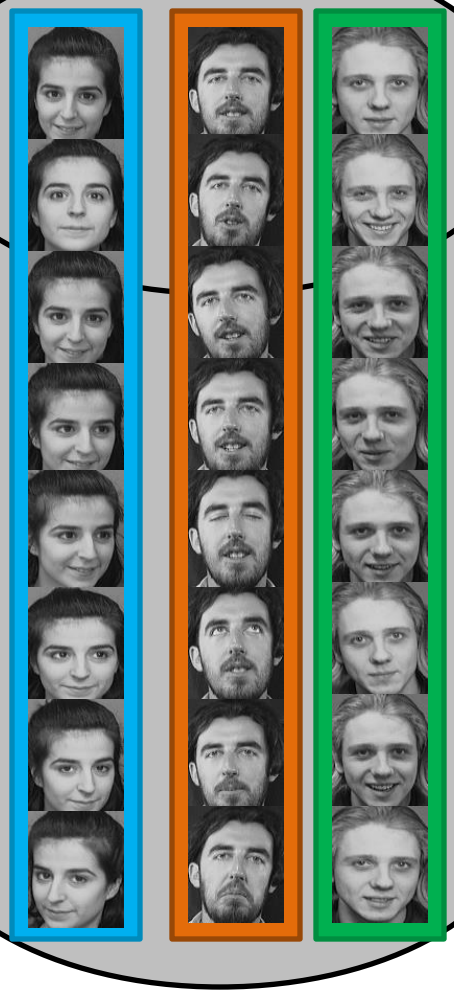
Face Recognition: A Convolutional Neural Network Approach

Instructor: Bhiksha Raj
Student: T. Hoang Ngan Le

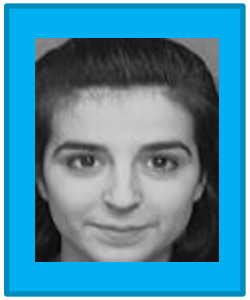
The Problem

Testing

Training



Recognition



Proposed System - Flowchart

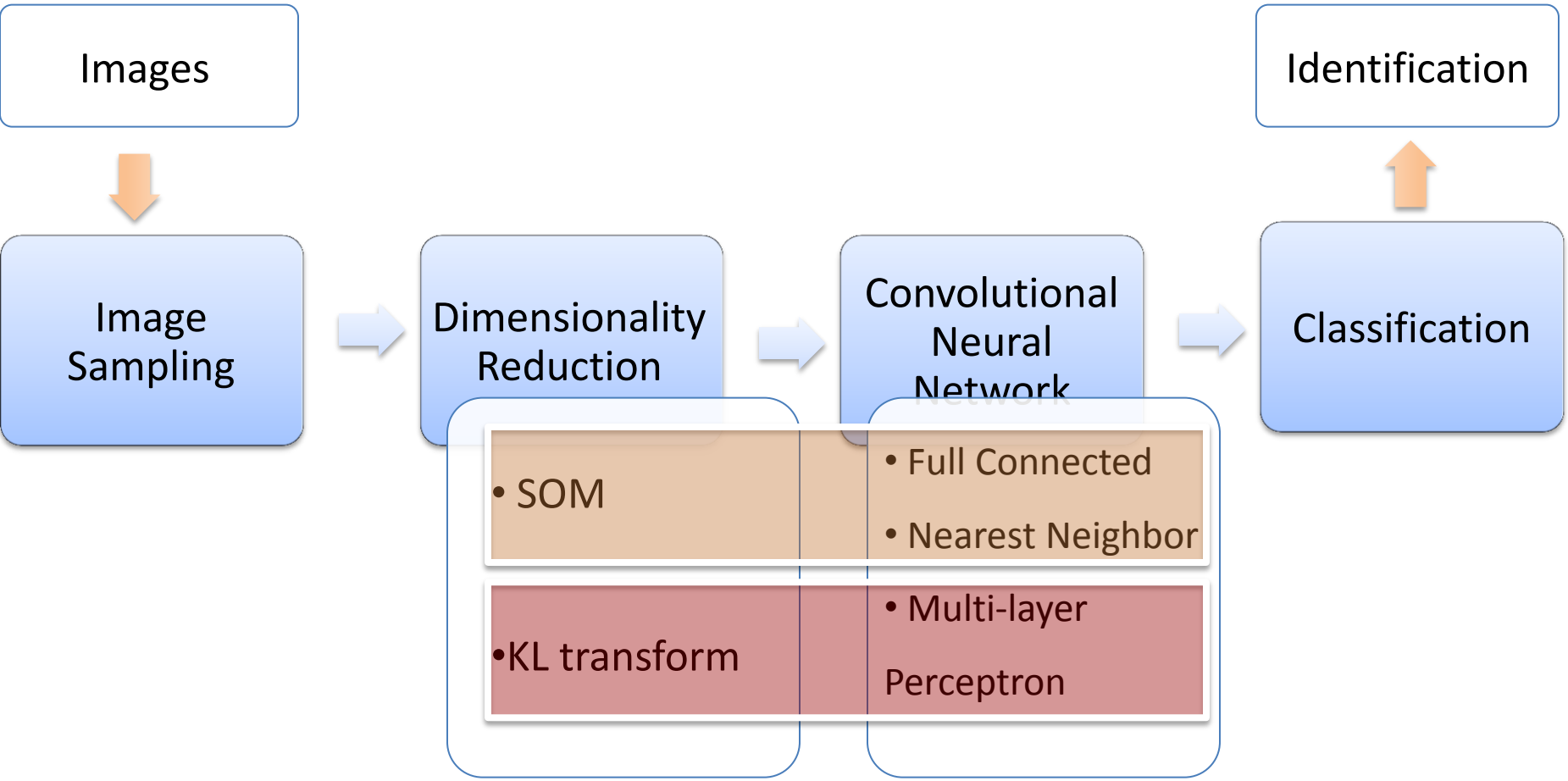
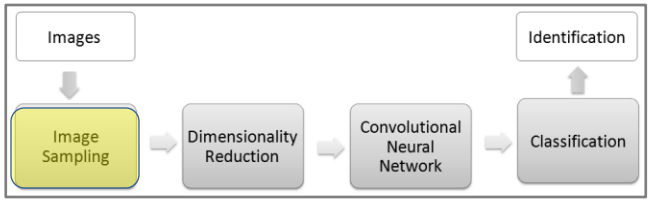
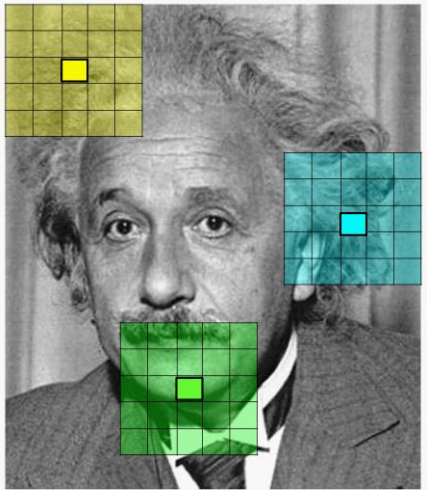
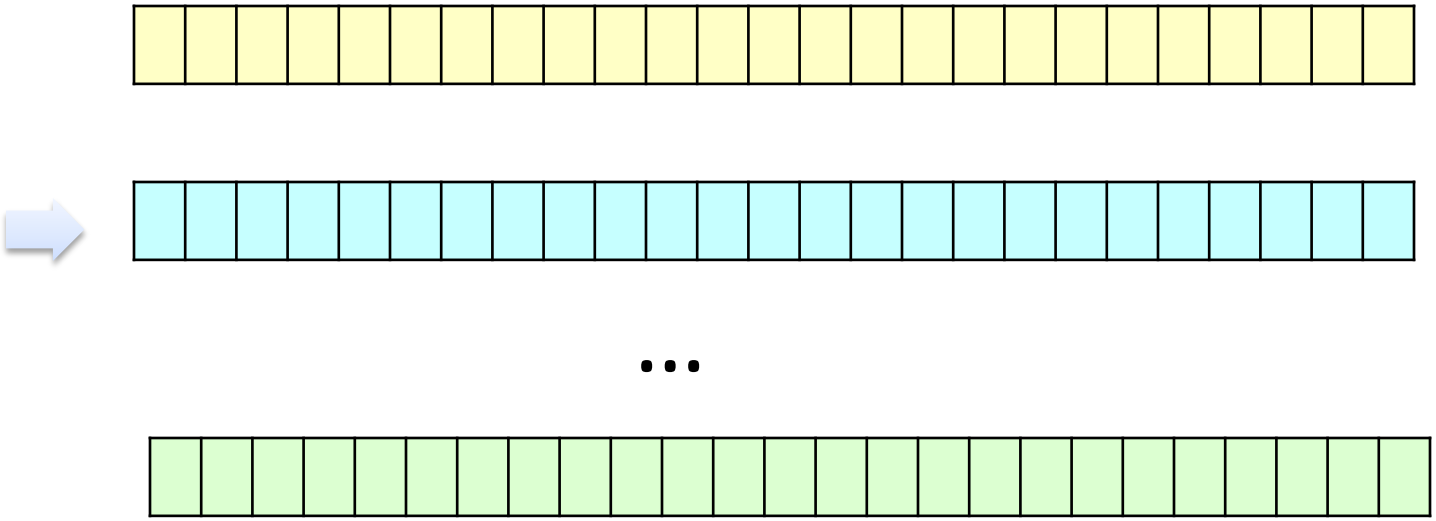


Image Sampling





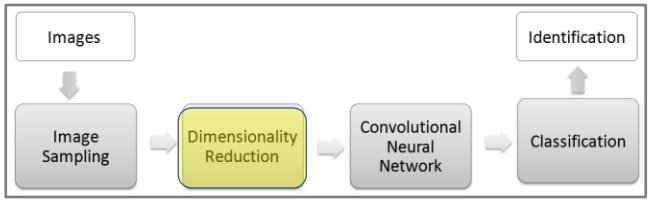
5×5

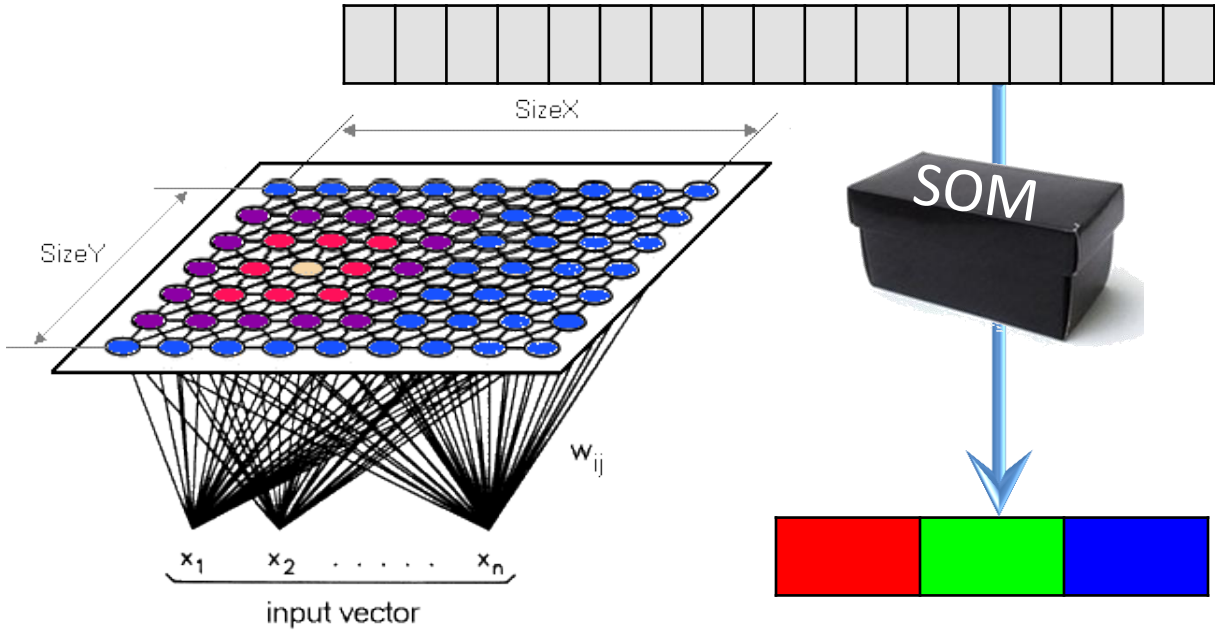


R^{25}

A window is stepped over the image and a vector is created at each location.

Dimensionality Reduction - SOM



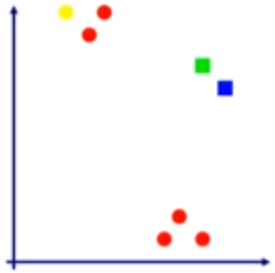


R^{25}

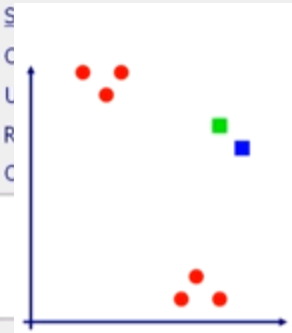
R^{3}

Dimensionality Reduction - SOM

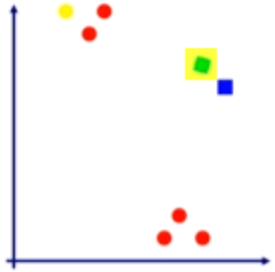
1



Operations



2



Operations

Select random input

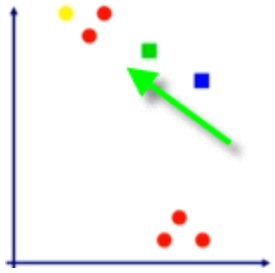
Compute winner neuron

Update neurons

Repeat for all input data

Classify input data

3



Operations

Select random input

Compute winner neuron

Update neurons

Repeat for all input data

Classify input data

4

Operations

Select random input

Compute winner neuron

Update neurons

Repeat for all input data

Classify input data

Operations

Select random input

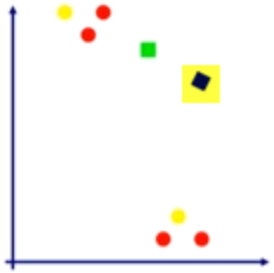
Compute winner neuron

Update neurons

Repeat for all input data

Classify input data

5



Operations

Select random input

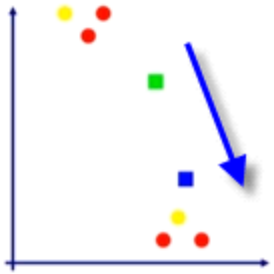
Compute winner neuron

Update neurons

Repeat for all input data

Classify input data

6



Operations

Select random input

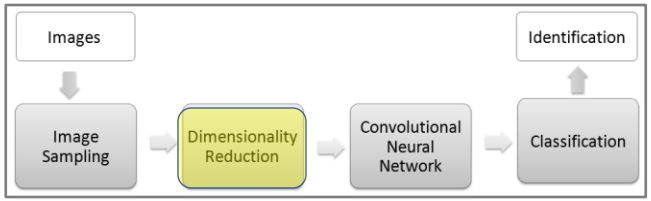
Compute winner neuron

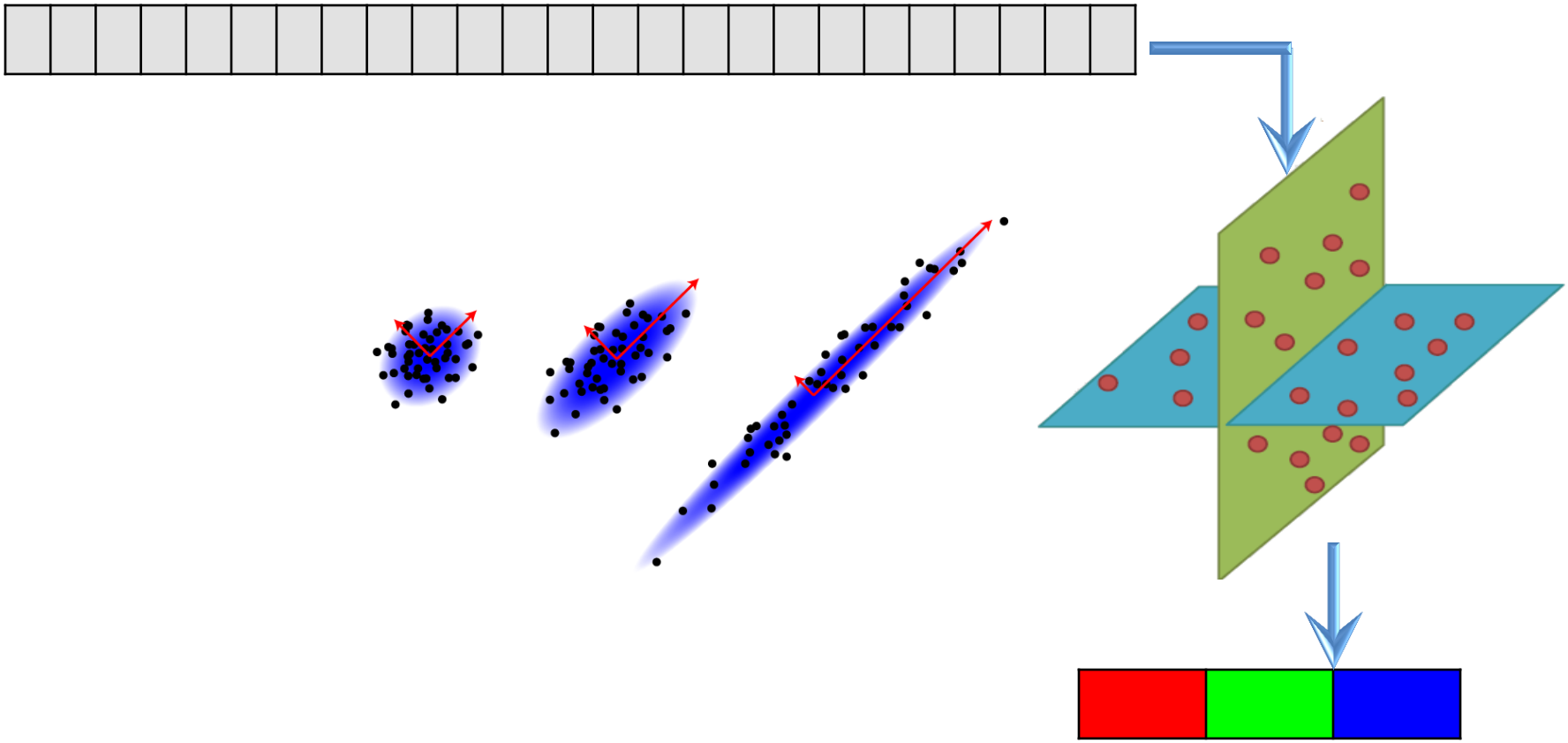
Update neurons

Repeat for all input data

Classify input data

Dimensionality Reduction - KL Transform





Dimensionality Reduction - KL Transform

- PCA
  - Objective function: $\mathbf{u}^\top \mathcal{C}\mathbf{u} - \lambda(\mathbf{u}^\top \mathbf{u} - 1)$

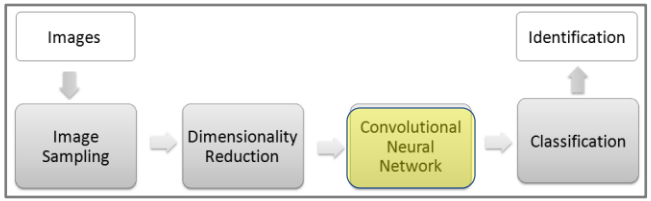
- Karhunen-Loeve (KL) transform
  - Objective function:

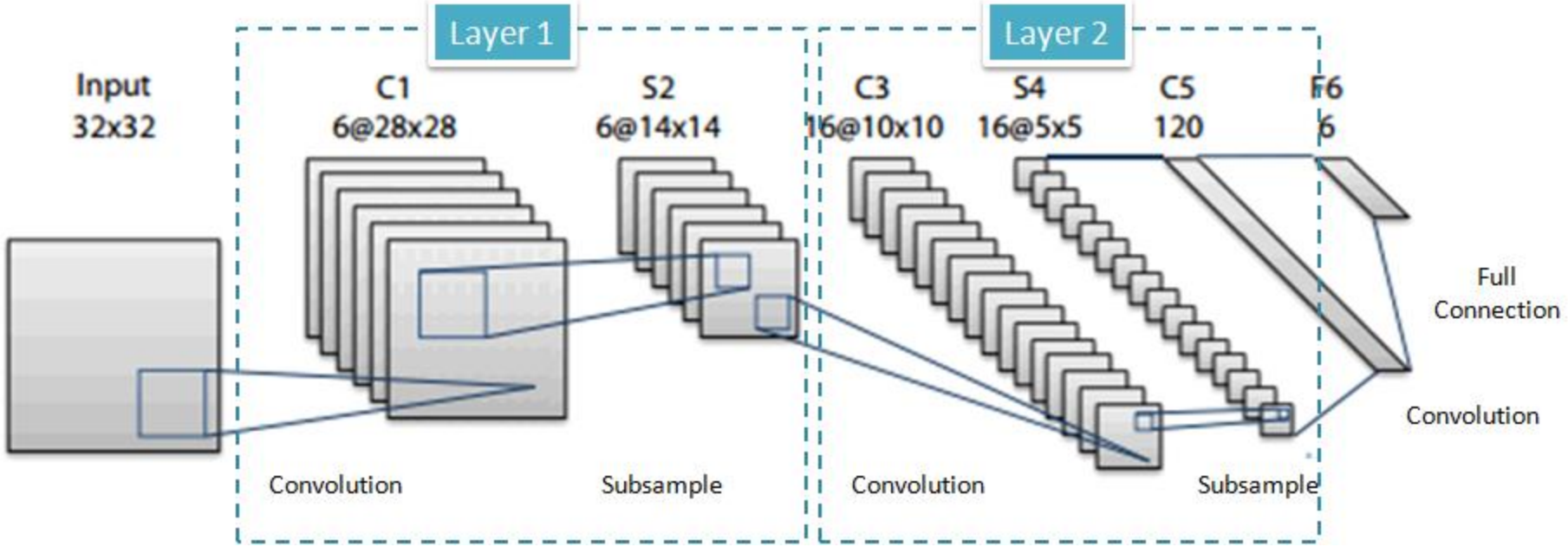
$$\mathbf{u}_2^\top \mathcal{C}\mathbf{u}_2 - \lambda(\mathbf{u}_2^\top \mathbf{u}_2 - 1) - \phi \mathbf{u}_2^\top \mathbf{u}_1$$

$$\hat{\mathbf{x}}_{k-1} = \mathbf{x} - \sum_{i=1}^{k-1} \mathbf{u}_i \mathbf{u}_i^\top \mathbf{x}$$

$$\mathbf{u}_k = \arg \max_{\|\mathbf{u}\|=1} E[(\mathbf{u}^\top \hat{\mathbf{x}}_{k-1})^2]$$

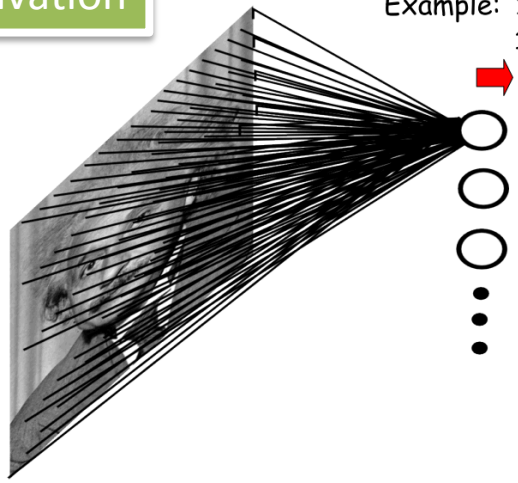
Convolutional Network



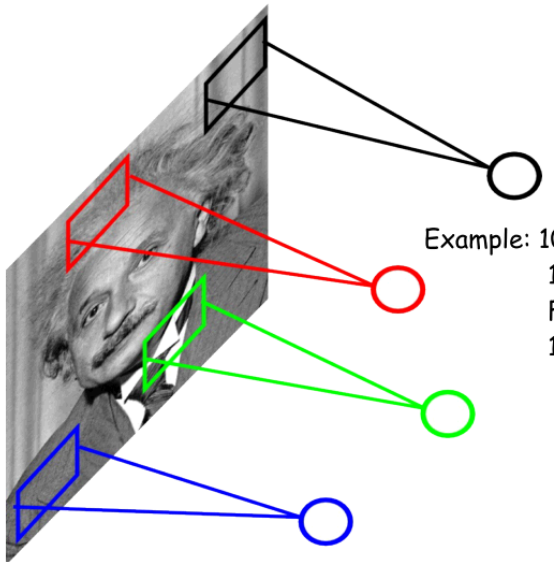


Convolutional Network

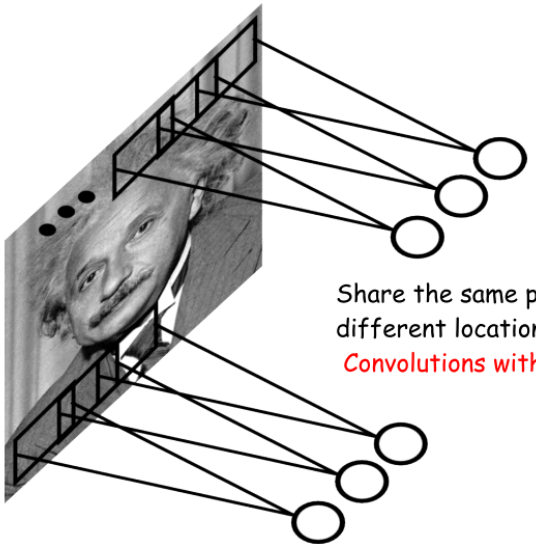
Motivation



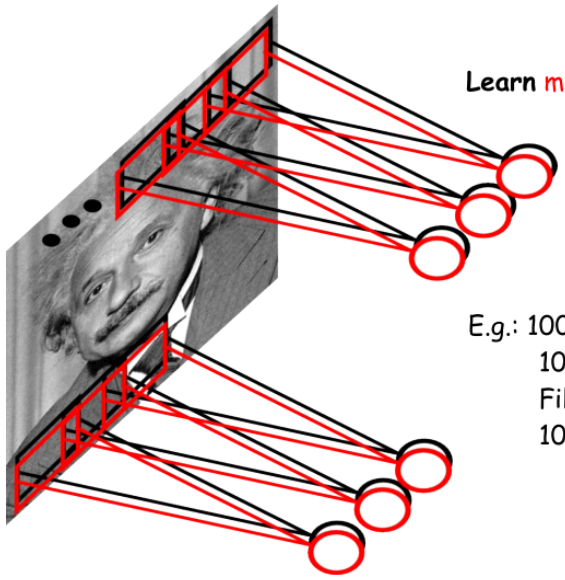
Example: 1000x1000 image
1M hidden units
10^12 parameters!!!



Example: 1000x1000 image
1M hidden units
Filter size: 10x10
100M parameters



Share the same parameters across different locations:
Convolutions with learned kernels



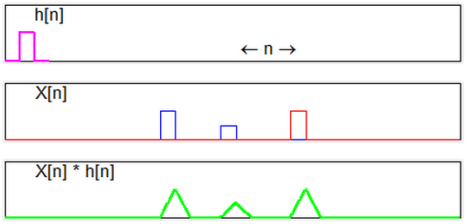
Learn multiple filters.

E.g.: 1000x1000 image
100 Filters
Filter size: 10x10
10K parameters

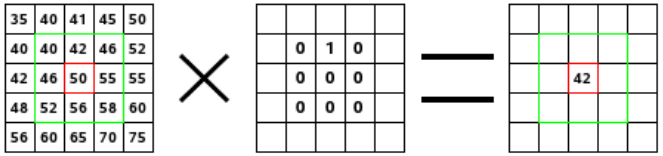
Convolutional Network

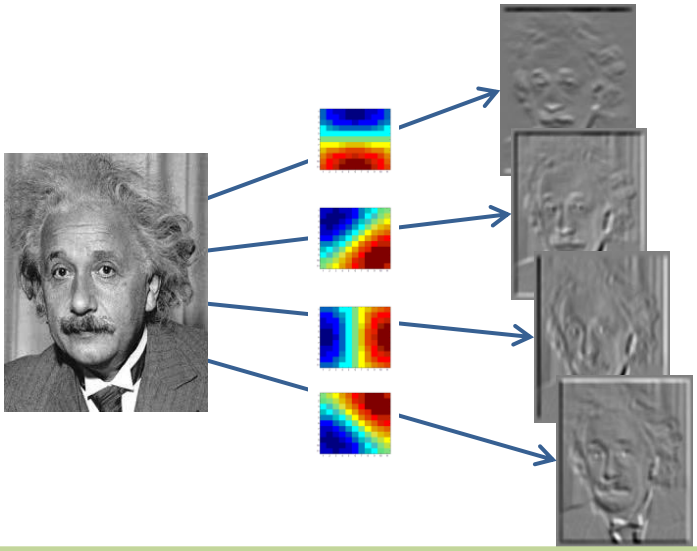
Convolution

1D



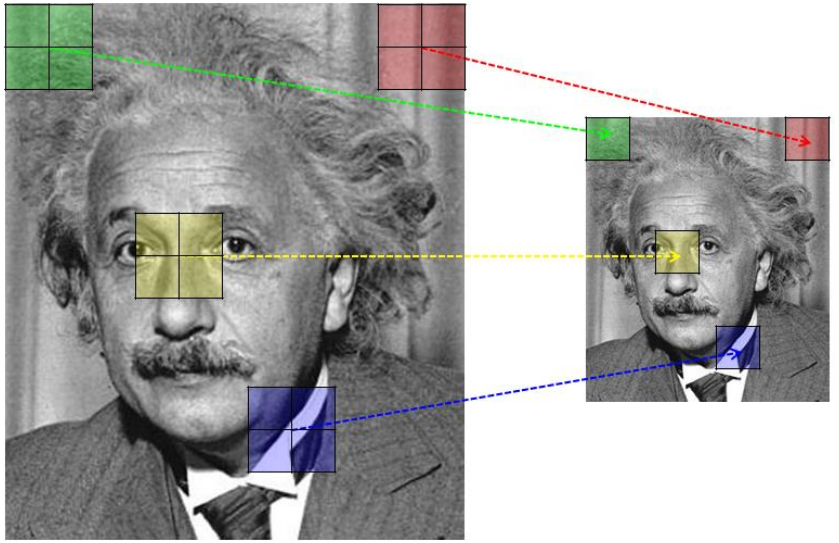
2D





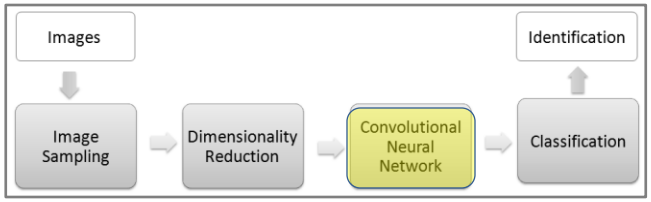
Subsample

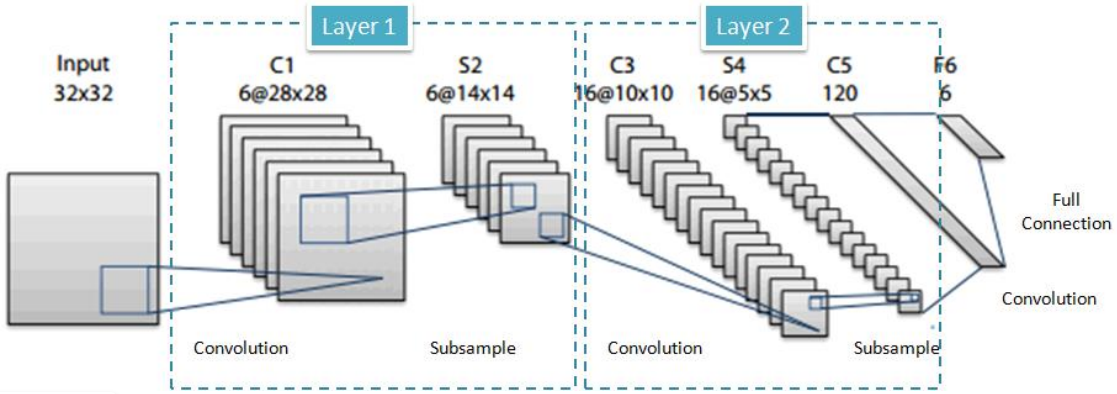
local averaging operator

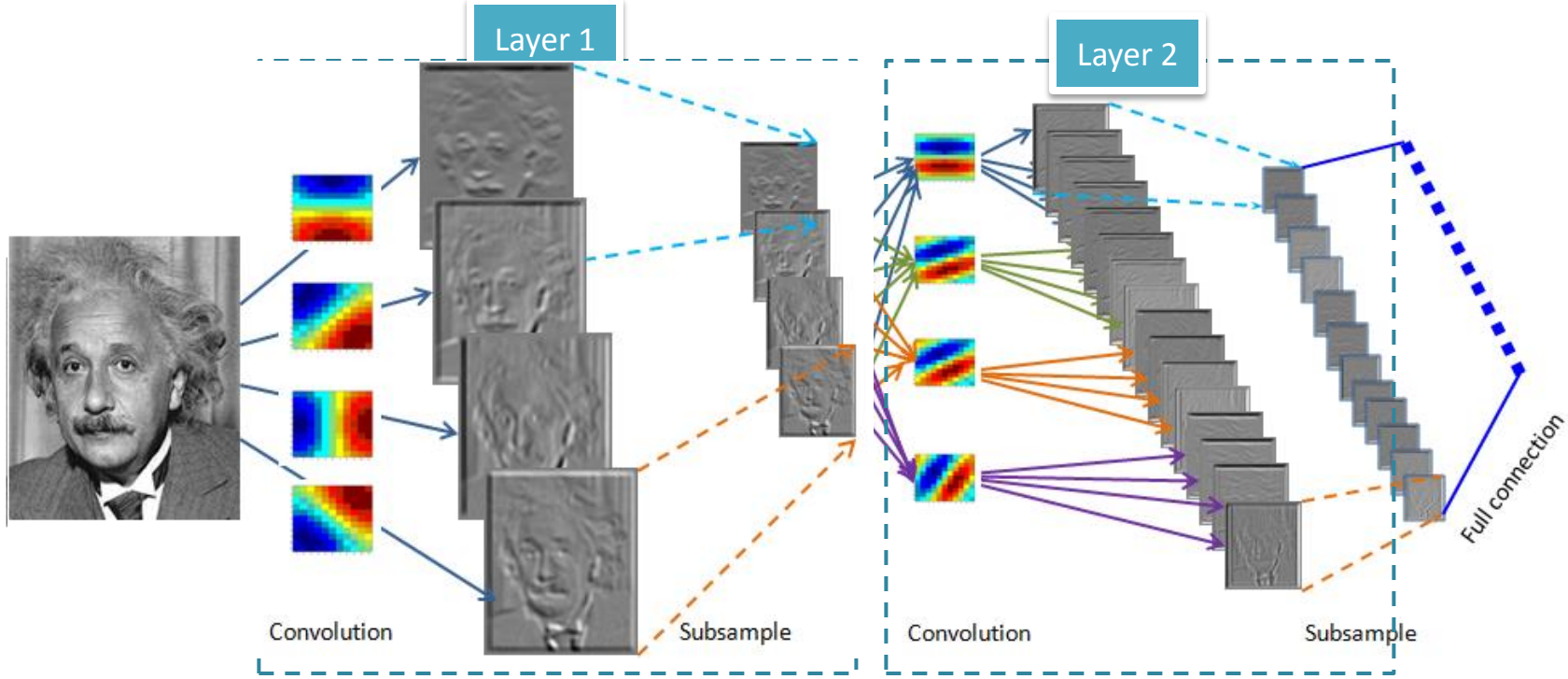


$$y_{ij} = \frac{1}{4}\left(x_{2i,2j} + x_{2i+1,2j} + x_{2i,2j+1} + x_{2i+1,2j+1}\right)$$

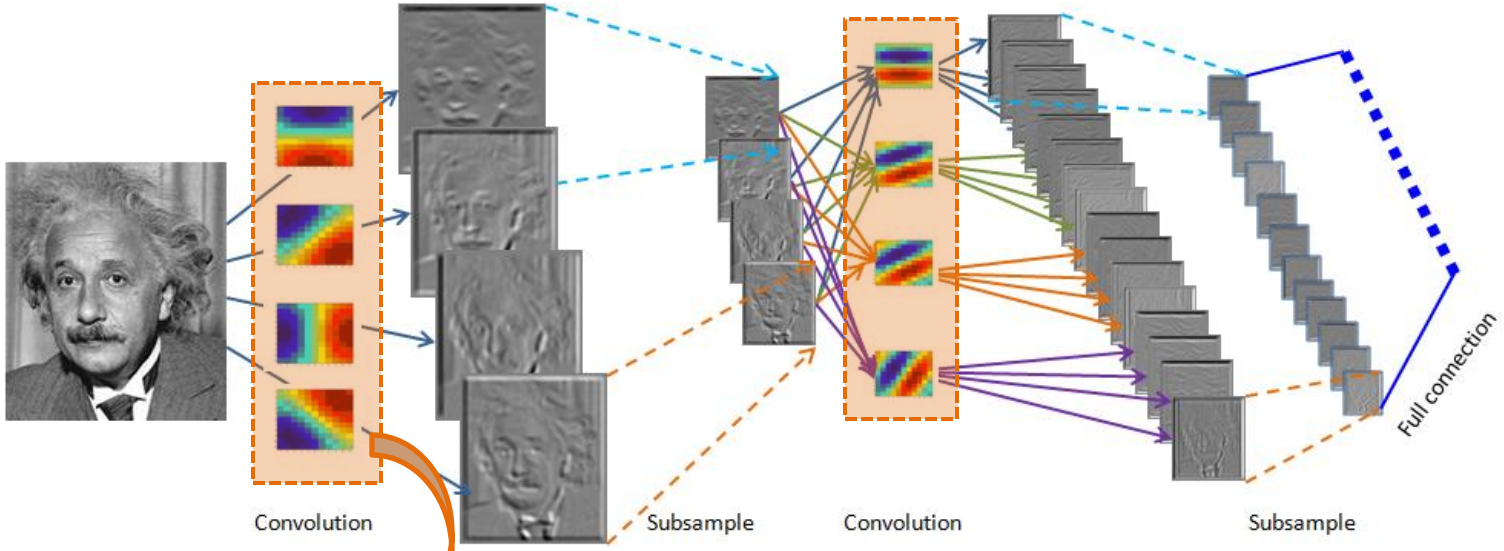
Convolutional Network

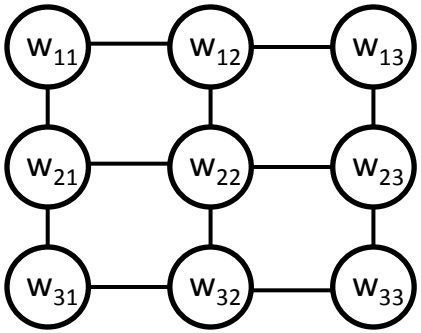






Convolutional Network





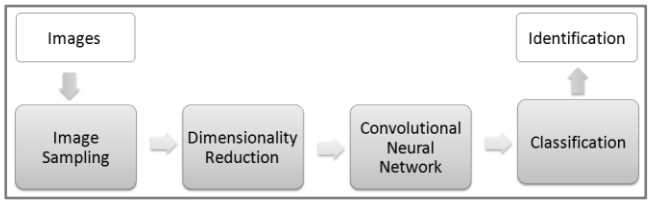
Backpropagation gradient-descent procedure

Backpropagationalgorithm for standard MLP

$$E_p = \frac{1}{2}||\mathbf{o_p} - \mathbf{t_p}||^2 = \frac{1}{2}\sum_{k=1}^{K}(o_{pk} - t_{pk})^2$$

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} + \Delta w_{ji}^{(l)} = w_{ji}^{(l)} - \lambda \frac{\partial E_p}{\partial w_{ji}^{(l)}}$$

Convolutional Neural Network - System



Images

Image Sampling

Dimensionality Reduction

SOM

K-L Transform

Convolution Neural Network

Feature Extraction

MLP Style Classifier

Nearest – Neighbor Classifier

Multi-Layer Perceptron

Classification

Convolutional Neural Network – Extensions

LeNet-5 http://yann.lecun.com/exdb/lenet/

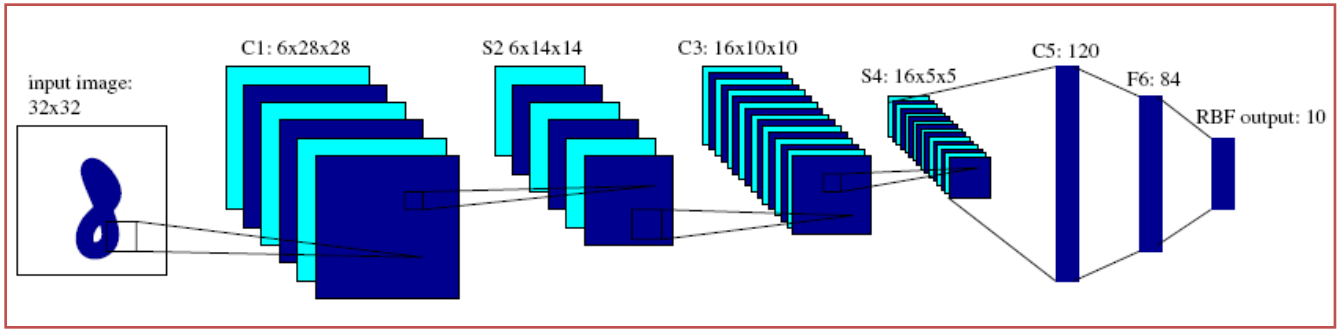
C1,C3,C5 : Convolutional layer.

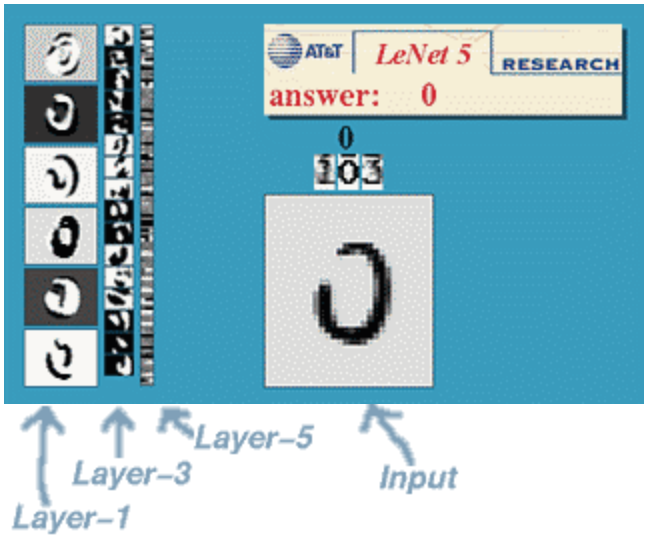
5 × 5 Convolution matrix.

S2 , S4 : Subsampling layer.

Subsampling by factor 2.

F6 : Fully connected layer.





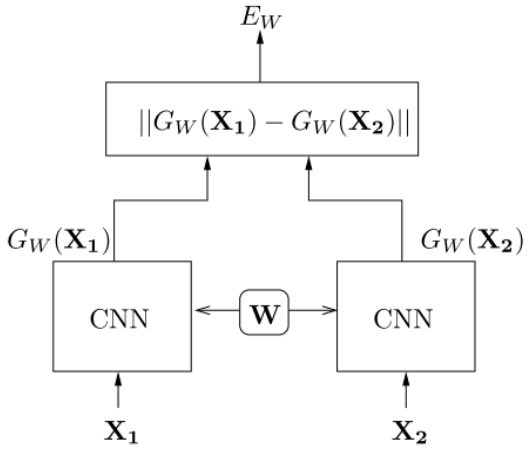
About 187,000 connection.

About 14,000 trainable weight

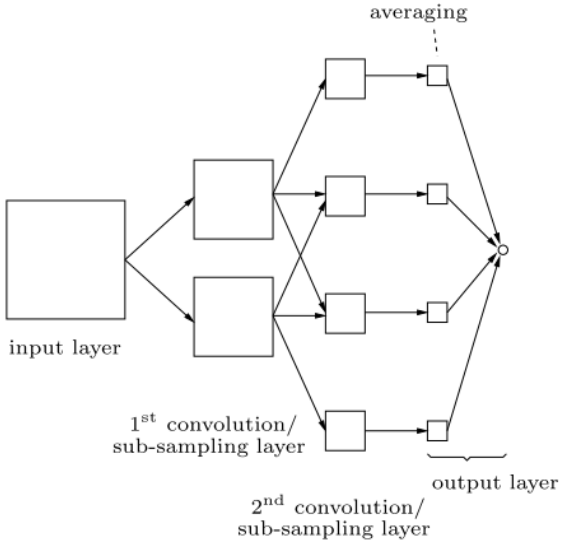
Convolutional Neural Network – Extension and variants

Space Displacement Neural Networks (**SDNN**)

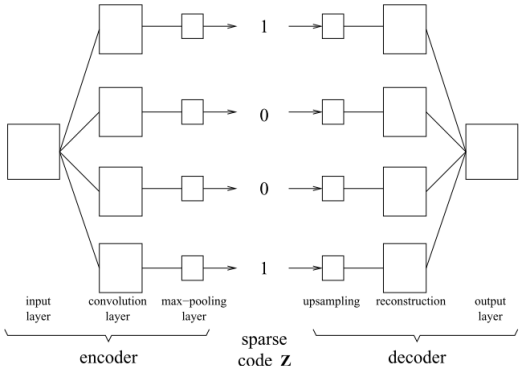
Siamese CNNs



Shunting **I**nhibitory **Co**nvolutional **N**eural **Net**works (**SICoNNet**)



Sparse Convolutional Neural Networks (**Sparse CNN**)



Convolutional Neural Network – Experiment & Comparison

200 training images and 200 test images from ORL database (AT&T).

Various Experiments
❑Variation of the number of output classes
❑Variation of the dimensionality of the SOM
❑Variation of the quantization level of the SOM
❑Variation of the image sample extraction ❑algorithm
❑Substituting the SOM with the KL transform
❑Replacing the CN with an MLP
…

Number of classes	10	20	40
Error rate	1.33%	4.33%	5.75%

SOM Dimension	1	2	3	4
Error rate	8.25%	6.75%	5.75%	5.83%

SOM Size	4	5	6	7	8	9	10
Error rate	8.5%	5.75%	6.0%	5.75%	3.83%	3.83%	4.16%

Input type	Pixel intensities	Differences w/base intensity
Error rate	5.75%	7.17%

Dimensionality reduction	Linear PCA	SOM
Error rate	5.33%	3.83%

	Linear PCA	SOM
MLP	41.2%	39.6%
CN	5.33%	3.83%

Comments

- Convolutional Neural Networks are a special kind of multi-layer neural networks.
- Like almost every other neural networks they are trained with a version of the back-propagation algorithm.
- Convolutional Neural Networks are designed to recognize visual patterns directly from pixel images with minimal preprocessing.
- Shared weights: all neurons in a feature share the same weights.
- In this way all neurons detect the same feature at different positions.
- Reduce the number of free parameters in the input image.


Thank you