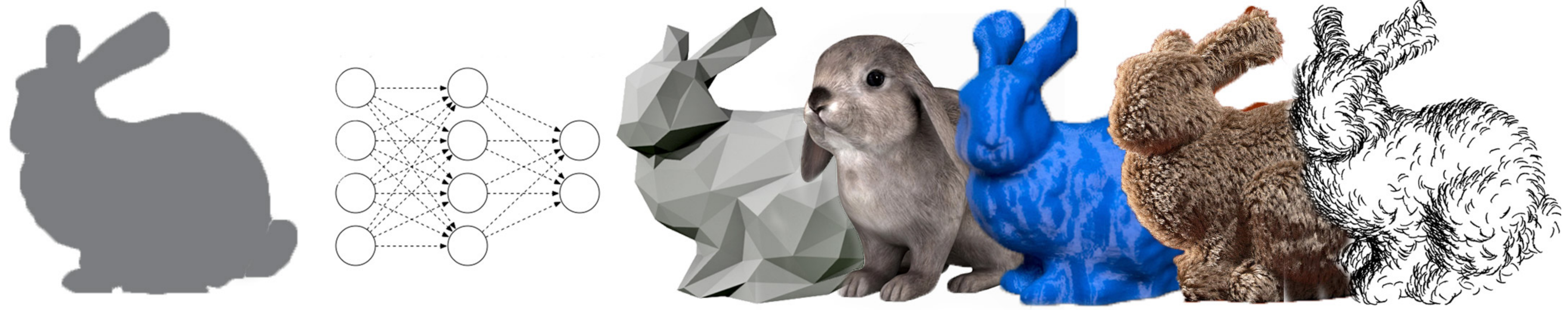


Deep Learning for Computer Graphics and Geometry Processing

Niloy Mitra — UCL

Iasonas Kokkinos — UCL

Federico Monti — USI Lugano

Emanuele Rodolà — La Sapienza

Michael Bronstein — Imperial College, USI Lugano

Or Litany — Stanford University, Facebook

Leonidas Guibas — Stanford University

http://geometry.cs.ucl.ac.uk/dl_for_CG/

Tutorial Organizers

Niloy Mitra

Tutorial Organizers

Niloy Mitra

Iasonas Kokkinos

Tutorial Organizers

Niloy Mitra

Iasonas Kokkinos

Federico Monti

Tutorial Organizers

Niloy Mitra

Iasonas Kokkinos

Federico Monti

Emanuele Rodolà

Tutorial Organizers

Niloy Mitra

Iasonas Kokkinos

Federico Monti

Emanuele Rodolà

Michael Bronstein

Tutorial Organizers

Niloy Mitra

Iasonas Kokkinos

Federico Monti

Emanuele Rodolà

Michael Bronstein

Or Litany

Tutorial Organizers

Niloy Mitra

Iasonas Kokkinos

Federico Monti

Emanuele Rodolà

Michael Bronstein

Or Litany

Leonidas Guibas

Tutorial Organizers

Niloy Mitra

Iasonas Kokkinos

Federico Monti

Emanuele Rodolà

Michael Bronstein

Or Litany

Leonidas Guibas

Timetable

			Niloy	Federico	Iasonas	Emanuele
Theory/Basics	Introduction	9:00	X	X	X	X
	Machine Learning Basics	~ 9:05	X			
	Neural Network Basics	~ 9:35		X		
	Alternatives to Direct Supervision (GANs)	~11:00			X	
State of the Art	Image Domain	~11:45			X	
	3D Domains (extrinsic)	~13:30	X			
	3D Domains (intrinsic)	~ 14:15				X
	Physics and Animation	~ 16:00	X			
	Discussion	~ 16:45	X	X	X	X

Sessions: A. 9:00-10:30 (**coffee**) B. 11:00-12:30 [**LUNCH**] C. 13:30-15:00 (**coffee**) D. 15:30-17:00

Code Examples

PCA/SVD basis
Linear Regression
Polynomial Regression
Stochastic Gradient Descent vs. Gradient Descent
Multi-layer Perceptron
Edge Filter 'Network'
Convolutional Network
Filter Visualization
Weight Initialization Strategies
Colorization Network
Autoencoder
Variational Autoencoder
Generative Adversarial Network

http://geometry.cs.ucl.ac.uk/dl_for_CG/

Scan me

Course Objectives

Course Objectives

- Provide an overview of the popular **ML algorithms** used in CG

Course Objectives

- Provide an overview of the popular **ML algorithms** used in CG
- Provide a quick overview of **theory** and **CG applications**
  - Many extra slides in the course notes + example code

Course Objectives

- Provide an overview of the popular **ML algorithms** used in CG
- Provide a quick overview of **theory** and **CG applications**
  - Many extra slides in the course notes + example code
- Progress in the last 3-5 years has been dramatic
  - We have organized them to help newcomers
  - Discuss the main **challenges and opportunities** specific to CG

Two-way Communication

Scan me

Two-way Communication

- Our aim is to convey what we found to be relevant so far
- You are invited/encouraged to give feedback

Scan me

Two-way Communication

- Our aim is to convey what we found to be relevant so far

- You are invited/encouraged to give feedback
  - Speakup. Please send us your criticism/comments/suggestions

Scan me

Two-way Communication

- Our aim is to convey what we found to be relevant so far

- You are invited/encouraged to give feedback
  - Speakup. Please send us your criticism/comments/suggestions
  - Ask questions, please!

- **Thanks to many people who helped so far with slides/comments**

Scan me

Representations in Computer Graphics

Representations in Computer Graphics

- Images (e.g., pixel grid)

- Volume (e.g., voxel grid)

Representations in Computer Graphics

- Images (e.g., pixel grid)
- Volume (e.g., voxel grid)
- Meshes (e.g., vertices/edges/faces)

Representations in Computer Graphics

- Images (e.g., pixel grid)
- Volume (e.g., voxel grid)
- Meshes (e.g., vertices/edges/faces)
- Pointclouds (e.g., point arrays)

Representations in Computer Graphics

- Images (e.g., pixel grid)
- Volume (e.g., voxel grid)
- Meshes (e.g., vertices/edges/faces)
- Pointclouds (e.g., point arrays)
- Animation (e.g., skeletal positions over time; cloth dynamics over time)

Representations in Computer Graphics

- Images (e.g., pixel grid)
- Volume (e.g., voxel grid)
- Meshes (e.g., vertices/edges/faces)
- Pointclouds (e.g., point arrays)
- Animation (e.g., skeletal positions over time; cloth dynamics over time)
- Physics simulations (e.g., fluid flow over space/time, object-body interaction)

Problems in Computer Graphics

- Feature detection (image features, point features) $\mathbb{R}^{m \times m} \rightarrow \mathbb{Z}$
- Denoising, Smoothing, etc. $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Embedding, Distance computation $\mathbb{R}^{m \times m, m \times m} \rightarrow \mathbb{R}^{d}$
- Rendering $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Animation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Physical simulation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Generative models $\mathbb{R}^{d} \rightarrow \mathbb{R}^{m \times m}$

Problems in Computer Graphics

- Feature detection (image features, point features) $\mathbb{R}^{m \times m} \rightarrow \mathbb{Z}$
- Denoising, Smoothing, etc. $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Embedding, Distance computation $\mathbb{R}^{m \times m, m \times m} \rightarrow \mathbb{R}^d$

analysis

- Rendering $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Animation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Physical simulation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Generative models $\mathbb{R}^d \rightarrow \mathbb{R}^{m \times m}$

Problems in Computer Graphics

- Feature detection (image features, point features) $\mathbb{R}^{m \times m} \rightarrow \mathbb{Z}$
- Denoising, Smoothing, etc. $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Embedding, Distance computation $\mathbb{R}^{m \times m, m \times m} \rightarrow \mathbb{R}^{d}$

analysis

- Rendering $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Animation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Physical simulation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Generative models $\mathbb{R}^{d} \rightarrow \mathbb{R}^{m \times m}$

synthesis

Goal: Learn a Parametric Function

$$f_\theta : \mathbb{X} \longrightarrow \mathbb{Y}$$

θ: function parameters, these are learned
$\mathbb{X}$: source domain
$\mathbb{Y}$: target domain

Goal: Learn a Parametric Function

$$f_\theta : \mathbb{X} \longrightarrow \mathbb{Y}$$

θ: function parameters, $\mathbb{X}$: source domain $\mathbb{Y}$: target domain
these are learned

Examples:

Goal: Learn a Parametric Function

$$f_\theta : \mathbb{X} \longrightarrow \mathbb{Y}$$

θ: function parameters, these are learned $\mathbb{X}$: source domain $\mathbb{Y}$: target domain

Examples:

Image Classification: $f_\theta : \mathbb{R}^{w \times h \times c} \longrightarrow \{0, 1, \ldots, k-1\}$

$w \times h \times c$: image dimensions k: class count

Goal: Learn a Parametric Function

$$f_\theta : \mathbb{X} \longrightarrow \mathbb{Y}$$

θ: function parameters, these are learned $\mathbb{X}$: source domain $\mathbb{Y}$: target domain

Examples:

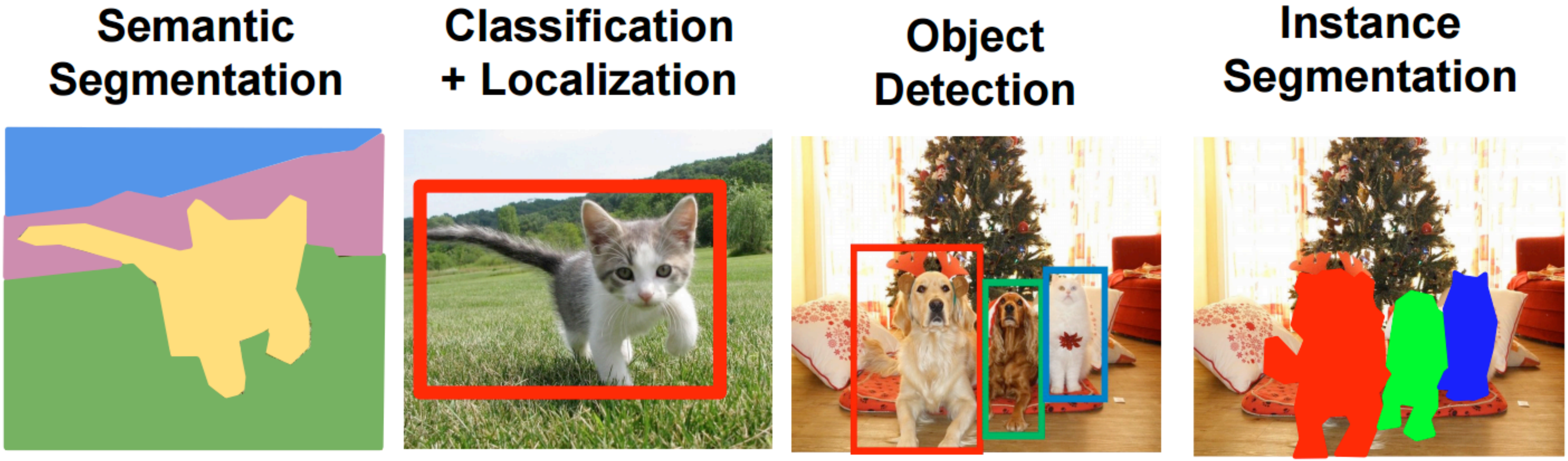
Image Classification: $f_\theta : \mathbb{R}^{w \times h \times c} \longrightarrow \{0, 1, \ldots, k-1\}$

$w \times h \times c$: image dimensions k: class count

Image Synthesis: $f_\theta : \mathbb{R}^{n} \longrightarrow \mathbb{R}^{w \times h \times c}$

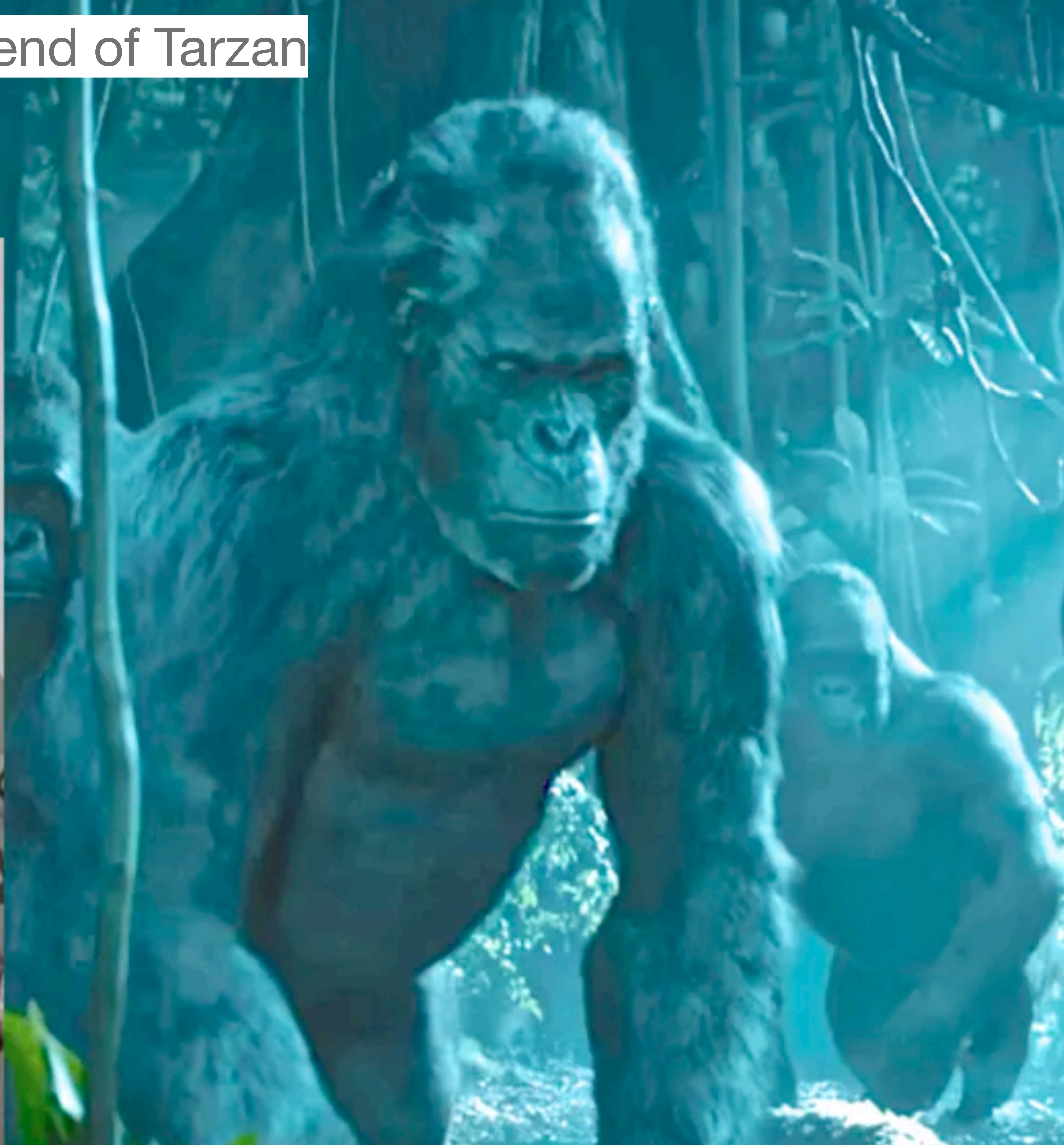
n : latent variable count $w \times h \times c$: image dimensions

Semantic Segmentation



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

The Legend of Tarzan



Pose Detection using CNNs

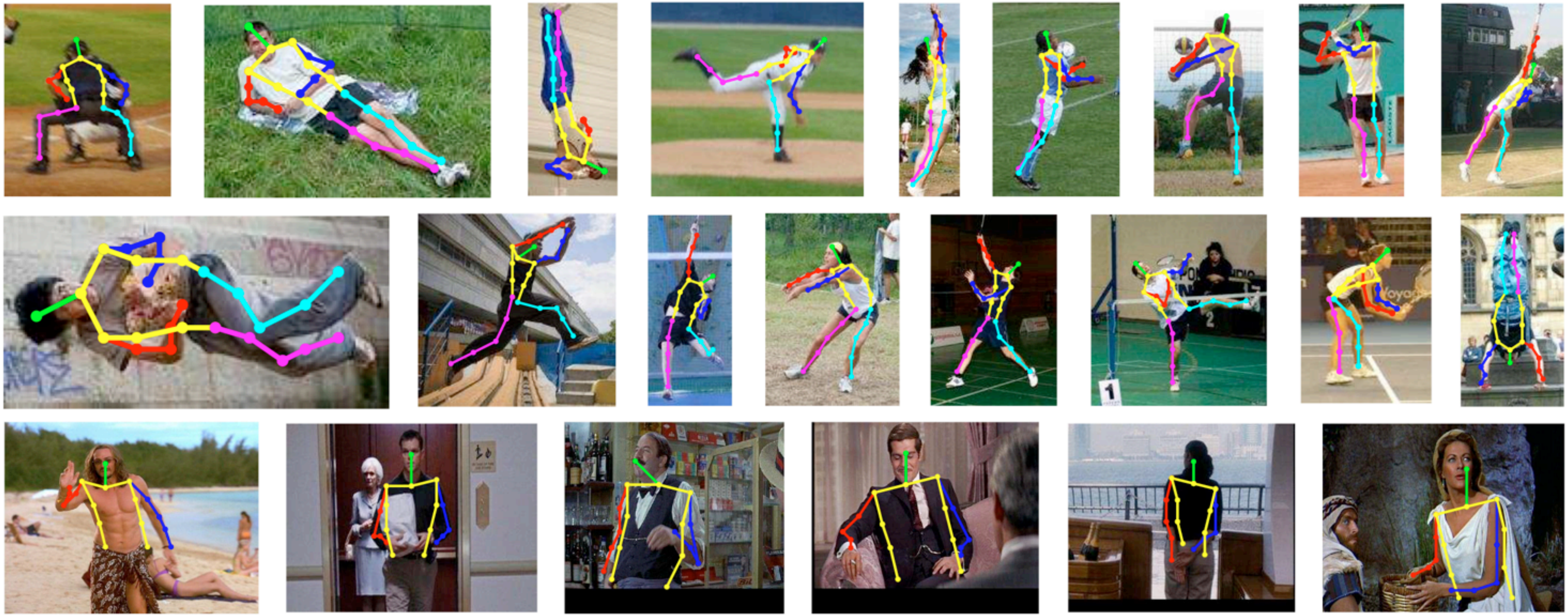


Image Denoising

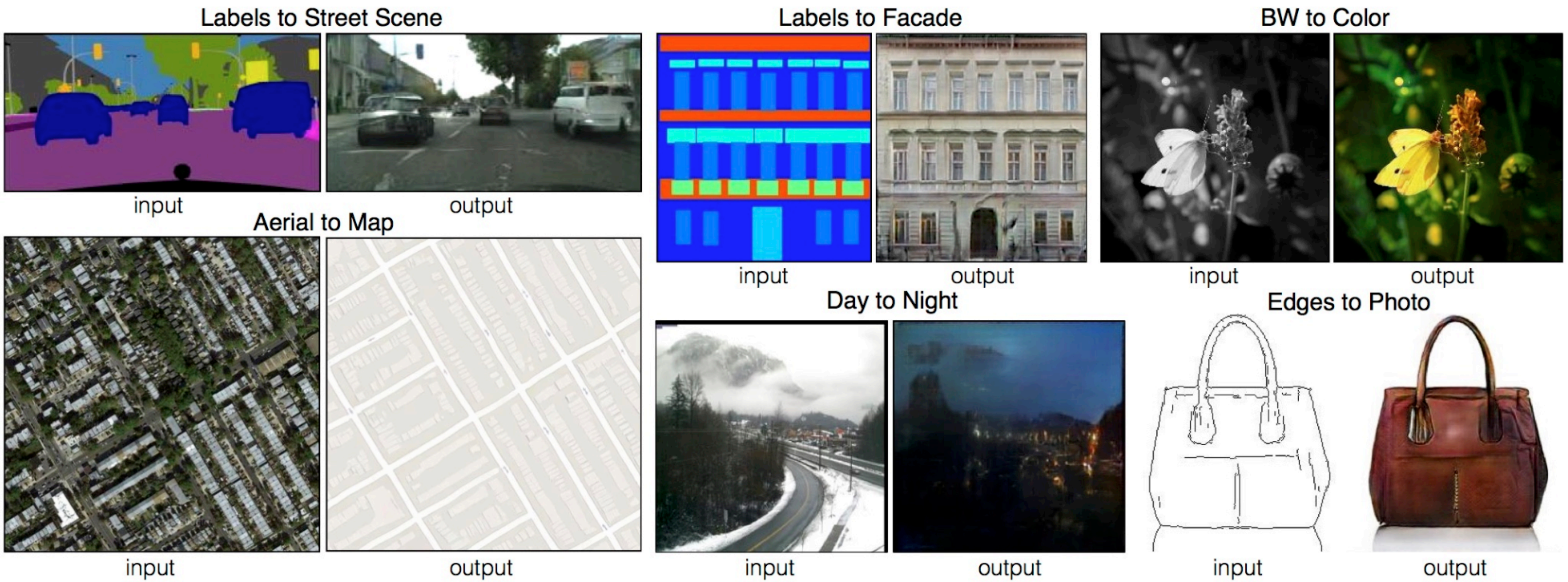
[Chaitanya et al. 2017, Siggraph]

Image Denoising

[Chaitanya et al. 2017, Siggraph]

Pix2Pix (Image Translation)

[Isola et al. 2017, CVPR]



Sketch to Face!

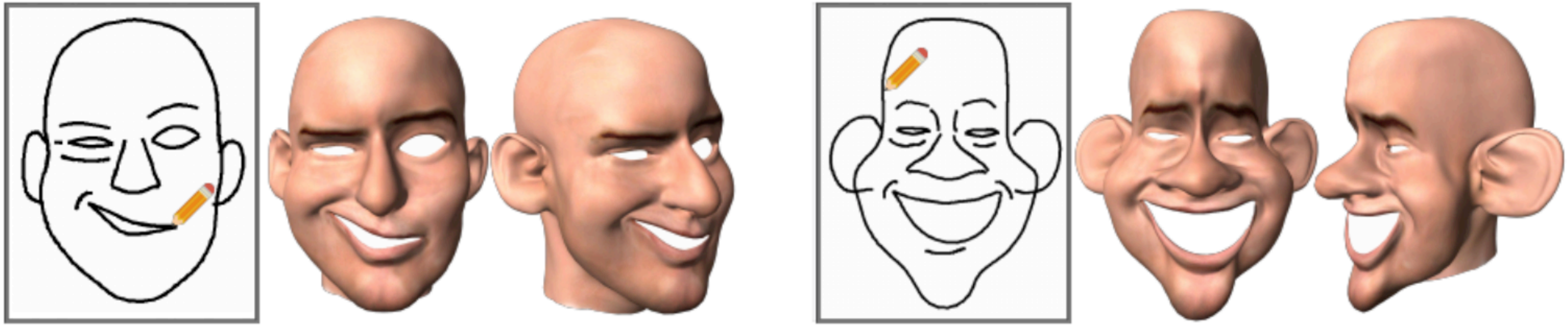
DeepSketch2Face: A Deep Learning Based Sketching System for 3D Face and Caricature Modeling

XIAOGUANG HAN, CHANG GAO, and YIZHOU YU, The University of Hong Kong

Sketch to Face!

DeepSketch2Face: A Deep Learning Based Sketching System for 3D Face and Caricature Modeling

XIAOGUANG HAN, CHANG GAO, and YIZHOU YU, The University of Hong Kong



[Wang et al. 2018, Siggraph Asia]

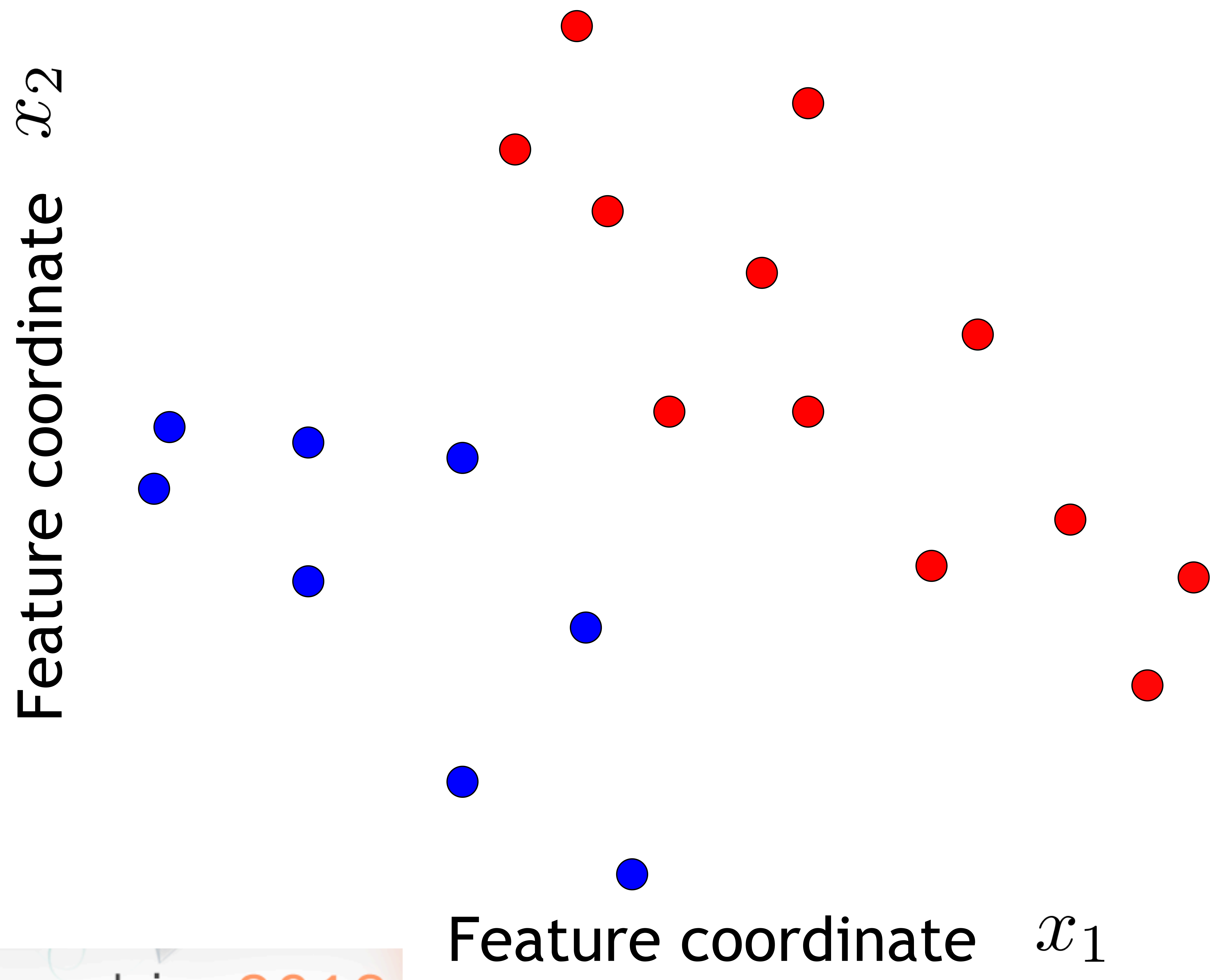
Real Images

[Wang et al. 2018, Siggraph Asia]

Real Images

Machine Learning 101: Linear Classifier

$$f_\theta : \mathbb{R}^n \longrightarrow \{0, 1\}$$

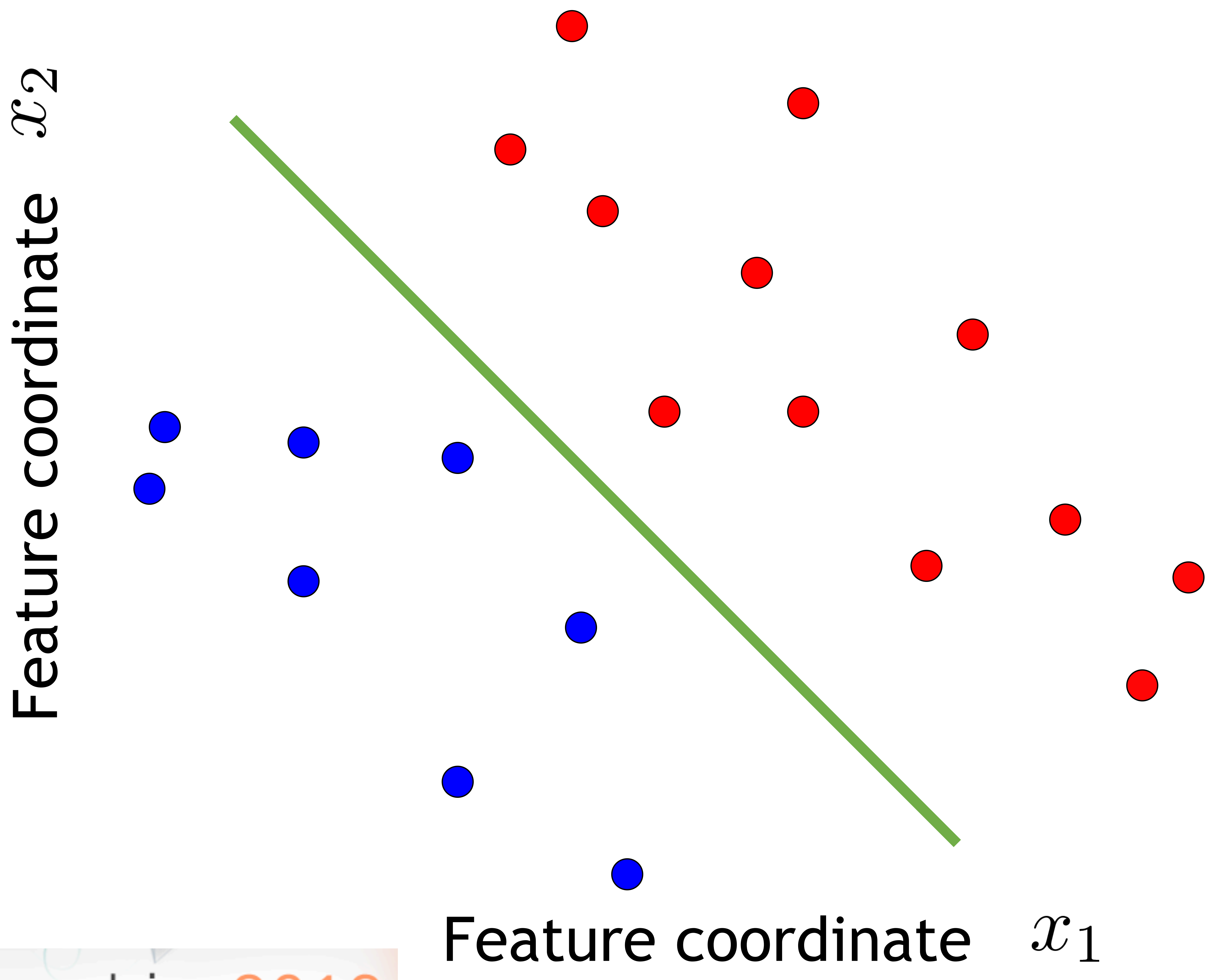


Each data point has a class label:

$$y^i = \begin{cases} 1 & (\text{red}) \\ 0 & (\text{blue}) \end{cases}$$

Machine Learning 101: Linear Classifier

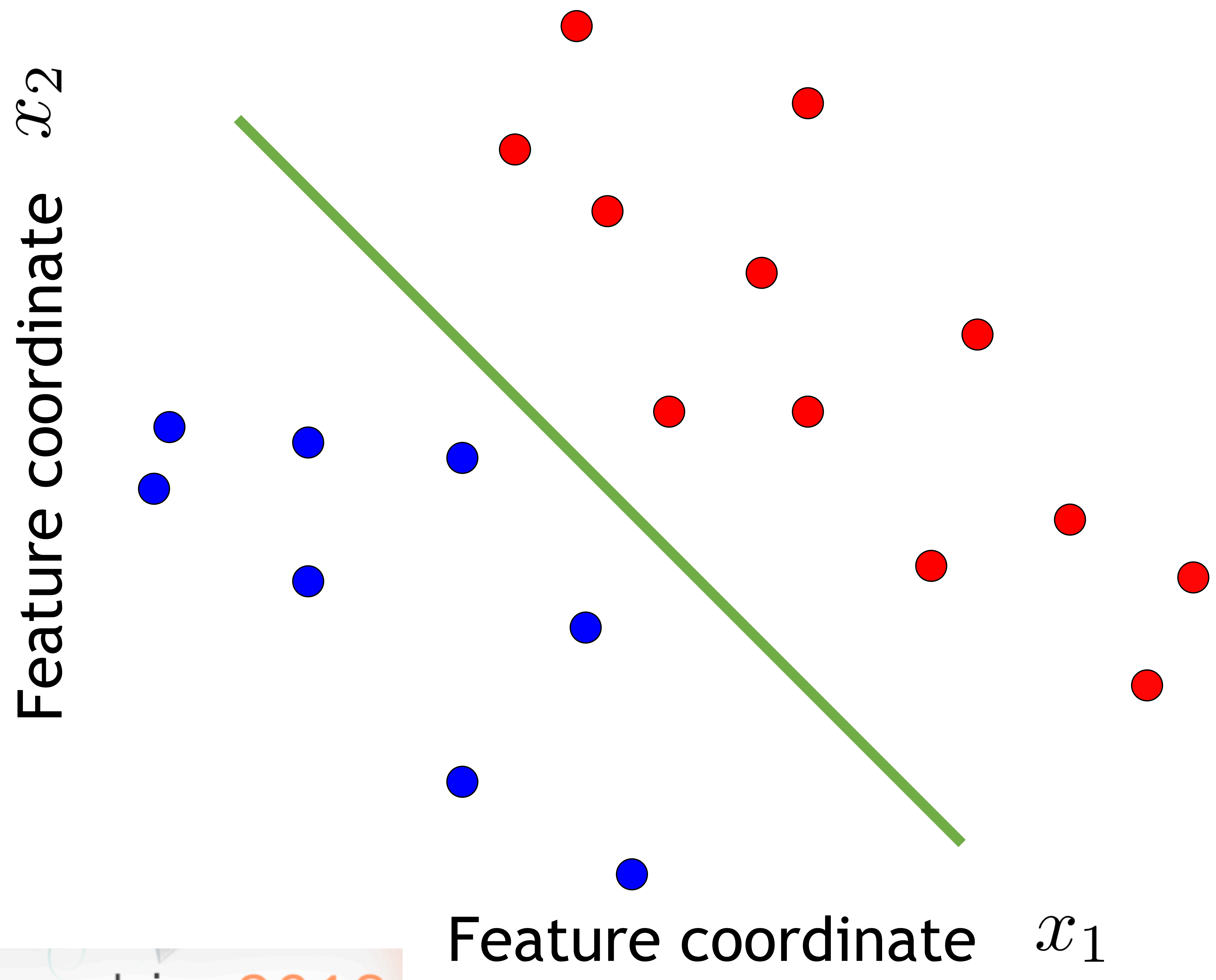
$$f_\theta : \mathbb{R}^n \longrightarrow \{0, 1\}$$



Each data point has a class label:

$$y^i = \begin{cases} 1 & (\ ) \\ 0 & (\ ) \end{cases}$$

Machine Learning 101: Linear Classifier



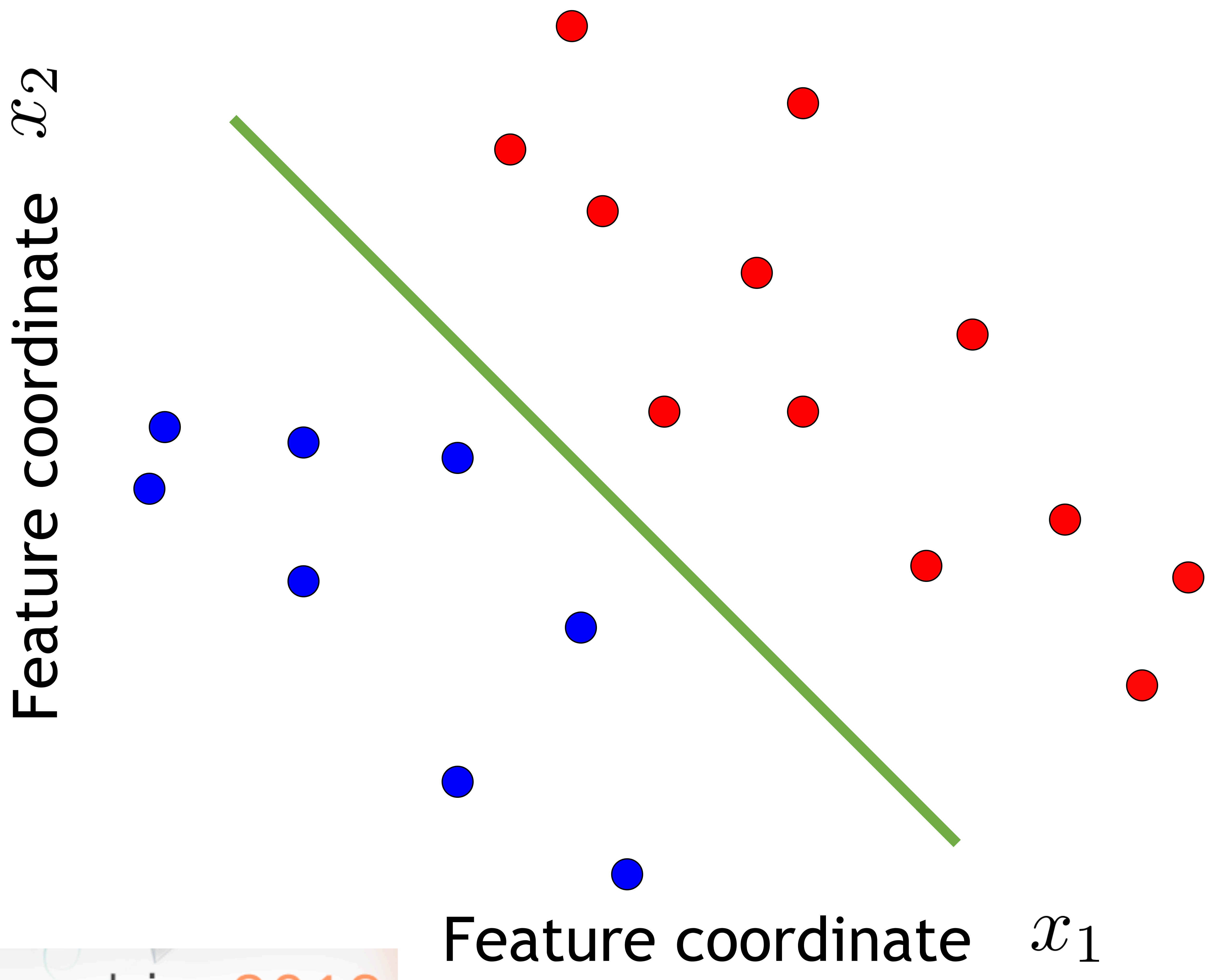
$$f_\theta : \mathbb{R}^n \longrightarrow \{0,1\}$$

$$f_\theta(x) = \begin{cases} 1 & \text{if } wx + b \geq 0 \\ 0 & \text{if } wx + b < 0 \end{cases}$$

Each data point has a class label:

$$y^i = \begin{cases} 1 & (\text{red}) \\ 0 & (\text{blue}) \end{cases}$$

Machine Learning 101: Linear Classifier



$$f_\theta : \mathbb{R}^n \longrightarrow \{0, 1\}$$

$$f_\theta(x) = \begin{cases} 1 & \text{if } wx + b \geq 0 \\ 0 & \text{if } wx + b < 0 \end{cases}$$

$$\boxed{\theta = \{w, b\}}$$

Each data point has a class label:

$$y^i = \begin{cases} 1 & (\bullet) \\ 0 & (\bullet) \end{cases}$$

Data-driven Algorithms (Supervised)

Labelled data
(supervision data)

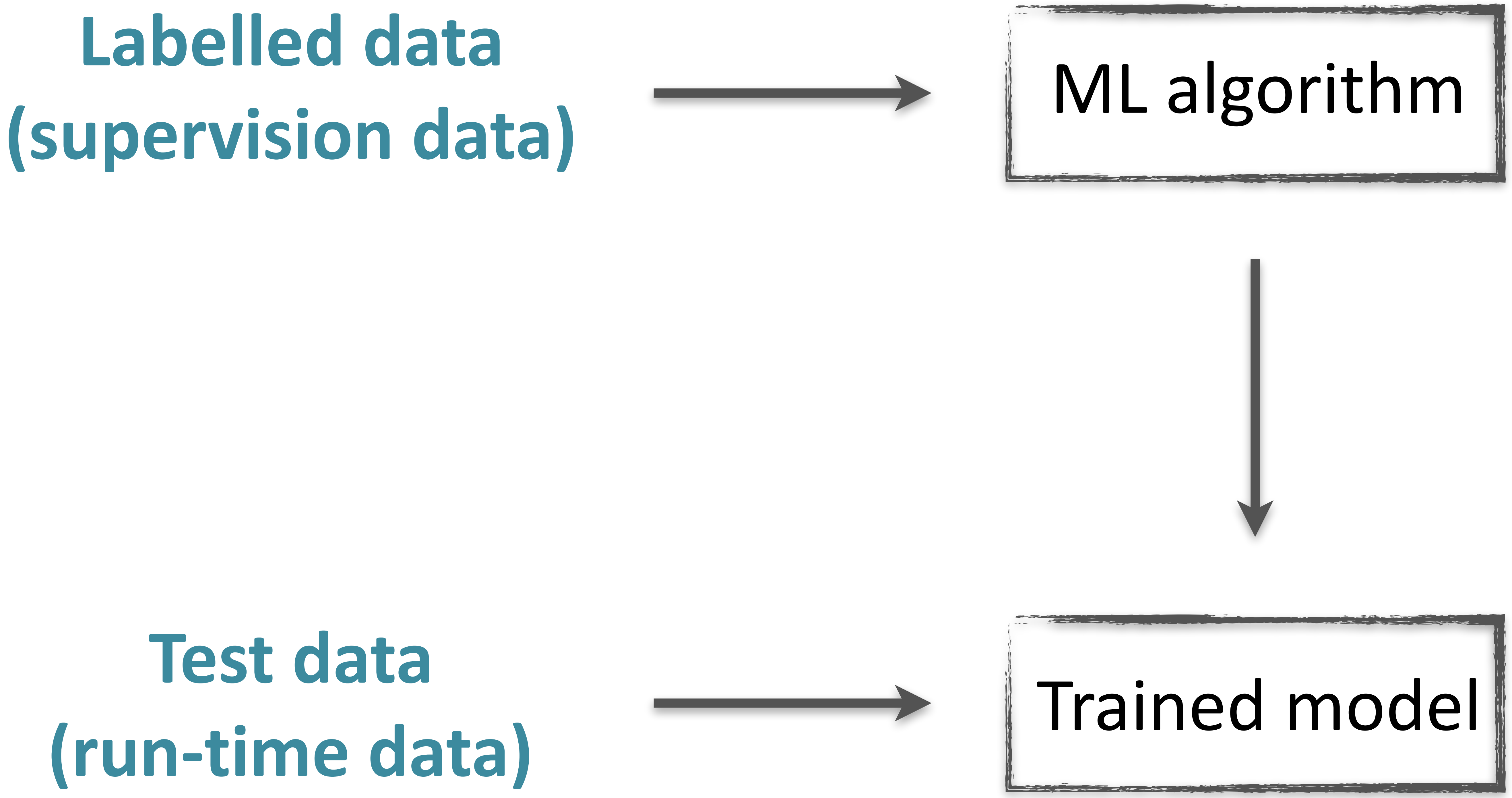
Data-driven Algorithms (Supervised)

Labelled data
(supervision data)

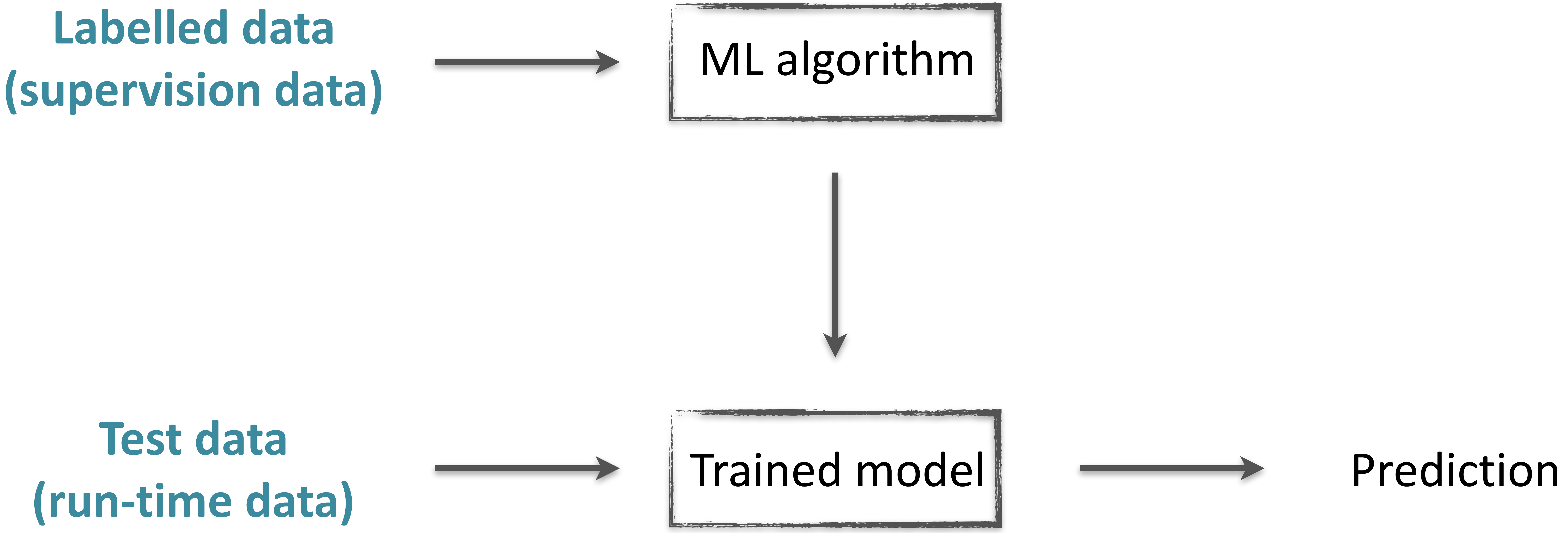
ML algorithm

Trained model

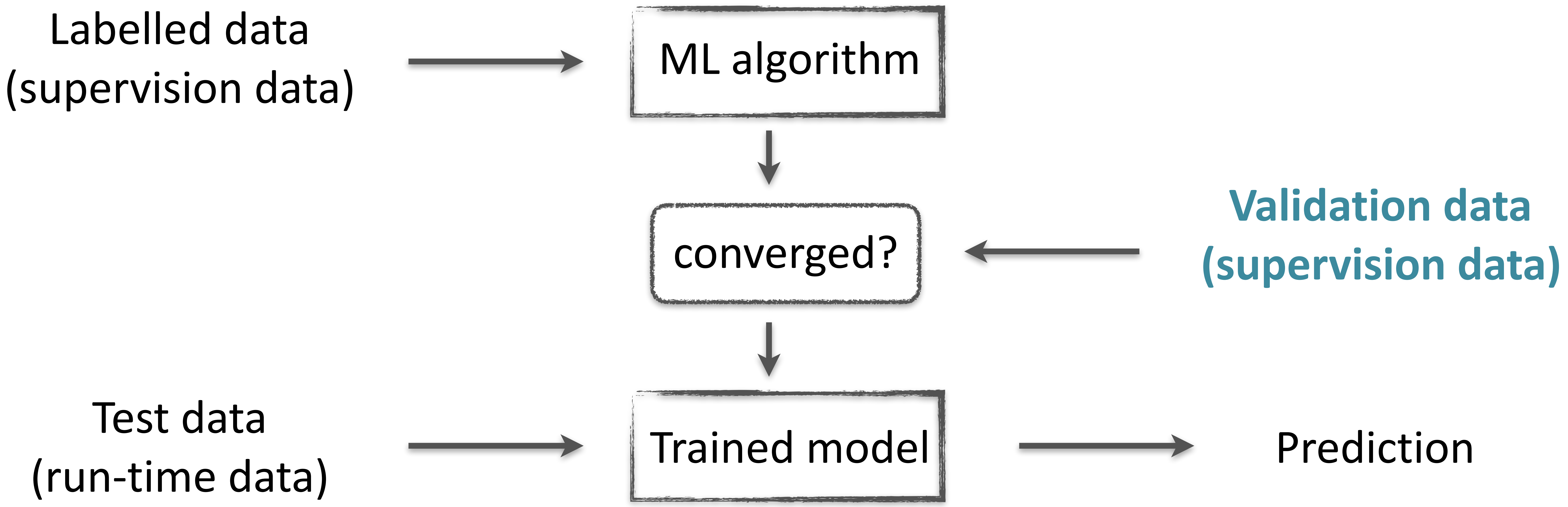
Data-driven Algorithms (Supervised)



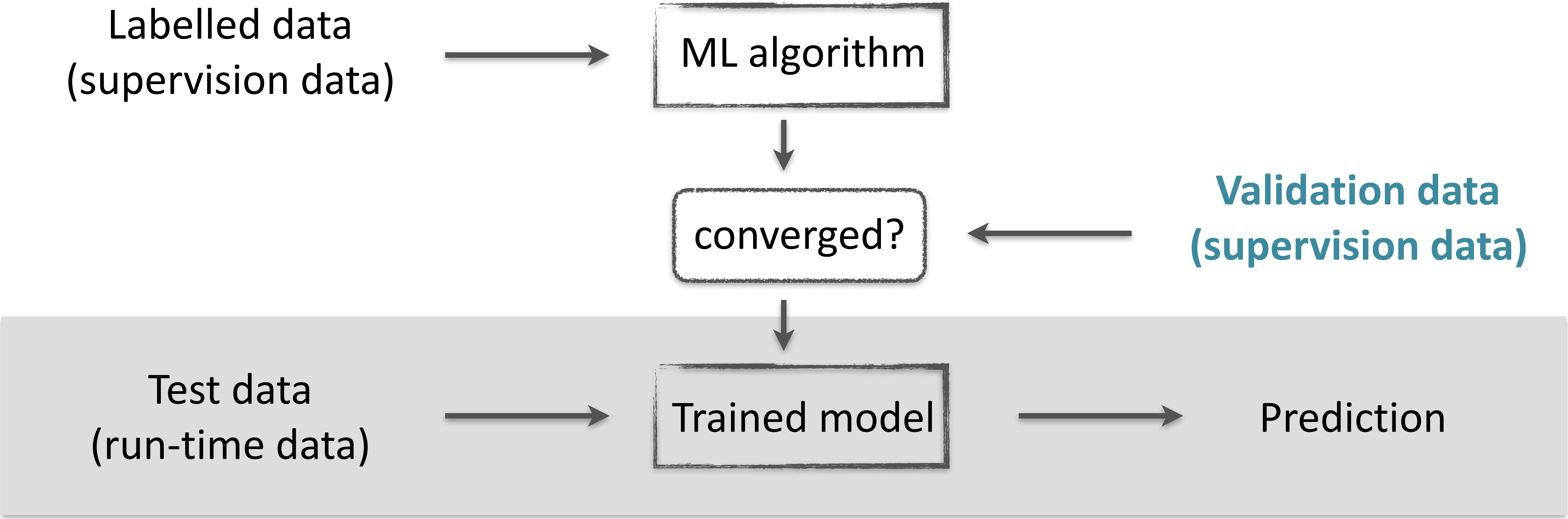
Data-driven Algorithms (Supervised)



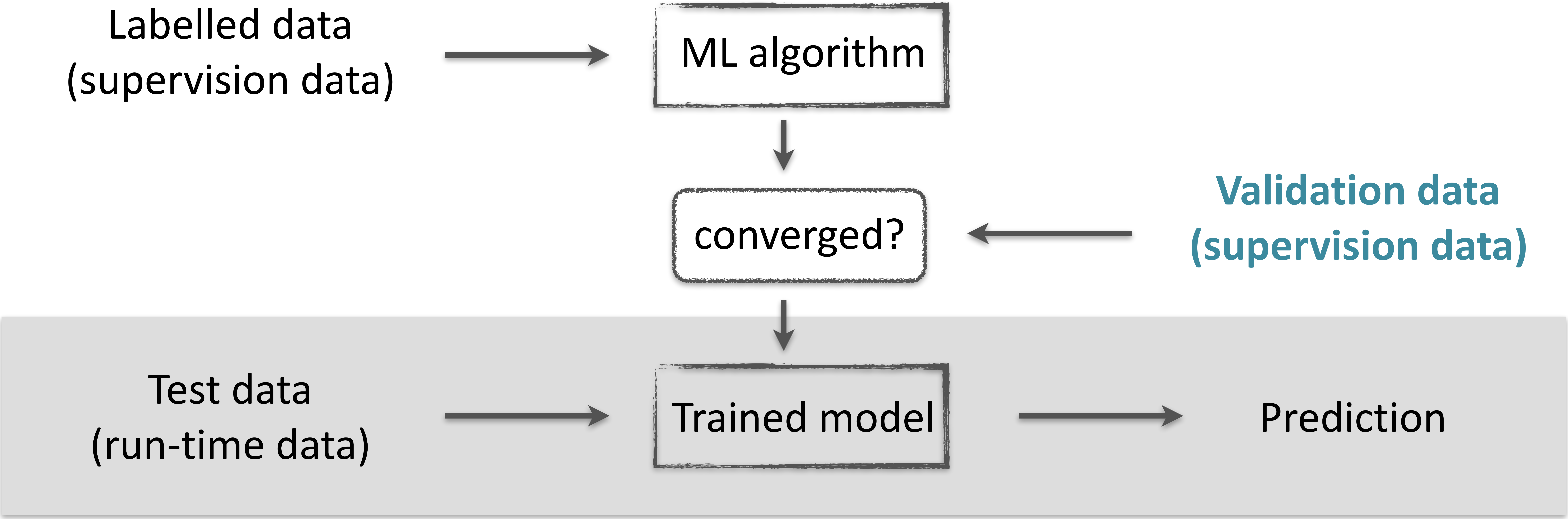
Data-driven Algorithms (Supervised)



Data-driven Algorithms (Supervised)

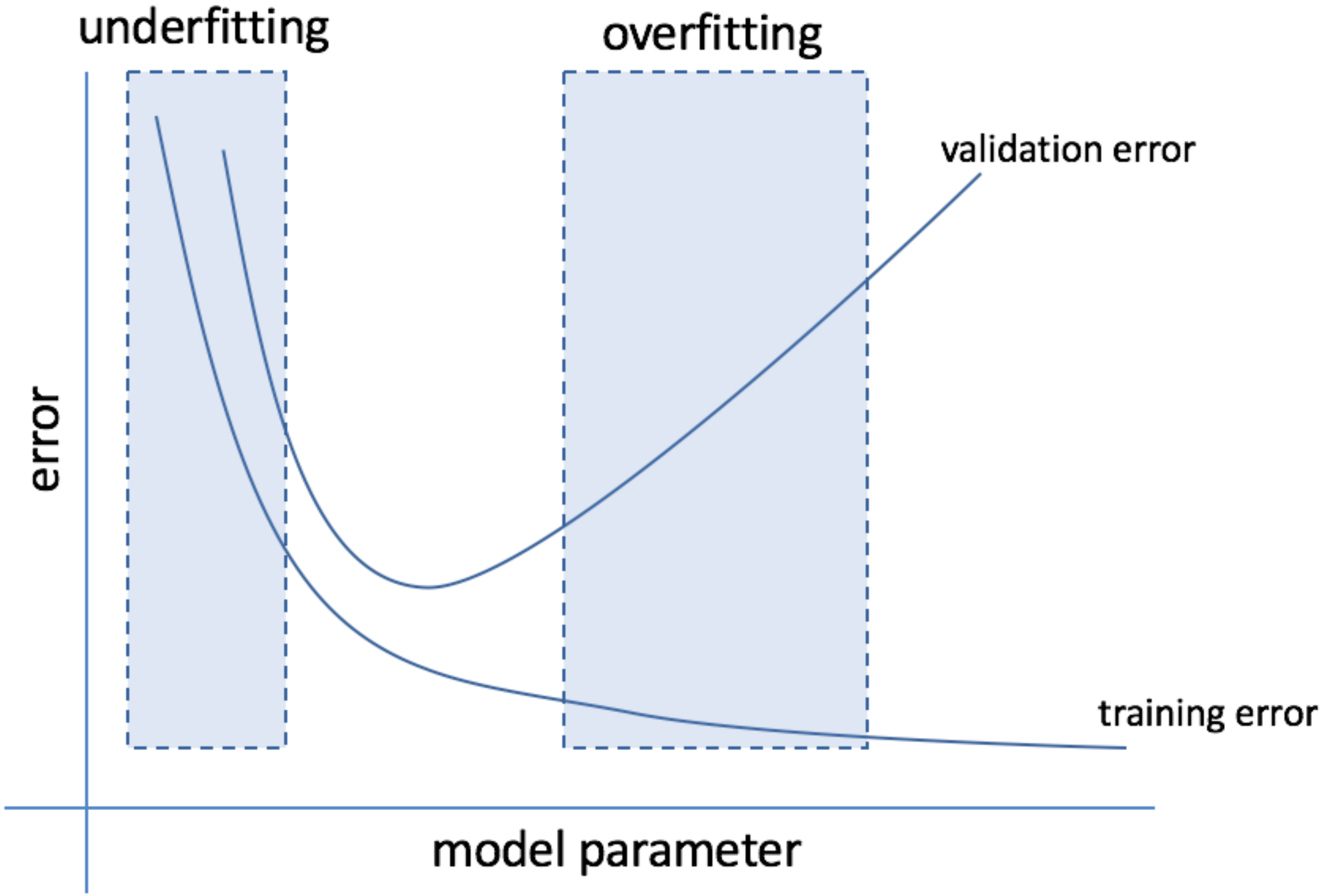


Data-driven Algorithms (Supervised)

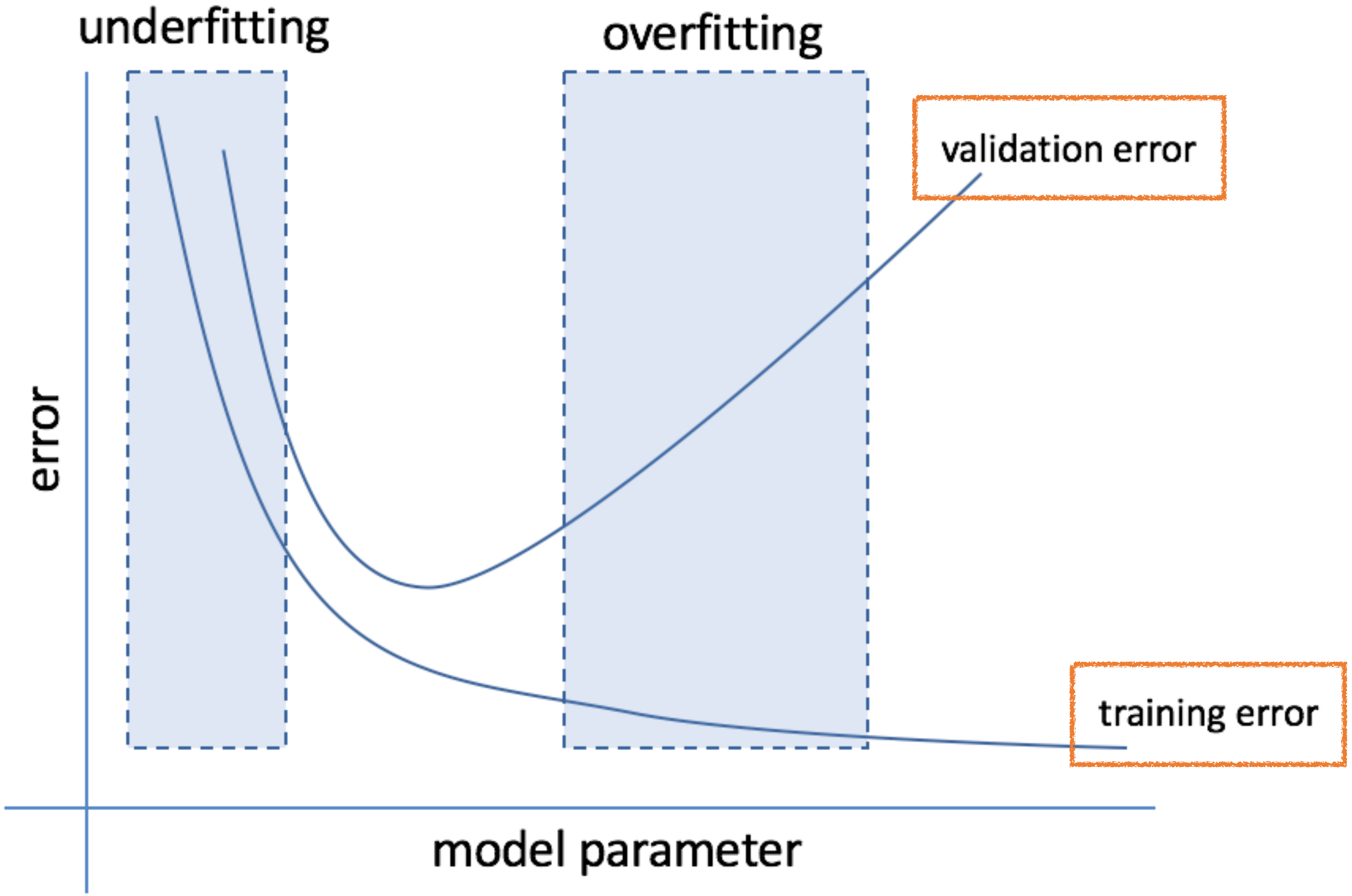


Implementation Practice: Training: 70%; Validation: 15%; Test 15%

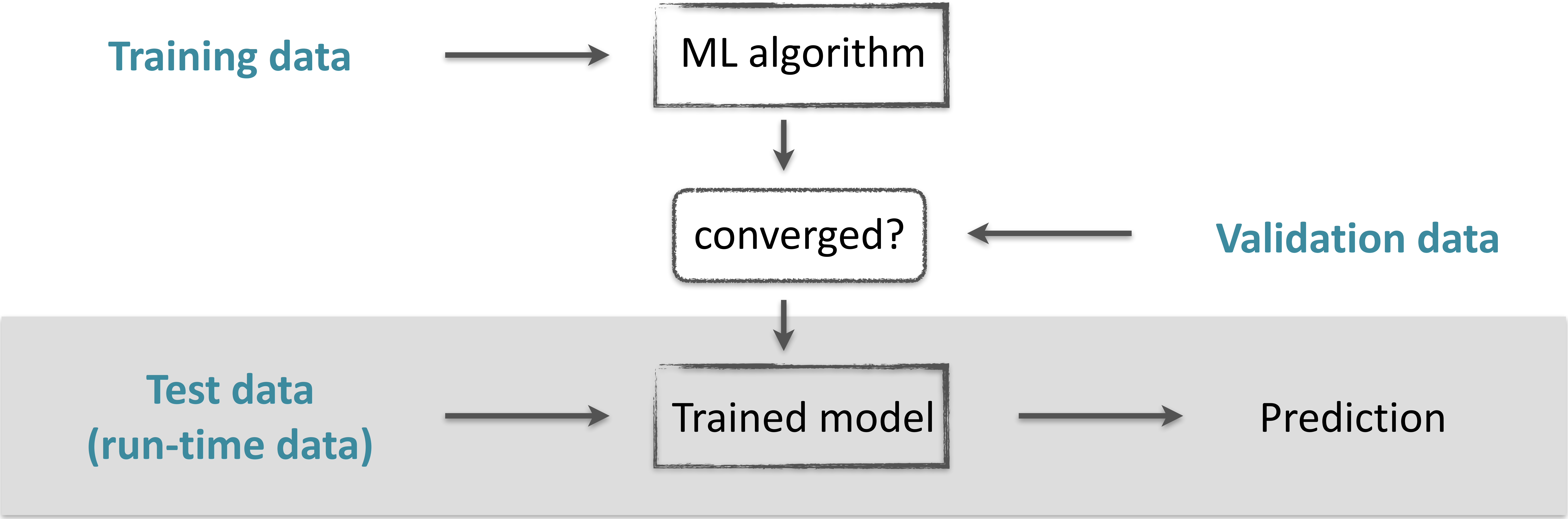
Training versus Validation Loss/Accuracy



Training versus Validation Loss/Accuracy

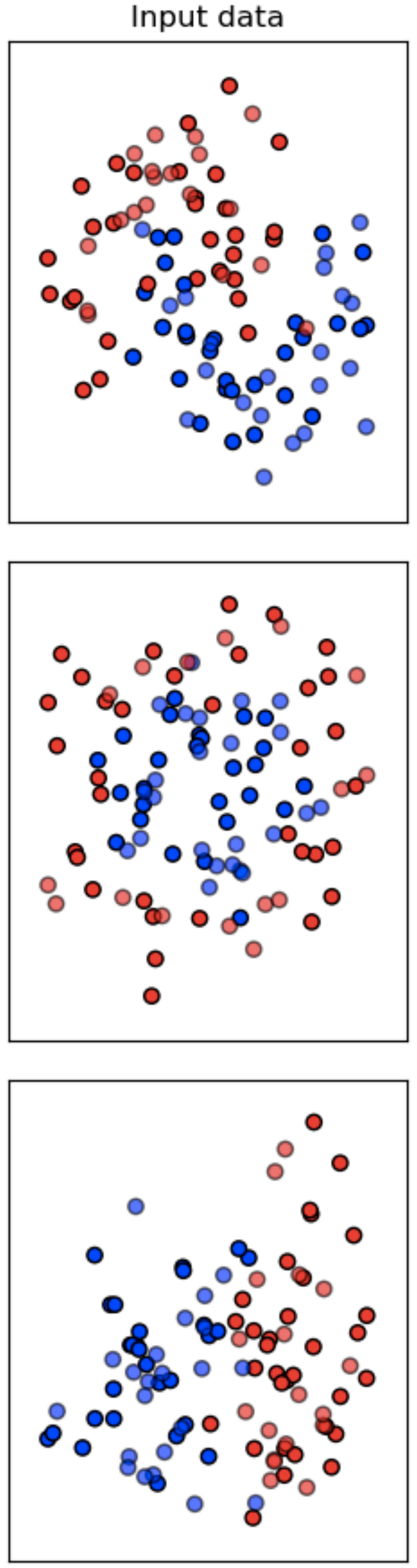


Data-driven Algorithms (Unsupervised)



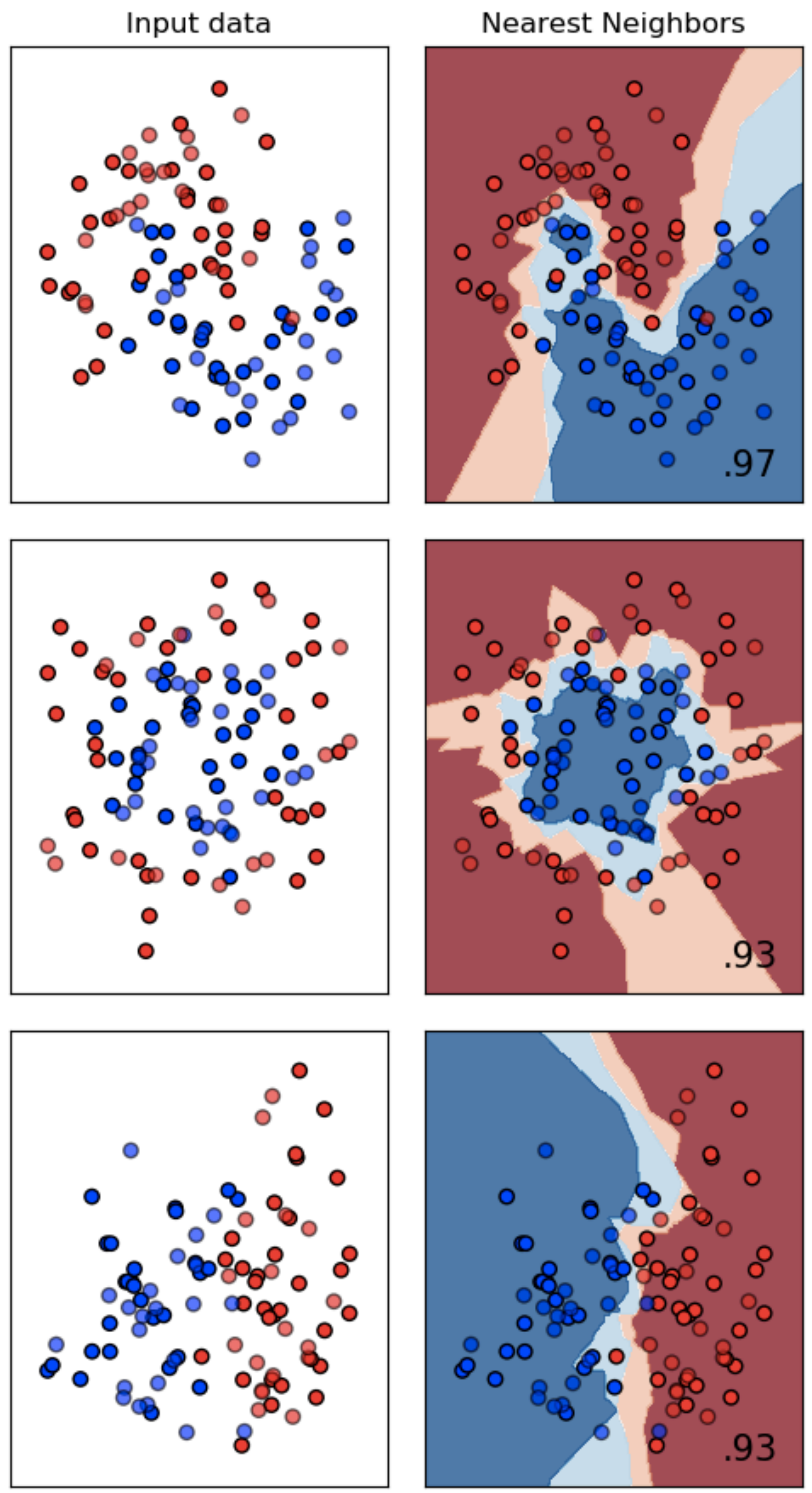
Implementation Practice: Training: 70%; Validation: 15%; Test 15%

Various ML Approaches (Supervised approaches)



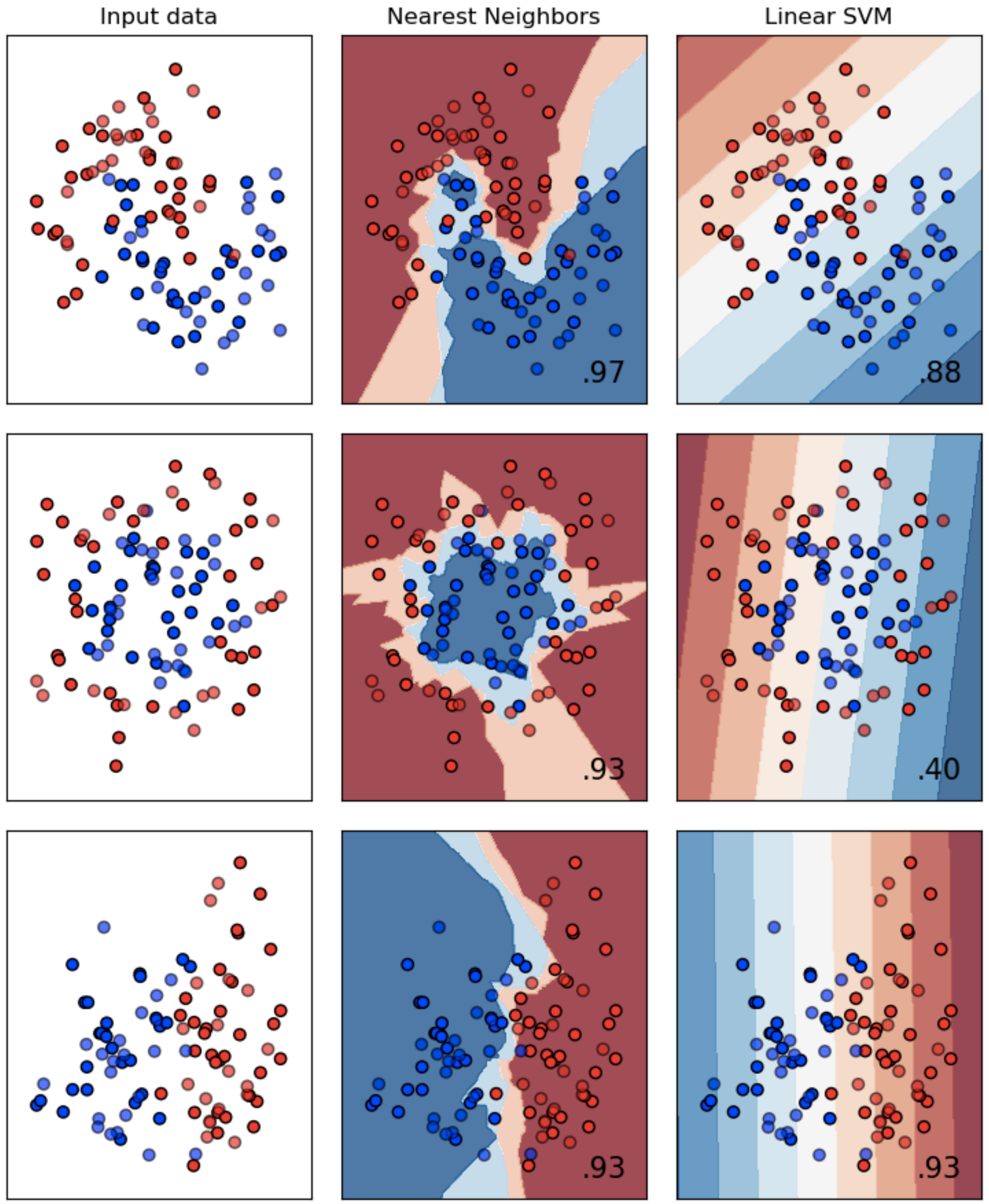
http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Various ML Approaches (Supervised approaches)



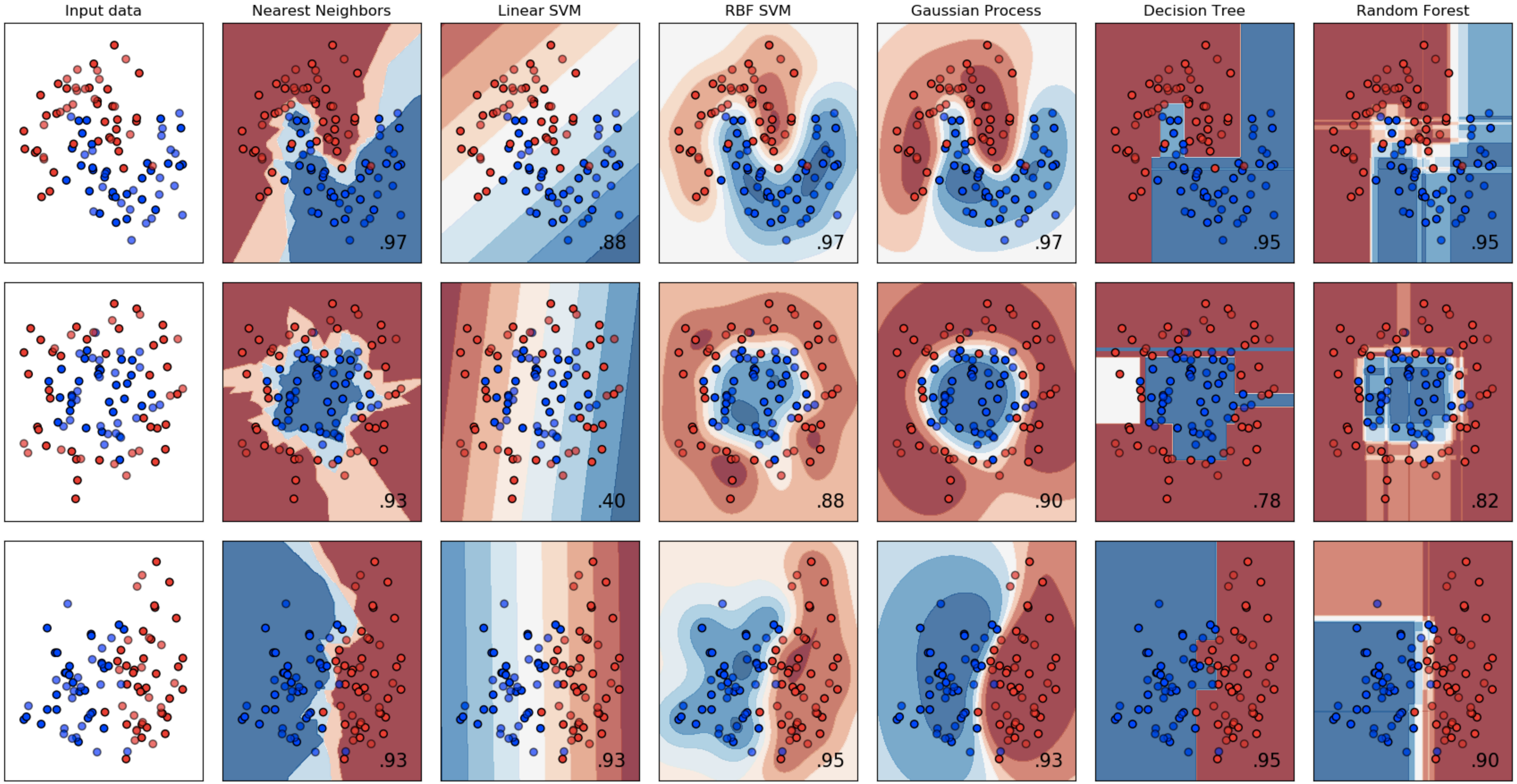
http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Various ML Approaches (Supervised approaches)



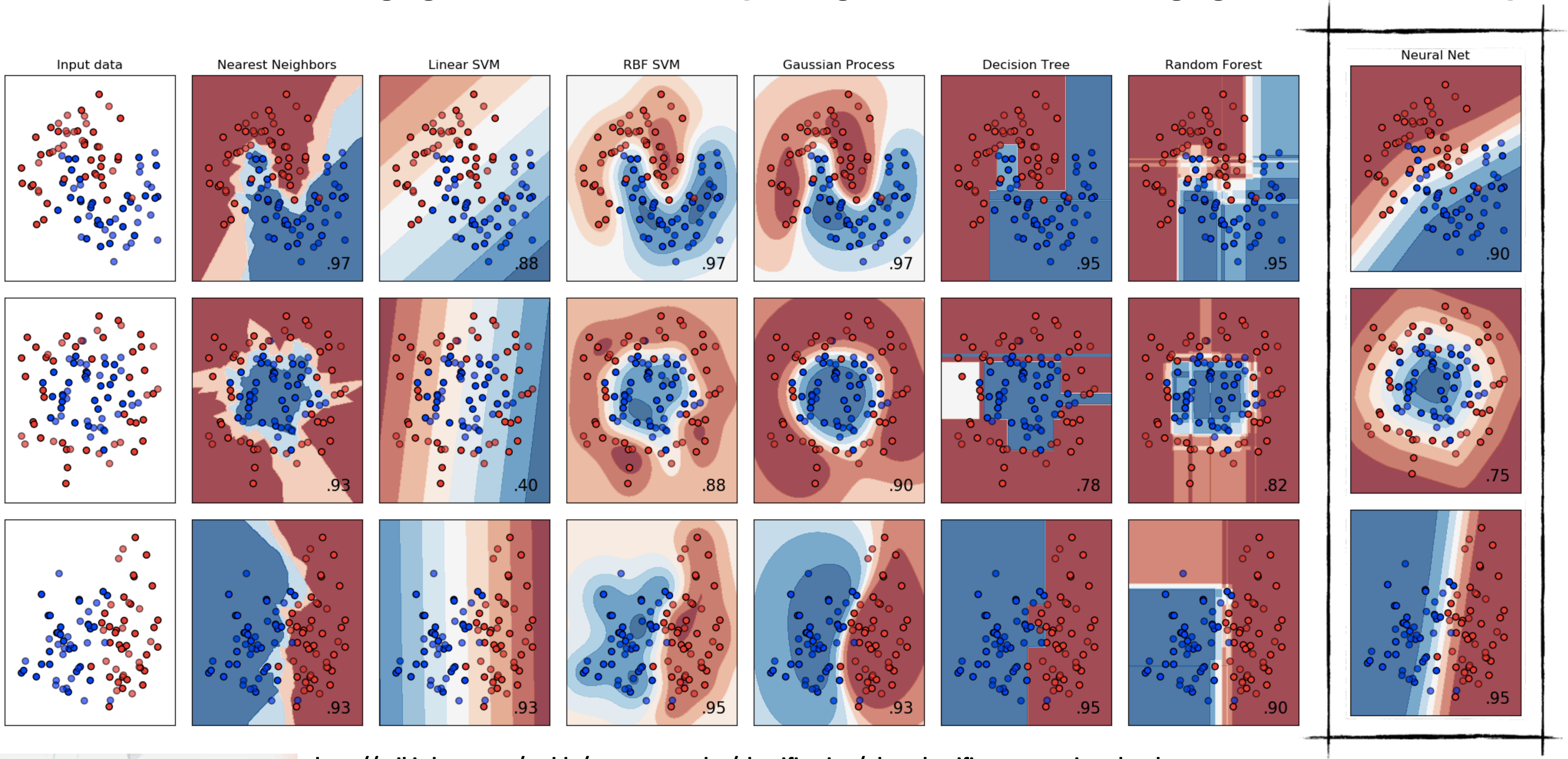
http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Various ML Approaches (Supervised approaches)



http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Various ML Approaches (Supervised approaches)

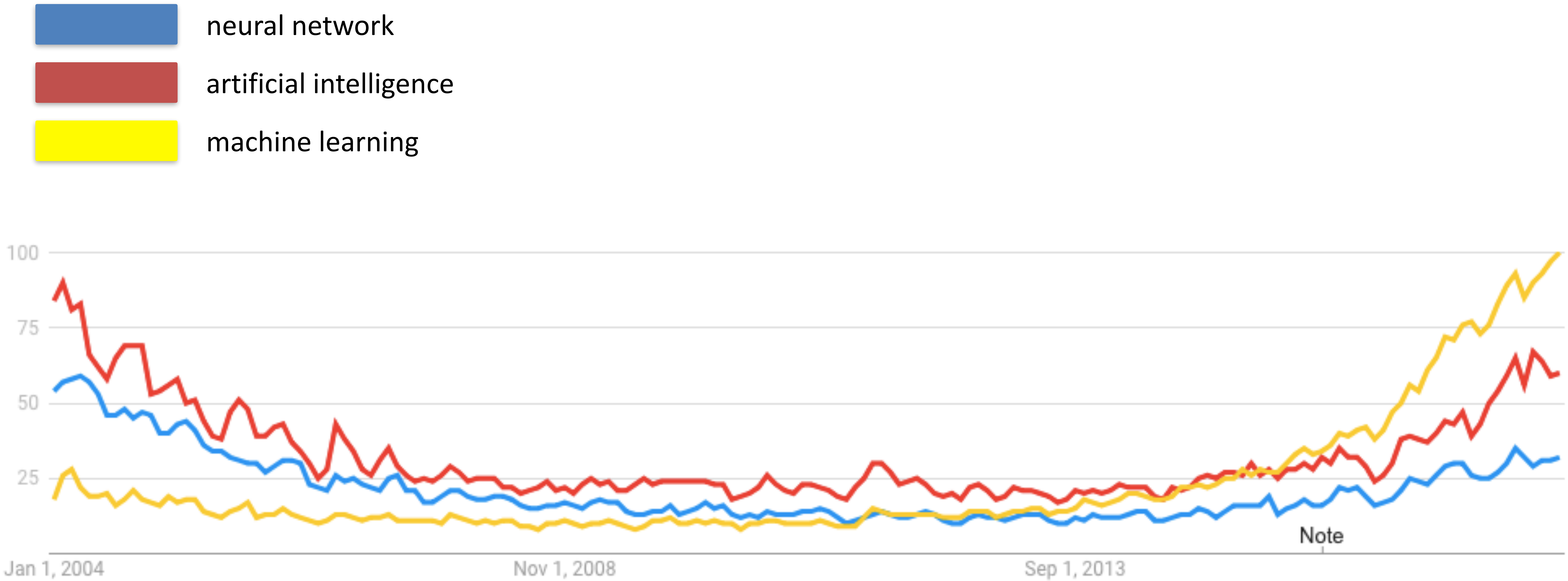


http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Rise of Learning

- 1958: Perceptron
- 1974: Backpropagation
- 1981: Hubel & Wiesel wins Nobel prize for ‘visual system’
- 1990s: SVM era
- 1998: CNN used for handwriting analysis
- **2012: AlexNet wins ImageNet**

Rise of Machine Learning



Rise of Machine Learning

neural network

artificial intelligence

machine learning

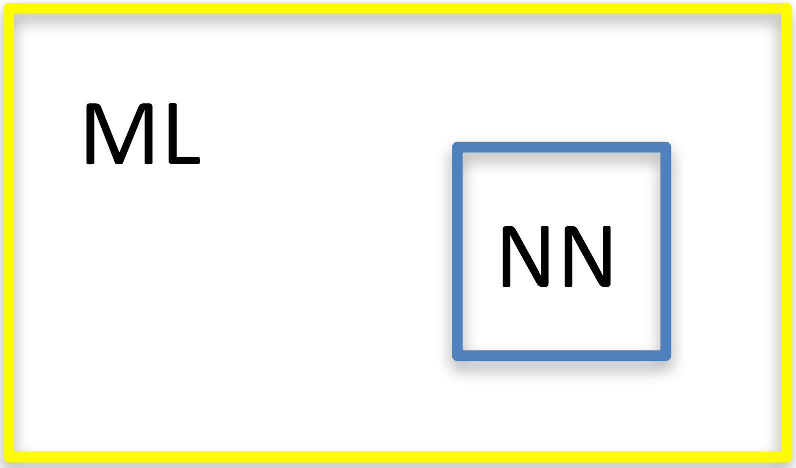
NN

Rise of Machine Learning

neural network

artificial intelligence

machine learning

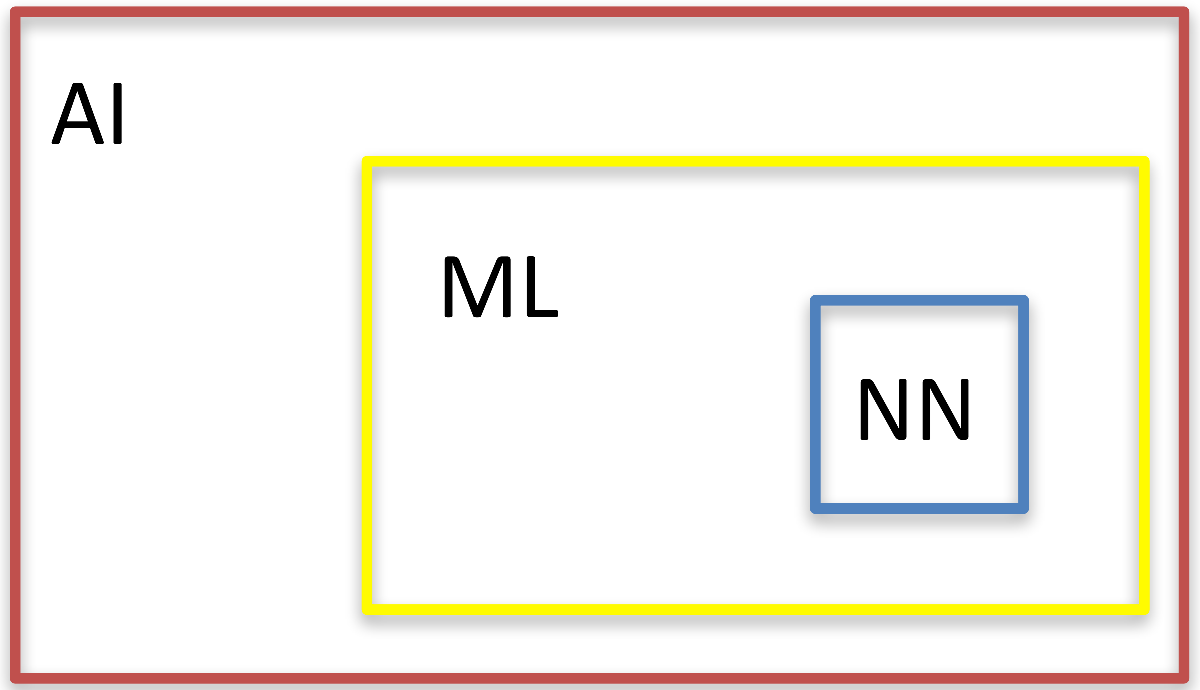


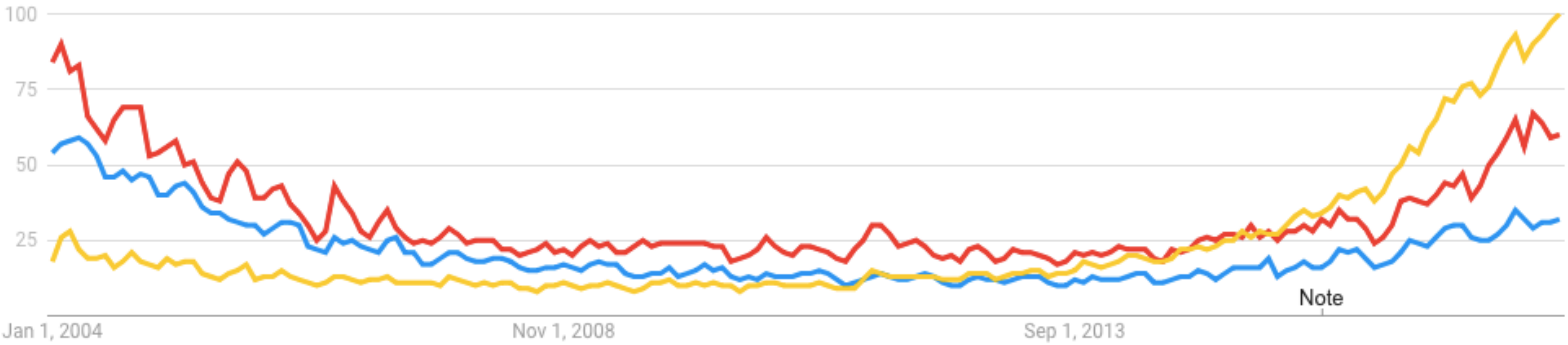
Rise of Machine Learning

neural network

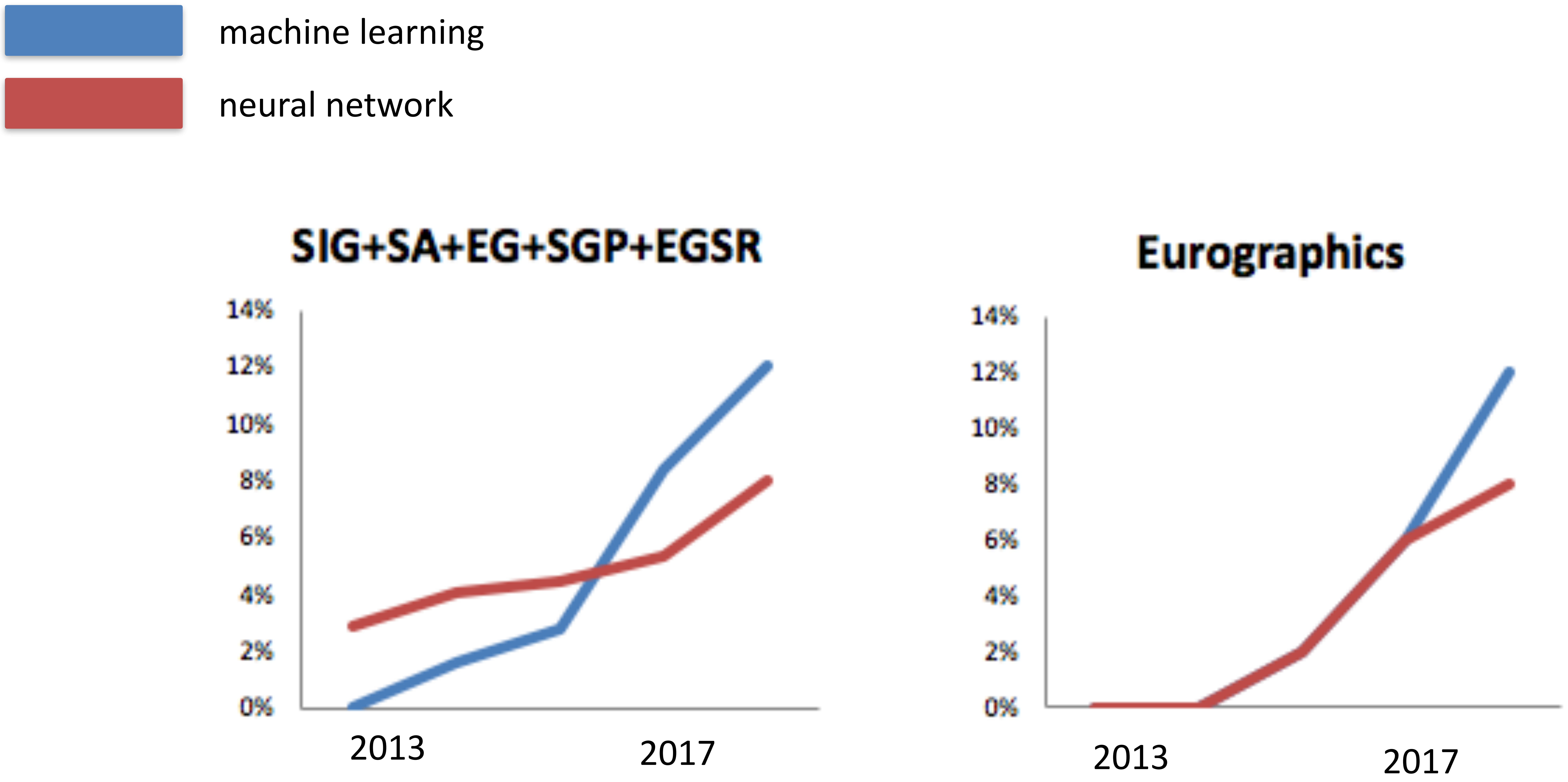
artificial intelligence

machine learning





Rise of Machine Learning (in Graphics)



What is Special about CG?

What is Special about CG?

1. **Image Processing** (image translation tasks)

What is Special about CG?

1. Image Processing (image translation tasks)

2. Many sources of input data — model building
   (e.g., images, scanners, motion capture)

What is Special about CG?

1. **Image Processing** (image translation tasks)

2. Many sources of input data — **model building**
   (e.g., images, scanners, motion capture)

3. Many sources of **synthetic data** — can serve as supervision data
   (e.g., rendering, animation)

What is Special about CG?

1. **Image Processing** (image translation tasks)

2. Many sources of input data — **model building**
   (e.g., images, scanners, motion capture)

3. Many sources of **synthetic data** — can serve as supervision data
   (e.g., rendering, animation)

4. Many problems in **generative models**

Main Challenges and Scope for Innovation

Main Challenges and Scope for Innovation

1. **Representation:** How is the data organised and structured?

Main Challenges and Scope for Innovation

1. **Representation:** How is the data organised and structured?

2. **Training data:** Is it synthetic or real, or mixed?

Main Challenges and Scope for Innovation

1. **Representation:** How is the data organised and structured?

2. **Training data:** Is it synthetic or real, or mixed?

3. **User control:** End-to-end or in small steps?

Main Challenges and Scope for Innovation

1. **Representation:** How is the data organised and structured?

2. **Training data:** Is it synthetic or real, or mixed?

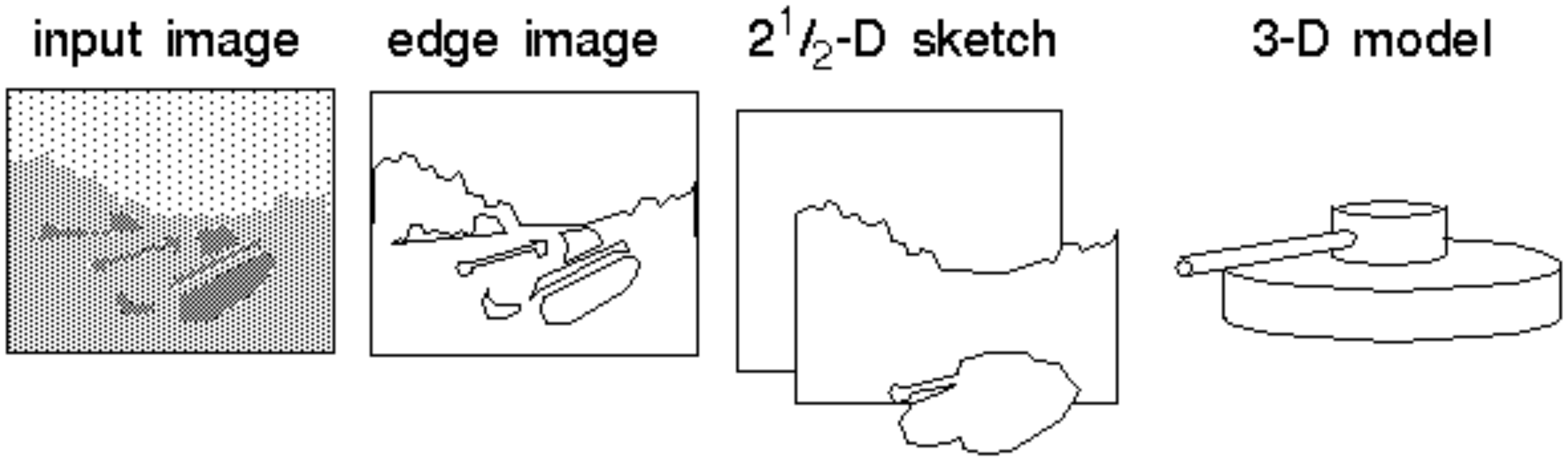
3. **User control:** End-to-end or in small steps?

4. **Loss functions:** Hand-crafted or learned from data?

End-to-end: **Learned Features**

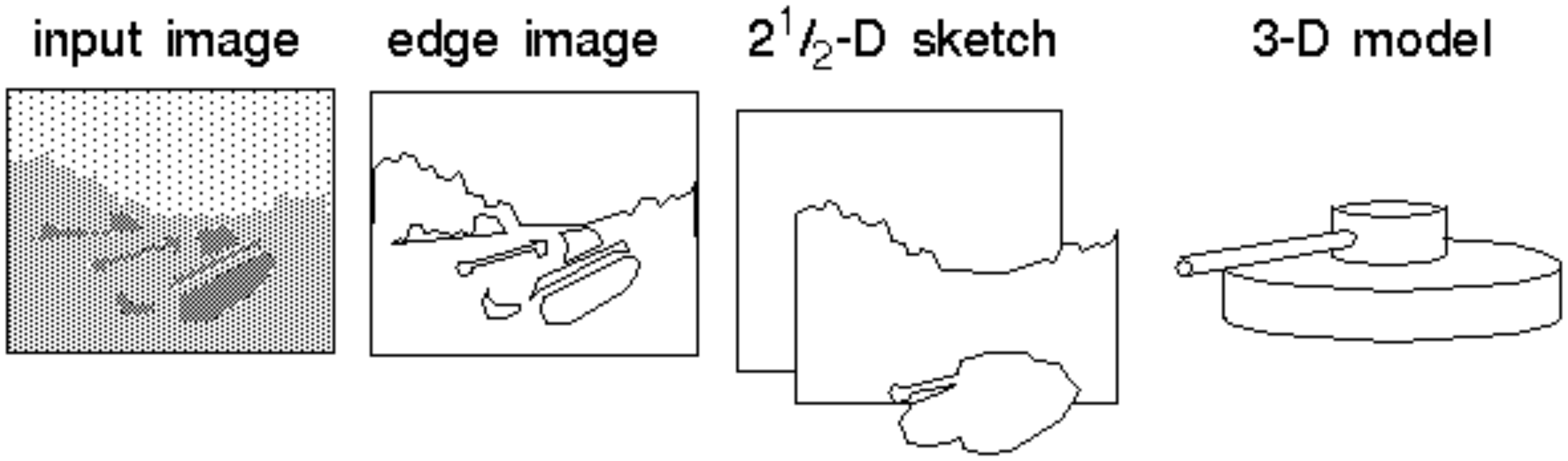
End-to-end: Learned Features

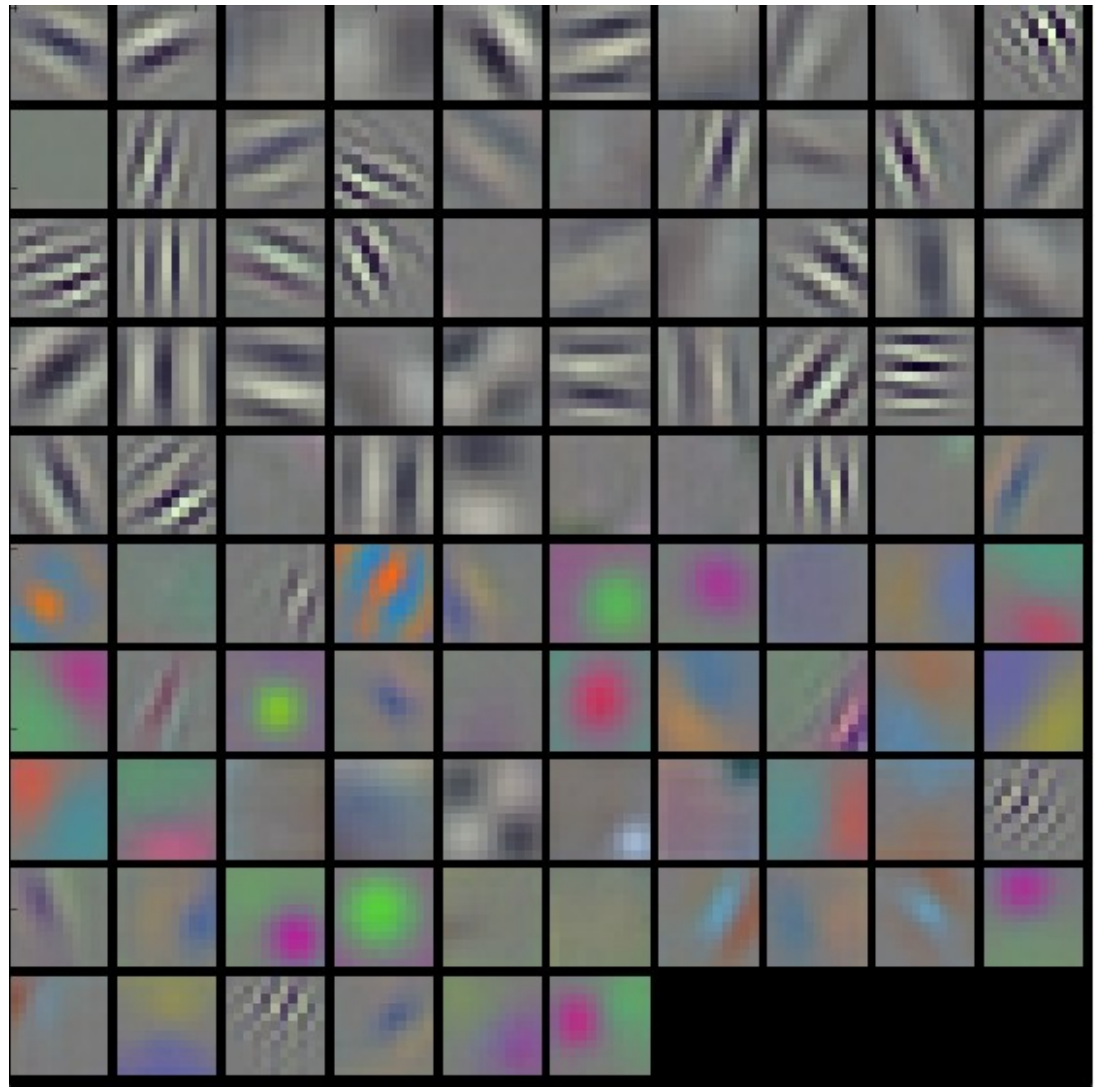
- *Old days*
  - Handcrafted feature extraction, e.g., edges or corners (hand-crafted)
  - Mostly with linear models (PCA, etc.)



End-to-end: Learned Features

- *Old days*
  - Handcrafted feature extraction, e.g., edges or corners (hand-crafted)
  - Mostly with linear models (PCA, etc.)
- *Now*
  - End-to-end
  - Move away from hand-crafted representations





End-to-end: Learned Loss

End-to-end: Learned Loss

- *Old days*
  - Evaluation came after
  - It was a bit optional
    - You might still have a good algorithm without a good way of quantifying it
    - Evaluation helped publishing

End-to-end: Learned Loss

- *Old days*
  - Evaluation came after
  - It was a bit optional
    - You might still have a good algorithm without a good way of quantifying it
    - Evaluation helped publishing
- *Now*
  - It is essential and build-in
  - If the loss is not good, the result is not good
  - (Extensive) Evaluation happens automatically

End-to-end: Learned Loss

- *Old days*
  - Evaluation came after
  - It was a bit optional
    - You might still have a good algorithm without a good way of quantifying it
    - Evaluation helped publishing
- *Now*
  - It is essential and build-in
  - If the loss is not good, the result is not good
  - (Extensive) Evaluation happens automatically
- While still much is left to do, this makes graphics much more reproducible

End-to-end: Real/Generated **Data**

End-to-end: Real/Generated **Data**

- *Old days*
  - Test with some toy examples
  - Deploy on real stuff
  - Maybe collect some performance data later

End-to-end: Real/Generated **Data**

- *Old days*
  - Test with some toy examples
  - Deploy on real stuff
  - Maybe collect some performance data later

- *Now*
  - Test and deploy need to be as identical **(in distribution)**
  - Need to collect data first
  - No two steps

Examples in Graphics

Geometry

Image manipulation

Animation

Rendering

Examples in Graphics

Geometry

Procedural modelling

Mesh segmentation

Learning deformations

Colorization

Sketch simplification

Image manipulation

Animation

BRDF estimation

Boxification

Fluid

Real-time rendering

Animation

Rendering

Denoising

Facial animation

PCD processing

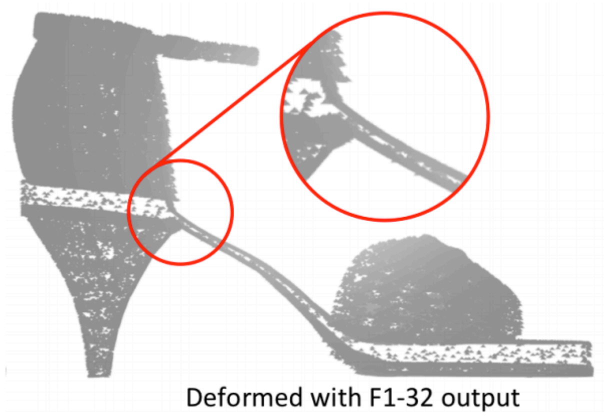
Examples in Graphics

Sketch simplification

Colorization

Procedural modelling

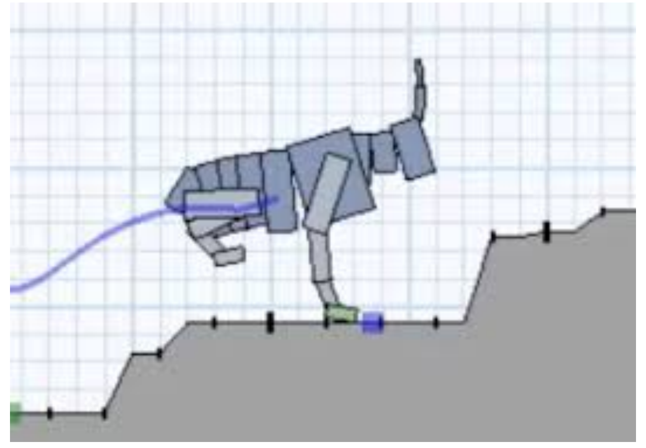
Mesh segmentation



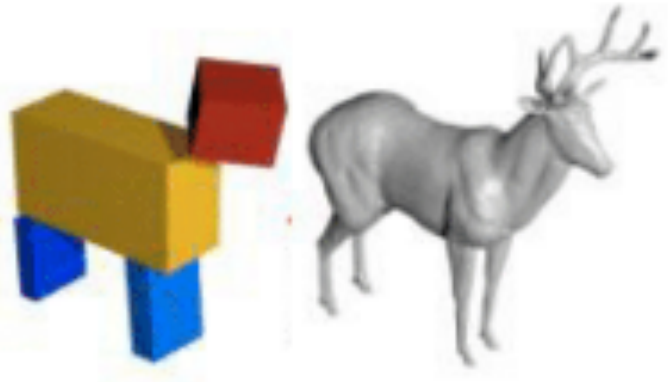
Learning deformations

BRDF estimation

Fluid



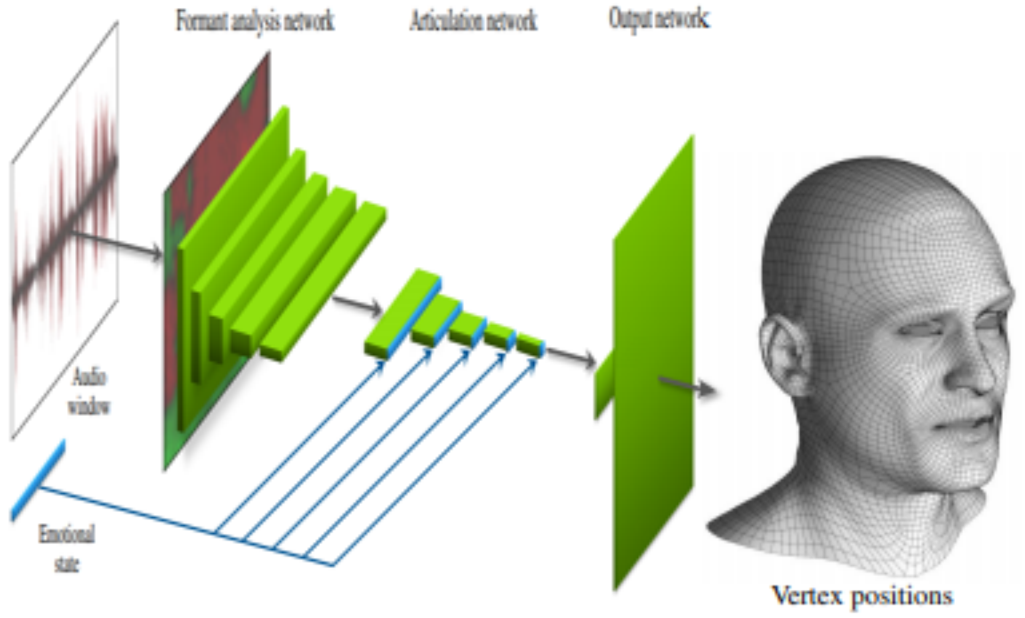
Animation



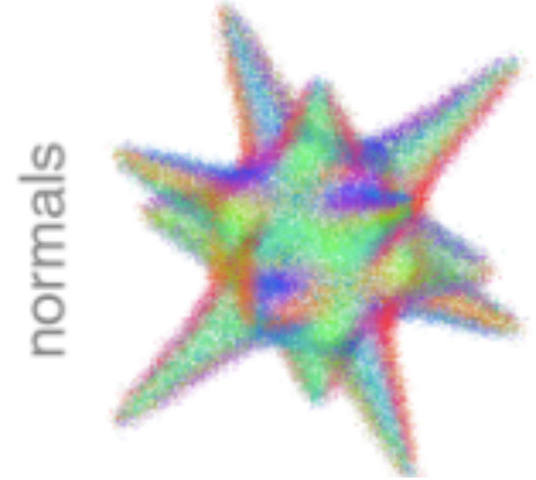
Boxification

Real-time rendering

Denoising

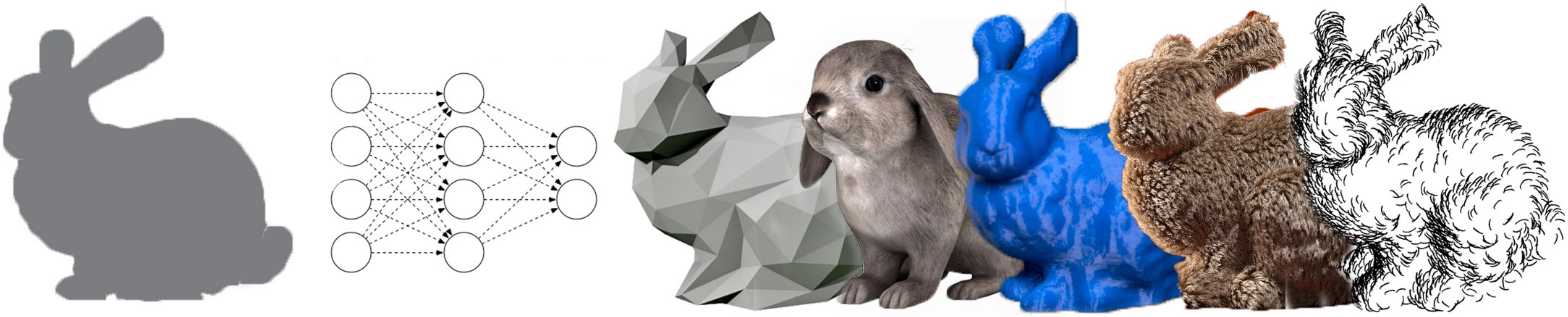


Facial animation



PCD processing

Course Information (slides/code/comments)



http://geometry.cs.ucl.ac.uk/dl_for_CG/

Scan me