Unsupervised Machine Learning in Optical Surface Inspection

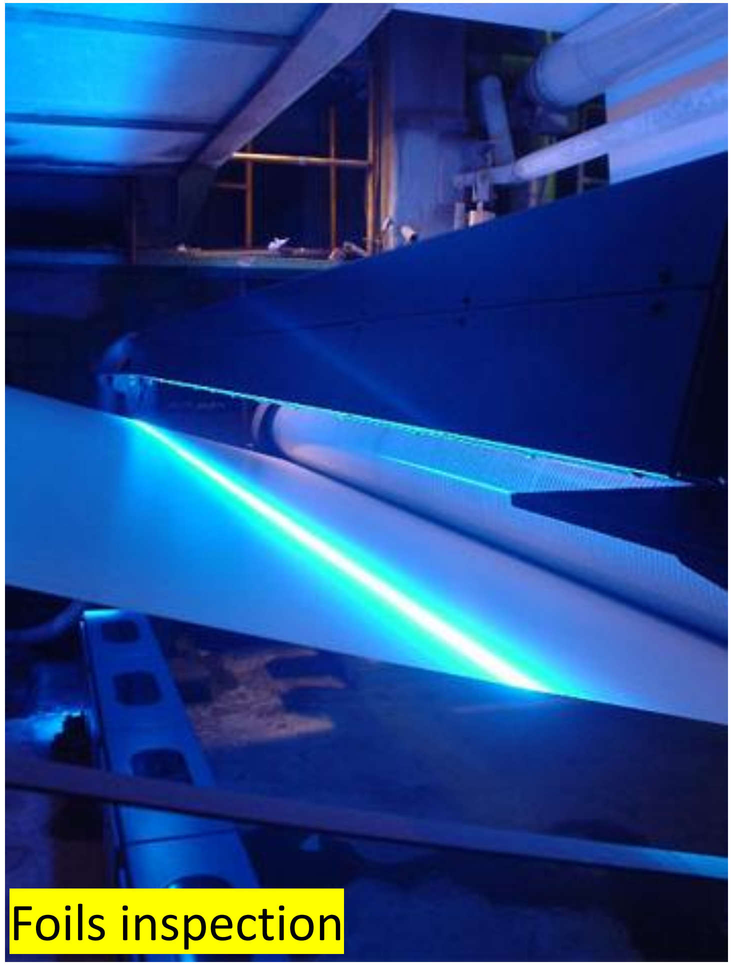
Brown-Bag-Seminar 2019, IOS, Konstanz

Matthias Hermann, HTWG-Konstanz, Institute for Optical Systems
Promotionsstart: 11/2017

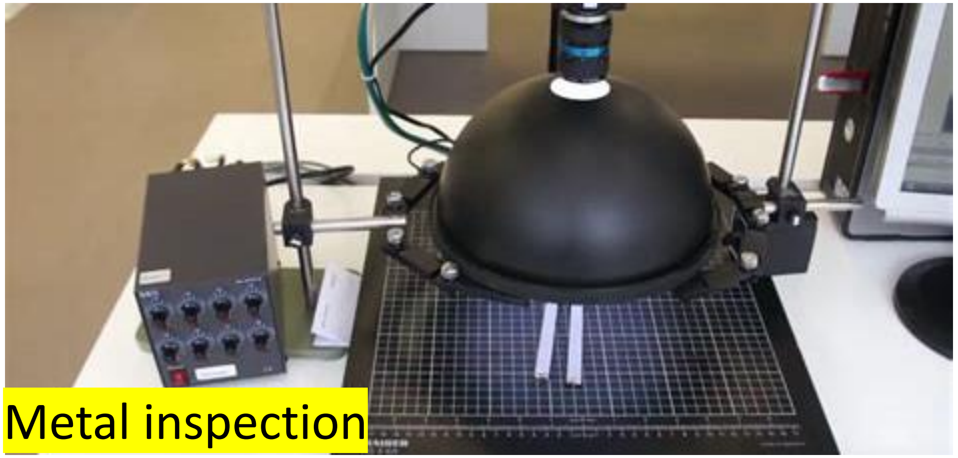
Advisors:
Prof. Dr. Georg Umlauf, HTWG-Konstanz, Institute for Optical Systems
Prof. Dr. Bastian Goldlücke, Universität Konstanz, Computer Vision/Image Analysis

BMBF-FZ: 13N14540

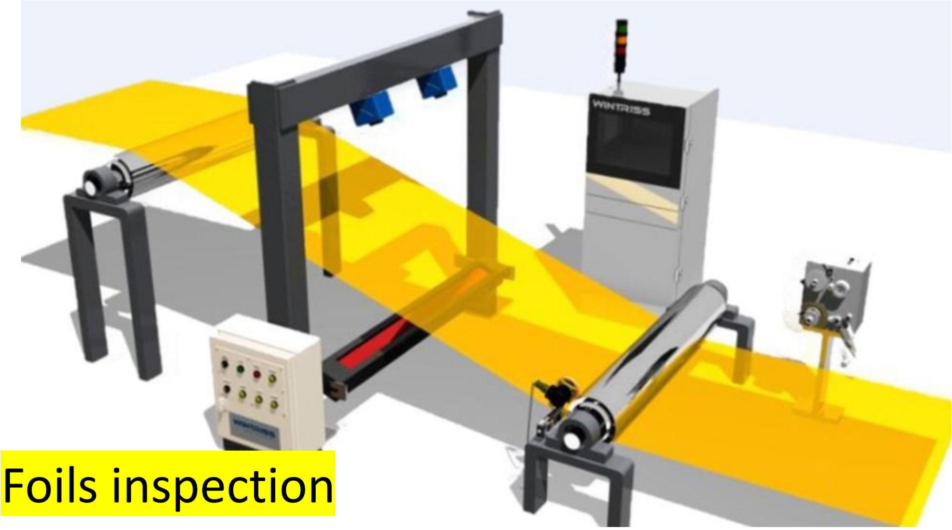
Optical surface inspection I



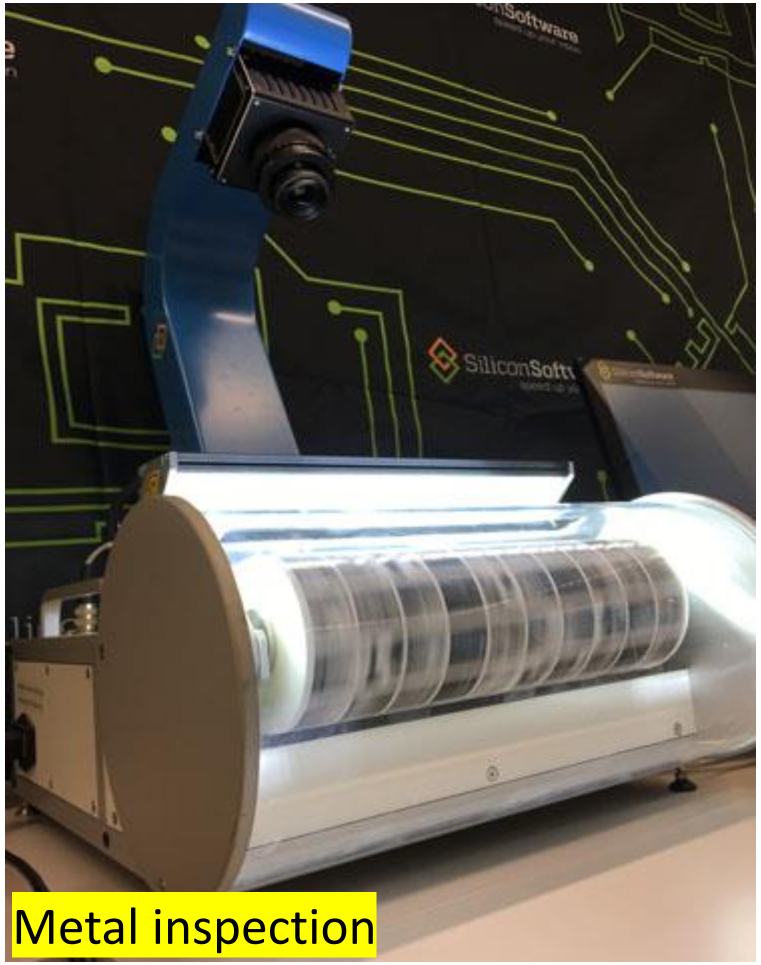
Foils inspection



Metal inspection



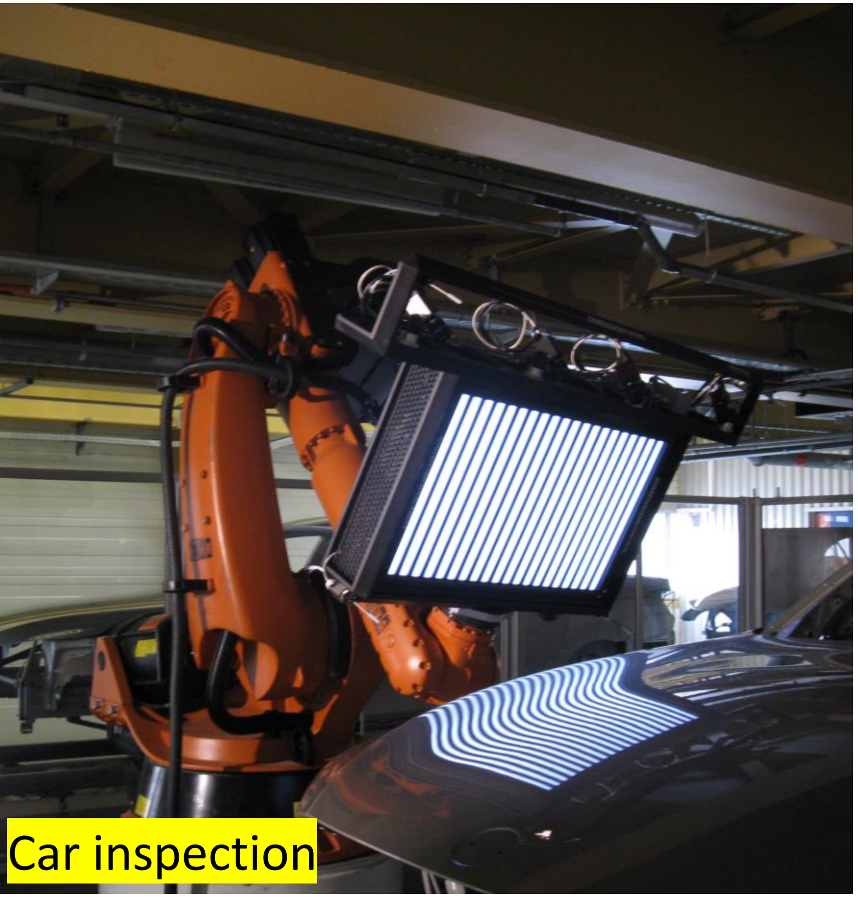
Foils inspection



Metal inspection

Source: Silicon Software, Wintriss Engineering, Weco, Kuka

Optical surface inspection II

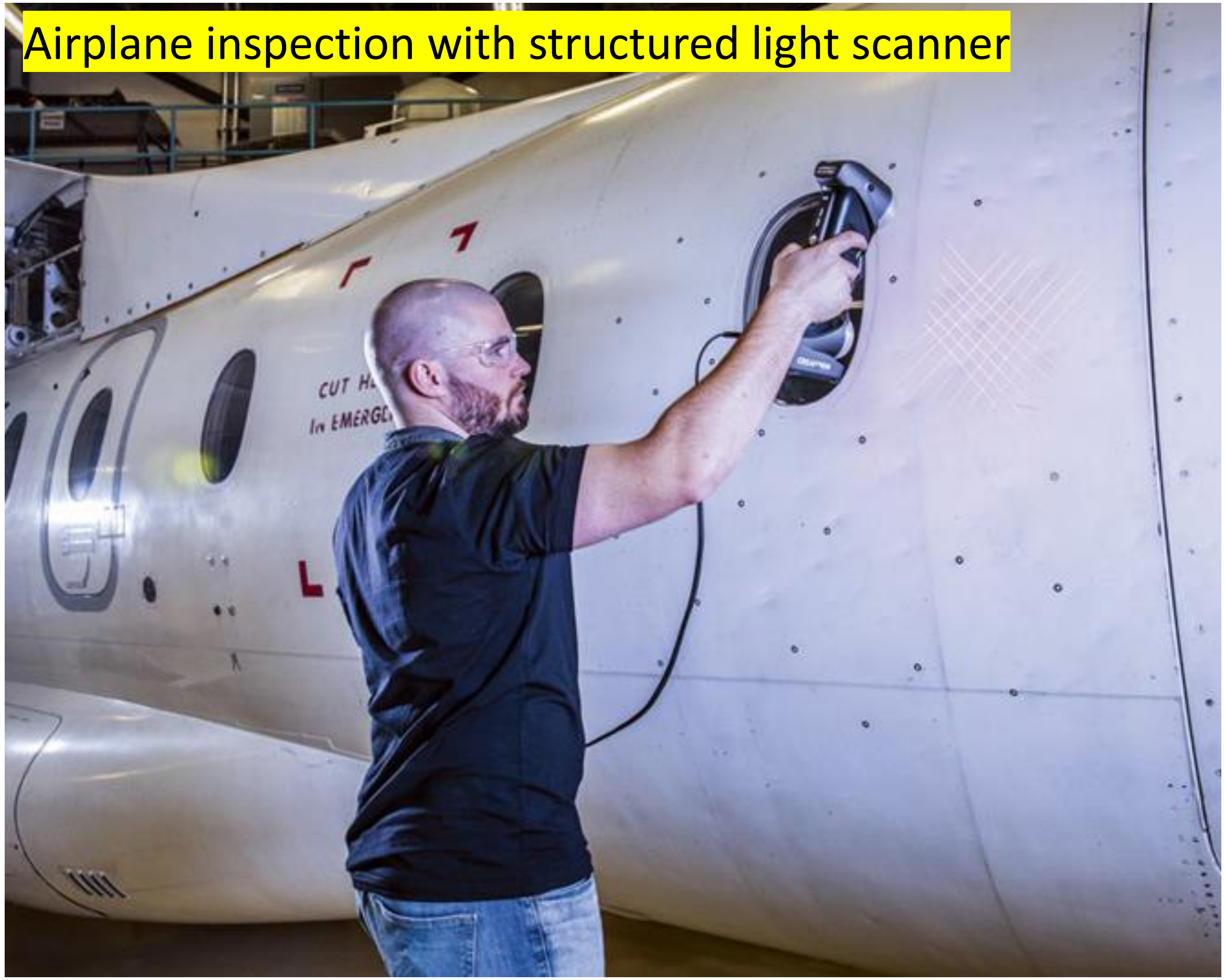

Car inspection


Food inspection


Wood inspection

Source: Silicon Software, Wintriss Engineering, Weco, Kuka

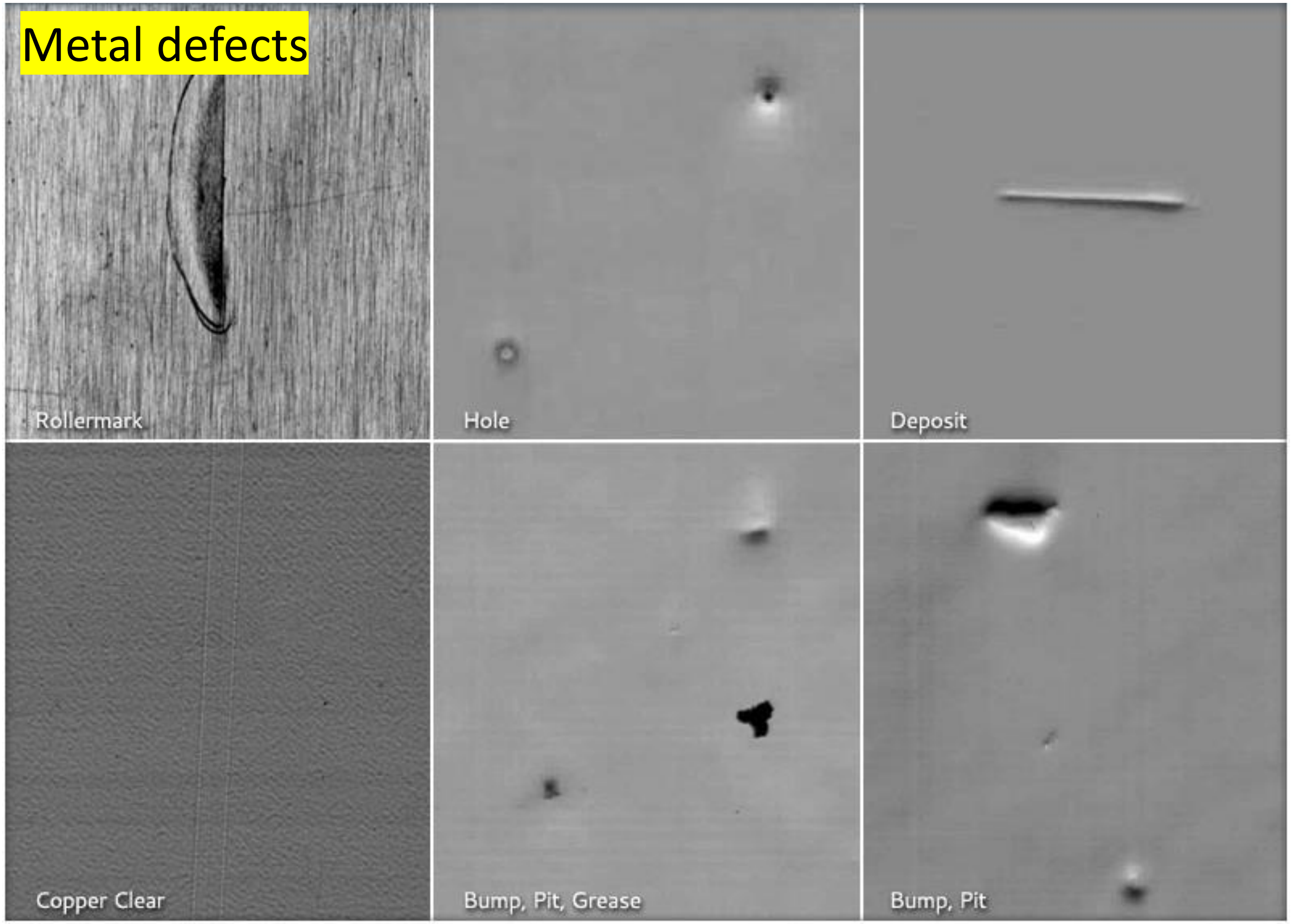
Microscopic and macroscopic applications



Source: creaform3d, Zeisscreaform3d, Zeiss

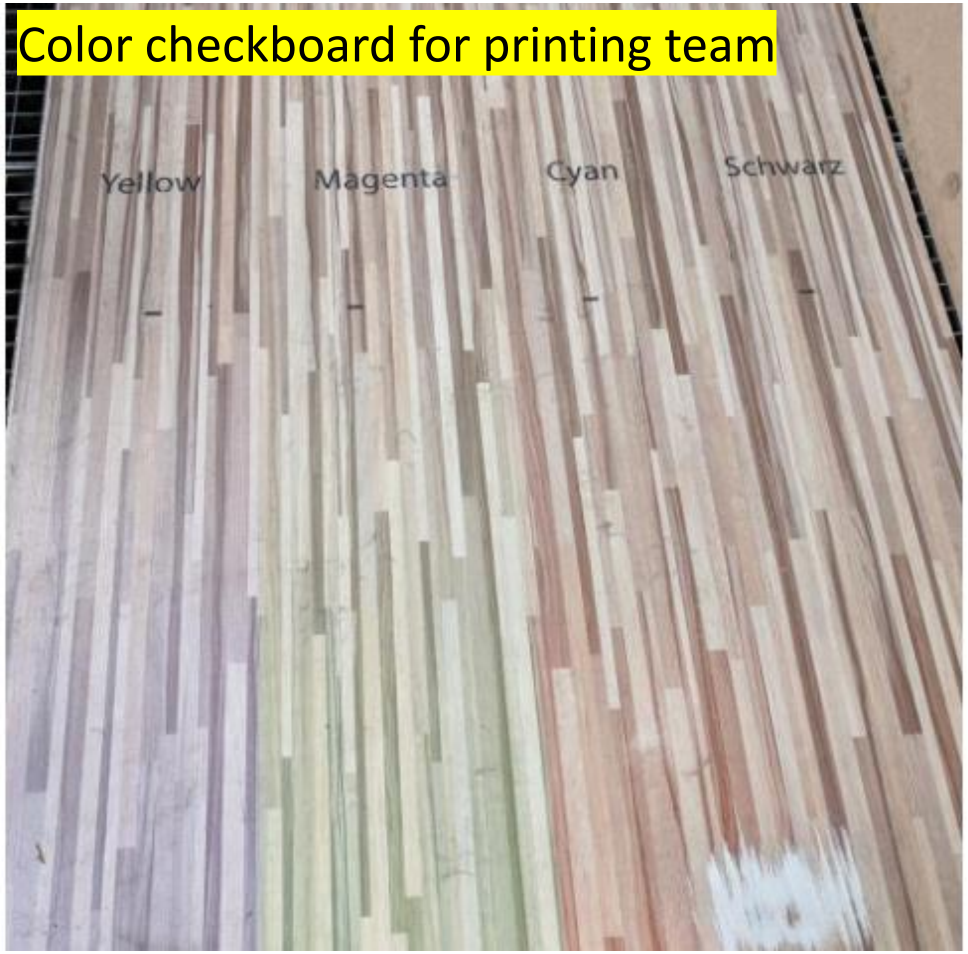
Typical applications of surface inspection

- Quality control in manufacturing
- Automation
- Medical imaging e.g. mammography
- Material sciences
- Reconstruction and repairing of things

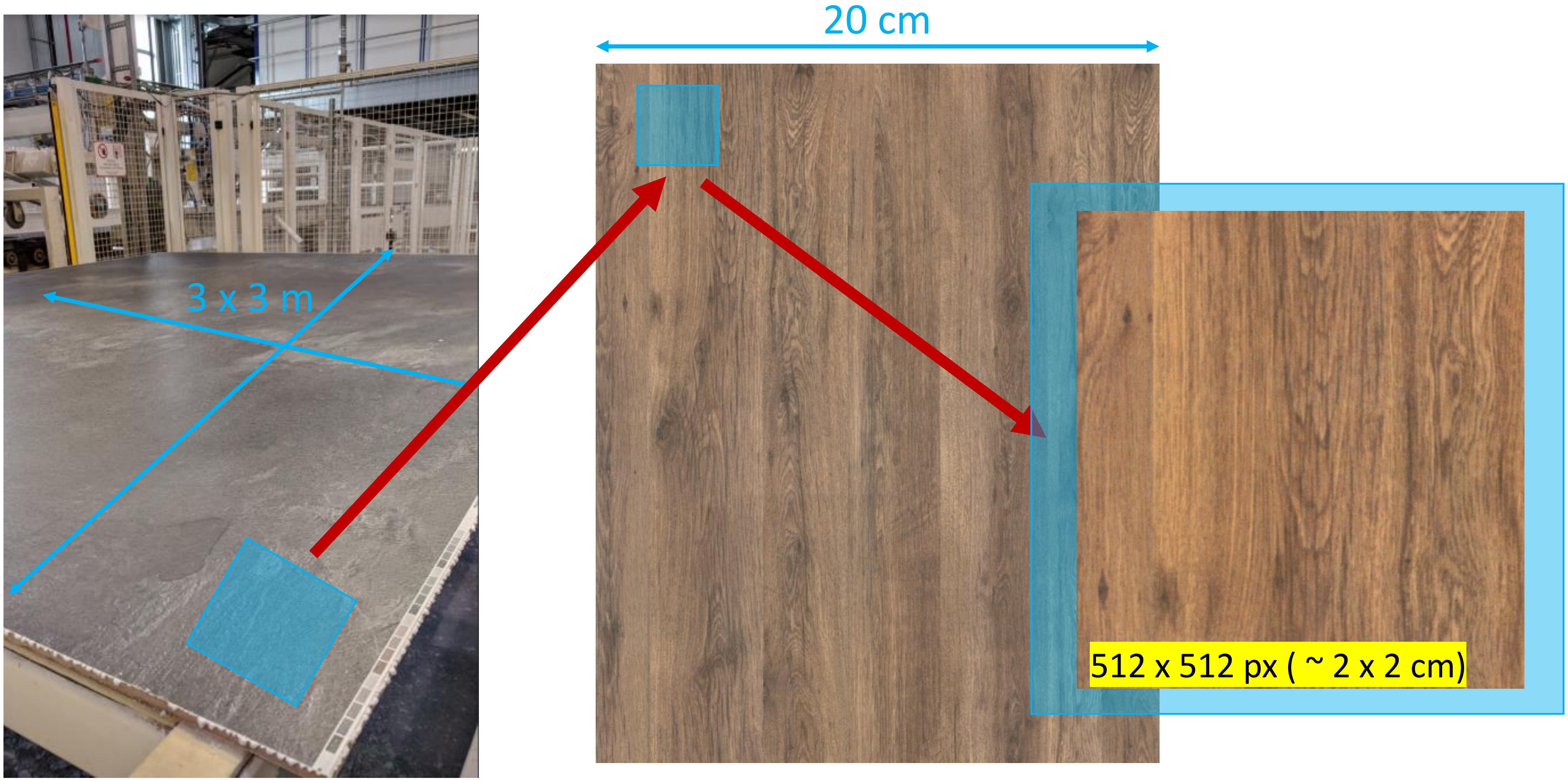


Source: creaform3d, Zeisscreaform3d, Zeiss

My Application: Digital Print Inspection



Inspecting digital printed wooden decors



Recorded Example I

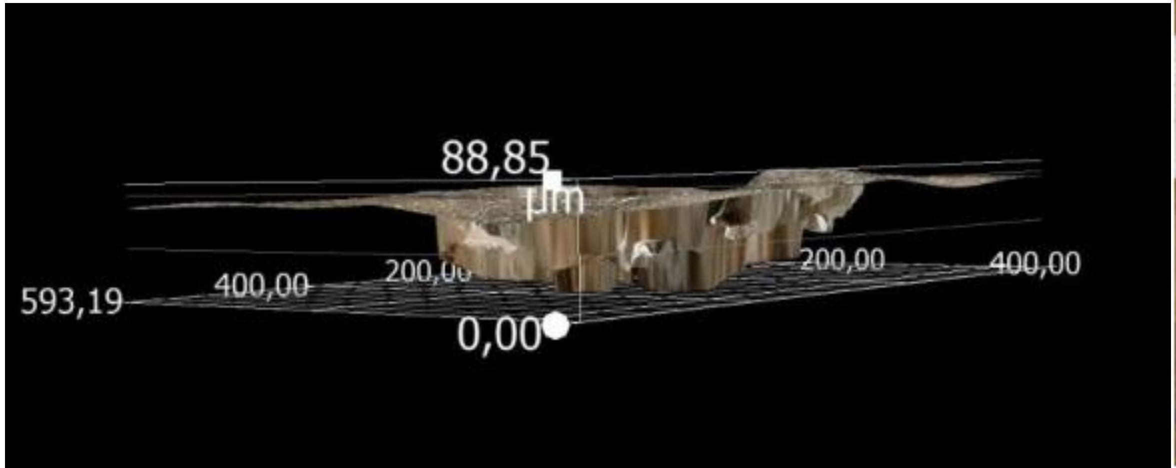
Recorded Example II

Error-free reference

Printing failure through nozzle head fault

Problem statement

„Detection of texture independent surface defects and digital printing errors in multi spectral color and depth images“



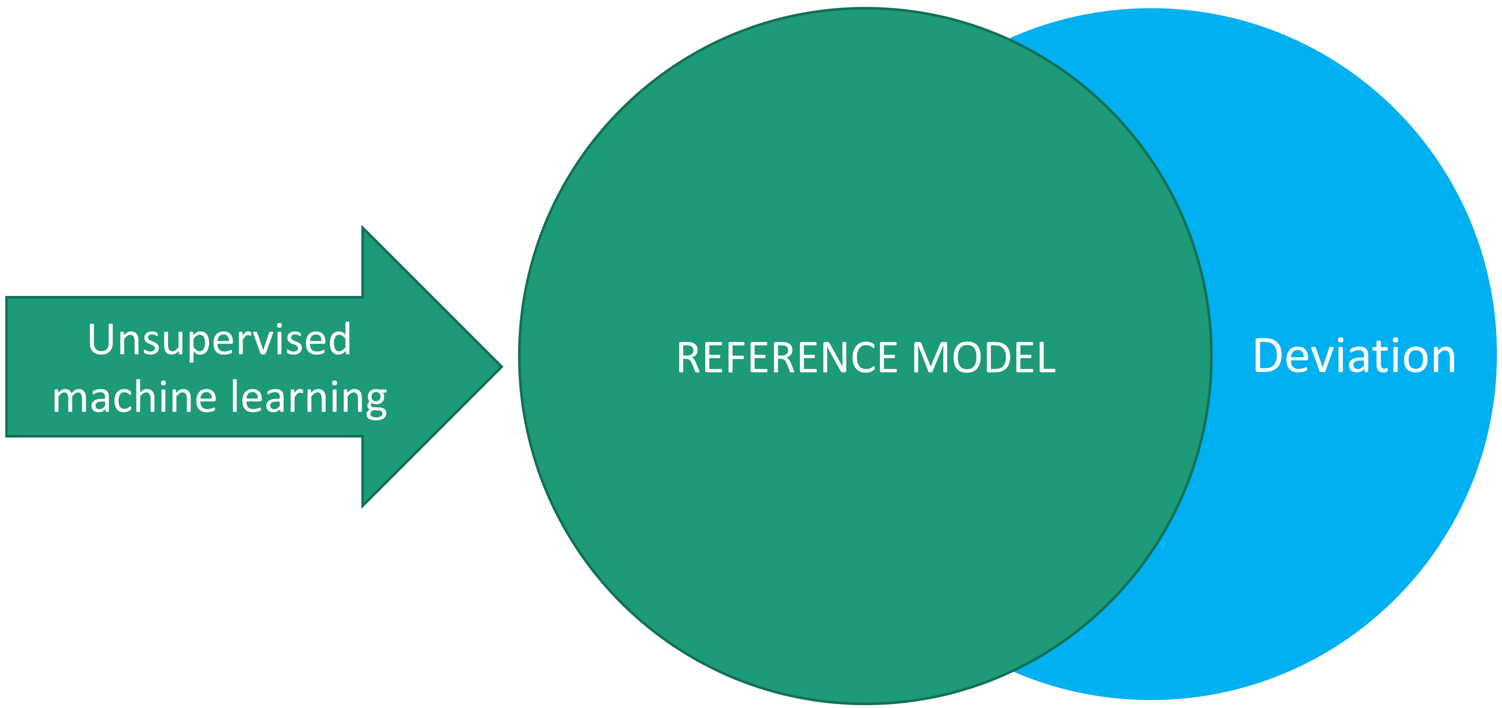
Classical approaches

Error models	Lighting models	Classification
- Edge amd blob detectors - Statistical methods - Color spaces - Feature spaces	- Lights and lasers - Reflectance - Registration - …	- Human expert - Machine learning - …
>> Creativity!	>> Engineering!	>> Data labeling!

Another approach:
Reference modeling and anomaly detection

A) Find a good model for
the reference (error-free) data.

B) Detect deviations to the model!



Machine Learning paradigms and applications

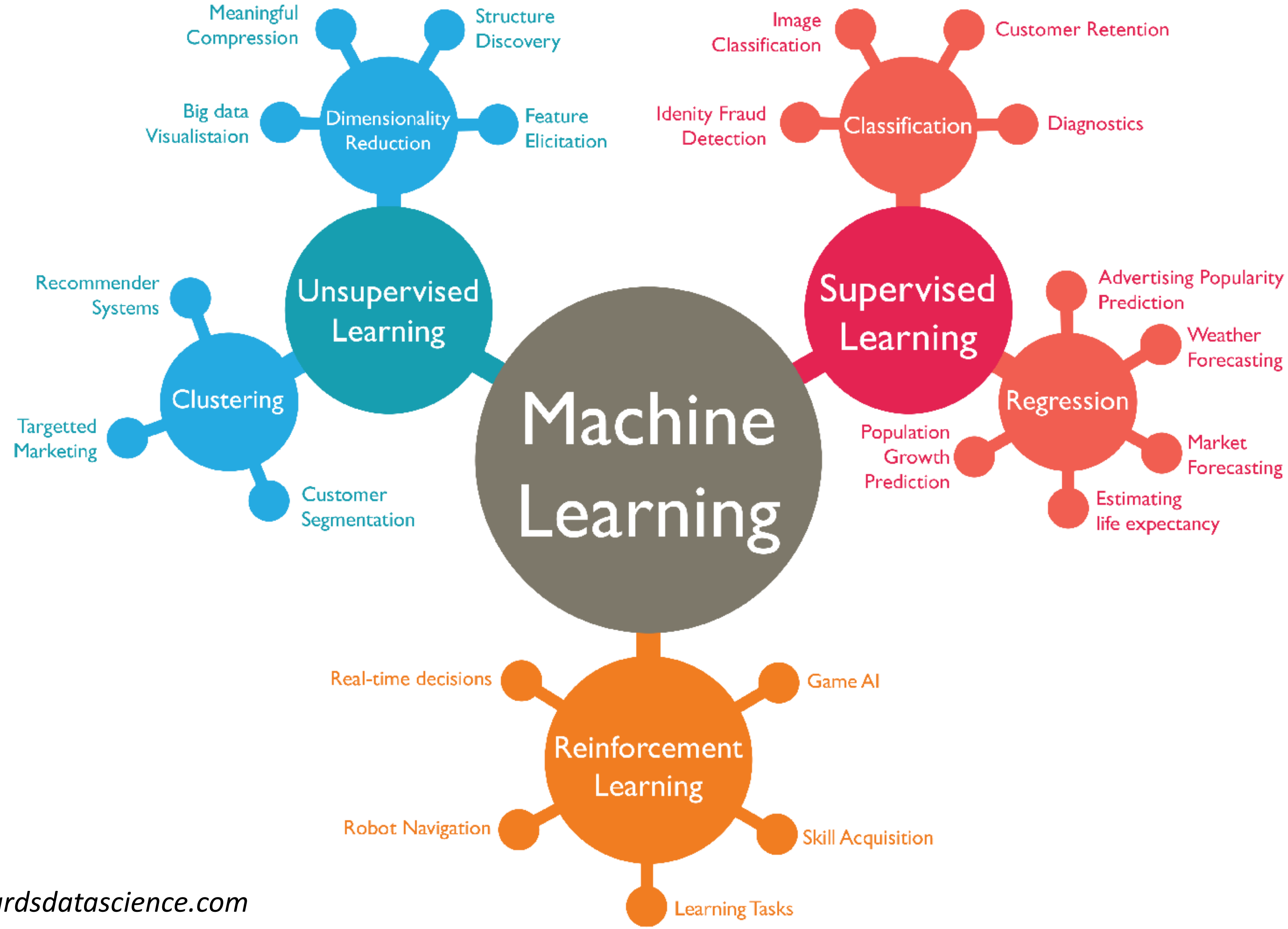


Figure adapted from towardsdatascience.com

Machine Learning paradigms and applications

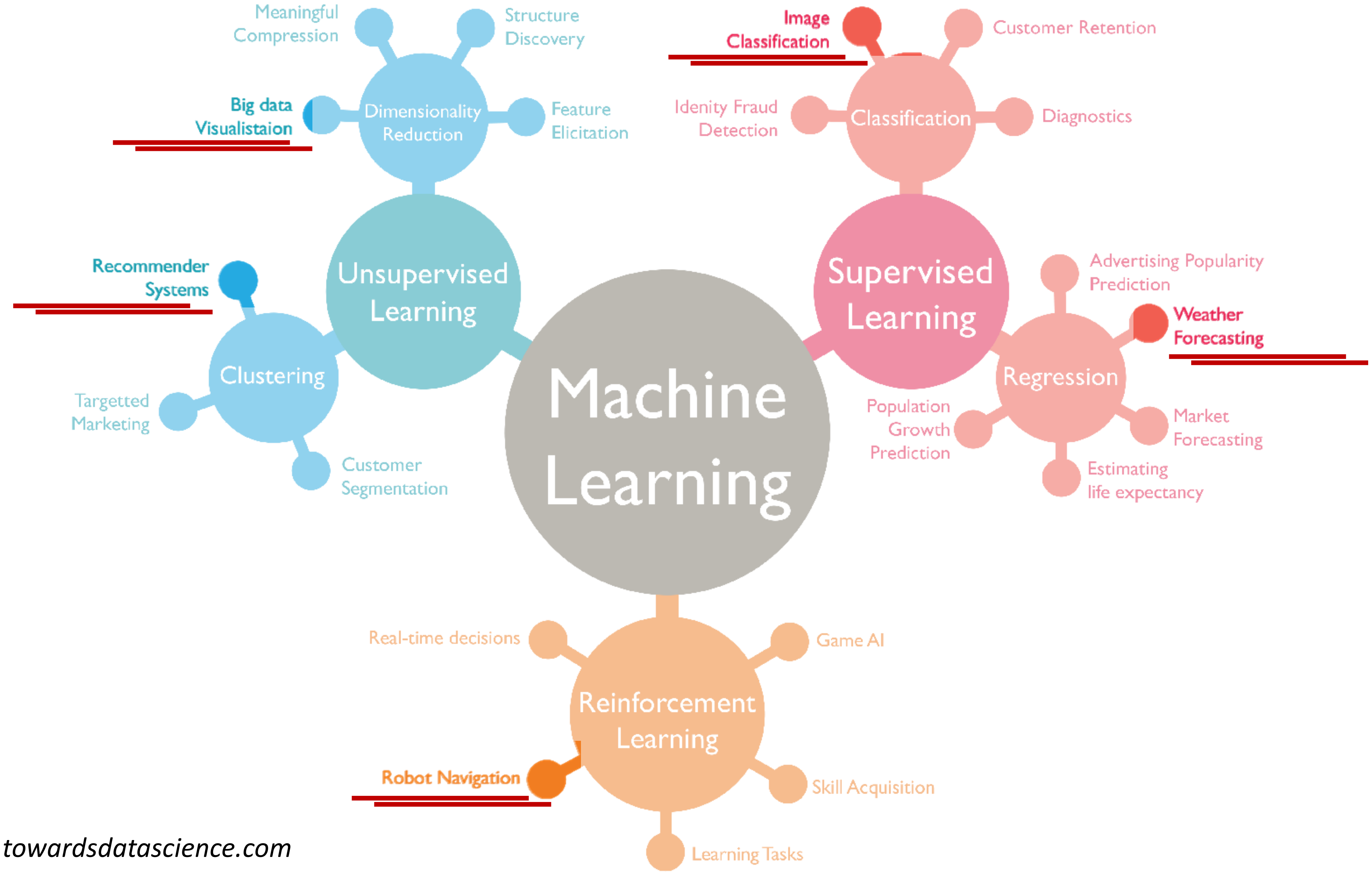


Figure adapted from towardsdatascience.com

What is data and what is a „model“?

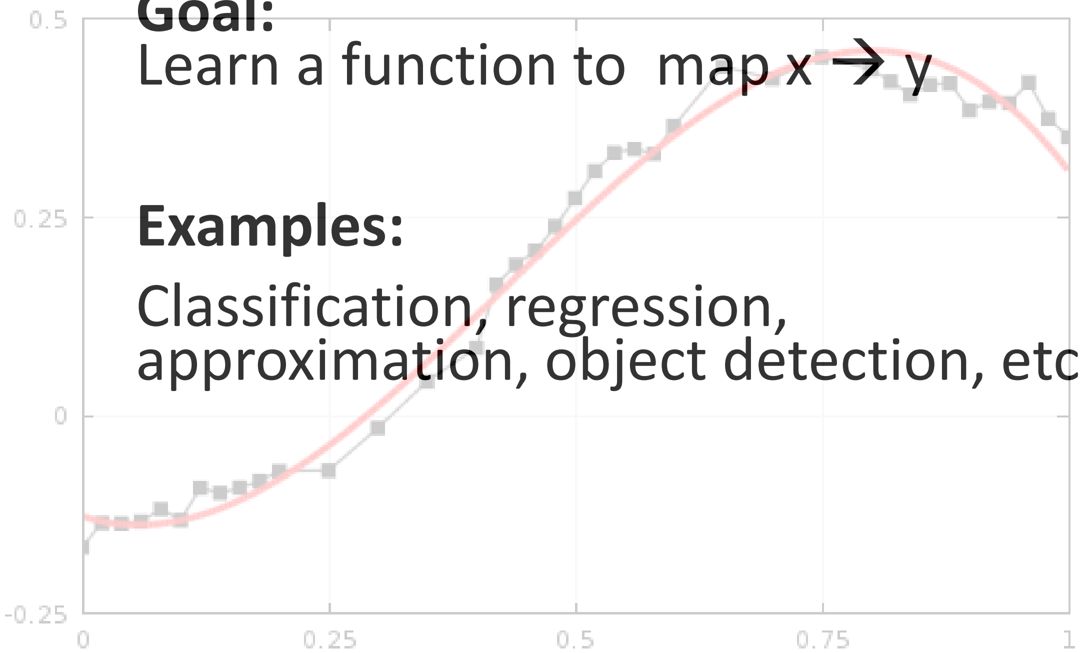
Supervised Models

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x → y

Examples:

Classification, regression, approximation, object detection, etc.



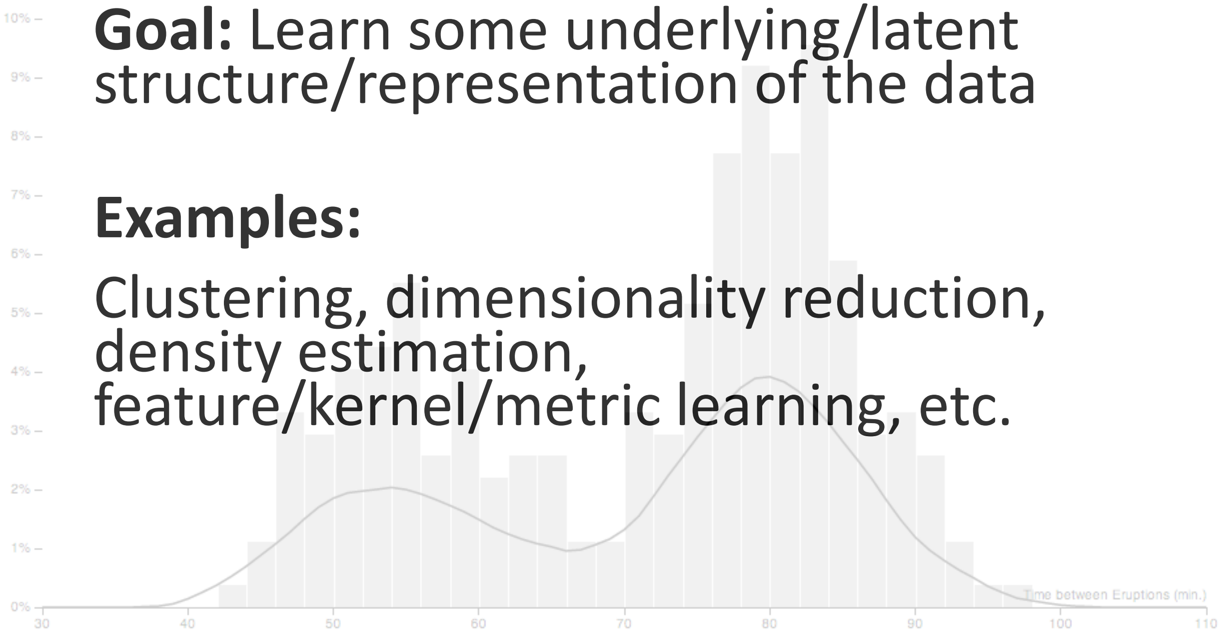
Unsupervised Models

Data: x
Just data, no labels!

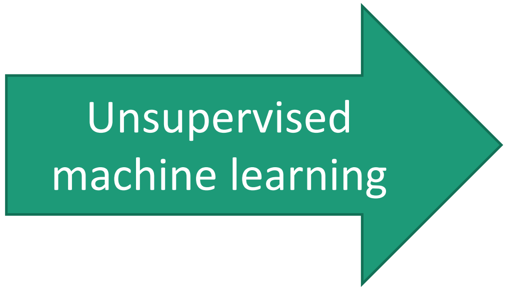
Goal: Learn some underlying/latent structure/representation of the data

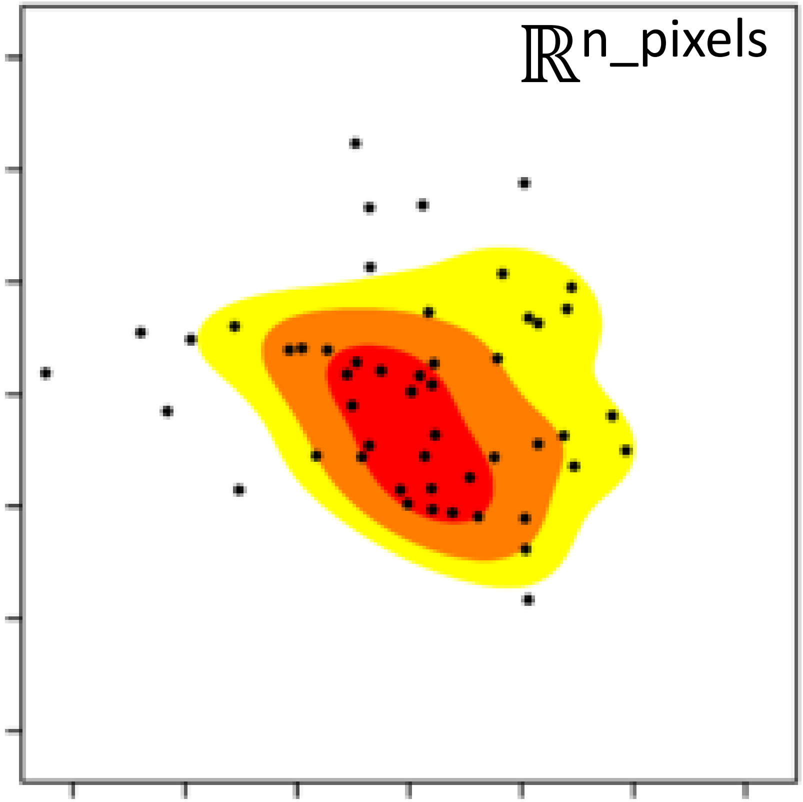
Examples:

Clustering, dimensionality reduction, density estimation, feature/kernel/metric learning, etc.



Represent data as high dimensional vectors or (coll.) points





Reference data ~ $p_{data}(x)$

Model data ~ $p_{model}(x)$

Learn a $p_{model}(x)$ similar to $p_{data}(x)$

Example (generative) models and samples

$p_{bedrooms}(x)$

$p_{faces}(x)$

$p_{bags}(x|z)$

Model Zoo in unsupervised machine Learning

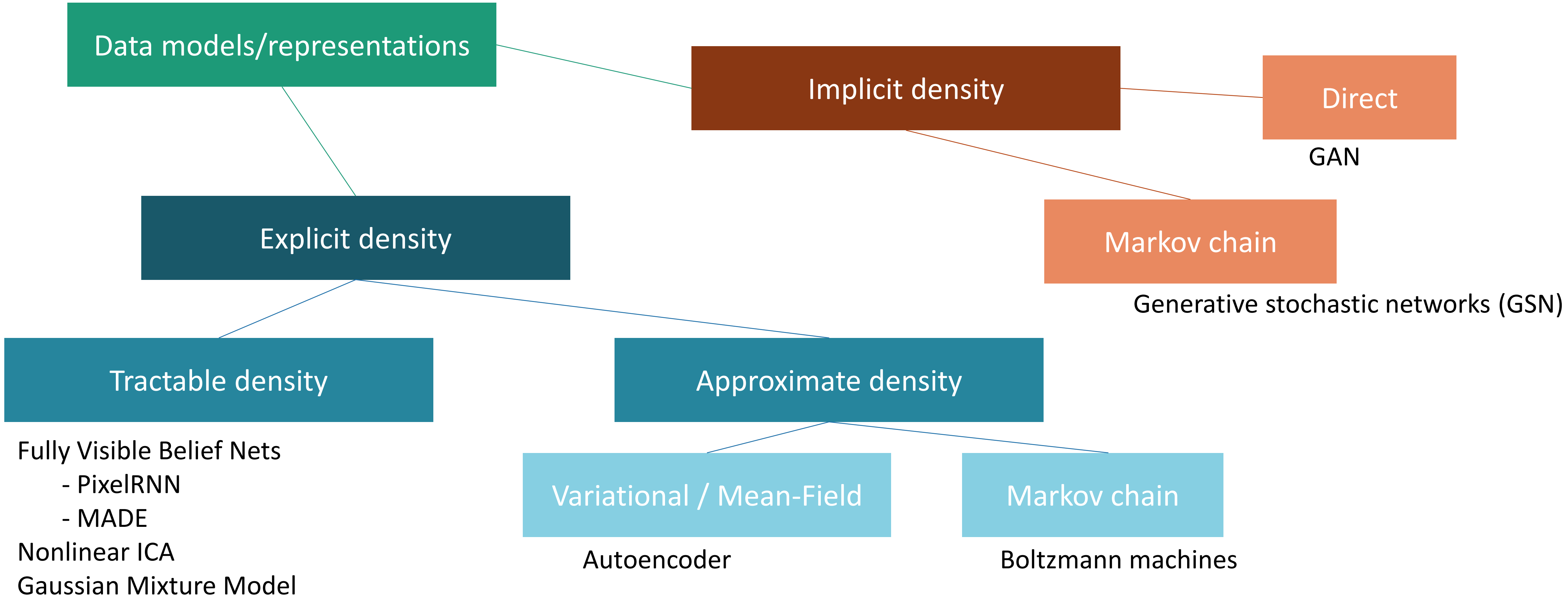


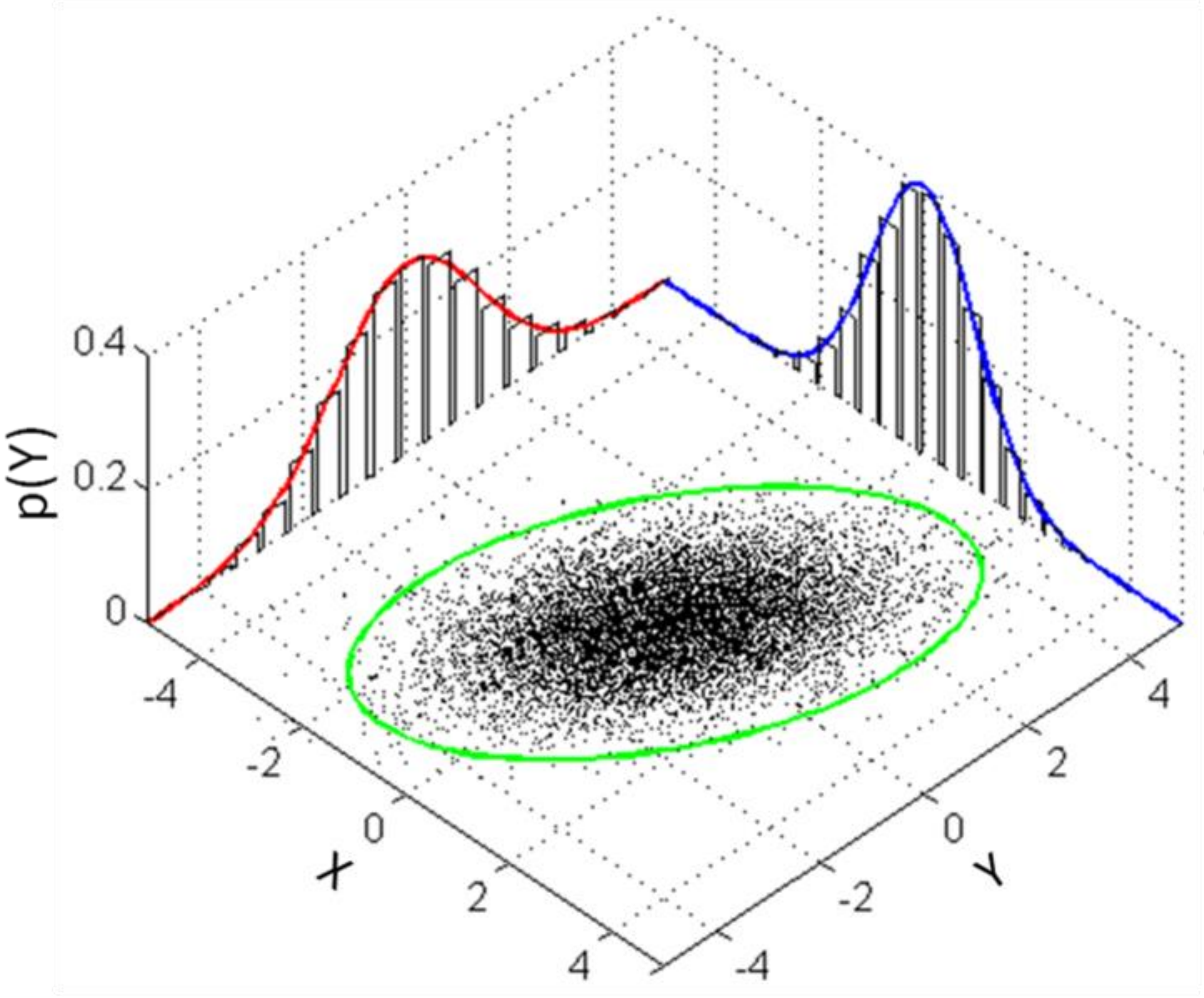
Figure adapted from Ian Goodfellow, Turotial on GANS, 2017

Example Algorithm: Gaussian approximation

$p_{model}(x) = N(\mu, \Sigma)$

a.k.a. multivariate Gaussian.

The green ellipse indicates the isocountour line for the first standard deviation (σ).



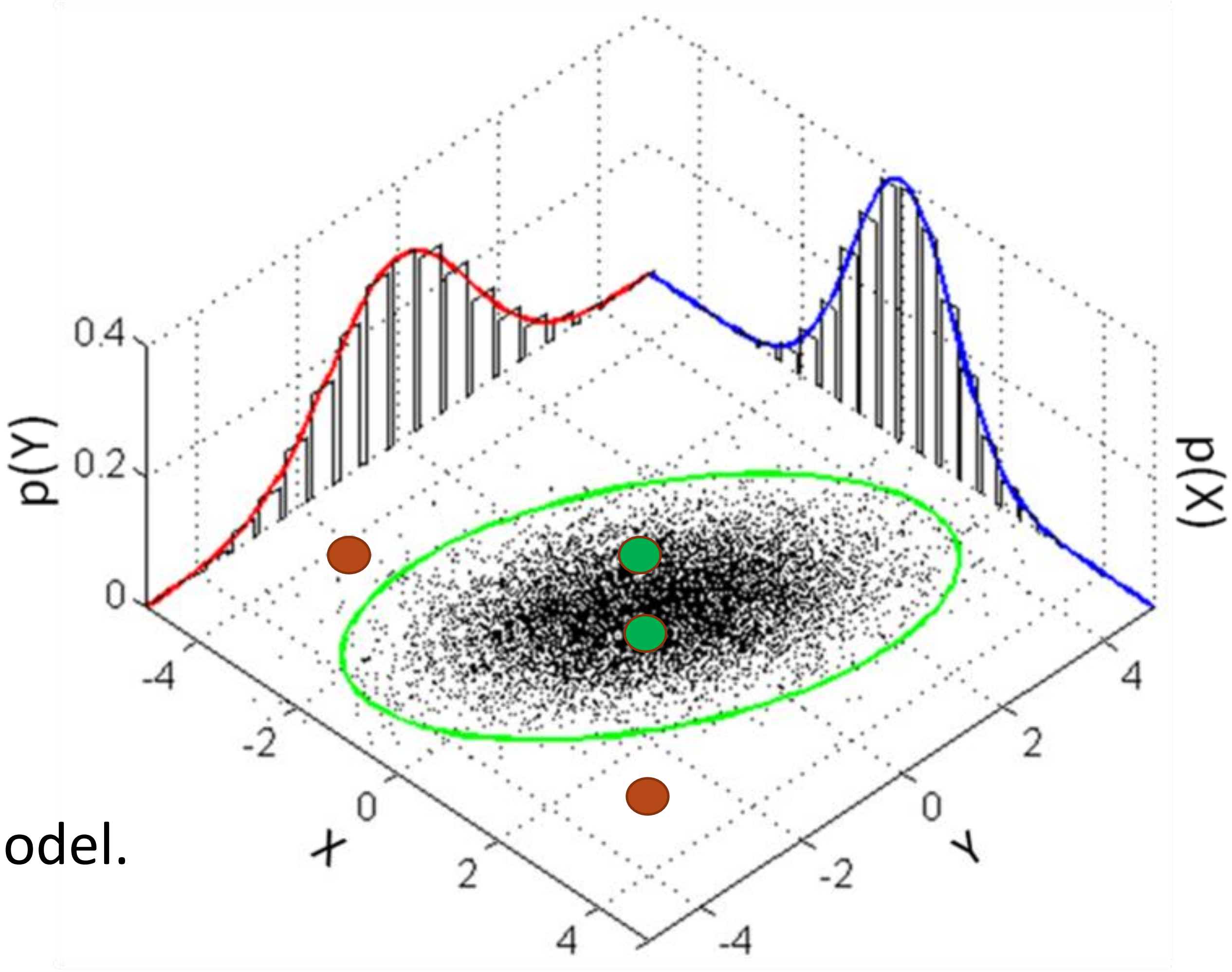
Example Algorithm: Gaussian approximation

$p_{model}(x) = N(\mu, \Sigma)$

Adding new datapoints to the model:

- Red data points are outliers,
- Green data poitns are inliers

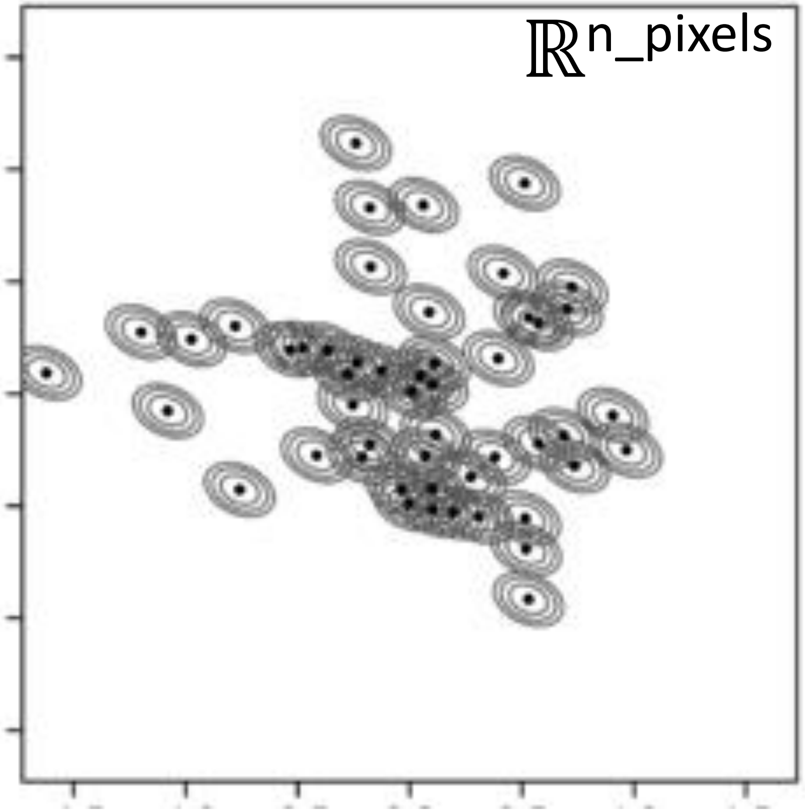
in terms of likelihood under the model.



Example algorithm: k-nearest neigbors

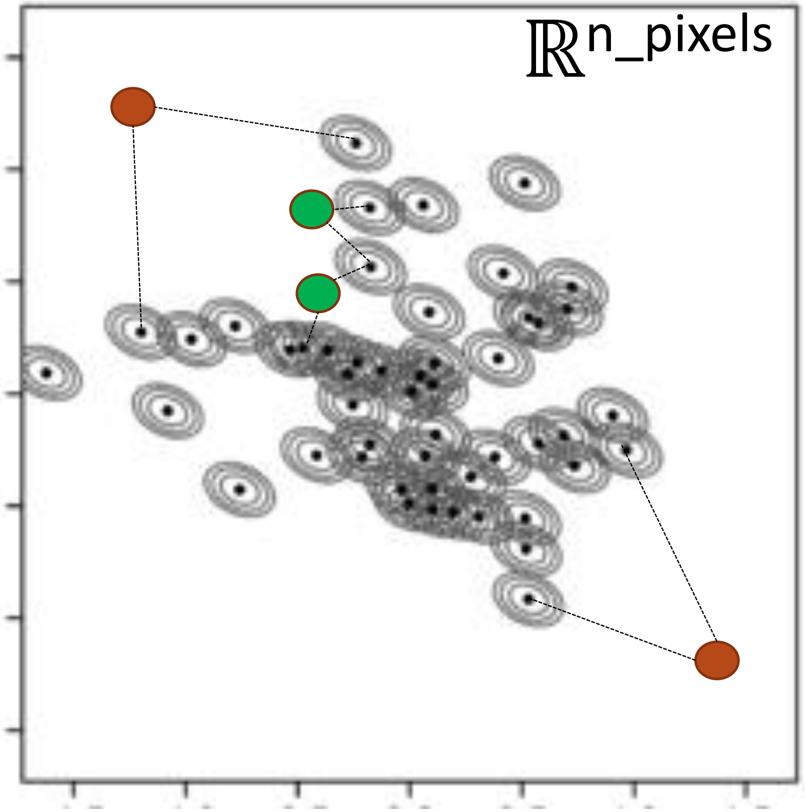
$p_{model}(x) = \frac{1}{K} \sum^{K} N(\mu(x)_n, \Sigma(x)_n)$

Modelling a gaussian distribution around each data point.



Example algorithm: k-nearest neigbors

$$p_{model}(x) = \frac{1}{K} \sum^{K} N(\mu(x)_n, \Sigma(x)_n)$$



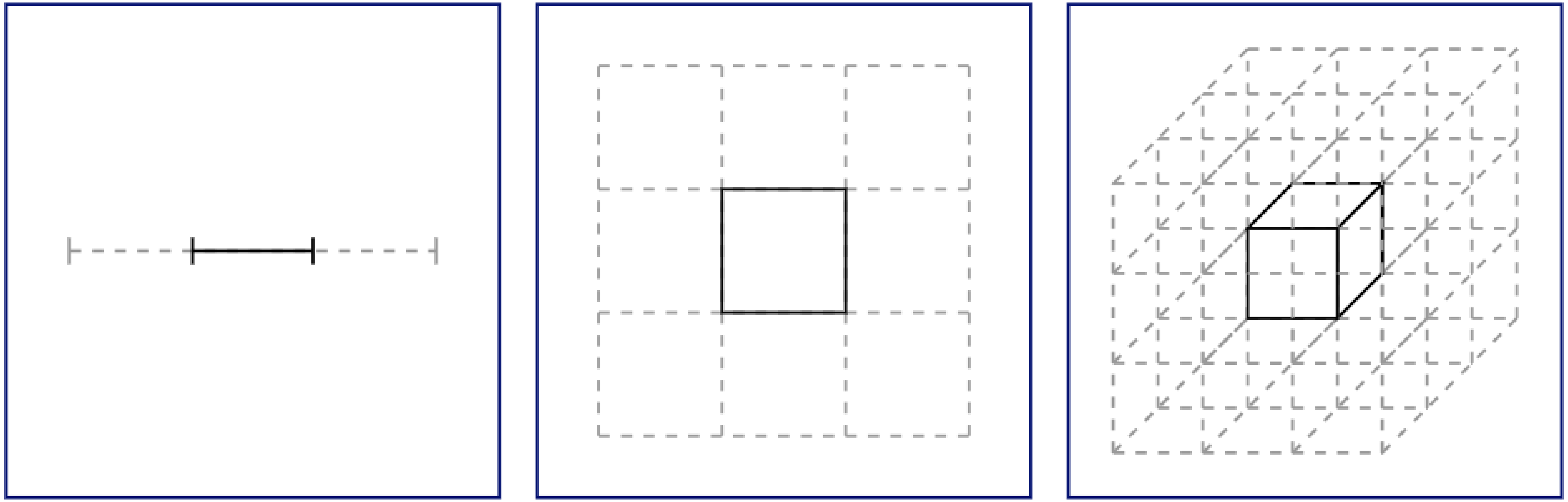
Adding new datapoints to the model:

- Red data points are outliers,
- Green data points are inliers

in terms of average distance (e.g. likelihood) to K neighbors.

But, major issues with high dimensional data

Relative **weight of center partition** decreases with higher dimensions.

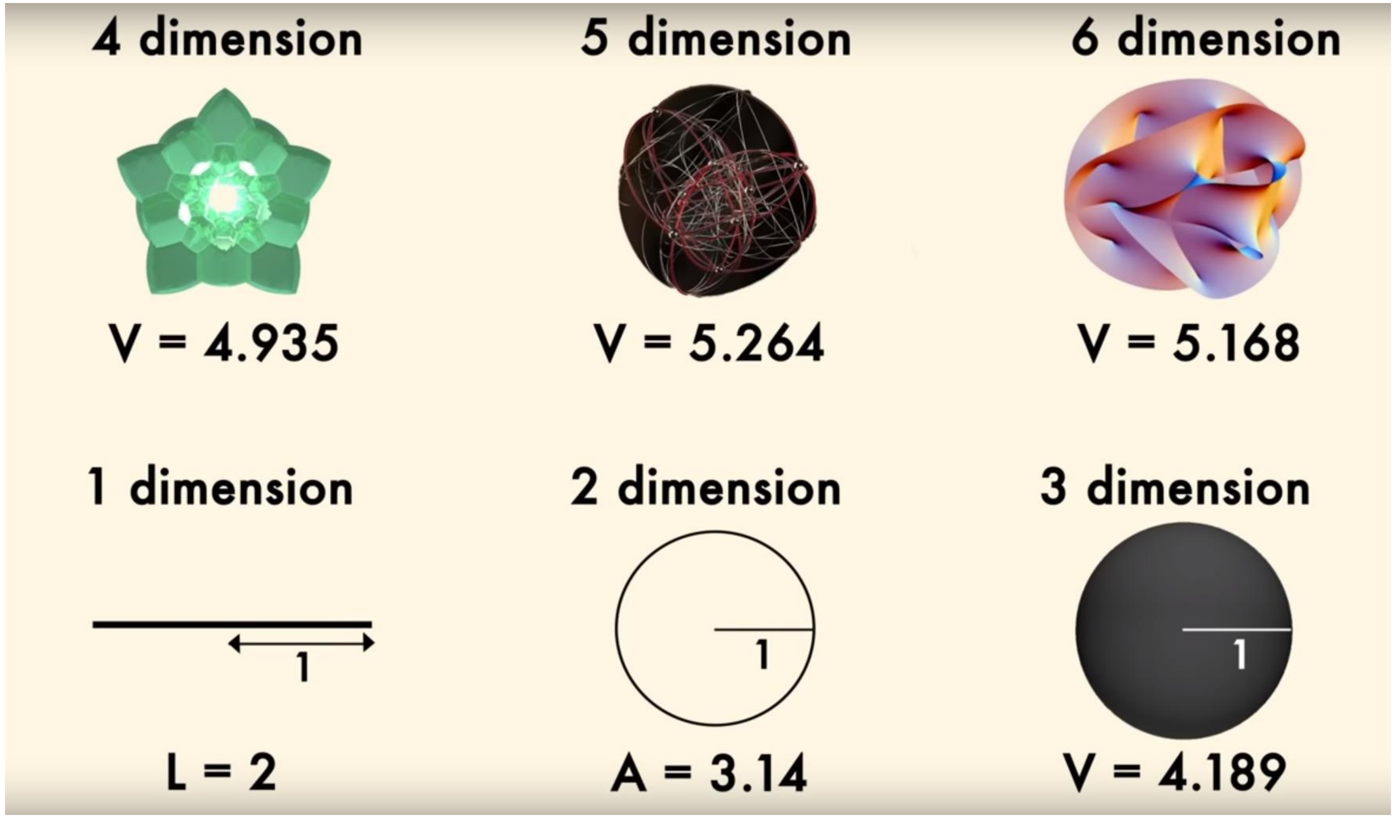


1/3 1/9 1/27

>> In high dimensional euclidean spaces a sphere has almost all of its volume on the surface.

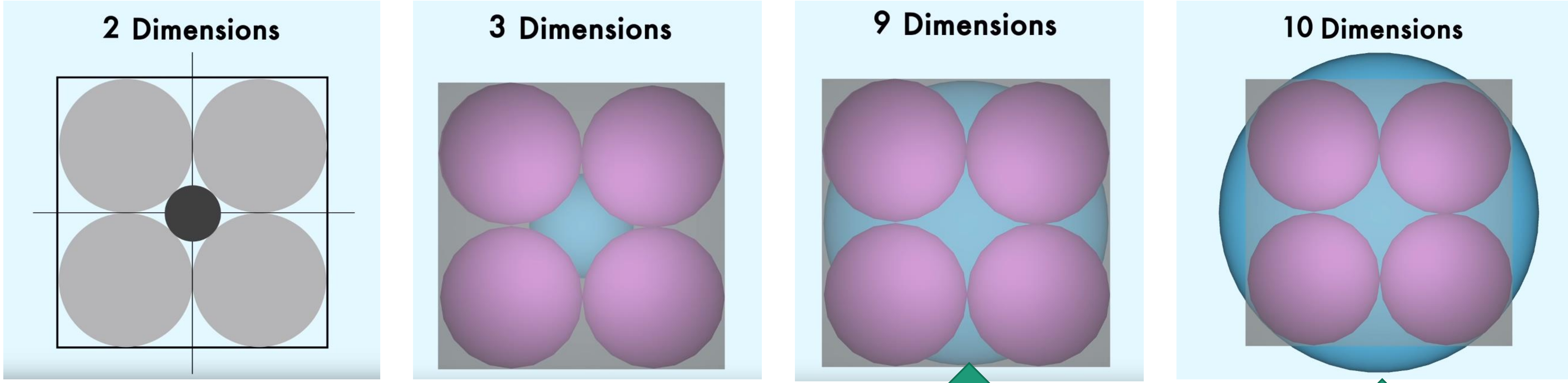
>> **Curse of dimensionality!** Masuring distances in higher dimensions does not work as expected (Concentration of measure principle)!

Volume measures depend on Dimension n



$$V_n(R) = \frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}+1\right)} R^n$$

Diameters have surprising effects, too



For dimensions < 262: $V_{sphere} < V_{cube}$

The enclosed sphere touches the unit-box…

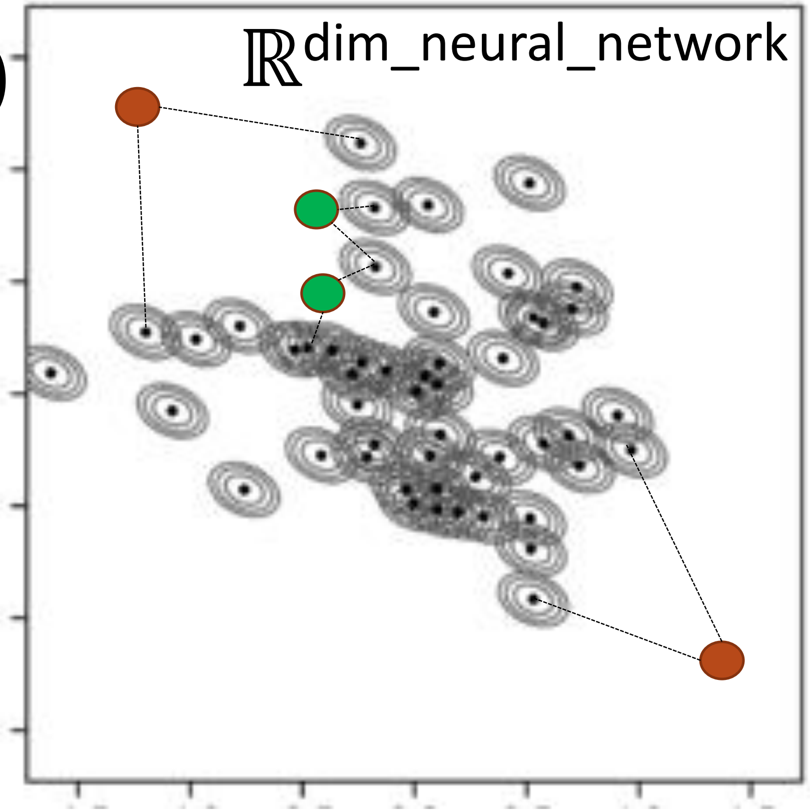
..and even breaks through!

Example algorithm: Using neural networks

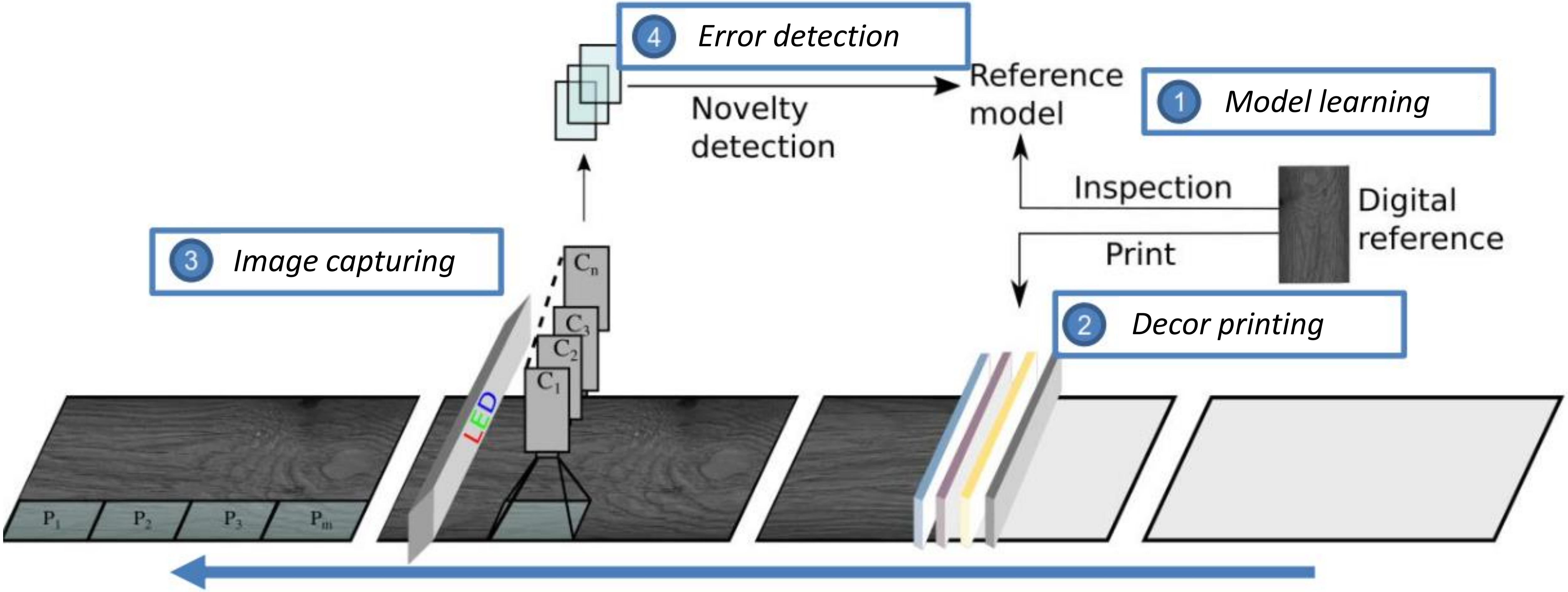
$$p_{model}(x) = \frac{1}{K} \sum^{K} N(\varphi_{\boldsymbol{\theta}}(x), \varphi_{\boldsymbol{\theta}}(x))$$

$$\textit{with neural network } \varphi(x).$$
$$E.g. \varphi(x) = \tanh(Wx + b).$$

- Neural networks may give better non-linear representations for data to fix dimensionality issues.
- Neural networks are parametric models that can handle 100k+ data samples.

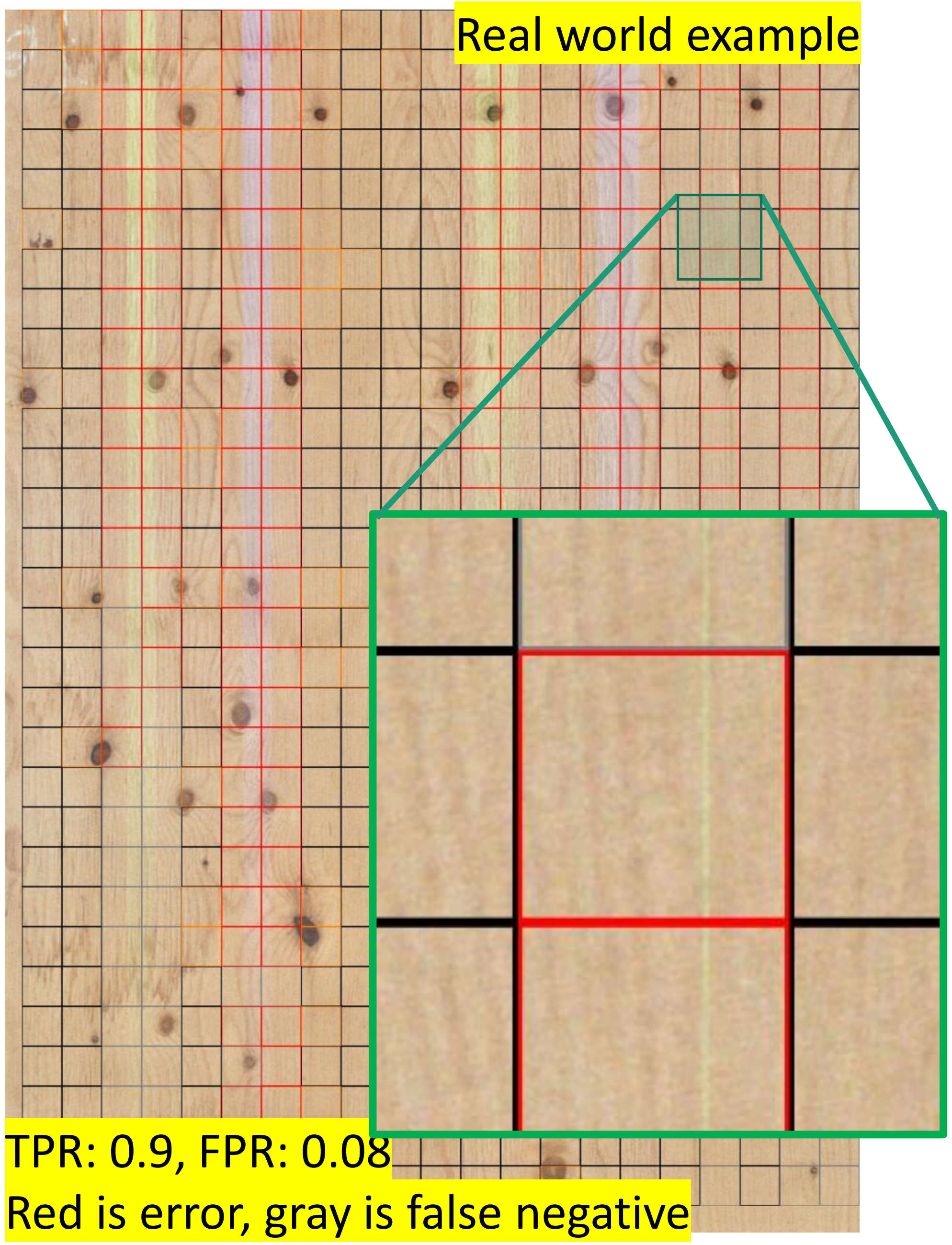


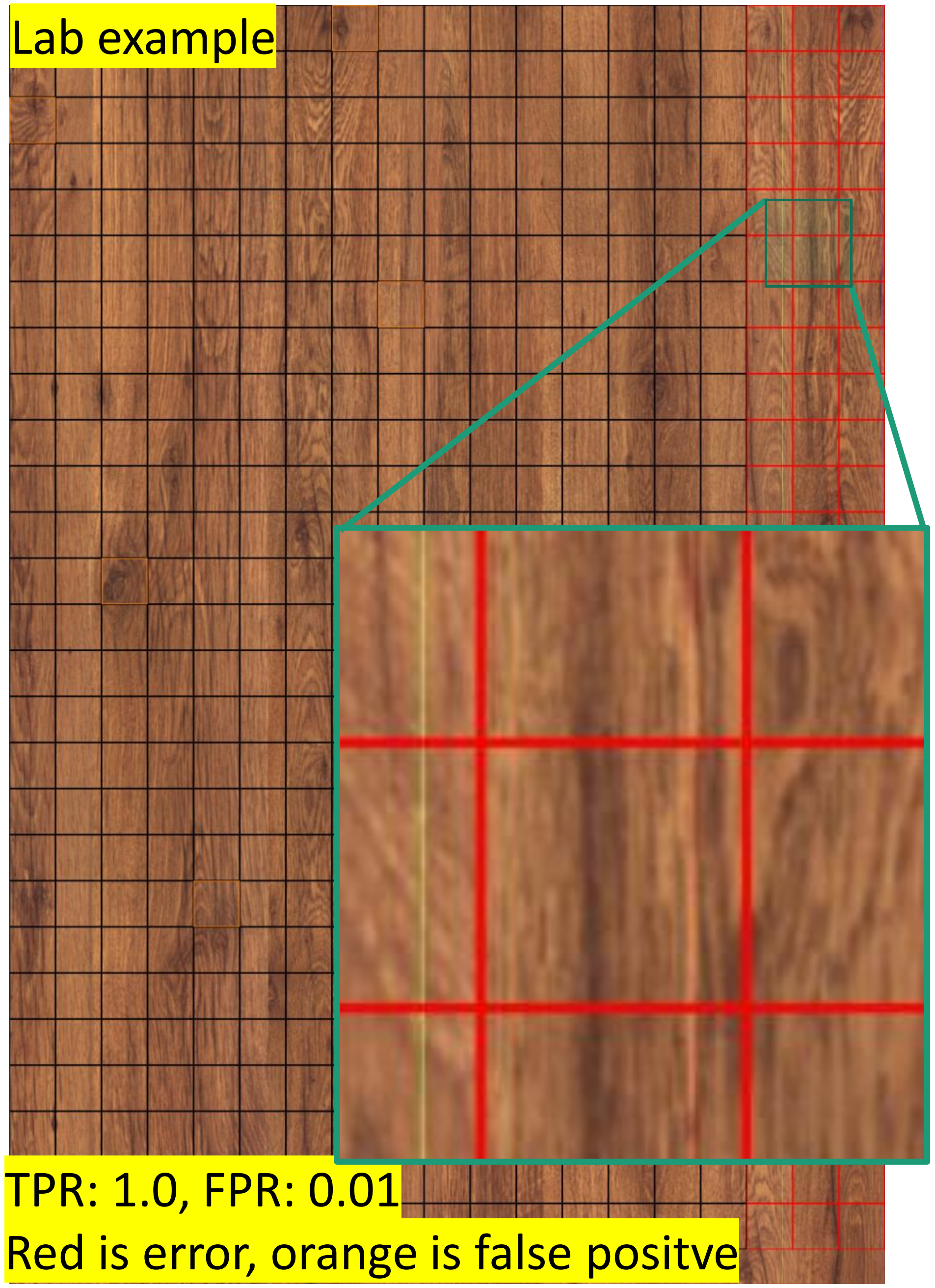
System design



Sampling rate: 80 kHz (14 GBit/s per camera, 144 GBit/s parallelized data rate)

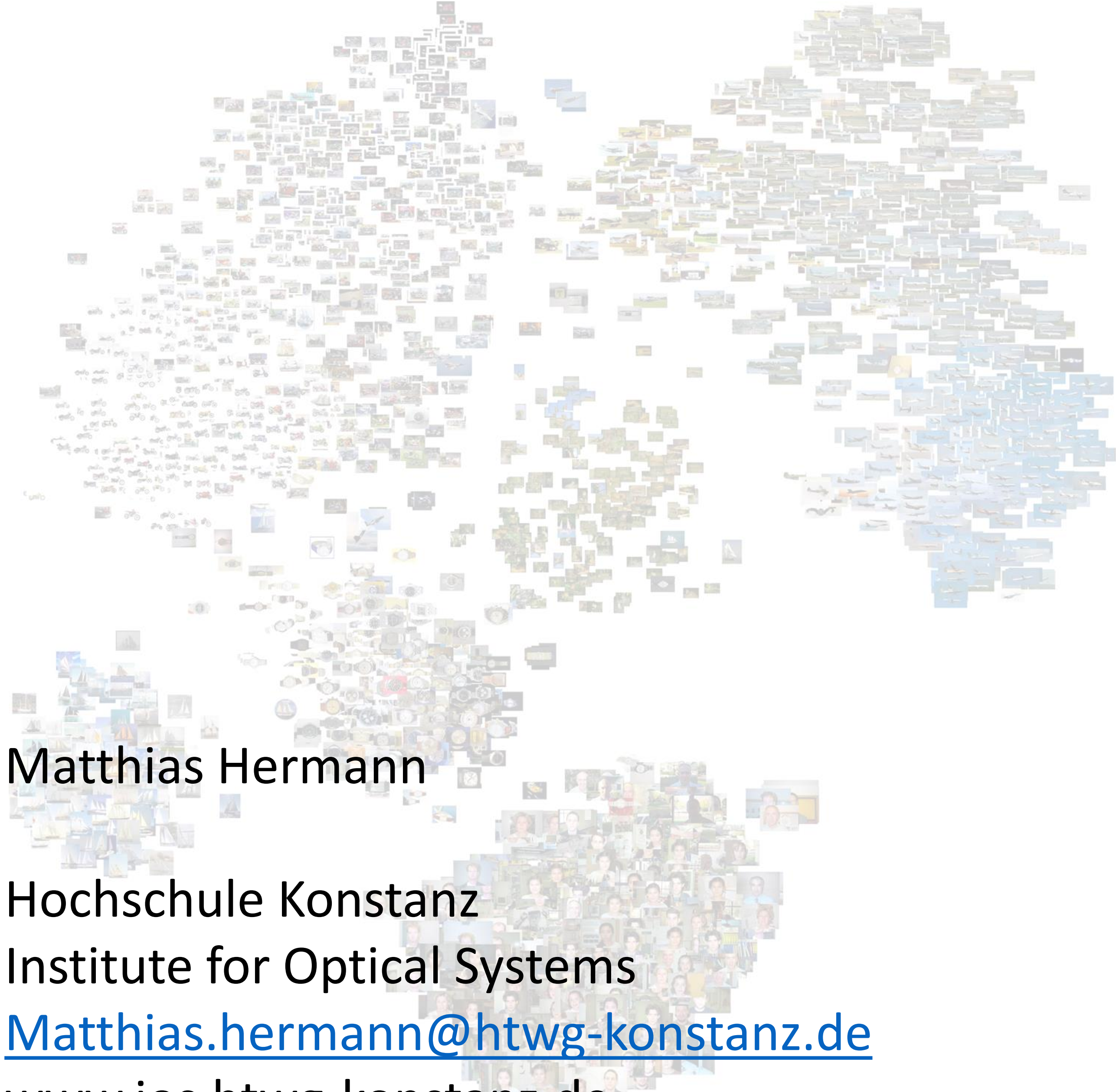
System performance





Next steps

- Collecting more validation real world data.
- Integrating model into FPGA.
- Extending detection model to more experimental methods.
- Extending data processing model to unstructured geometric data.



Thanks for your attention!

Matthias Hermann

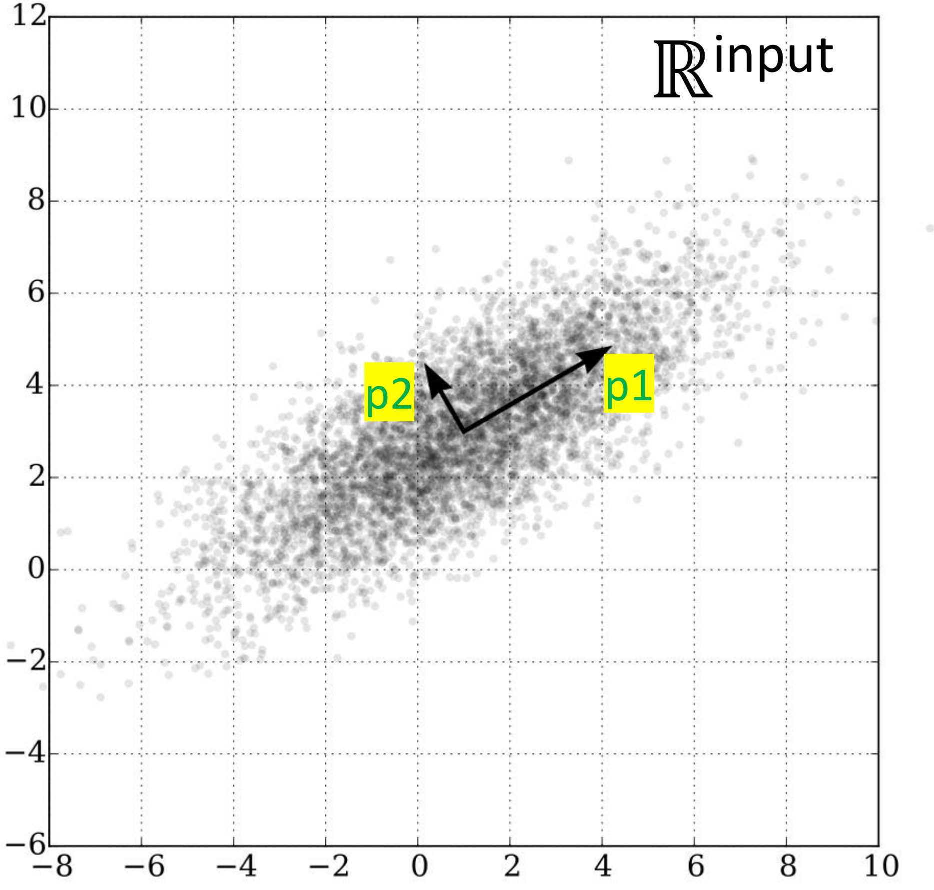
Hochschule Konstanz
Institute for Optical Systems
Matthias.hermann@htwg-konstanz.de
www.ios.htwg-konstanz.de

INSTITUT FÜR OPTISCHE SYSTEME

Literature

- *H. Bay; T. Tuytelaars & L. van Gool (2006). "SURF: Speeded Up Robust Features". Proceedings of the 9th European Conference on Computer Vision, Springer LNCS volume 3951, part 1. pp. 404–417.*
- Kim, M. S., Chen, Y. R., Cho, B. K., Chao, K., Yang, C. C., Lefcourt, A. M., & Chan, D. (2007). Hyperspectral reflectance and fluorescence line-scan imaging for online defect and fecal contamination inspection of apples. *Sensing and Instrumentation for Food Quality and Safety*, *1*(3), 151.
- Tsai, I. S., & Hu, M. C. (1996). Automatic inspection of fabric defects using an artificial neural network technique. *Textile Research Journal*, *66*(7), 474-482.
- Xie, X. (2008). A review of recent advances in surface defect detection using texture analysis techniques. *ELCVIA: electronic letters on computer vision and image analysis*, *7*(3), 1-22.
- Aiger, D., & Talbot, H. (2012). The phase only transform for unsupervised surface defect detection. In *Emerging Topics In Computer Vision And Its Applications* (pp. 215-232).
- Zimek, A., Schubert, E., & Kriegel, H. P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *5*(5), 363-387.
- Marimont, R.B.; Shapiro, M.B. (1979). "Nearest Neighbour Searches and the Curse of Dimensionality". *IMA J Appl Math*. **24**(1): 59–70. doi:10.1093/imamat/24.1.59.
- Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001, January). On the surprising behavior of distance metrics in high dimensional space. In International conference on database theory (pp. 420-434). Springer, Berlin, Heidelberg.
- Raj, B. (2017). Introduction to Deep Learning. *Carnegie Mellon's School of Computer Science.*

Recap PCA (Principal Component Analysis)



$$w_1 = argmax_{||w||=1} \left|\left|Xw\right|\right|^2 = w^T X^T X w$$

$$w_2 = argmax_{||w||=1} \left|\left|(X - X w_1 w_1^T) w\right|\right|^2$$

$$w_3 = \ldots$$

- Yields Transformation $T = XW$ $with$ W $p \times p$ and T $n \times p$
- Maximizing variance of the projected data X

Wikipedia.org

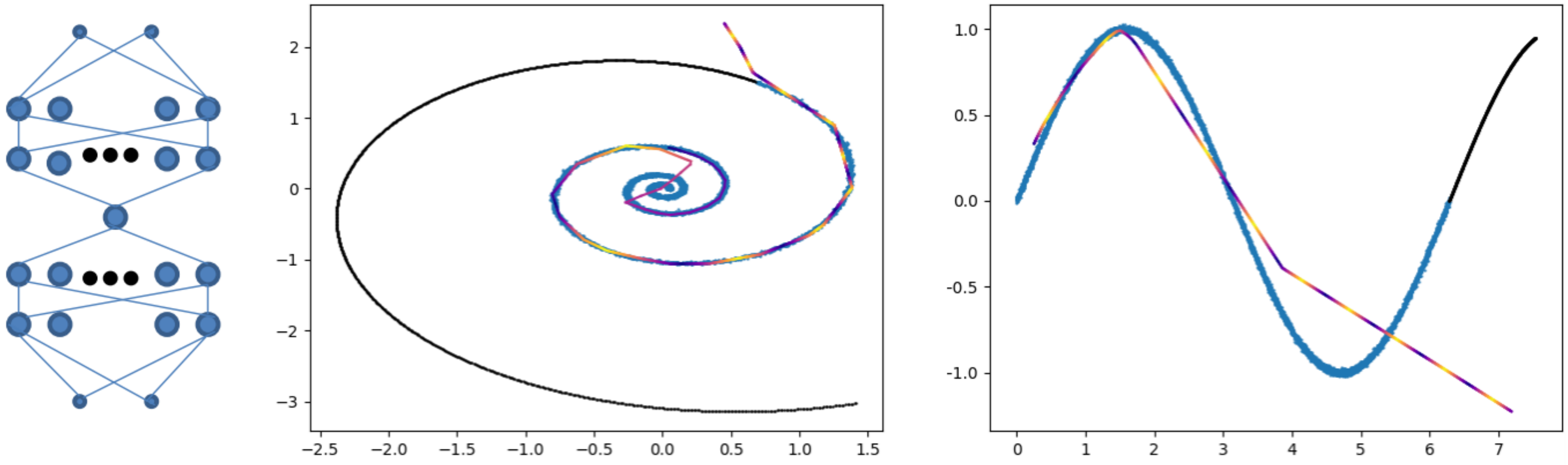
Recap PCA (Limitations I)

- No probabilistic model for observed data.
- Computation-intensive variance-covariance matrix needs to be calculated.
- Does not work properly for outlying data and incomplete data.

Bishop, 1999

Recap PCA (Limitations II)

- PCA (and autoencoders) are a discriminative models:
  - Varying hidden layer value z only generates data along the learned manifold
  - Any input will result in an output along learned manifold



Raj, 2017

Probabilistic PCA (Motivation)

- Maximum-likelihood estimates can be computed for elements associated with principle components.
- Conventional PCA will assign low reconstruction cost to data points that are close to the principal subspace even if they lie far away from training data.
- Addresses limitations of regular PCA (and autoencoders).

But much more important!

- Autoencoders can be viewed as non-linear PCA
- Variational autoencoders can be viewed as non-linear PPCA (special case of Factor Analysis)

Bishop, 2006

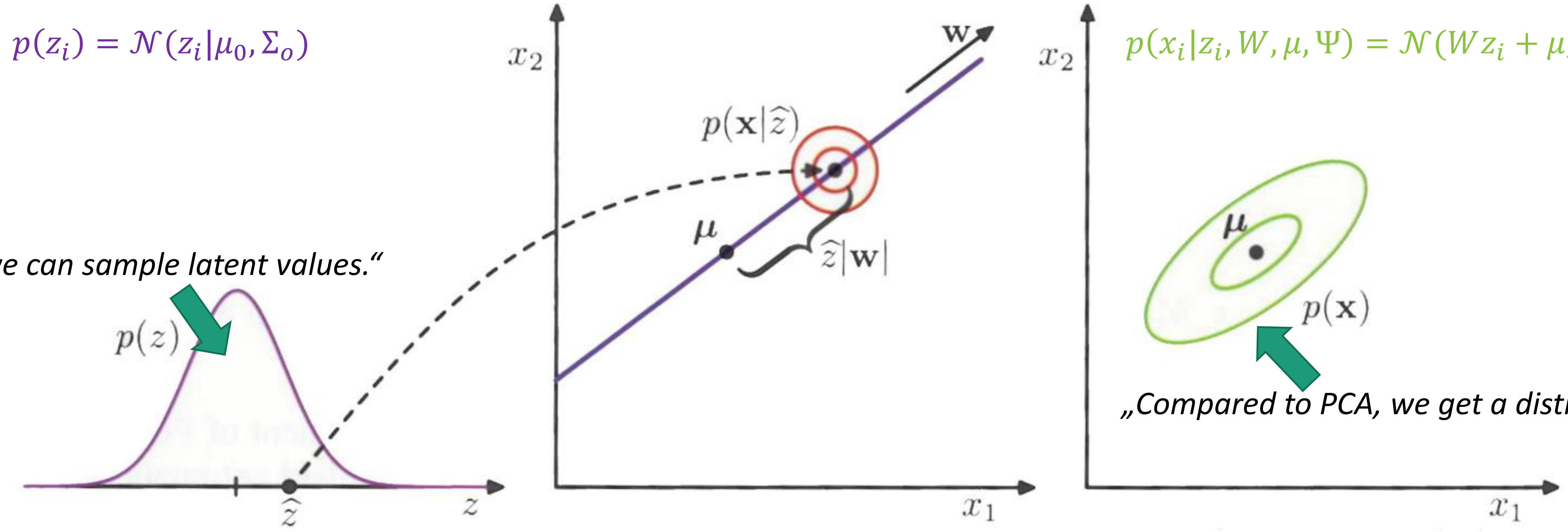
Probabilistic PCA (Model)

- Generative model: Assumes that data are generated from real values.

$p(z_i) = \mathcal{N}(z_i|\mu_0, \Sigma_o)$

$p(x_i|z_i, W, \mu, \Psi) = \mathcal{N}(Wz_i + \mu, \Psi)$

„No we can sample latent values.“



„Compared to PCA, we get a distributio

Where $\boldsymbol{W} \epsilon \mathbb{R}^{D\times L}, \boldsymbol{\Psi} \epsilon \mathbb{R}^{D\times D}, \boldsymbol{\Psi}$ is diagonal, in PPCA further $\sigma^2\mathbf{I}$

Bishop, 2006

Typical way (Marginal distribution of observed x_i)

$$p(x_i|W,\mu,\Psi) = \int \mathcal{N}(Wz_i + y, \Psi)\mathcal{N}(z_i|\mu_o, \Sigma_0)dz_i$$

Find: $p\left(x_i|\widehat{W},\hat{\mu},\Psi\right) = \mathcal{N}(x_i|\hat{\mu},\Psi + \widehat{W}\widehat{W}^T)$

$where\ \hat{\mu} = W\mu_0 + \mu\ and\ \widehat{W} = W\Sigma_0^{\frac{1}{2}}\ and\ let\ p(z_i) = \mathcal{N}(z_i|0,I)$

Solution path:

- ELBO!
- Proof ELBO is tight!
- EM-Algorithm!

Bishop, 2006

Another Perspective (Log-Likelihood)

Consider the log-likelihood of the marginal distribution with latent variable z and model parameters Θ:

$$\ell(\theta) = \sum_{i=1}^{N} \log p(x_i|\theta) = \sum_{i=1}^{N} \log \int p(x_i, z_i|\theta)\, dz_i$$

Problem: We have a log outside of the integral which would cause inefficient integration per datapoint.

Bishop, 2006

Another Perspective (Expected Log-Likelihood)

Better would be observing z_i:

$$\ell(\theta) = \sum_{i=1}^{N} \log p(x_i, z_i|\theta) \qquad (complete\ log - likelihood)$$

Take Expectation!

$$\mathbb{E}_{q(z)}[\ell(\theta)] = \int q(z_i)\ell(\theta)dz_i = \sum_{i=1}^{N} \int q(z_i) \log p(x_i, z_i|\theta)dz_i$$

- Finding the q that maximizes this is the E step of EM
- Finding the Θ that maximizes this is the M step of EM

Bishop, 2006

Approach (Expected Log-Likelihood)

$$argmax_\theta \mathbb{E}_{q(z)}[\log p(X, Z|\theta)]$$
$$= argmax_\theta \mathbb{E}_{q(z)}[\log p(X|Z, \theta)] + \mathbb{E}_{q(z)}[\log p(z)]$$

„Does not depend on Θ"

1) Compute optimal q-values (i.e. „Project X into Z-space")
2) Sample „optimal q-values"
3) Optimize for Θ

➔ *Is still, EM-Algorithm*

Bishop, 2006; Raj, 2017

Approach (Expected Log-Likelihood)

$$\begin{aligned} & argmax_\theta \mathbb{E}_{q(z)}[\log p(X,Z|\theta)] \\ = \; & argmax_\theta \mathbb{E}_{q(z)}[\log p(X|Z,\theta)] + \mathbb{E}_{q(z)}[\log p(z)] \end{aligned}$$

„Does not depend on Θ“

1) Compute optimal q-values (i.e. „Project X into Z-space“)
2) Sample „optimal q-values“
3) Optimize for Θ

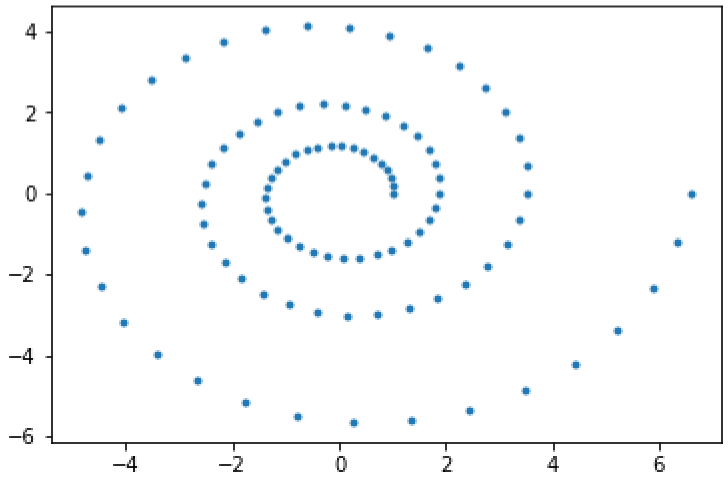
➔ *Is still, EM-Algorithm*

Final derivation for completeness:

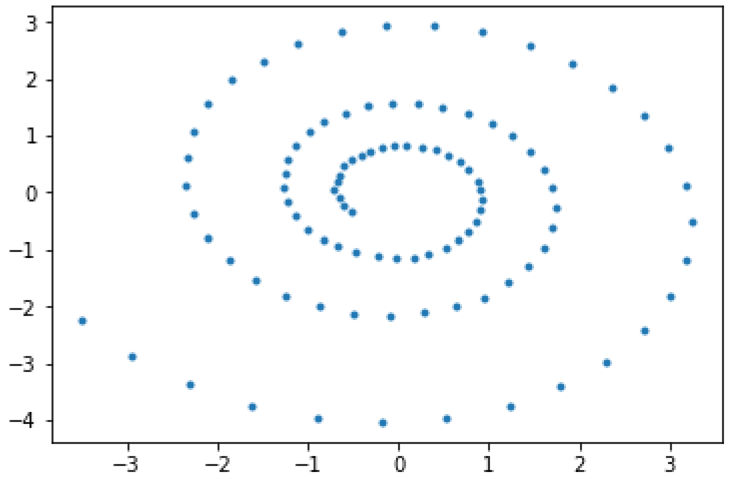
$$= \operatorname{argmax}_{\boldsymbol{W},\boldsymbol{\Psi}} - \frac{N}{2}\log\det(\boldsymbol{\Psi}) - \sum_{i=1}^{N}\left(\frac{1}{2}\boldsymbol{x}_i^T\boldsymbol{\Psi}^{-1}\boldsymbol{x}_i - \boldsymbol{x}_i^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\mathbb{E}_{q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i] + \frac{1}{2}\operatorname{tr}\left(\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\mathbb{E}_{q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^T]\right)\right)$$

Bishop, 2006; Raj, 2017

Example (1 dimensional latent space)



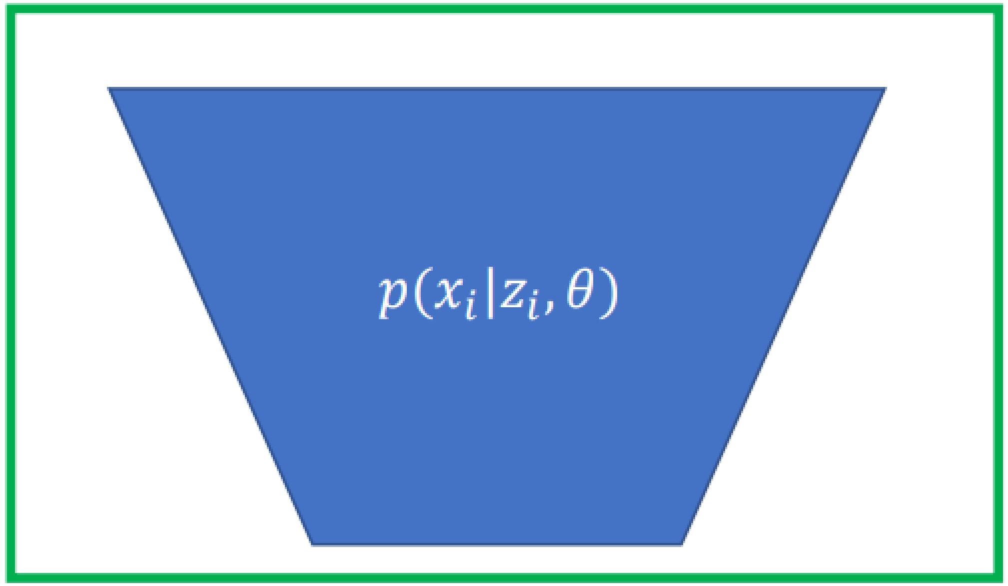
Training data



Samples from the model $x_i \sim N(Wz + \mu, 0)$

The plot recovers (up to rotation) the original 2D data quite well.

Further reading…

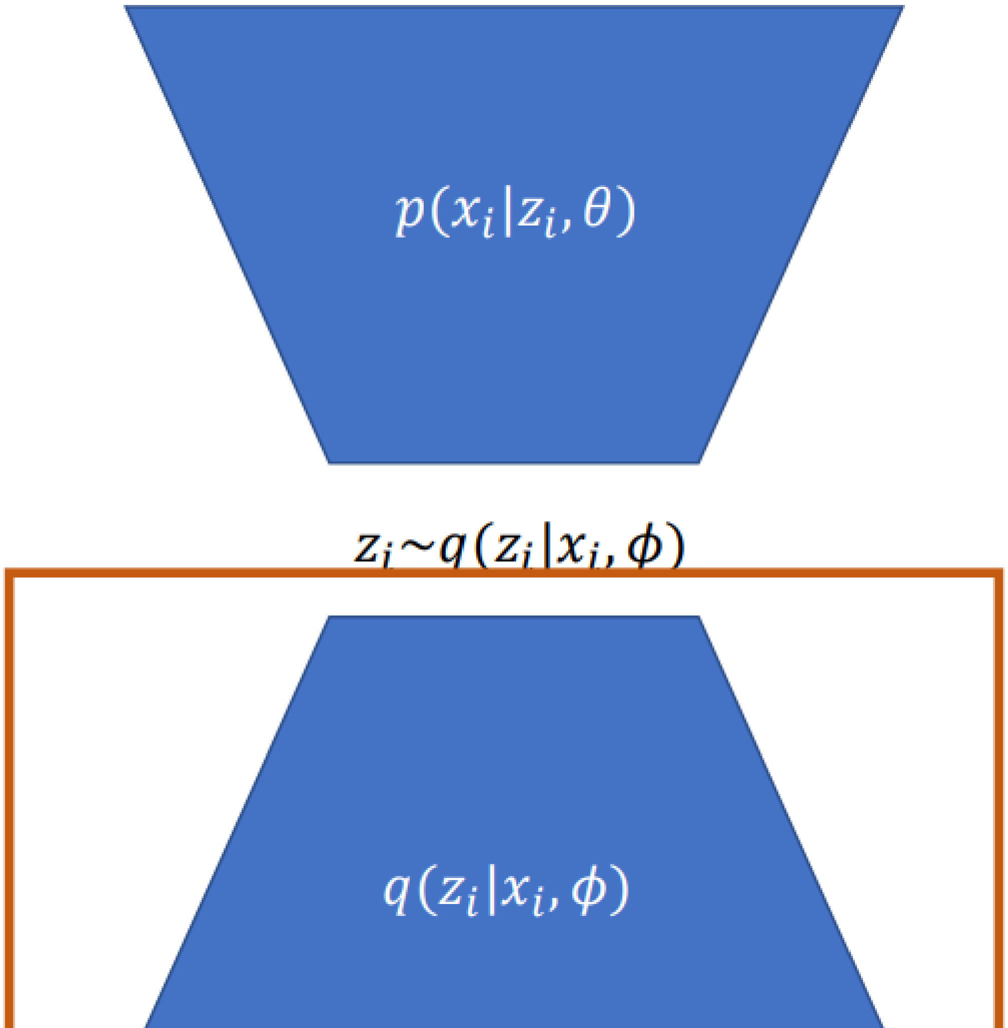


$p(z_i)$

a) Assume a generative model with a latent variable distributed according to some distribution $p(z_i)$

b) The observed variable is distributed according to a conditional distribution $p(x_i \mid z_i , \theta)$

Further reading…



a) We also create a weighting distribution $q(z_i \mid x_i, \phi)$

b) This will play the same role as $q(z_i)$ in the EM algorithm, as we will see.

c) But no it depends on the input x_i

→ *See you next time, Variational Autoencoder*