OUTPUT OF A CNN



Trained CNN

"Castle" probability

0.02

0.01

0.11

0.81

…

…

…

0.1

"Wheel" probability

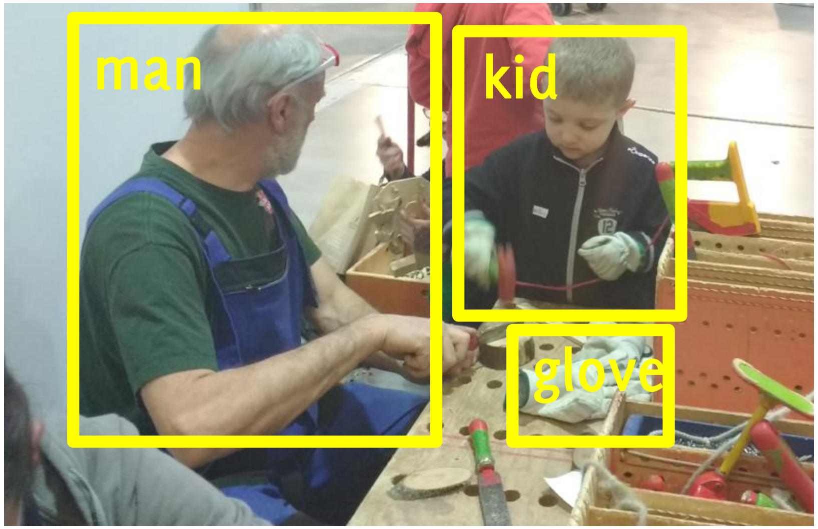
OBJECT DETECTION TASK

Given a fixed set of categories and an input image which contains an unknown and varying number of instances

Draw a bounding box on each object instance

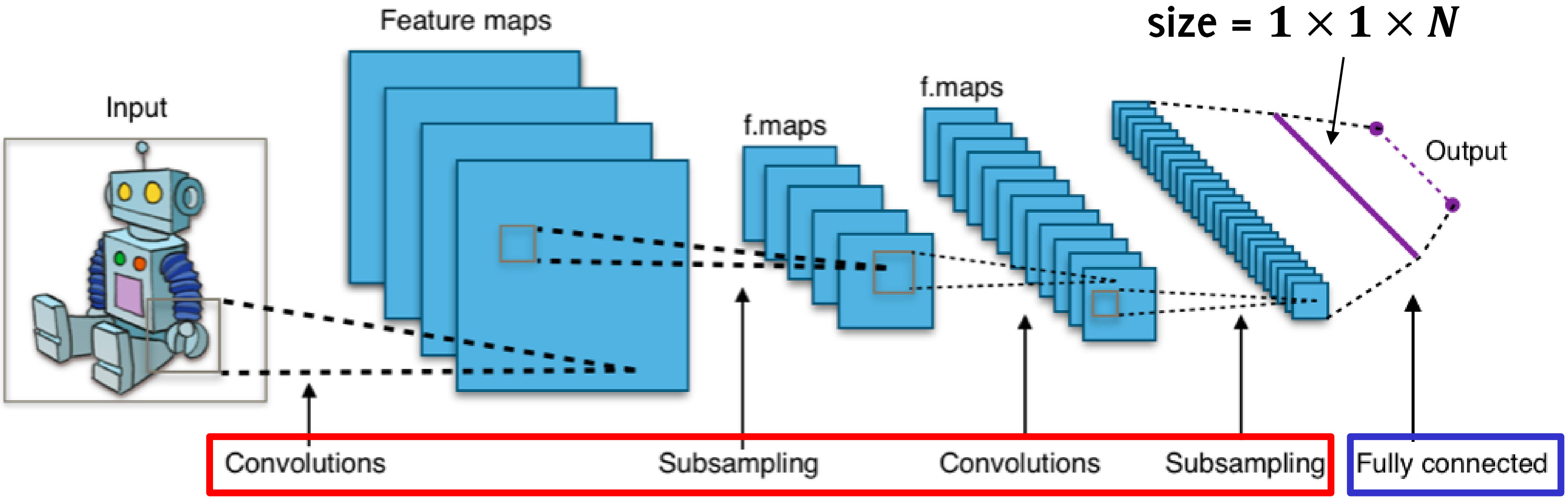
A training set of annotated images with labels and bounding boxes for each object is required

MAN: (x,y,h,w)
KID: (x,y,h,w)
GLOVE: (x,y,h,w)



CONVOLUTIONAL NEURAL NETWORKS (CNN)

The typical architecture of a convolutional neural network



CNNs are meant to process fixed-size input (e.g. 224 x 224 x 3).

The **convolutional and subsampling layers** operate in a sliding manner over image having arbitrary size

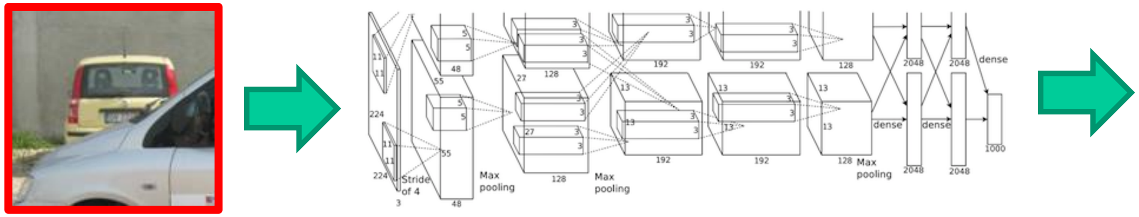
The **fully connected** layer constrains the input to a fixed size.

THE STRAIGHTFORWARD SOLUTION: SLIDING WINDOW

1000 x 2000 pixels

- Slide on the image a window of that size and classify each region.
- Assign the predicted label to the central pixel

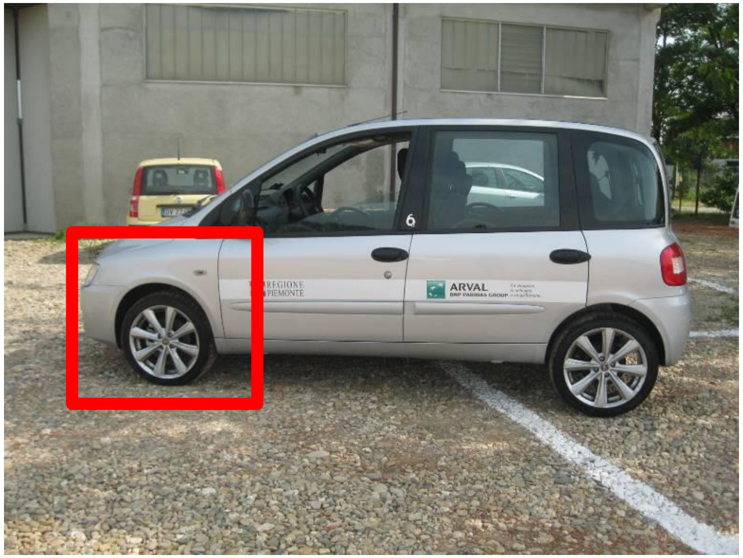
Adopt the whole machinery seen so far to each crop of the image



car

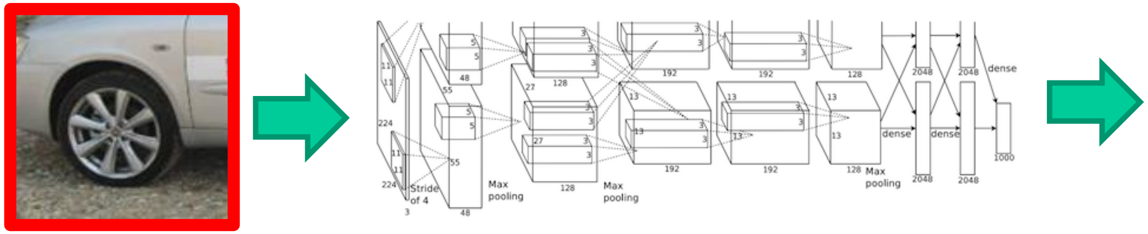
THE STRAIGHTFORWARD SOLUTION: SLIDING WINDOW

1000 x 2000 pixels



- A pretrained model is meant to process a fixed input size (e.g. 224 x 224 x 3)
- Slide on the image a window of that size and classify each region.
- Assign the predicted label to the central pixel

Adopt the whole machinery seen so far to each crop of the image



wheel

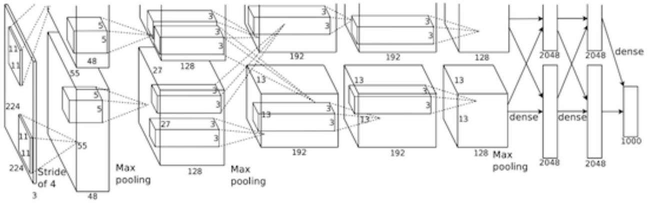
THE STRAIGHTFORWARD SOLUTION: SLIDING WINDOW

1000 x 2000 pixels

- A pretrained model is meant to process a fixed input size (e.g. 224 x 224 x 3)
- Slide on the image a window of that size and classify each region.
- Assign the predicted label to the central pixel

Adopt the whole machinery seen so far to each crop of the image

The background class has to be included!



background

MANY DRAWBACKS…

Cons:

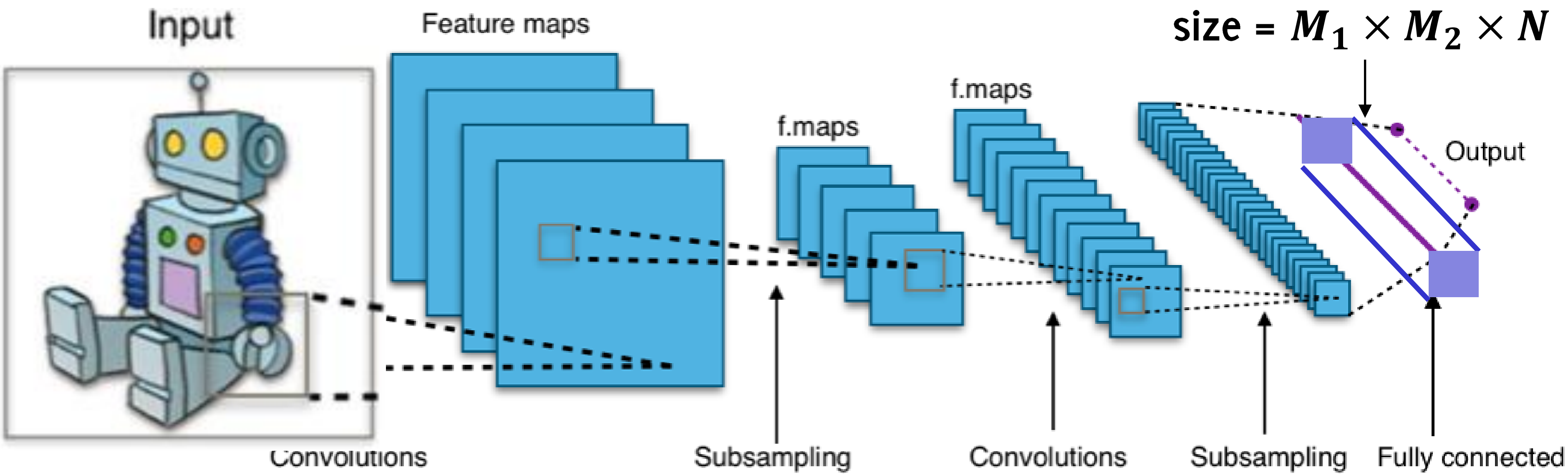
- **Very inefficient!** Does not re-use features that are «shared» among overlapping crops
- How to choose the crop size?
- Difficult to detect objects at different scales!
- A huge number of crops of different sizes should be considered….

Plus:

- The CNN is trained for the simpler image classification task

CONVOLUTIONAL NEURAL NETWORKS (CNN)

The typical architecture of a convolutional neural network

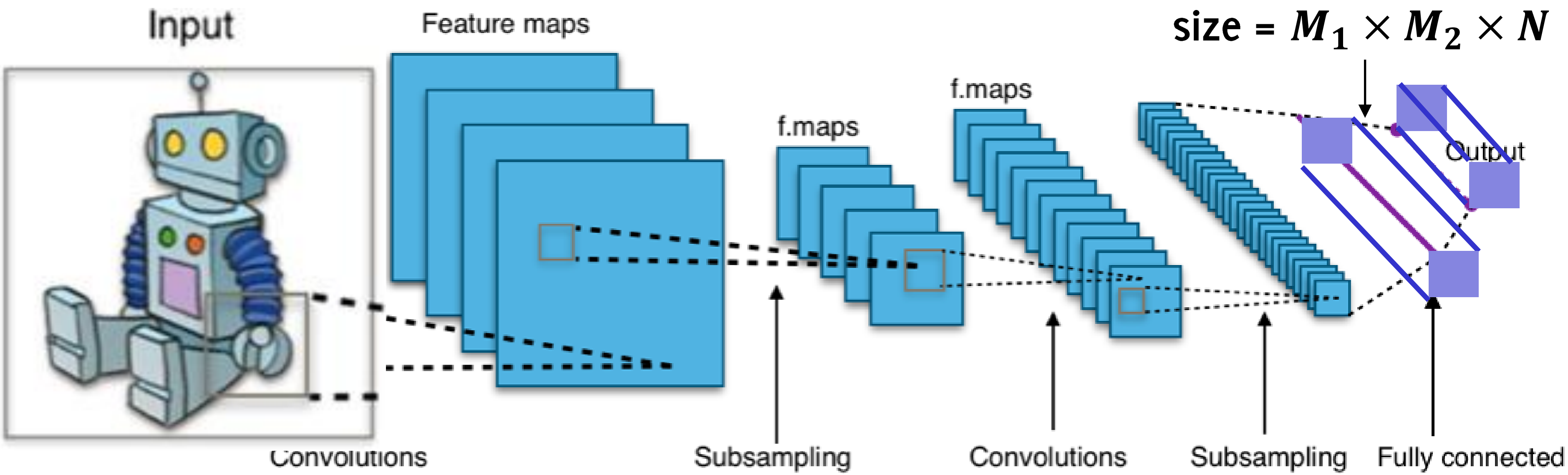


Applying the first CNN layers to larger images yield larger volumes through all the network until the input of the FC layer.

The FC network can not be used to compute class scores.

FULLY CONVOLUTIONAL NEURAL NETWORKS (F-CNN)

The typical architecture of a convolutional neural network



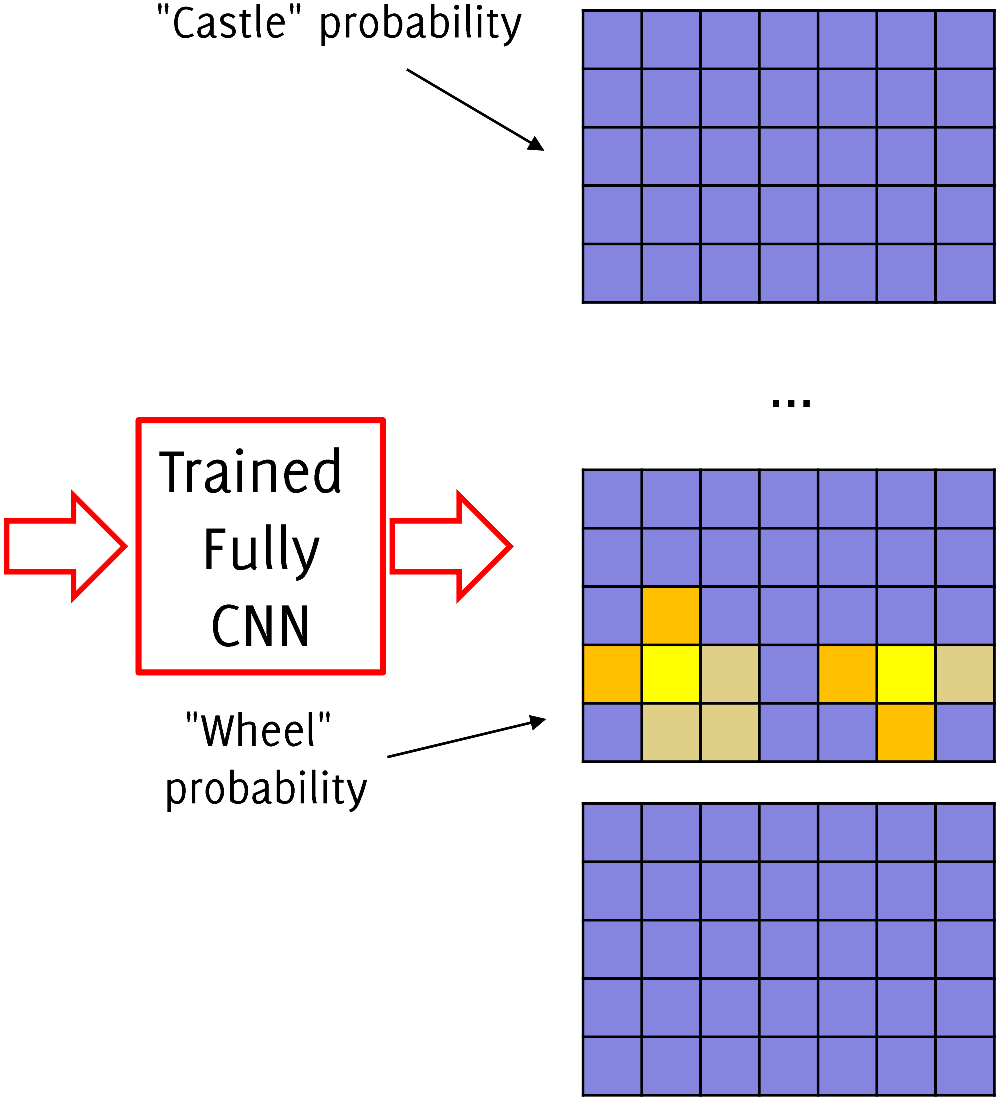
Since the **FC** is linear, it can be **represented as convolution** against L filters of size $1 \times 1 \times N$ (each one contains the FC weights)

Convolutional filters can be applied to volumes of **any size**, yielding images as outputs. The CNN becomes **fully convolutional**

Long, J., Shelhamer, E., Darrell, T. *"Fully convolutional networks for semantic segmentation"*. CVPR 2015

OUTPUT OF A F-CNN AS HEATMAPS

An larger image than those used for training the network



"Castle" probability

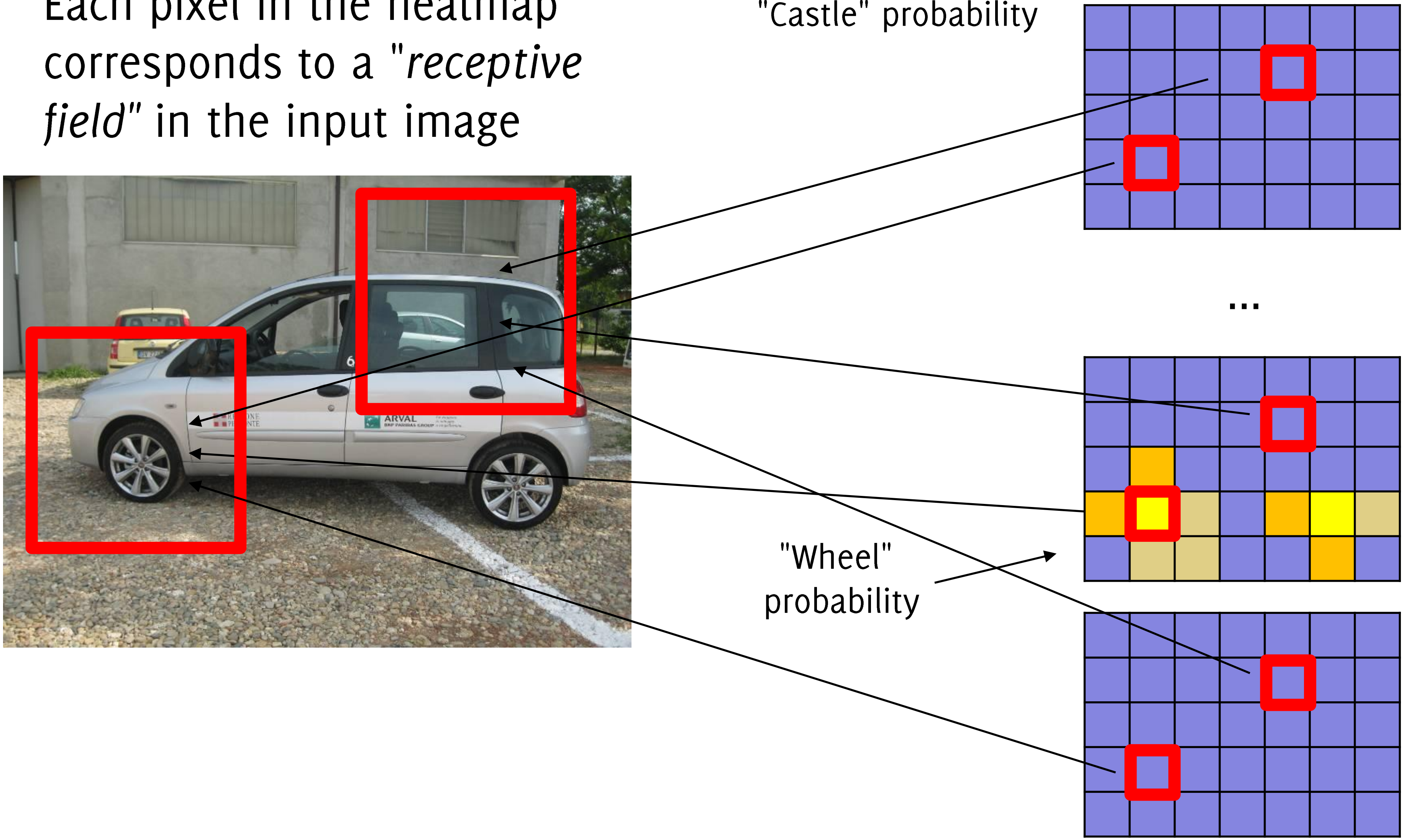
Trained Fully CNN

...

"Wheel" probability

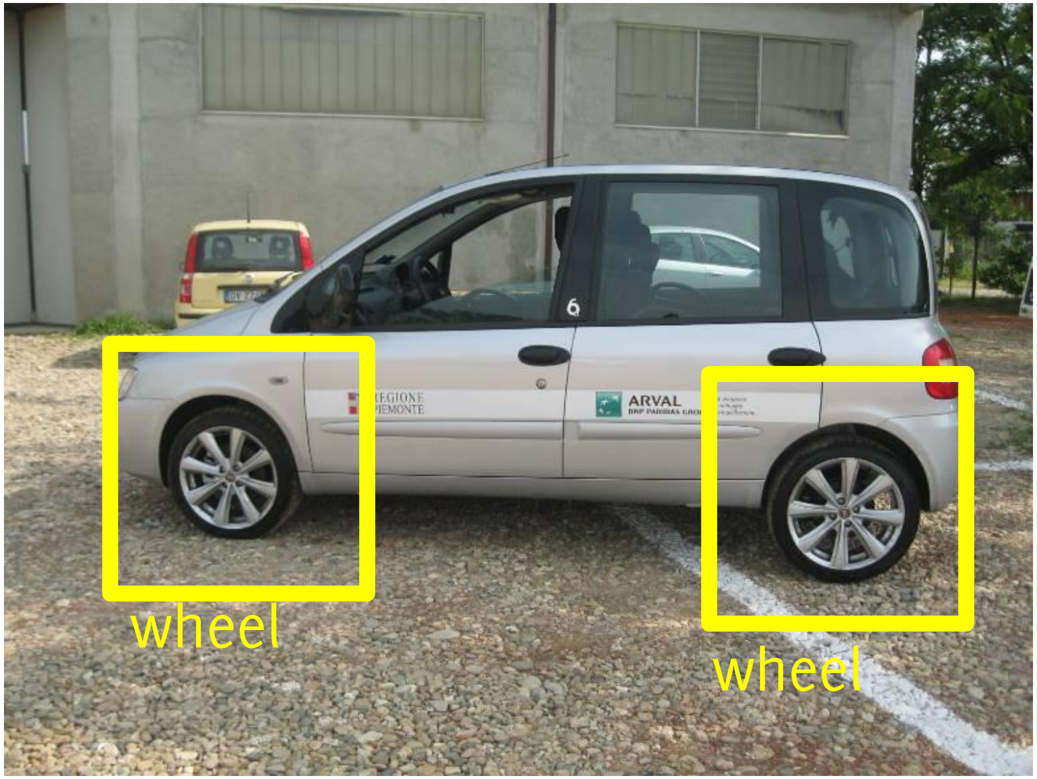
OUTPUT OF A F-CNN AS HEATMAPS

Each pixel in the heatmap corresponds to a "*receptive field*" in the input image



OBJECT DETECTION FROM THE HEATMAPS

Then apply some aggregation strategy on the heatmaps to perform object detection by a pre-trained CNN network.

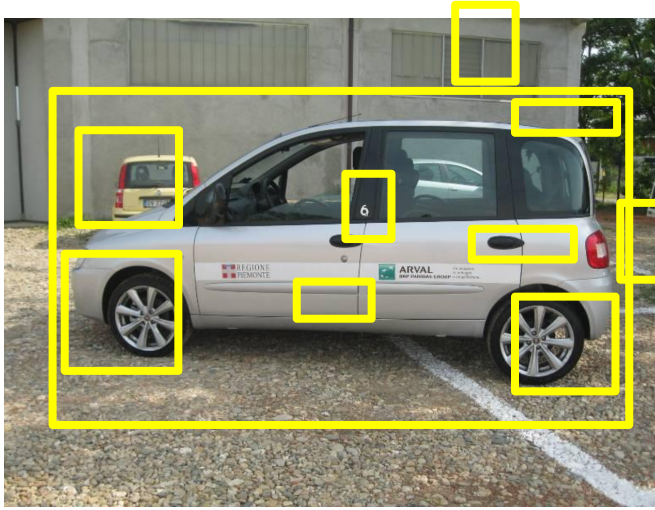


REGION PROPOSAL

Region proposal algorithms (and networks) are meant to select all the objects in an image and provide a bounding box around each of them

Algorithms with rather high recall (but low precision) were there before the deep learning advent

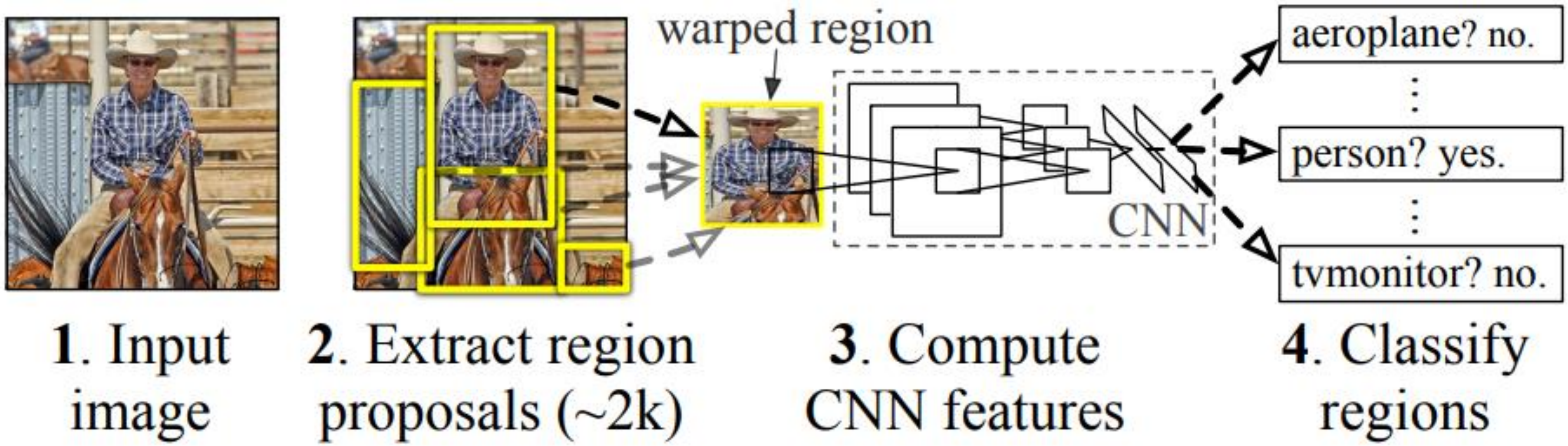
The idea is to apply first a region proposal algorithm and fed them to a classification network to the proposal regions



R-CNN

Object detection by means of region proposal (R stands for **regions**)

SVM +
BB regressor



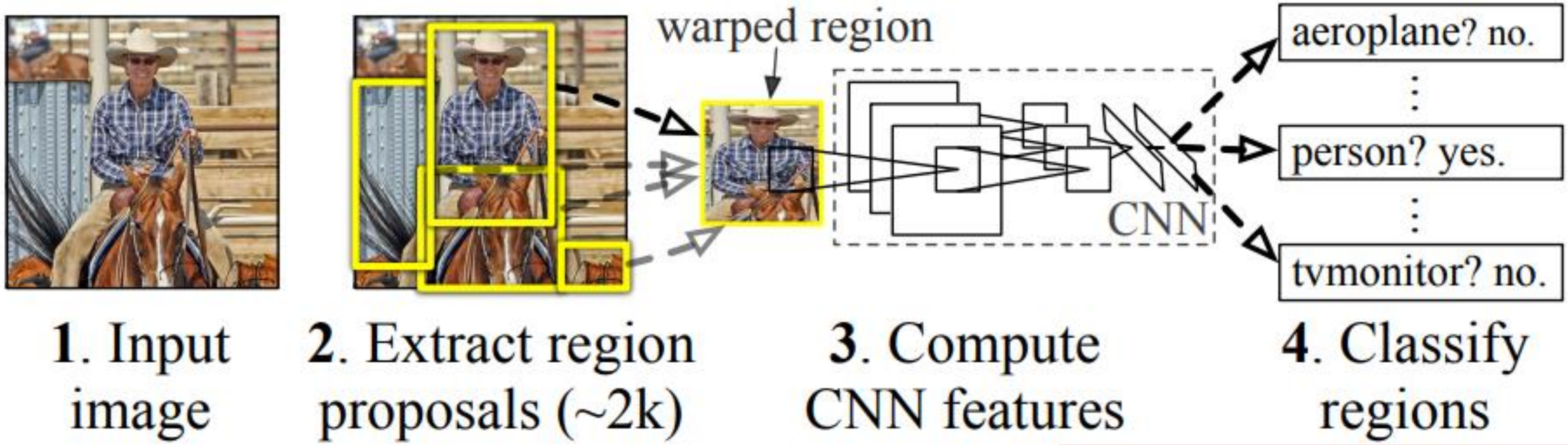
Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *CVPR 2014*

R-CNN

Object detection by means of region proposal

Warping is necessary when CNN has the FC layer

SVM + BB regressor



There is no learning in the region proposal algorithm

It is also possible to refine the region by training a regression network.
Region of interest can exceed image

R-CNN LIMITATIONS

Ad-hoc training objectives and not an end-to-end training

- Fine-tune network with softmax classifier (log loss)
- Train post-hoc linear SVMs (hinge loss)
- Train post-hoc bounding-box regressions (least squares)

Region proposals are from a different algorithm and that part has **not been optimized** for the detection by CNN

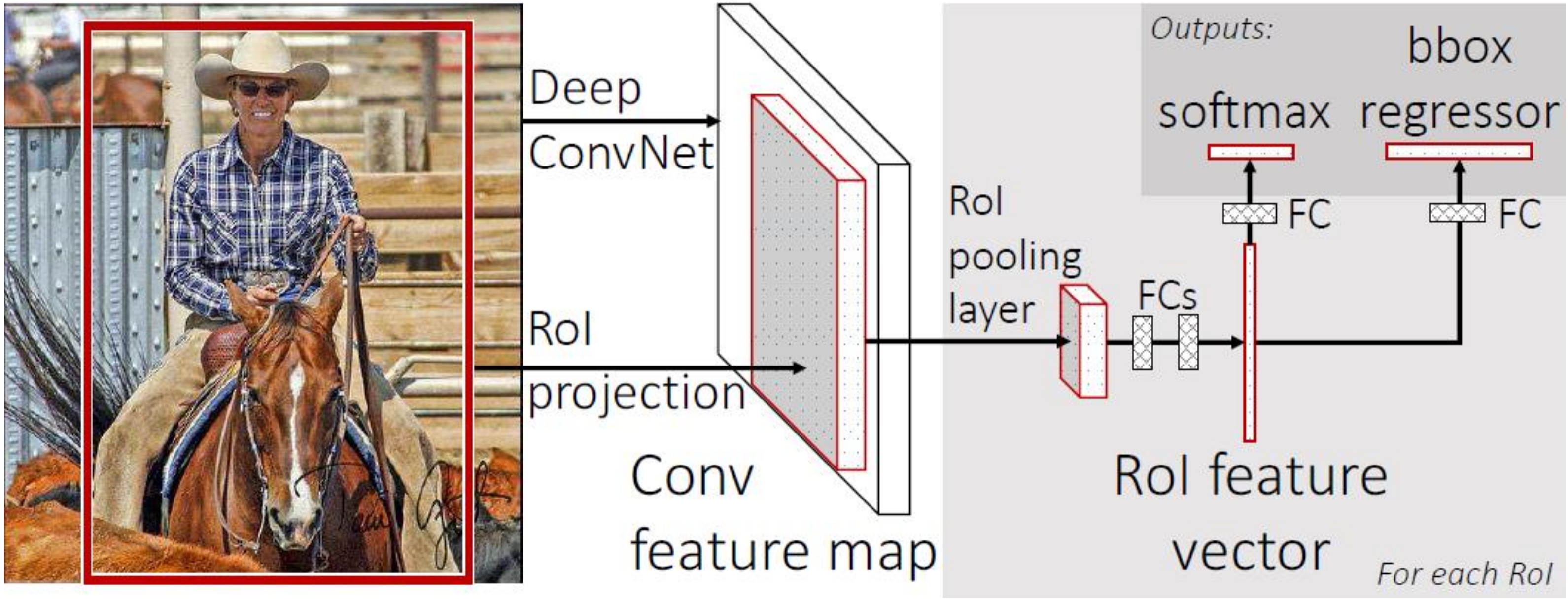
Training is slow (84h), takes a lot of disk space

Inference (detection) **is slow** since the CNN has to be executed on each region proposal (no feature re-use)

- 47s / image with VGG16

FAST R-CNN

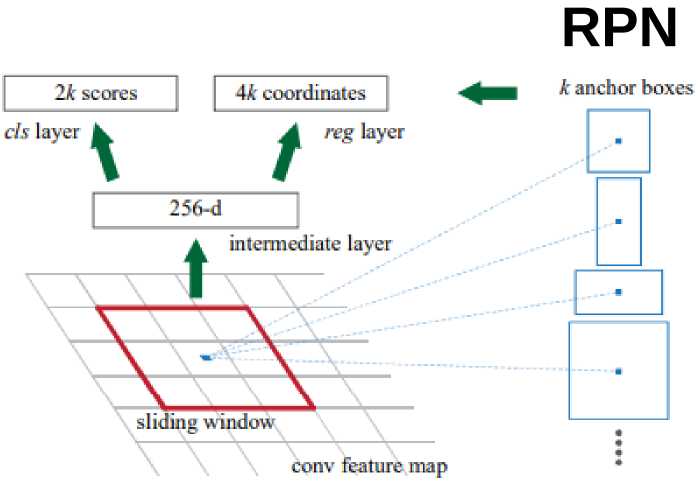
1. The whole image is fed to a CNN that extracts feature maps.
2. **Region proposals** are identified from the image and **projected into the feature maps** (re-use convolutional computation)



Girshick, Ross. "Fast r-cnn." ICCV 2015

FASTER R-CNN

- A region proposal network (RPN) is a Fully Convolutoinal NN (3x3 filter size) that replaces the ROI extraction algorithm.
- RPN operates on feature maps of the conv. layers of the Fast R-CNN
- Given the image, it provides a set of BB with their *objectness score*
- The network becomes much faster (0.2s test time per image)



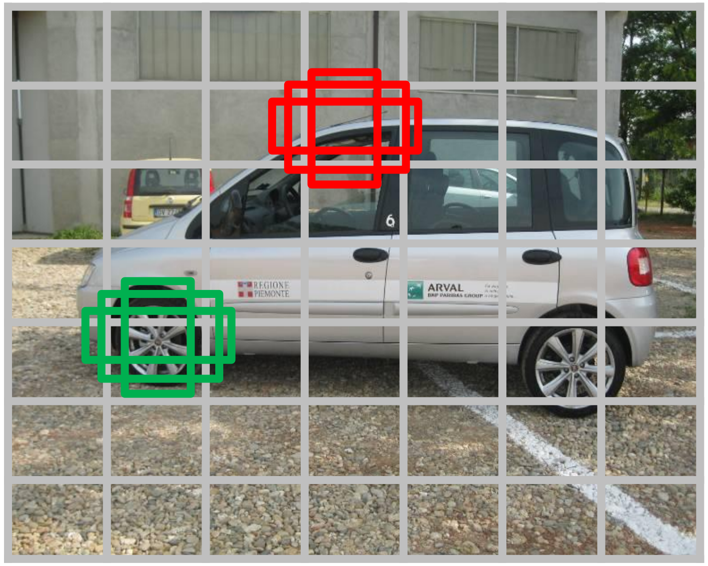
Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *NIPS 2015*

YOLO

1. divide the image in a coars grid (e.g. 7x7)
2. each grid cell there are B base-bounding boxes associated
3. For each cell and bounding box we want to predict:
   - The **bounding box offset**, to better match the object: (dx, dy, dh, dw, objectness_score)
   - The **class associated** to the bounding box over the C considered categories

So, the output of the network is

$$7 \times 7 \times B \times (5 + C)$$



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *CVPR* 2016.

FROM SUPERVISED TO SEMI-SUPERVISED

Training a CNN requires **a lot of labeled data**

In real world applications **anomalous data** are very difficult to collect

Semi-supervised approaches require only normal data and **are more appealing**

Semi-supervised approaches

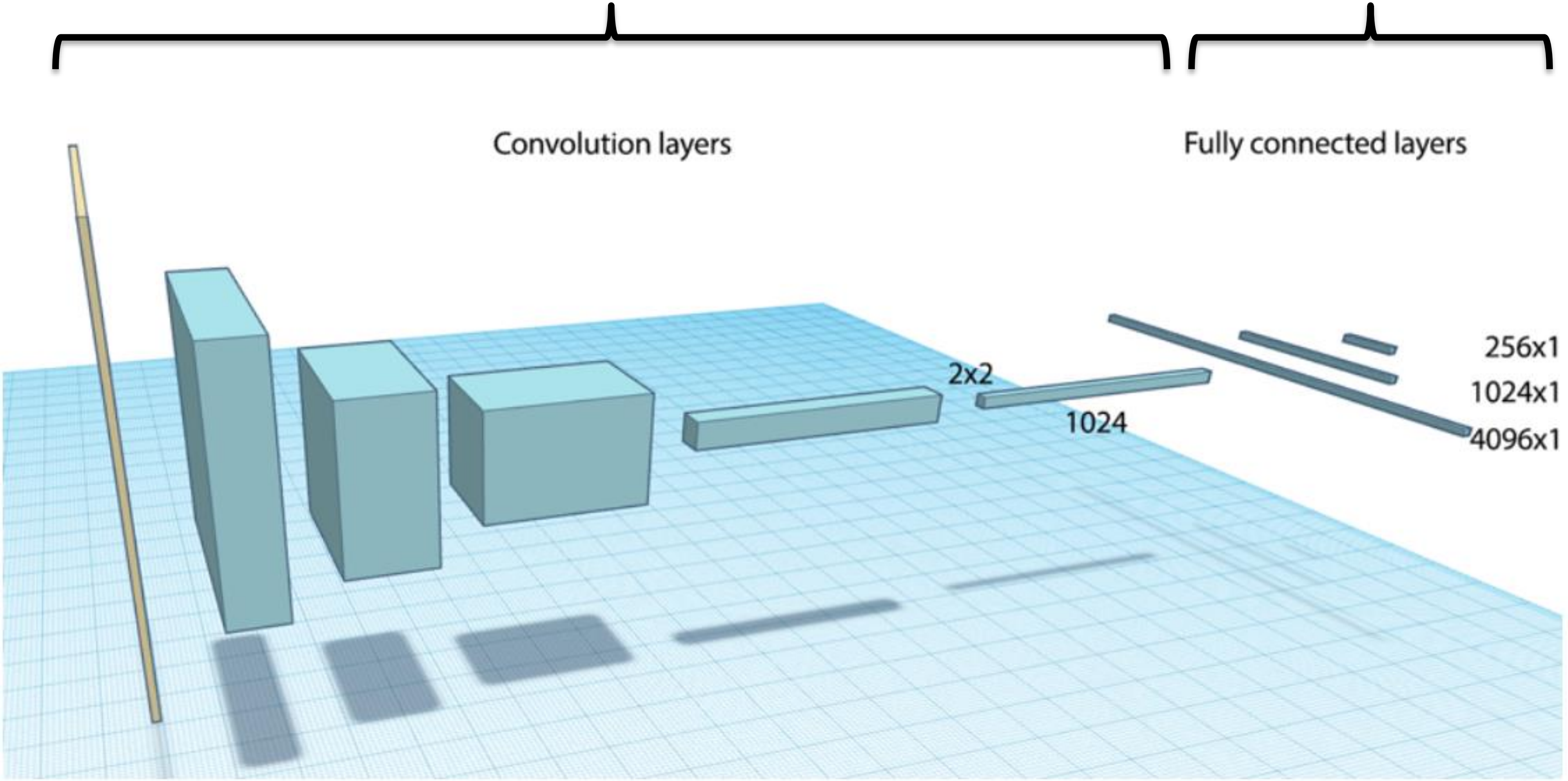
SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Autoencoders
  - Self-supervised learning
  - Domain-based
- Generative models

CNN AS DATA-DRIVEN FEATURE EXTRACTOR

Extract high-level features from pixel data

Classify

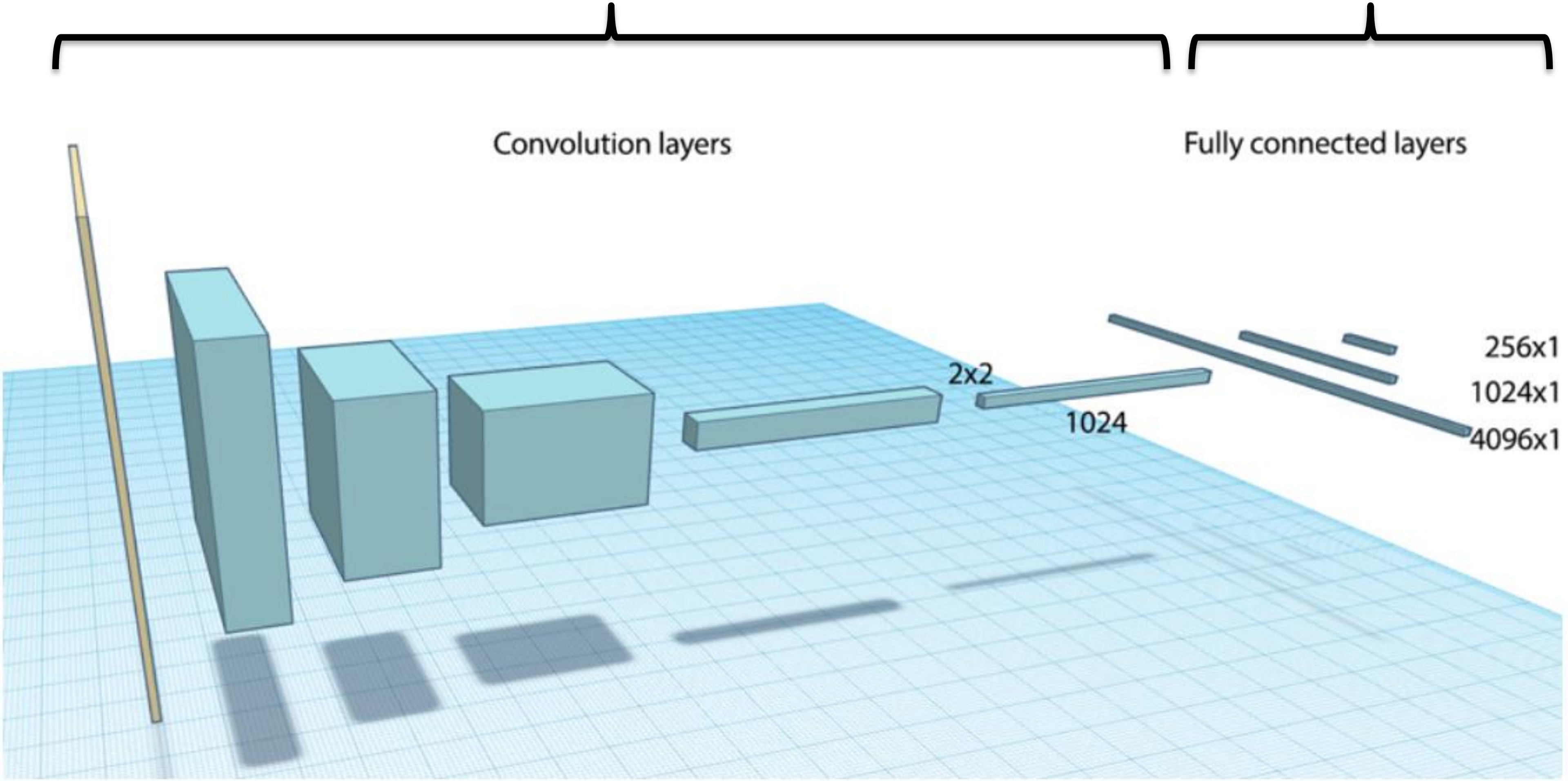


CNN AS DATA-DRIVEN FEATURE EXTRACTOR

The feature vector extracted from the last layer can be modeled as a random vector

Extract high-level features from pixel data

Classify



SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Autoencoders
  - Self-supervised learning
  - Domain-based
- Generative models

TRANSFER LEARNING

Idea:

- Use a pretrained network CNN (e.g. AlexNet), that was trained for a different task and on a different dataset
- Throw away the last layer(s)
- Use the CNN to build a new dataset TR' from TR:

$$TR' = \{\psi(\boldsymbol{s_i}),\ \boldsymbol{s_i} \in TR\}$$

- Train your favorite anomaly detector on TR'

TRANSFER LEARNING

- Features extracted from a CNN, i.e., $\psi(\boldsymbol{s})$ is typically very large for deep networks (e.g. ResNET). **Reduce data-dimensionality** by PCA defined on a set of normal features
- **Anomalies** can be **detected by measuring distance w.r.t. normal features**, possibly using clustering to speed up performance.
- Thresholds can be computed by the three-sigma rule or bootstrap.

Napoletano P., Piccoli F., Schettini R., *"Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity"*, Sensors 2018

TRANSFER LEARNING

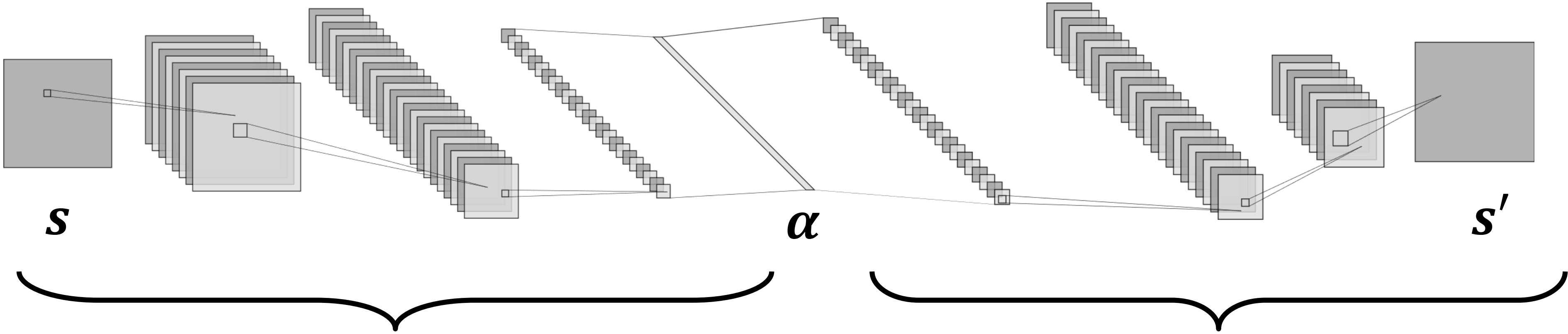
Pros: pretrained networks are very powerful models, since they usually trained on datasets with million of images

Cons: the network is **not trained on normal** data. Meaningful structures in normal images might not be successfully captured by network trained on images from a different domain (e.g. medical vs natural images)

SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Autoencoders (revisited)
  - Self-supervised learning
  - Domain-based
- Generative models

AUTOENCODERS (REVISITED)



$\boldsymbol{s}$ $\boldsymbol{\alpha}$ $\boldsymbol{s}'$

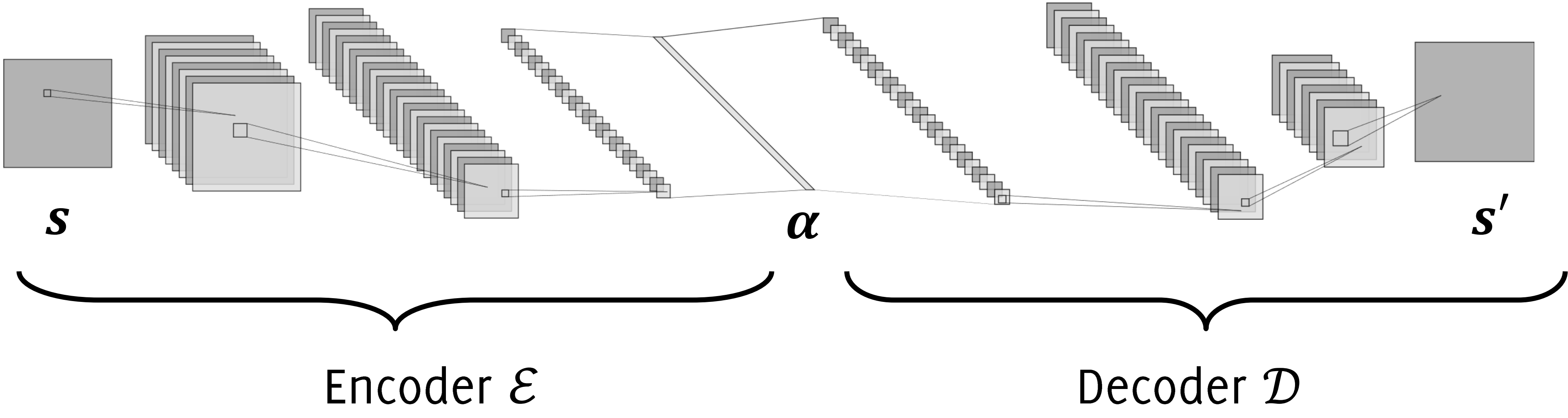
Encoder $\mathcal{E}$

Decoder $\mathcal{D}$

Autoencoders can be trained directly on normal data by minimizing the reconstruction loss:

$$\sum_{\boldsymbol{s} \in TR} \left\| \boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big) \right\|_2$$

AUTOENCODERS (REVISITED)



Encoder $\mathcal{E}$

Decoder $\mathcal{D}$

We can fit a **density model** (e.g. Gaussian Mixture) on $\boldsymbol{\alpha} = \mathcal{E}(\boldsymbol{s})$:

$$\boldsymbol{\alpha} \sim \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i} ,$$

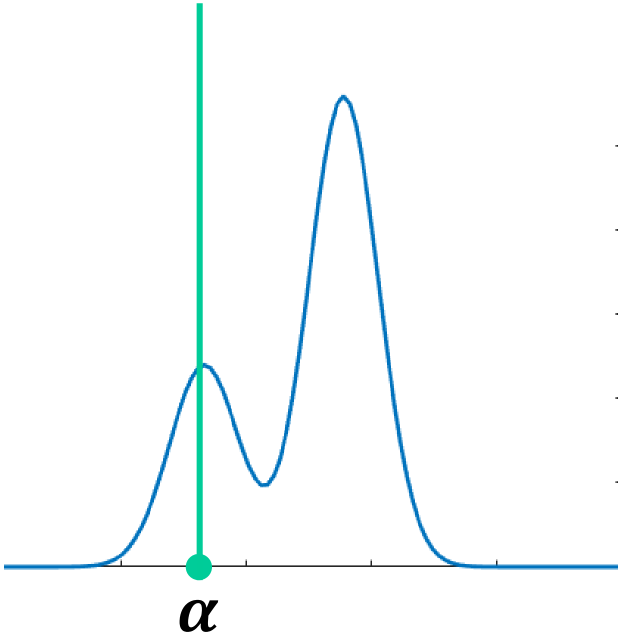
Where $\varphi_{\boldsymbol{\mu}_i, \Sigma_i}$ is the pdf of $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{i}}, \Sigma_i\ )$

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i,\Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k,\Sigma_k}(\boldsymbol{\alpha}_n)}$$



Bishop, “Pattern recognition and machine learning”

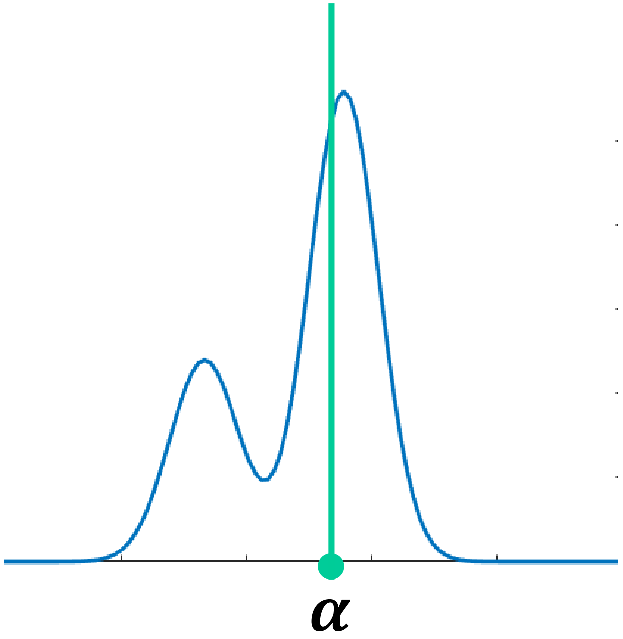
$$\gamma_1 \sim 1$$
$$\gamma_2 \sim 0$$

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i,\Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k,\Sigma_k}(\boldsymbol{\alpha}_n)}$$



$$\gamma_1 \sim 0$$
$$\gamma_2 \sim 1$$

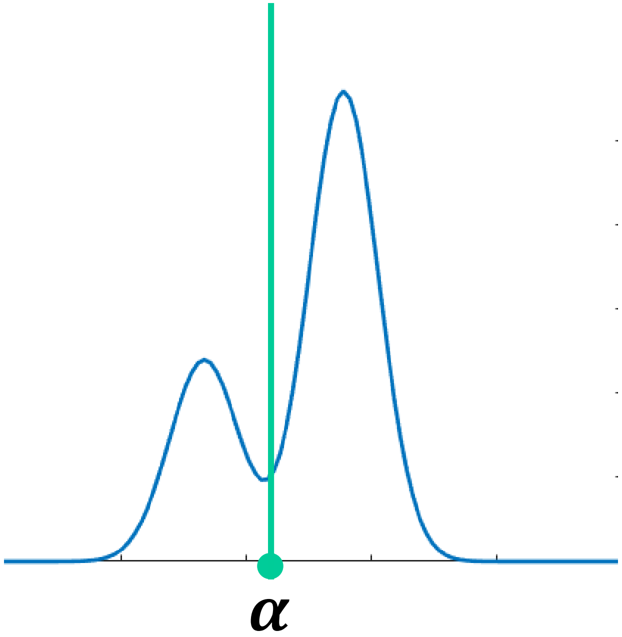
Bishop, “Pattern recognition and machine learning”

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i,\Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k,\Sigma_k}(\boldsymbol{\alpha}_n)}$$



$$\gamma_1 \sim \frac{1}{2}$$

$$\gamma_2 \sim \frac{1}{2}$$

Bishop, “Pattern recognition and machine learning”

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i,\Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k,\Sigma_k}(\boldsymbol{\alpha}_n)}$$

- **M-step:** update the parameters of the Gaussian Mixture

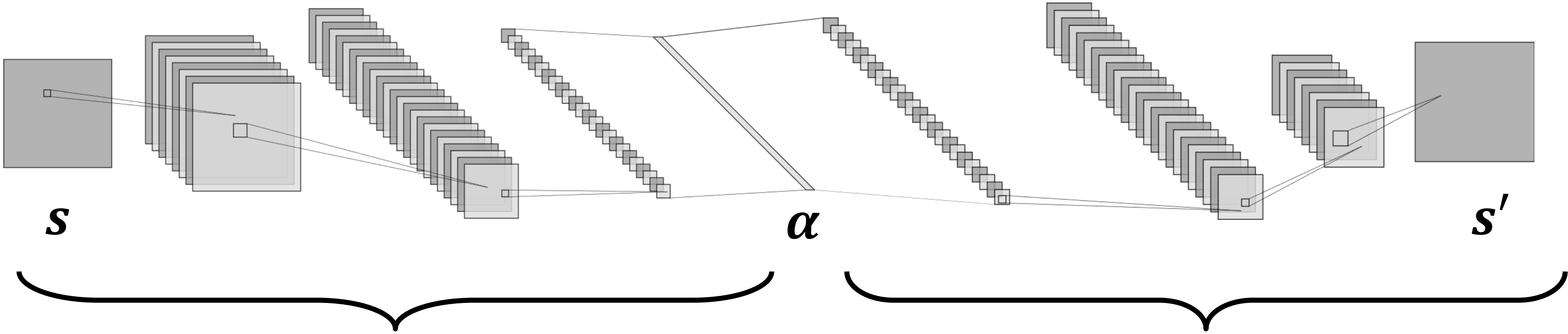
$$\pi_i = \frac{1}{N}\sum_n \gamma_{n,i}$$

$$\mu_i = \frac{\sum_n \gamma_{n,i}\boldsymbol{\alpha}_n}{\Sigma_n \gamma_{n,i}}$$

$$\Sigma_i = \frac{\sum_n \gamma_{n,i}(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)^T}{\Sigma_n \gamma_{n,i}}$$

Bishop, “Pattern recognition and machine learning”

AUTOENCODERS (REVISITED)

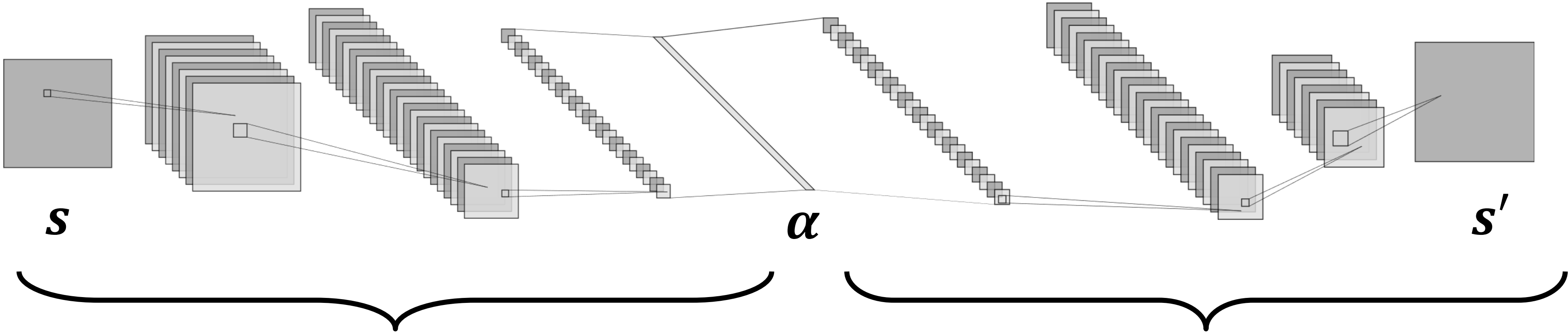


Encoder $\mathcal{E}$ Decoder $\mathcal{D}$

We can compute the likelihood of a test sample $\boldsymbol{s}$ as:

$$\mathcal{L}(\boldsymbol{s}) = \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\mathcal{E}(\boldsymbol{s})),$$

AUTOENCODERS (REVISITED)



Encoder $\mathcal{E}$ Decoder $\mathcal{D}$

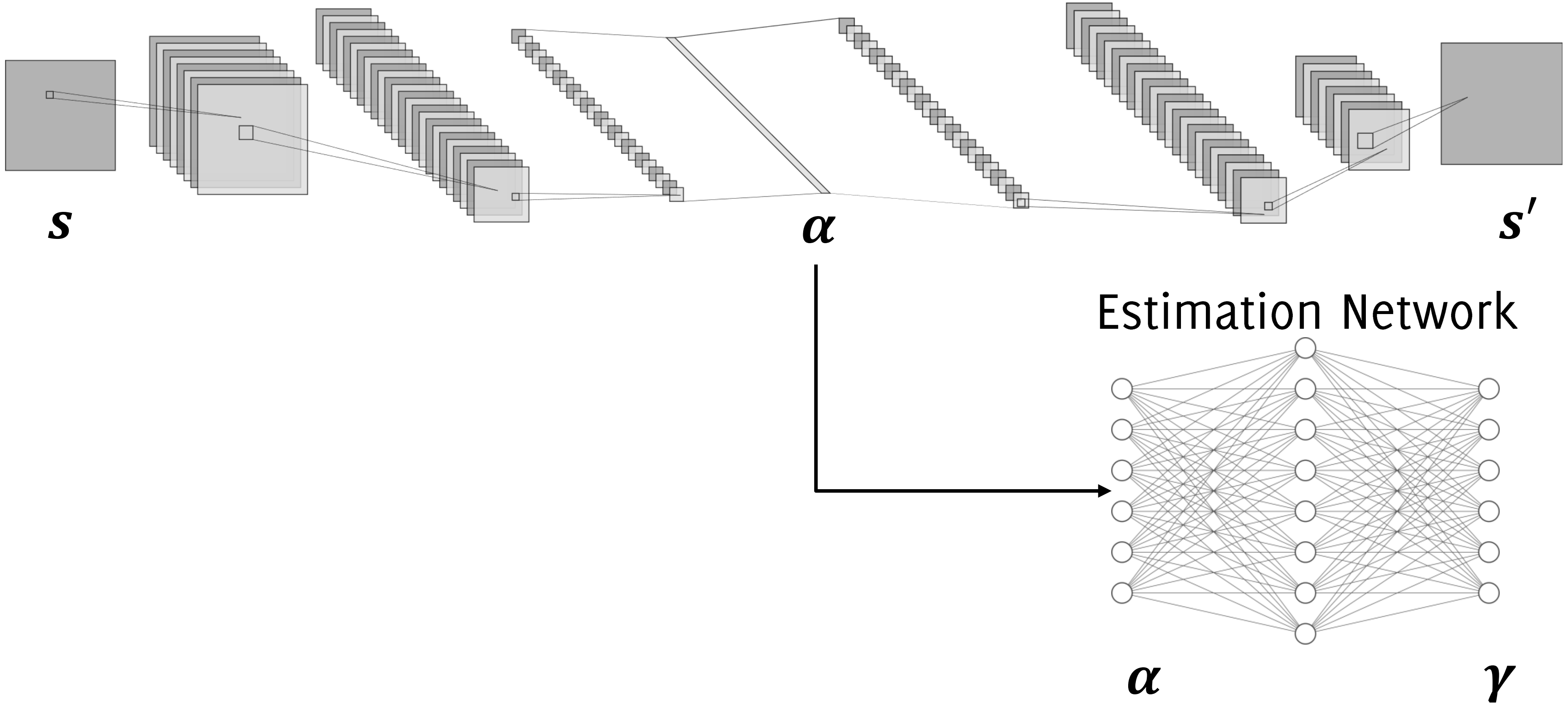
We can compute the likelihood of a test sample $\boldsymbol{s}$ as:

$$\mathcal{L}(\boldsymbol{s}) = \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i,\Sigma_i}(\mathcal{E}(\boldsymbol{s})),$$

The autoencoder and the Gaussian Mixture are not jointly learned!

JOINT LEARNING OF AUTOENCODER AND DENSITY MODEL

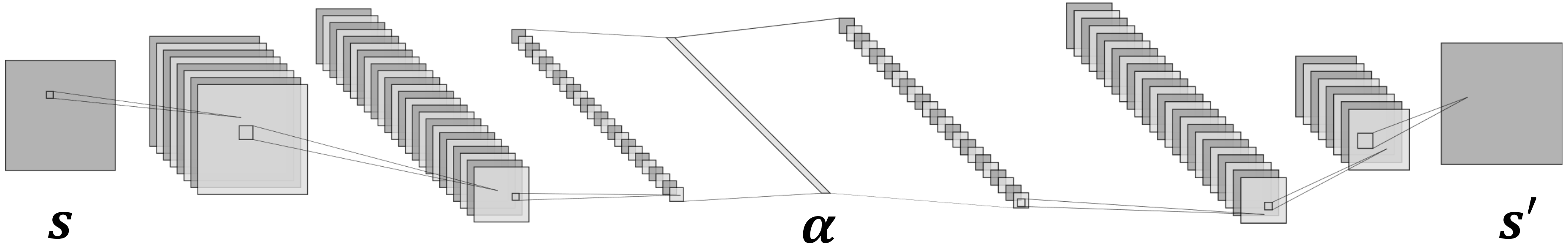
Idea: given a training set of N samples use a NN to predict the membership weights of each sample

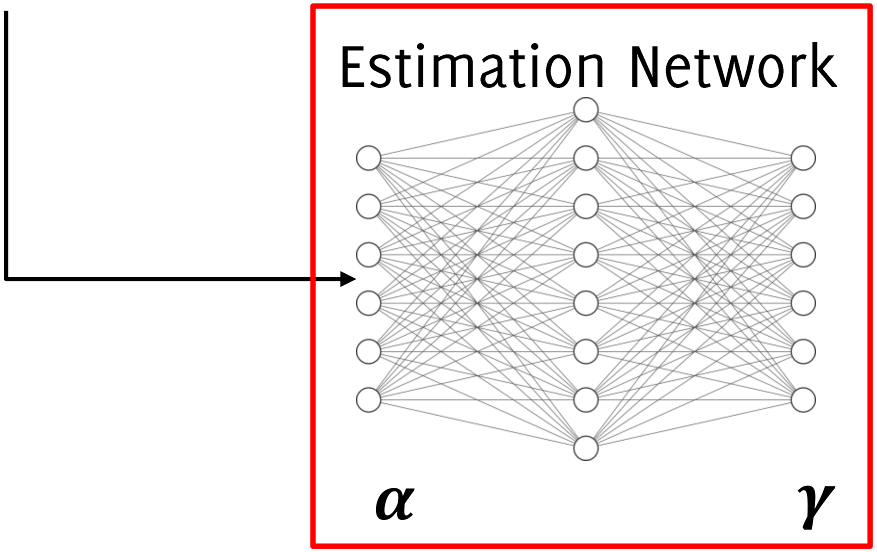


Zong et al, *"Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection"*, *ICLR 2018*

JOINT LEARNING OF AUTOENCODER AND DENSITY MODEL

Idea: given a training set of N samples use a NN to predict the membership weights of each sample





E-step

Estimate the GM parameters as:

$$\pi_i = \frac{1}{N}\sum_n \gamma_{n,i}$$

$$\mu_i = \frac{\sum_n \gamma_{n,i}\boldsymbol{\alpha}_n}{\sum_n \gamma_{n,i}}$$

$$\Sigma_i = \frac{\sum_n \gamma_{n,i}(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)^T}{\sum_n \gamma_{n,i}}$$

M-step

Zong et al, *"Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection"*, *ICLR 2018*

DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL

Minimize the loss:

$$\min \sum_{\boldsymbol{s}} \left\|\boldsymbol{s} - \mathcal{D}\left(\mathcal{E}(\boldsymbol{s})\right)\right\|_2^2 + \lambda \mathcal{R}(\mathcal{E}(\boldsymbol{s}))$$

Where

$$\mathcal{R}(\boldsymbol{\alpha}) = -\log \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha})$$

Additional regularizations has to be imposed on Σ_i to avoid trivial solution

$\mathcal{R}(\mathcal{E}(\boldsymbol{s}))$ can be used an anomaly score for a sample $\boldsymbol{s}$

Zong et al, *"Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection", ICLR 2018*

SOME REMARKS

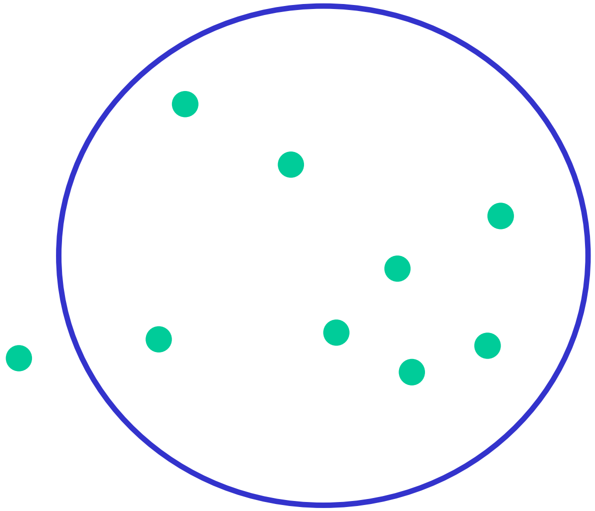
- The **estimation network** introduces a **regularization** that helps to avoid local optima of the recontruction error
- The autoencoder is then able to extract meaningful feature from normal data
- Density estimation enables anomaly detection, but it is a **more complicated** task

SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
    - Transfer learning
    - Autoencoders
    - Domain-based
    - Self-supervised learning
- Generative models

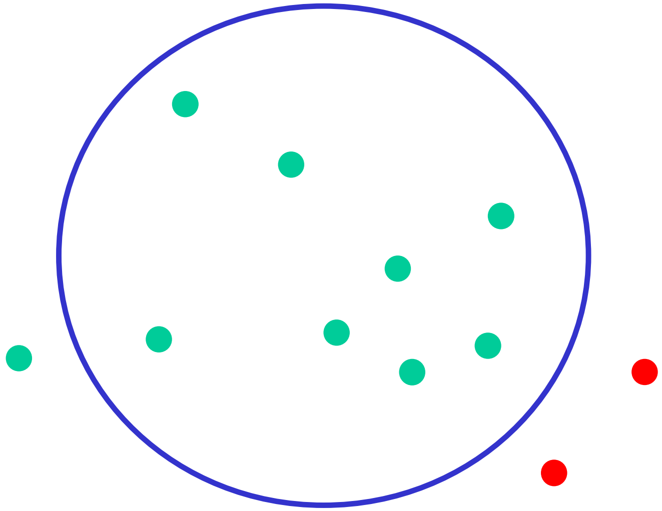
SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

We want to find an **hypersphere** that, in the feature space, **encloses most of the normal** data



SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

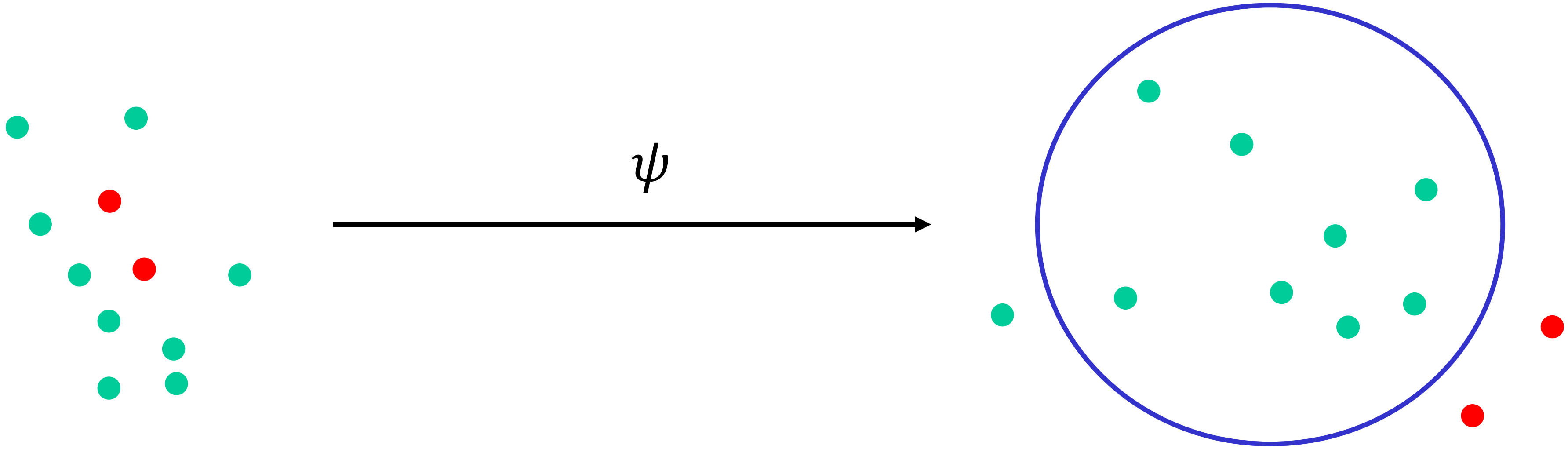
We want to find an **hypersphere** that, in the feature space, **encloses most of the normal** data



We expect that anomalous data lie outside the sphere

SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

Typically the sphere is computed in a **high** (possibly infinite) **dimensional** feature **space**

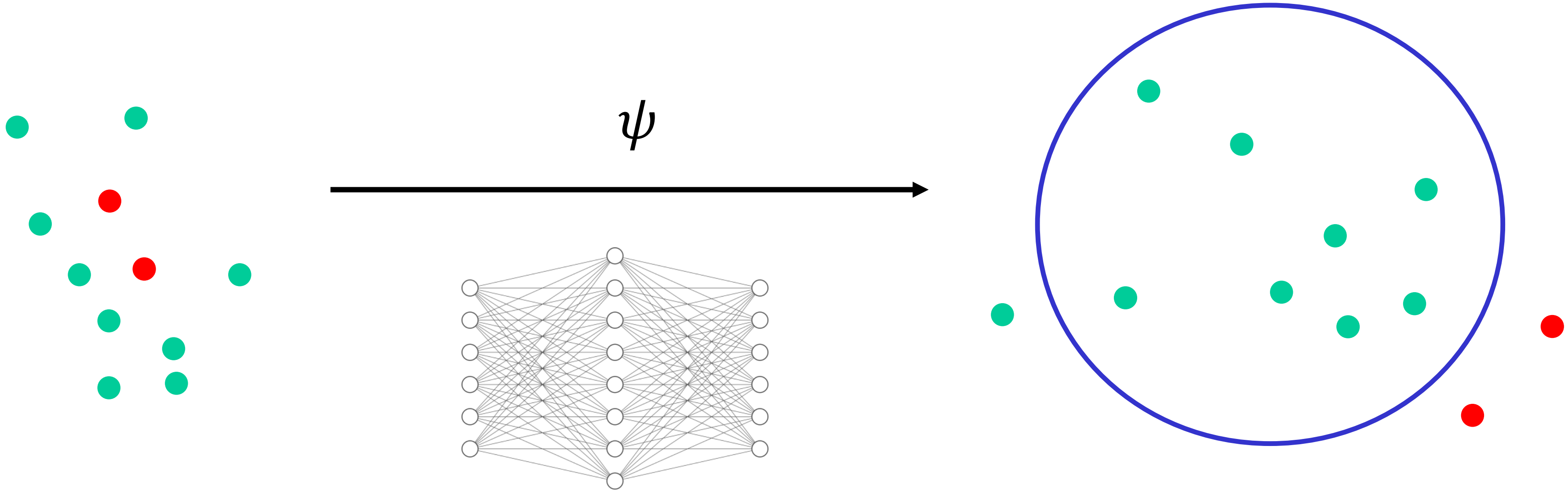


Feature are defined using **kernels**

- Polinomial kernel
- Gaussian kernel

SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

Idea: can we learn the feature from normal data using a neural network?



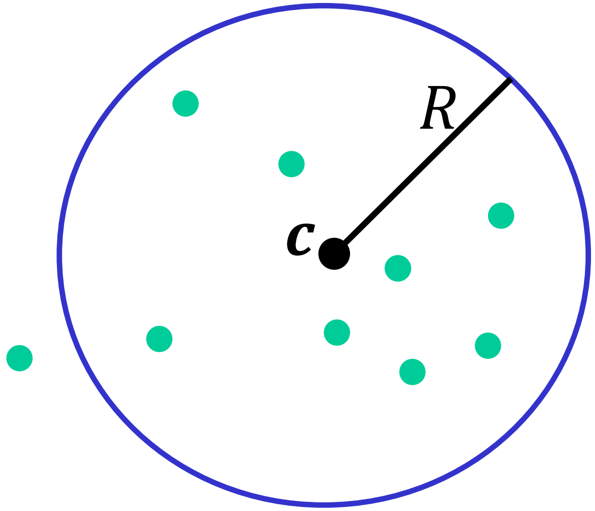
Ruff et al, *"Deep One-Class Classification"*, ICML 2018

SOFT-BOUNDARY DEEP SVDD

Minimize the loss:

$$\min_{R,\boldsymbol{\theta}} R^2 + \frac{1}{\nu N}\sum_{n=1}^{N} \max\{0, ||\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n) - \boldsymbol{c}||^{\mathbf{2}} - R^2\} + \lambda||\boldsymbol{\theta}||^2$$

- The samples $\boldsymbol{s}_n$ such that $\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n)$ is inside the sphere do not contribute to the loss
- ν provides a bound on the False Positive Rate
- $\lambda||\boldsymbol{\theta}||^2$ is a regularization term



A test sample $\boldsymbol{s}$ is anomalous if $\psi_{\boldsymbol{\theta}}(\boldsymbol{s}) - \boldsymbol{c} > R$

Ruff et al, *“Deep One-Class Classification”, ICML 2018*

SOFT-BOUNDARY DEEP SVDD

Minimize the loss:

$$\min_{R,\boldsymbol{\theta}} R^2 + \frac{1}{\nu N}\sum_{n=1}^{N} \max\{0, \left|\left|\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n) - \boldsymbol{c}\right|\right|^{\mathbf{2}} - R^2\} + \lambda\left|\left|\boldsymbol{\theta}\right|\right|^2$$

Remarks:

- Some contraints must be imposed on the network $\psi_{\boldsymbol{\theta}}$ to avoid trivial solutions:
  - **No bias terms**
  - **Unbounded** activations
- $\boldsymbol{c}$ is not optimized but has to be precomputed from data
  - $\boldsymbol{c}$ must be different from $\boldsymbol{c}_0 = \psi_0(\boldsymbol{s})$

Ruff et al, *"Deep One-Class Classification"*, *ICML 2018*

A SIMPLER FORMULATION: DEEP SVDD

$$\min_{\boldsymbol{\theta}} + \frac{1}{N}\sum_{n=1}^{N} ||\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n) - \boldsymbol{c}||^2 + \lambda ||\boldsymbol{\theta}||^2$$

Cons:

- No bound on the FPR provided by ν
- A threshold has to be chosen for the anomaly score:

$$||\psi_{\boldsymbol{\theta}}(\boldsymbol{s}) - \boldsymbol{c}||^2$$

Ruff et al, *“Deep One-Class Classification”, ICML 2018*

SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Autoencoders
  - Domain-based
  - Self-supervised learning
- Generative models

SELF-SUPERVISED LEARNING

We can **build** a **labeled dataset** for multiclass **classification** from normal data

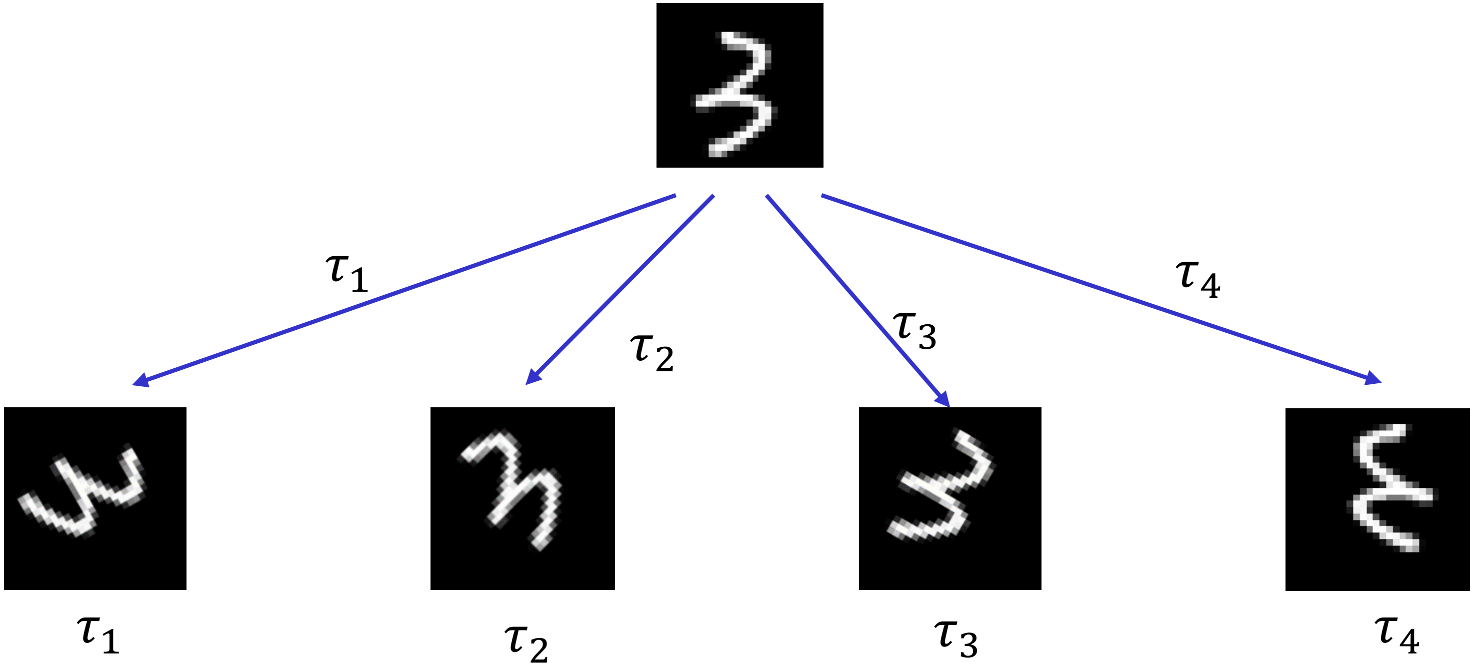
- Consider a set of T transformation $\mathcal{T} = \{\tau_1, \dots, \tau_T\}$
- Apply each transformation τ_i to every $\mathbf{s} \in TR$:

$$TR_{new} = \{(\tau_i(\boldsymbol{s}), i) \mid \boldsymbol{s} \in TR, i = 1, \dots, T\}$$

- Train a CNN on TR_{new}
- The output of the last layer of the CNN is used as feature vector

Example:

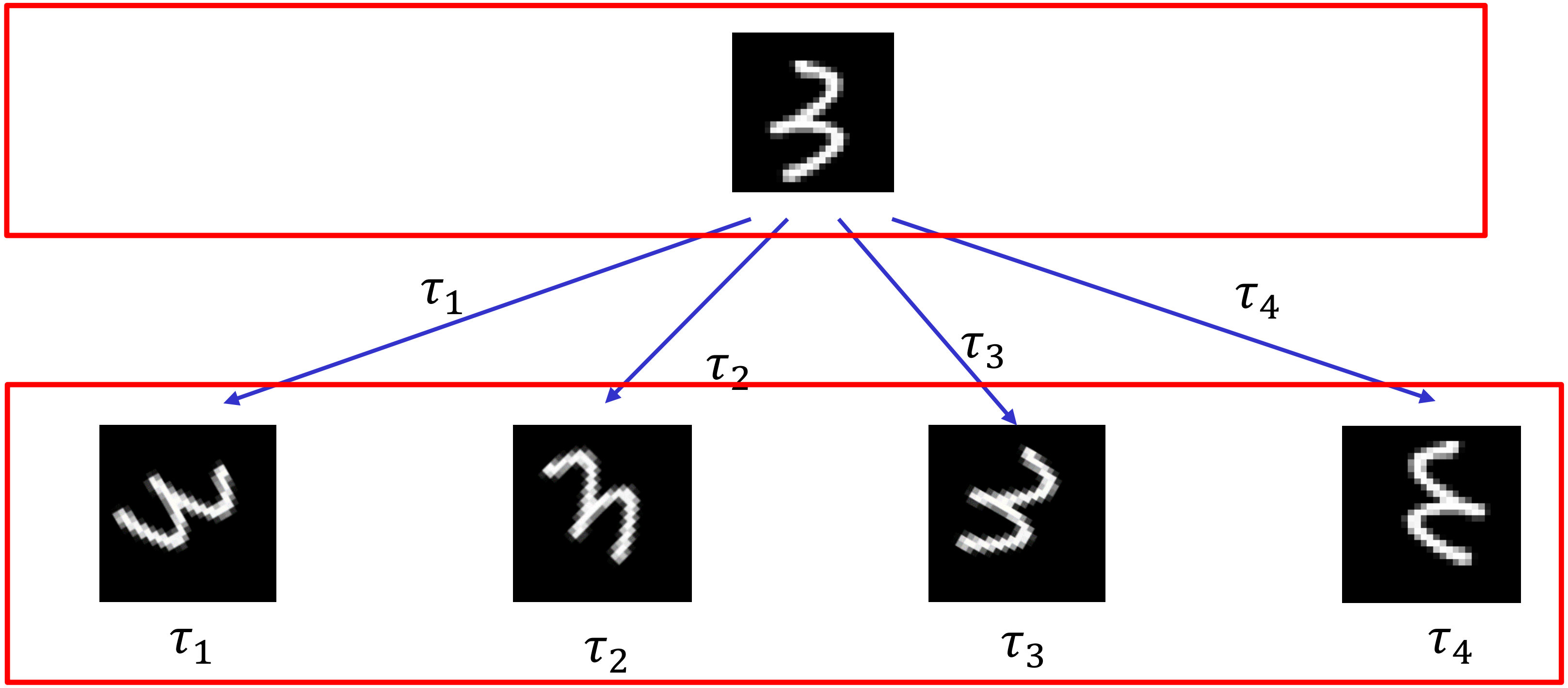
- TR contains only images representing digit 3
- $\mathcal{T}$ contains rotations and horizontal/vertical flips



SELF-SUPERVISED LEARNING

Example:

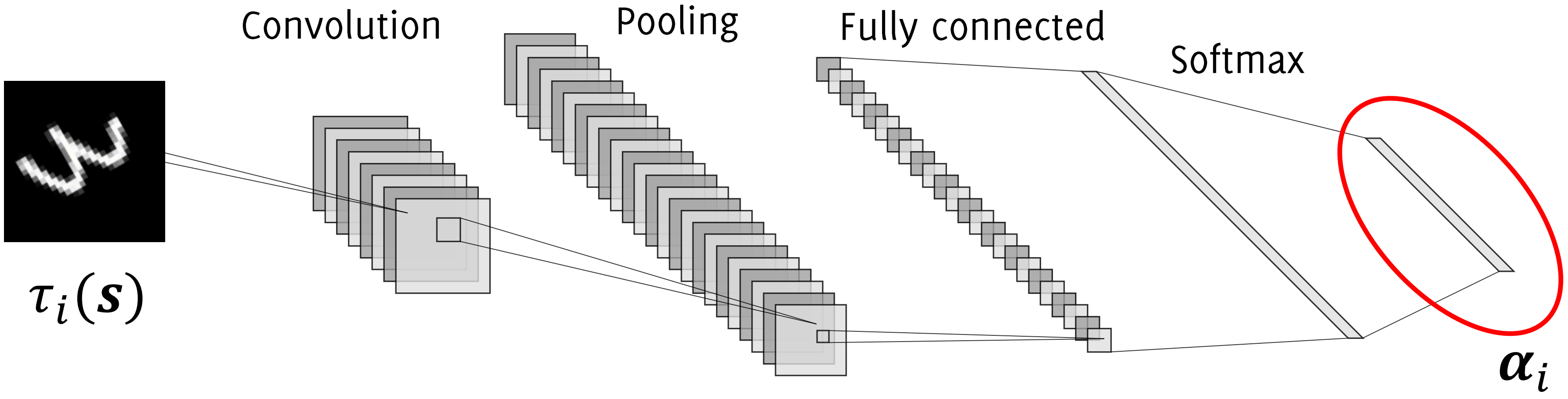
- TR contains only images representing digit 3
- $\mathcal{T}$ contains rotations and horizontal/vertical flips



Training set of normal data

Labeled training set with T classes

SELF-SUPERVISED LEARNING



Use the output of the last layer as **feature vector:**

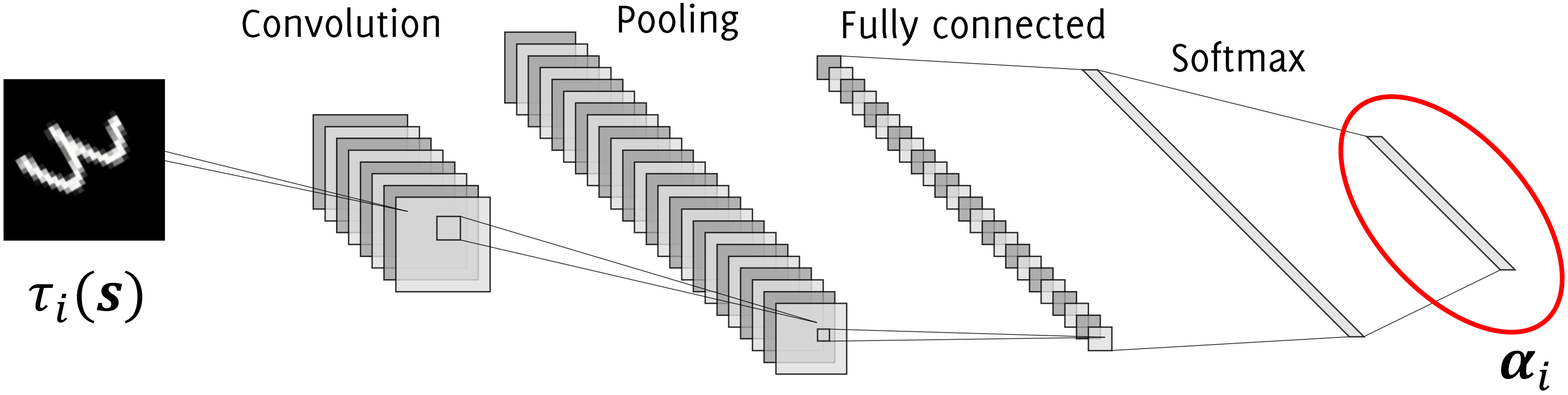
$$\boldsymbol{\alpha}_i = \psi(\boldsymbol{\tau}_i(\boldsymbol{s})) \in [0,1]^T$$

Estimate the conditional distributions $P(\boldsymbol{\alpha}_i|\tau_i)$ for each τ_i

Parametric distributions such as Dirichlet distribution can be used

Golan, El-Yaniv, *“Deep Anomaly Detection Using Geometric Transformations”*, NeurIPS 2018

SELF-SUPERVISED LEARNING



Compute the anomaly score as

$$score(\boldsymbol{s}) = -\sum_{i} \log P(\boldsymbol{\alpha}_i | \tau_i)$$

Golan, El-Yaniv, *“Deep Anomaly Detection Using Geometric Transformations”*, NeurIPS 2018

SELF-SUPERVISED LEARNING

The set of transformation has to be properly chosen:

- if during training the trained **classifier cannot discriminate** the transformed samples, it **does not extract meaningful feature** for anomaly detection
- **Non-geometric transformations** (Gaussian blur, gamma correction, sharpening) might eliminate important feature and **are less performing** than geometric ones

Golan, El-Yaniv, *"Deep Anomaly Detection Using Geometric Transformations"*, NeurIPS 2018

SEMI-SUPERVISED APPROACHES

- CNN as data-driven feature extractor
  - Transfer learning
  - Autoencoders
  - Domain-based
  - Self-supervised learning
- Generative models

GENERATIVE MODELS

Goal:

generative models generate, given a training set of images (data) S, other images (data) that are similar to those in S

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN approach:

Do not look for an **explicit density model** ϕ_S describing the manifold of natural images.

Just find out a **model** able to **generate samples** that looks like training samples $S \subset \mathbb{R}^n$

Instead of sampling from ϕ_S, just use:

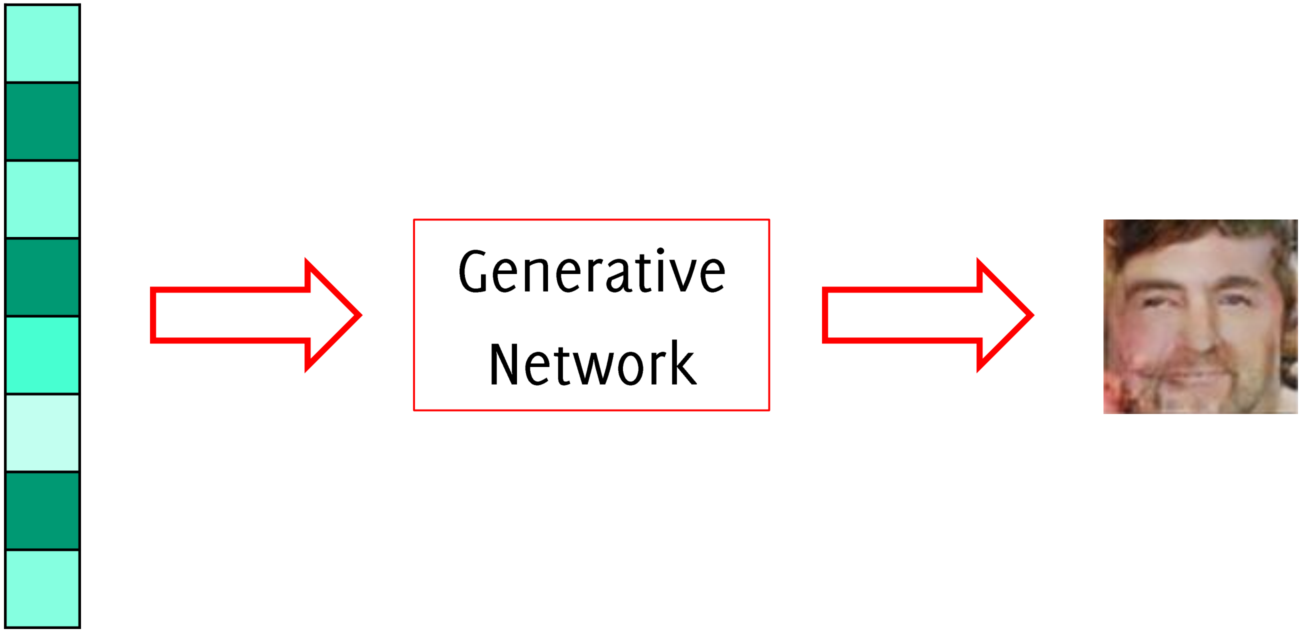
- Sample a seed from a known distribution ϕ_z
- Feed this seed to a learned transformation that generates realistic samples, as if they were drawn from ϕ_S

Use a **neural network to learn this transformation**

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN approach:



Generative Network

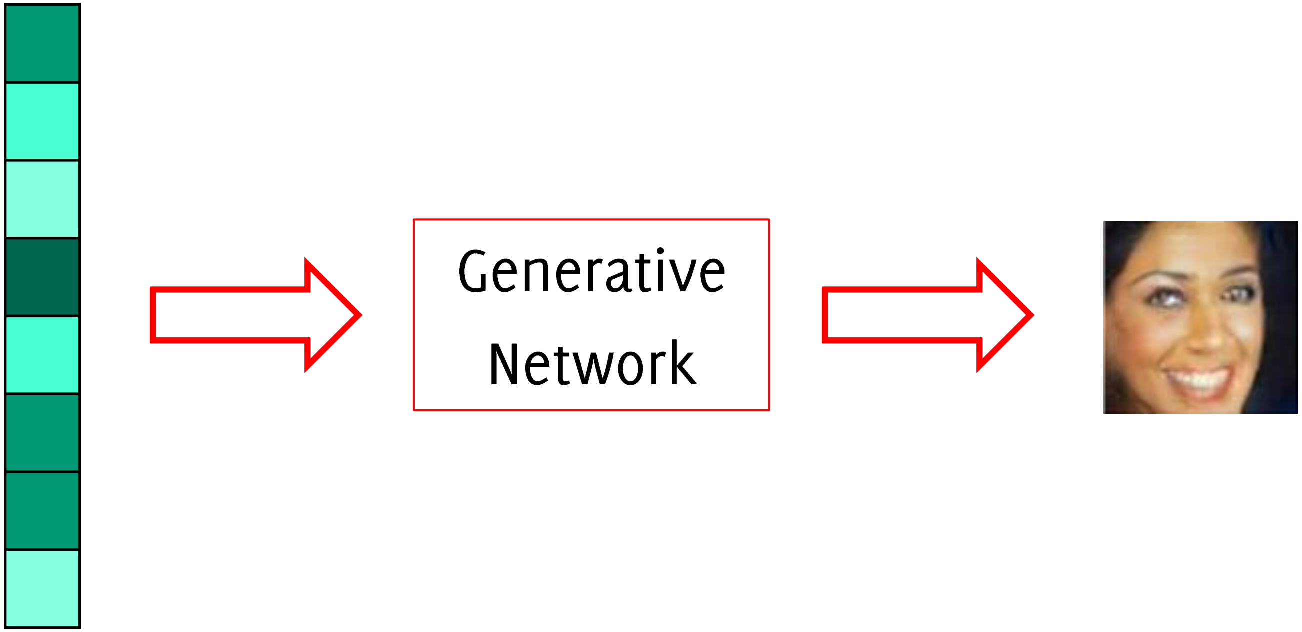
$z \sim \phi_z$

Draw a sample from the noise distribution

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN approach:



Generative Network

$z \sim \phi_z$

Draw a sample from the noise distribution

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN solution: Train a pair of neural networks with different tasks that compete in a sort of **two player game**.

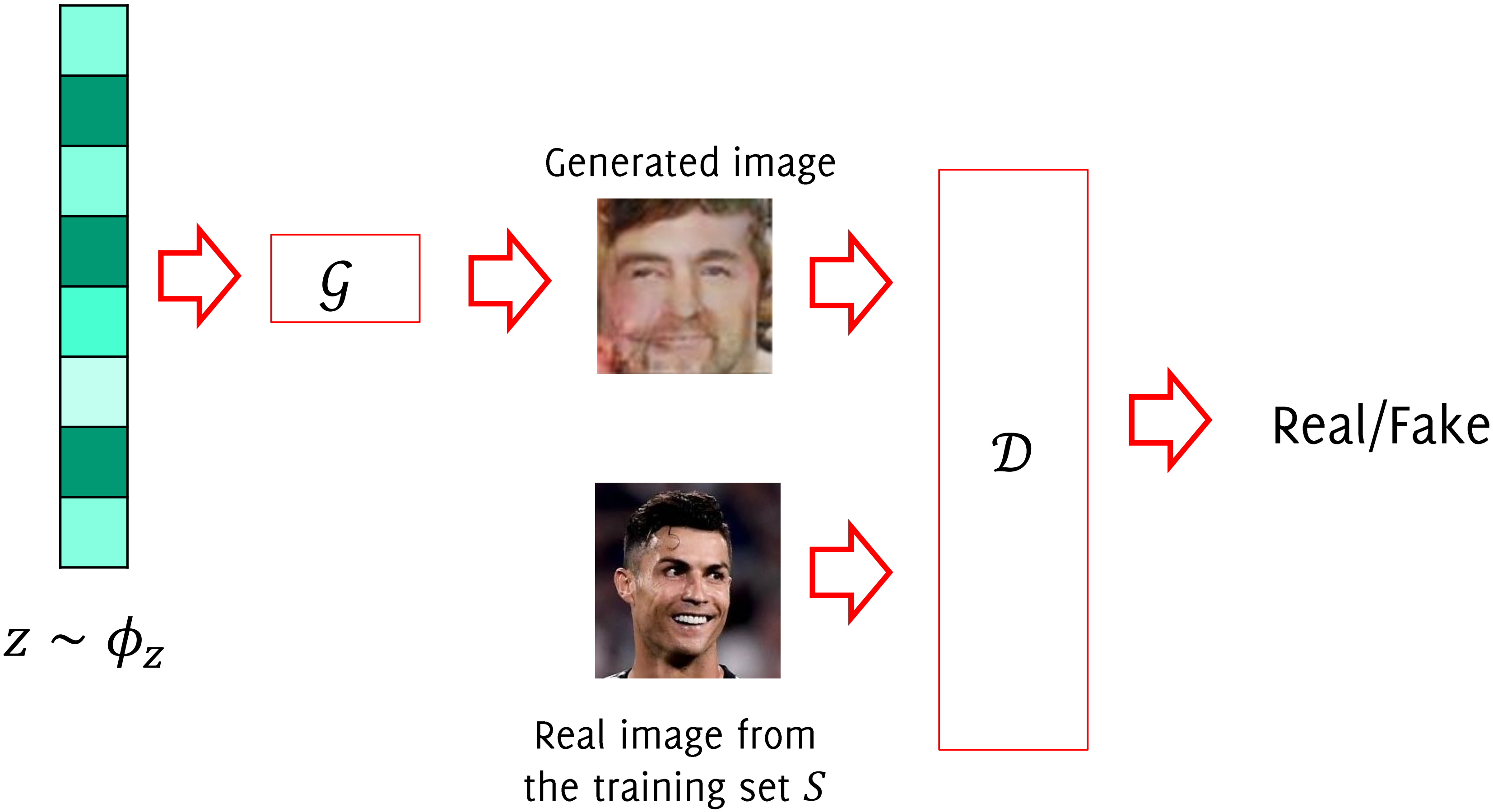
These models are:

- Generator $\mathcal{G}$ that produces realistic samples e.g. taking as input some random noise. $\mathcal{G}$ tries to fool the discriminator
- Discriminator $\mathcal{D}$ that takes as input an image and assess whether it is real or generated by $\mathcal{G}$

Train the two and at the end, keep only $\mathcal{G}$

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GAN ARCHITECTURE



$z \sim \phi_z$

$\mathcal{G}$

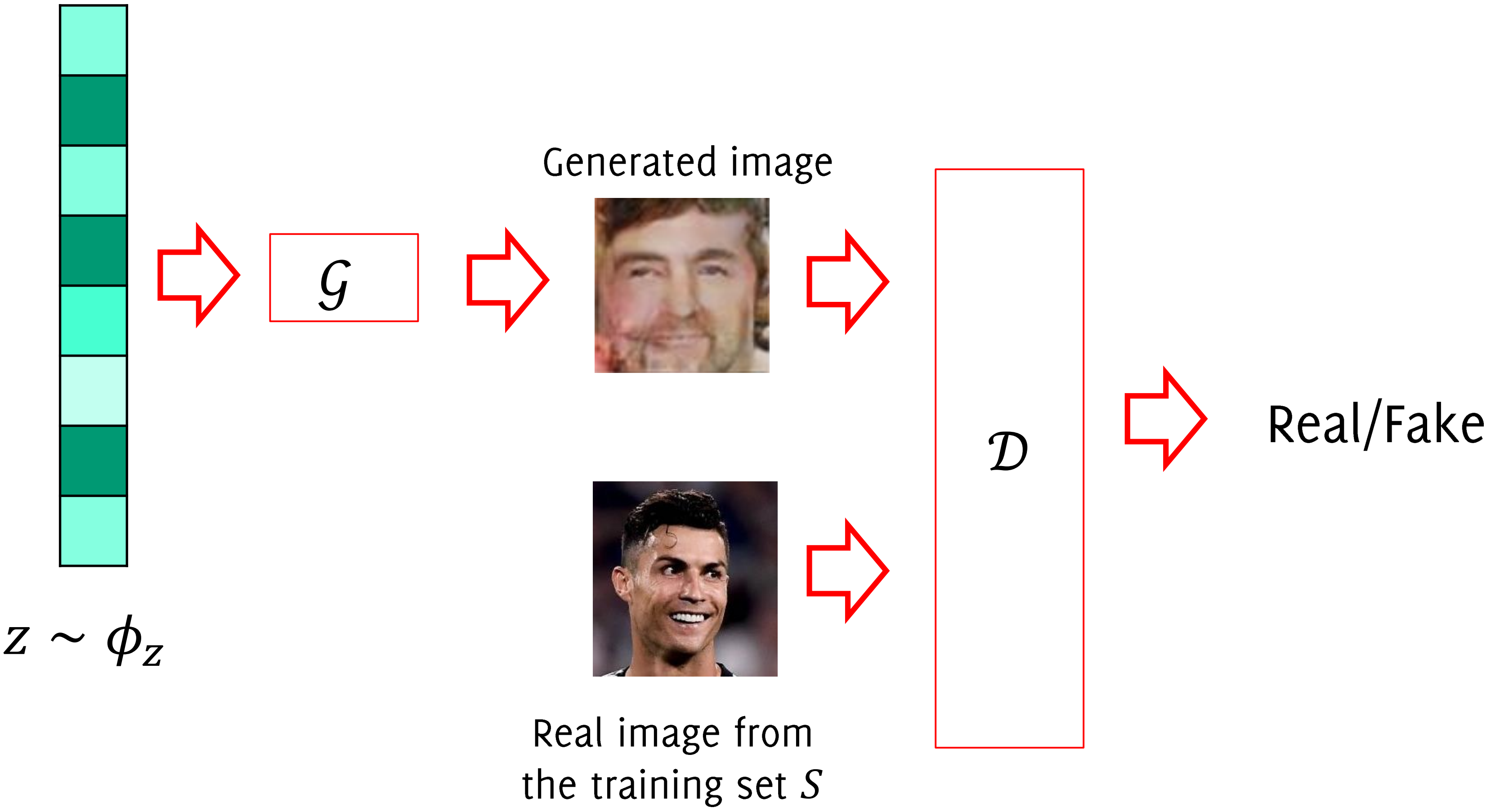
Generated image

Real image from the training set S

$\mathcal{D}$

Real/Fake

GAN ARCHITECTURE



$z \sim \phi_z$

$\mathcal{G}$

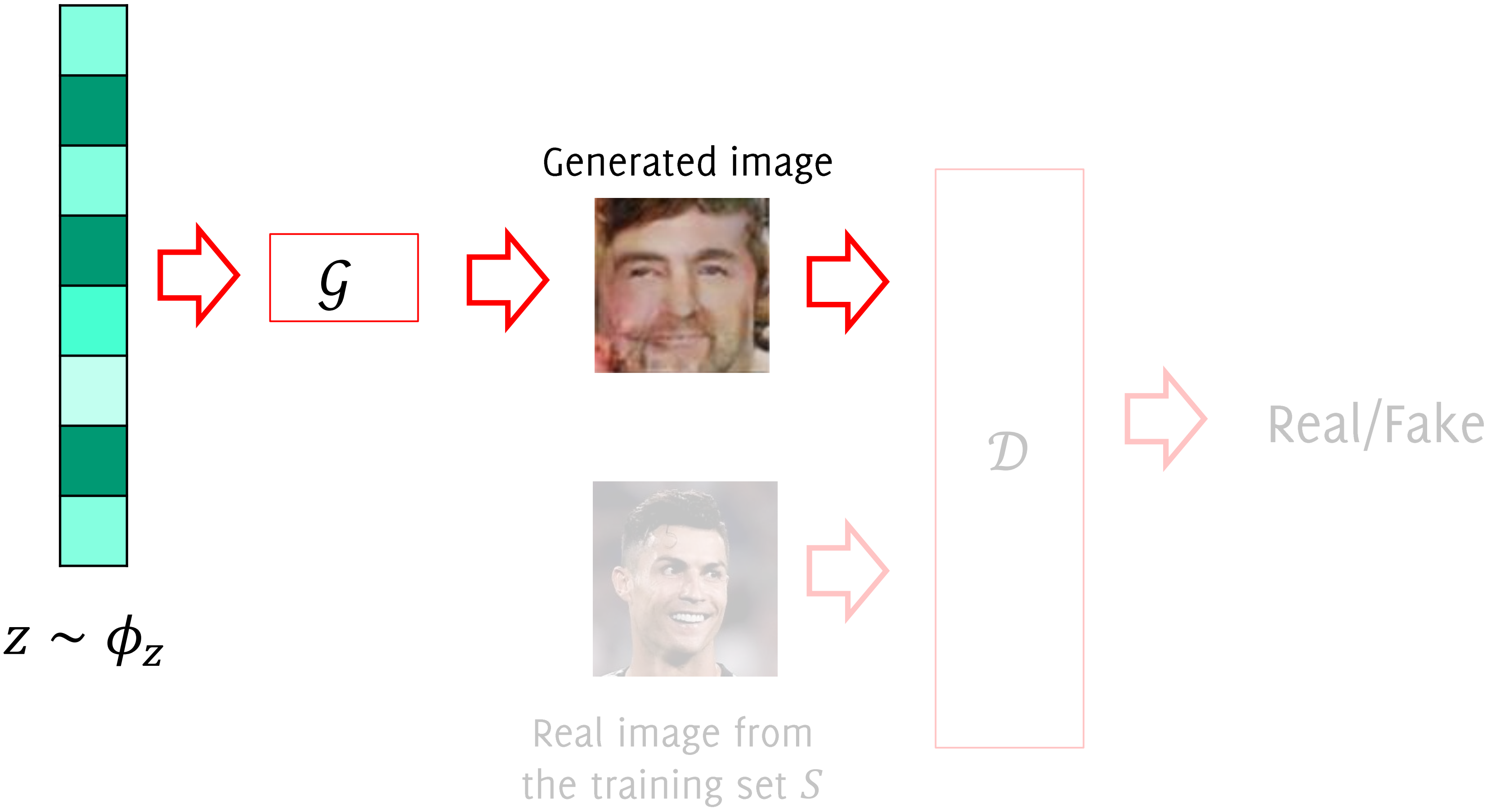
Generated image

Real image from the training set S

$\mathcal{D}$

Real/Fake

USING GAN



Discriminator $\mathcal{D}$ is completely useless and as such dropped. After a successful GAN training, $\mathcal{D}$ is not able to distinguish the real/fake

GAN

Both $\mathcal{D}$ and $\mathcal{G}$ are conveniently chosen as Neural Networks

Setting up the stage

Our networks take as input:

- $\mathcal{D} = \mathcal{D}(\boldsymbol{s})$
- $\mathcal{G} = \mathcal{G}(\boldsymbol{z}\,)$

$\boldsymbol{s} \in \mathbb{R}^n$ is an input image (either real or generated by $\mathcal{G}$) and $\boldsymbol{z} \in \mathbb{R}^d$ is some random noise to be fed to the generator.

Our network give as output:

$$\mathcal{D}(\cdot)\colon \mathbb{R}^n \to [0,1]$$

the posteriori for an input to be a true image (1)

$$\mathcal{G}(\cdot)\colon \mathbb{R}^d \to \mathbb{R}^n$$

the generated image

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GAN TRAINING

A good discriminator is such:

- $\mathcal{D}(\boldsymbol{s})$ is maximum when $s \in S$
- $1 - \mathcal{D}(\boldsymbol{s})$ is maximum when s was generated from $\mathcal{G}$
- $1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)$ is maximum when $\boldsymbol{z} \sim \phi_Z$

Training $\mathcal{D}$ consists in maximizing the binary cross-entroy

$$\max_{\mathcal{D}}\Big(\mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{z\sim\phi_Z}\big[\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))\big]\Big)$$

GAN TRAINING

A good discriminator is such:

- $\mathcal{D}(\boldsymbol{s})$ is maximum when $s \in S$
- $1 - \mathcal{D}(\boldsymbol{s})$ is maximum when s was generated from $\mathcal{G}$
- $1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)$ is maximum when $\boldsymbol{z} \sim \phi_Z$

Training $\mathcal{D}$ consists in maximizing the binary cross-entroy

$$\max_{\mathcal{D}}\Big(\mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{z\sim\phi_Z}\big[\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))\big]\Big)$$

This has to be 1 since $s \sim \phi_S$, thus images are real

This has to be 0 since $\mathcal{G}(\boldsymbol{z})$ is a generated (fake) image

GAN TRAINING

A good discriminator is such:

- $\mathcal{D}(\boldsymbol{s})$ is maximum when $s \in S$
- $1 - \mathcal{D}(\boldsymbol{s})$ is maximum when s was generated from $\mathcal{G}$
- $1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)$ is maximum when $\boldsymbol{z} \sim \phi_Z$

Training $\mathcal{D}$ consists in maximizing the binary cross-entroy

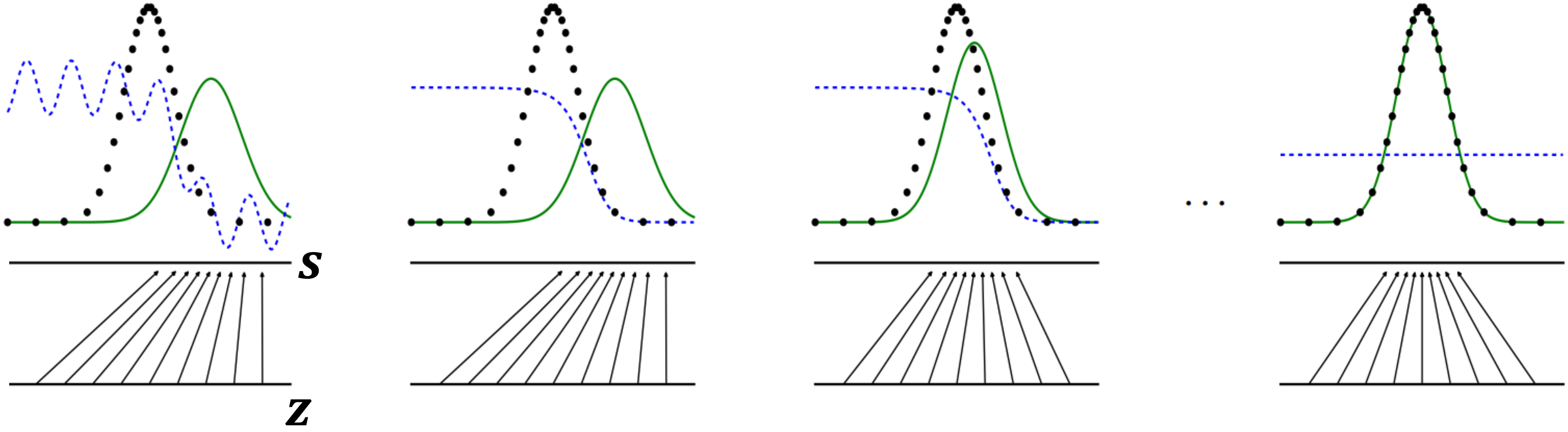
$$\max_{\mathcal{D}}\Big(\mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{z\sim\phi_Z}\Big[\log(1-\mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))\Big]\Big)$$

A good generator $\mathcal{G}$ is the one which makes $\mathcal{D}$ to fail

$$\min_{\mathcal{G}}\max_{\mathcal{D}}\Big(\mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{z\sim\phi_Z}\Big[\log(1-\mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))\Big]\Big)$$

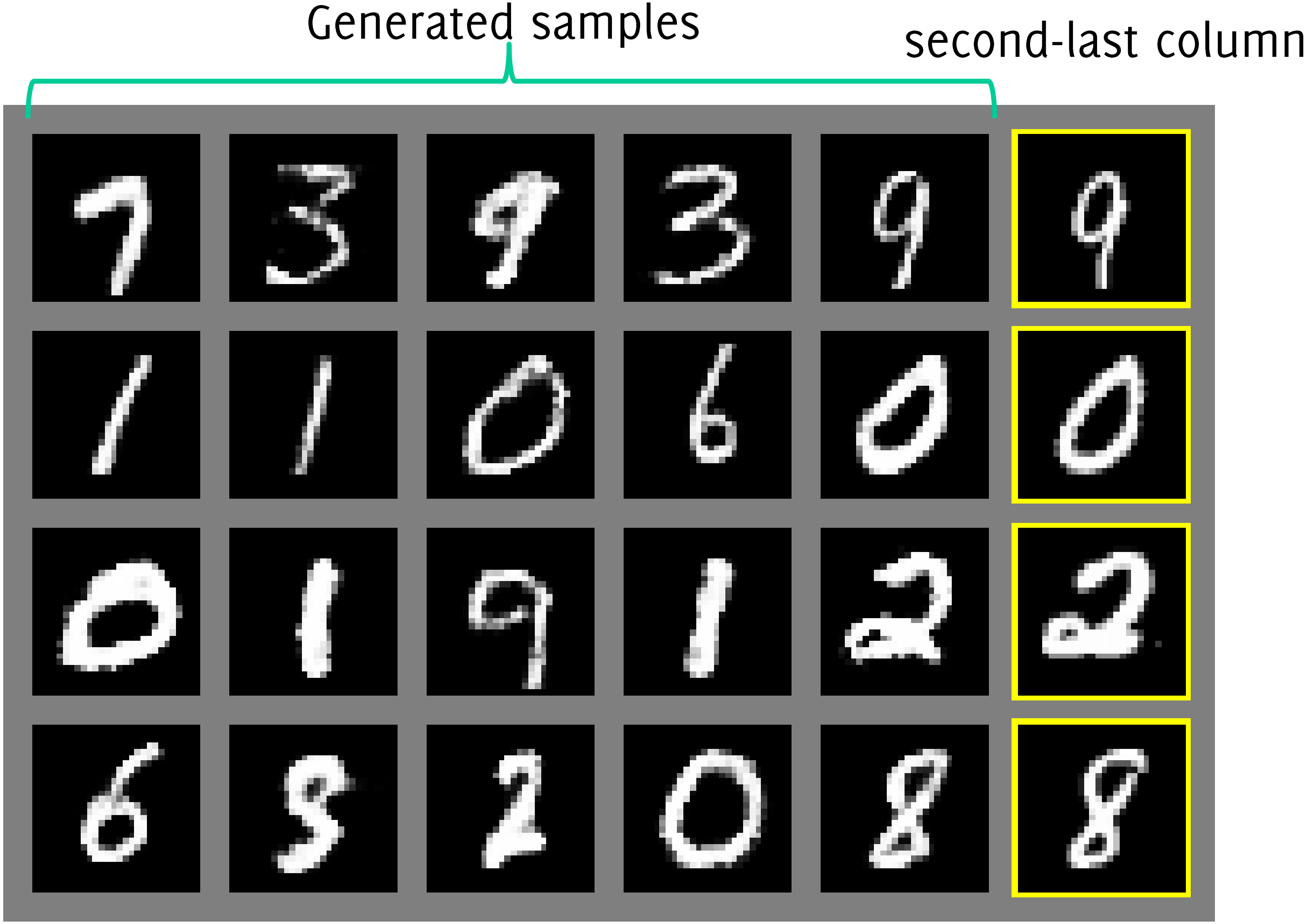
ILLUSTRATION

·······	$\boldsymbol{\phi}_{\boldsymbol{s}}$,	$\boldsymbol{s}$ real
──	$\boldsymbol{\phi}_{\mathcal{G}(\boldsymbol{z})}$	$\mathcal{G}(\boldsymbol{z})$ fake
·······	$\mathcal{D}(\cdot)$	$\mathcal{D}$ posterior



Goodfellow, I. et al "Generative adversarial nets" NIPS 2014

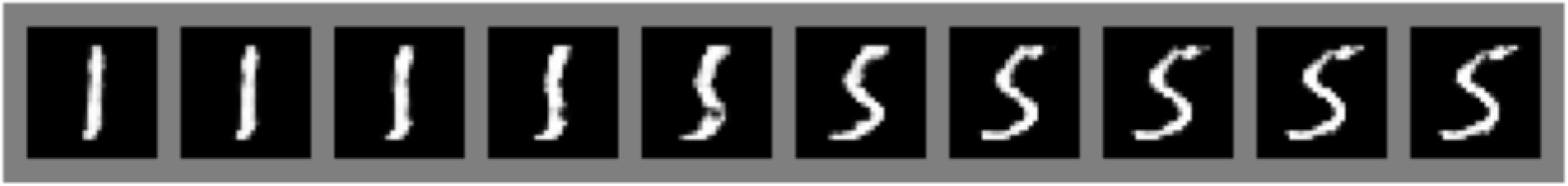
MNIST

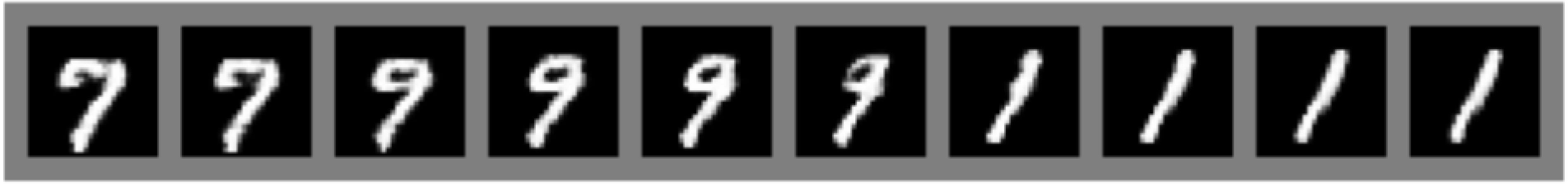


Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

INTERPOLATION IN THE LATENT SPACE

We can interpolate between two points in the latent space and obtain smooth transitions from a digit to another one





Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

AT THE END OF THE DAY…

The discriminator $\mathcal{D}$ is discarded

The generator $\mathcal{G}$ and ϕ_Z are preserved as generative model

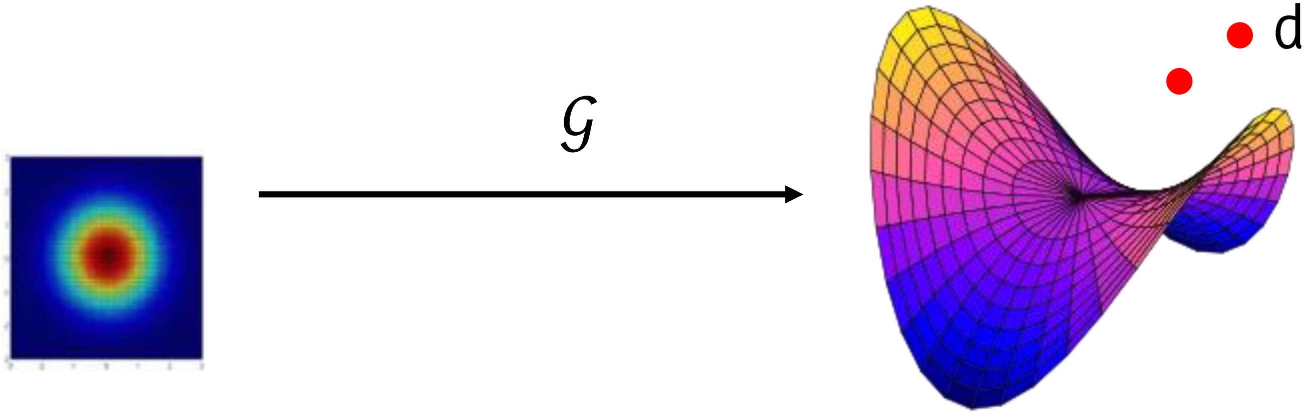
Remarks:

- The training is rather unstable, need to carefully synchronize the two steps (many later works in this direction, e.g. Wasserstein GAN)
- Training by standard tools: backpropagation and dropout
- Theoretical results provided
- Generator does not use S directly during training
- Generator performance is difficult to assess quantitatively
- **There is no explicit expression for the generator**, it is provided in an implicit form -> you cannot compute the likelihood of a sample w.r.t. the learned GAN

GAN FOR ANOMALY DETECTION

Idea: let us train a GAN on normal data. We expect that the generator $\mathcal{G}$ cannot generate any anomalous sample $\boldsymbol{s}$.

Problem: Given a test sample $\boldsymbol{s}$ how can we determine if it could be generated by $\mathcal{G}$?



$\mathcal{G}$

Anomalous data

Latent space

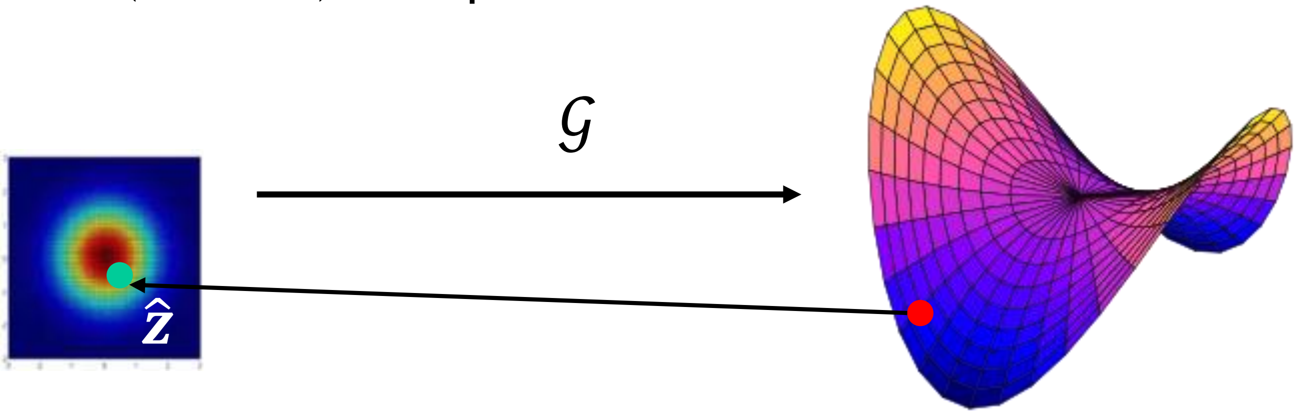
Manifold $\mathcal{M}$ of normal data

ANOGAN

Project the test sample $\boldsymbol{s}$ on the manifold $\mathcal{M}$ by solving the optimization problem:

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\left(\mathcal{G}(\boldsymbol{z})\right)\right)$$

- $\left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right|$ ensures that $\boldsymbol{s}$ is well approximated by the generator
- $\log\left(1 - \mathcal{D}\left(\mathcal{G}(\boldsymbol{z})\right)\right)$ ensures that the projection $\mathcal{G}(\hat{\boldsymbol{z}})$ is similar to a real (normal) sample



Latent space

Manifold $\mathcal{M}$ of normal data

ANOGAN

Project the test sample $\boldsymbol{s}$ on the manifold $\mathcal{M}$ by solving the optimization problem:

$$\hat{\mathbf{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$$

- $\left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right|$ ensures that $\boldsymbol{s}$ is well approximated by the generator
- $\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$ ensures that the projection $\mathcal{G}(\hat{\mathbf{z}})$ is similar to a real (normal) sample (since $\mathcal{G}$ fools $\mathcal{D}$)

Anomaly score:

$$score(\boldsymbol{s}) = \left|\left|\mathcal{G}(\hat{\mathbf{z}}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\hat{\mathbf{z}})\big))$$

Schlegl et al, "*Unsupervised anomaly detection with generative adversarial networks to guide marker discovery*", IPMI 2017

ANOGAN

Project the test sample $\boldsymbol{s}$ on the manifold $\mathcal{M}$ by solving the optimization problem:

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)\right)$$

- $\left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right|$ ensures that $\boldsymbol{s}$ is well approximated by the generator
- $\log\left(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)\right)$ ensures that the projection $\mathcal{G}(\hat{\boldsymbol{z}})$ is similar to a real (normal) sample (since $\mathcal{G}$ fools $\mathcal{D}$)

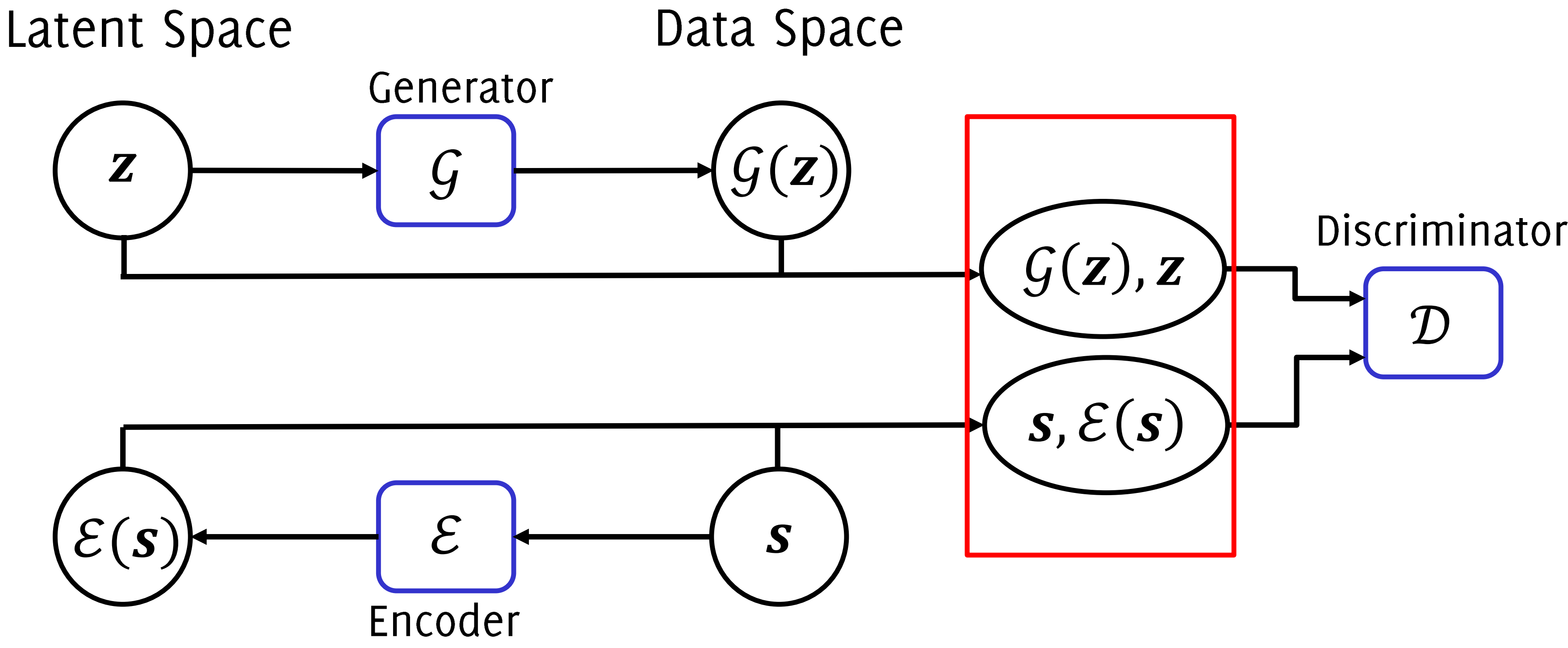
Anomaly score:

$$score(\boldsymbol{s}) = \left|\left|\mathcal{G}(\hat{\boldsymbol{z}}) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\big(\mathcal{G}(\hat{\boldsymbol{z}})\big)\right)$$

We need to solve an optimization problem for each test sample!

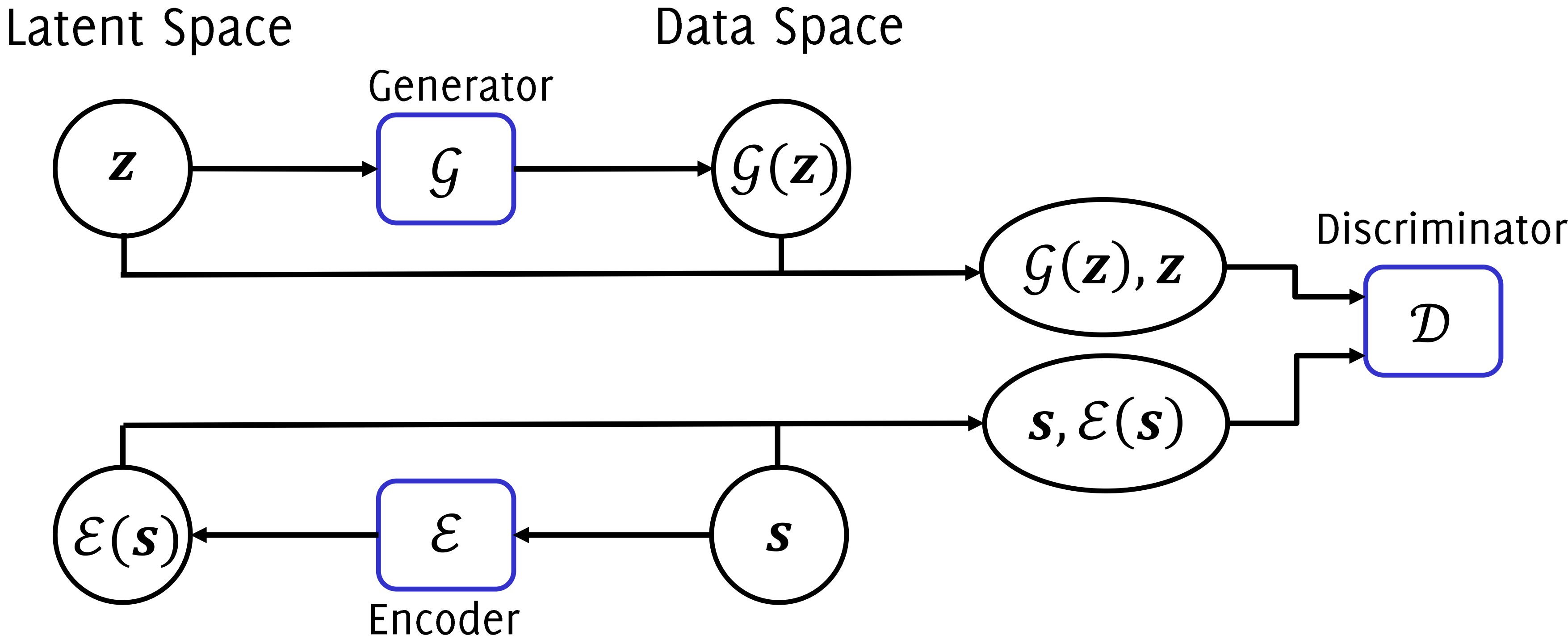
Schlegl et al, "*Unsupervised anomaly detection with generative adversarial networks to guide marker discovery*", IPMI 2017

BIDIRECTIONAL GAN



Donahue et al, "*Adversarial feature learning*", ICLR 2017

BIDIRECTIONAL GAN

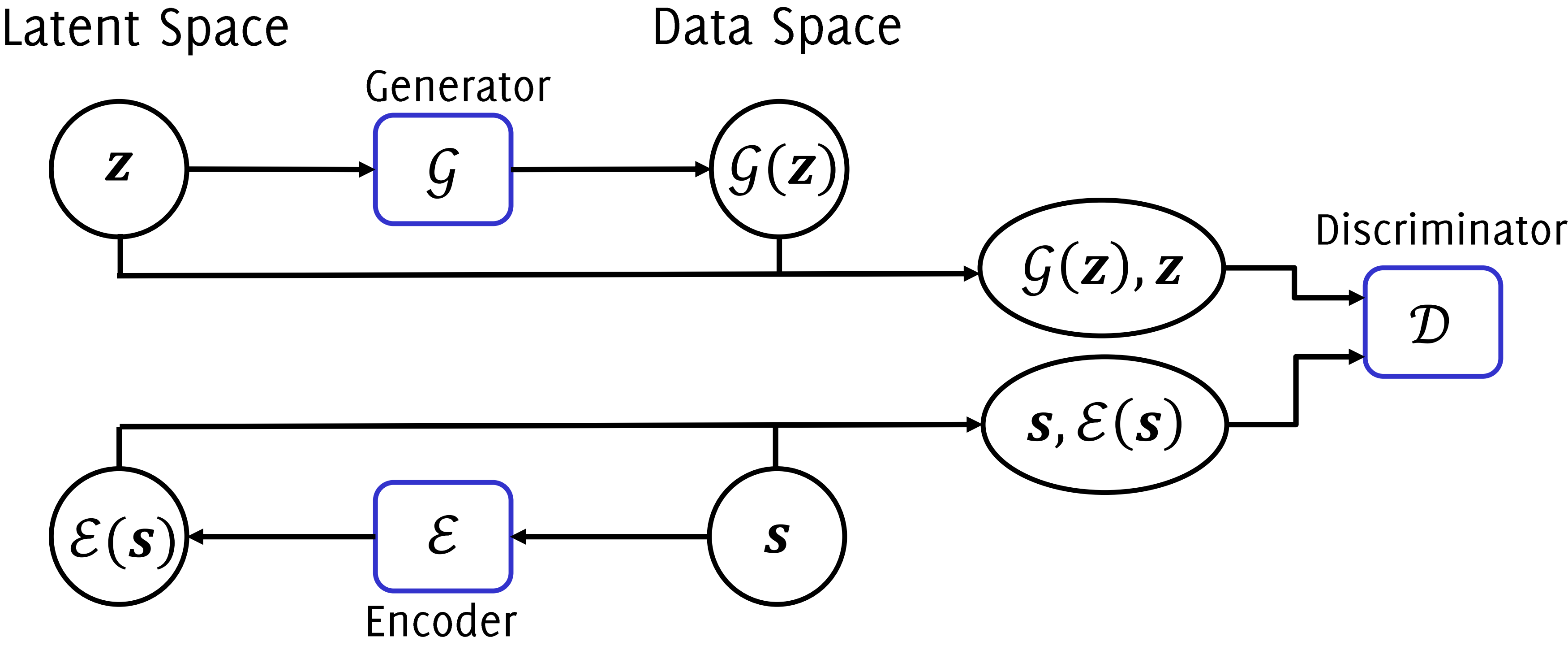

Latent Space
Data Space
Generator
$\boldsymbol{z}$
$\mathcal{G}$
$\mathcal{G}(\boldsymbol{z})$
$\mathcal{G}(\boldsymbol{z}), \boldsymbol{z}$
Discriminator
$\mathcal{D}$
$\boldsymbol{s}, \mathcal{E}(\boldsymbol{s})$
$\mathcal{E}(\boldsymbol{s})$
$\mathcal{E}$
$\boldsymbol{s}$
Encoder

$$\min_{\mathcal{G},\mathcal{E}} \max_{\mathcal{D}} \mathcal{L}(\mathcal{D}, \mathcal{E}, \mathcal{G})$$

$$\mathcal{L}(\mathcal{D}, \mathcal{E}, \mathcal{G}) = \mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s}, \mathcal{E}(\boldsymbol{s})] + \mathrm{E}_{z\sim\phi_Z}[\log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z}), \boldsymbol{z}))]$$

Donahue et al, "*Adversarial feature learning*", ICLR 2017

BIDIRECTIONAL GAN



Latent Space
Data Space
Generator
$\boldsymbol{z}$
$\mathcal{G}$
$\mathcal{G}(\boldsymbol{z})$
$\mathcal{G}(\boldsymbol{z}), \boldsymbol{z}$
Discriminator
$\mathcal{D}$
$\boldsymbol{s}, \mathcal{E}(\boldsymbol{s})$
$\mathcal{E}(\boldsymbol{s})$
$\mathcal{E}$
$\boldsymbol{s}$
Encoder

It can be proved that on the manifold $\mathcal{M}$ (i.e. on normal data):

$$\mathcal{E} = \mathcal{G}^{-1}$$

Donahue et al, "*Adversarial feature learning*", ICLR 2017

ANOGAN IMPROVED

We can use BiGAN to efficiently invert the generator in AnoGAN:

$$\cancel{\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\left(\mathcal{G}(\boldsymbol{z})\right)\right)}$$

$$\hat{\boldsymbol{z}} = \mathcal{E}(\boldsymbol{s})$$

Zenati et al, "*Efficient GAN-based anomaly detection*", Workshop ICLR 2018

ANOGAN IMPROVED

We can use BiGAN to efficiently invert the generator in AnoGAN:

$$\cancel{\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z})))}$$

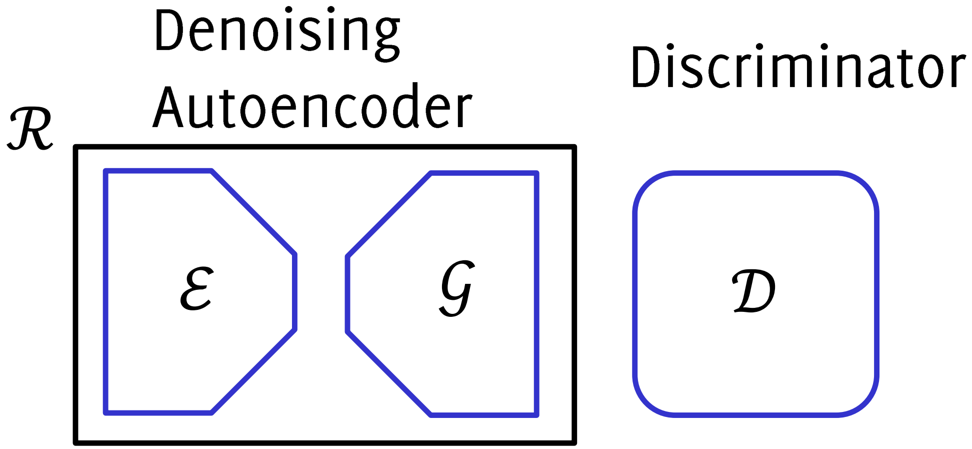
$$\hat{\boldsymbol{z}} = \mathcal{E}(\boldsymbol{s})$$

$$\begin{aligned} score(\boldsymbol{s}) &= \left|\left|\mathcal{G}(\hat{\boldsymbol{z}}) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\left(\mathcal{G}(\hat{\boldsymbol{z}})\right)\right) = \\ &= \left|\left|\mathcal{G}(\mathcal{E}(\boldsymbol{s})) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\left(\mathcal{G}(\mathcal{E}(\boldsymbol{s}))\right)\right) \end{aligned}$$

Zenati et al, "*Efficient GAN-based anomaly detection*", Workshop ICLR 2018

AUTOENCODER AS GENERATIVE MODELS

Noise is added to real images

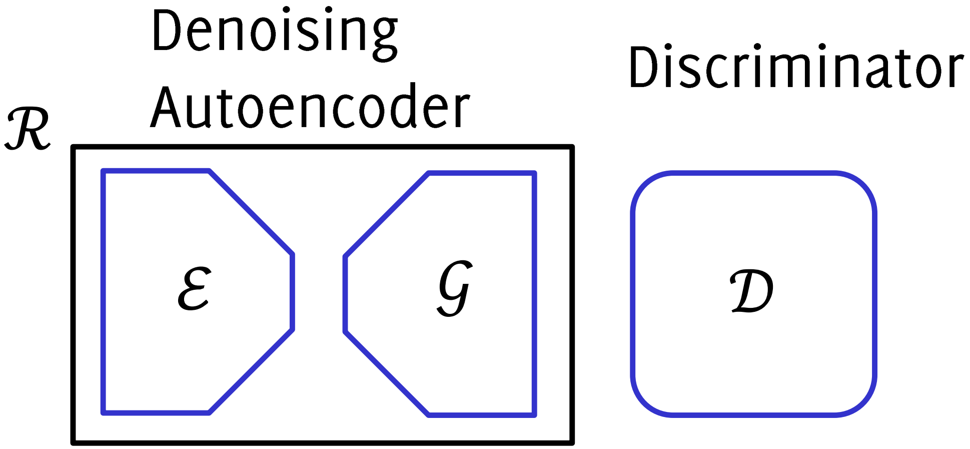


$$\boldsymbol{s} + \mathcal{N}(0,\sigma^2) \longrightarrow \boldsymbol{z} \longrightarrow \boldsymbol{s}' \longrightarrow [0,1]$$

$$\min_{\mathcal{R}} \max_{\mathcal{D}} \Big(\mathrm{E}_{\boldsymbol{s}\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{\tilde{\boldsymbol{s}}\sim\phi_S+\mathcal{N}(0,\sigma^2)}\big[\log(1-\mathcal{D}(\mathcal{R}(\tilde{\boldsymbol{s}})))\big]\Big)$$

Sabokrou et al, "*Adversarially learned one-class classifier for novelty detection*", CVPR 2018

AUTOENCODER AS GENERATIVE MODELS

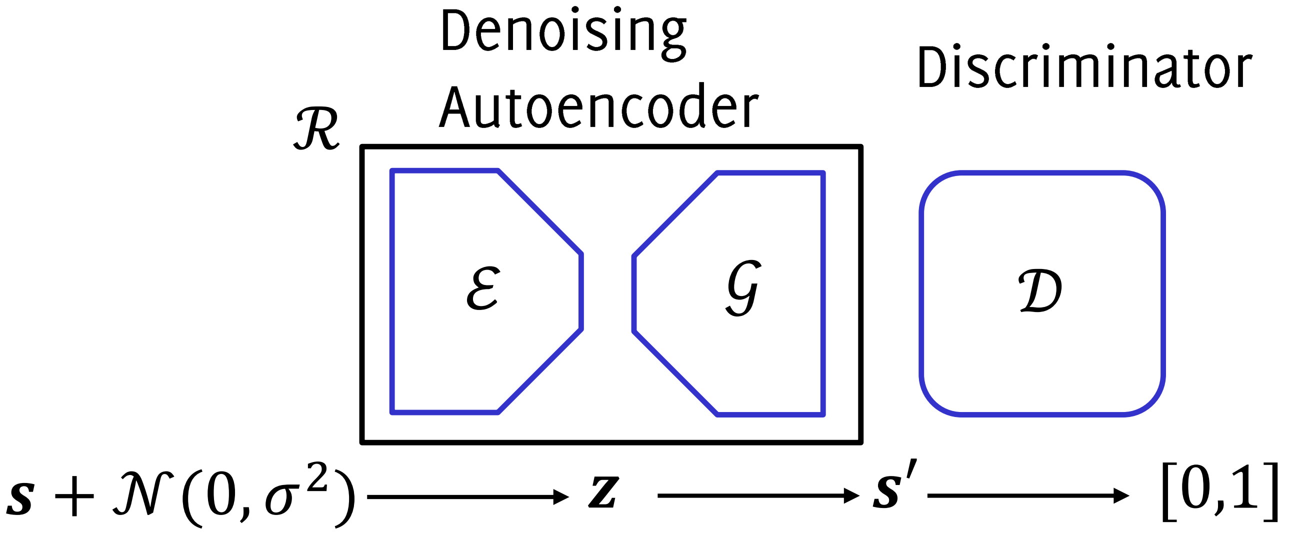


$$\boldsymbol{s} + \mathcal{N}(0, \sigma^2) \longrightarrow \boldsymbol{z} \longrightarrow \boldsymbol{s}' \longrightarrow [0,1]$$

$$\min_{\mathcal{R}} \max_{\mathcal{D}} \Big(\mathrm{E}_{\boldsymbol{s}\sim\phi_S}[\log \mathcal{D}(\boldsymbol{s})] + \mathrm{E}_{\boxed{\tilde{\mathbf{s}}\sim\phi_S+\mathcal{N}(0,\sigma^2)}}\big[\log\big(1 - \mathcal{D}\big(\mathcal{R}(\tilde{\mathbf{s}})\big)\big)\big]\Big)$$

Sabokrou et al, "*Adversarially learned one-class classifier for novelty detection*", CVPR 2018

AUTOENCODER AS GENERATIVE MODELS



$$\mathbf{s} + \mathcal{N}(0,\sigma^2) \longrightarrow \mathbf{z} \longrightarrow \mathbf{s}' \longrightarrow [0,1]$$

$$\min_{\mathcal{R}} \max_{\mathcal{D}} \left(\mathrm{E}_{\mathbf{s}\sim\phi_S}[\log \mathcal{D}(\mathbf{s})] + \mathrm{E}_{\tilde{\mathbf{s}}\sim\phi_S+\mathcal{N}(0,\sigma^2)}\left[\log\left(1-\mathcal{D}\left(\mathcal{R}(\tilde{\mathbf{s}})\right)\right)\right]\right)$$

We expect that $\mathcal{R}$ can successfully reconstruct (thus, fool $\mathcal{D}$) only normal sample:

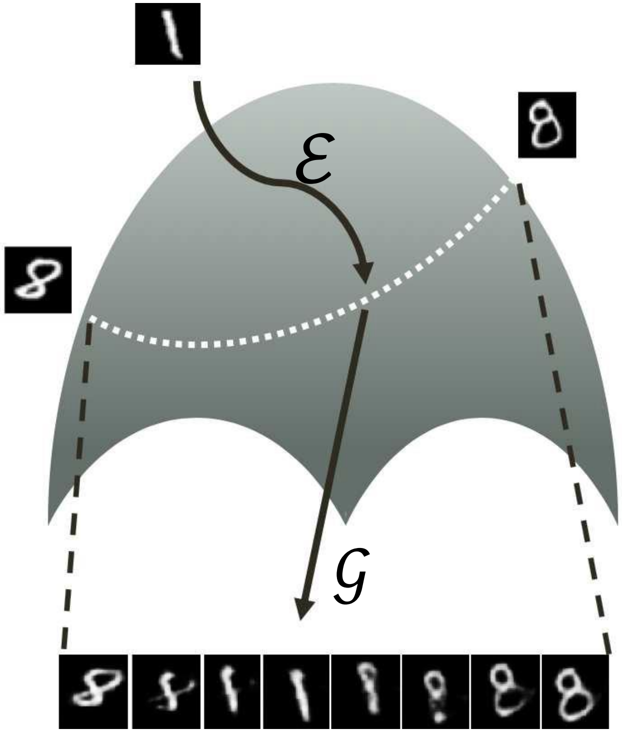
$$score(\mathbf{s}) = \mathcal{D}(\mathcal{R}(\mathbf{s}))$$

Sabokrou et al, "*Adversarially learned one-class classifier for novelty detection*", CVPR 2018

AUTOENCODER AS GENERATIVE MODELS

The generator $\mathcal{G}$ may be able to generate samples also of **anomalous class**

In this case it would be **impossible** to use $\mathcal{G}$ to **discriminate** between normal and anomalous samples



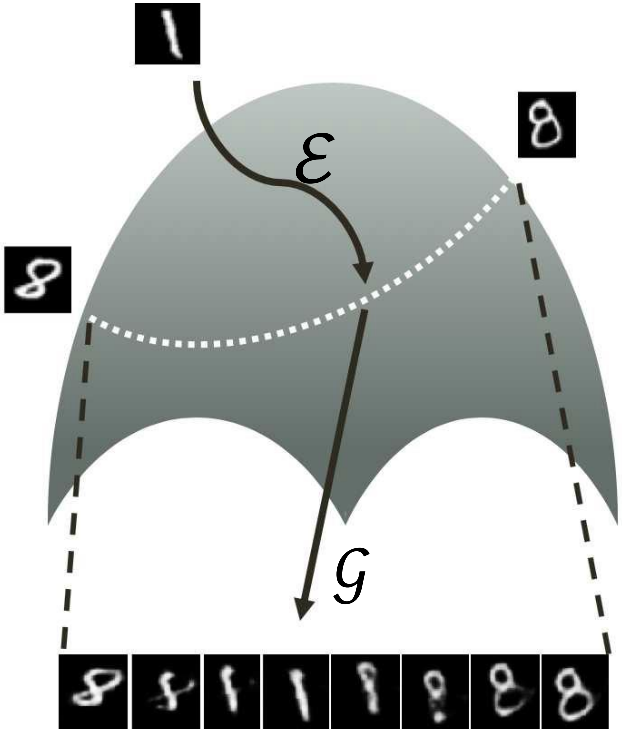
Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

AUTOENCODER AS GENERATIVE MODELS

The generator $\mathcal{G}$ may be able to generate samples also of **anomalous class**

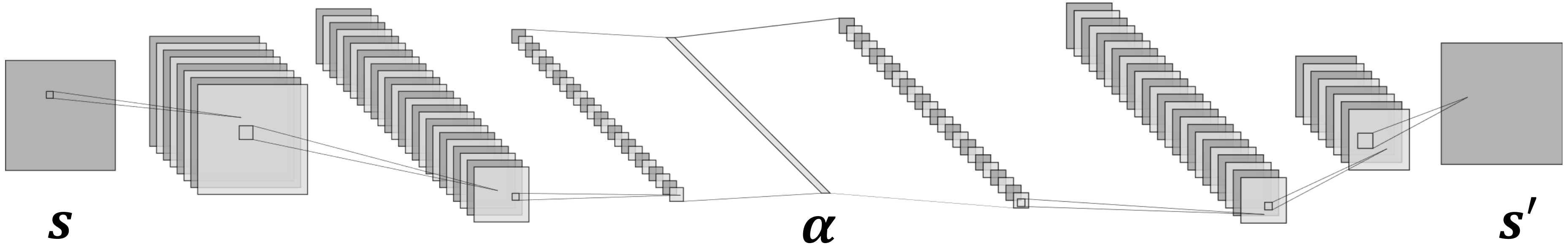
In this case it would be **impossible** to use $\mathcal{G}$ to **discriminate** between normal and anomalous samples

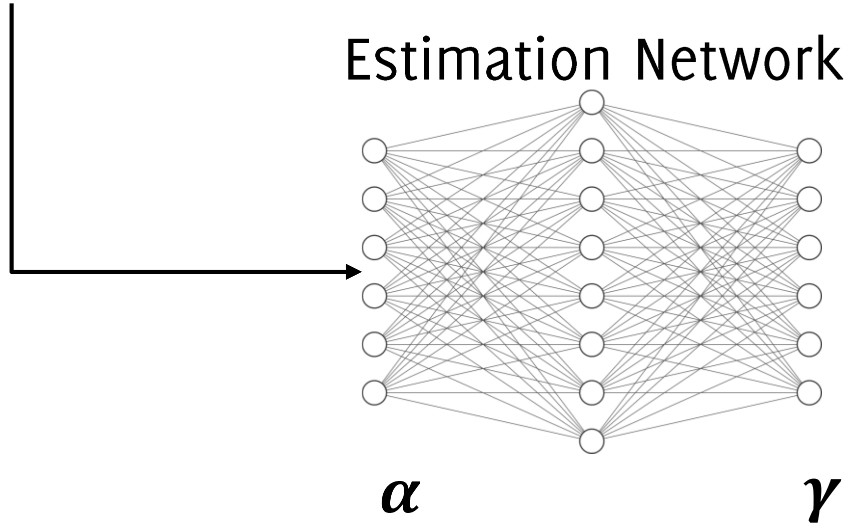
Idea: we can enforce a known distribution on the latent space



Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

AUTOENCODER AS GENERATIVE MODELS





Estimate the GM parameters as:

$$\pi_i = \frac{1}{N}\sum_n \gamma_{n,i}$$

$$\mu_i = \frac{\sum_n \gamma_{n,i}\boldsymbol{\alpha}_n}{\sum_n \gamma_{n,i}}$$

$$\Sigma_i = \frac{\sum_n \gamma_{n,i}(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)^T}{\sum_n \gamma_{n,i}}$$

Zong et al, *"Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection"*, ICLR 2018

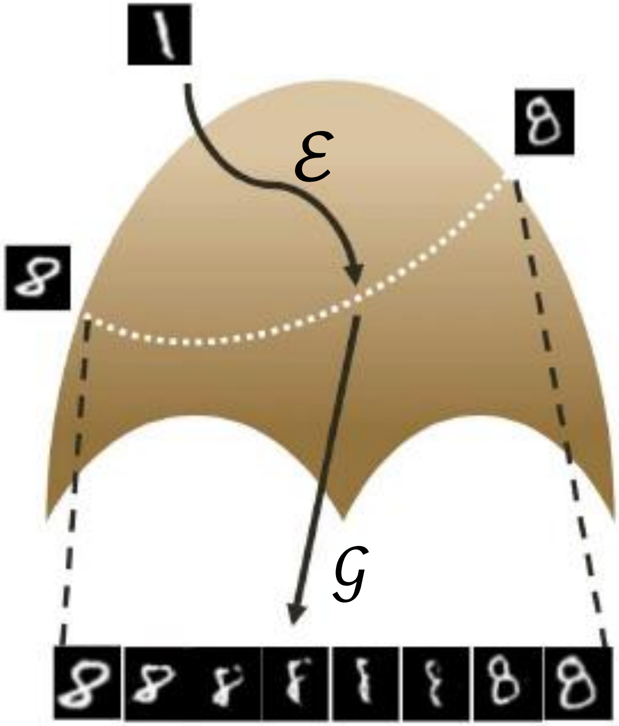
AUTOENCODER AS GENERATIVE MODELS

The generator $\mathcal{G}$ may be able to generate samples also of **anomalous class**

In this case it would be **impossible** to use $\mathcal{G}$ to **discriminate** between normal and anomalous samples

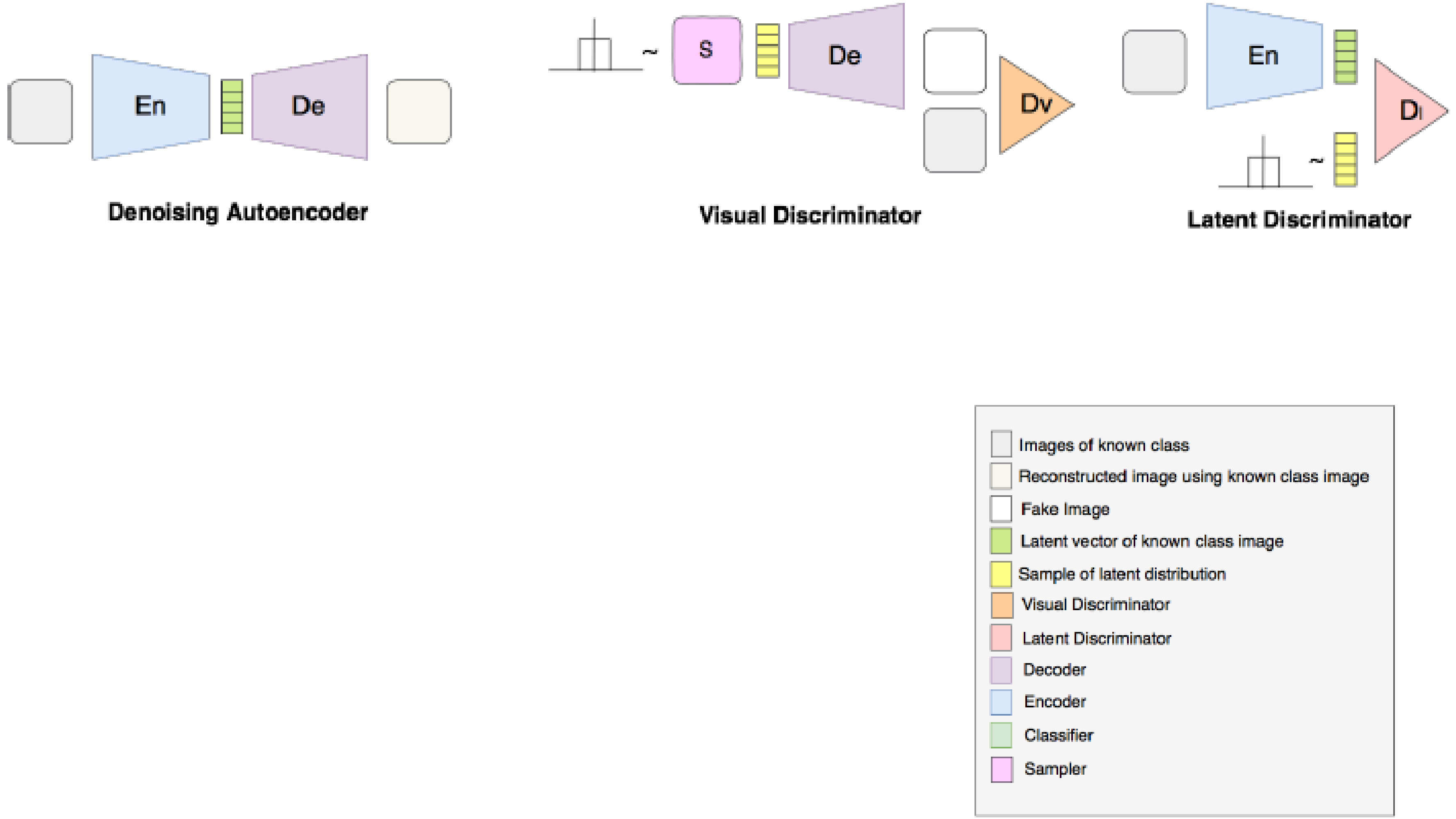
Idea: we can enforce a known distribution on the latent space

This can be done using a **latent discriminator** on the latent space



Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

MUCH MORE COMPLICATED ARCHITECTURE



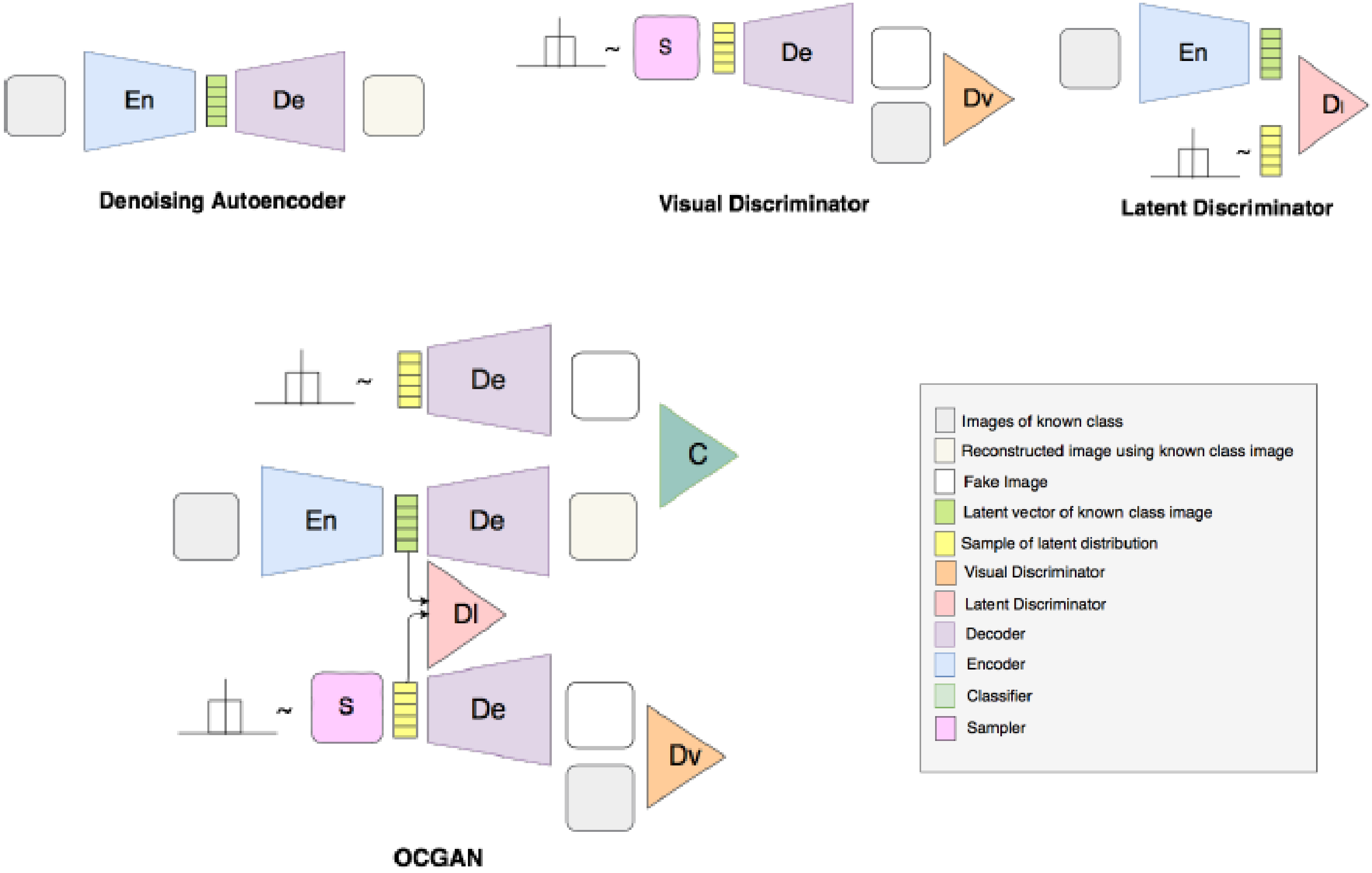
Denoising Autoencoder

Visual Discriminator

Latent Discriminator

Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

MUCH MORE COMPLICATED ARCHITECTURE



Denoising Autoencoder

Visual Discriminator

Latent Discriminator

OCGAN

Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

Concluding Remarks

A FEW CONCLUDING REMARKS

Nowadays, anomaly detection problems are **ubiquitous** in **engineering and applied sciences.**

The presented general **framework encompasses most of algorithms** in the literature, which often **boil down to**

- **Feature extraction**
- **Definition of suitable statistics**
- **Applying decision rules** to a set of **random variables.**

A FEW CONCLUDING REMARKS

When **data** are characterized by **complex structures**, as in case of images and signals, the feature extraction phase is the most critical one.

Data-driven models provide **meaningful representations** to images, that can be used to extract good feature for detection.

Nowadays the most powerful algorithms for feature extraction are based on deep learning, and in particular **Convolutional Neural Networks**

A FEW CONCLUDING REMARKS

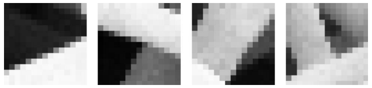
CNNs can be used either:

- As **data-driven feature extractor** that are put on top of an anomaly detector designed for random vectors
    - The best performance are achieved where the CNN and the anomaly detector are **jointly learned**
- As a **generative model** that allows to sample from the distribution of normal images
    - This generator has to be somehow inverted for anomaly detection

SOME RESOURCES

Annotated Datasets:

- http://web.mi.imati.cnr.it/ettore/NanoTWICE/
- https://www.kaggle.com/c/severstal-steel-defect-detection/data



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

SOME RESOURCES

Public software:

- Anomaly detection using sparse representations
  http://home.deib.polimi.it/boracchi/Projects/projects.html
  https://home.deib.polimi.it/carrerad/projects.html
- https://github.com/PramuPerera/OCGAN
- https://github.com/izikgo/AnomalyDetectionTransformations
- https://github.com/houssamzenati/Efficient-GAN-Anomaly-Detection.git
- https://github.com/lukasruff/Deep-SVDD