Computer Vision

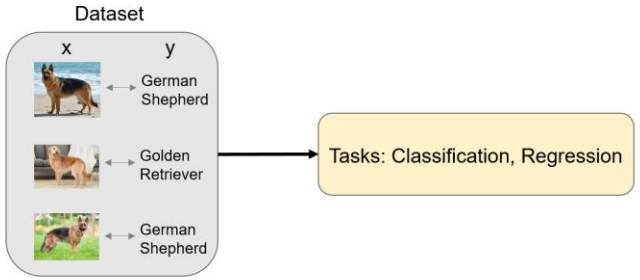
Lecture 5: Feature Extraction

Oliver Bimber

JOHANNES KEPLER
UNIVERSITY LINZ

Last Week: Machine Learning

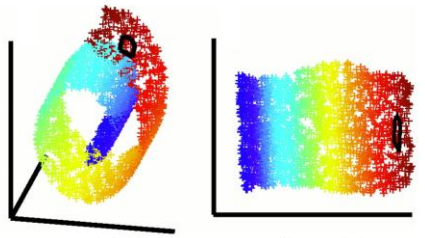
Supervised Learning



Use labeled Data for a Prediction Task.

Dimensionality Reduction

Given Data Points in *d* Dimensions, convert them to Data Points in *k*<*d* Dimensions with minimal loss of Information.

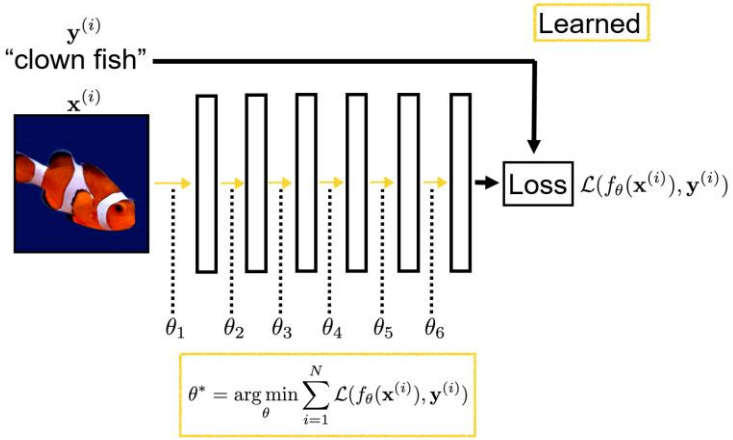


Original Data in 3D

Same data in 2D

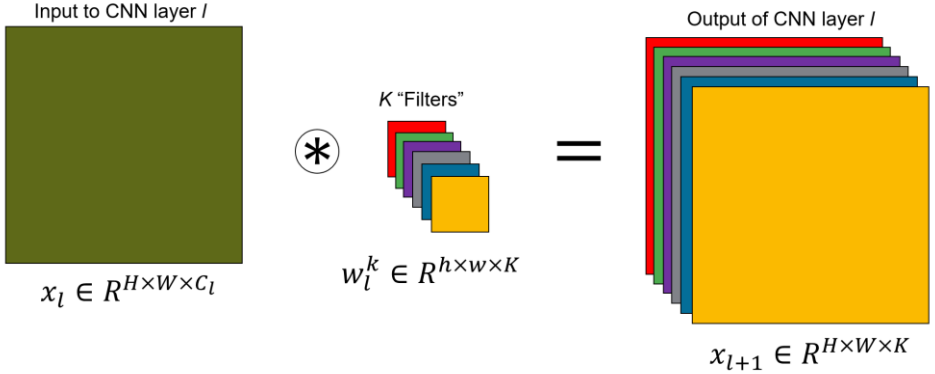
JOHANNES KEPLER UNIVERSITY LINZ

Deep Learning



JOHANNES KEPLER UNIVERSITY LINZ

Convolutional Neural Networks (CNNs)



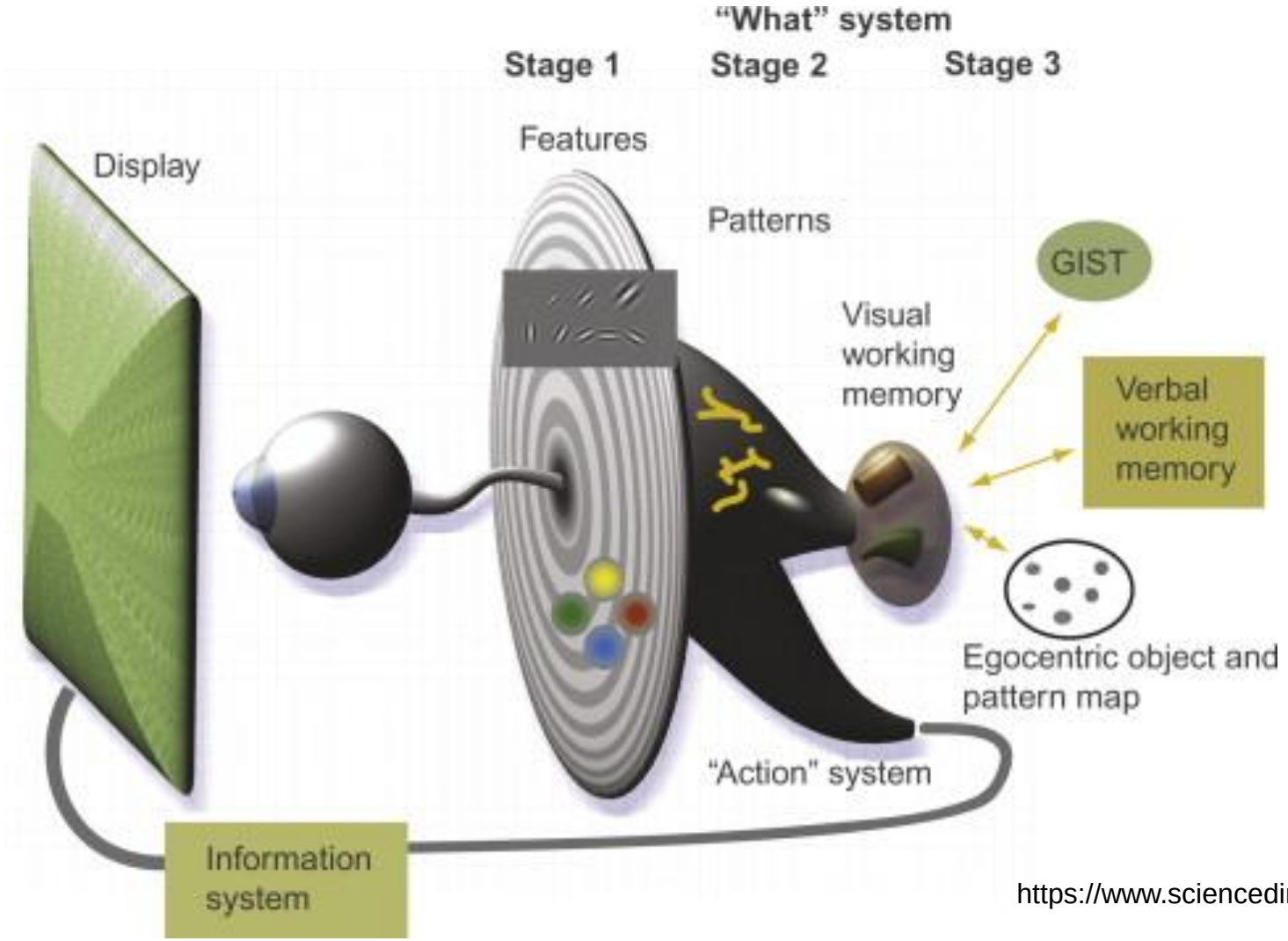
JOHANNES KEPLER UNIVERSITY LINZ

JOHANNES KEPLER UNIVERSITY LINZ

Course Overview

CW	Topic	Date	Place	Lab
41	Introduction and Course Overview	07.10.2025	Zoom	Lab 1
42	Capturing Digital Images	14.10.2025	Zoom	Lab 2
43	Digital Image Processing	21.10.2025	Zoom	Assignment 1
44	Machine Learning	28.10.2025	Zoom	
45	Feature Extraction	04.11.2025	Zoom	Open Lab 1
46	Segmentation	11.11.2025	Zoom	Assignment 2
47	Optical Flow	18.11.2025	Zoom	Open Lab 2
48	Object Detection	25.11.2025	Zoom	Assignment 3
49	Multi-View Geometry	02.12.2025	Zoom	Open Lab 3
50	3D Vision	09.12.2025	Zoom	Assignment 4
3	Trends in Computer Vision	13.01.2026	Zoom	
4	Q&A	20.01.2026	Zoom	Open Lab 4
5	Exam	27.01.2026	HS1 (Linz), S1/S3 (Vienna), S5 (Bregenz)	
9	Retry Exam	24.02.2026	tba	

Human Visual Processing



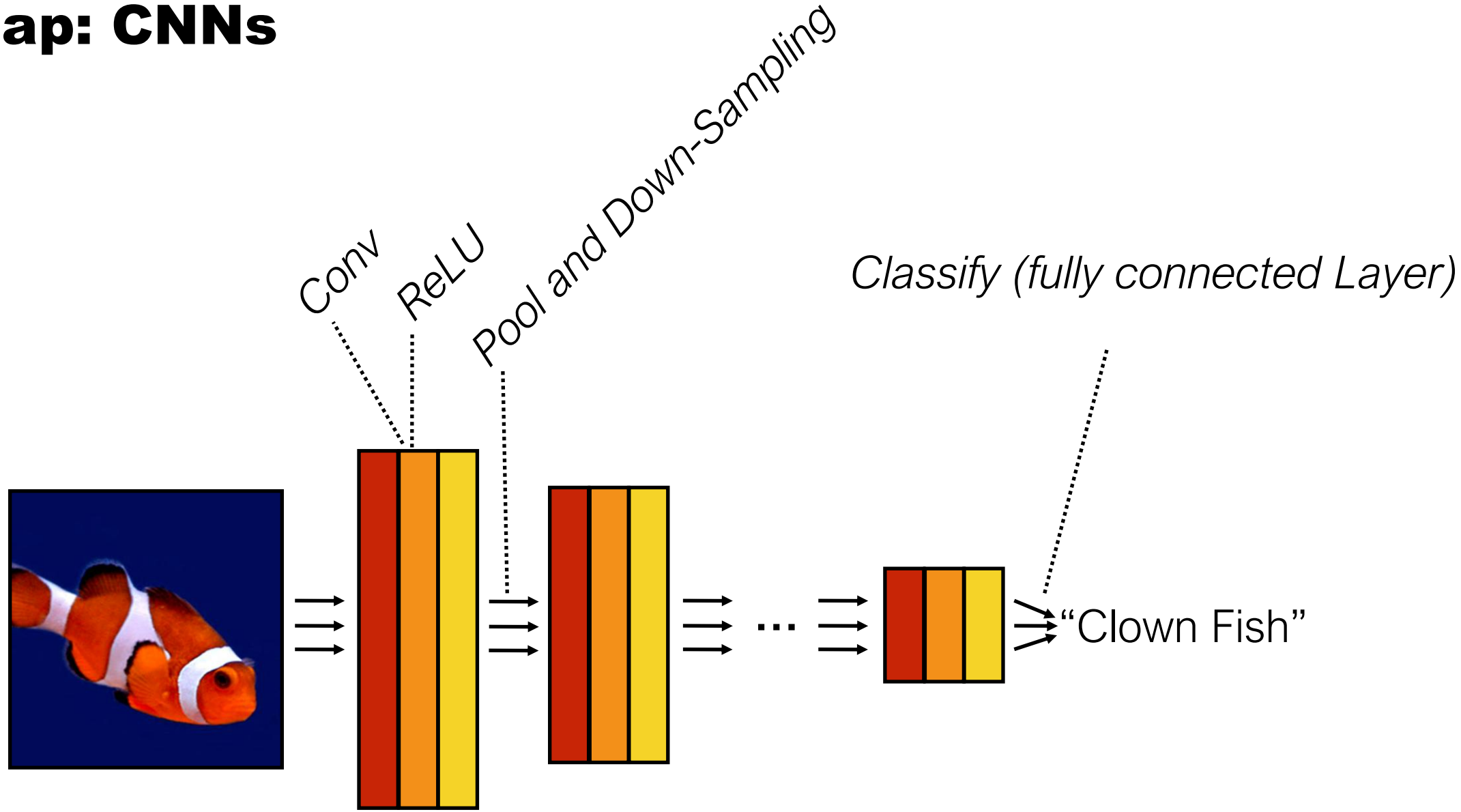
Early Visual Processing
The visual image is broken down into different kinds of features, particularly color differences, local edge and texture information, and local motion information.

Pattern Perception
Patterns are formed based on low-level features and on the task demands of visual thinking. Patterns consist of entities such as continuous contours, areas of a common texture, color, or motion.

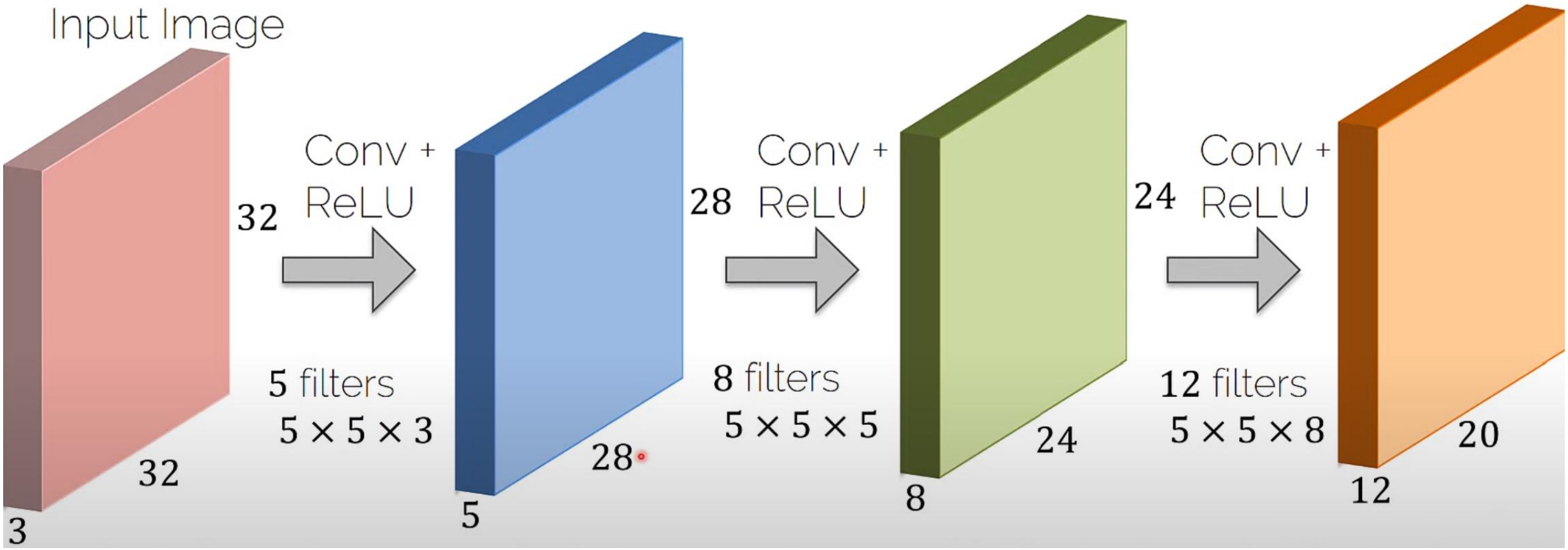
Working Memory
Only a few simple patterns can be held in working memory at any given instant. A visual query pattern can be held in working memory, forming the basis for active visual search through the direction of attention. Attention controls what visual information is held and stored. The semantic meaning or gist of an object or scene can be activated. For items to be processed into long-term memory, deeper semantic coding is needed (and sleep).

https://www.sciencedirect.com/topics/computer-science/visual-processing

Recap: CNNs


Conv
ReLU
Pool and Down-Sampling
Classify (fully connected Layer)
...
"Clown Fish"

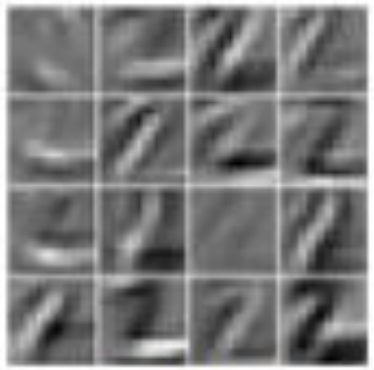
Recap: Convolutional Layers



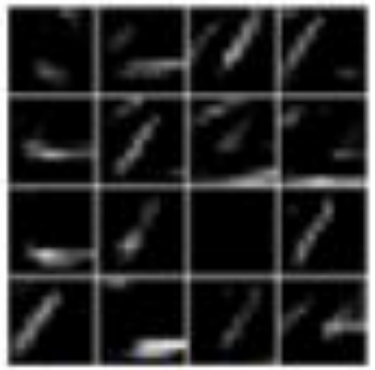
Recap: Activation (Feature) Maps

conv1

relu1



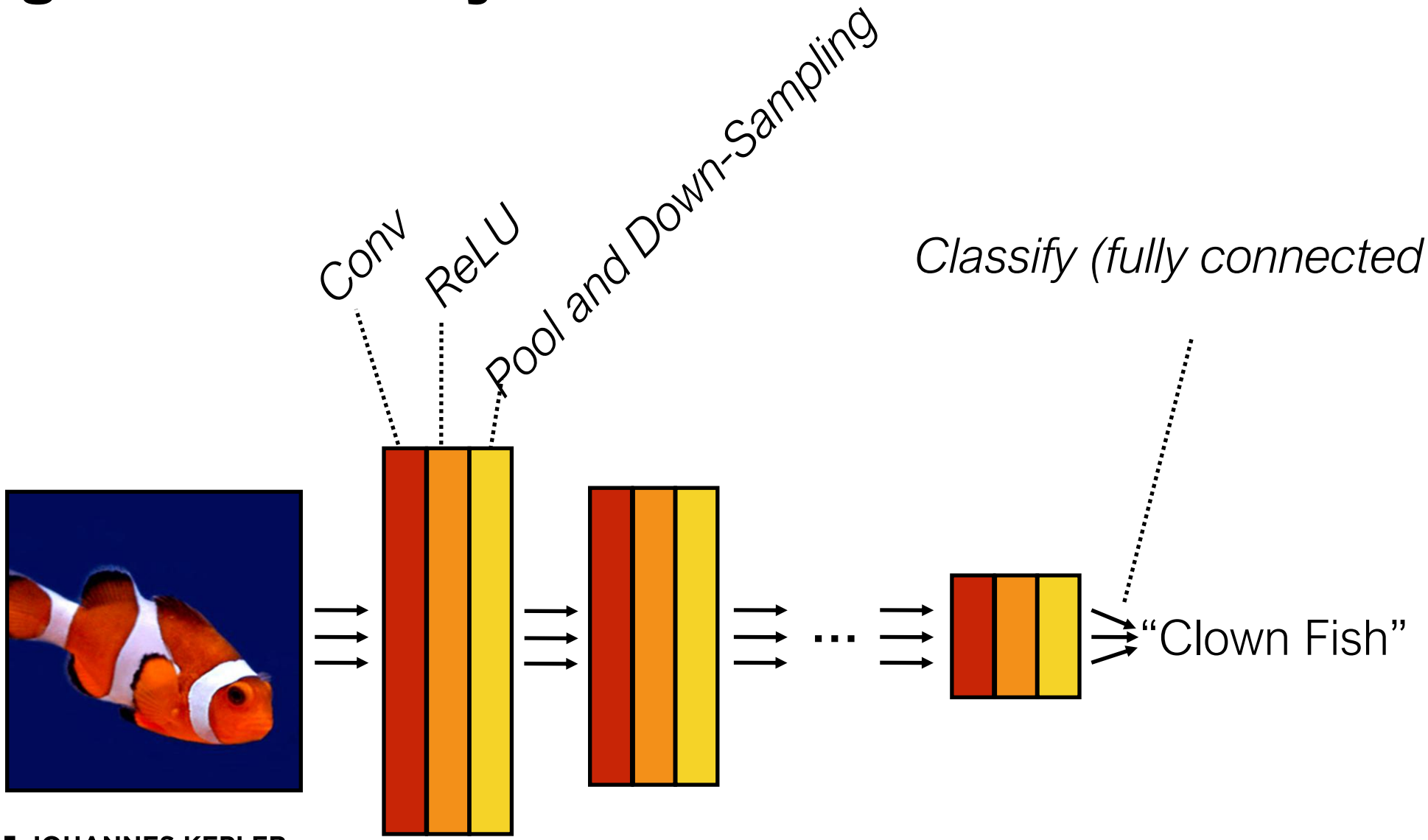
conv2



relu2

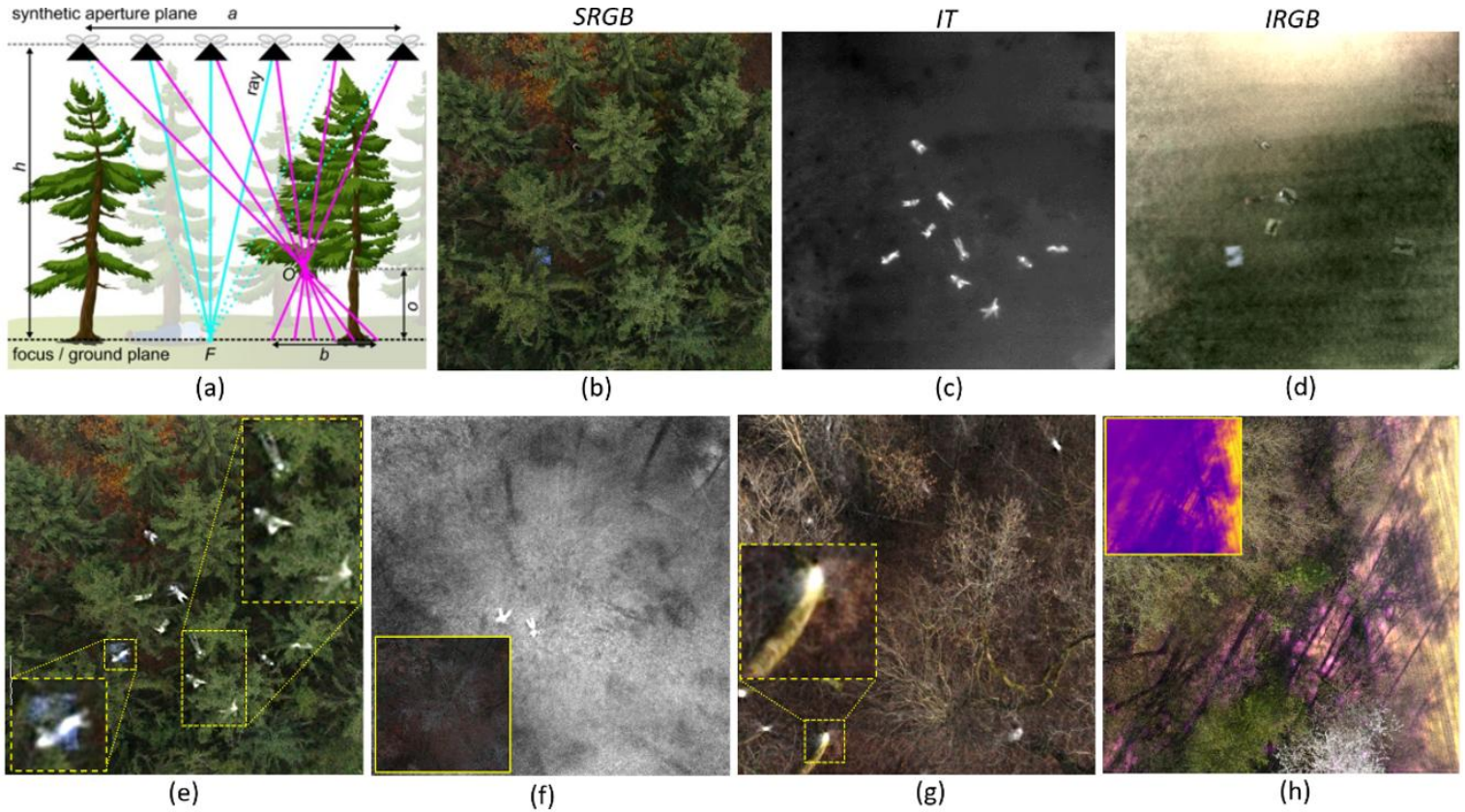
- Each Layer can be thought of as a set of C **Feature Maps** aka **Channels**
- Each Feature Map is an NxM Image
- **ReLU** (Rectified Linear Unit, *f(x)=max(0,x)*) Activation Function is used to introduce Nonlinearity in a Neural Network, helping mitigate the Vanishing Gradient Problem

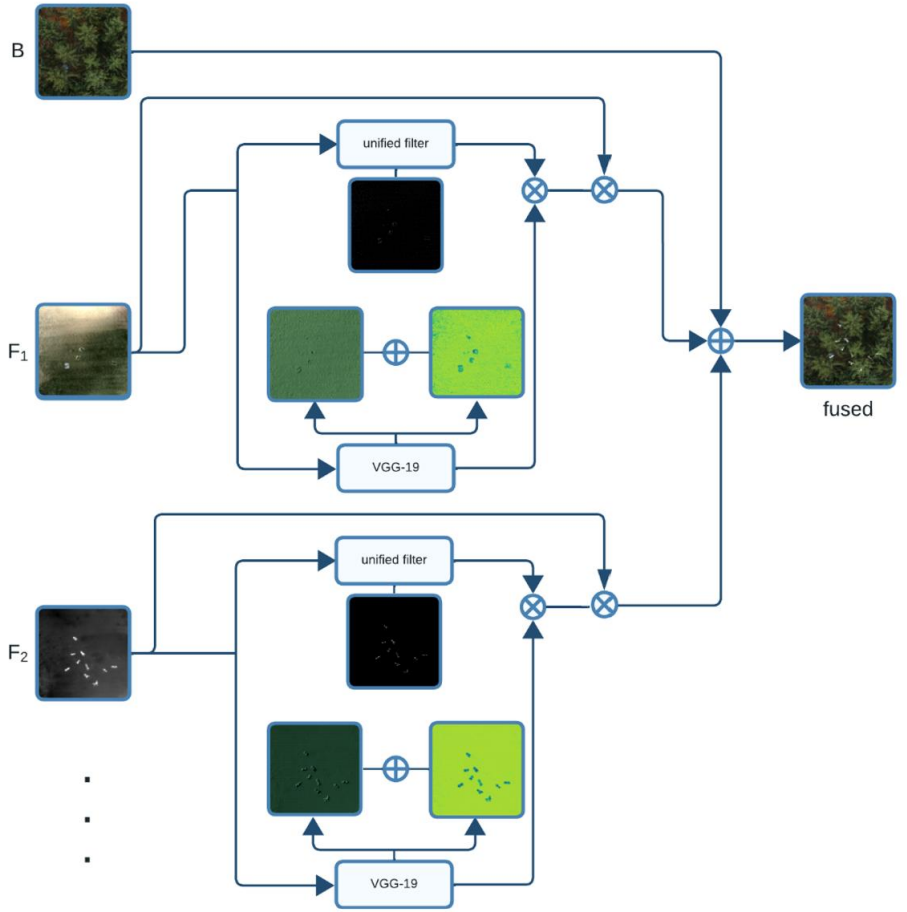
Using Features Beyond Classification



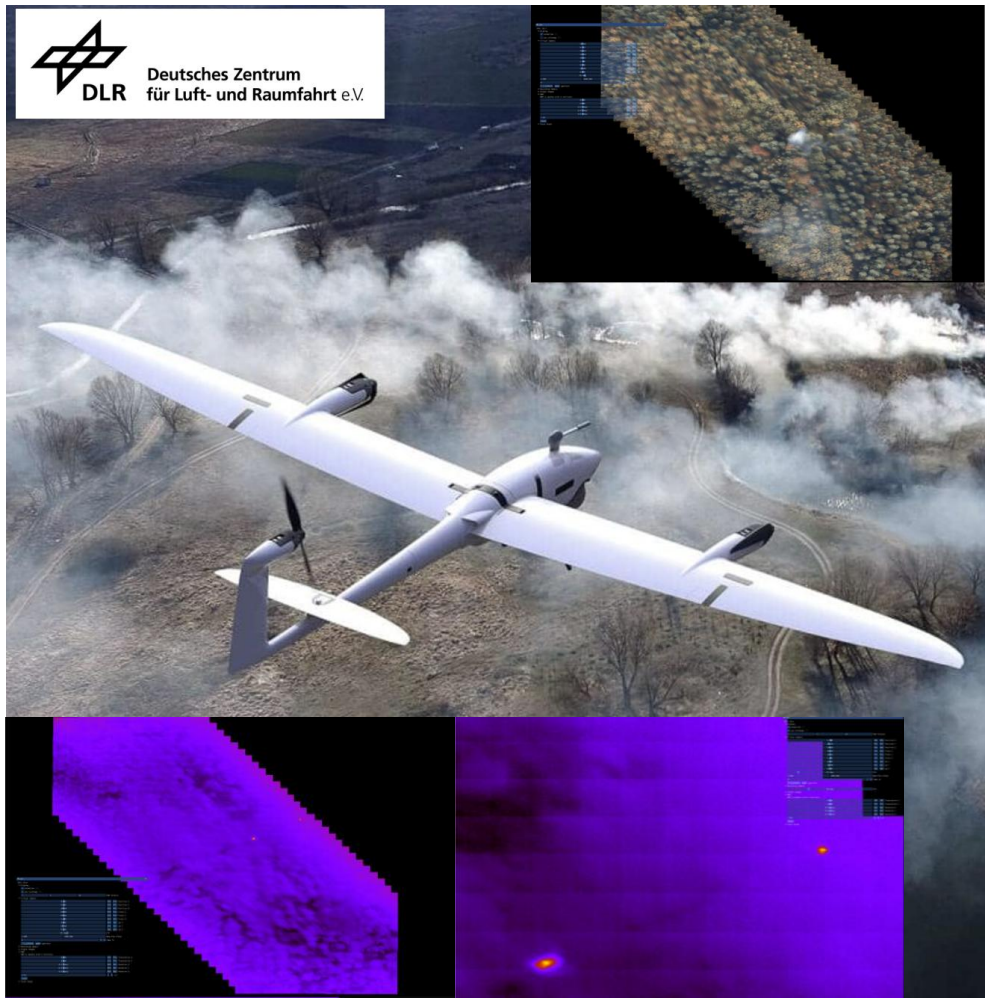
Airborne Optical Sectioning

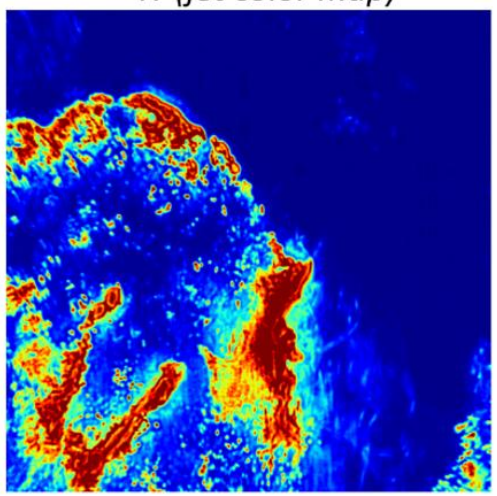
Research Example: Image Fusion with VGG-Net Features

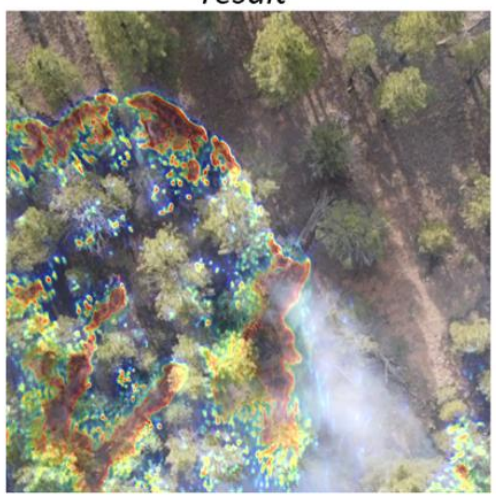


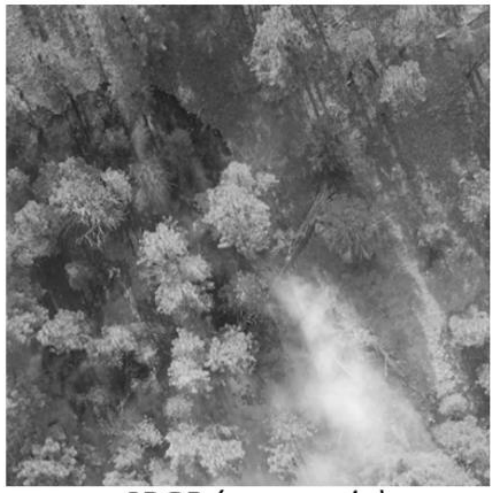


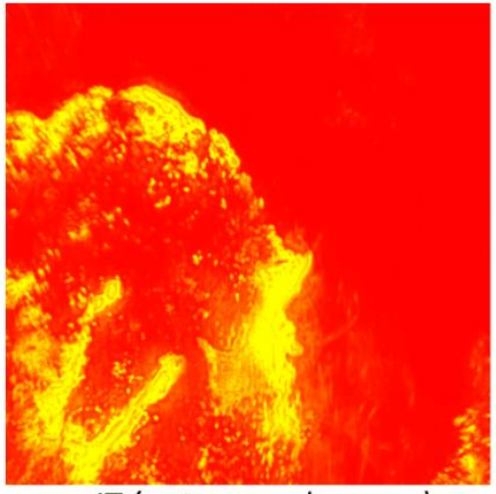
Research Example: Image Fusion with VGG-Net Features

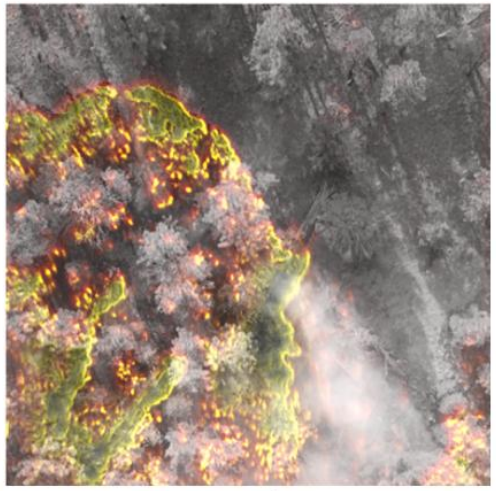






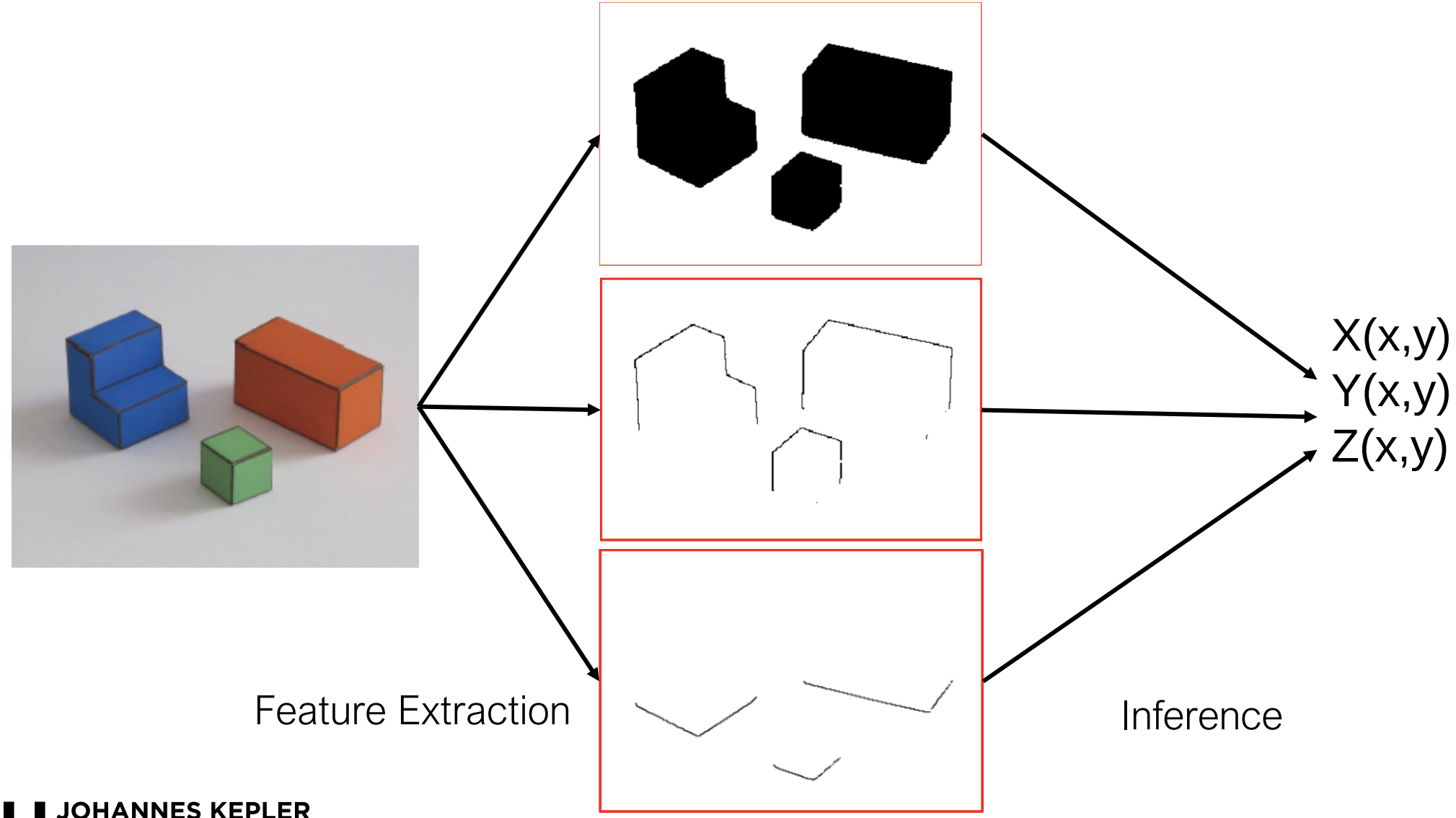




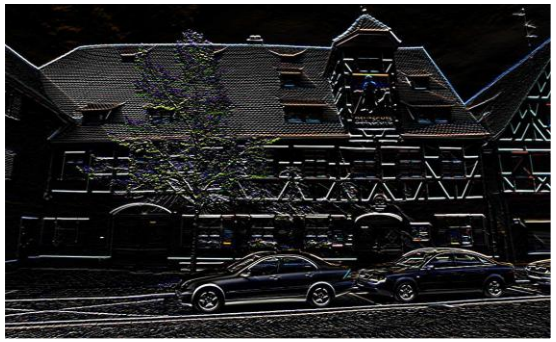


JOHANNES KEPLER
UNIVERSITY LINZ

Role of Image Features in Computer Vision



What are Image Features



$$\frac{\partial f(x,y)}{\partial x} \approx f(x+1,y) - f(x-1,y)$$

$$H = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

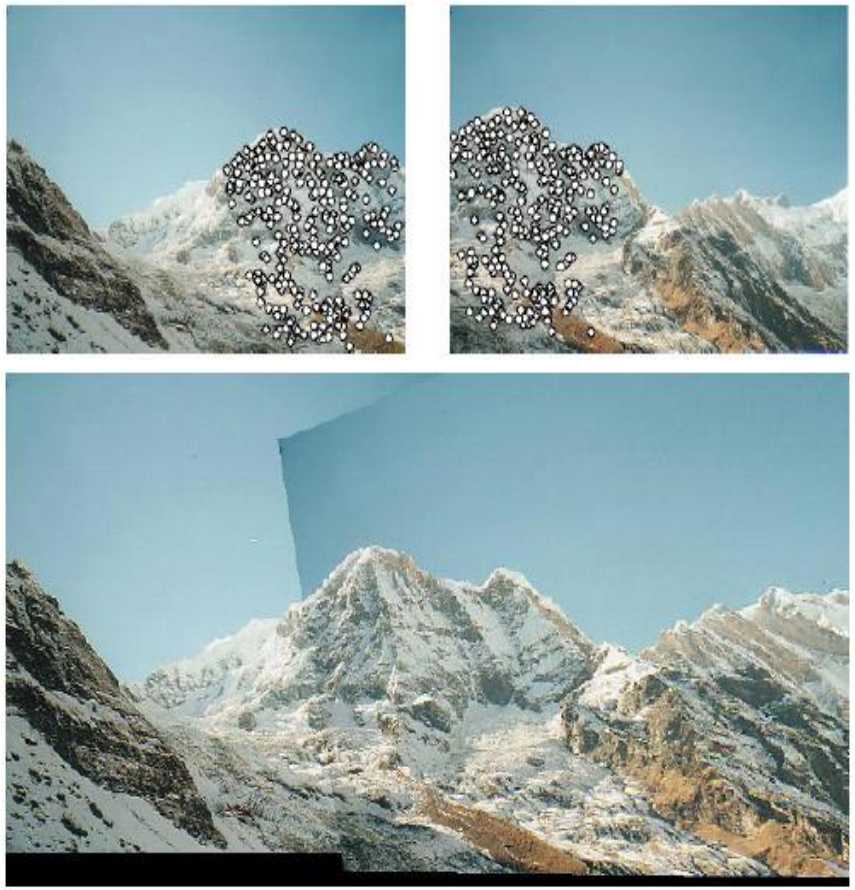
$$\frac{\partial f(x,y)}{\partial y} \approx f(x,y+1) - f(x,y-1)$$

$$H = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad H = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

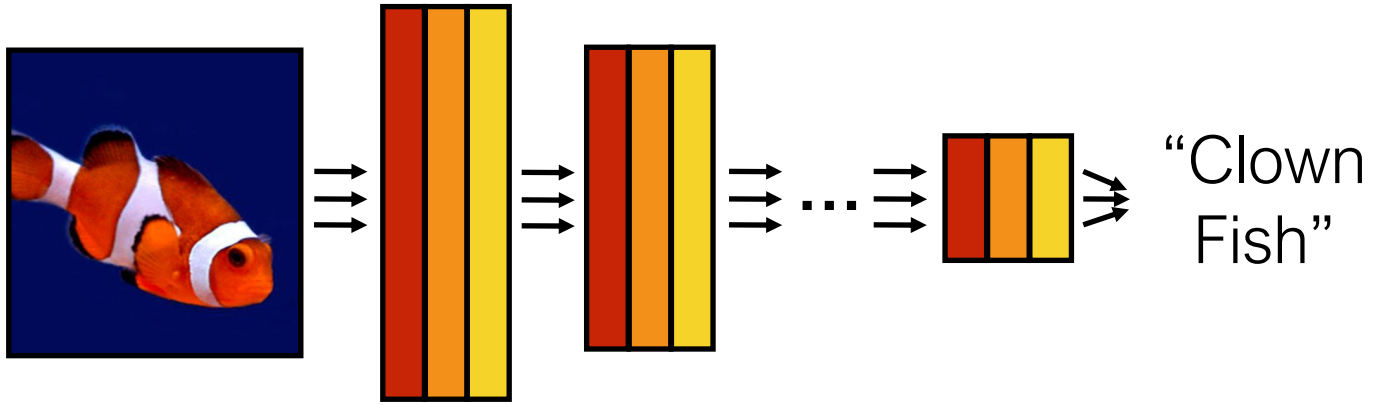
Example: Partial Derivation

- Not every Pixel in an Image is equally important to understand its Content
- Image Features mark those Pixels that are important
- Convolution Kernels can be used to identify them
- But what are good (i.e., invariant) Features and what are the Kernels to get them?

What do we need Image Features for?



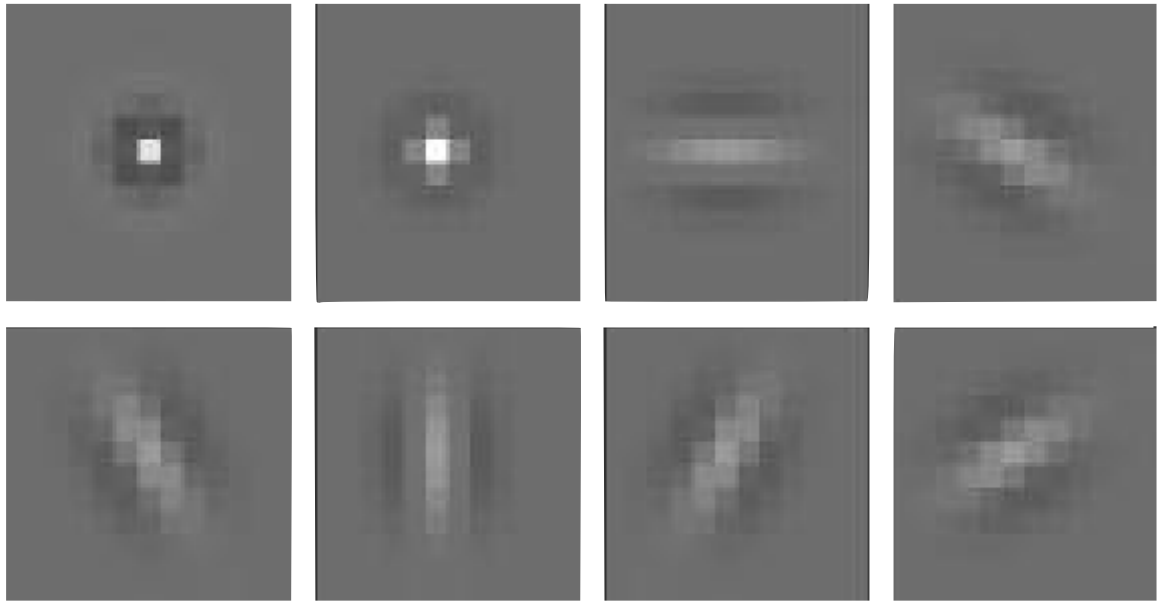
Example: Image Registration



Example: Classification

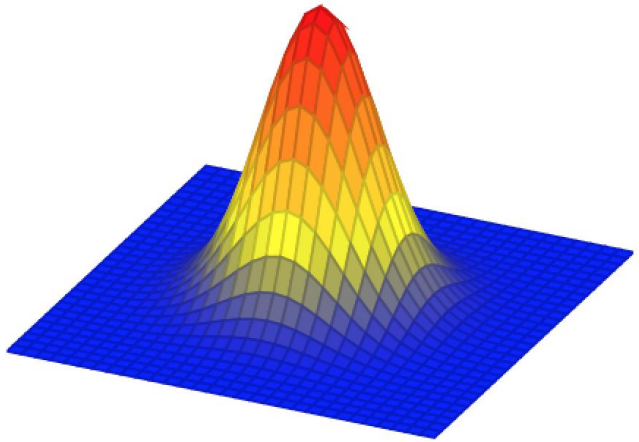
Model-Based vs. Learning-Based Feature Extraction

- **Model-Based:**
  - manually designed, general Filter Kernels (e.g., adapted to the Human Visual System -- spots and Bars at different Orientations)
- **Learning-Based:**
  - Application specific Filter Kernels are automatically learned from Training Data (e.g., CNN)

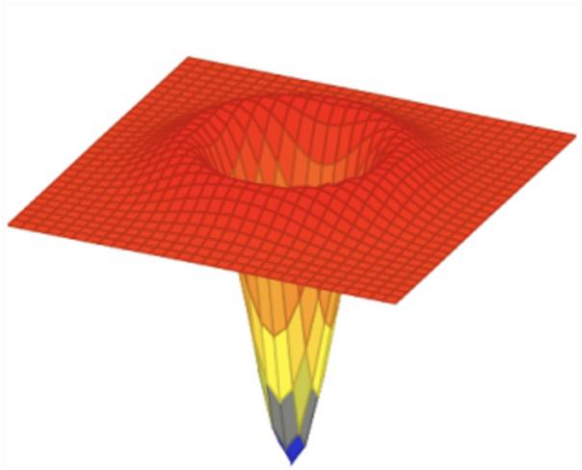


Filter Bank of two Spot and six Bar Filters

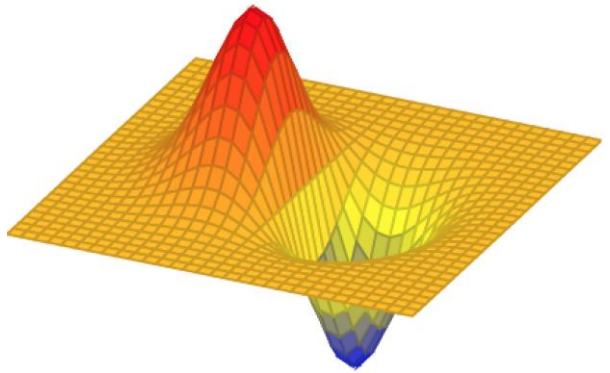
Shift-Invariance (Lots of useful Kernels)



Gaussian



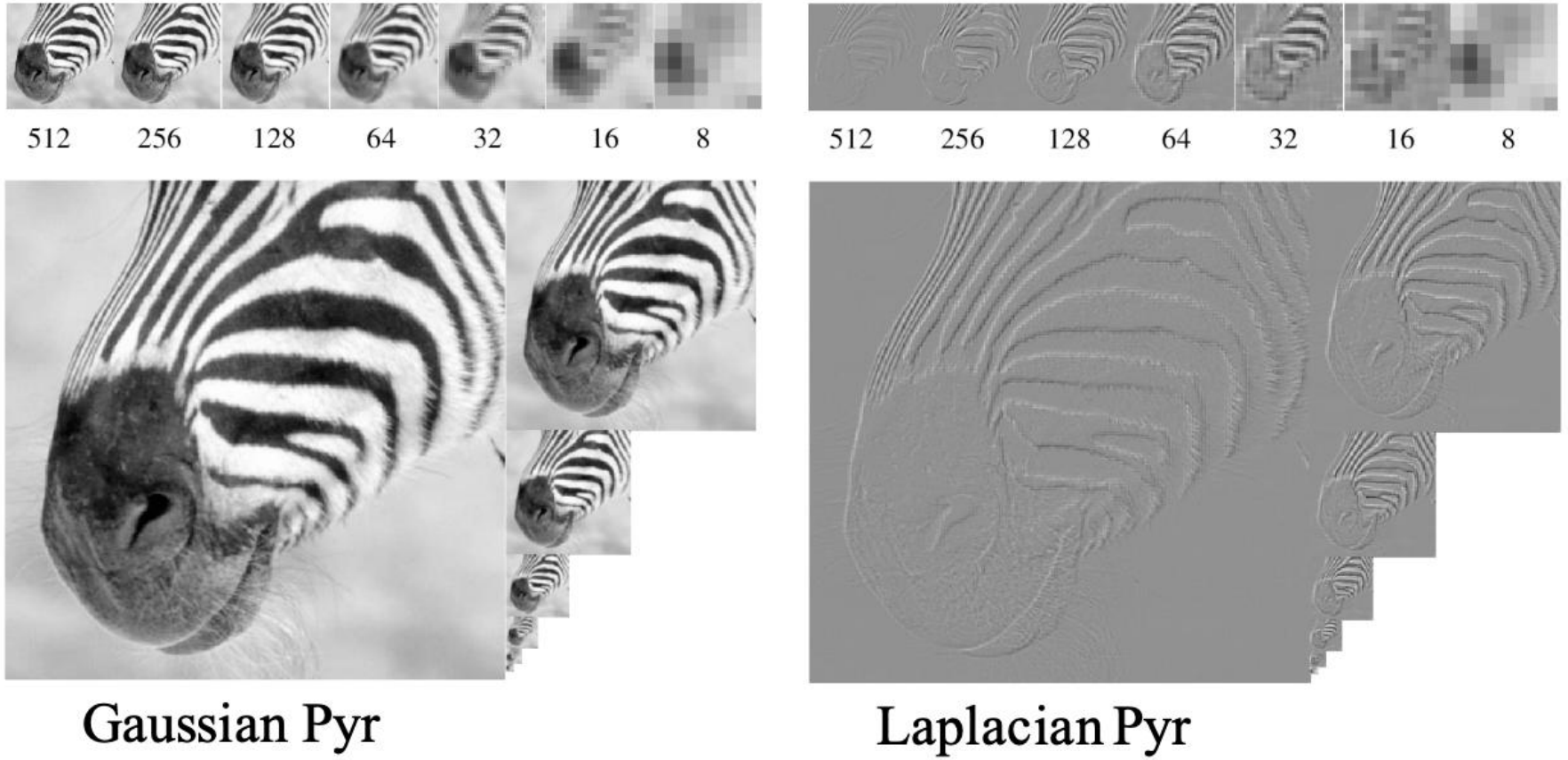
Laplacian



Gaussian Derivative

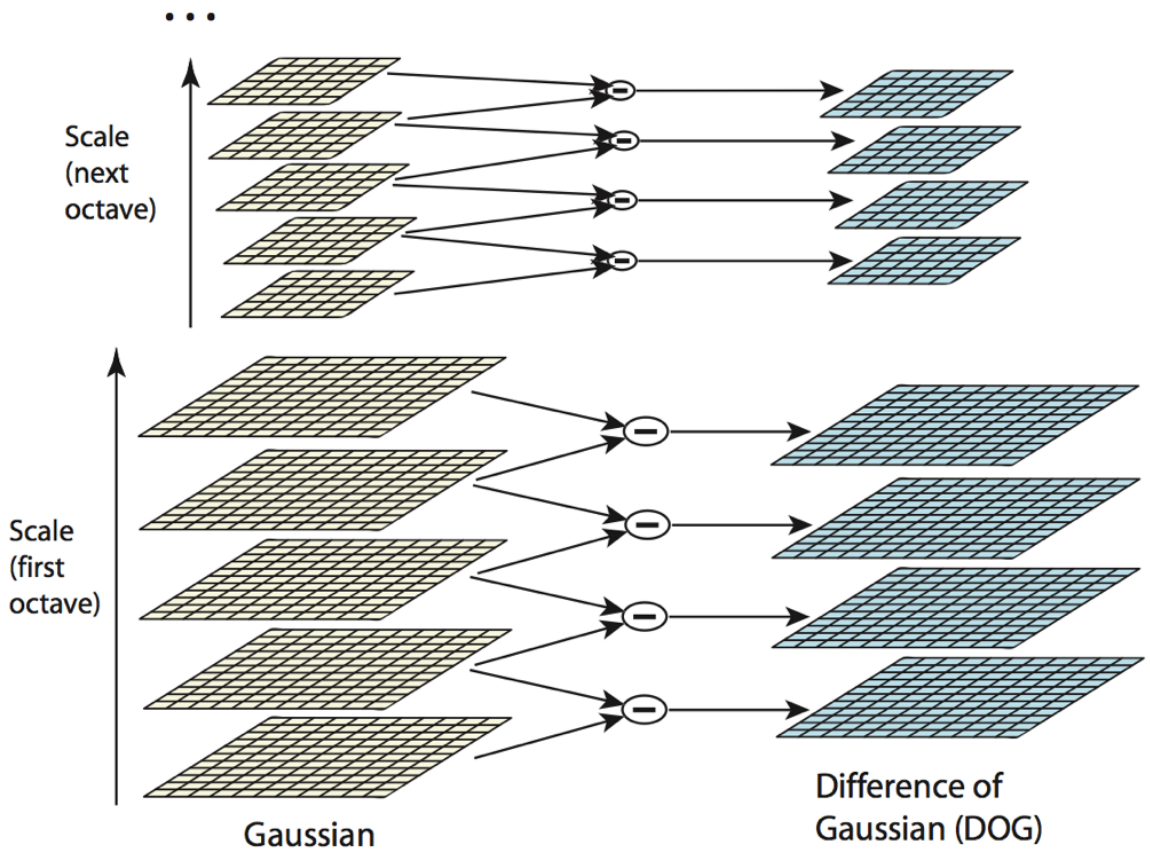
…and many more…

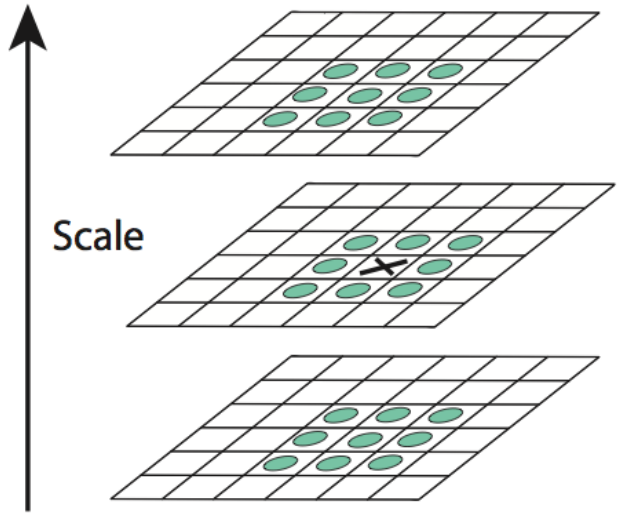
Scale-Invariance (Lots of Image Pyramids)



…and many more…

Example: Scale-Invariant Feature Transform (SIFT)





(1) Compute Scale Space (Difference of Gaussian, DoG = Laplacian Image Pyramid)

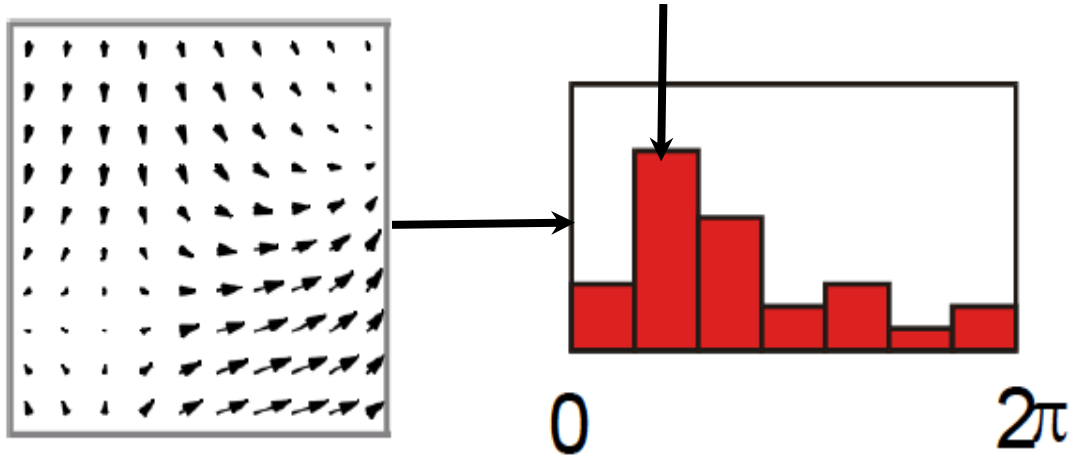
(2) Detect local Extrema in DoG Pyramid Layers (i.e., high Gradients wrt. Neighborhood)

Example: Scale-Invariant Feature Transform (SIFT)

unfiltered

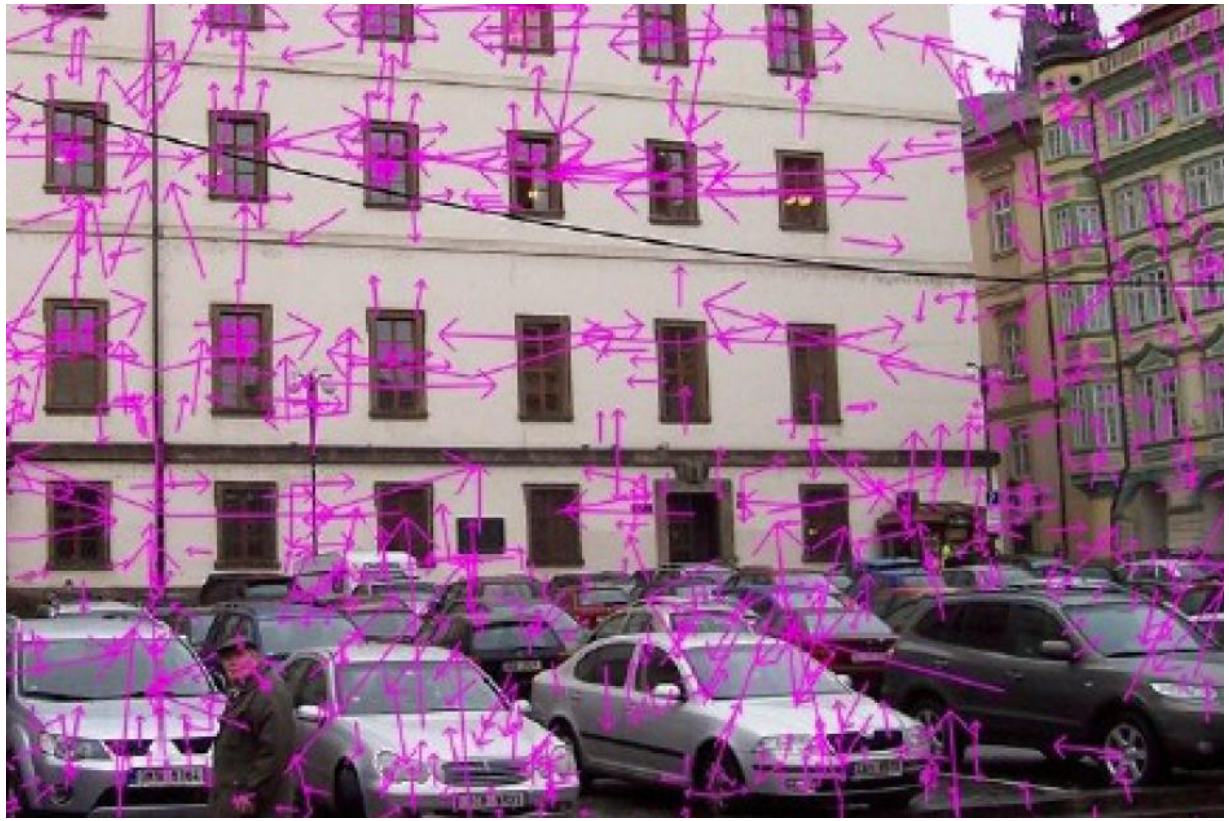
filtered

(3) Filter out Features in low-contrast Regions (Noise)

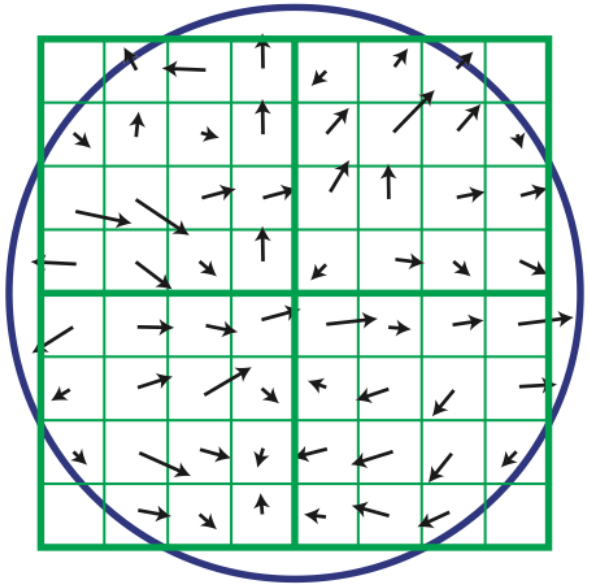


(4) Determine Feature (i.e., Gradient) Orientations and sort them into Histogram (largest Bin = main Orientation)

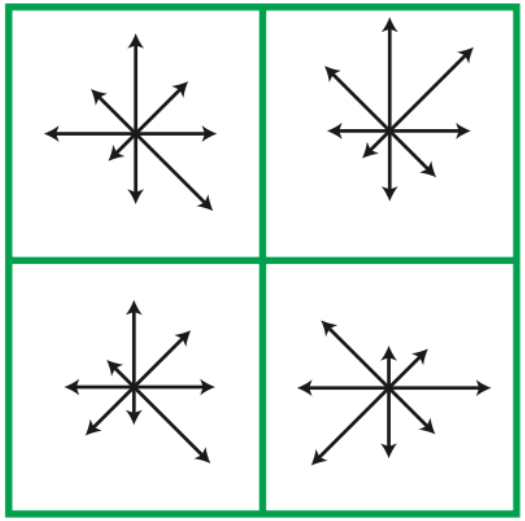
Example: Scale-Invariant Feature Transform (SIFT)



Assignment of Main Orientations to Feature Points



4 image tiles (4x4 Pixels each)

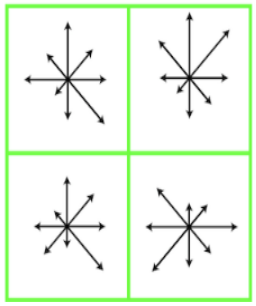


Orientation Histograms (8 bins) for each Tile

(5) Define Descriptor for each Feature. Example: for 4 neighboring Image Tiles (each 4x4 Pixels) determine Orientation Histogram (e.g., 8 Bins) → 4x8=32 Entries of Feature Descriptor (SIFT used 4x4=16 Tiles instead of 2x2=4 Tiles → 16x8=128 Entries of Feature Descriptor)

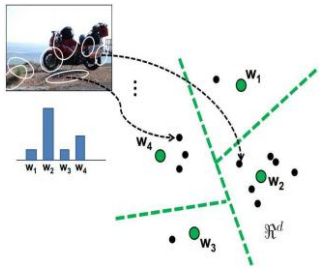
Model-Based Classification

Low-Level Features

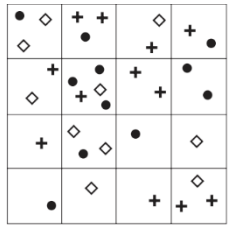


Features
(SIFT,…)

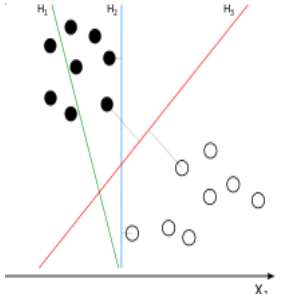
Mid-Level Features



Clustering



Pooling



Classification

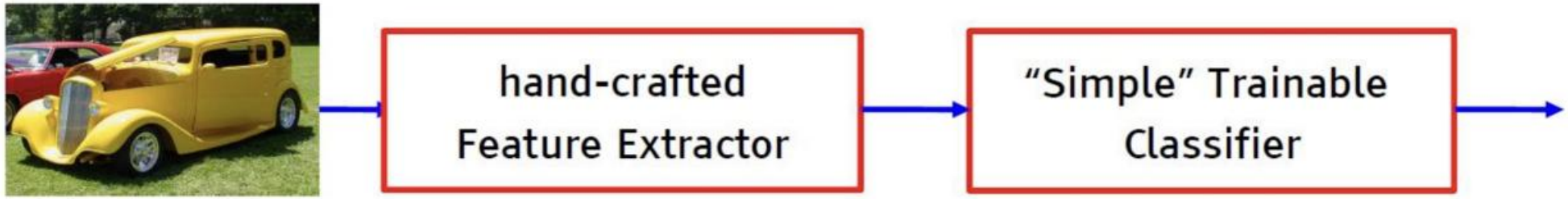
Cat

Feature Representation

Problem: Manually constructed Features. Can we learn them from Data?

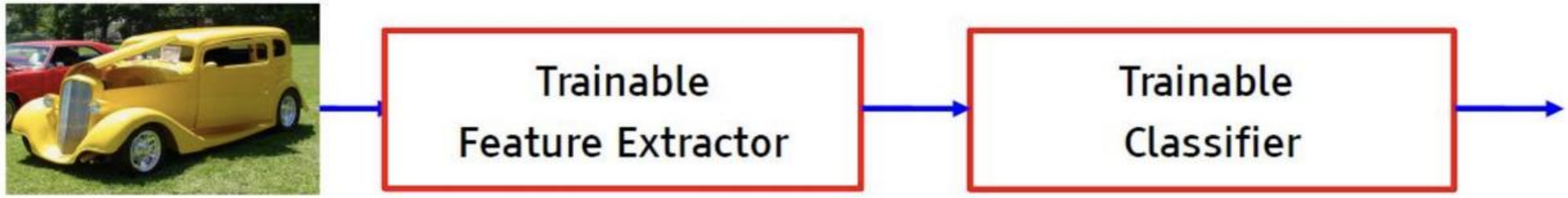
Model-Based vs. Learning-Based Feature Extraction

- Fixed engineered features (or kernels) + trainable classifier

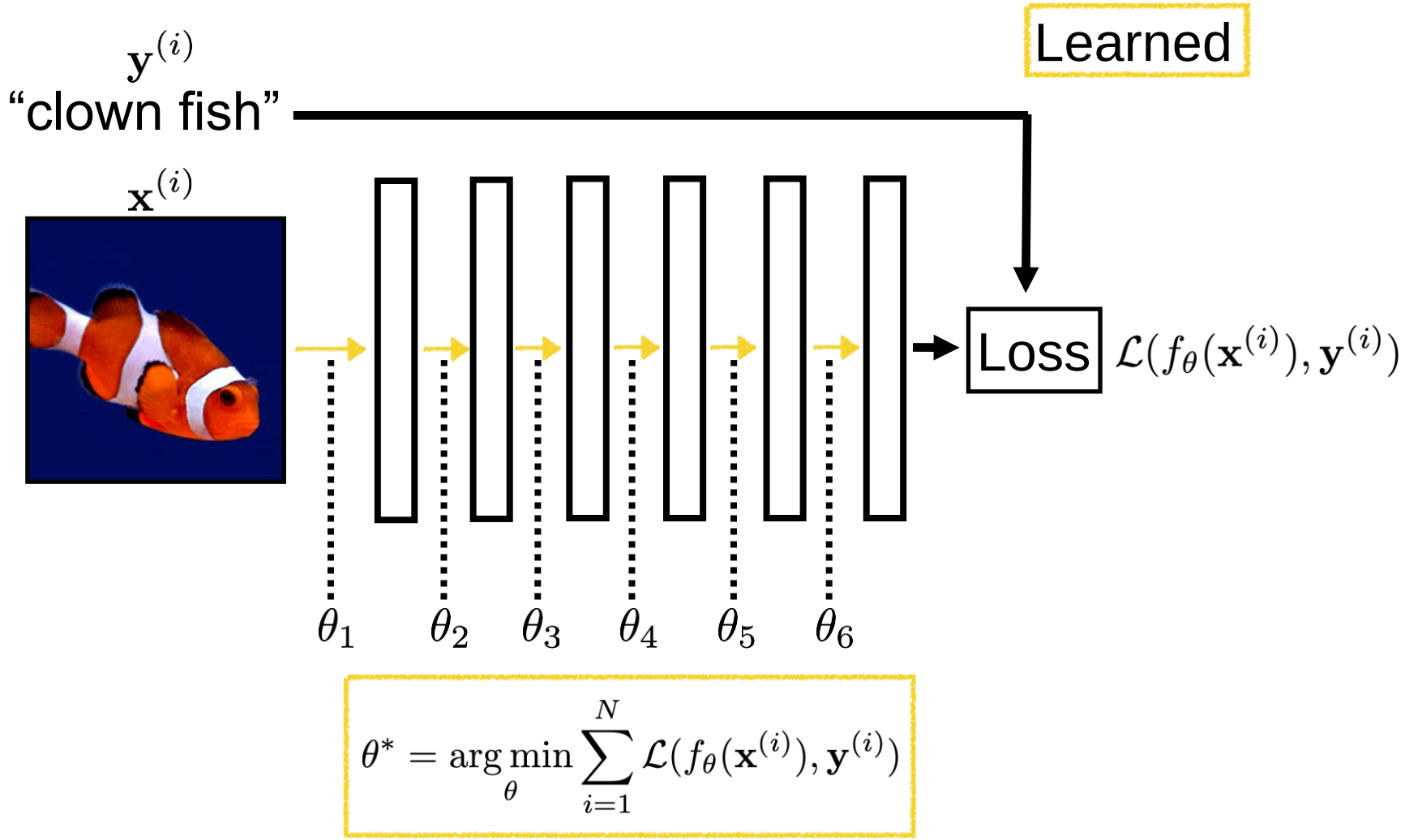


VS.

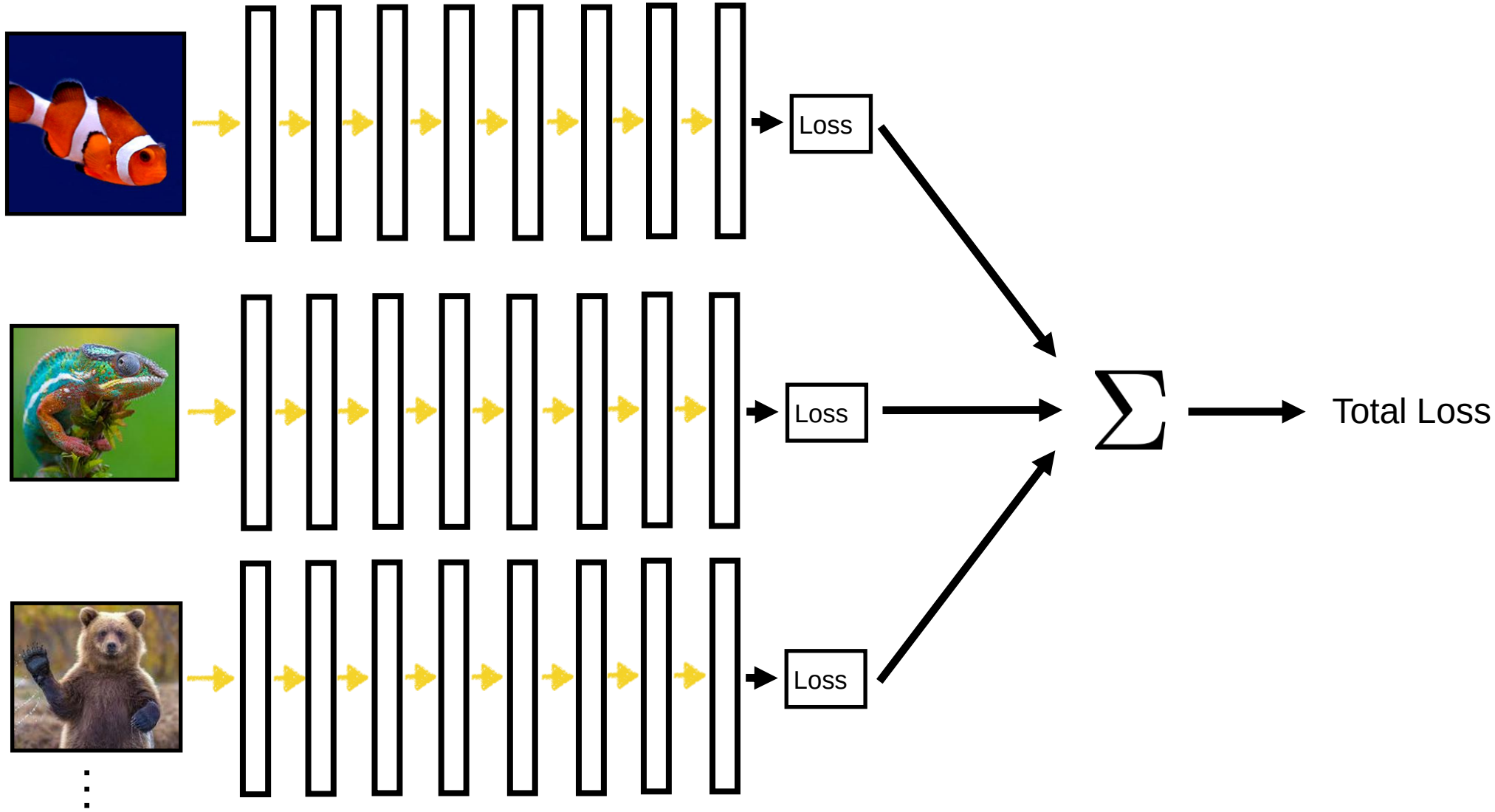
- End-to-end learning / feature learning / deep learning



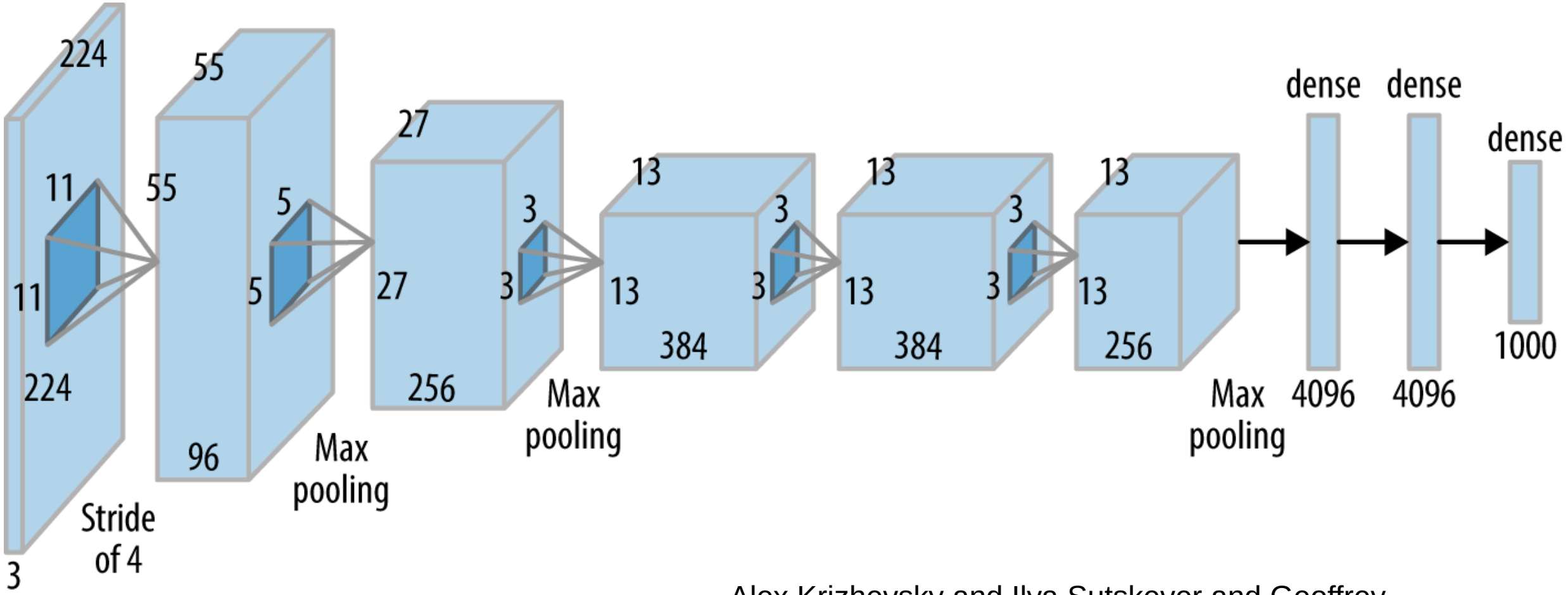
Feature Extraction (for Classification) with Deep NNs



Feature Extraction (for Classification) with Deep NNs

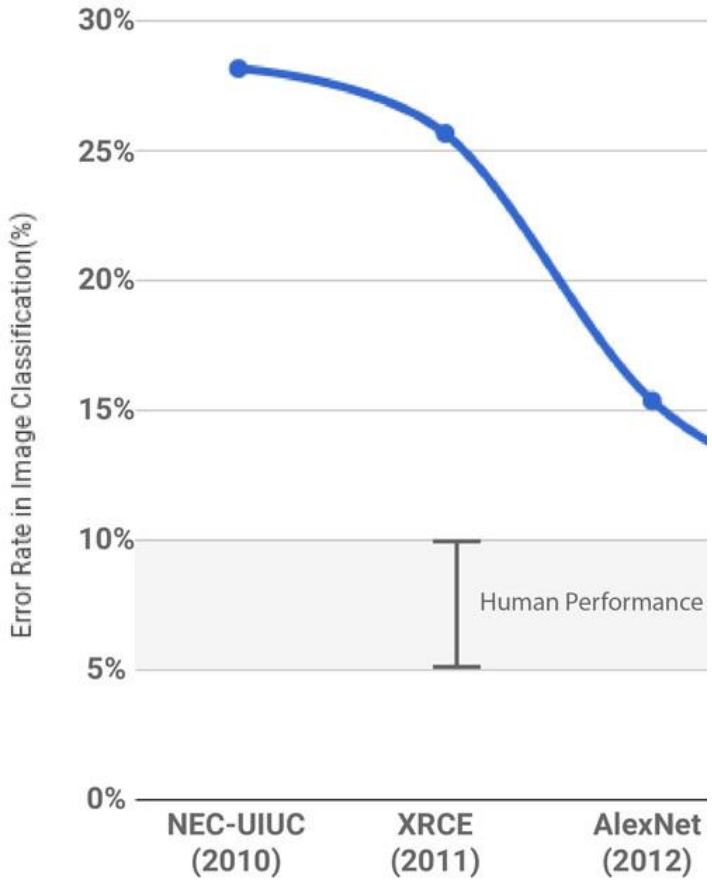


Example: AlexNet (2012)



Alex Krizhevsky and Ilya Sutskever and Geoffrey

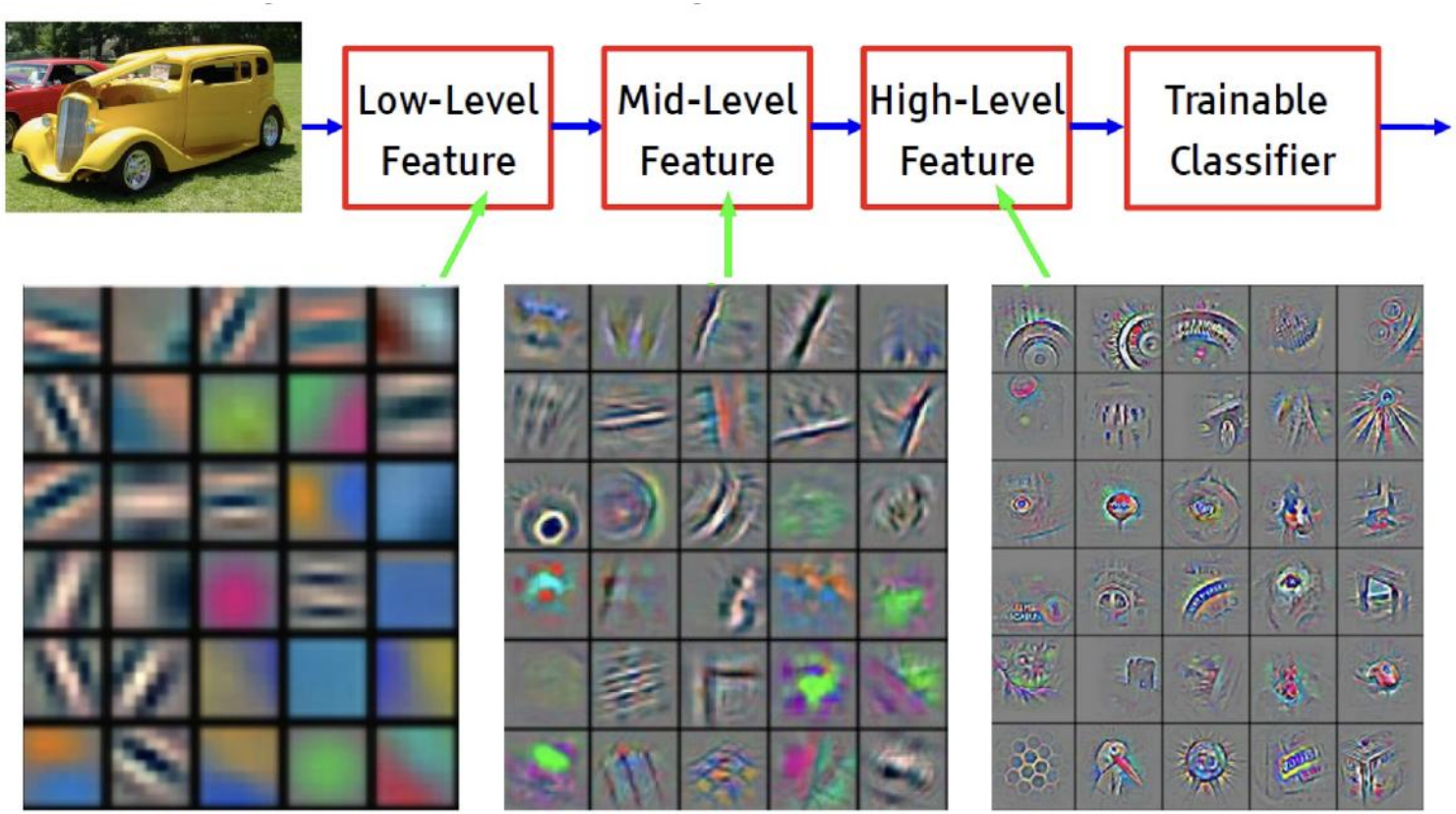
Development of Architectures



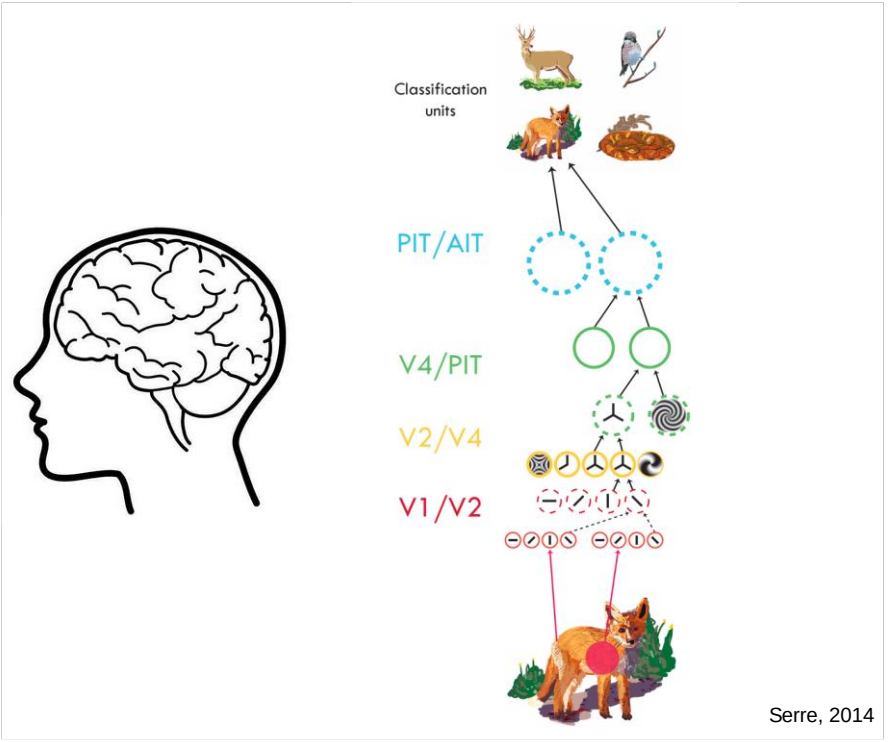
Decline in top 5 Error Rate for Neural Network Architecture for Image Classification

Top 5 Error Rate = correct Label not among top 5 Classifications

From Low-Level to High-Level Features



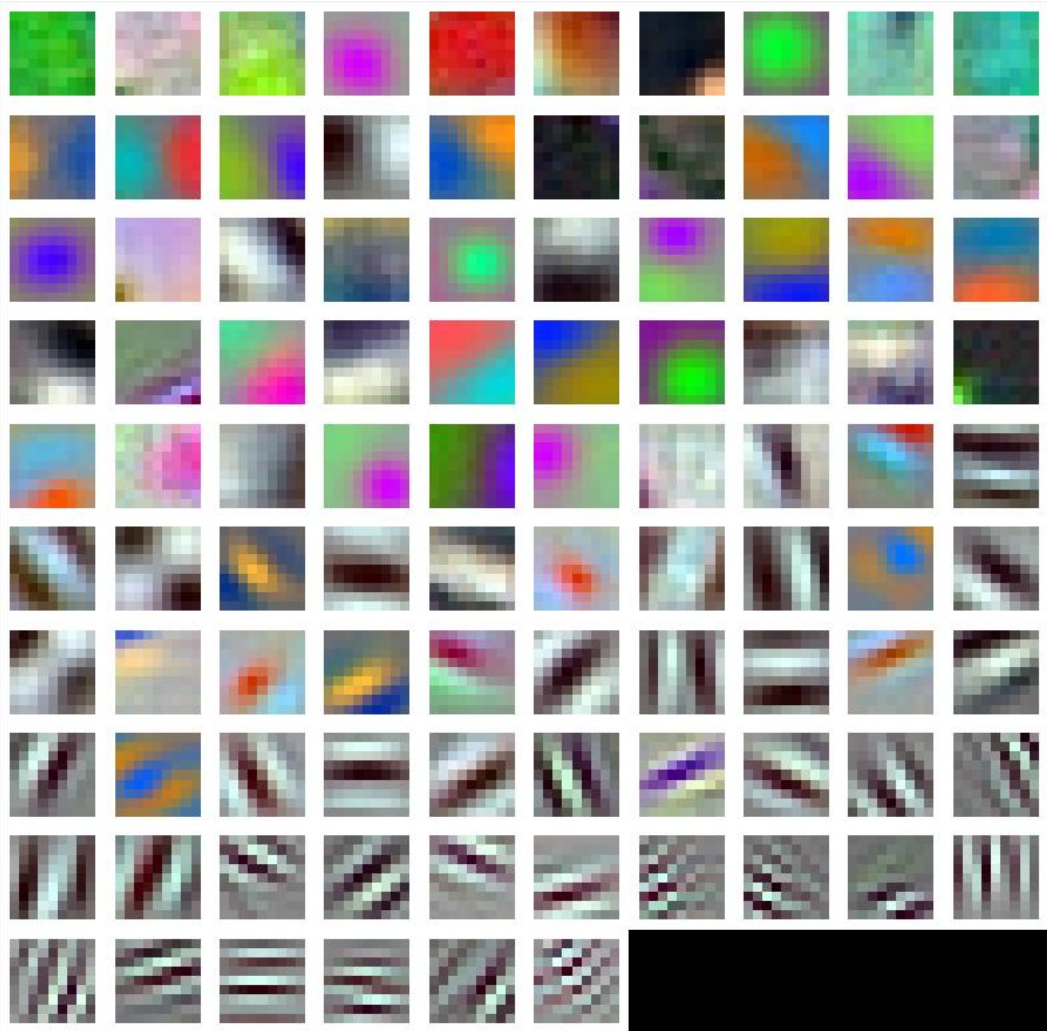
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]



Serre, 2014

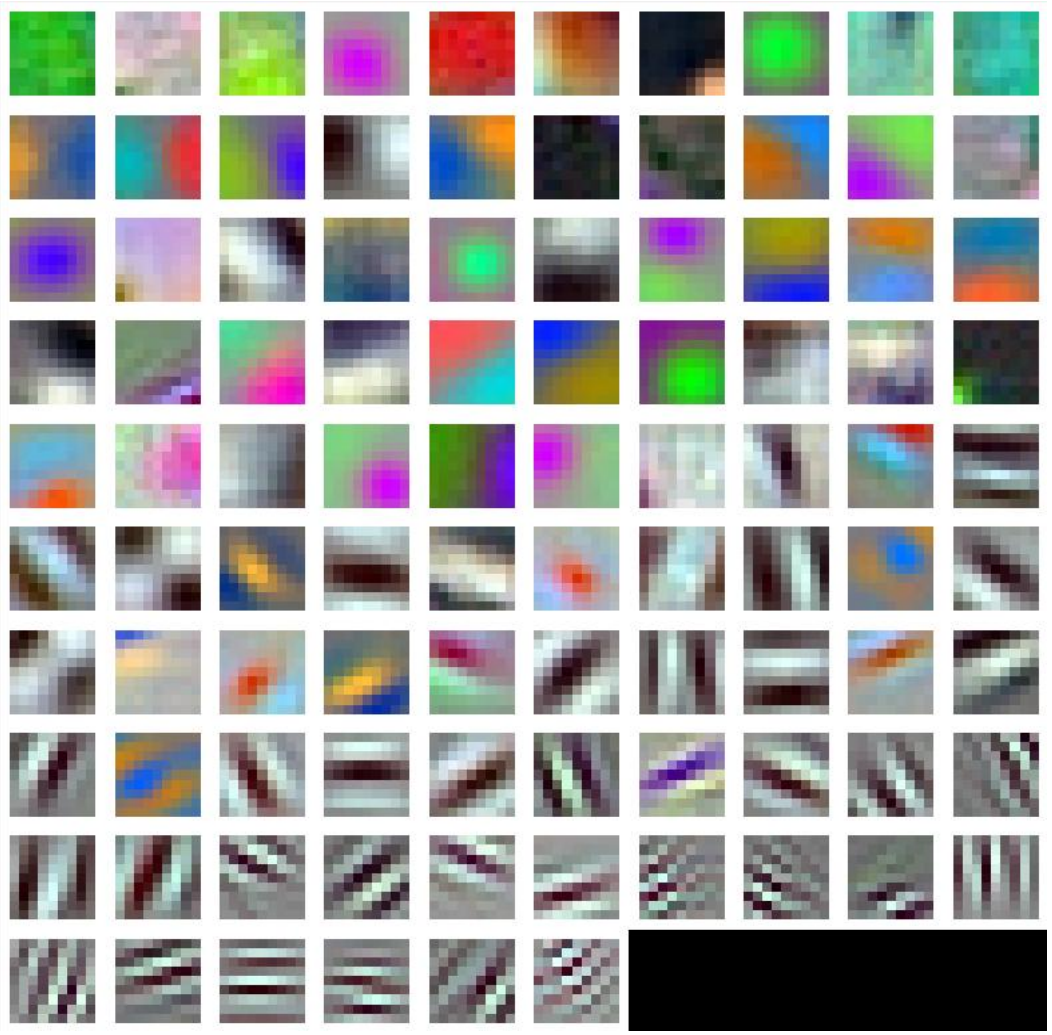
See inbuilt feature map visualization (PyTorch / TensorFlow), Lucid, Captum, tf-explain, TorchCAM, Keras-vis, TensorBoard, Ne, tronDeep Visualization Toolbox...

Example: 96 Kernels in Conv1

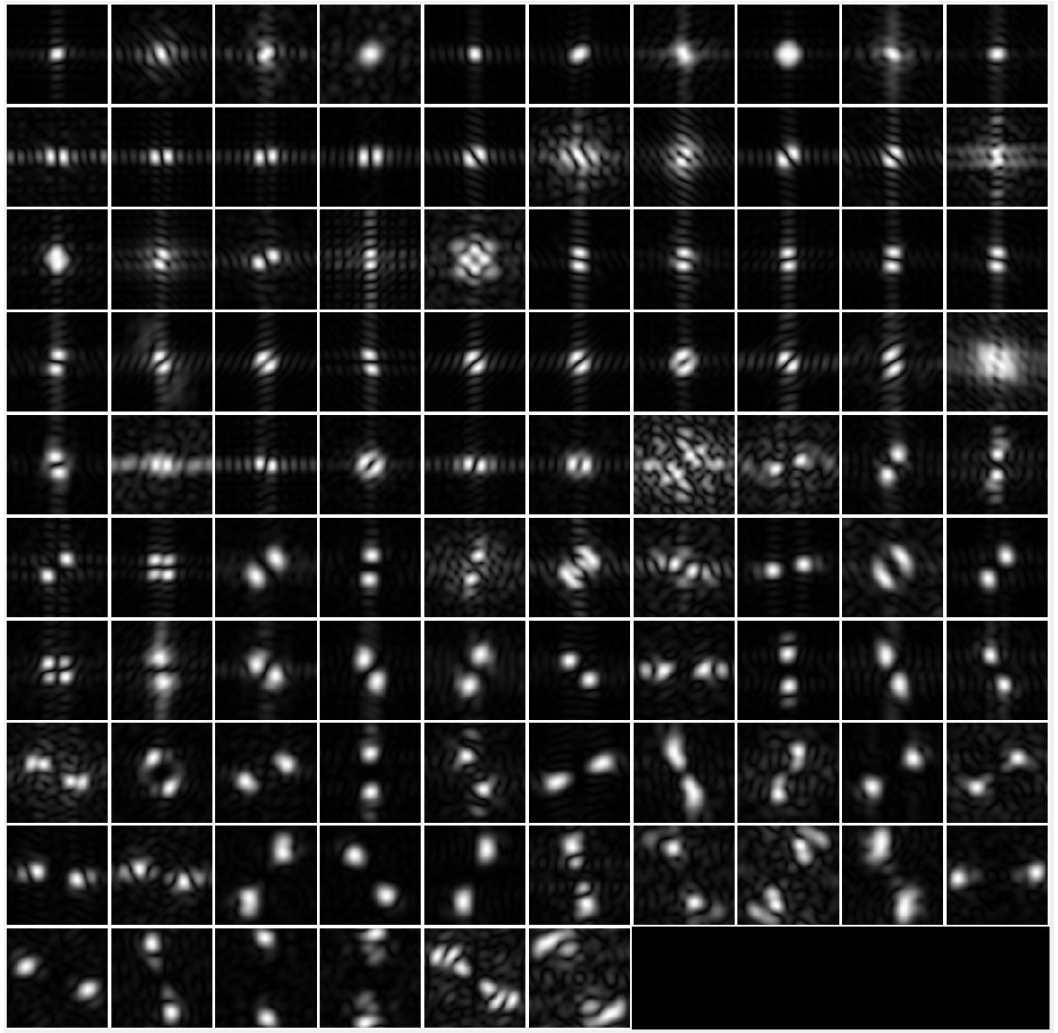


Spatial Domain

Example: 96 Kernels in Conv1



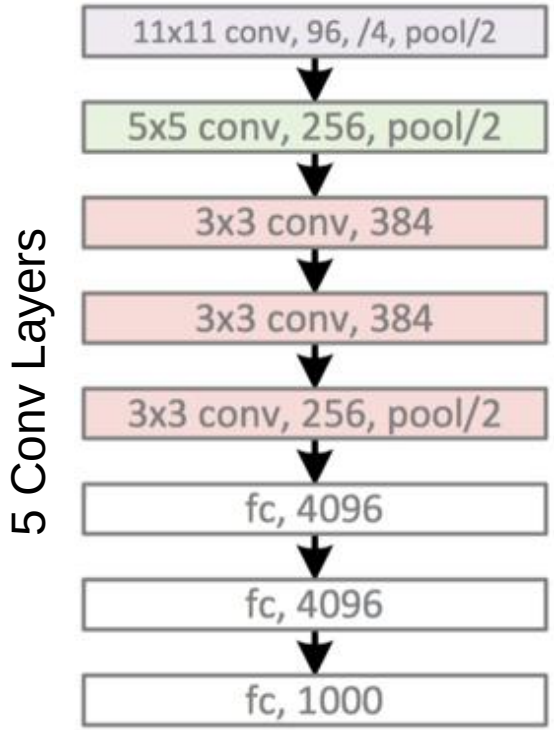
Spatial Domain



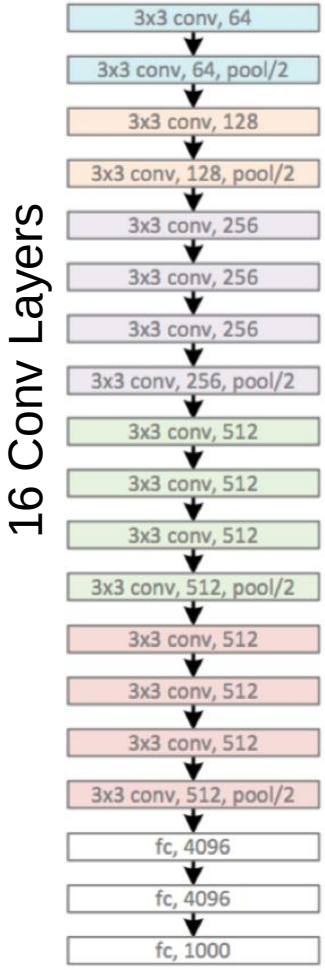
Frequency Domain

Development of Architectures

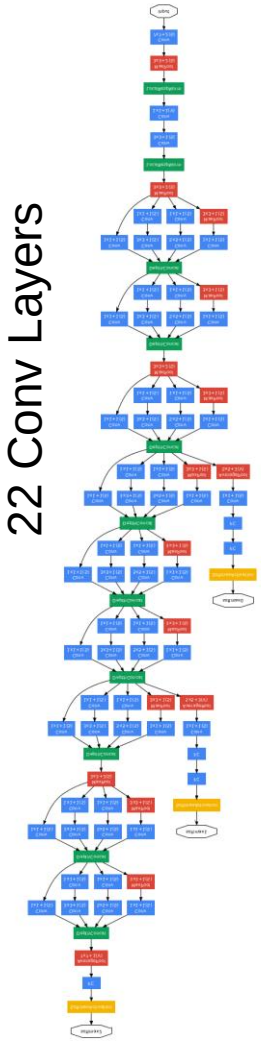
AlexNet (2012)



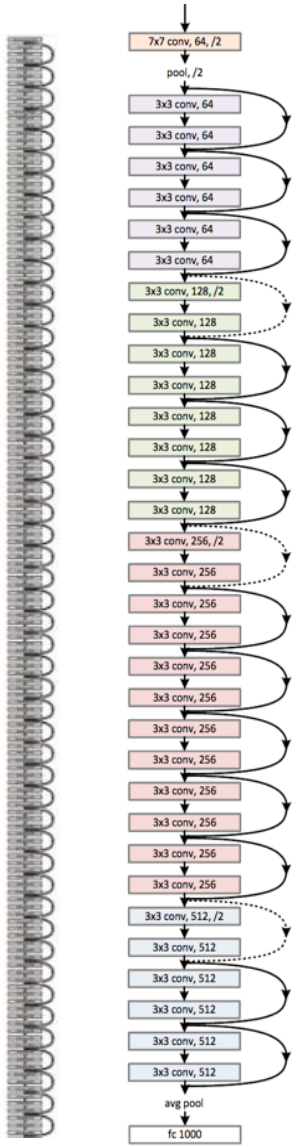
VGG-Net (2014)



GoogleNet (2015)

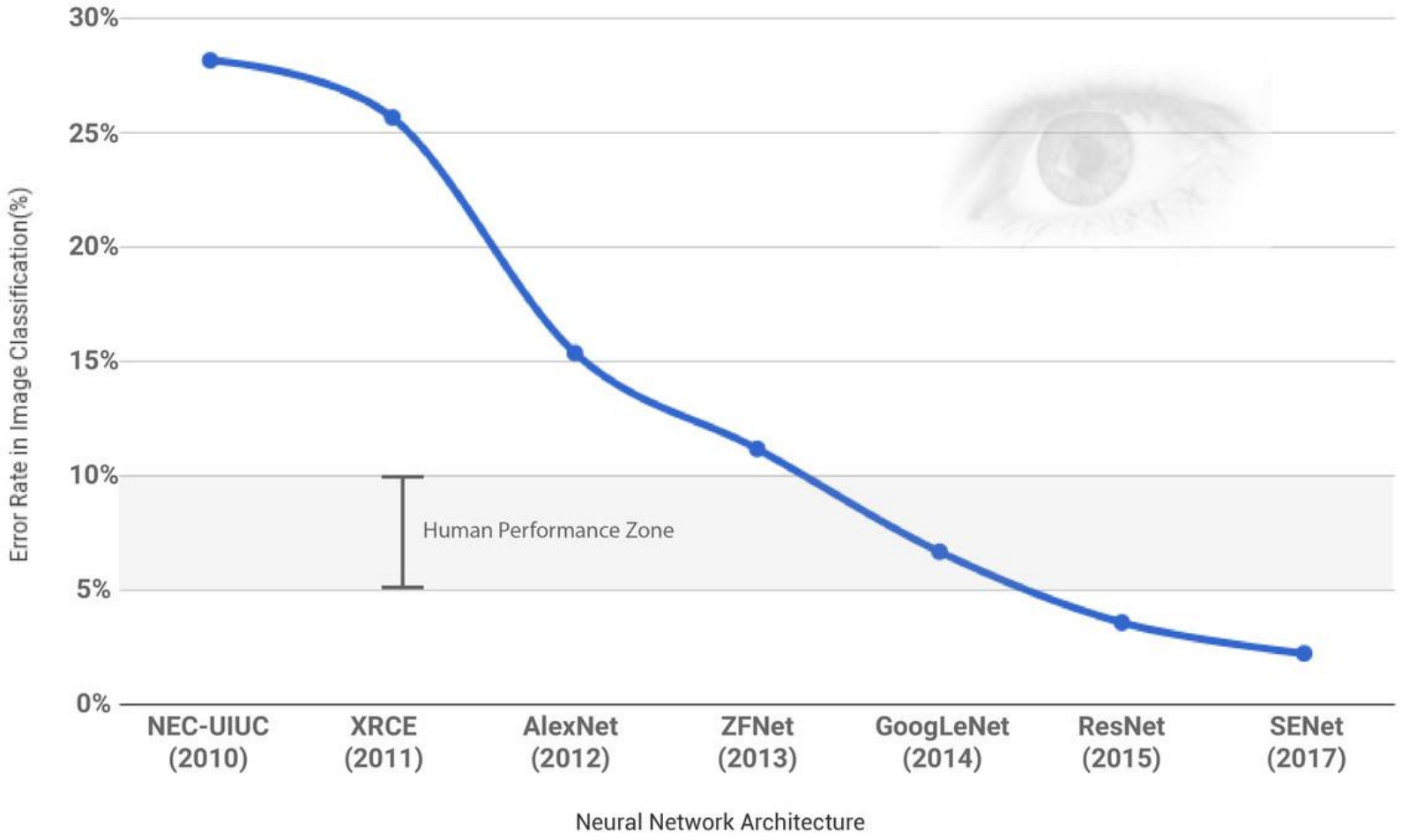


ResNet (2016)



…going really deep…

Development of Architectures



Decline in top 5 Error Rate for Neural Network Architecture for Image Classification

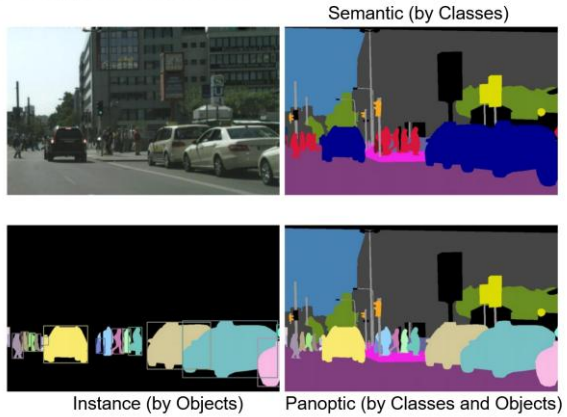
Top 5 Error Rate = correct Label not among top 5 Classifications

Course Overview

CW	Topic	Date	Place	Lab
41	Introduction and Course Overview	07.10.2025	Zoom	Lab 1
42	Capturing Digital Images	14.10.2025	Zoom	Lab 2
43	Digital Image Processing	21.10.2025	Zoom	Assignment 1
44	Machine Learning	28.10.2025	Zoom	
45	Feature Extraction	04.11.2025	Zoom	Open Lab 1
46	Segmentation	11.11.2025	Zoom	Assignment 2
47	Optical Flow	18.11.2025	Zoom	Open Lab 2
48	Object Detection	25.11.2025	Zoom	Assignment 3
49	Multi-View Geometry	02.12.2025	Zoom	Open Lab 3
50	3D Vision	09.12.2025	Zoom	Assignment 4
3	Trends in Computer Vision	13.01.2026	Zoom	
4	Q&A	20.01.2026	Zoom	Open Lab 4
5	Exam	27.01.2026	HS1 (Linz), S1/S3 (Vienna), S5 (Bregenz)	
9	Retry Exam	24.02.2026	tba	

Next Week: Segmentation

Types of Segmentation



JOHANNES KEPLER UNIVERSITY LINZ

Example: Clustering by Graph Eigenvectors

affinity between element i and j
X
association of elements j with cluster n

$$w_n^T A w_n = \lambda \qquad A w_n = \lambda w_n$$

association of elements j with cluster n X

$$\begin{bmatrix} w_{n,0} \dots w_{n,j} \end{bmatrix} \begin{bmatrix} a_{0,0} & \dots & a_{i,0} \\ \dots & \dots & \dots \\ a_{0,j} & \dots & a_{i,j} \end{bmatrix} \begin{bmatrix} w_{n,0} \\ \dots \\ w_{n,j} \end{bmatrix}$$

JOHANNES KEPLER UNIVERSITY LINZ

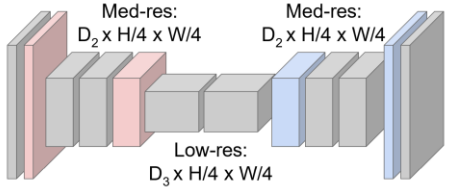
Segmentation using CNNs

Downsampling: Pooling, Strided Convolution

Design Network as a Bunch of Convolutional Layers, with **Downsampling** and **Upsampling** inside the Network!

Upsampling: Strided Transposed Convolution (or Unpooling)

Input:
3 x H x W



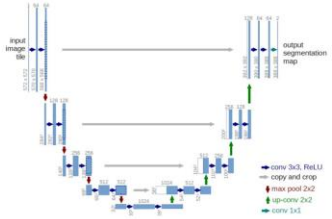
High-res:
D1 x H/2 x W/2

High-res:
D1 x H/2 x W/2

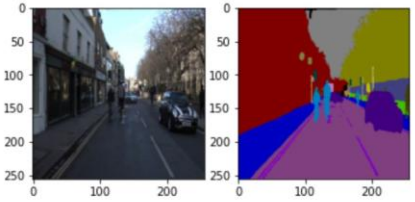
Predictions:
H x W

JOHANNES KEPLER UNIVERSITY LINZ

Connected Autoencoders (U-Nets)



Connected Autoencoder (U-Net)



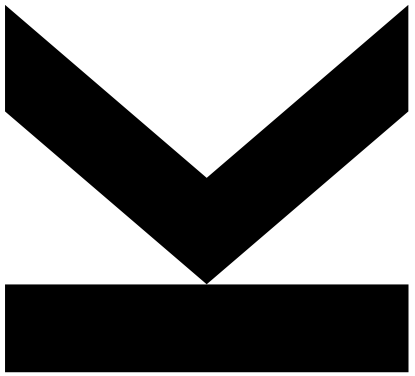
Predicted Segmentation

- U-Nets overcome this problem by connecting corresponding encoder-decoder layers with skip connections:
  - the output of an encoder level is skip-connected (concatenation) with the input of the corresponding decoder level

JOHANNES KEPLER UNIVERSITY LINZ

JOHANNES KEPLER UNIVERSITY LINZ

Thank You



JOHANNES KEPLER
UNIVERSITY LINZ