Image Analysis with Deep Convolutional Neural Networks

AI in Medicine

Ricardo Henao

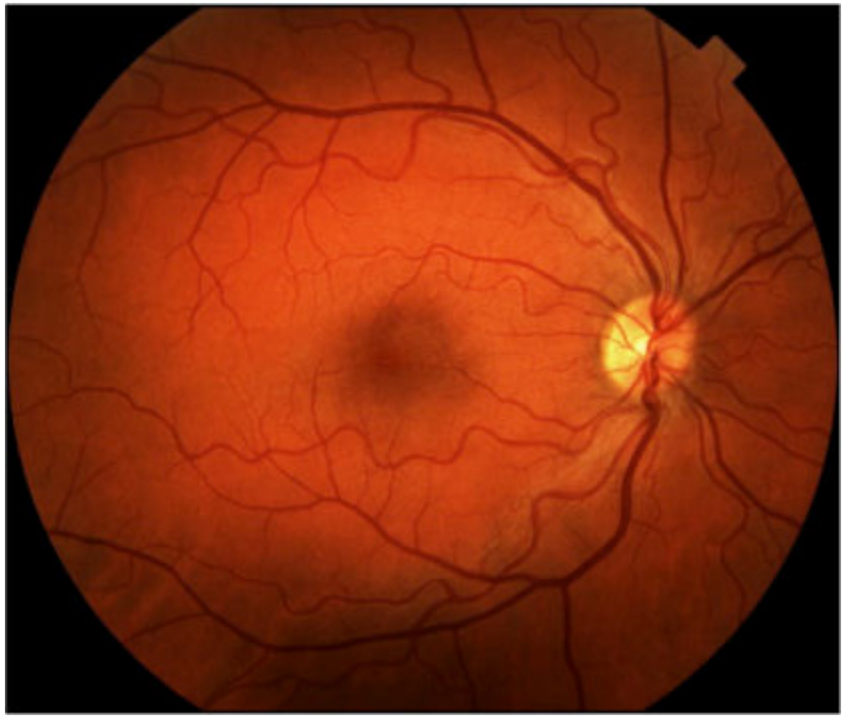
أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology
Computational Bioscience Research Center

مبادرة الصحة الذكية
Smart-Health Initiative

Deep Learning for Image Analysis

Diabetic Retinopathy Classification

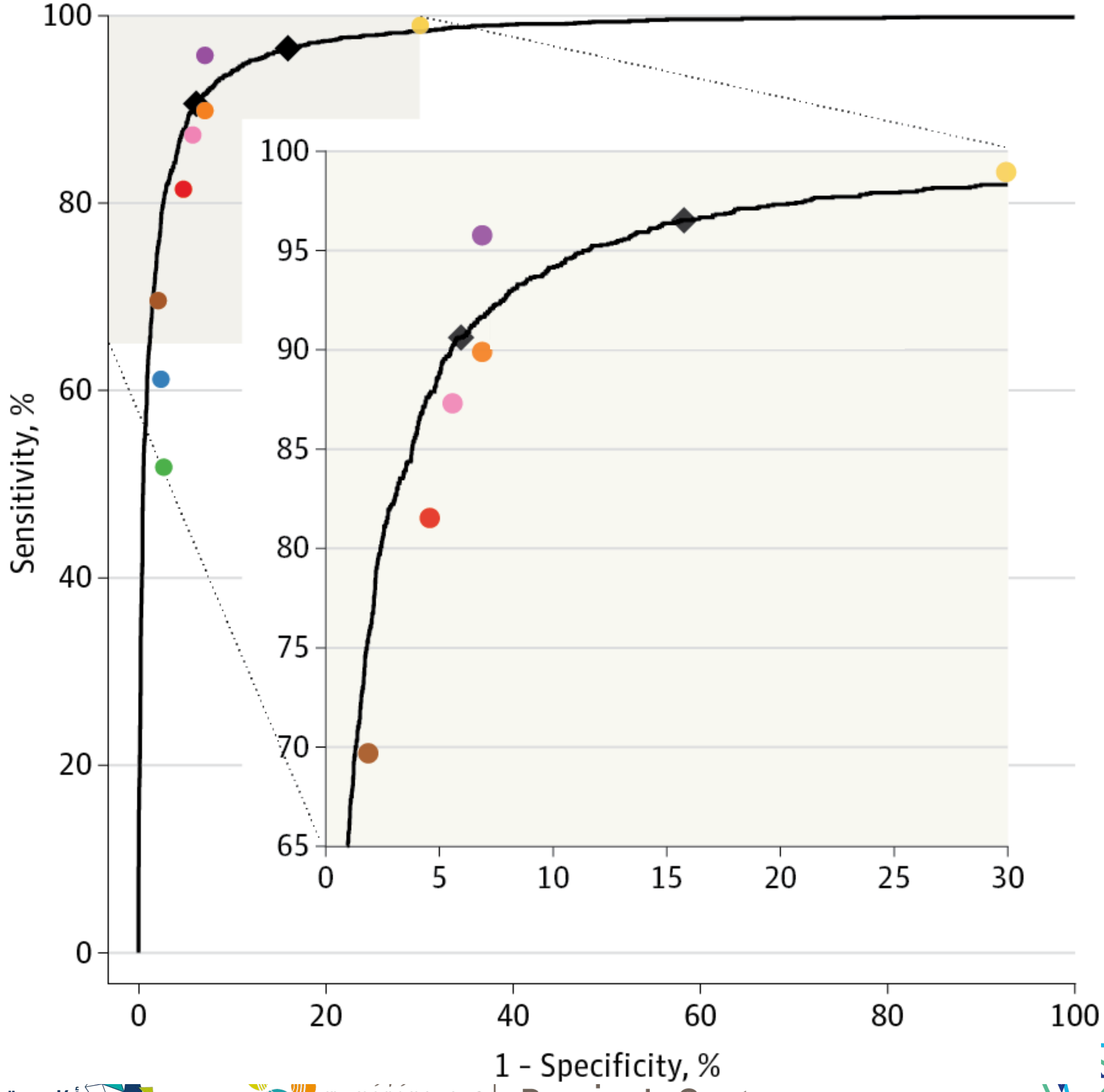


Healthy Retina

Unhealthy Retina

Deep Learning for Image Analysis

Diabetic Retinopathy Classification



$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{total number of positives in the dataset}}$$

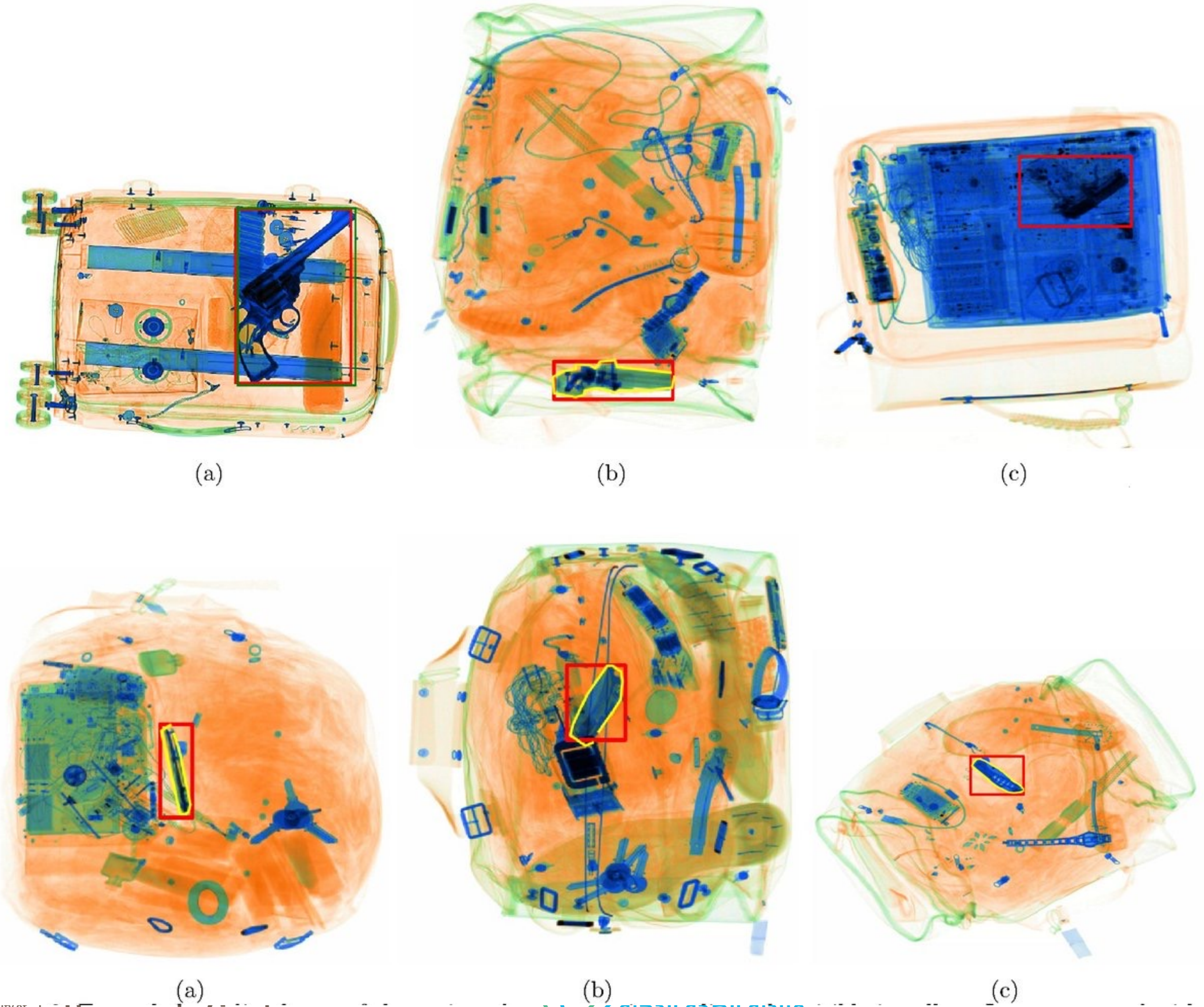
$$\text{specificity} = \frac{\text{number of true negatives}}{\text{total number of negatives in the dataset}}$$

Gulshan et al. *JAMA* (2016)

See also: Ting et al. *JAMA* (2017)

Deep Learning for Image Analysis

TSA Screening



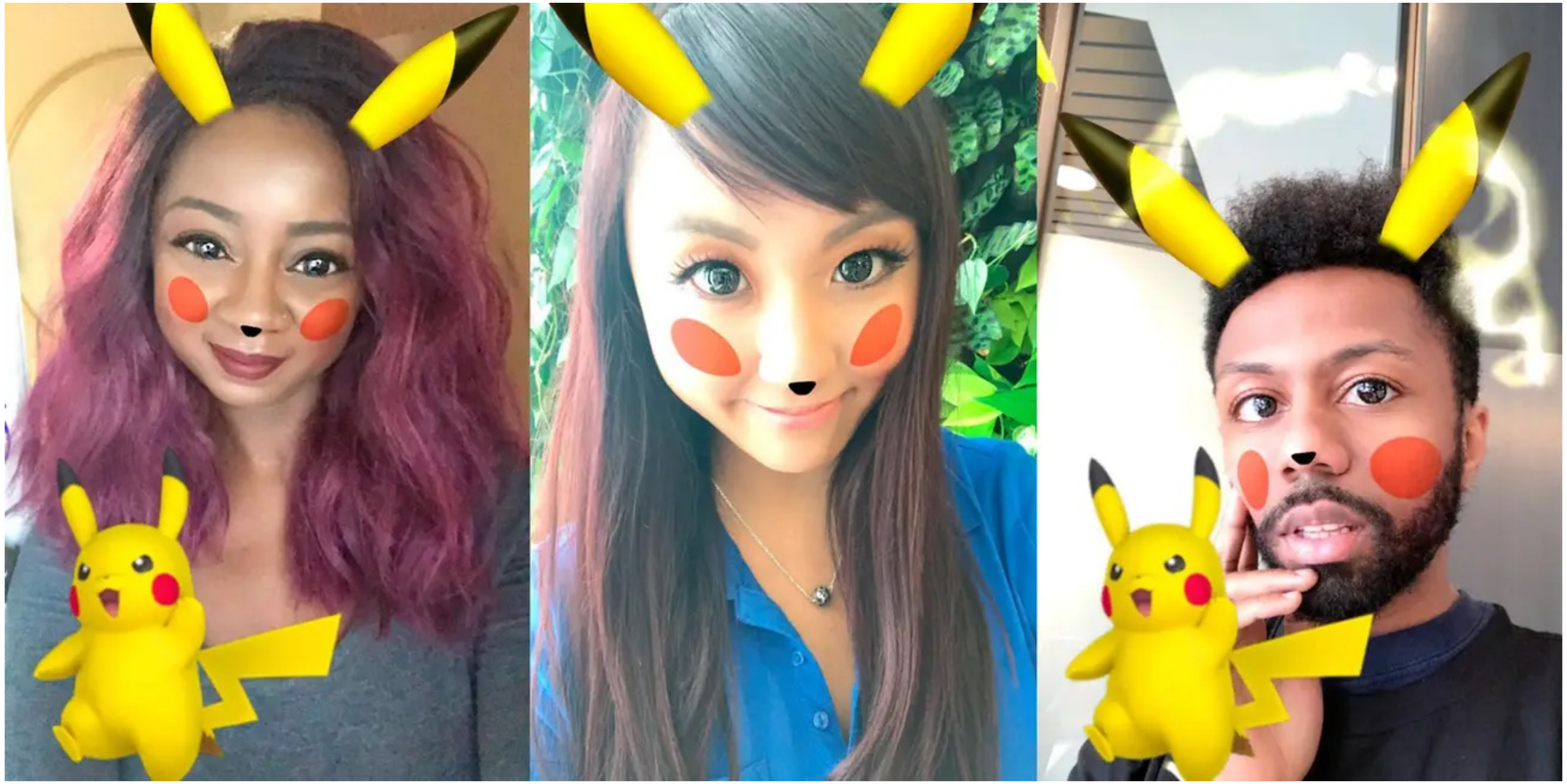
Liang et al. *SPIE* (2018)

Deep Learning for Image Analysis

Face Recognition

Deep Learning for Image Analysis

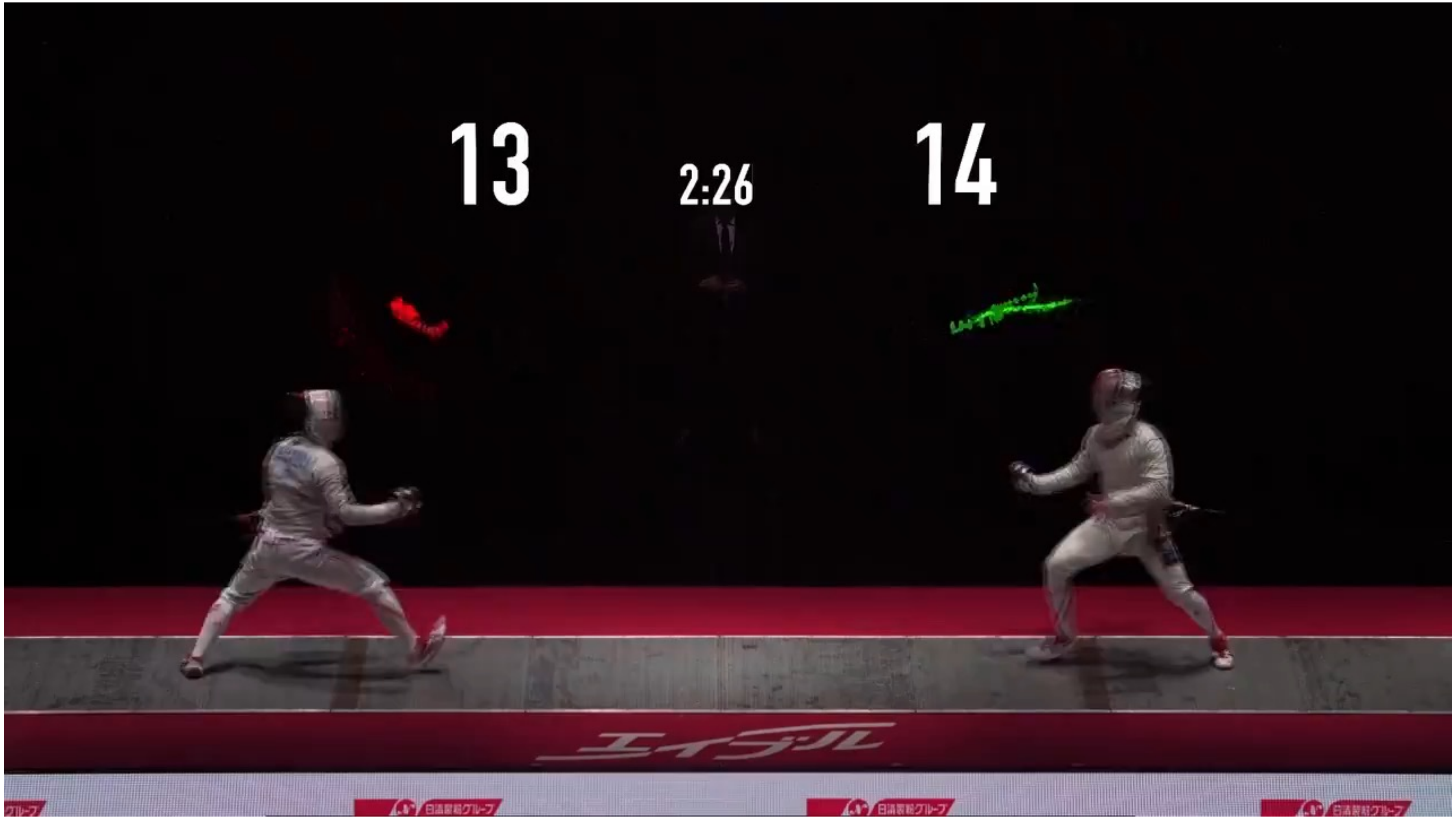
Face Recognition



businessinsider

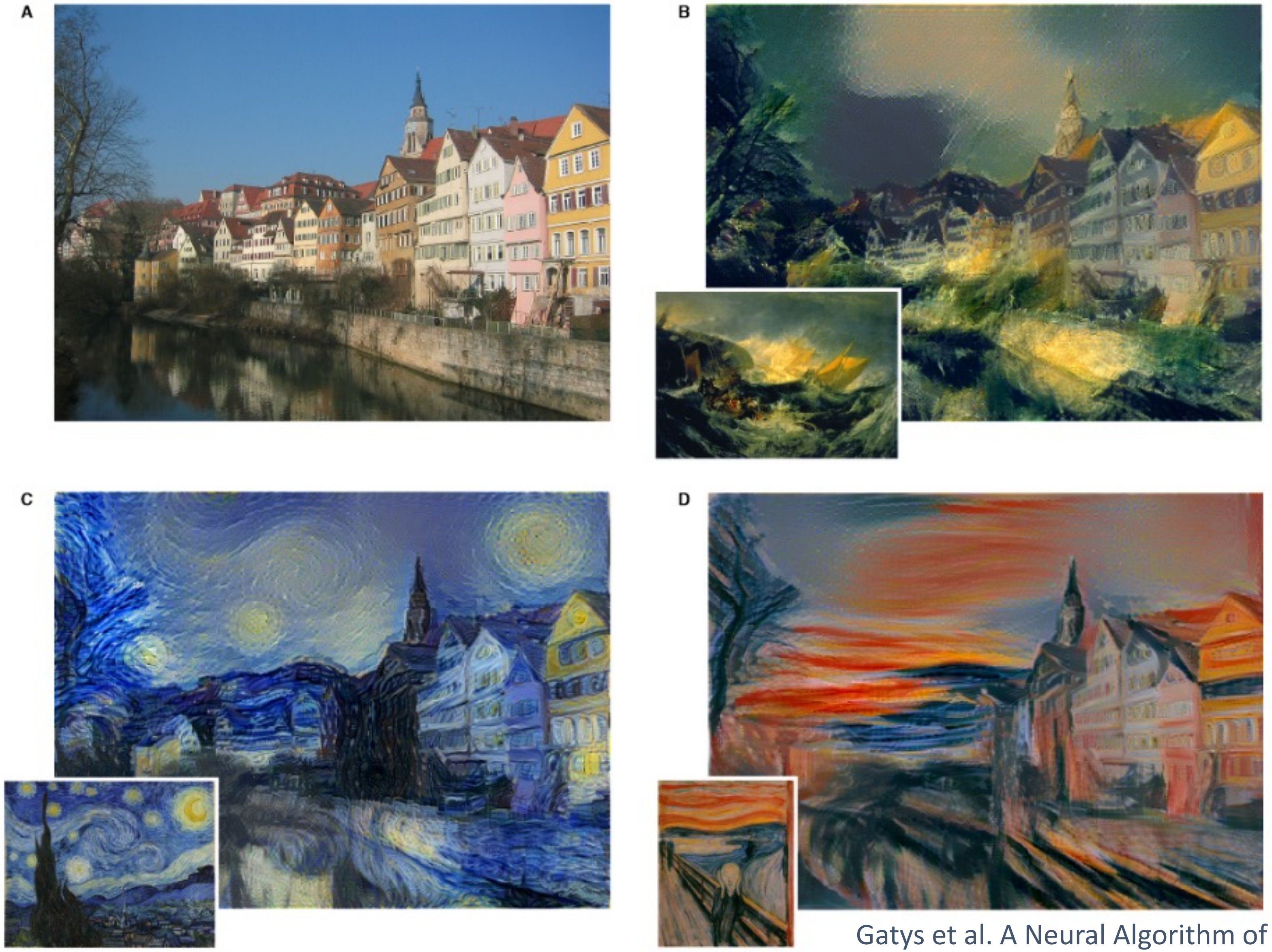
Deep Learning for Image Analysis

Automatic 3D Body Tracking in Video



Deep Learning for Image Analysis

Style Transfer and Harmonization



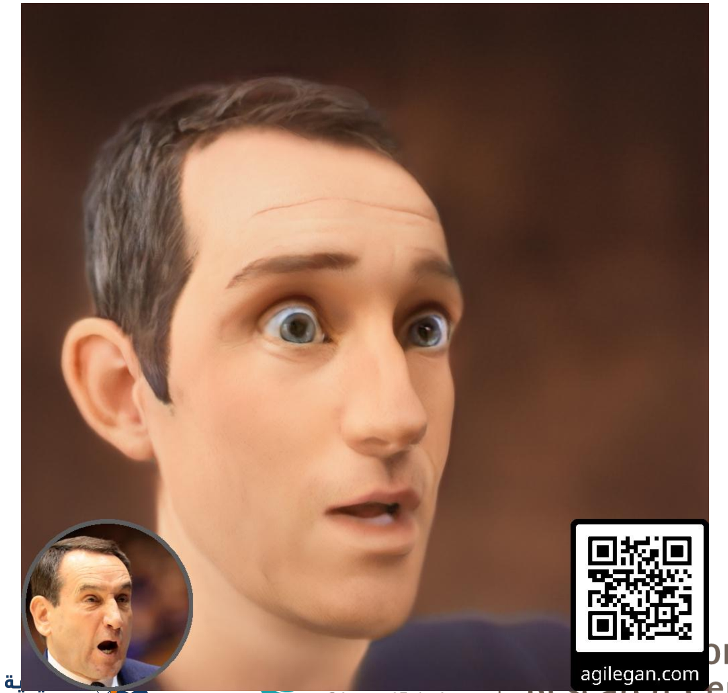
Gatys et al. A Neural Algorithm of Artistic Style. *arXiv* (2015)

Deep Learning for Image Analysis

Stylizing Portraits

www.agilegan.com

Cartoon

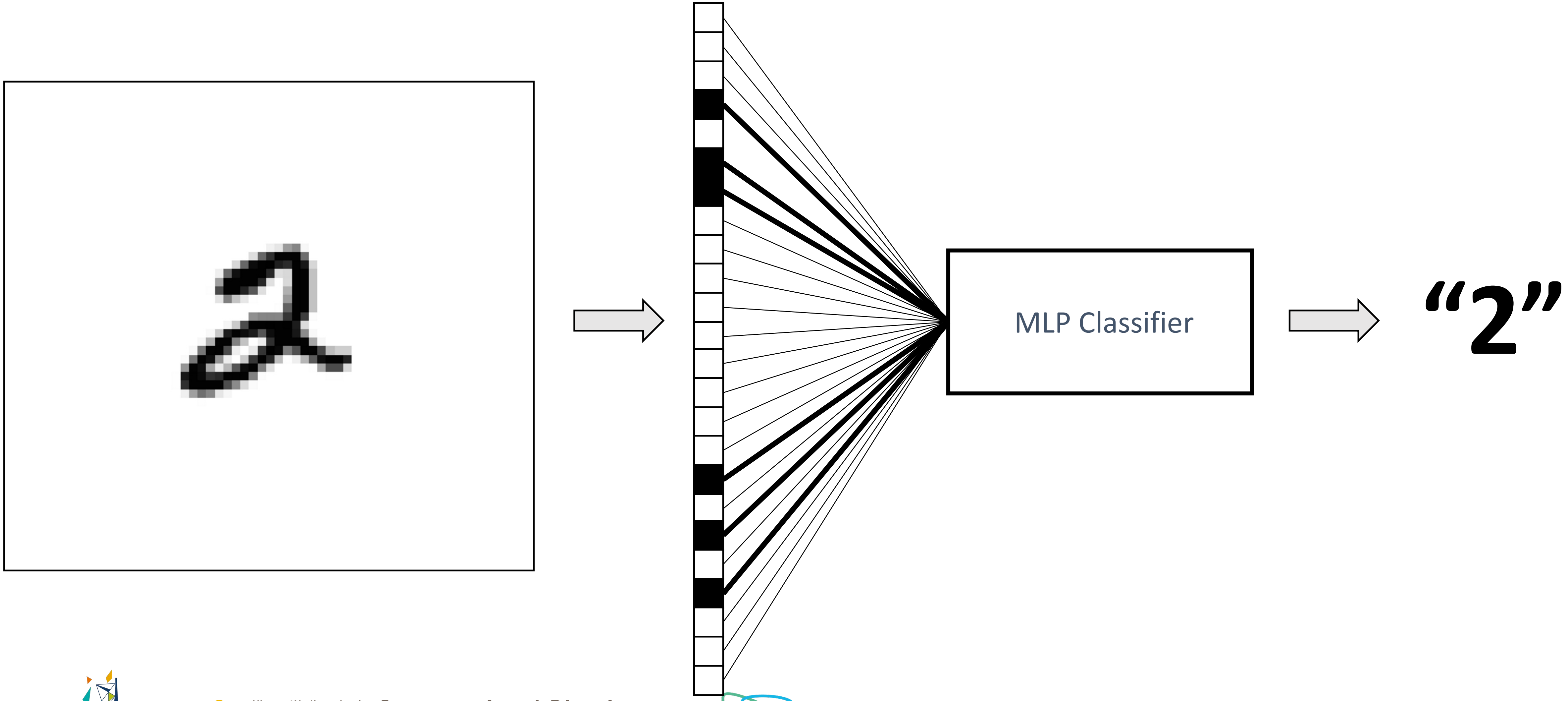


Painting

Comic

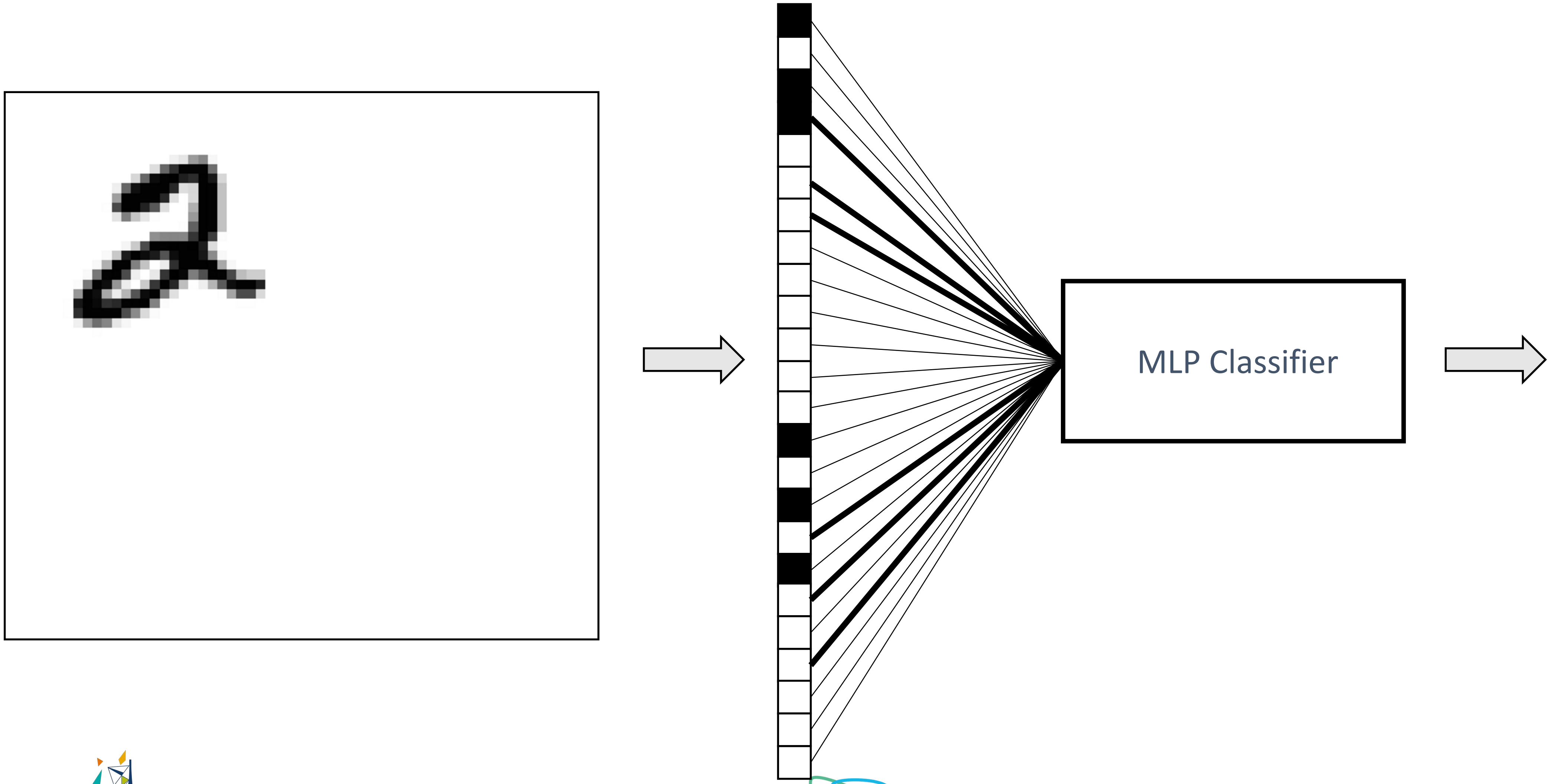
Convolutional Neural Networks Find Structure Anywhere In Images

Consider the multi-layer perceptron for digit recognition:



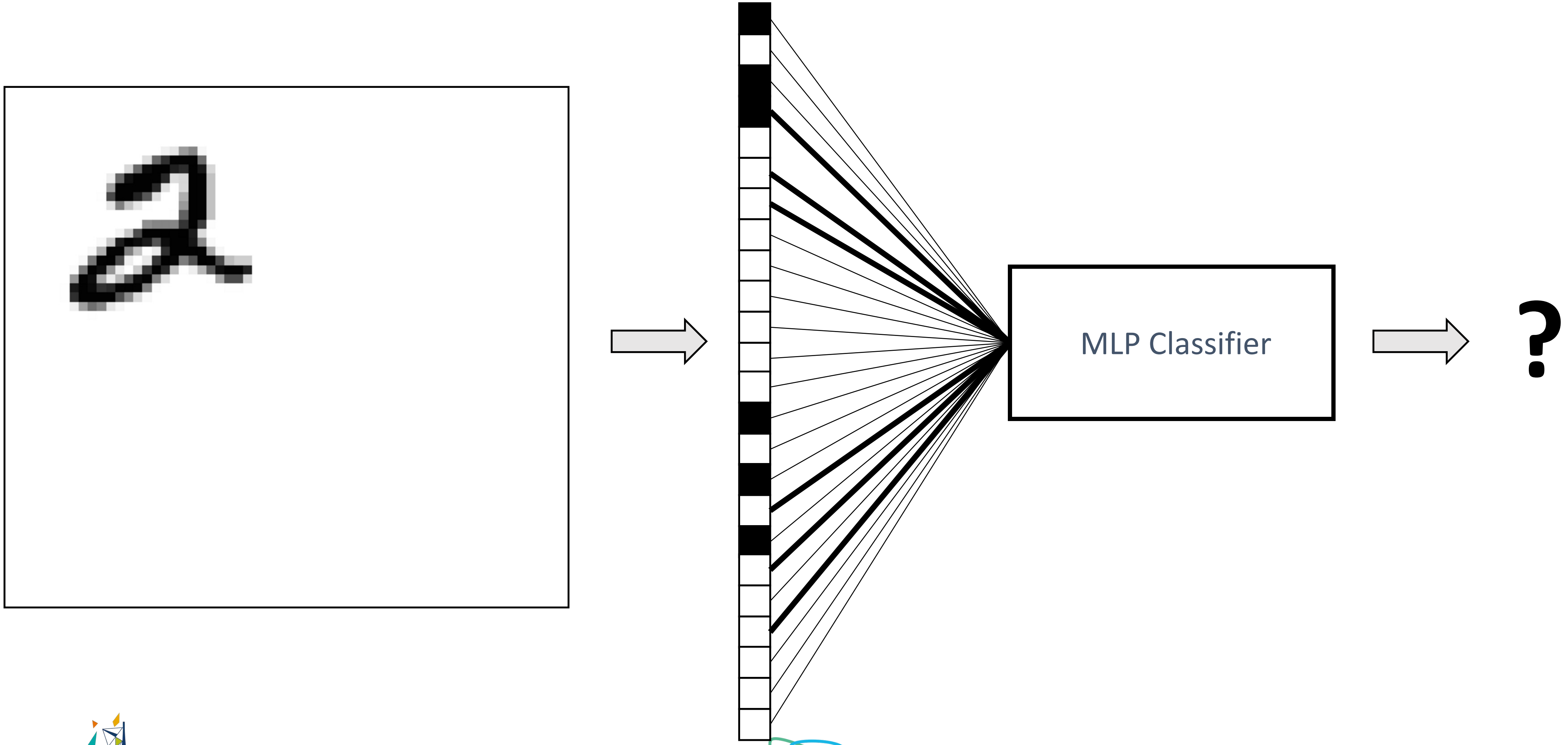
Convolutional Neural Networks Find Structure Anywhere In Images

Consider the multi-layer perceptron for digit recognition:

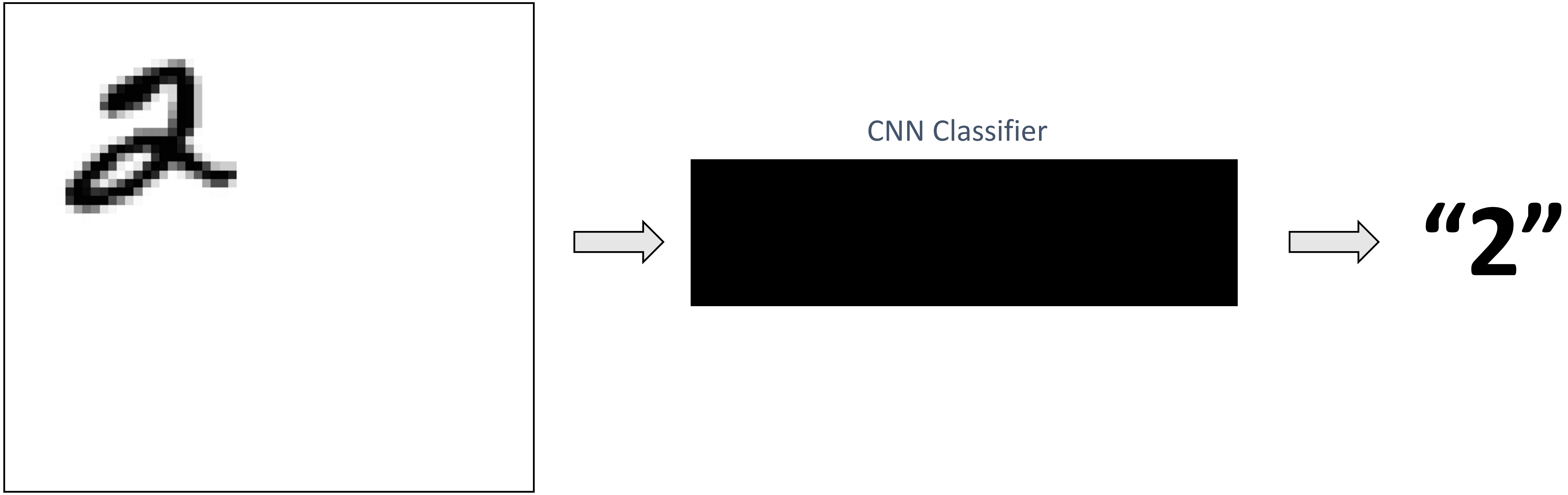


Convolutional Neural Networks Find Structure Anywhere In Images

Consider the multi-layer perceptron for digit recognition:

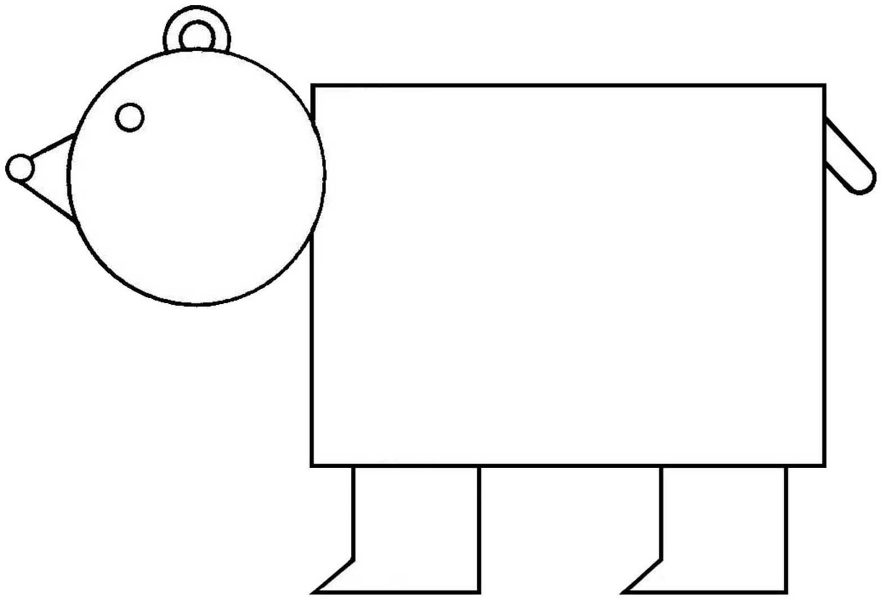


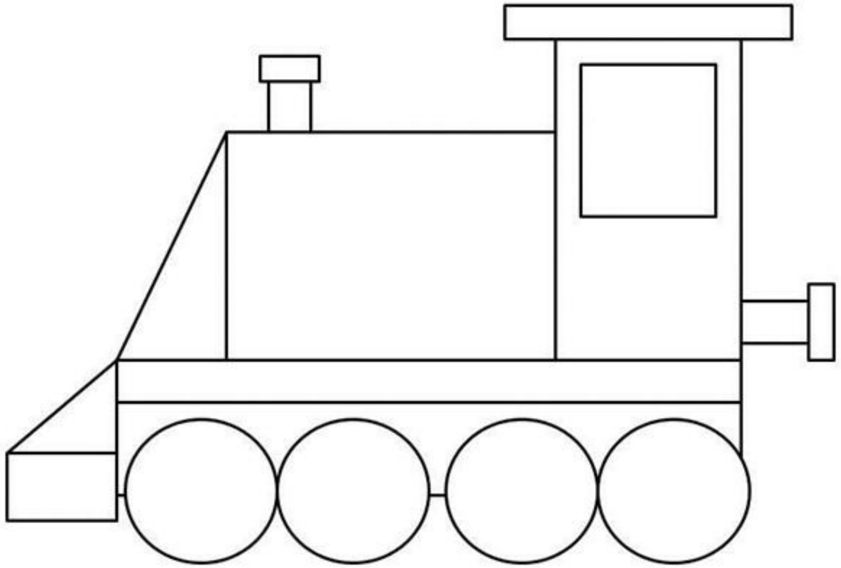
Convolutional Neural Networks Find Structure Anywhere In Images



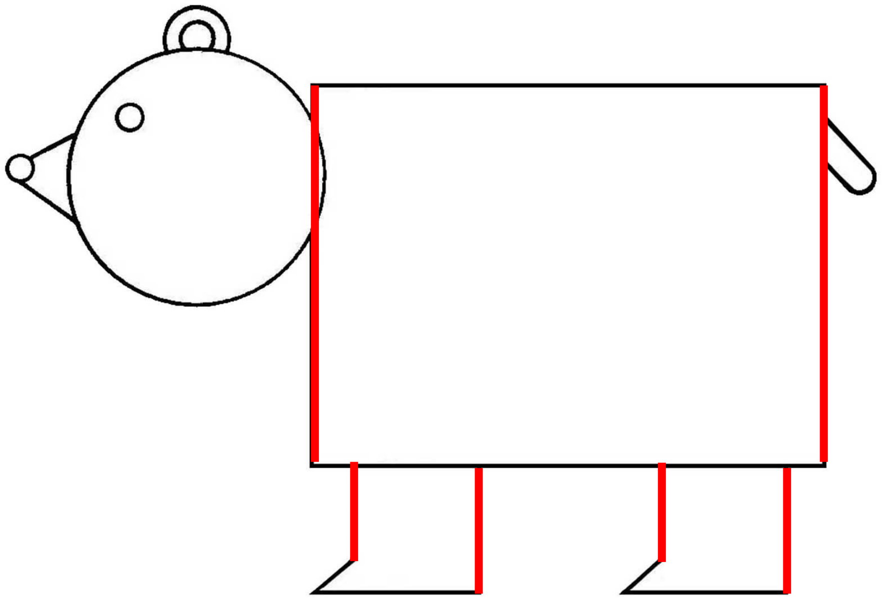
CNNs Take Advantage of Repeated, Hierarchical Structure in Images

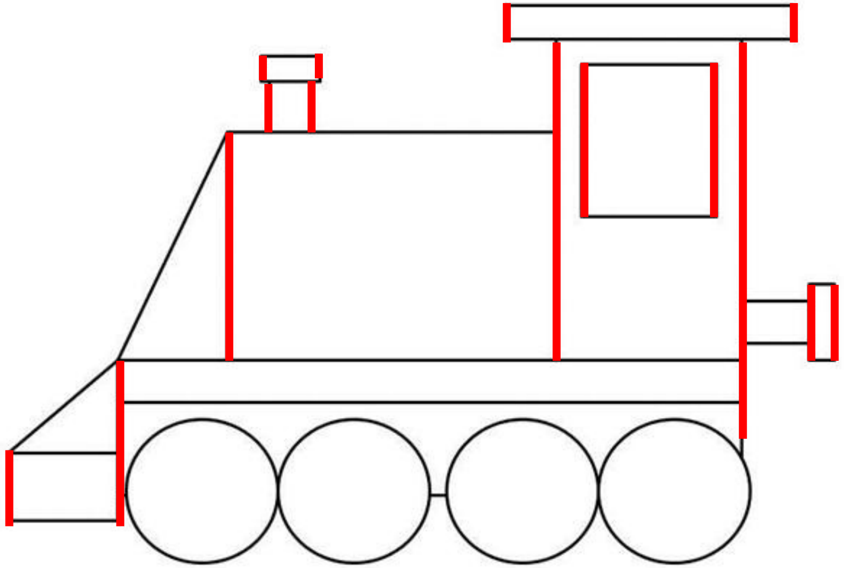
CNNs Take Advantage of Repeated, Hierarchical Structure in Images





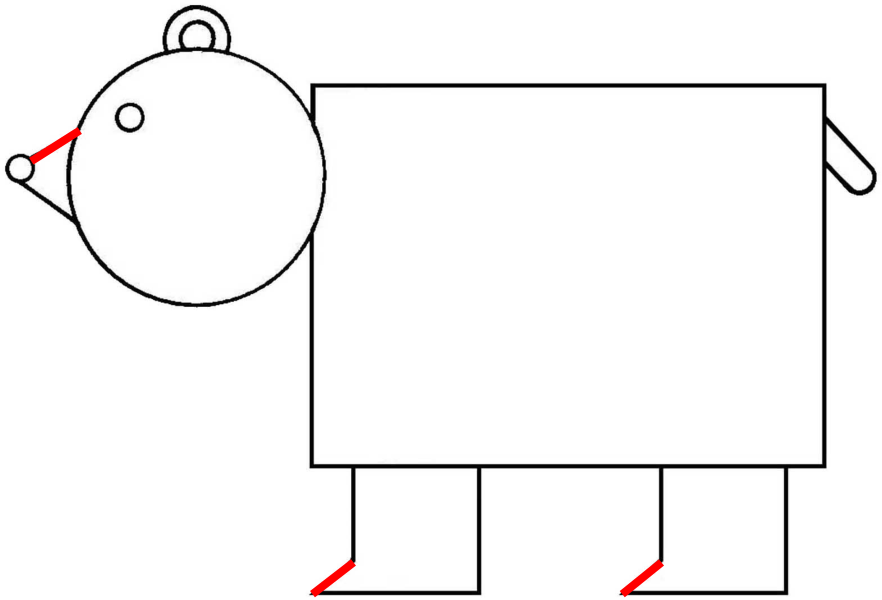
CNNs Take Advantage of Repeated, Hierarchical Structure in Images

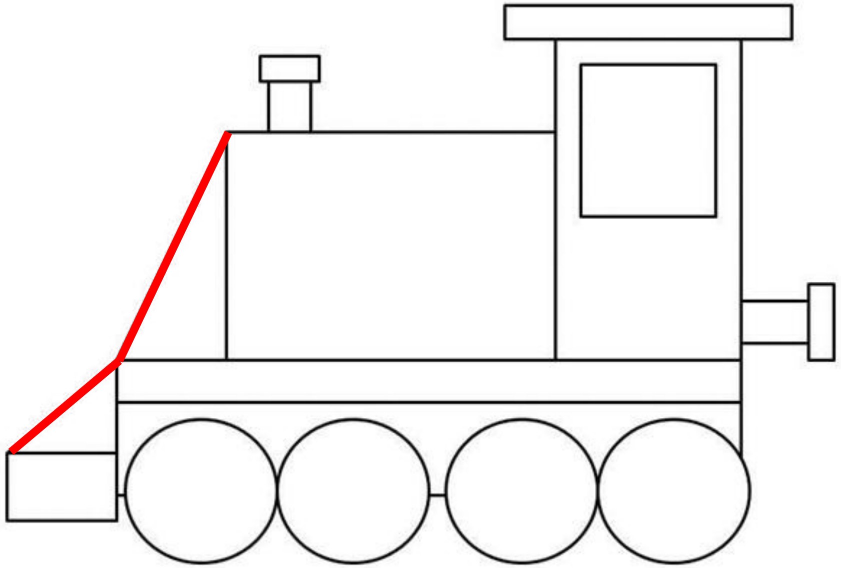




Low-level structure: lines, curves

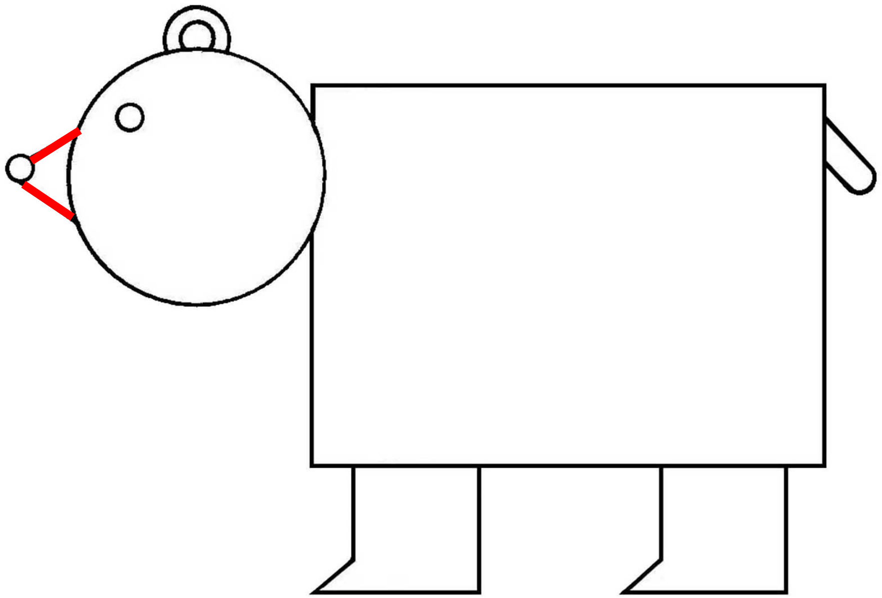
CNNs Take Advantage of Repeated, Hierarchical Structure in Images

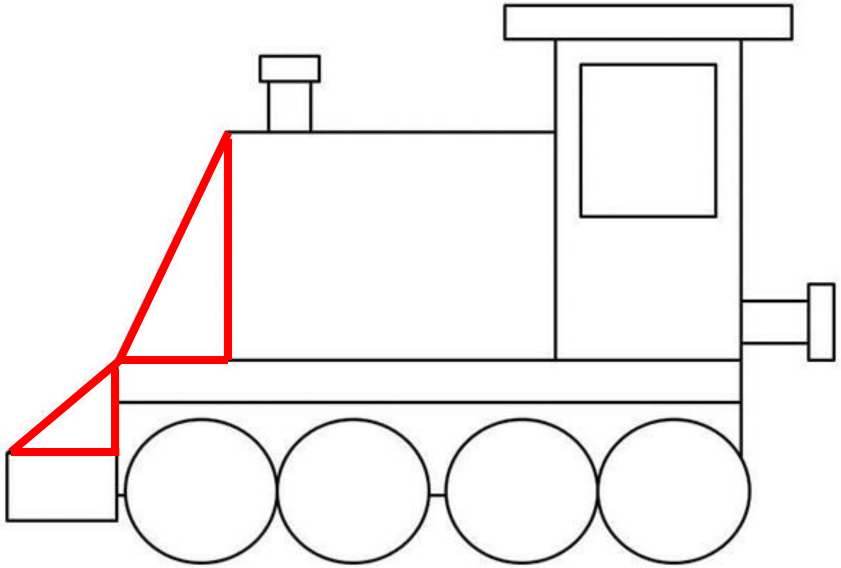




Low-level structure: lines, curves

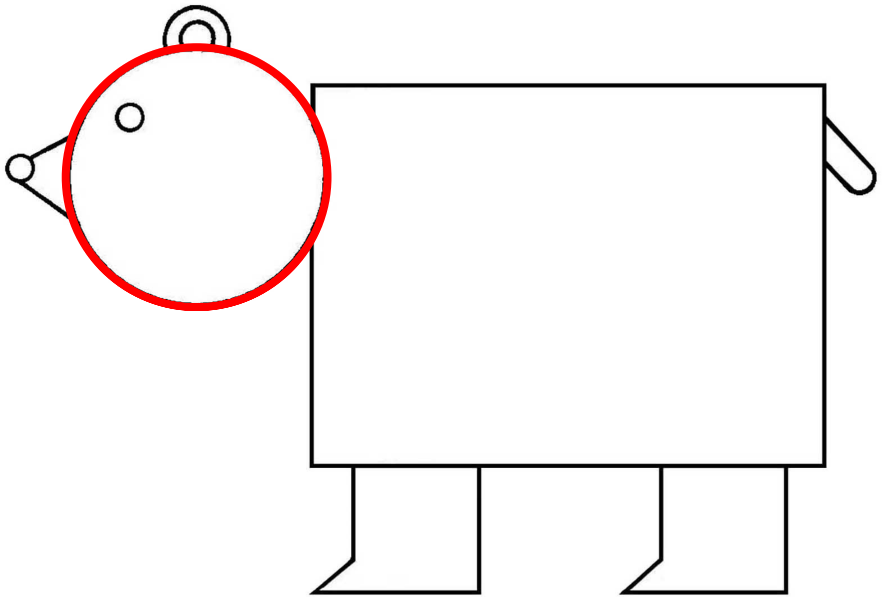
CNNs Take Advantage of Repeated, Hierarchical Structure in Images

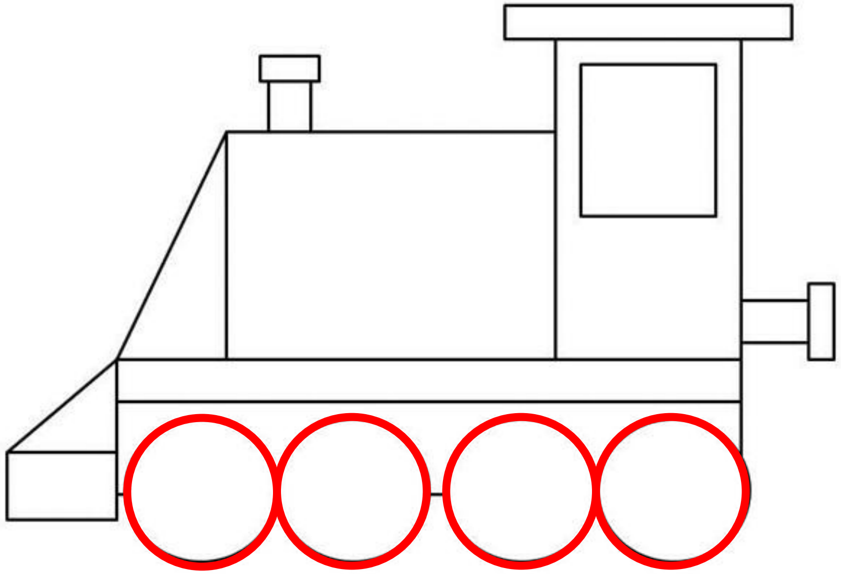




Mid-level structure: shapes

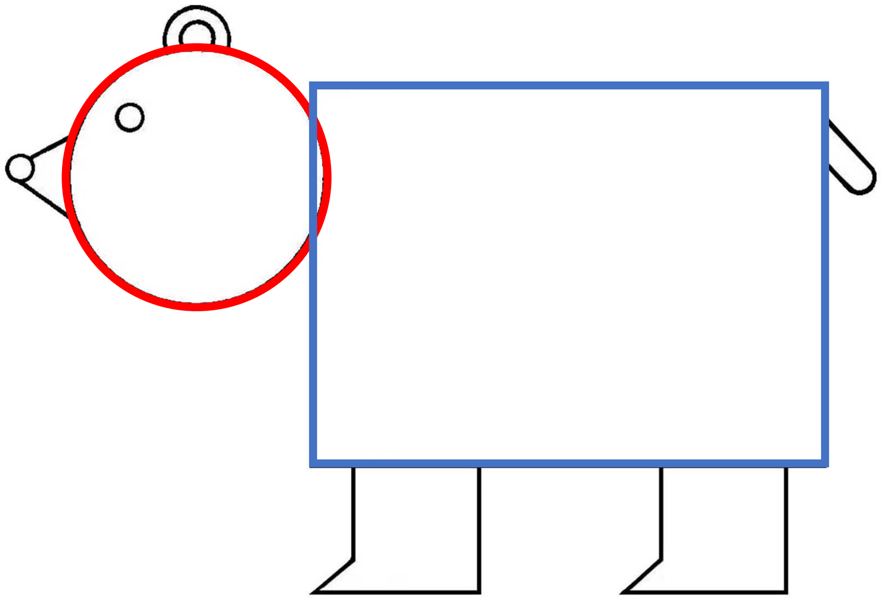
CNNs Take Advantage of Repeated, Hierarchical Structure in Images

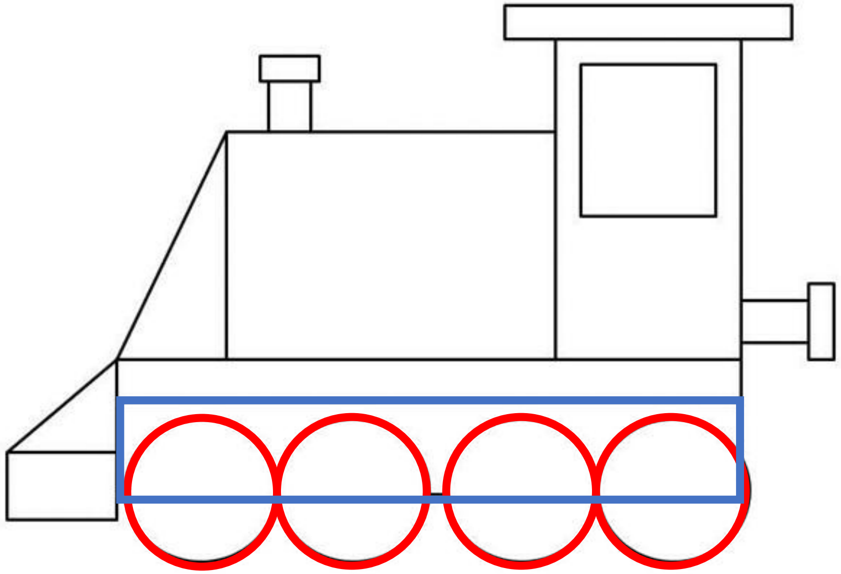




Mid-level structure: shapes

CNNs Take Advantage of Repeated, Hierarchical Structure in Images

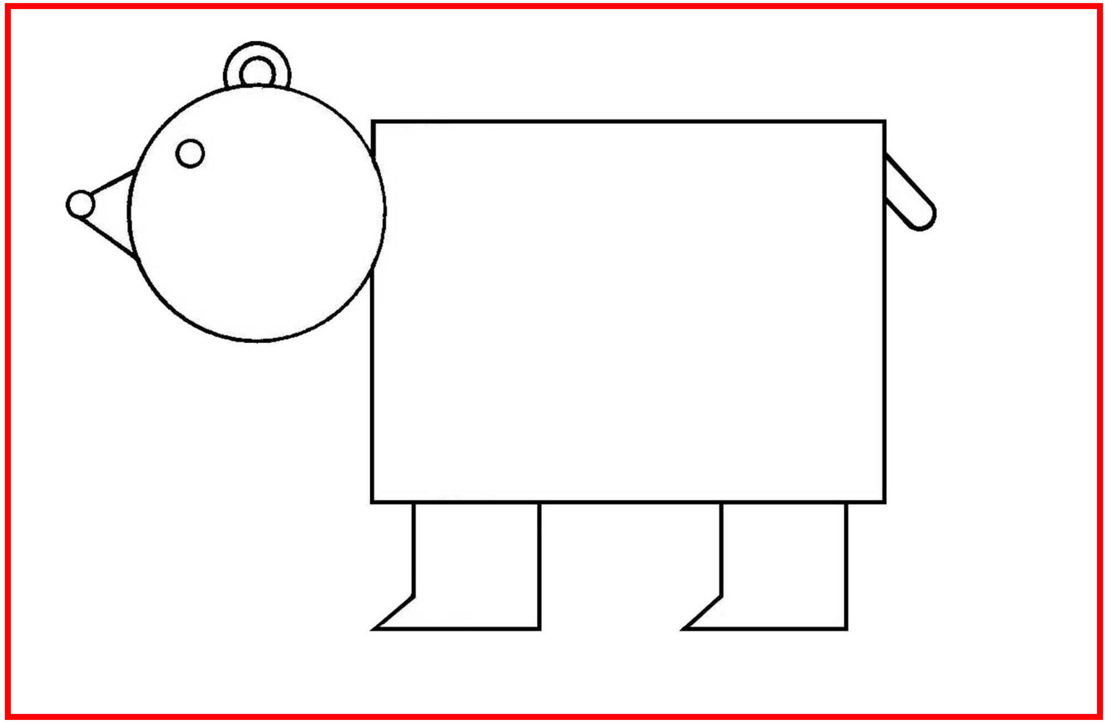




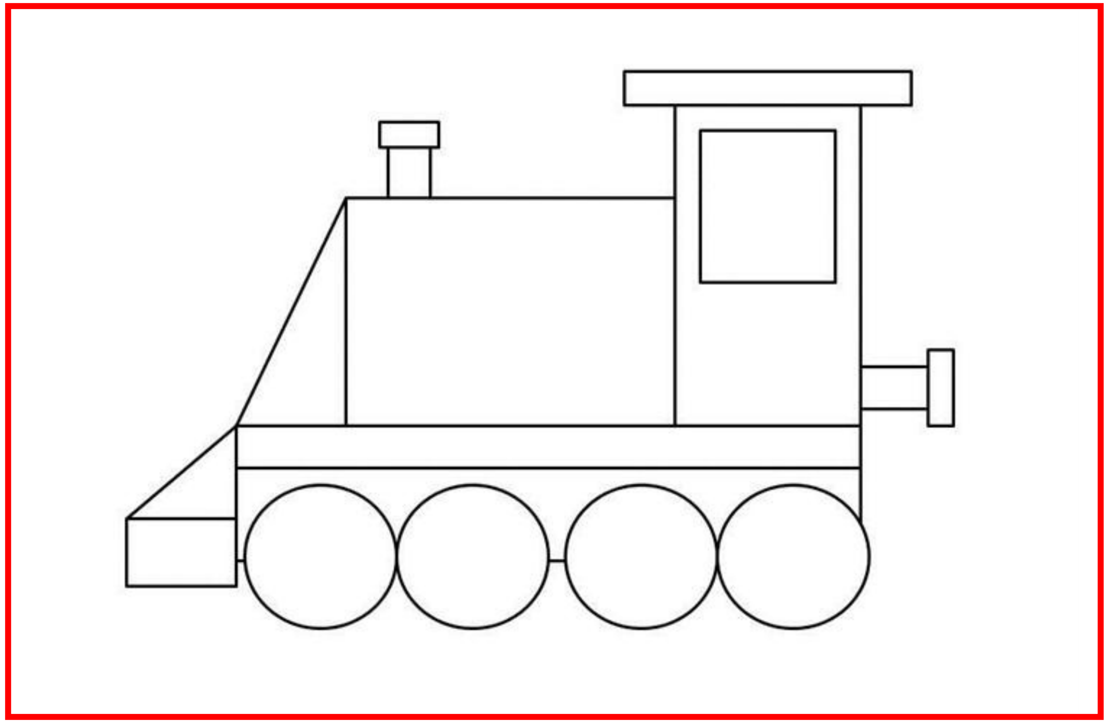
High-level structure: groups of shapes

CNNs Take Advantage of Repeated, Hierarchical Structure in Images

Bear

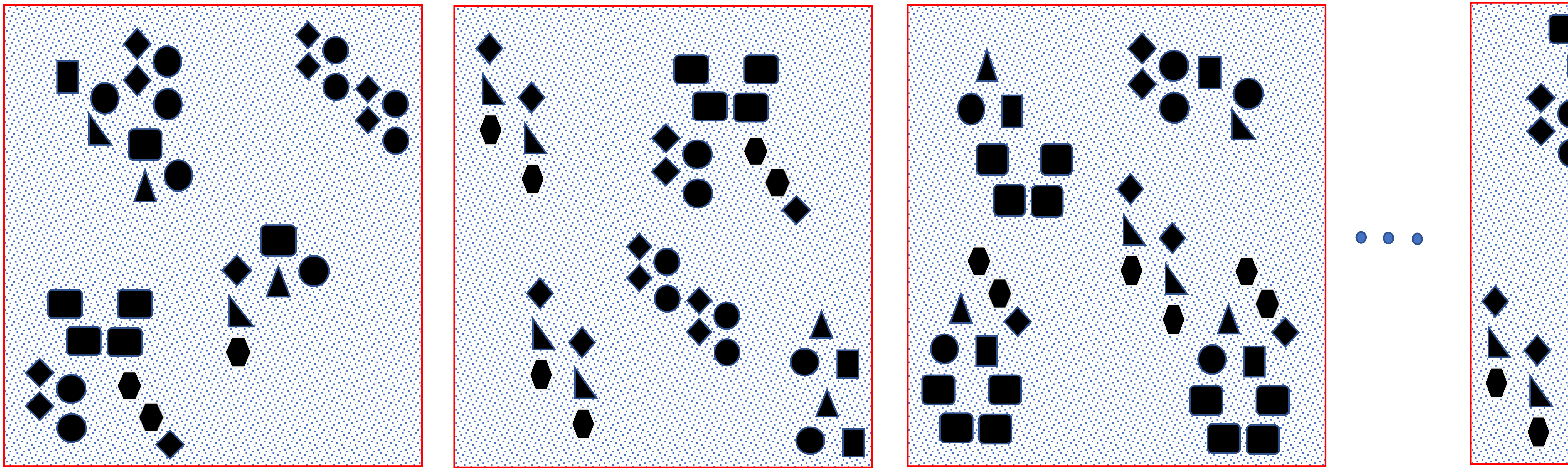


Train

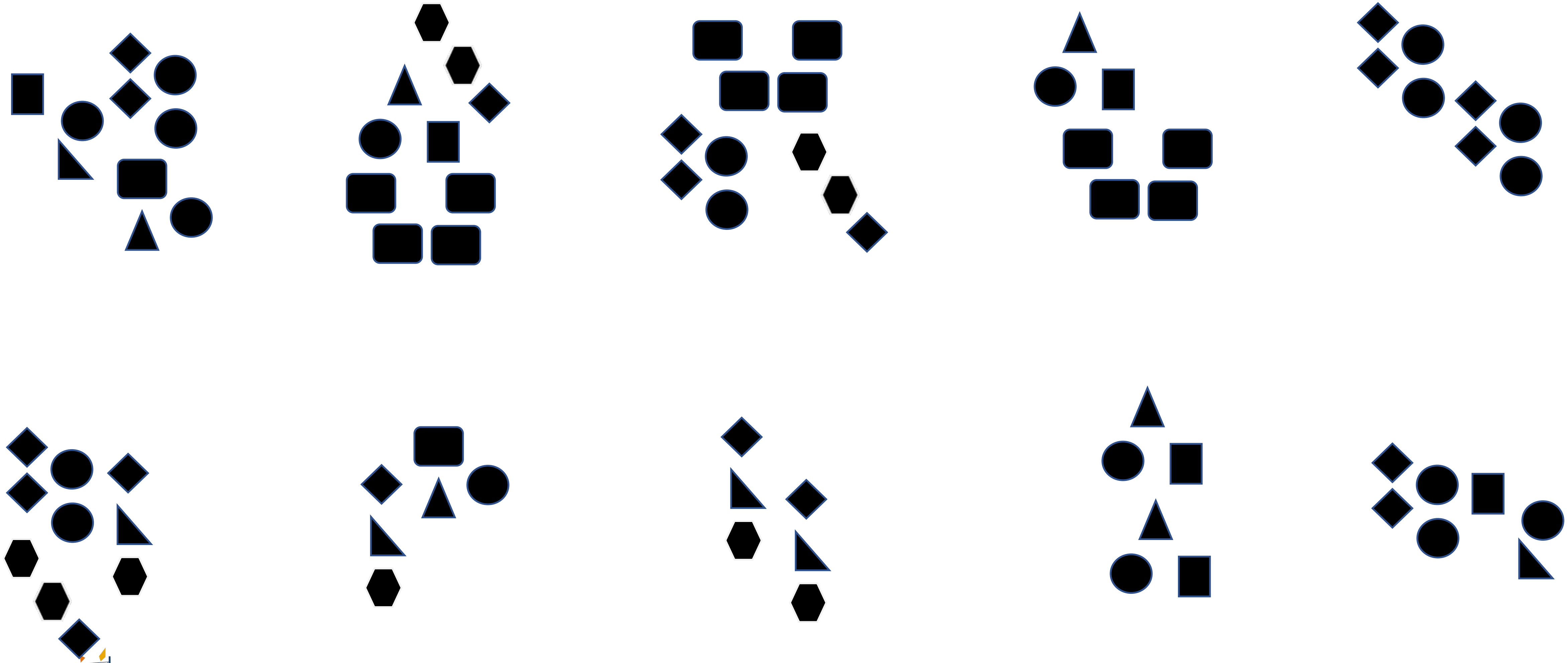


High-level structure: groups of shapes → objects

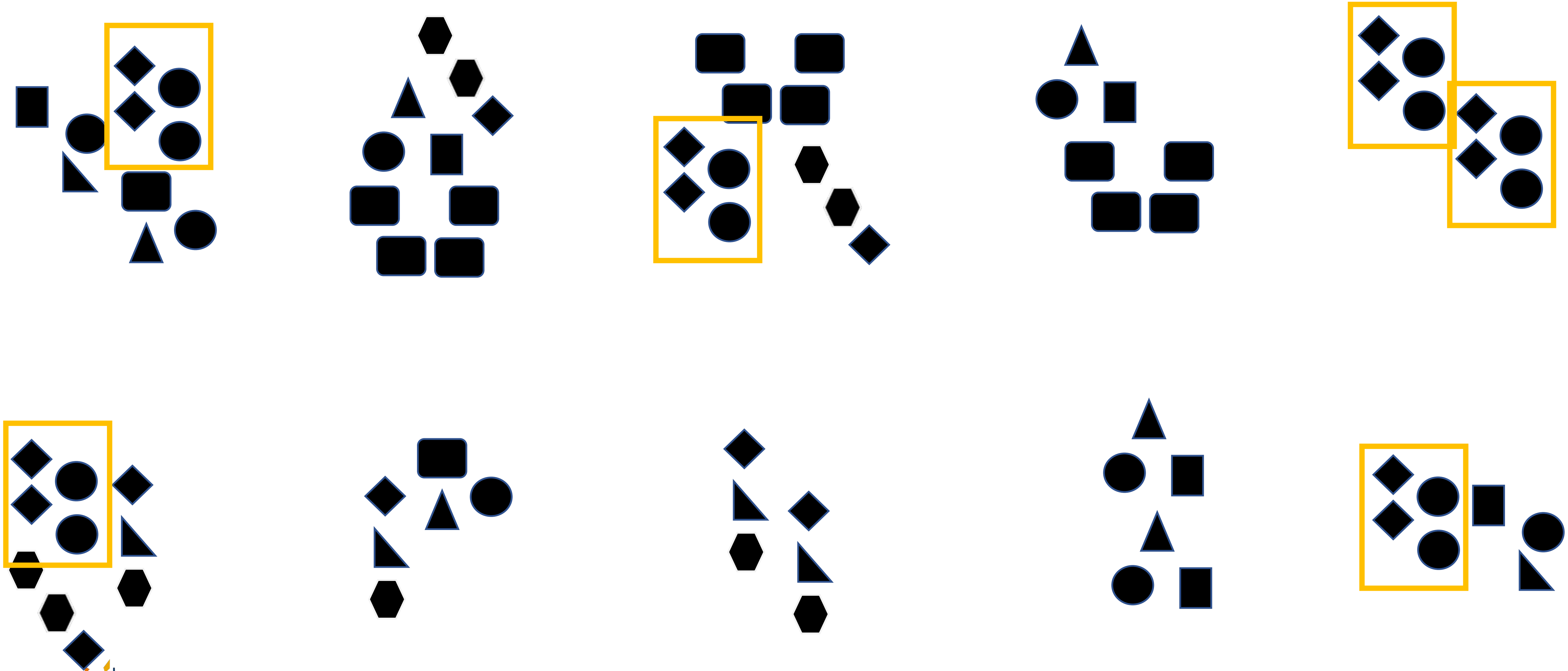
Consider a Set of "Toy" Images,
for illustration of how this structure can be extracted by an algorithm



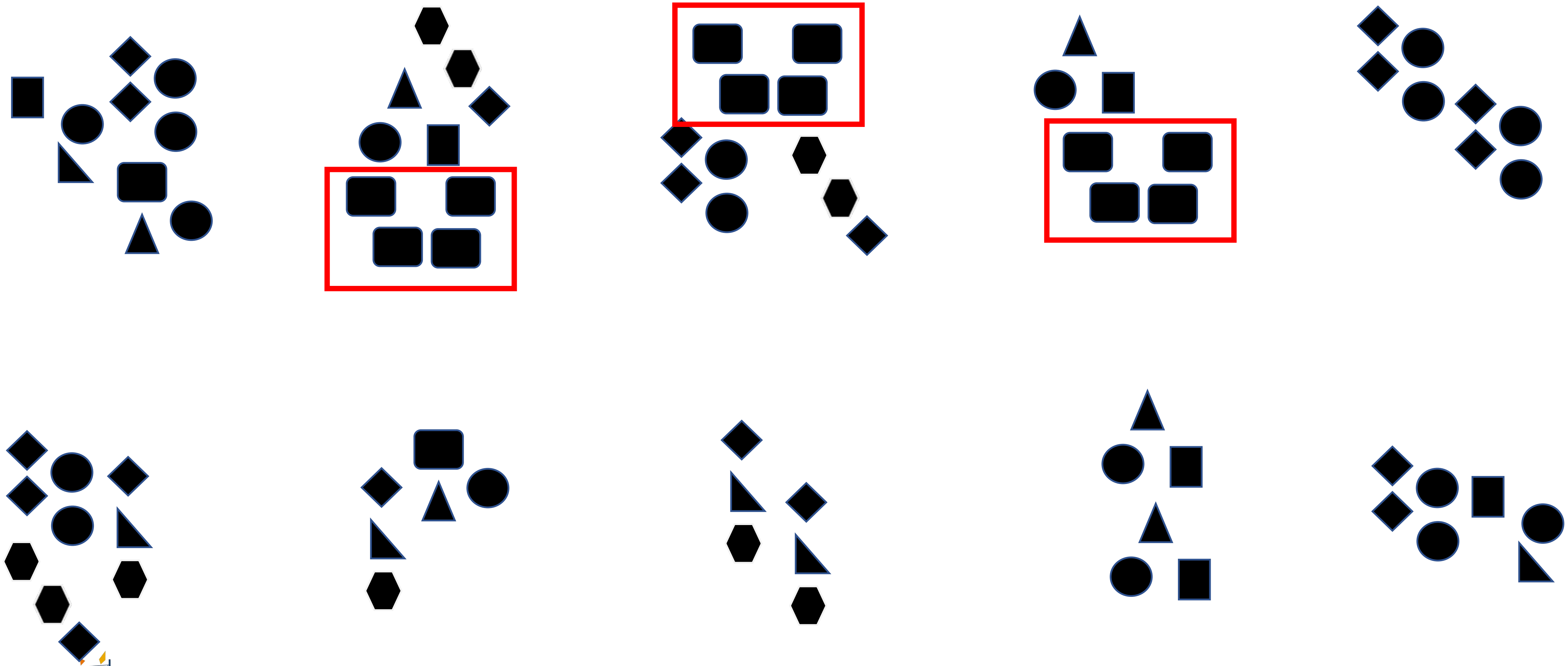
High-Level Motifs/Structure



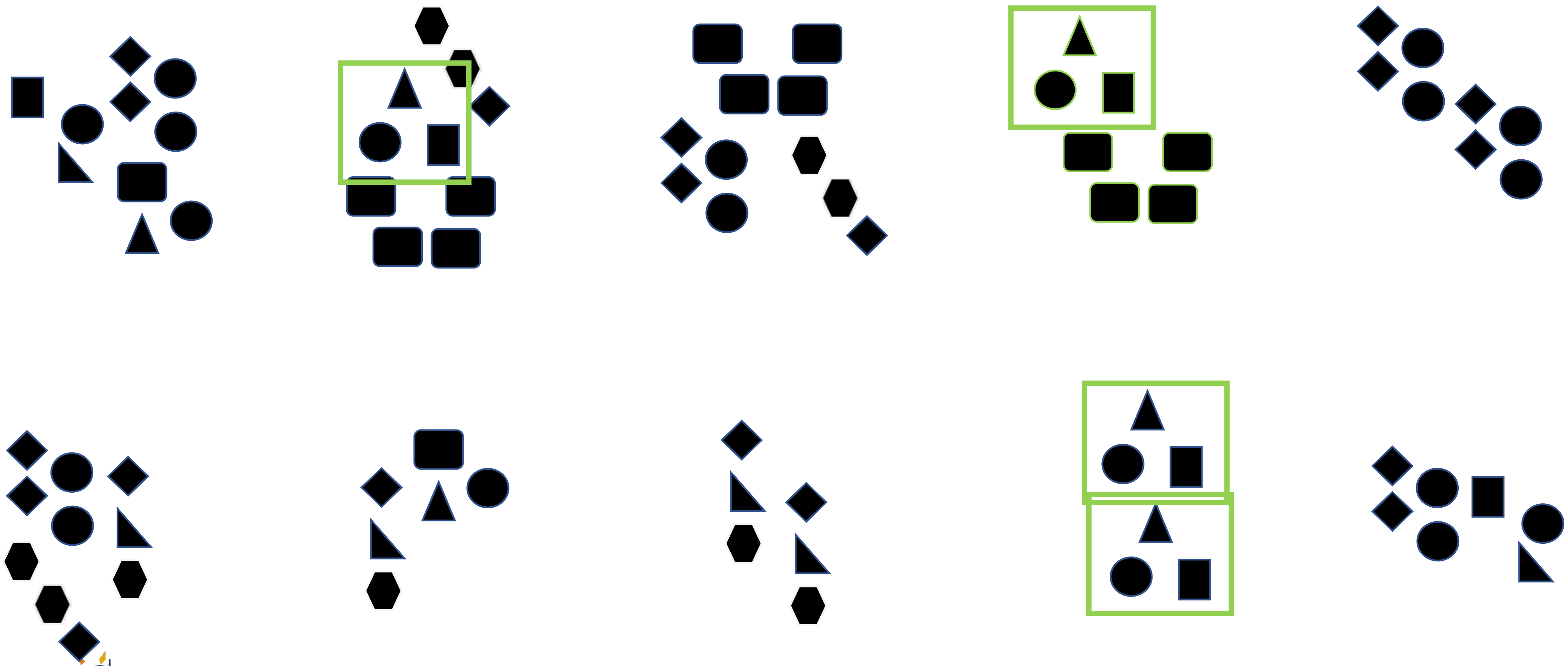
Shared Substructure Within Motifs



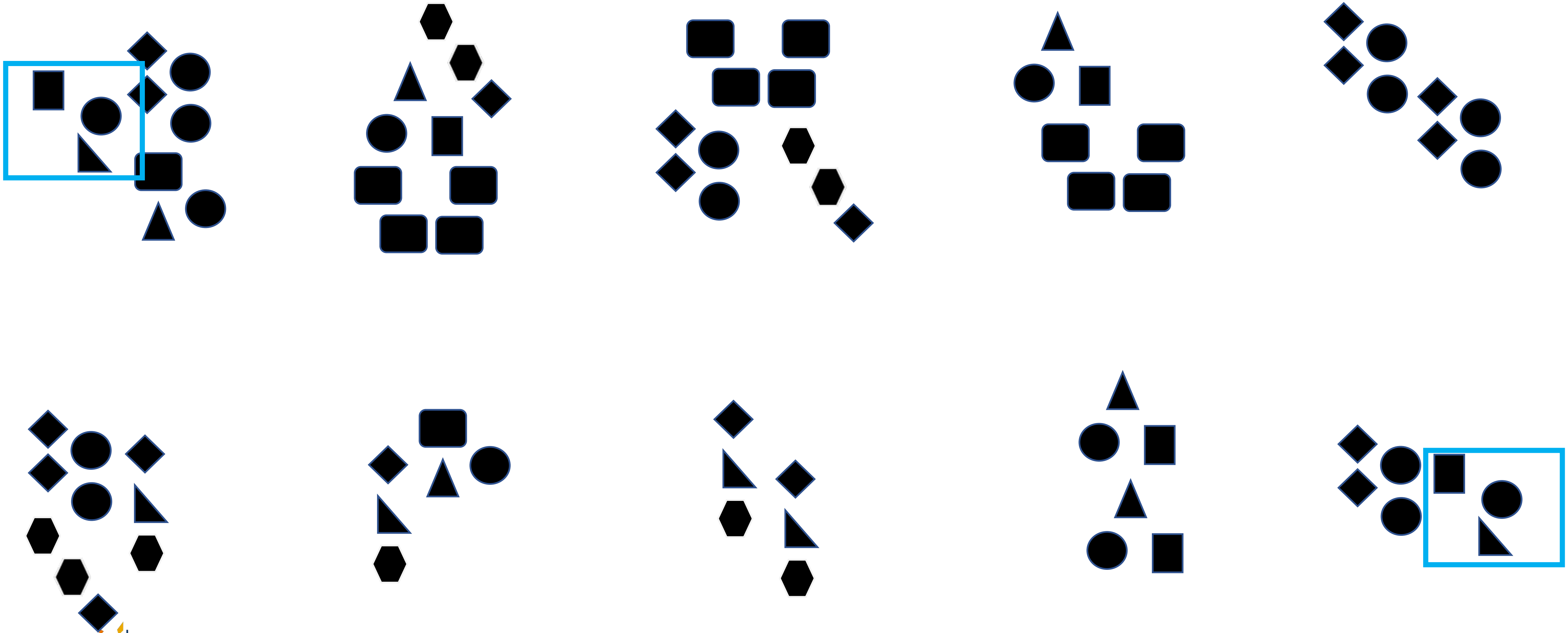
Shared Substructure Within Motifs



Shared Substructure Within Motifs



Shared Substructure Within Motifs

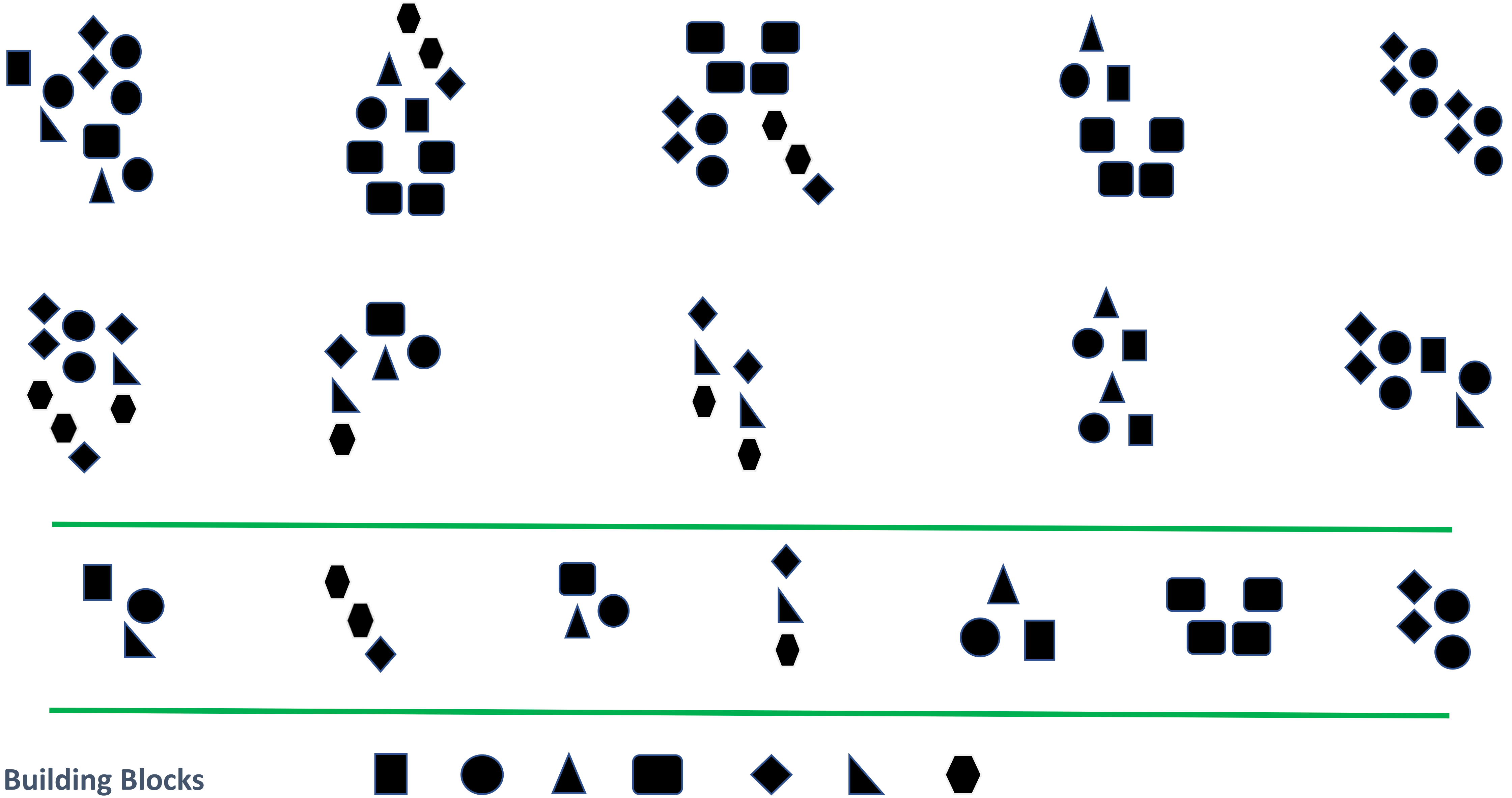


Hierarchical Representation of Images

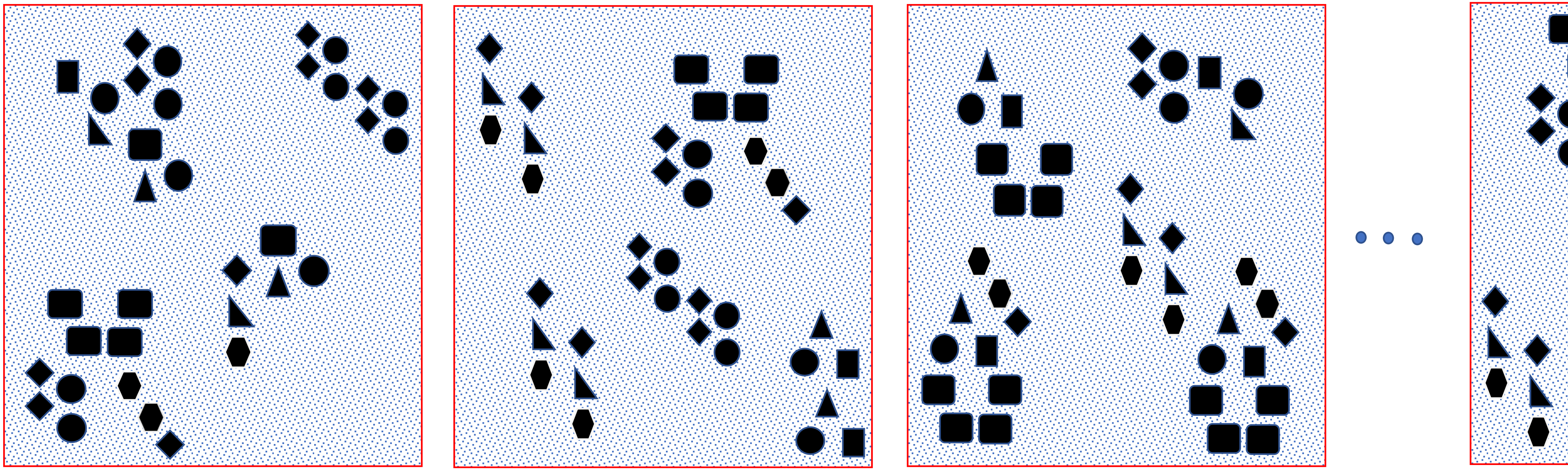
Layer 3:
Motifs

Layer 2:
Sub-Motifs

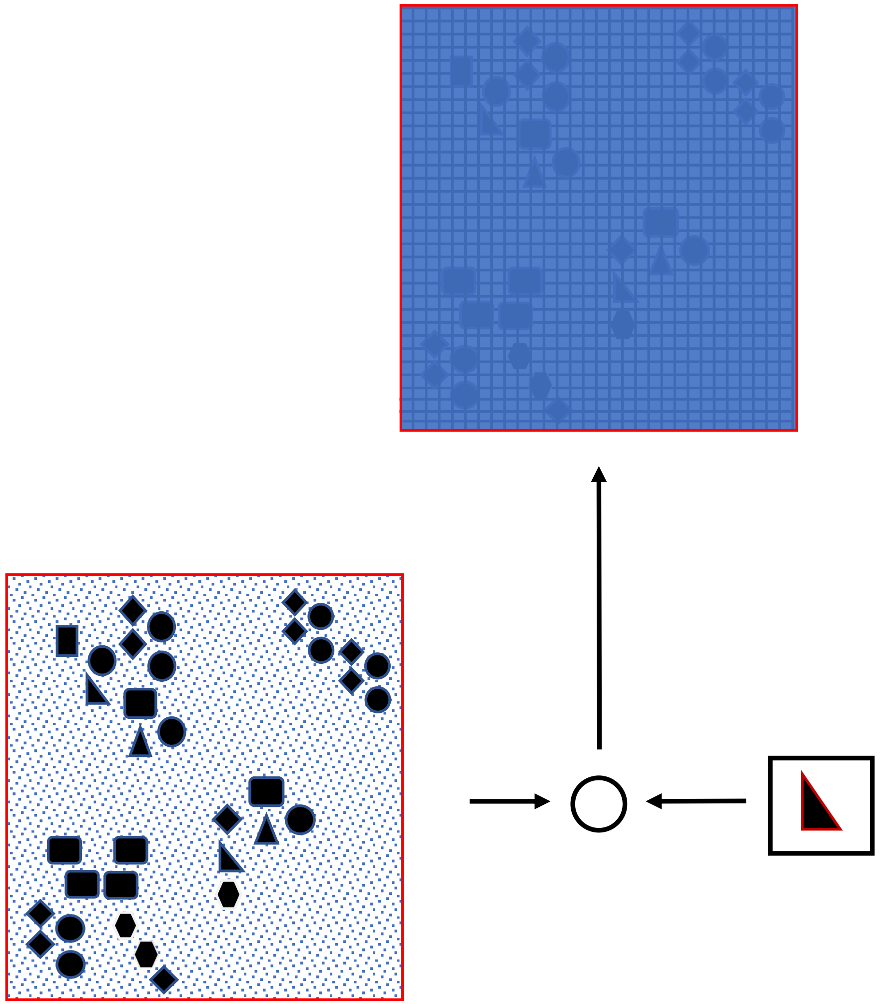
Layer 1:
Fundamental Building Blocks



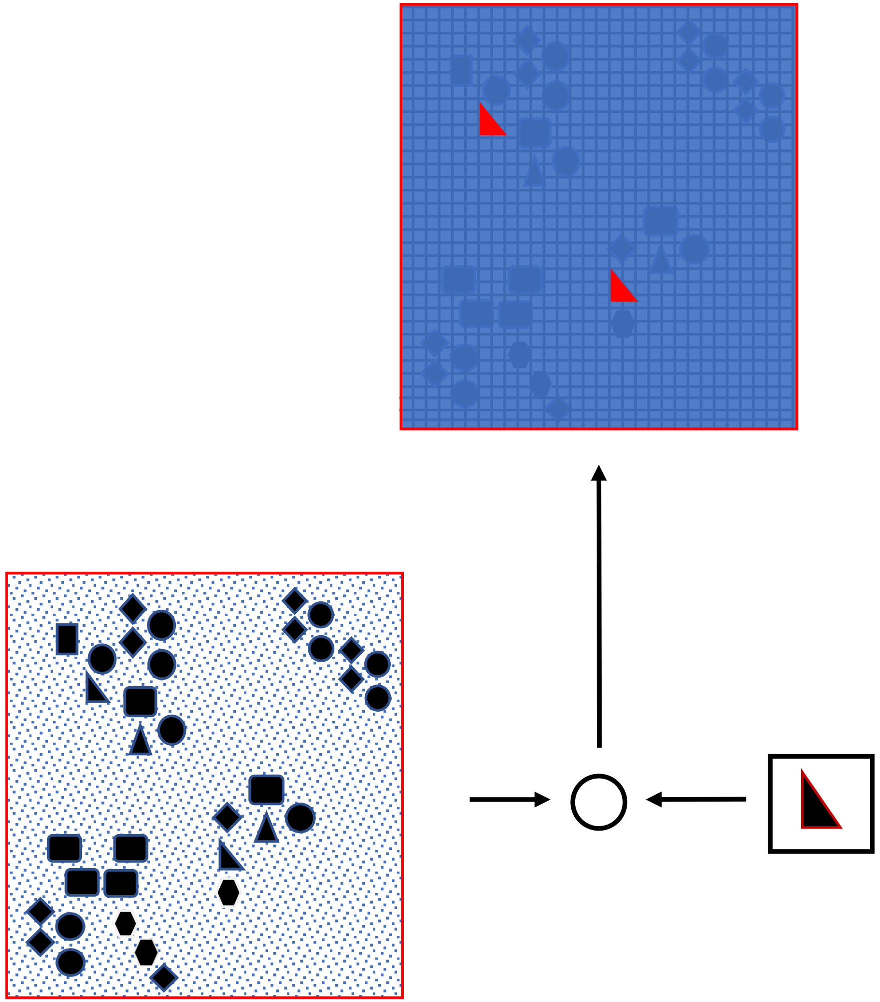
Recall the Data/Images



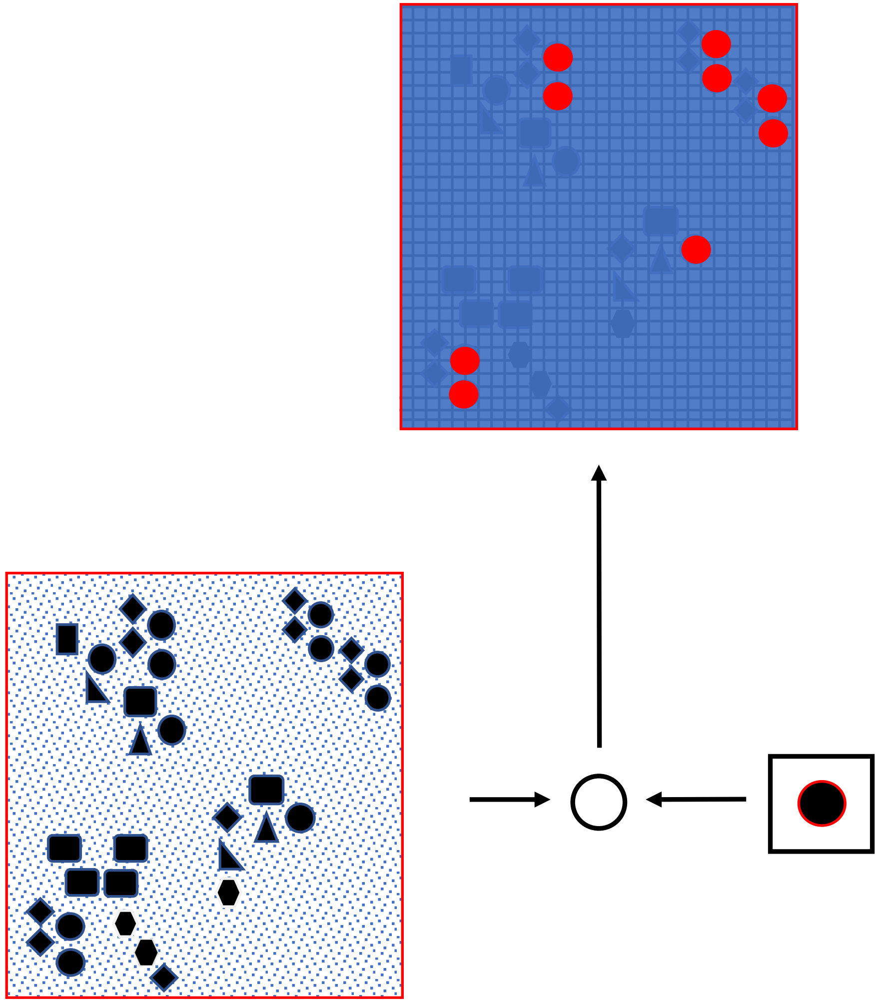
Convolutional Filter



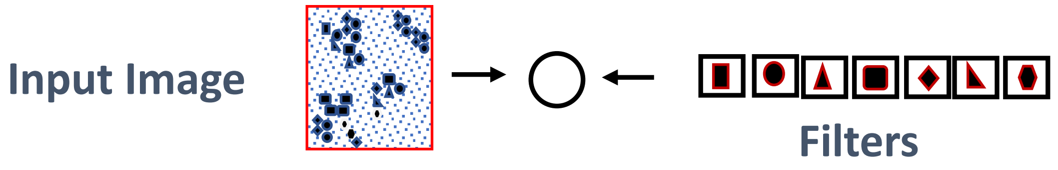
Convolutional Filter

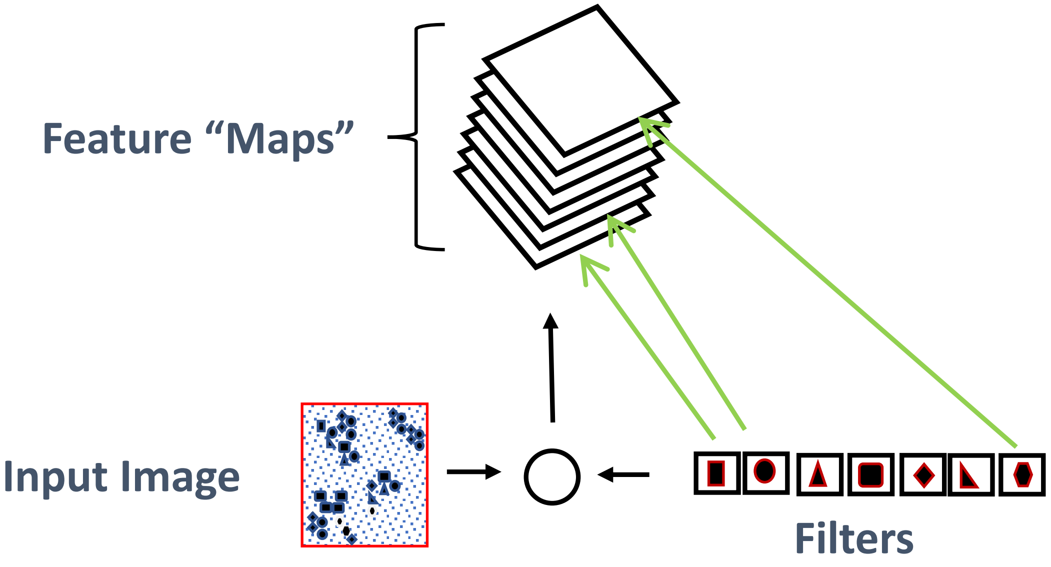


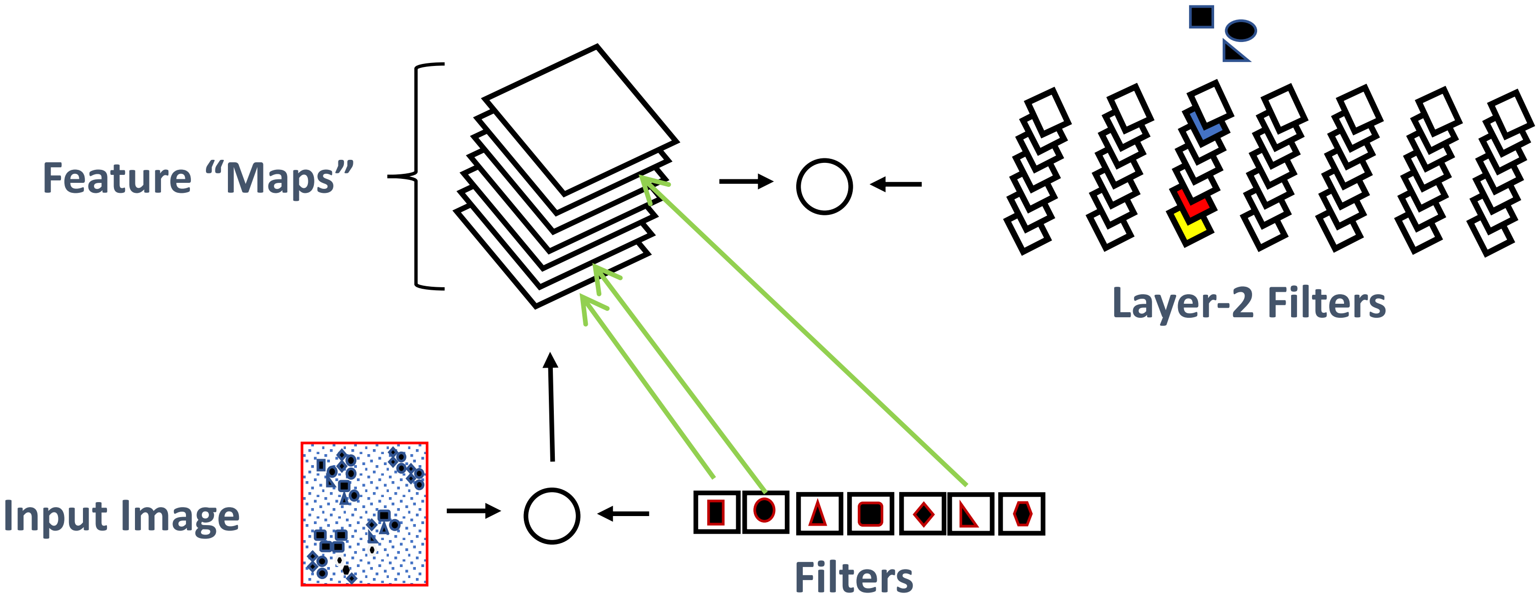
Convolutional Filter

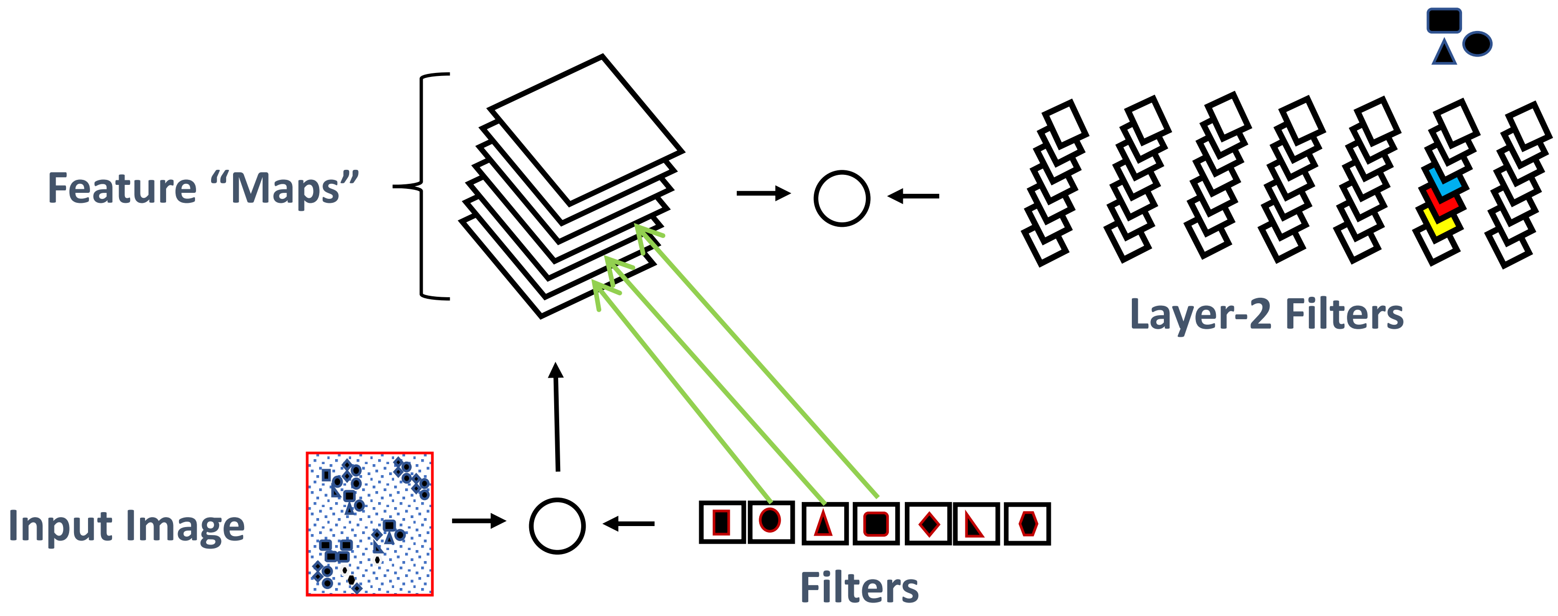


Multiple Filters, One for Each Building Block

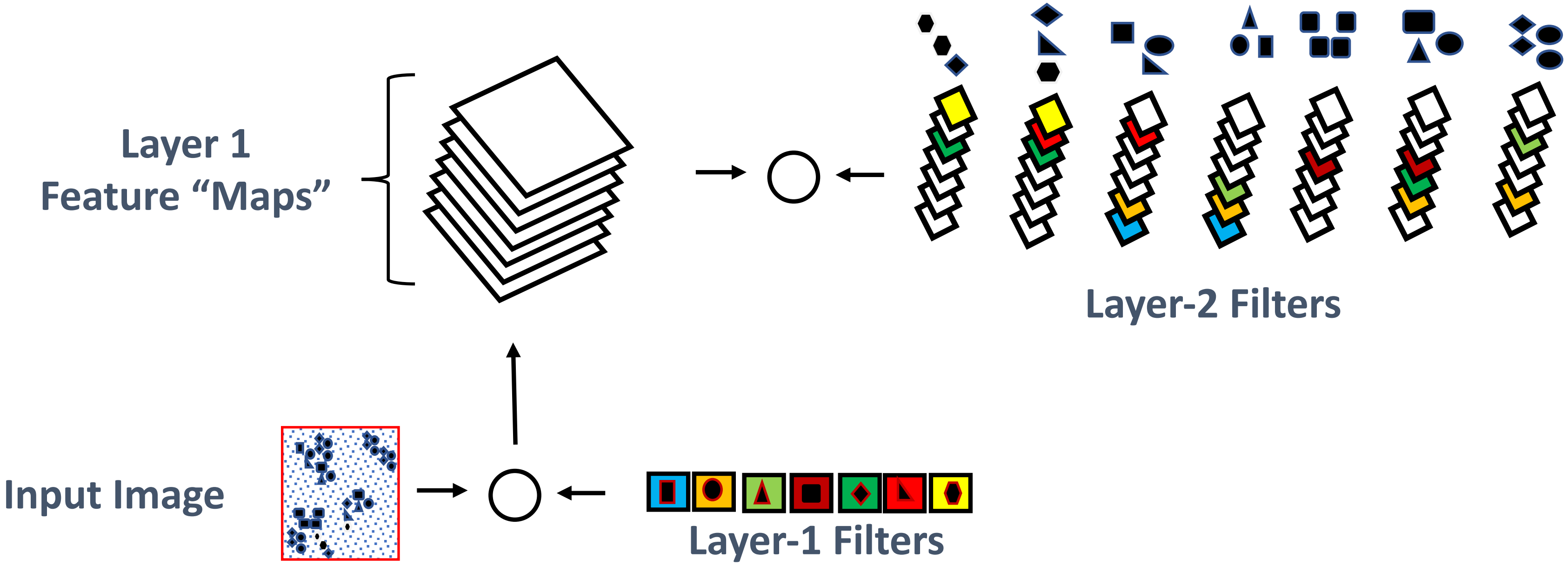


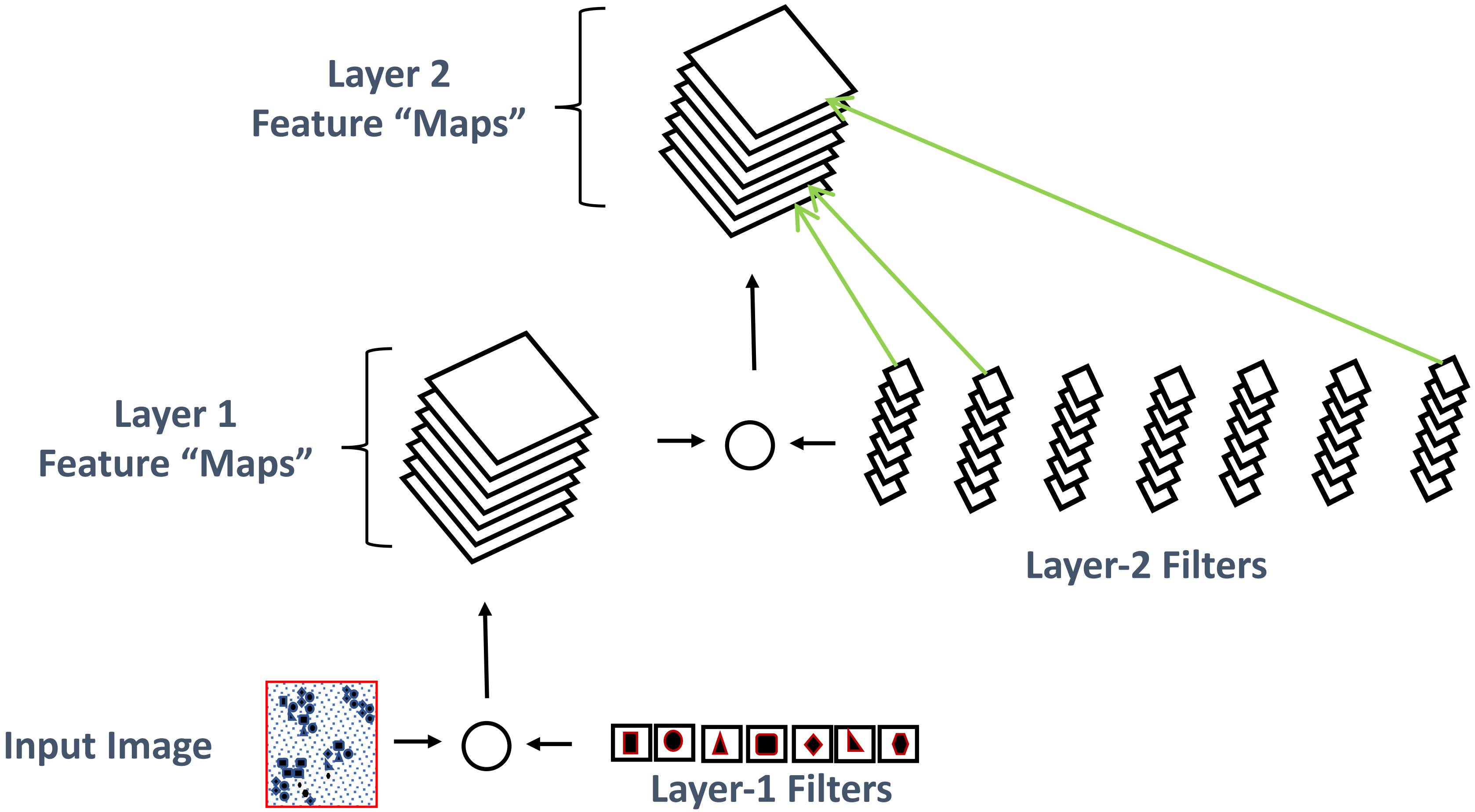

Feature "Maps"
Input Image
Filters


Feature "Maps"
Layer-2 Filters
Input Image
Filters

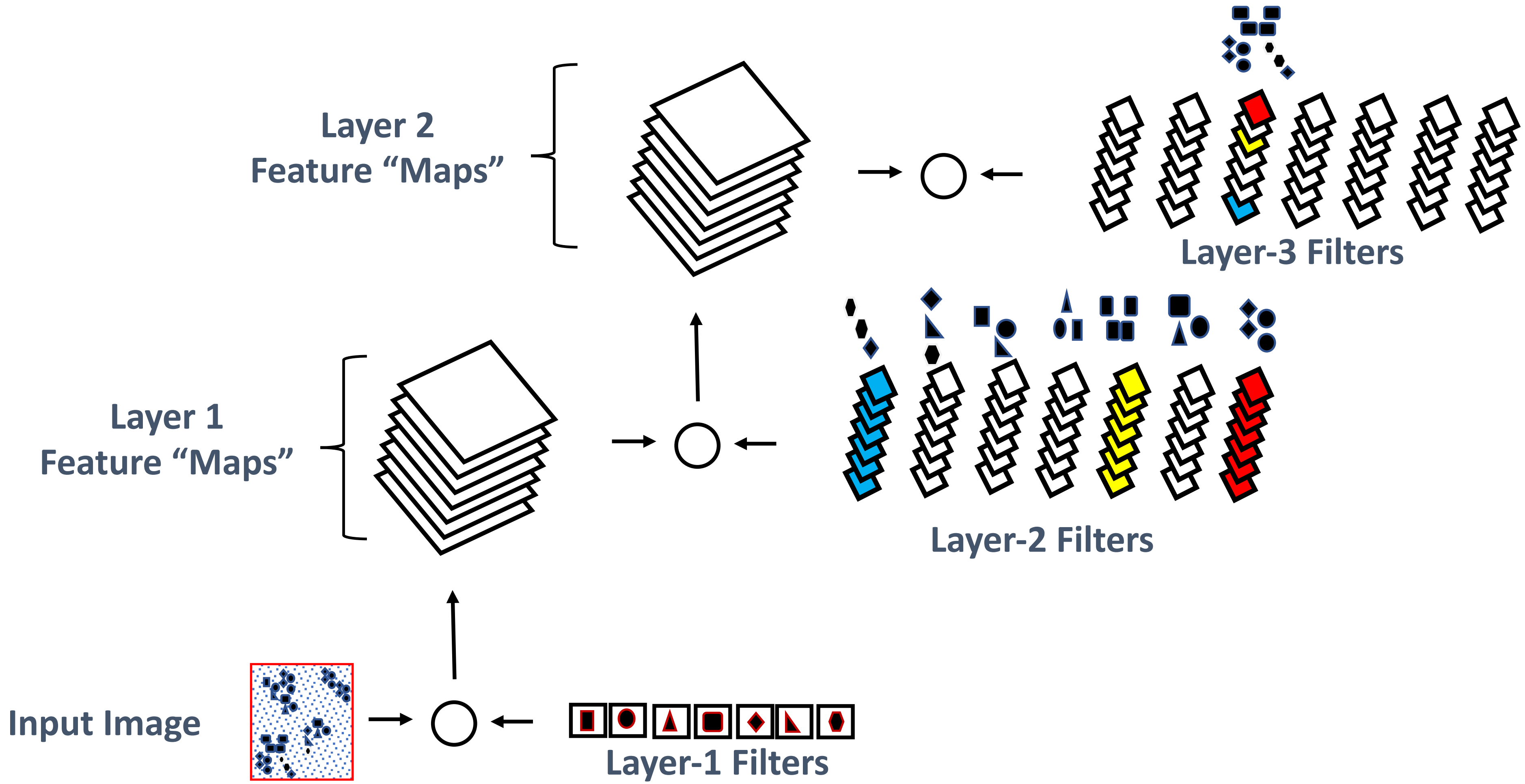


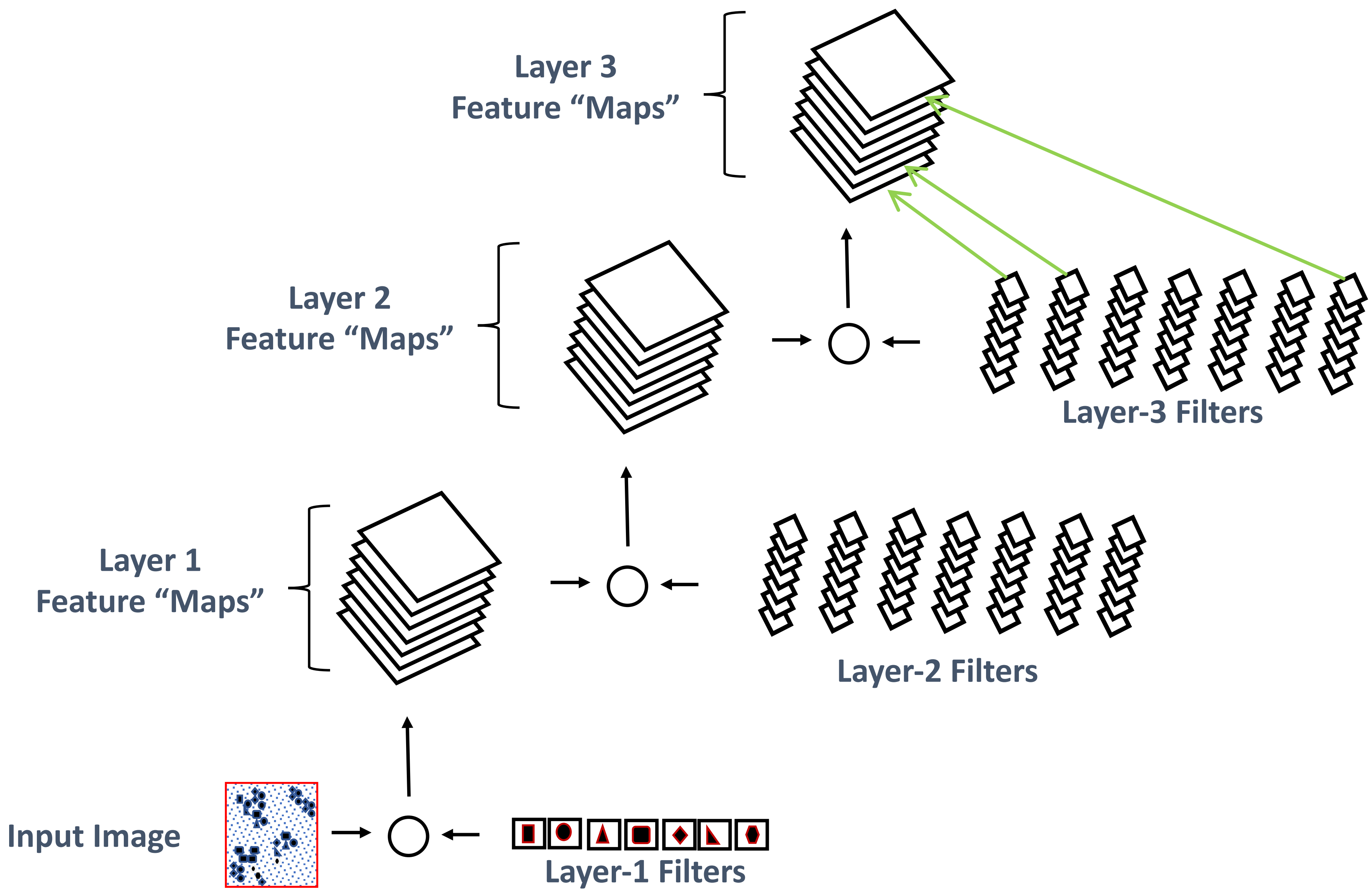
Feature "Maps"
Layer-2 Filters
Input Image
Filters


Layer 1
Feature "Maps"
Layer-2 Filters
Input Image
Layer-1 Filters



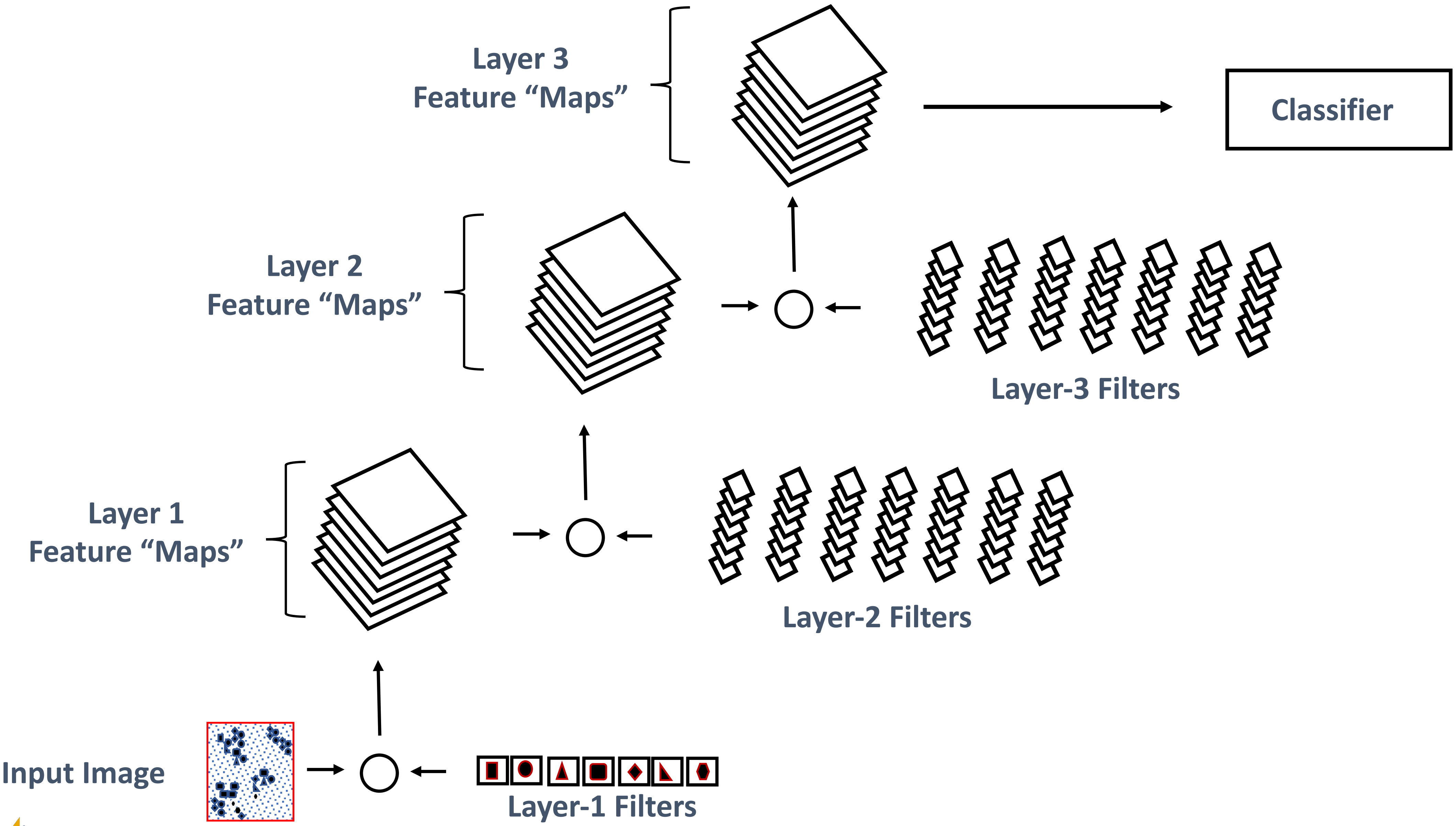
Layer 2
Feature “Maps”
Layer 1
Feature “Maps”
Layer-2 Filters
Input Image
Layer-1 Filters


Layer 2
Feature “Maps”
Layer-3 Filters
Layer 1
Feature “Maps”
Layer-2 Filters
Input Image
Layer-1 Filters



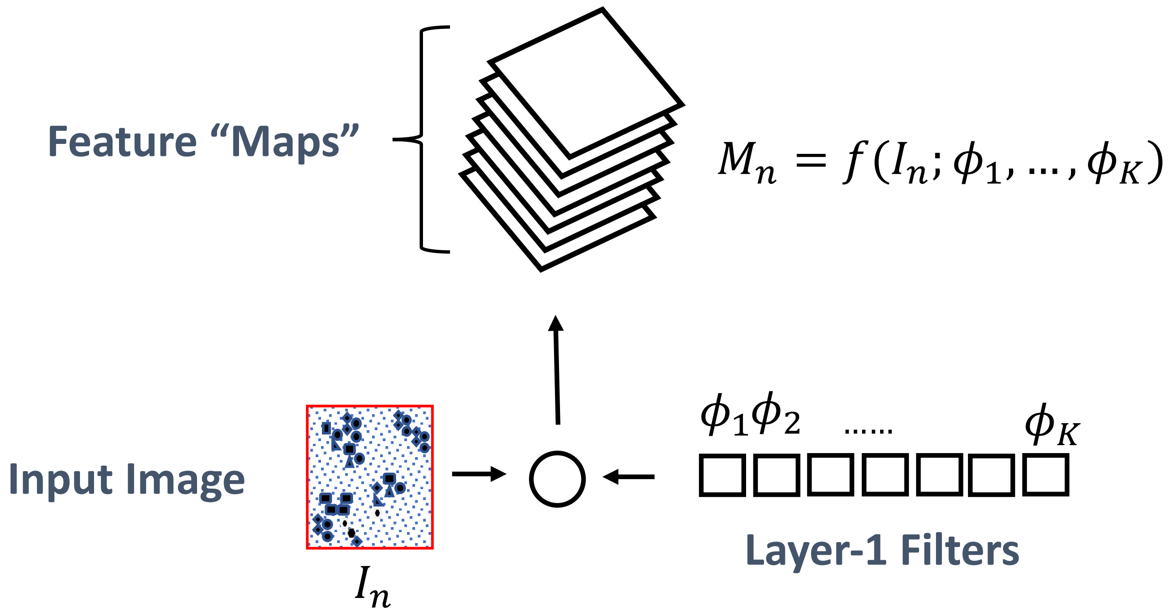
Layer 3
Feature "Maps"
Layer 2
Feature "Maps"
Layer-3 Filters
Layer 1
Feature "Maps"
Layer-2 Filters
Input Image
Layer-1 Filters

Deep Analysis Architecture

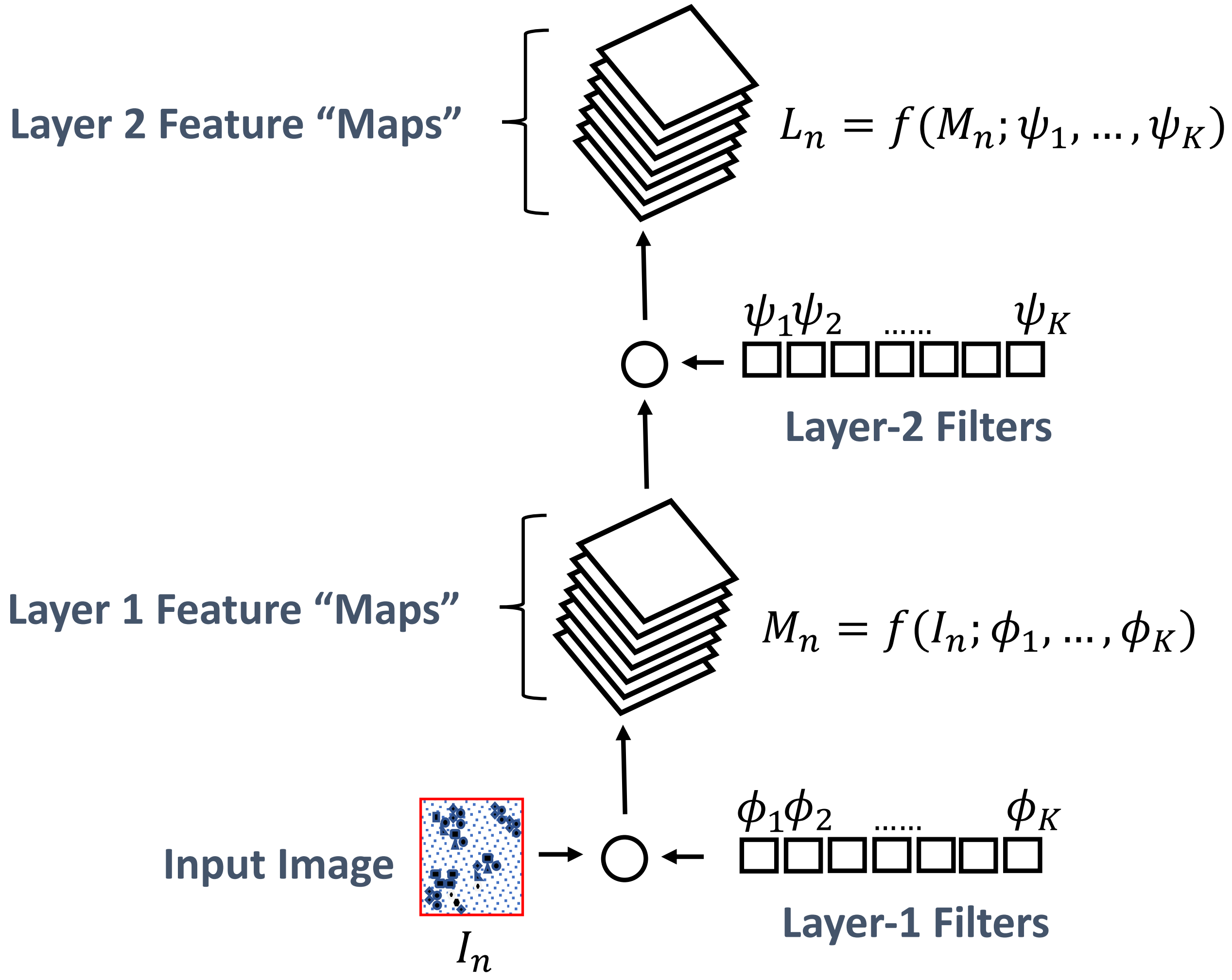


Given Images, How Do We Learn Model Parameters?

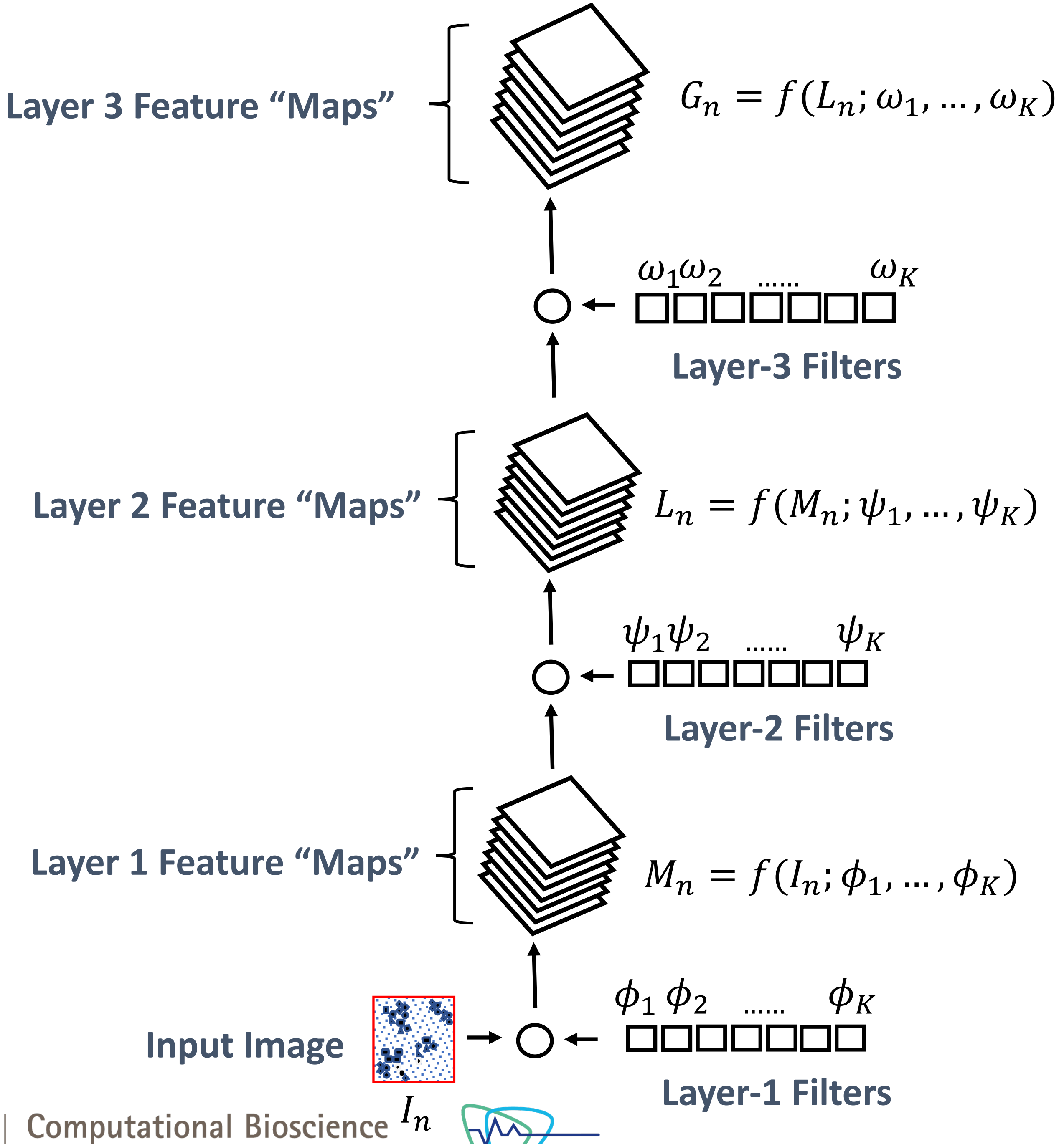
- The previous discussion was an illustration for motivating the “deep” algorithm concept
- Demonstrated using “toy” images
- How do we build such an algorithm in practice, given a large set of training images?

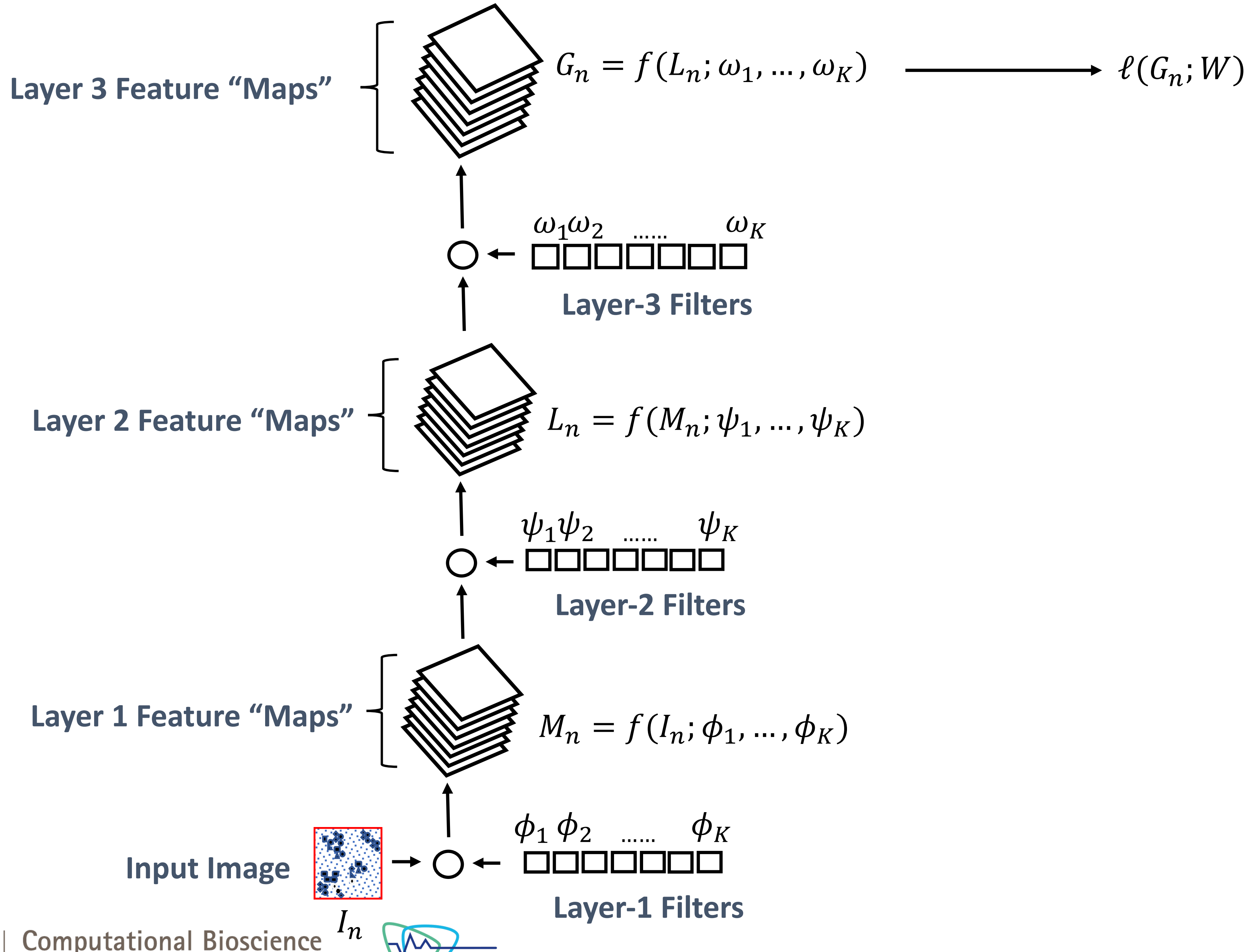


Feature “Maps”
$M_n = f(I_n; \phi_1, \dots, \phi_K)$
Input Image
I_n
$\phi_1 \phi_2$ ...... ϕ_K
Layer-1 Filters

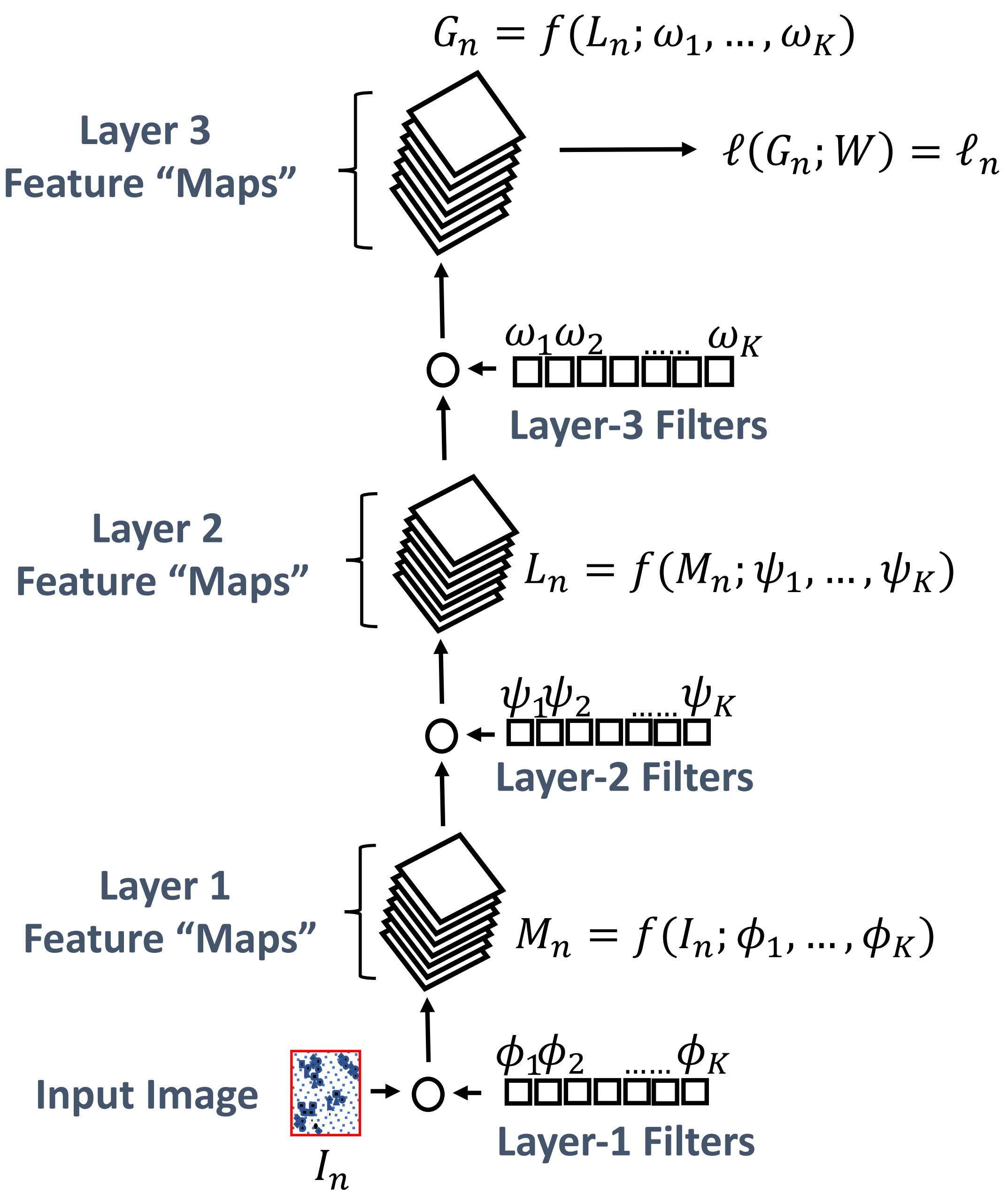


Layer 2 Feature "Maps"
$L_n = f(M_n; \psi_1, \ldots, \psi_K)$
$\psi_1 \psi_2$ ...... ψ_K
Layer-2 Filters
Layer 1 Feature "Maps"
$M_n = f(I_n; \phi_1, \ldots, \phi_K)$
Input Image
I_n
$\phi_1 \phi_2$ ...... ϕ_K
Layer-1 Filters


Layer 3 Feature “Maps”
$G_n = f(L_n; \omega_1, \dots, \omega_K)$
$\omega_1 \omega_2$ …… ω_K
Layer-3 Filters
Layer 2 Feature “Maps”
$L_n = f(M_n; \psi_1, \dots, \psi_K)$
$\psi_1 \psi_2$ …… ψ_K
Layer-2 Filters
Layer 1 Feature “Maps”
$M_n = f(I_n; \phi_1, \dots, \phi_K)$
$\phi_1 \phi_2$ …… ϕ_K
Input Image
I_n
Layer-1 Filters



Layer 3 Feature "Maps"
$G_n = f(L_n; \omega_1, \ldots, \omega_K)$
$\ell(G_n; W)$
$\omega_1 \omega_2$ ...... ω_K
Layer-3 Filters
Layer 2 Feature "Maps"
$L_n = f(M_n; \psi_1, \ldots, \psi_K)$
$\psi_1 \psi_2$ ...... ψ_K
Layer-2 Filters
Layer 1 Feature "Maps"
$M_n = f(I_n; \phi_1, \ldots, \phi_K)$
$\phi_1\ \phi_2$ ...... ϕ_K
Input Image
I_n
Layer-1 Filters

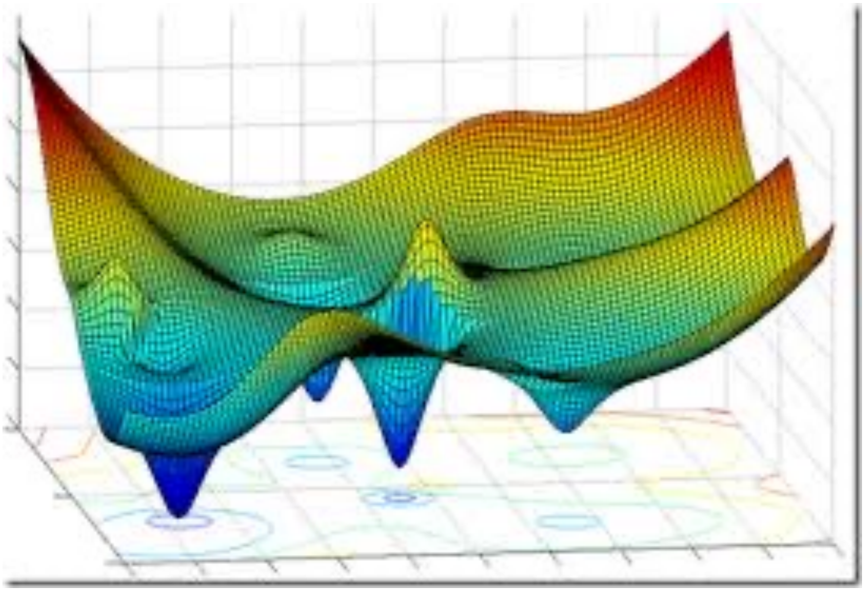


- Assume we have labeled images $\{I_n, y_n\}_{n=1,N}$
- I_n is image n, $y_n \in \{+1, -1\}$ is associated label
- Risk function of model parameters:

$$E(\Phi, \Psi, \Omega, W) = 1/N \sum_{n=1}^{N} loss(y_n, \ell_n)$$

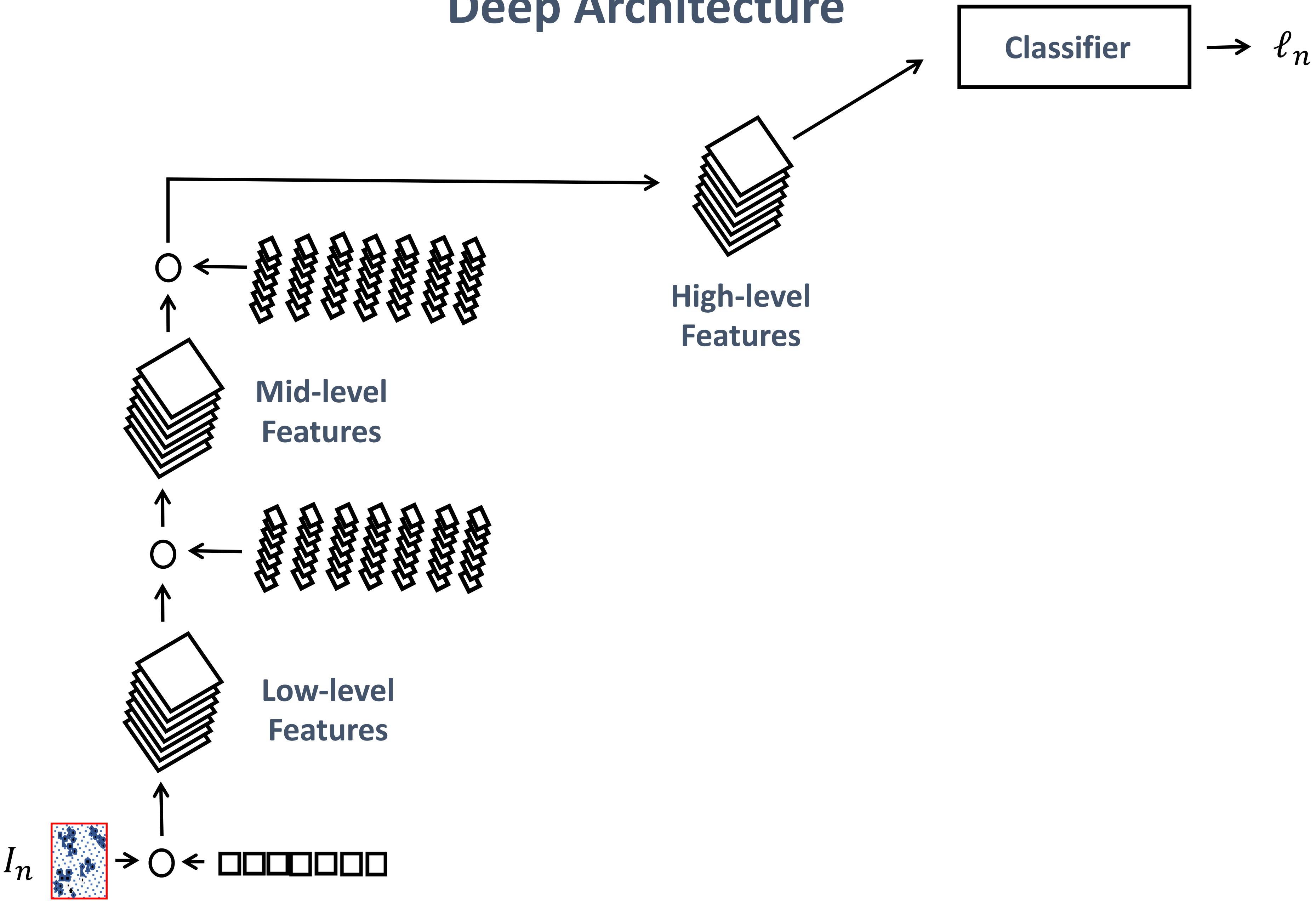
- Find model parameters $\widehat{\Phi}, \widehat{\Psi}, \widehat{\Omega}, \widehat{W}$ that minimize $E(\Phi, \Psi, \Omega, \mathrm{W})$

Cost Function *vs.* Model Parameters



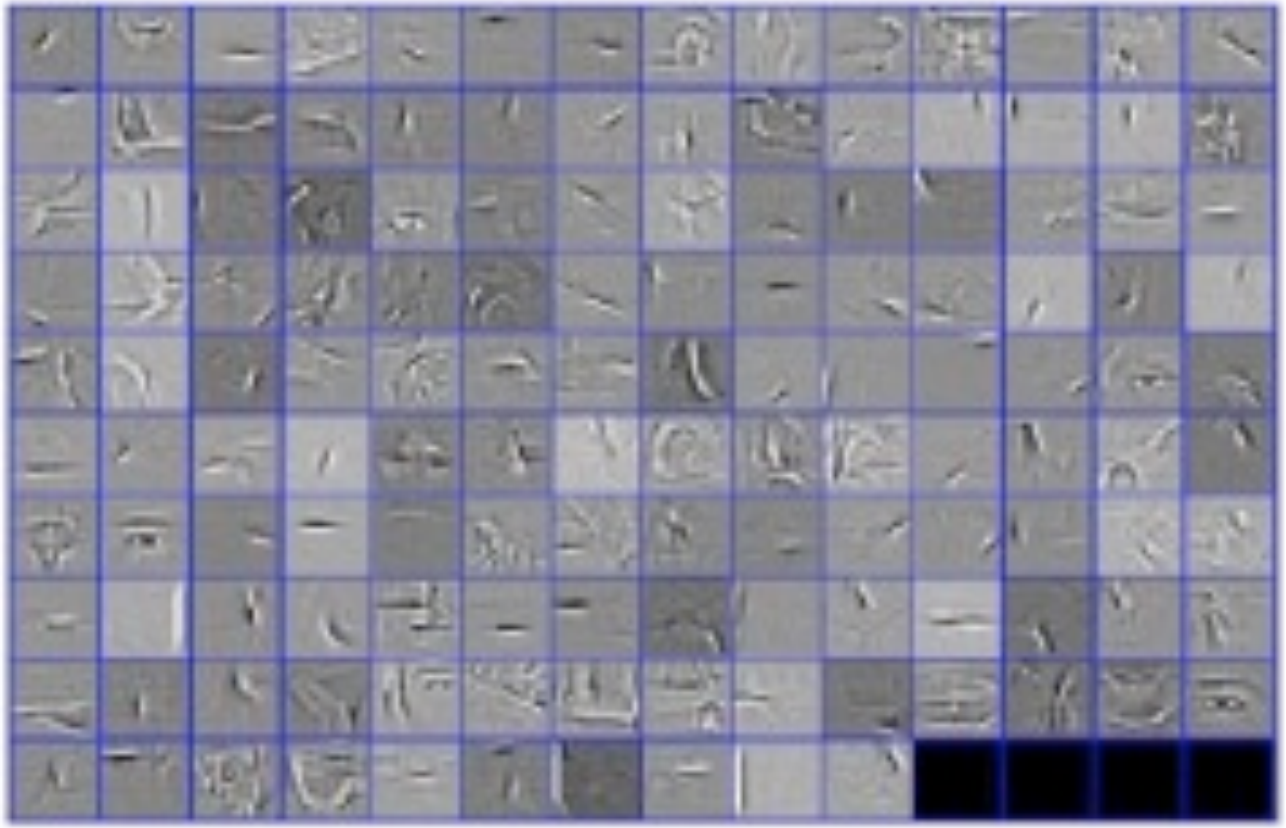
- High-dimensional function, as a consequence of a large number of model parameters
- Typically, many local minima
- May be expensive to compute, for sophisticated models & large quantity of training images

Deep Architecture

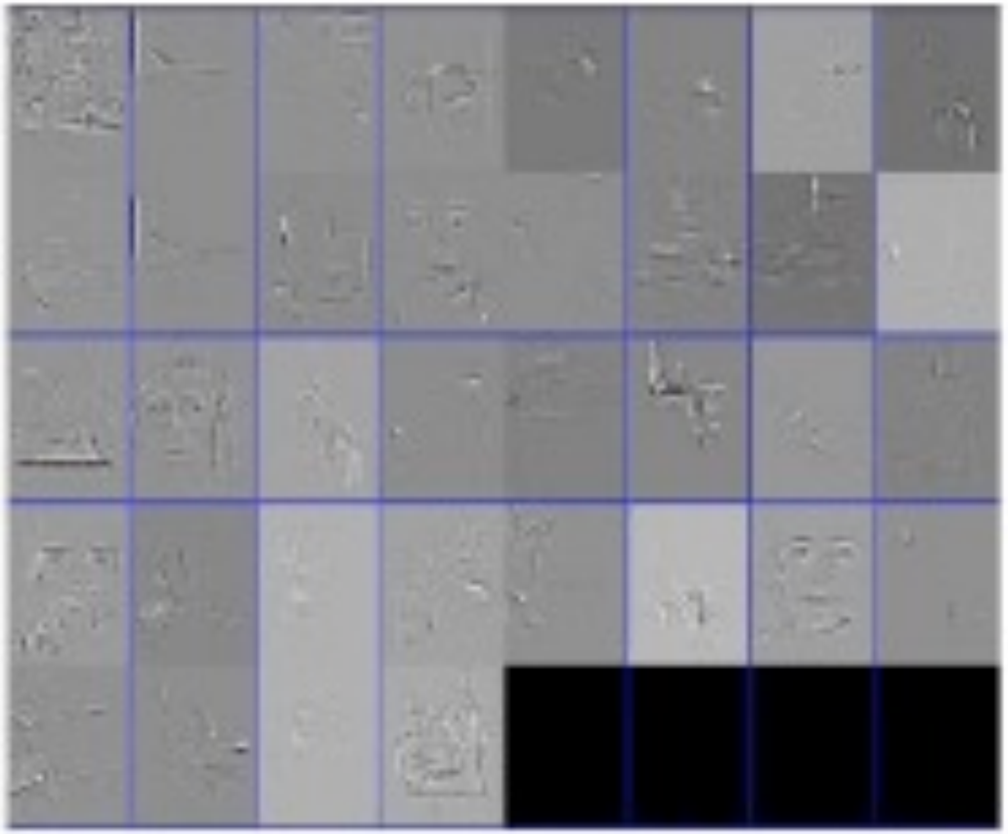


Layer 1

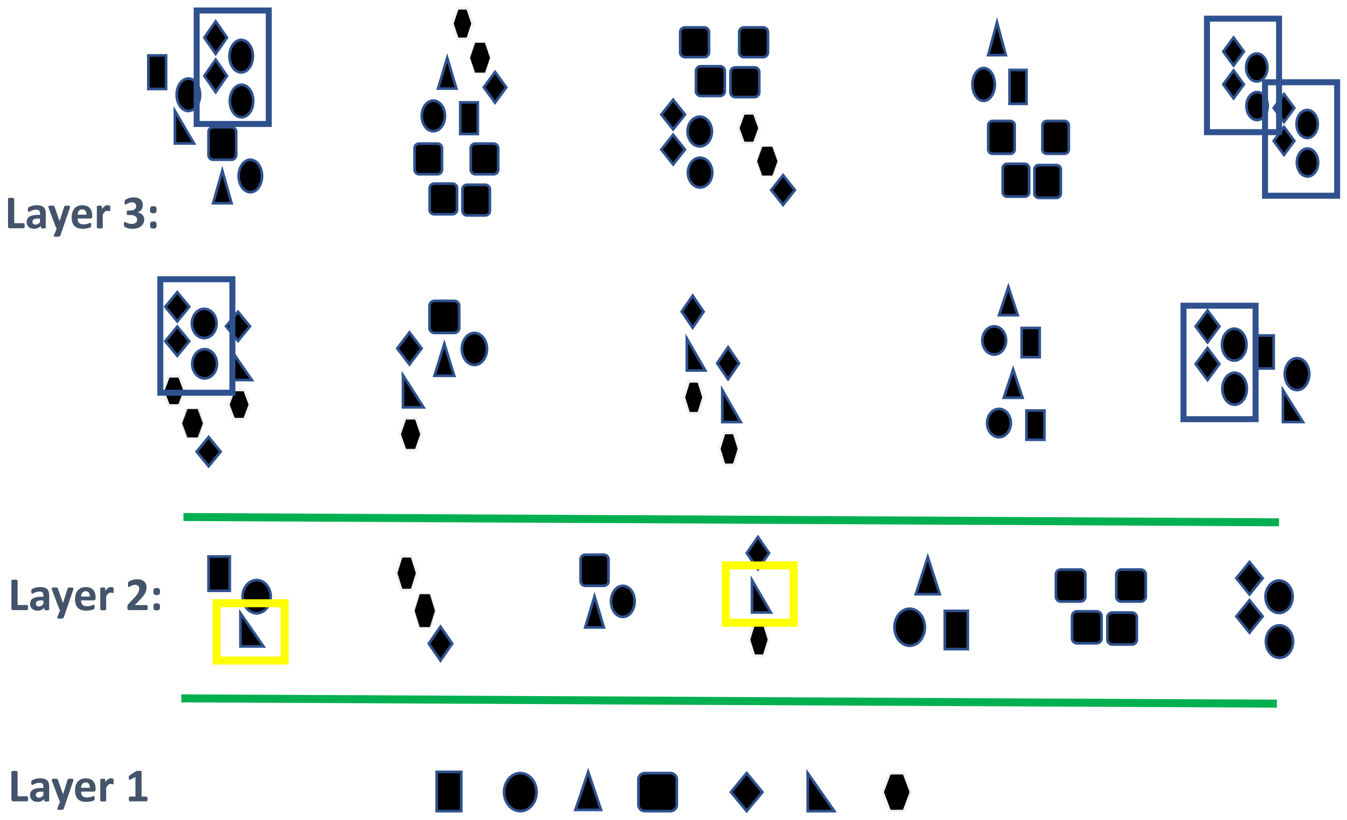
Layer 2



Layer 3



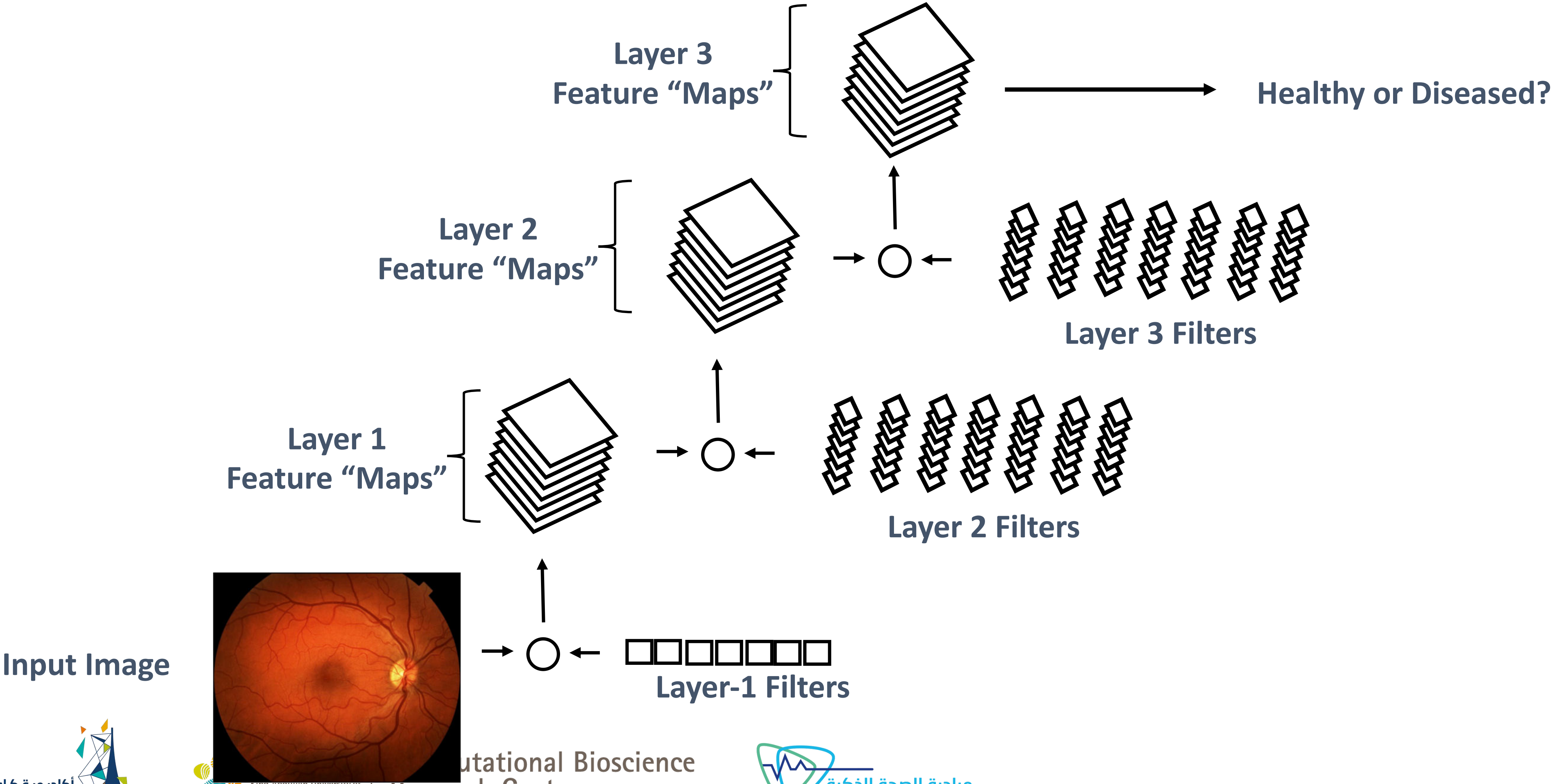
Advantage of Hierarchical Features?



- By learning and sharing statistical similarities within high-level motifs, we better leverage all training data

- If we do not use such a hierarchy, top-level motifs would be learned in isolation of each other

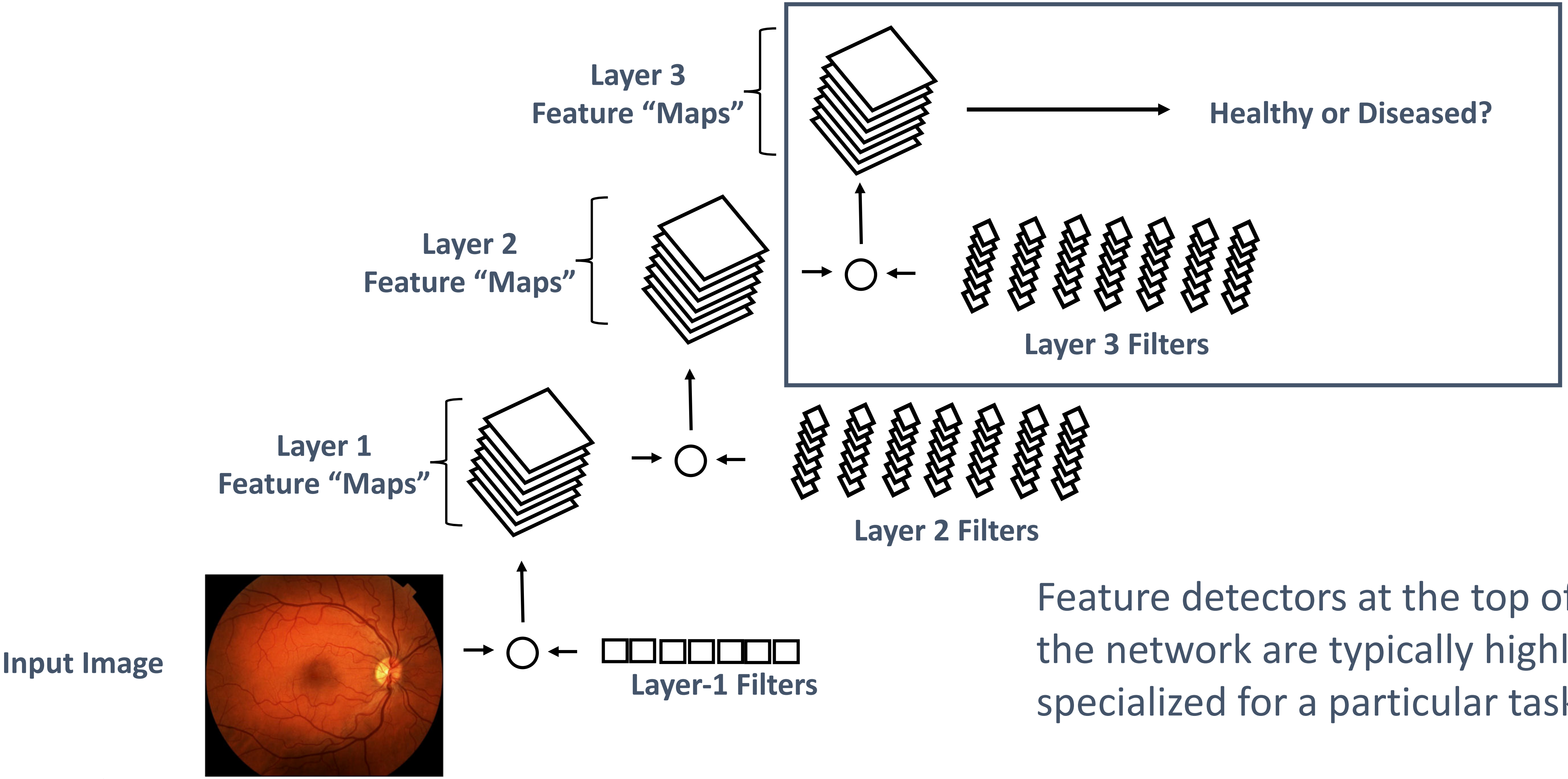
Transfer Learning



Transfer Learning

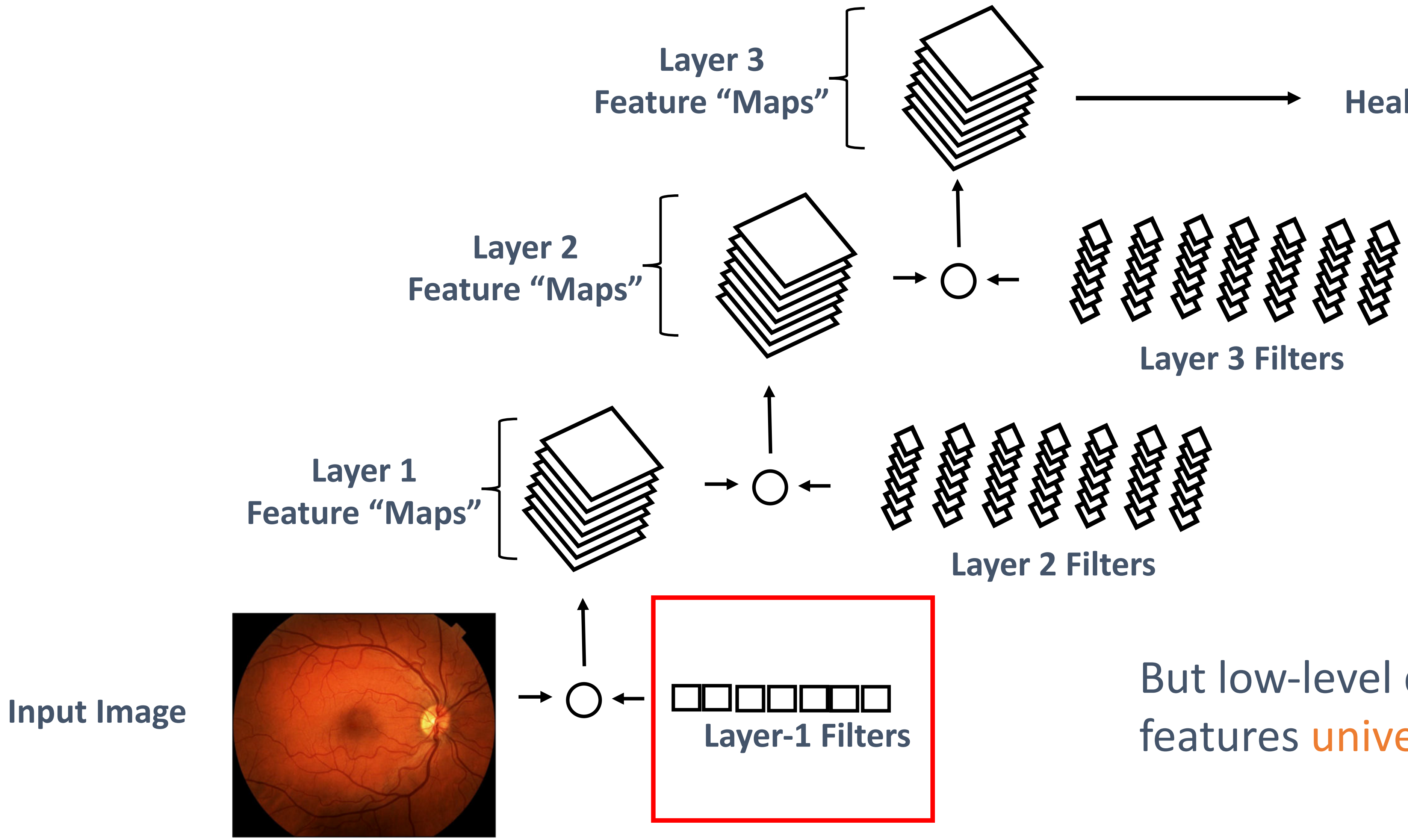
"To speed up the training, batch normalization as well as pre-initialization using weights from the same network trained to classify objects in the ImageNet data set were used. Pre-initialization also improved performance"

Transfer Learning



Feature detectors at the top of the network are typically highly specialized for a particular task

Transfer Learning

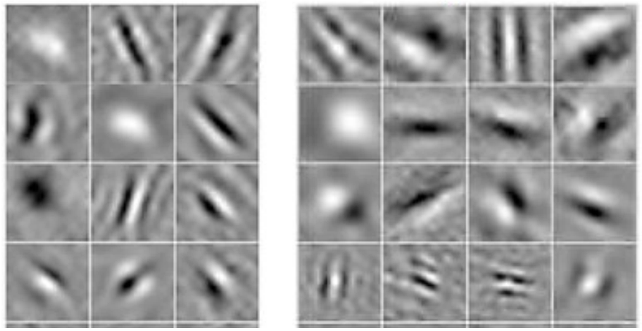


But low-level detectors extract features universal to all images

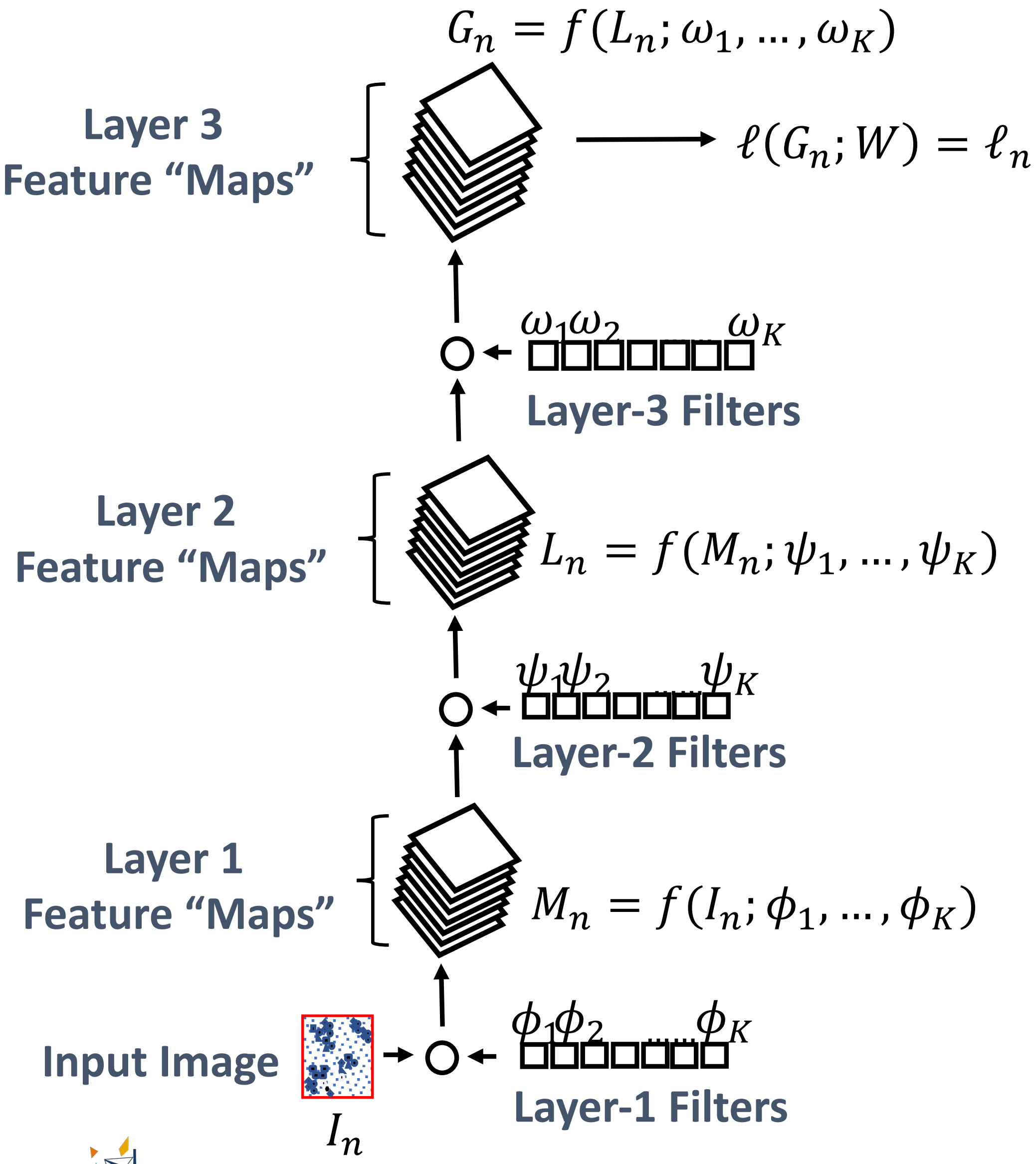
Transfer Learning

Layer 1 Filters,
Convolutional Neural Network

Neuron Receptive Fields,
Macaque Visual Cortex



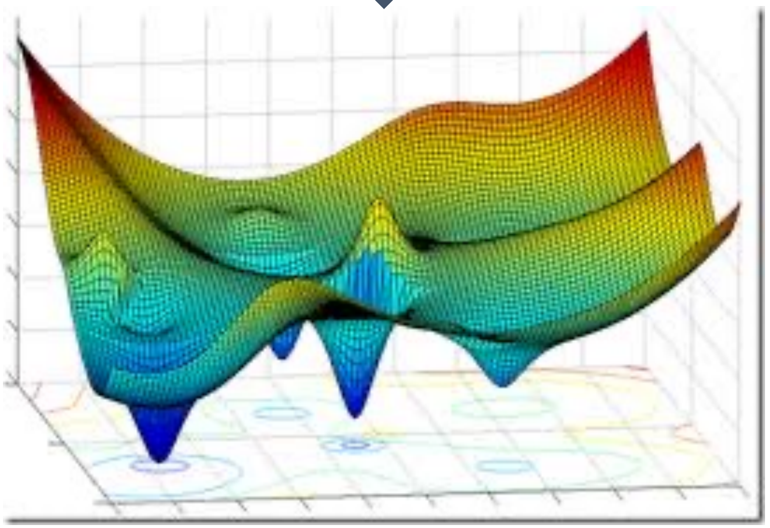
Big Picture



- Assume we have labeled images $\{I_n, y_n\}_{n=1,N}$
- I_n is image n, $y_n \in \{+1, -1\}$ is associated label
- Risk function of model parameters:

$$E(\Phi, \Psi, \Omega, W) = 1/N \sum_{n=1}^{N} loss(y_n, \ell_n)$$

- Find model parameters $\widehat{\Phi}, \widehat{\Psi}, \widehat{\Omega}, \widehat{W}$ that minimize $E(\Phi, \Psi, \Omega, \mathrm{W})$



Summary

- Convolutional neural networks learn to recognize **high-level structure** in images by building **hierarchical representations of features**

- Features are extracted via spatial convolutions with **filters**

- Filters are learned via iterative minimization of a risk function

- Convolutional neural networks have shown capabilities beyond human performance for image analysis