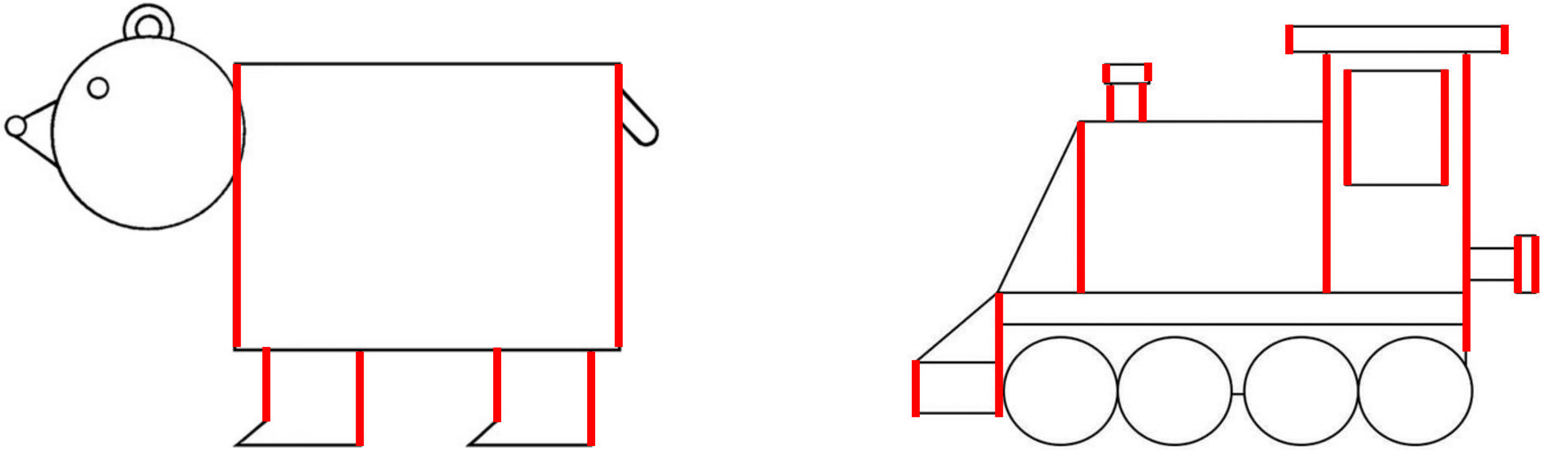
Deep CNNs

Matthew Engelhard

Many slides created by Tim Dunn

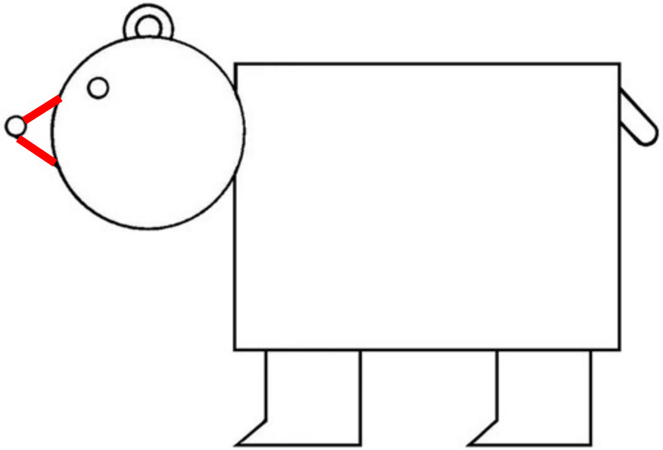
CNNs Take Advantage of Repeated, Hierarchical Structure in Images

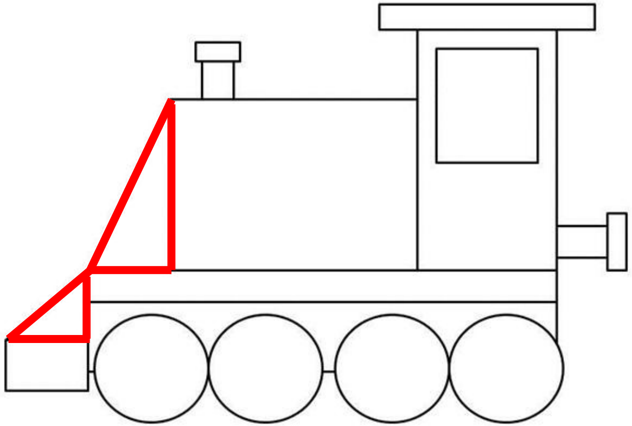
CNNs Take Advantage of Repeated, Hierarchical Structure in Images



Low-level structure: lines, curves

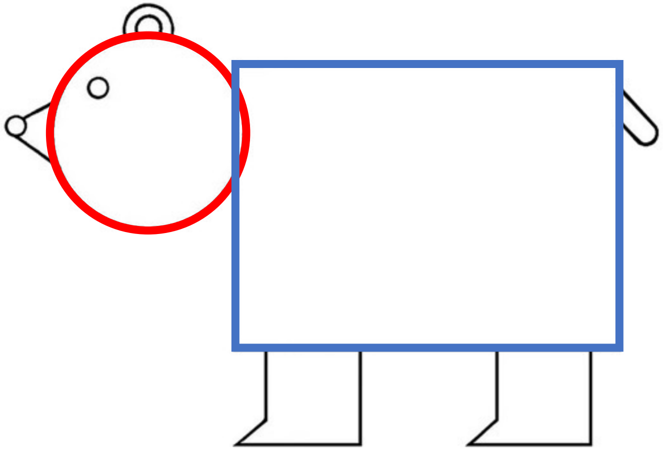
CNNs Take Advantage of Repeated, Hierarchical Structure in Images

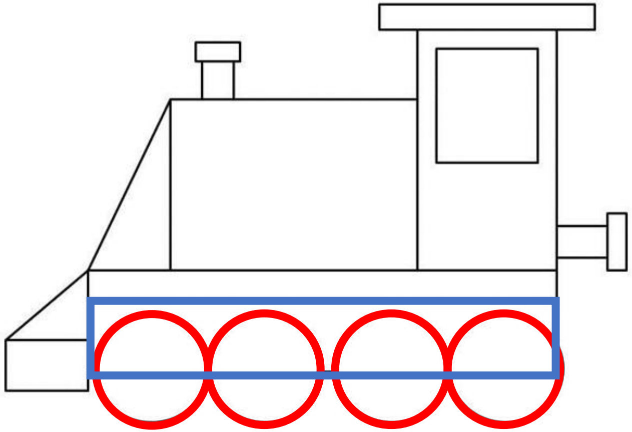




Mid-level structure: shapes

CNNs Take Advantage of Repeated, Hierarchical Structure in Images

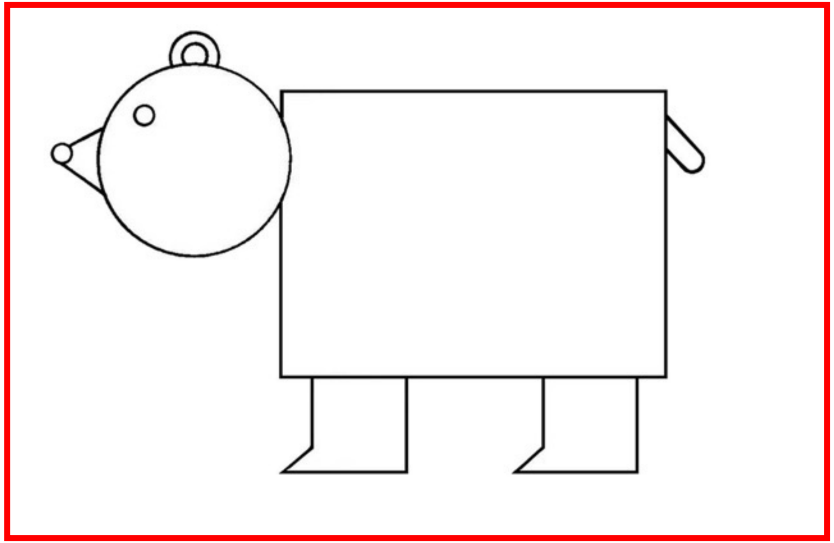




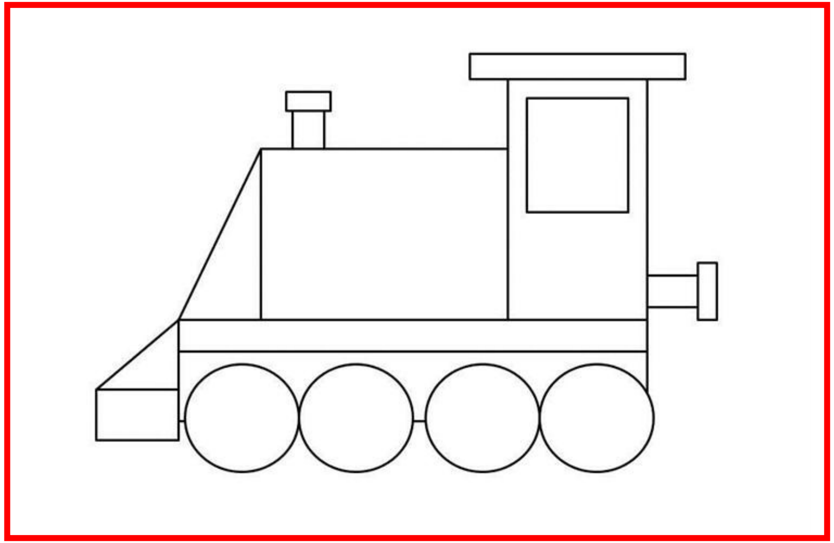
High-level structure: groups of shapes

CNNs Take Advantage of Repeated, Hierarchical Structure in Images

Bear



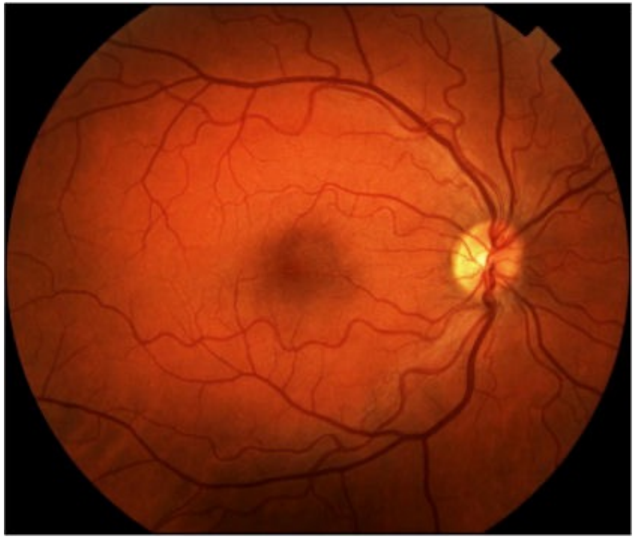
Train



High-level structure: groups of shapes $\rightarrow$ objects

Deep Learning for Image Analysis

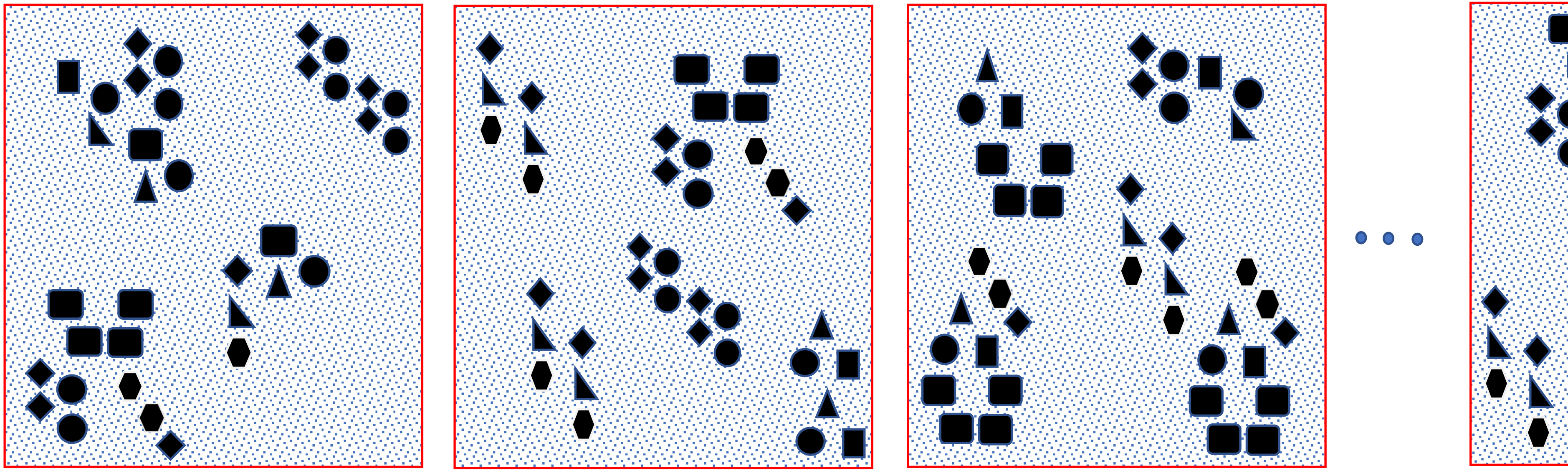
Diabetic Retinopathy
Classification



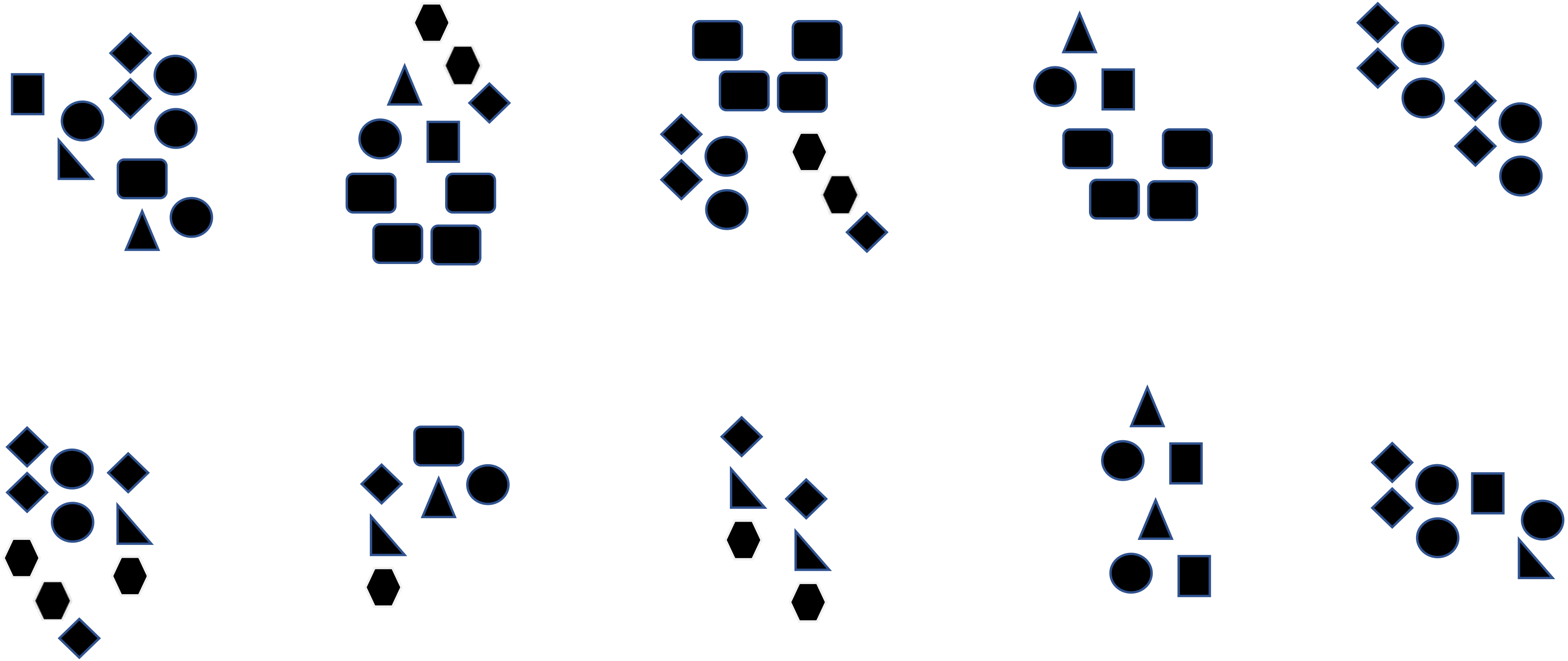
Healthy Retina

Unhealthy Retina

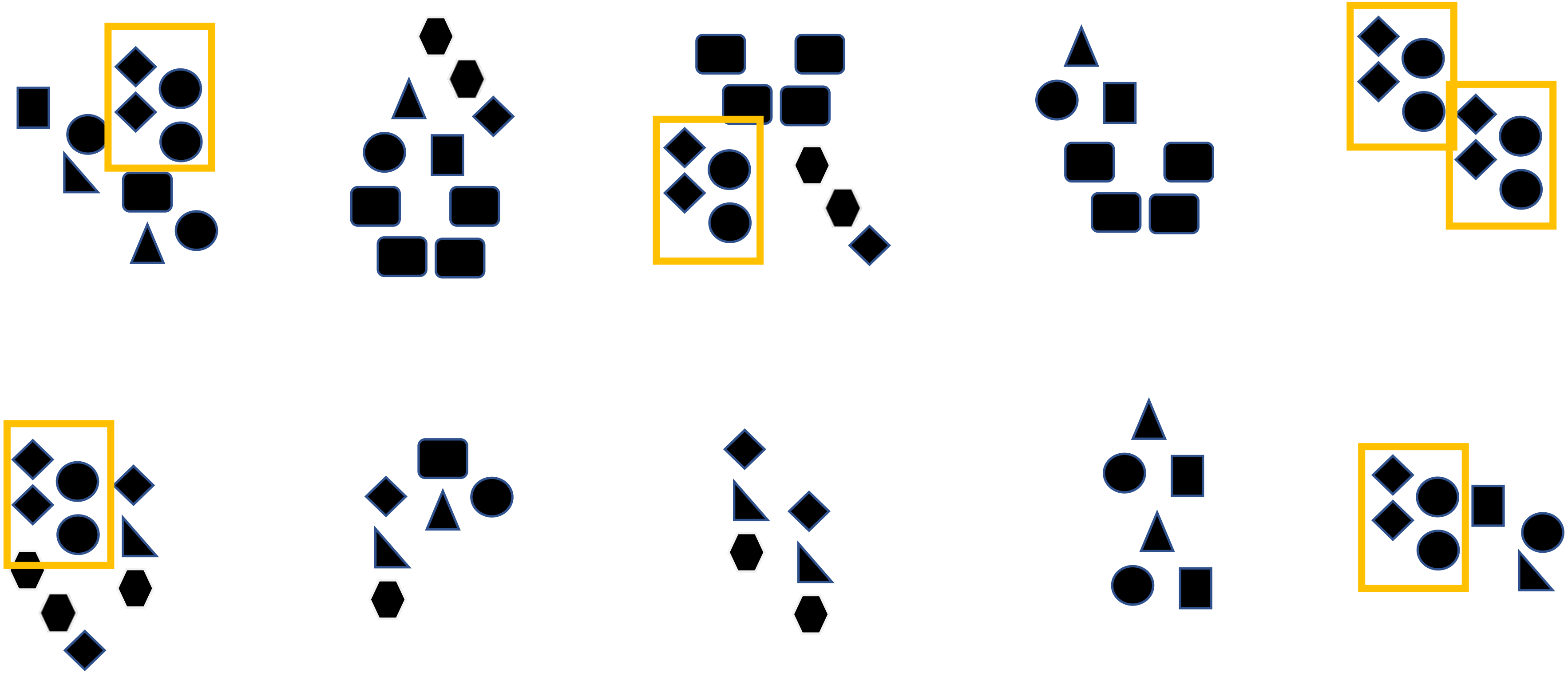
Consider a Set of "Toy" Images,
for illustration of how this structure can be extracted by an algorithm



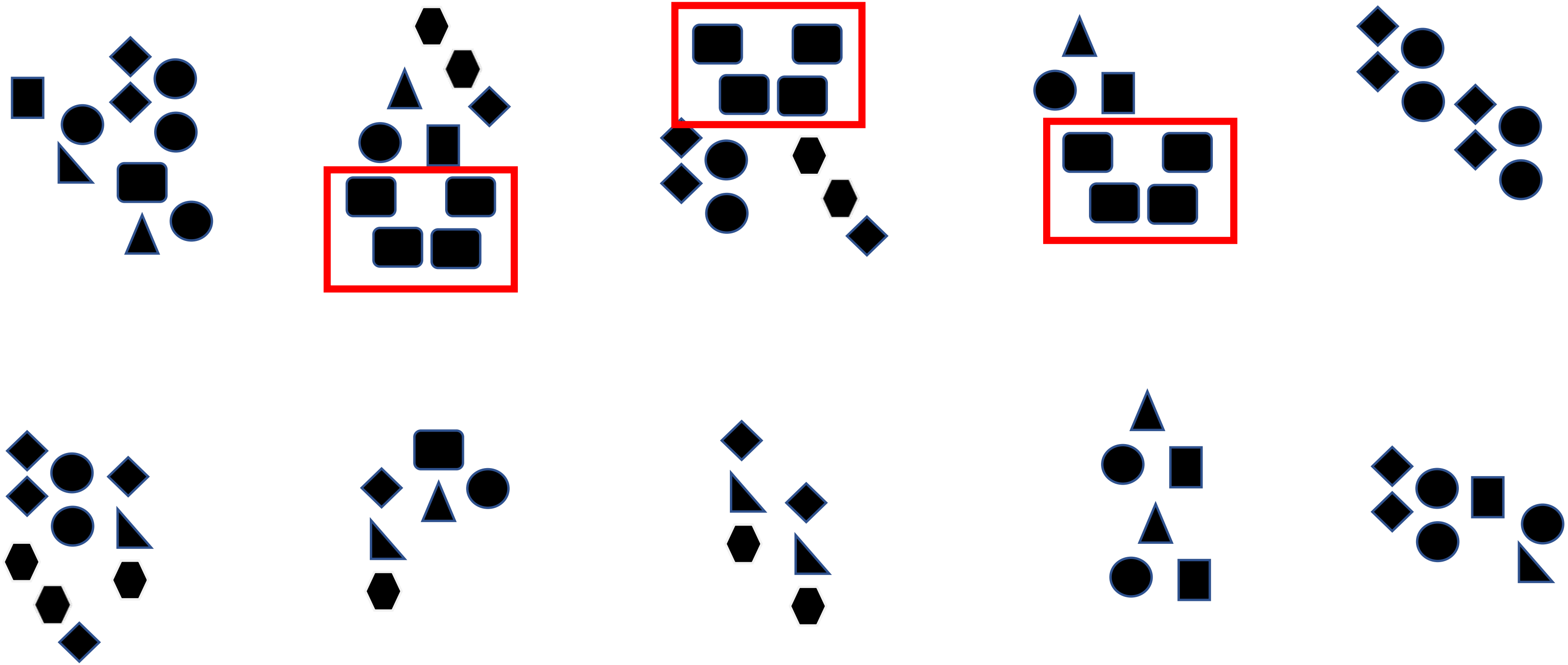
High-Level Motifs/Structure



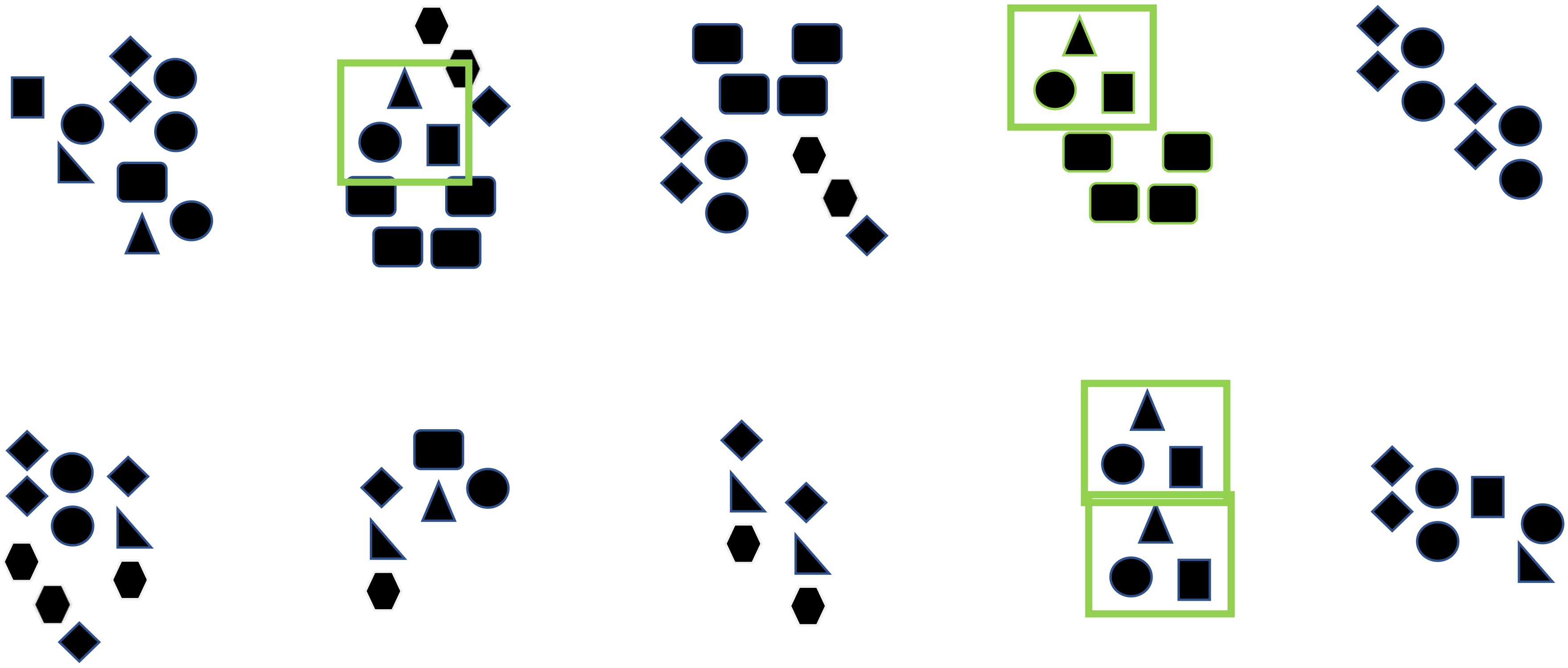
Shared Substructure Within Motifs



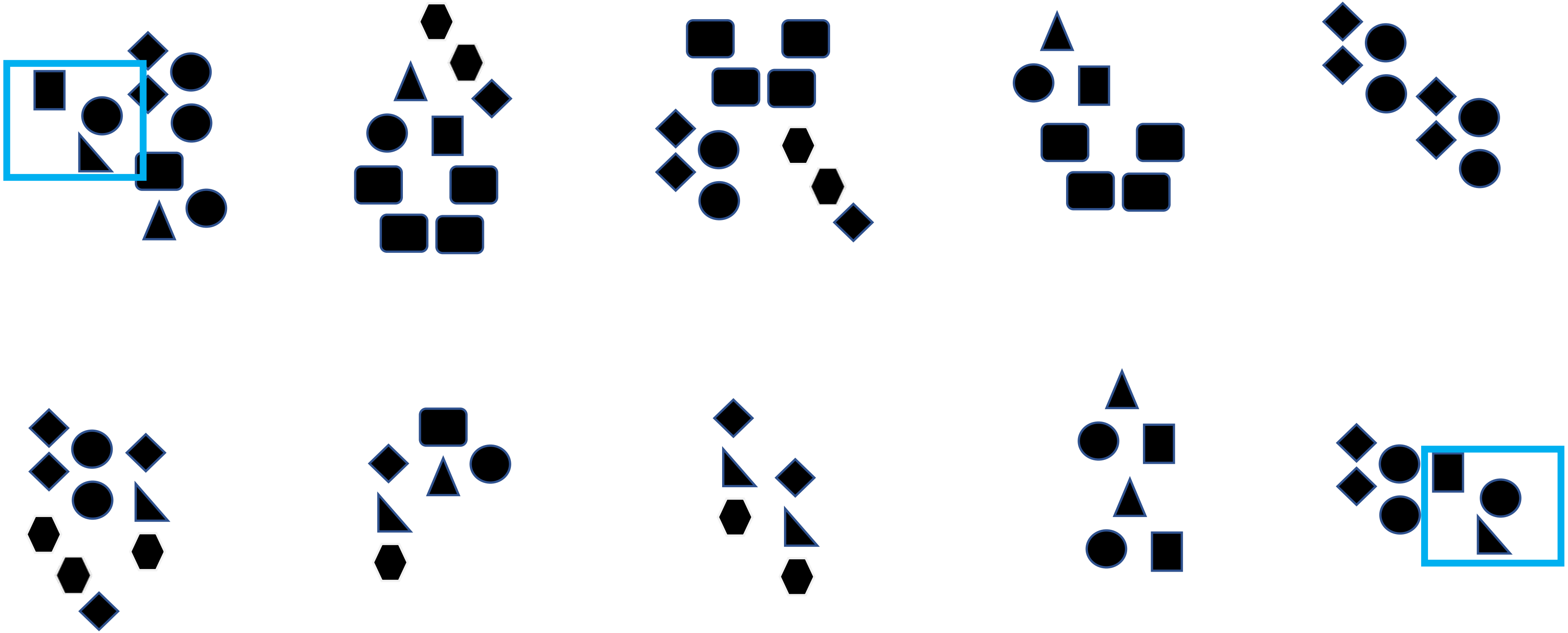
Shared Substructure Within Motifs



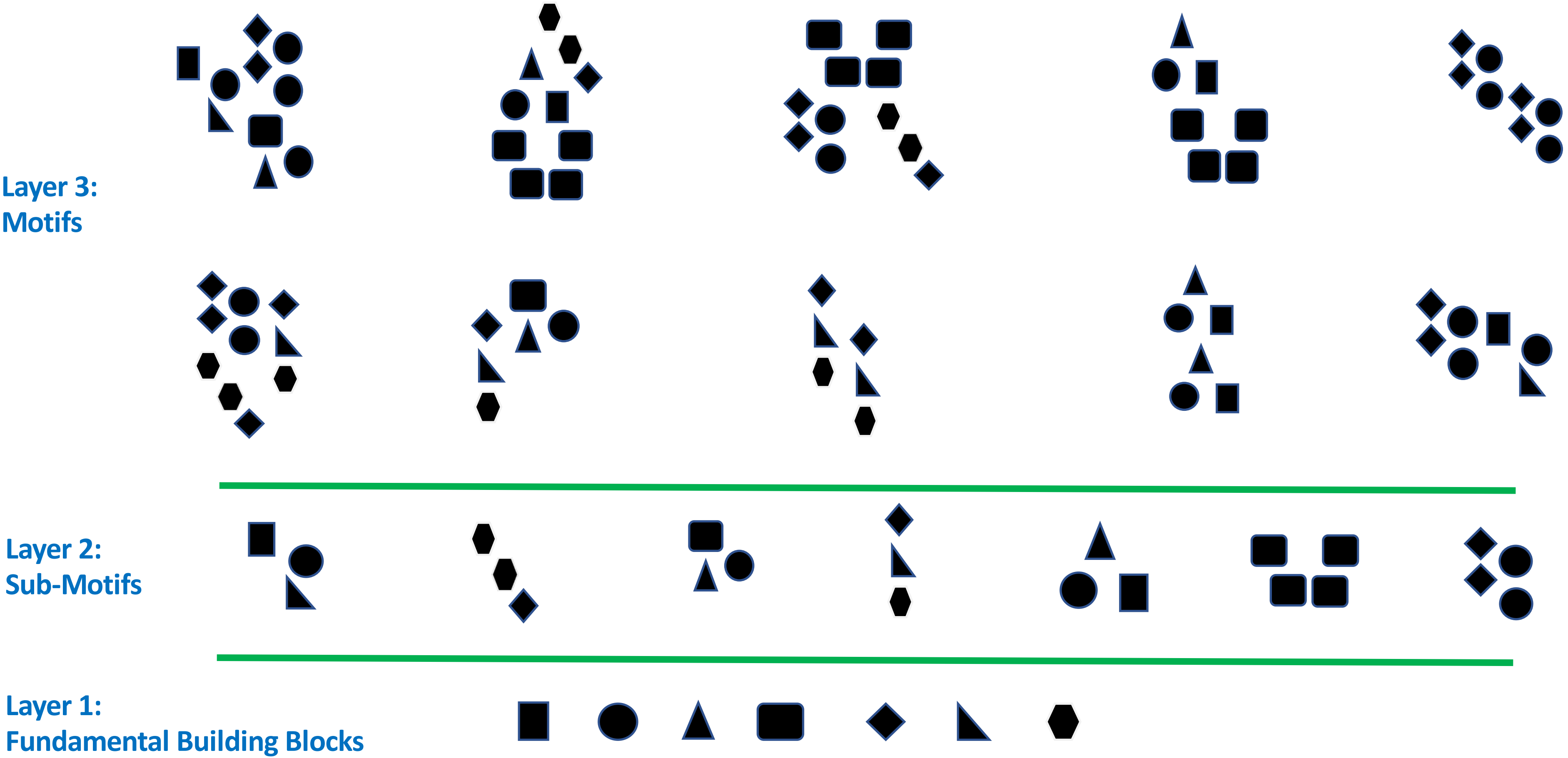
Shared Substructure Within Motifs



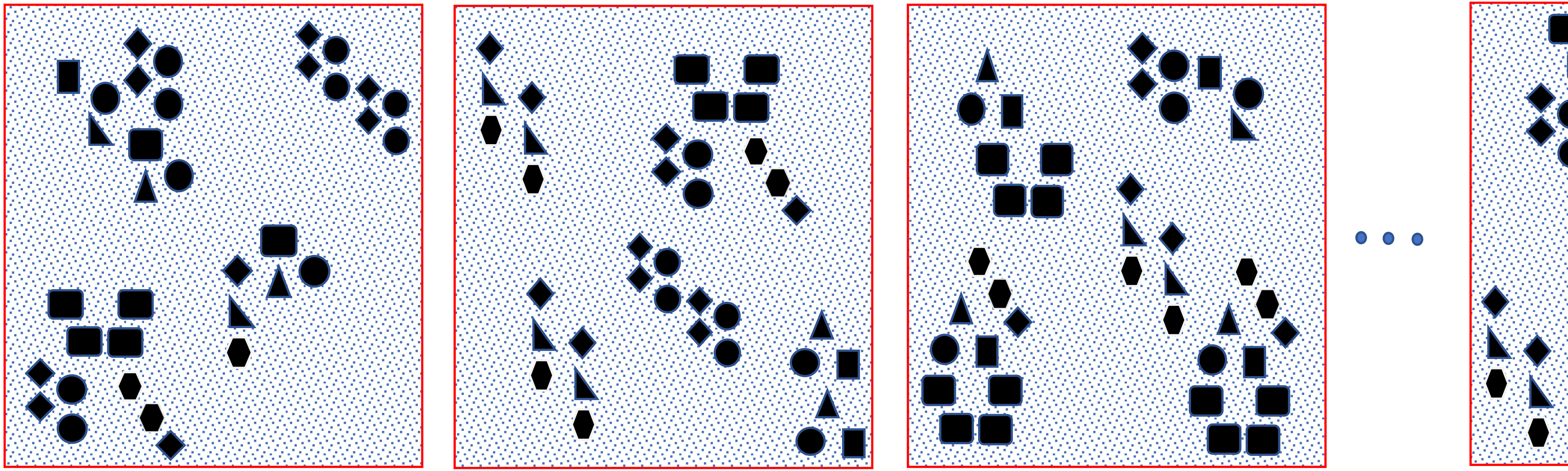
Shared Substructure Within Motifs



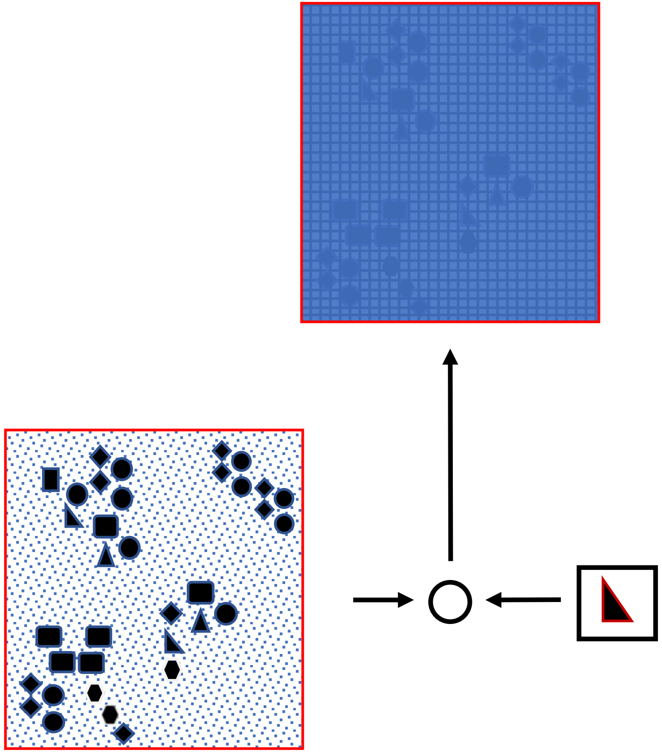
Hierarchical Representation of Images



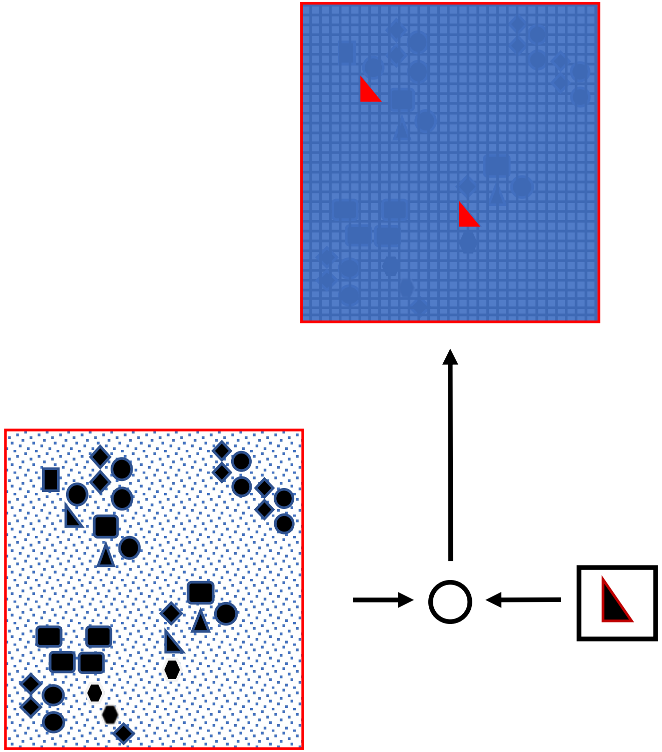
Recall the Data/Images



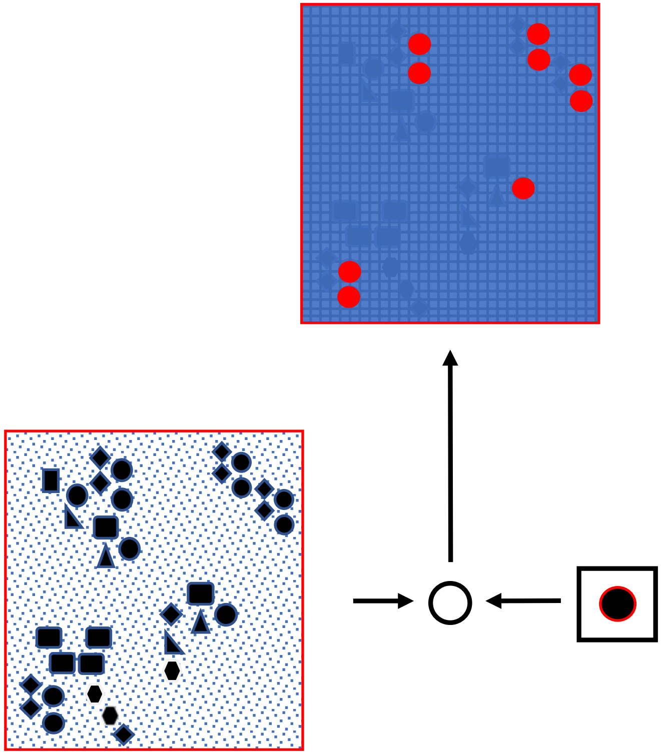
Convolutional Filter



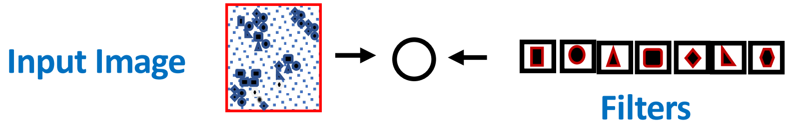
Convolutional Filter

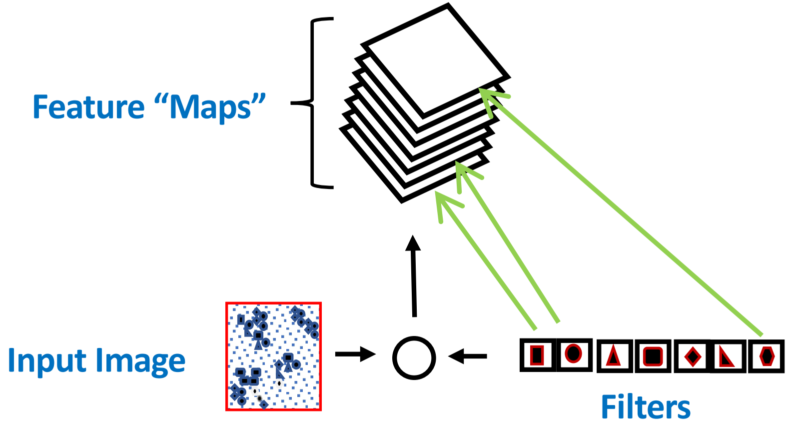


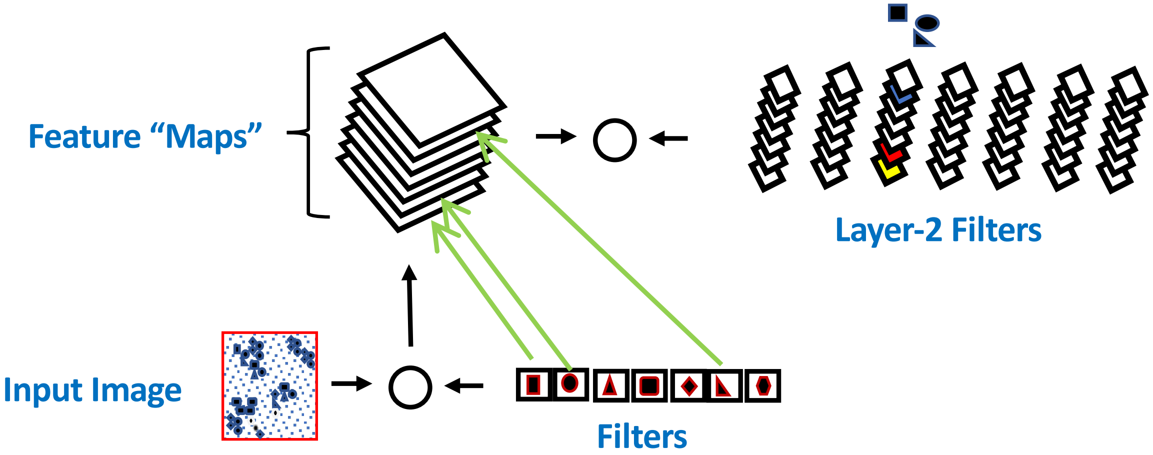
Convolutional Filter



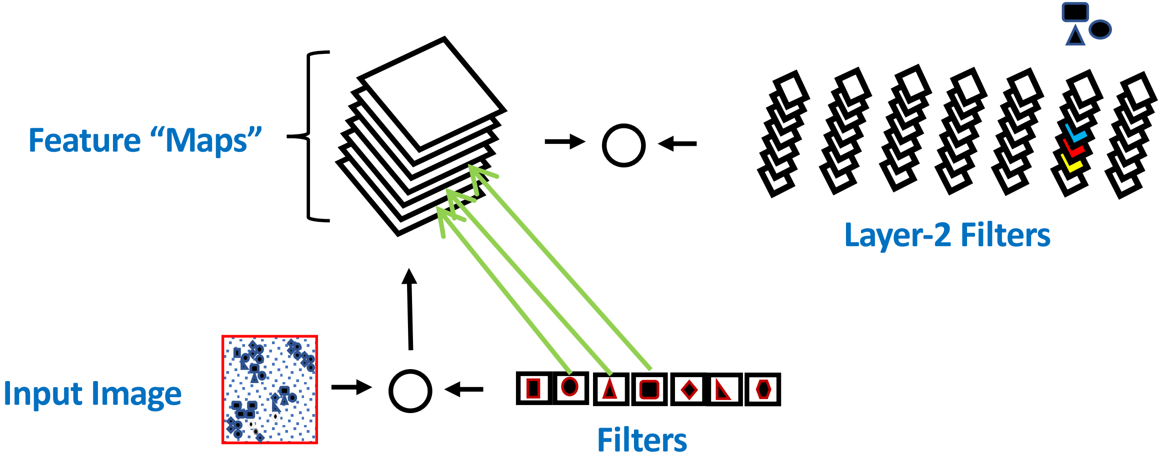
Multiple Filters, One for Each Building Block

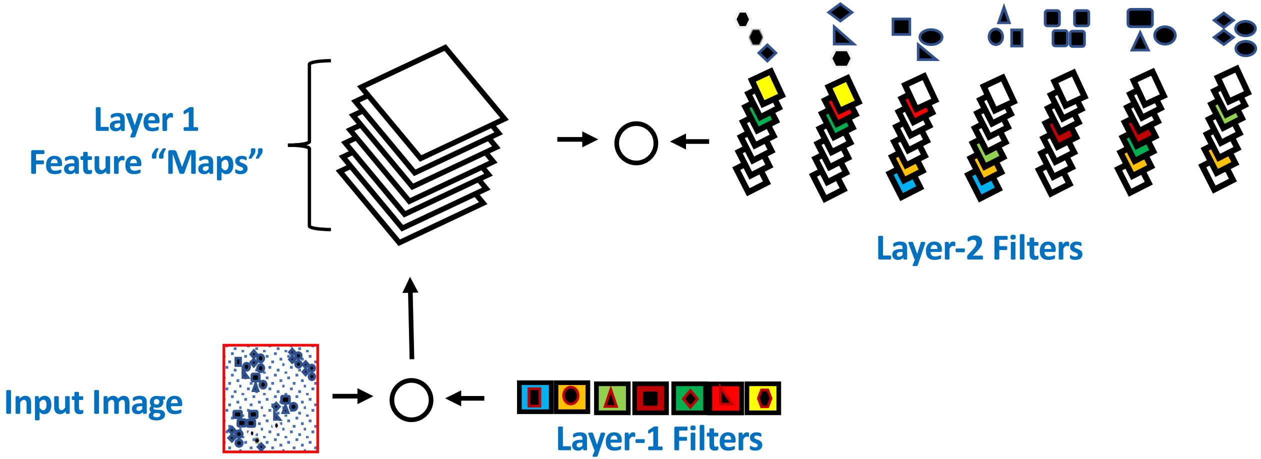


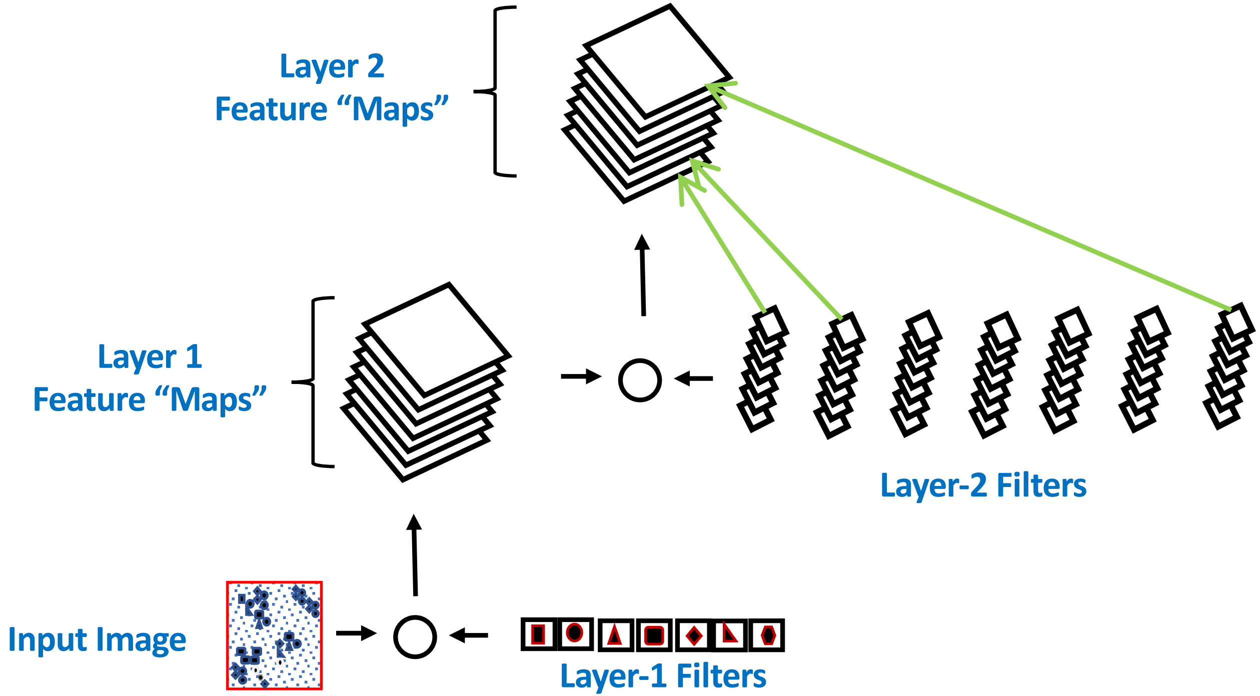

Feature "Maps"
Input Image
Filters



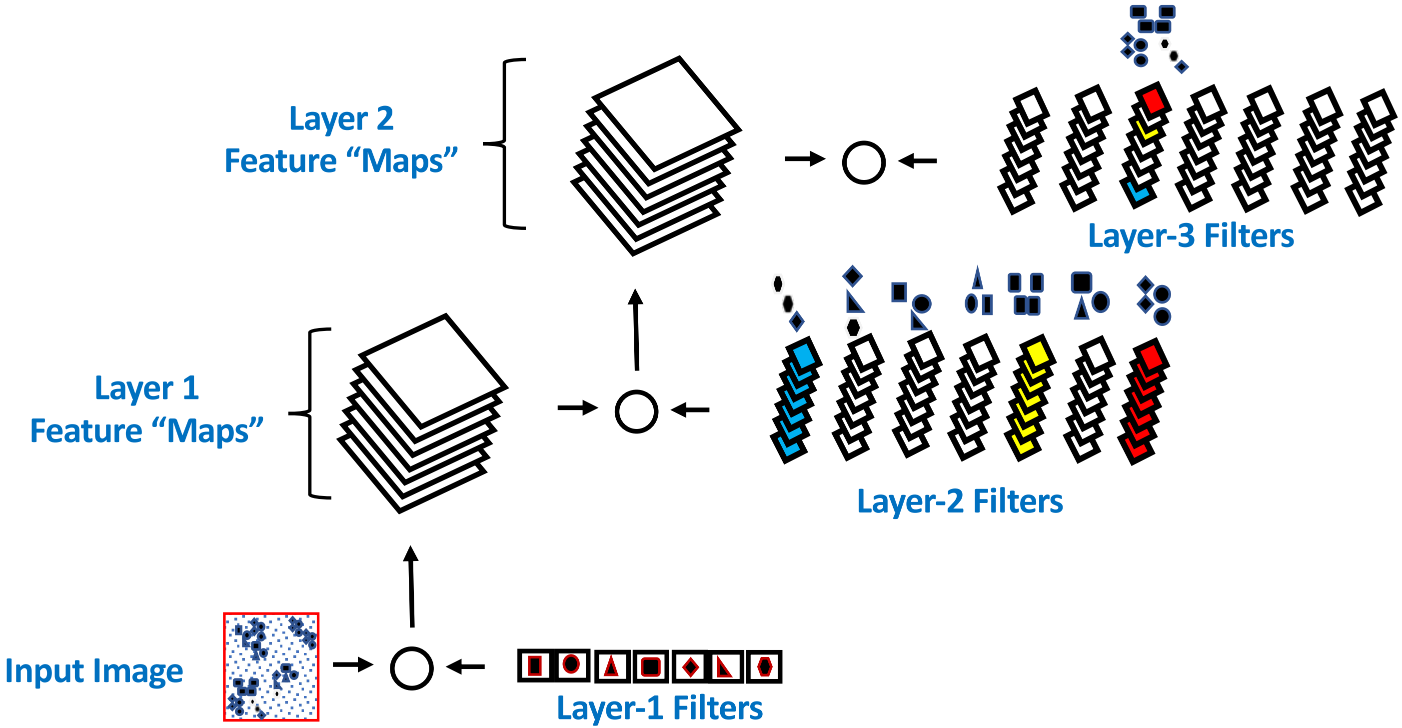
Feature “Maps”
Layer-2 Filters
Input Image
Filters

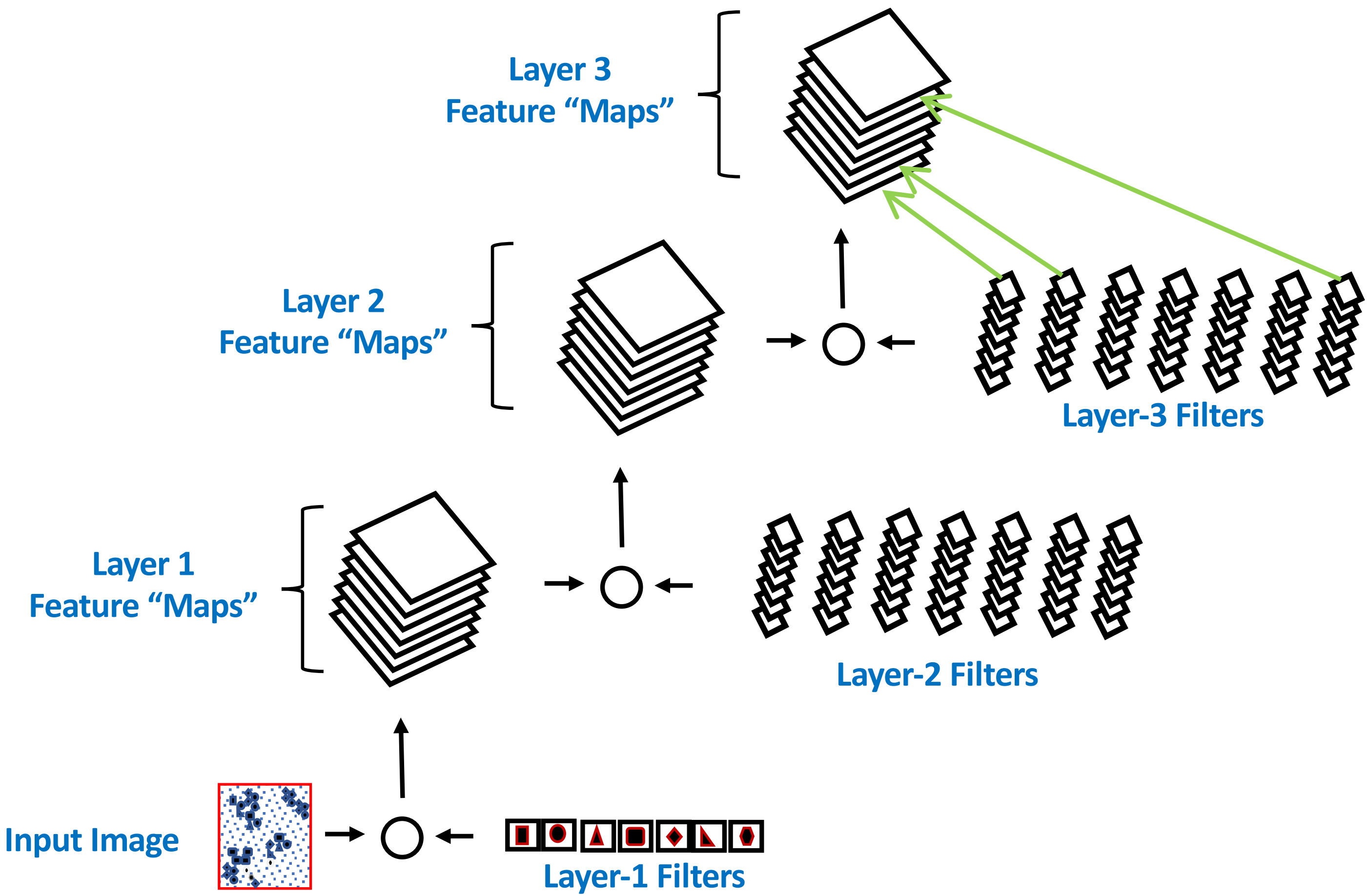

Feature "Maps"
Layer-2 Filters
Input Image
Filters


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

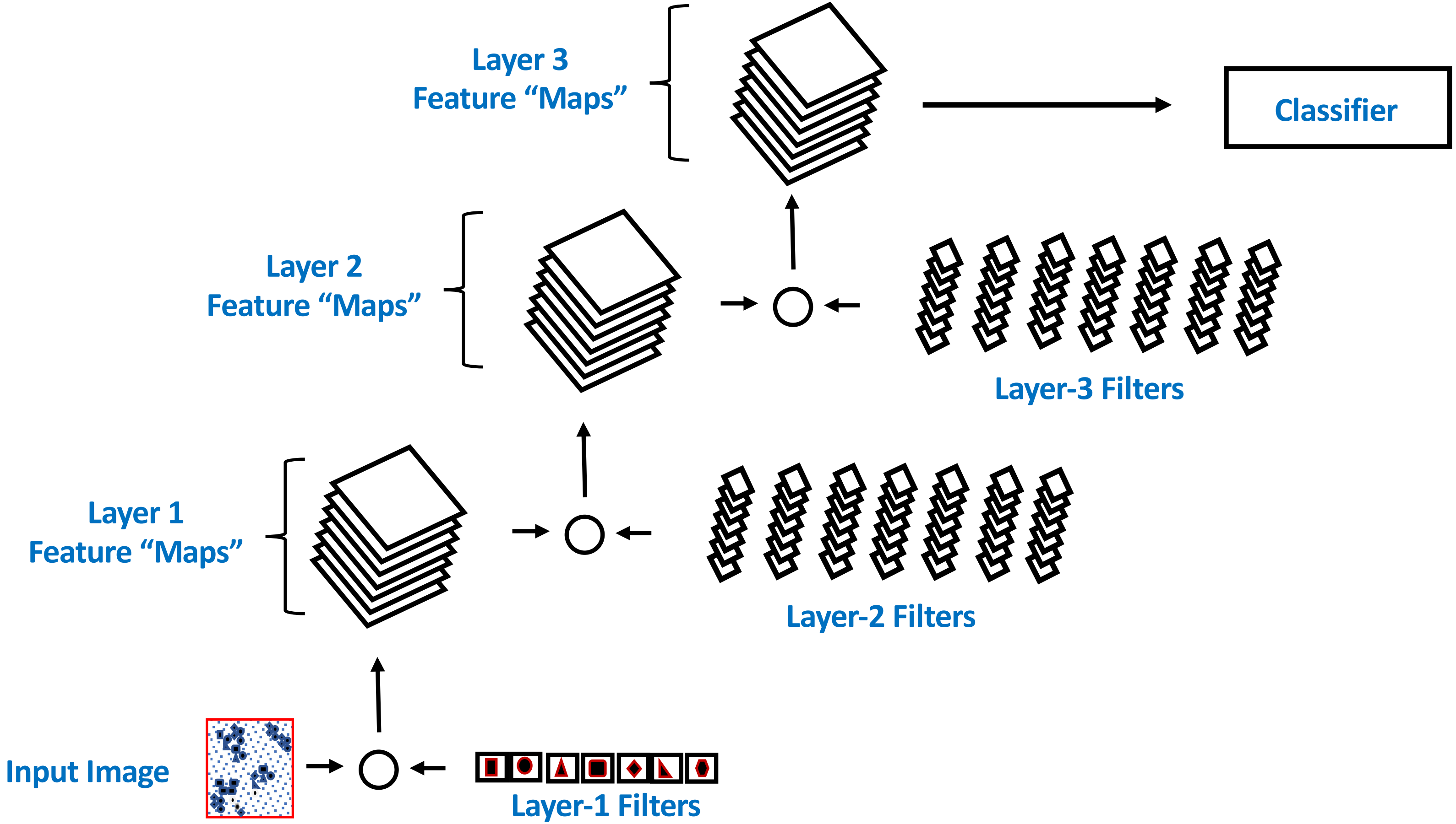


Layer 2
Feature “Maps”
Layer 1
Feature “Maps”
Layer-2 Filters
Input Image
Layer-1 Filters

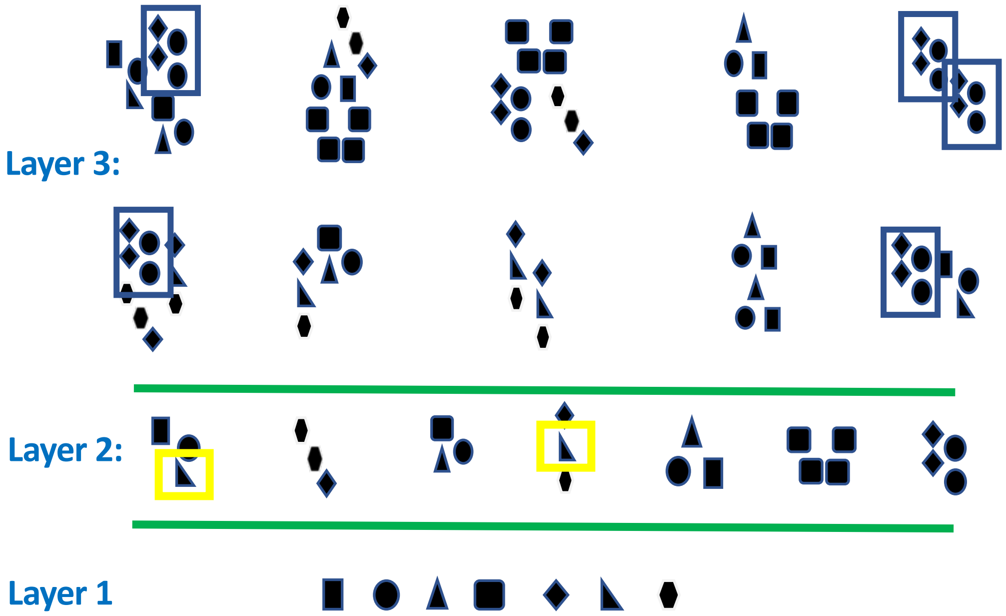

Layer 2
Feature "Maps"
Layer-3 Filters
Layer 1
Feature "Maps"
Layer-2 Filters
Input Image
Layer-1 Filters


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

Deep CNN Architecture



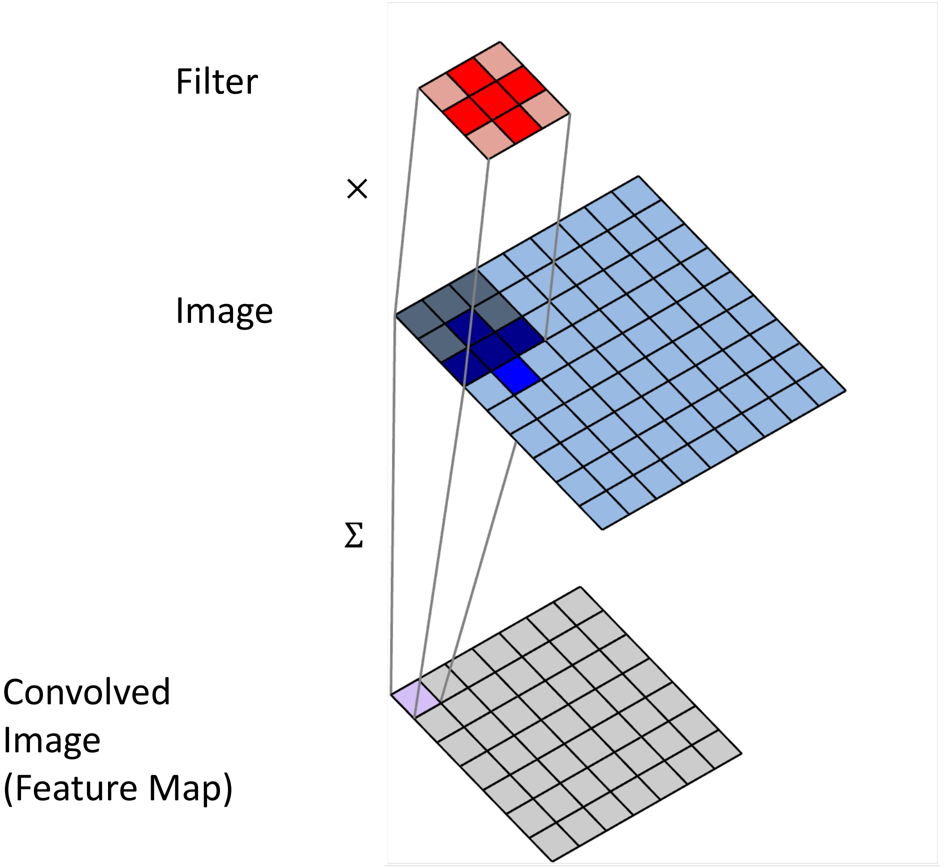
Advantage of Hierarchical Features?



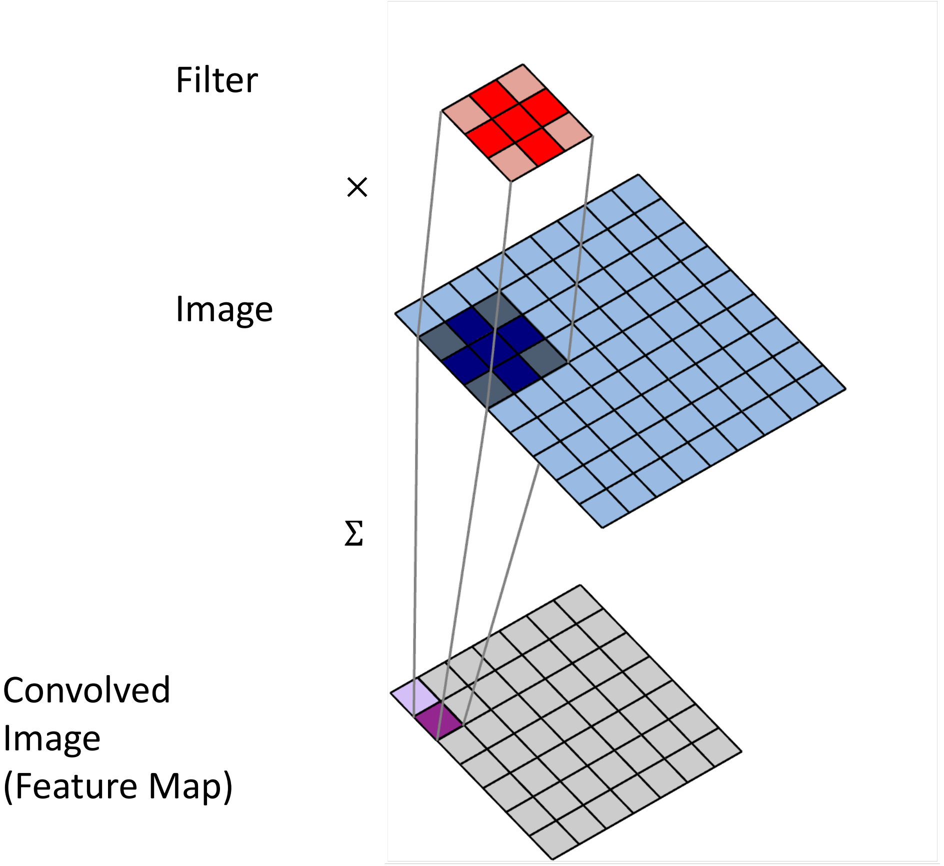
- By learning and sharing statistical similarities within high-level motifs, we better leverage all training data

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

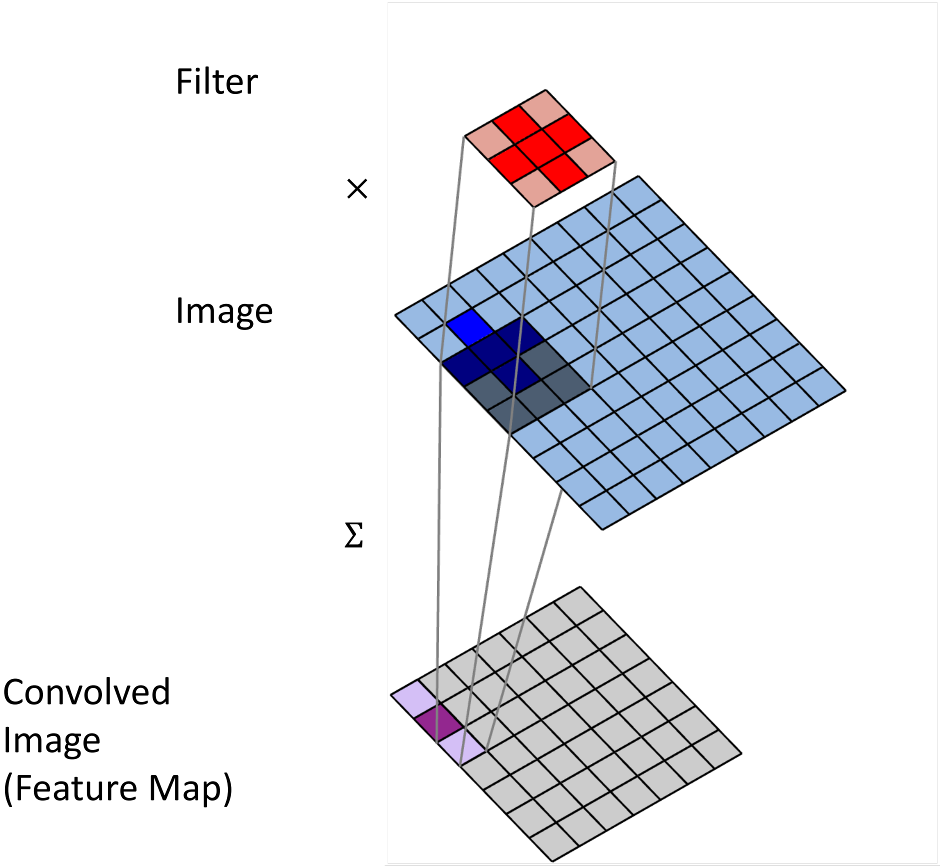
2D Spatial Convolution



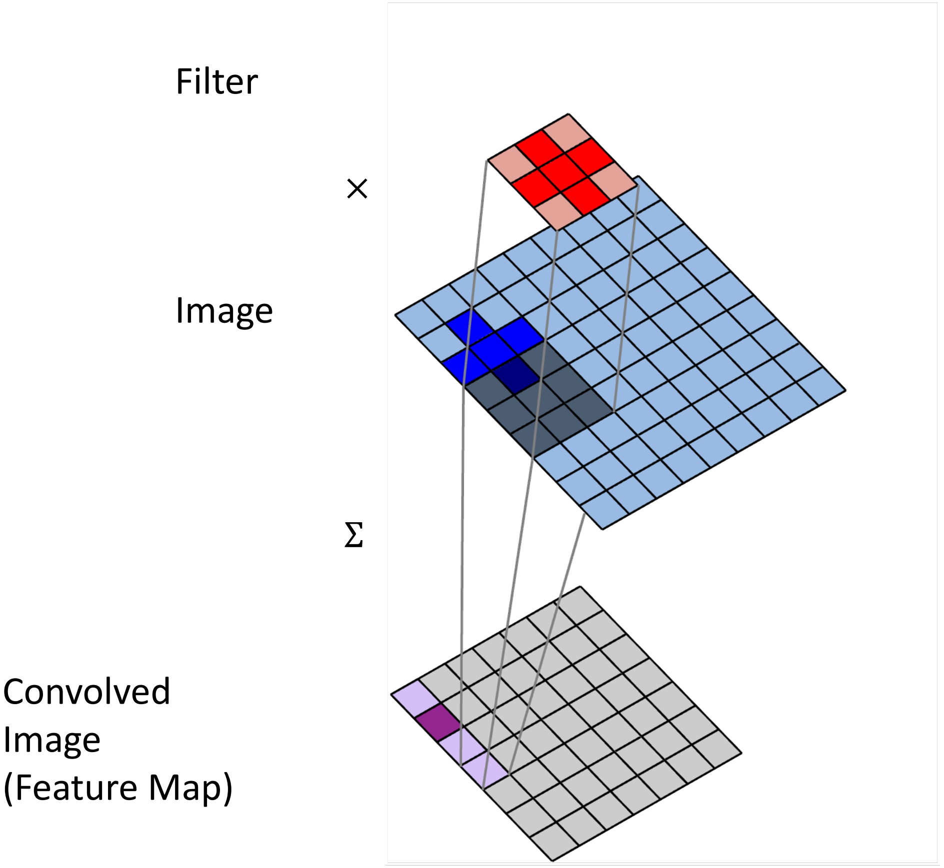
2D Spatial Convolution



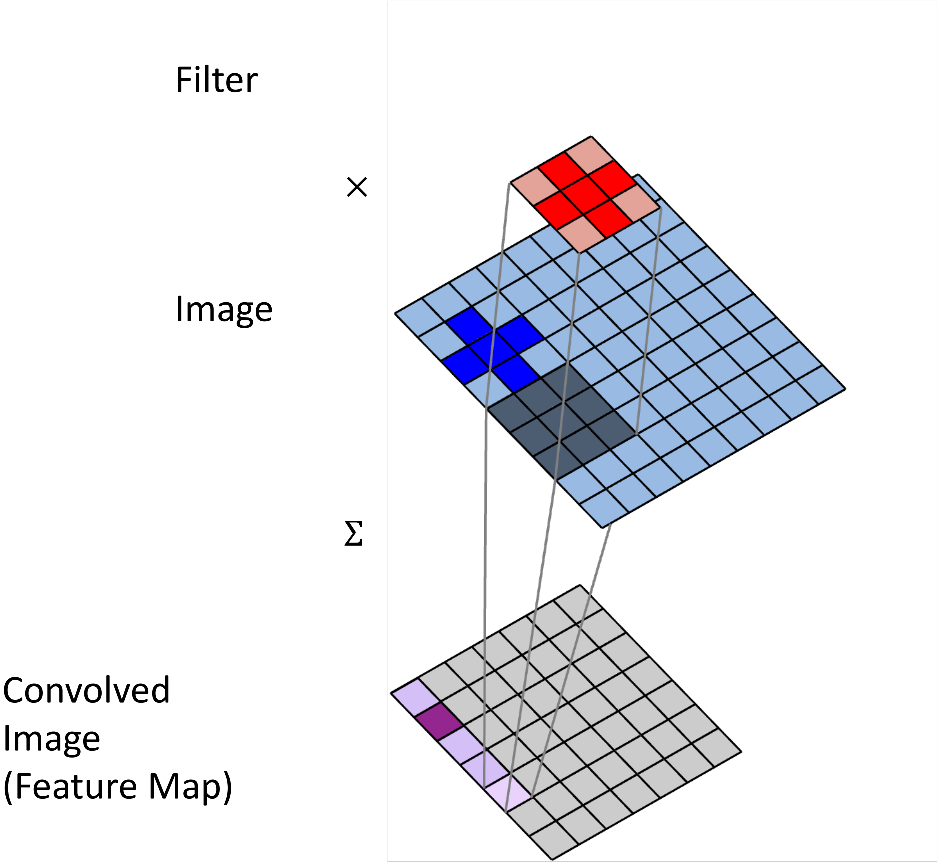
2D Spatial Convolution



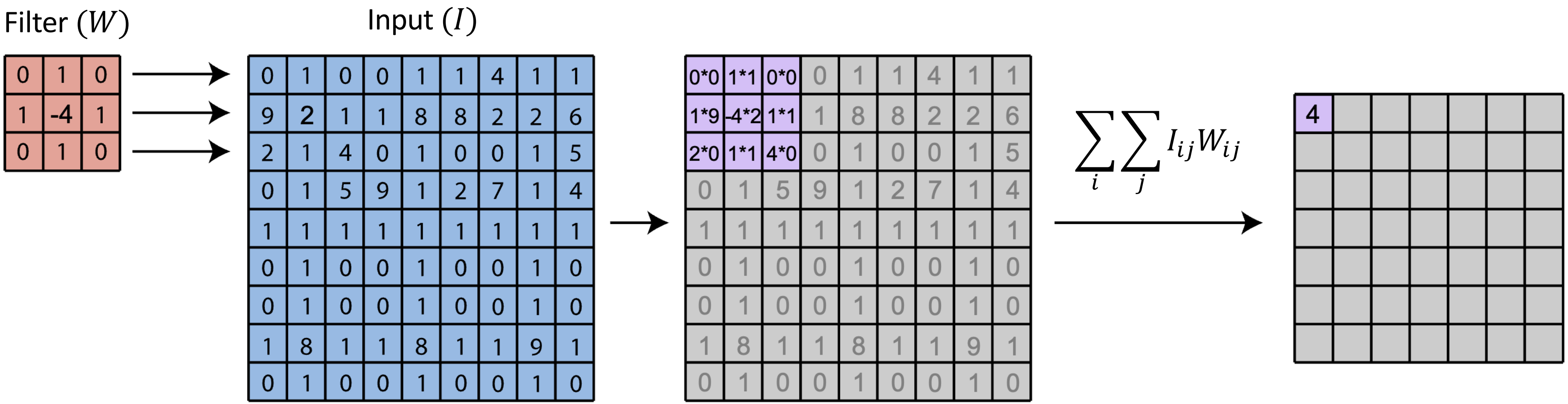
2D Spatial Convolution



2D Spatial Convolution



2D Spatial Convolution



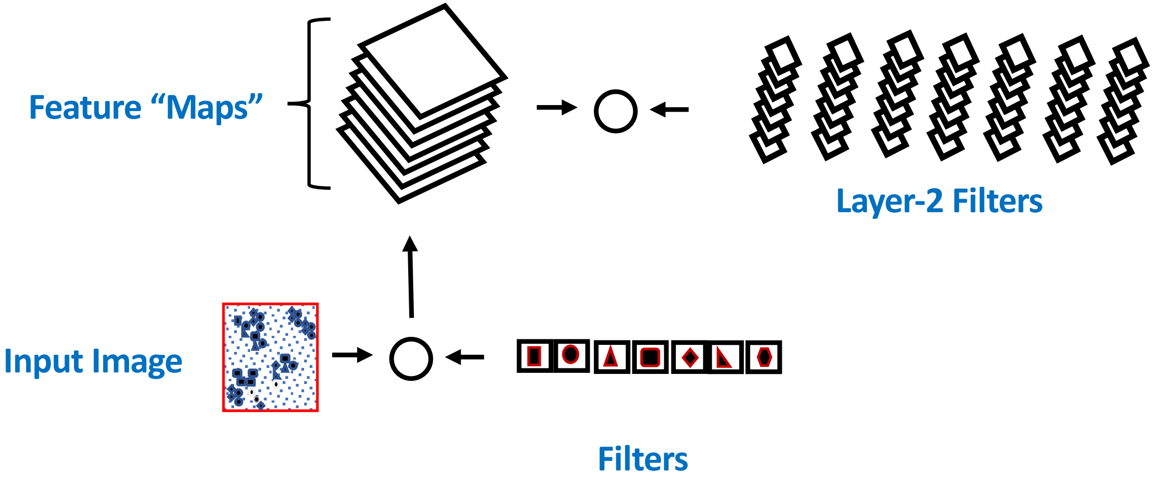
2D Spatial Convolution

Filter (W)

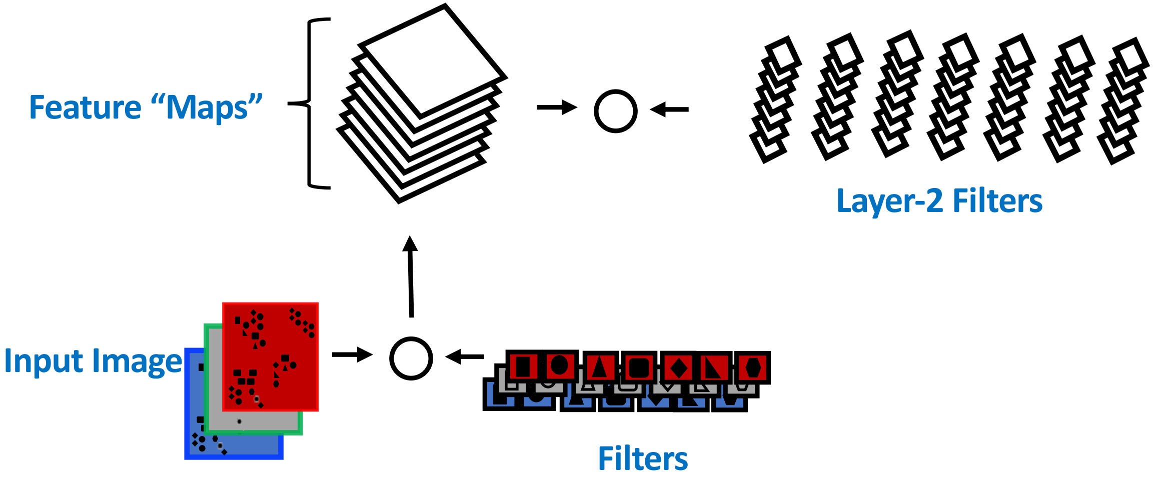
0	1	0
1	-4	1
0	1	0

Input (I)

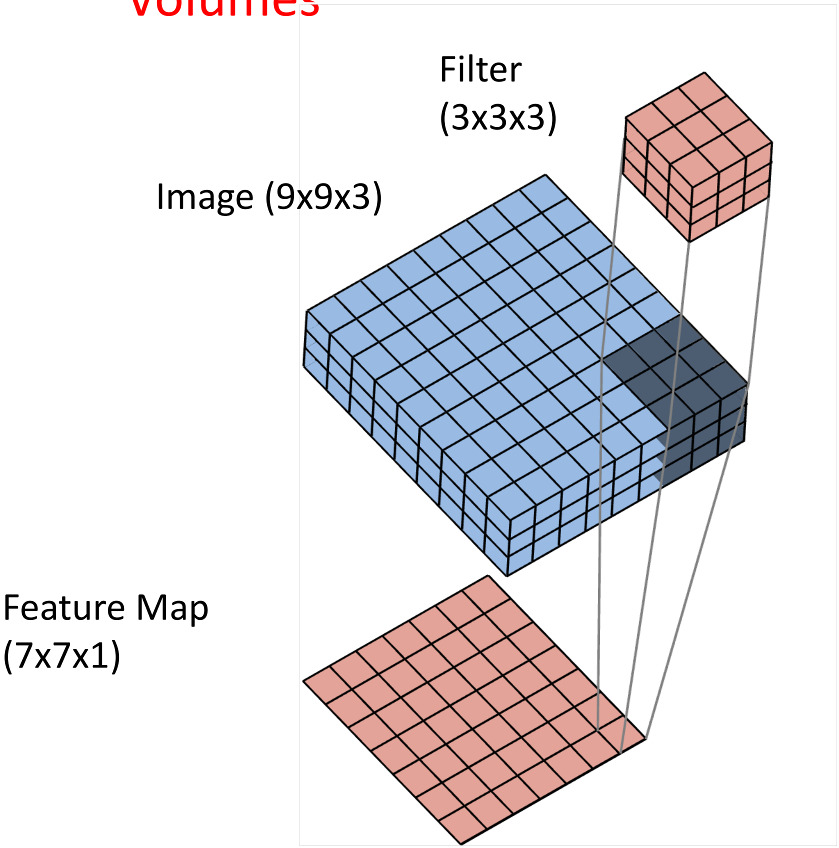
Filters Operate Over Input Volumes



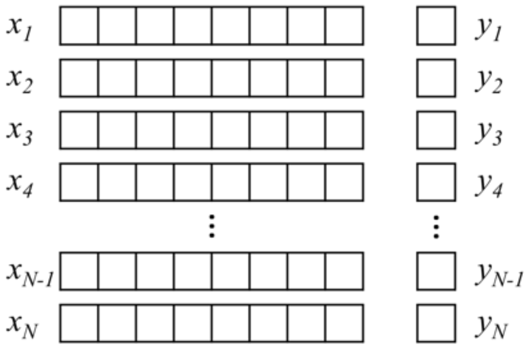
Filters Operate Over Input Volumes



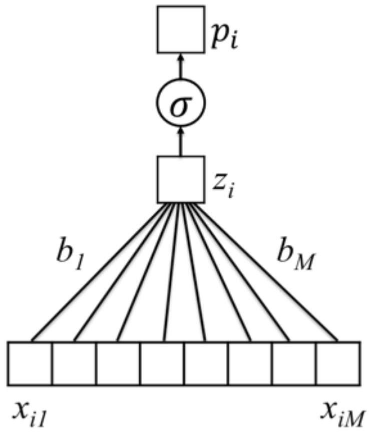
Filters Operate Over Input Volumes



Given Labeled Training Images, How do we Learn the Parameters of the CNN?

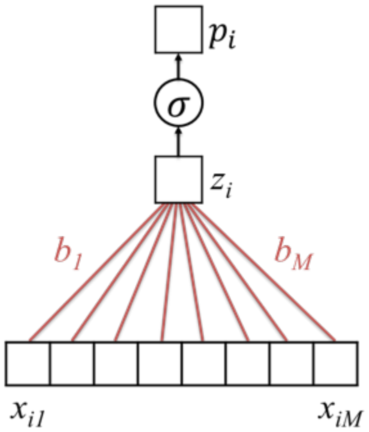


Training Set



$$p_i = \sigma(b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM})$$

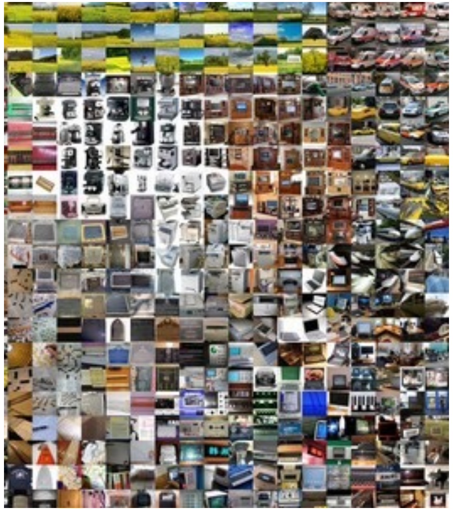
Untrained Logistic Regression Model (or "Network")

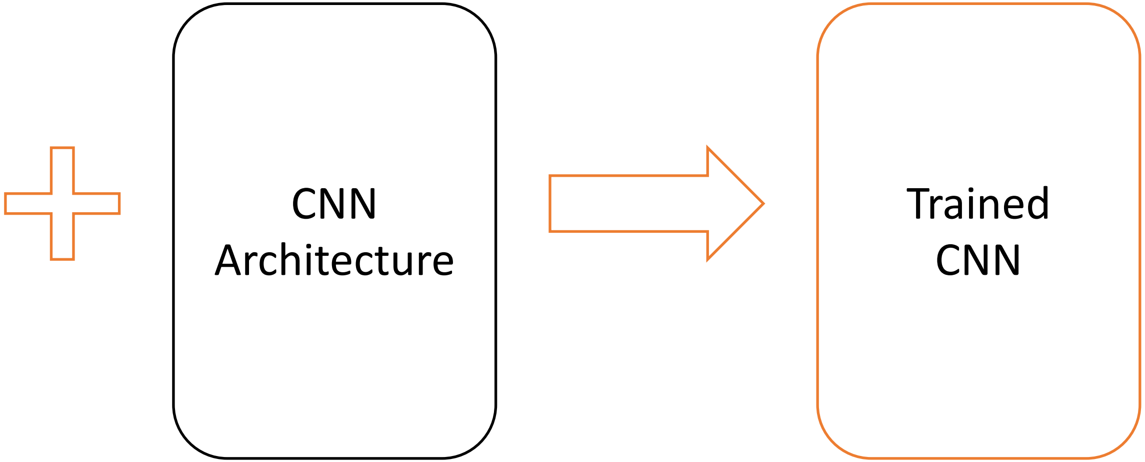


$$b = (b_0, \ldots b_M)$$

Trained Model (with learned parameters)

Given Labeled Training Images, How do we Learn the Parameters of the CNN?





Architecture (specified) vs Parameters (learned)

Architecture:

- Number of layers
- Layer types (e.g. convolutional, pooling, fully connected)
- Number of filters in each layer
- Shape and size of filters

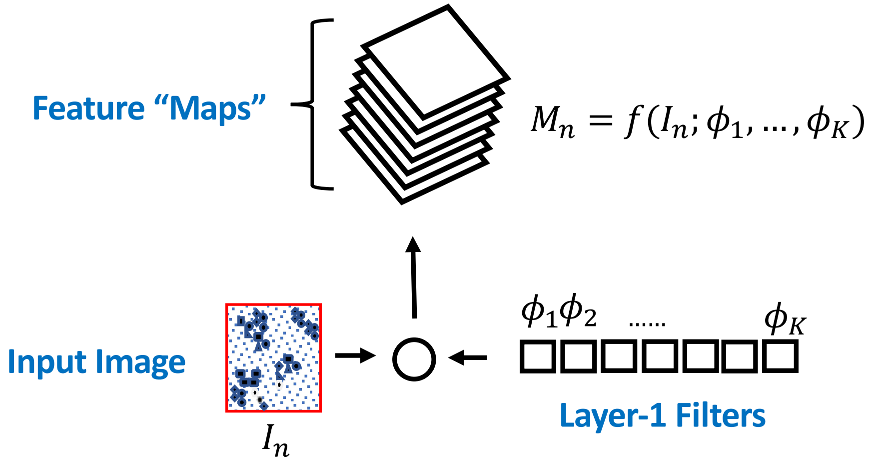
Use 3x3 filters
In layer 1

Parameters:

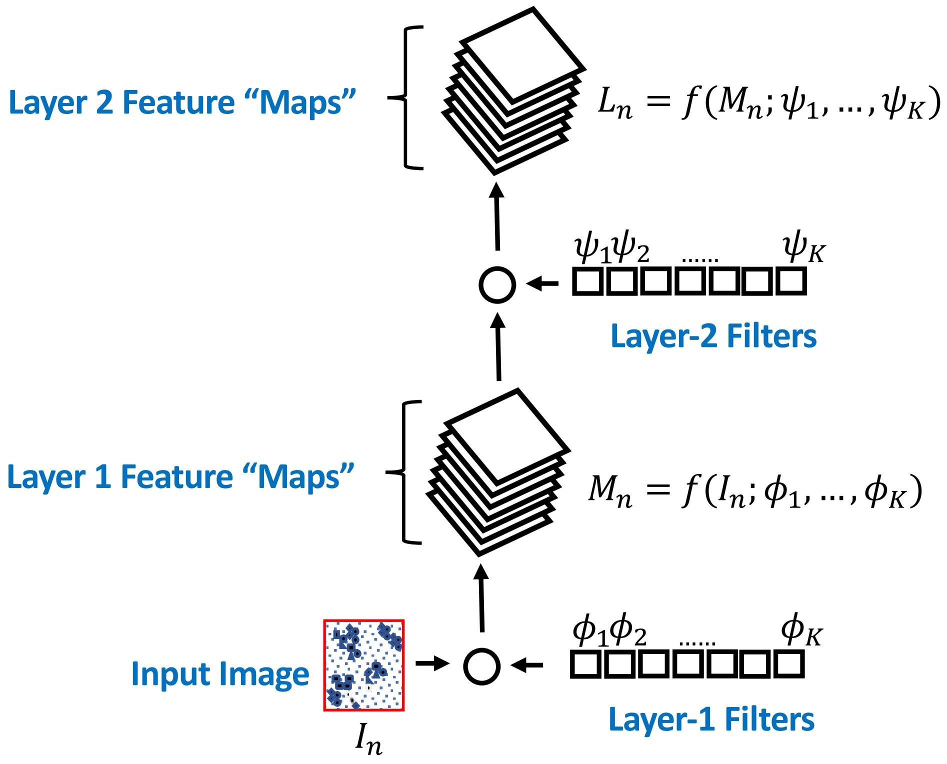
- Individual Elements of each filter
- Parameters of other layers

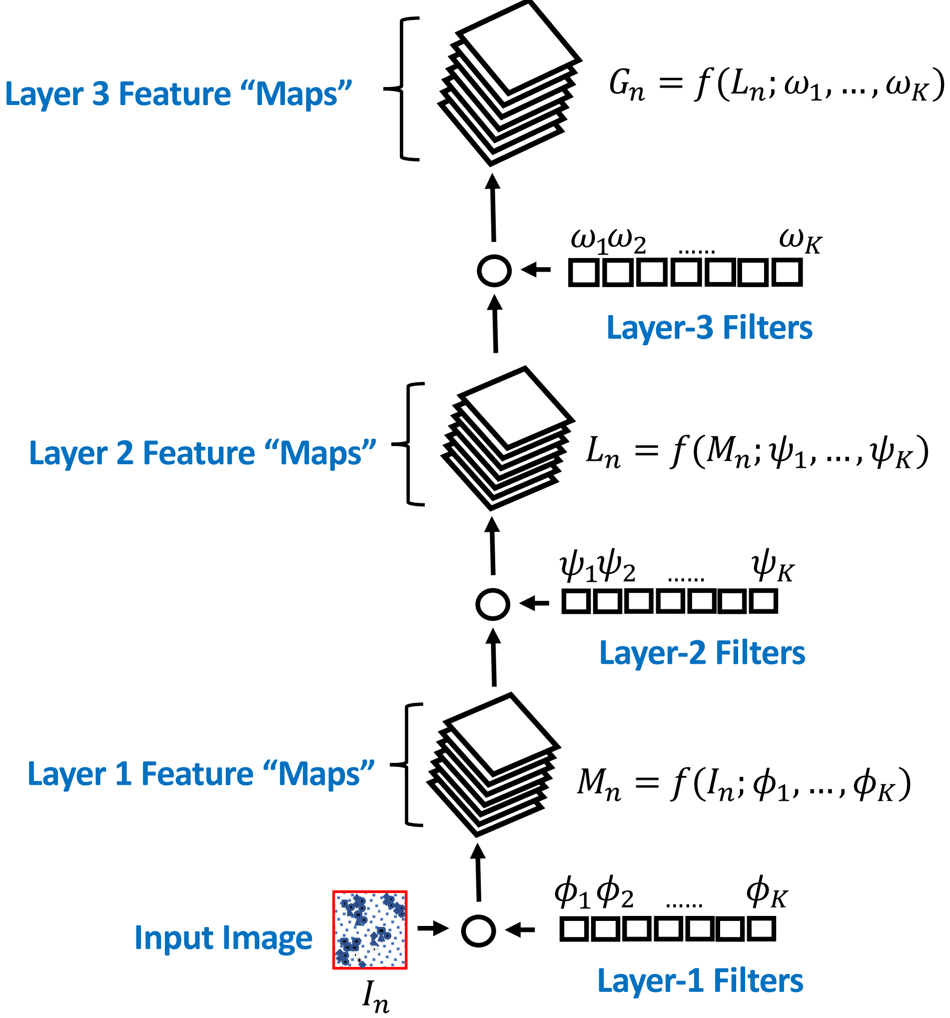
Learn values of
Each layer 1 filter

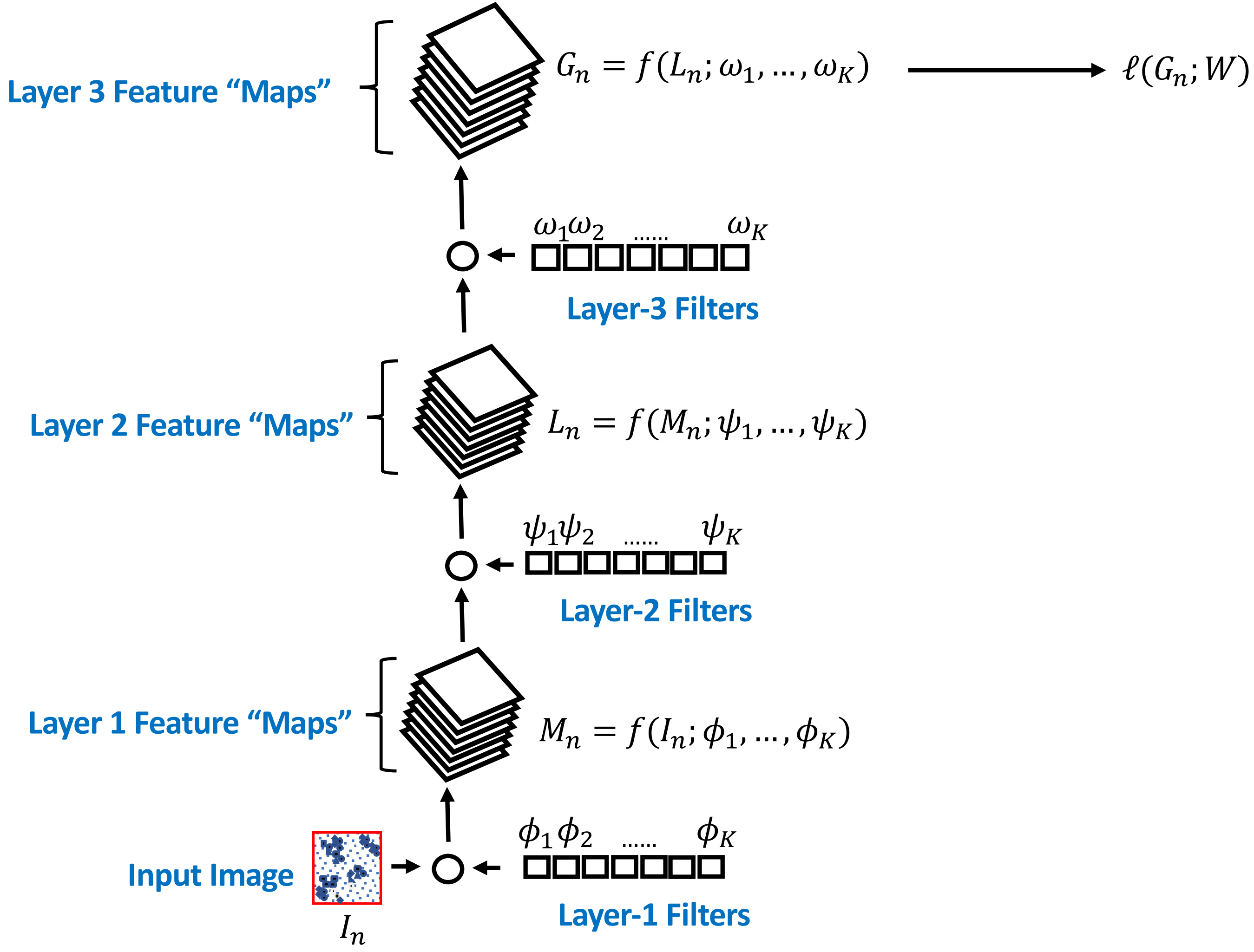
-1	1	-1
1	-1	1
-1	1	-1



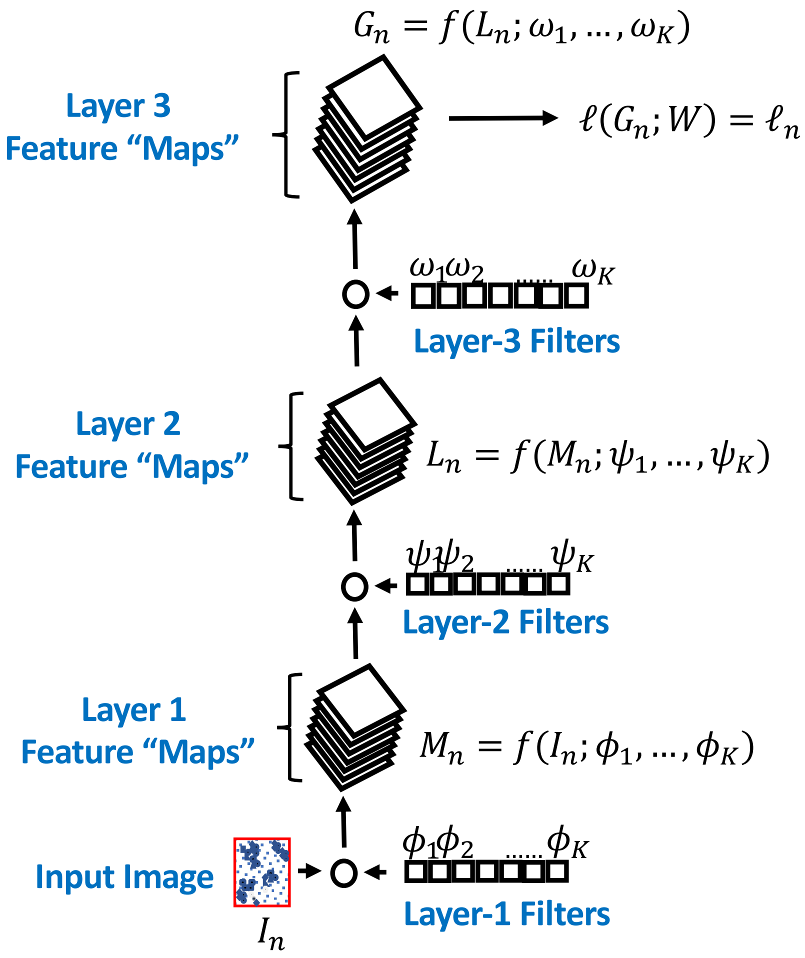
Feature “Maps”
$M_n = f(I_n; \phi_1, \ldots, \phi_K)$
Input Image
I_n
$\phi_1 \phi_2$ ...... ϕ_K
Layer-1 Filters


Layer 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”
$G_n = f(L_n; \omega_1, \dots, \omega_K)$
$\omega_1 \omega_2$ ...... 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
- Average loss, which depends on model parameters:

$$1/N \sum_{n=1}^{N} loss(y_n, \ell_n)$$

- Find specific parameters that minimize the average loss

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