Unimodal Representations

Đàm Quang Tuấn

Lecture Objectives

- Dimension of heterogeneity
- Image representations
  - Image gradients, edges, kernels
- Convolution neural network (CNN)
  - Convolution and pooling layers
- Visualizing CNNs
- Region-based CNNs
- Sequence modeling with convolution networks
- Team matching event

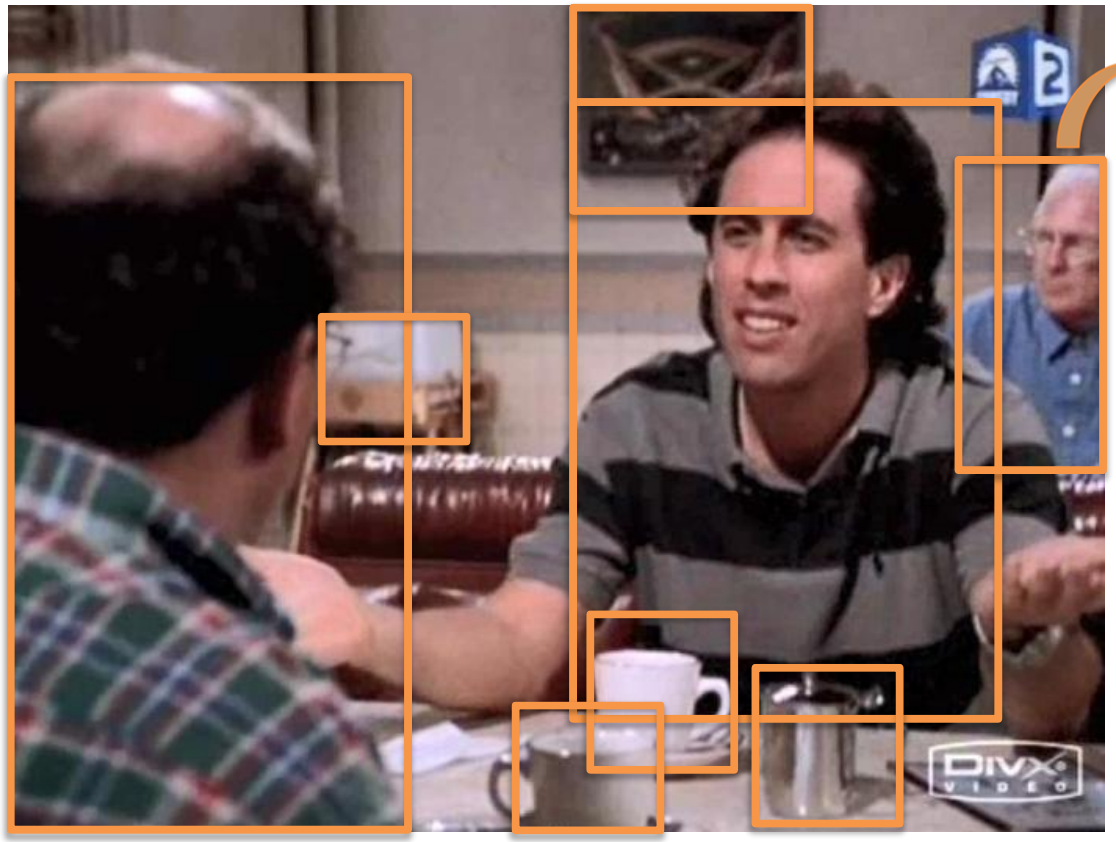
Image Representations

How Would You Describe This Image?

?

88
88
85
38
20
22
24
21
23
82
80
79
35
25
26
28
22
22
84
78
80
⋮

Object-Based Visual Representation



“person” label +

Appearance descriptor

- Age
- Expression
- Clothes …

Feature vector

88
88
85
38
20
22
24
21
23
82
80
79
35
25
26
28
22
22
84
78
80

Object Descriptors

How to represent and detect an object?

Many approaches over the years…

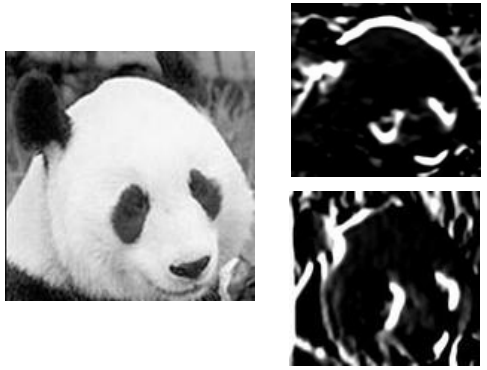
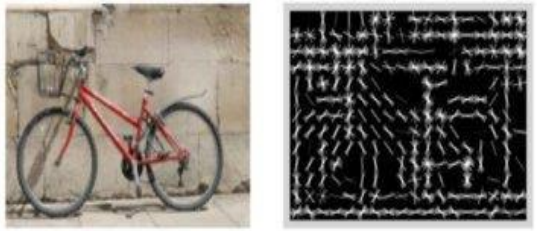


Image gradient

Edge detection



Histograms of Oriented Gradients

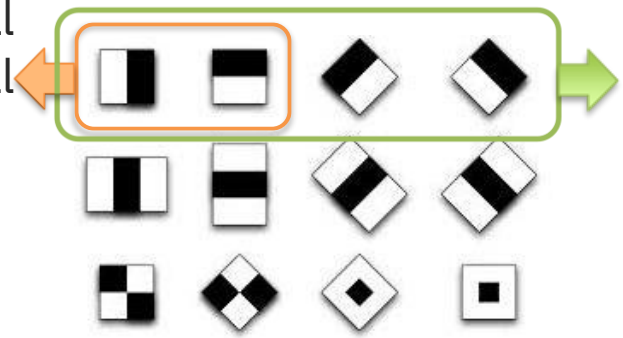
Optical Flow

Object Descriptors

How to represent and detect an object?

Many approaches over the years…

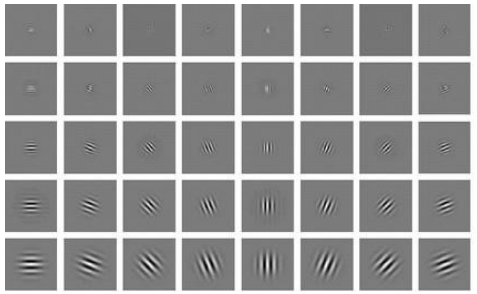
Horizontal and vertical gradients



Oriented gradients

Haar Wavelets

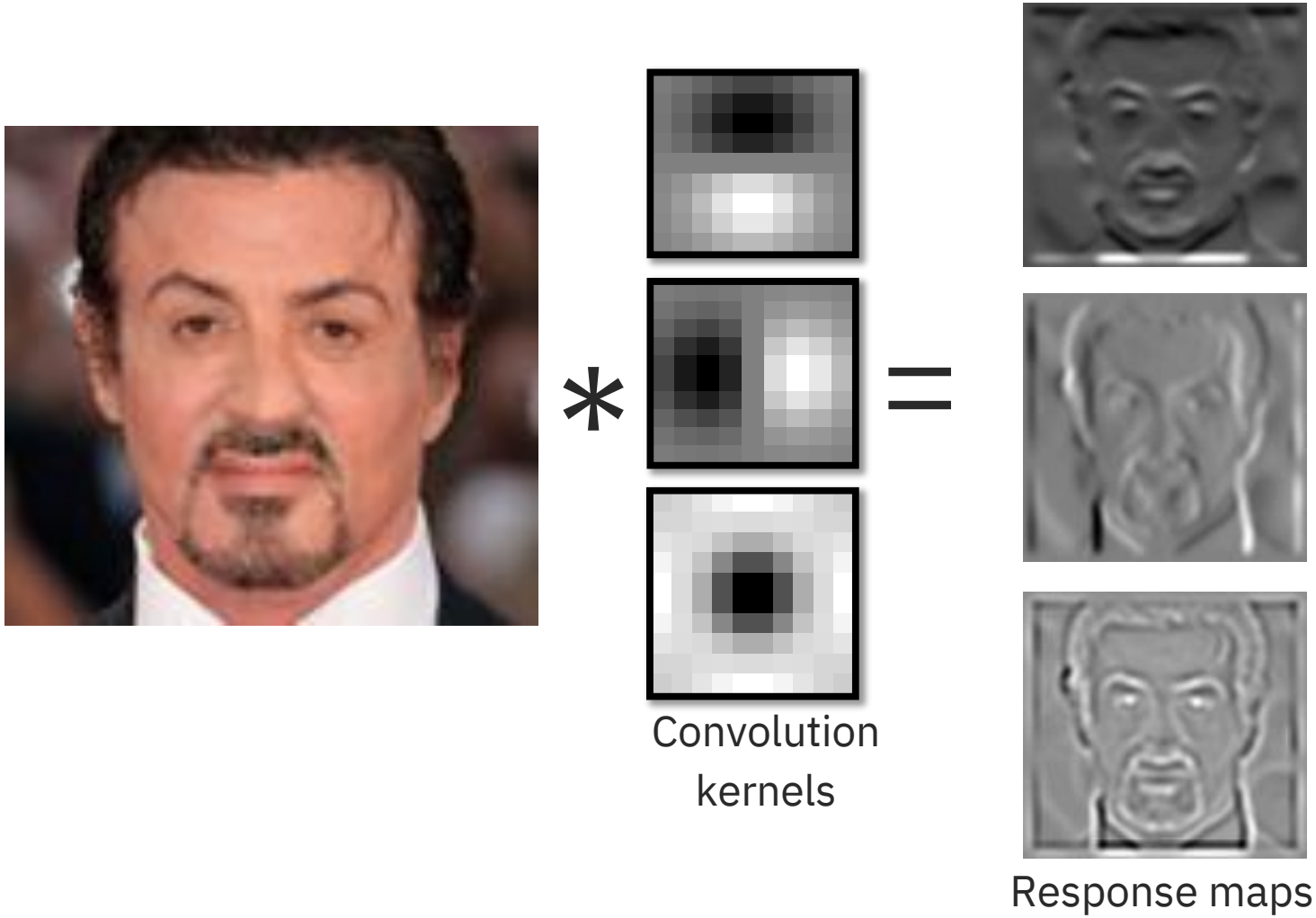
Templates tested on the image *(i.e., convolution kernels)*



Gabor filters

Inspired by visual cortex

Convolution Kernels

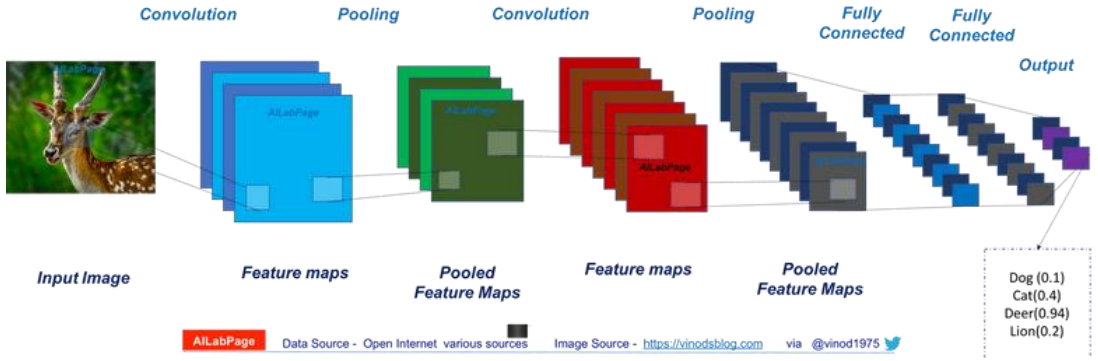


Object Descriptors

How to represent and detect an object?

Many approaches over the years…

Convolutional Neural Network (CNN)



More details about CNNs is coming…

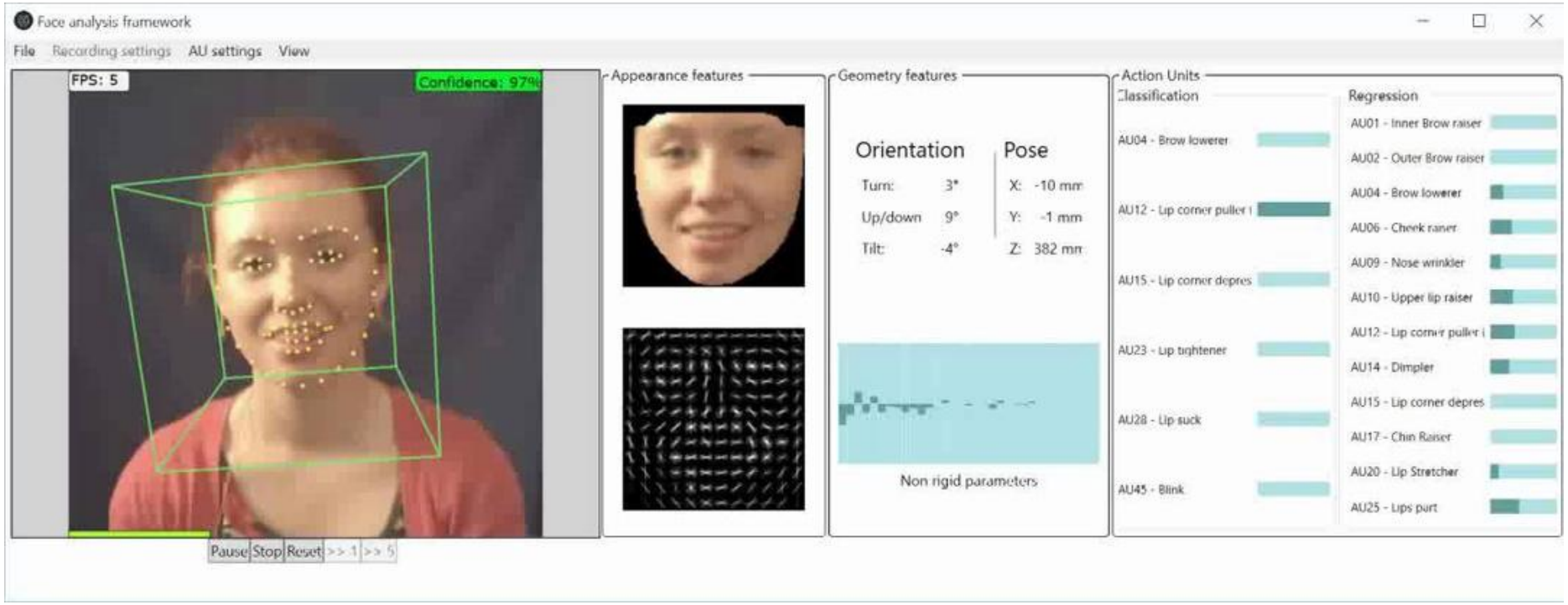
… and we will also talk about visual Transformers in coming weeks…

And images are more than a list of objects!

One representation, lots of tasks

https://github.com/facebookresearch/detectron2

Facial expression analysis



[**OpenFace: an open sourcefacial behavior analysis toolkit**, T. Baltrušaitiset al., 2016]

Articulated Body Tracking: OpenPose

https://github.com/CMU-Perceptual-Computing-Lab/openpose

See appendix for list of available tools for automatic visual behavior analysis

Convolutional Neural Networks

Why using Convolutional Neural Networks?

Goal: building more abstract, hierarchical visual representations

Key advantages:

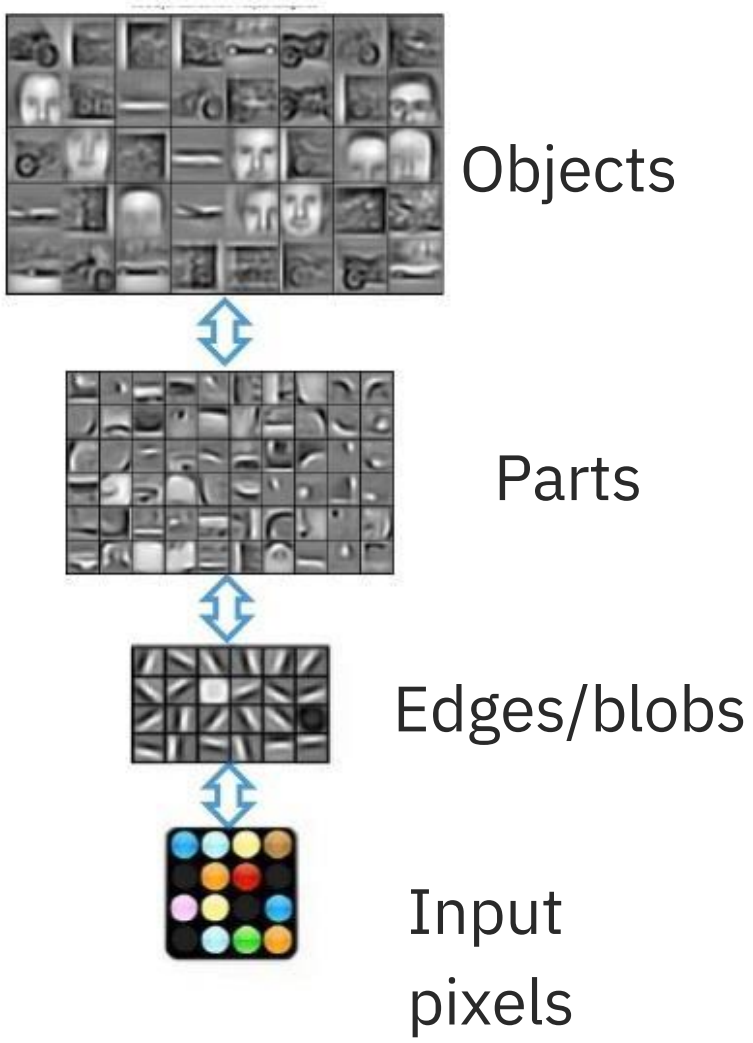
1)Inspired from visual cortex

2)Encourages visual abstraction

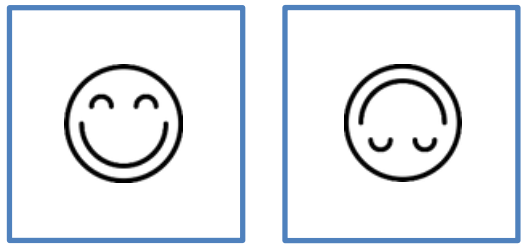
3)Exploits *translation invariance*

4)Kernels/templates are learned

5)Fewer parameters than MLP



Translation Invariance



2 Data Points –Which one is up?

- MLP can easily learn this task (possibly with only 1 neuron!)

What happens if the face is slightly translated?

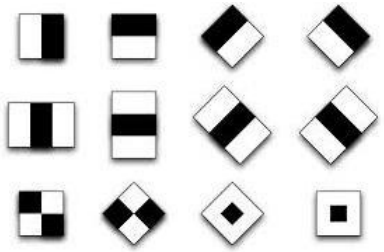
- The model should still be able to classify it

Conventional MLP models are not translation invariant!

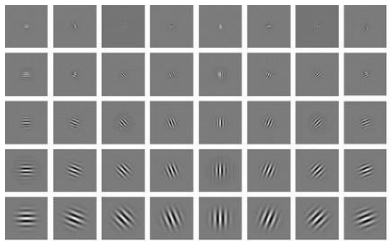
- But CNNs are kernel-based, which helps with translation invariance and reduce number of parameters

Predefined vs Learned Kernels

Predefined kernels



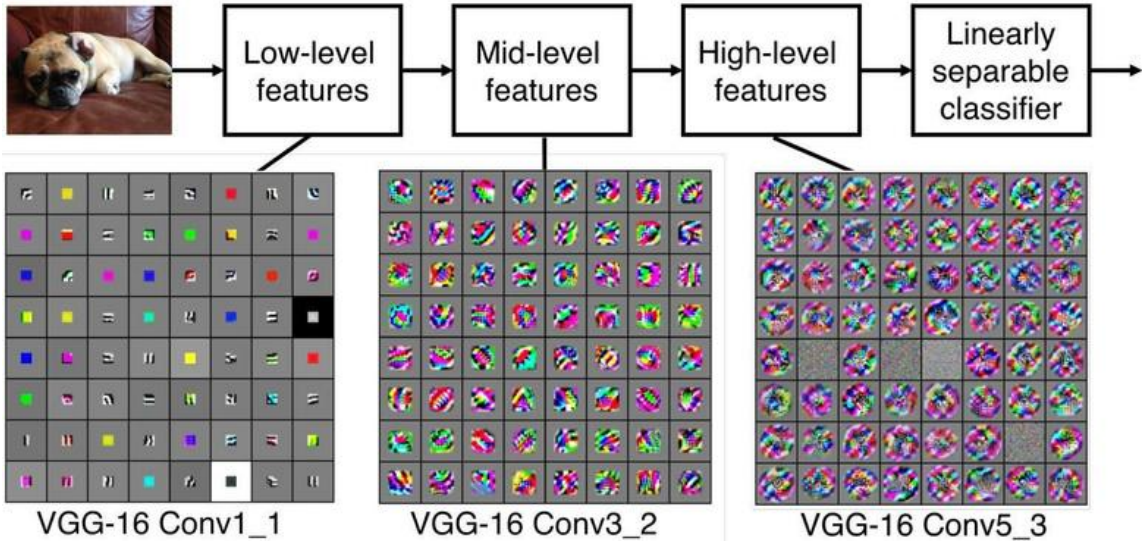
Haar Wavelets



Gabor filters

Learned kernels

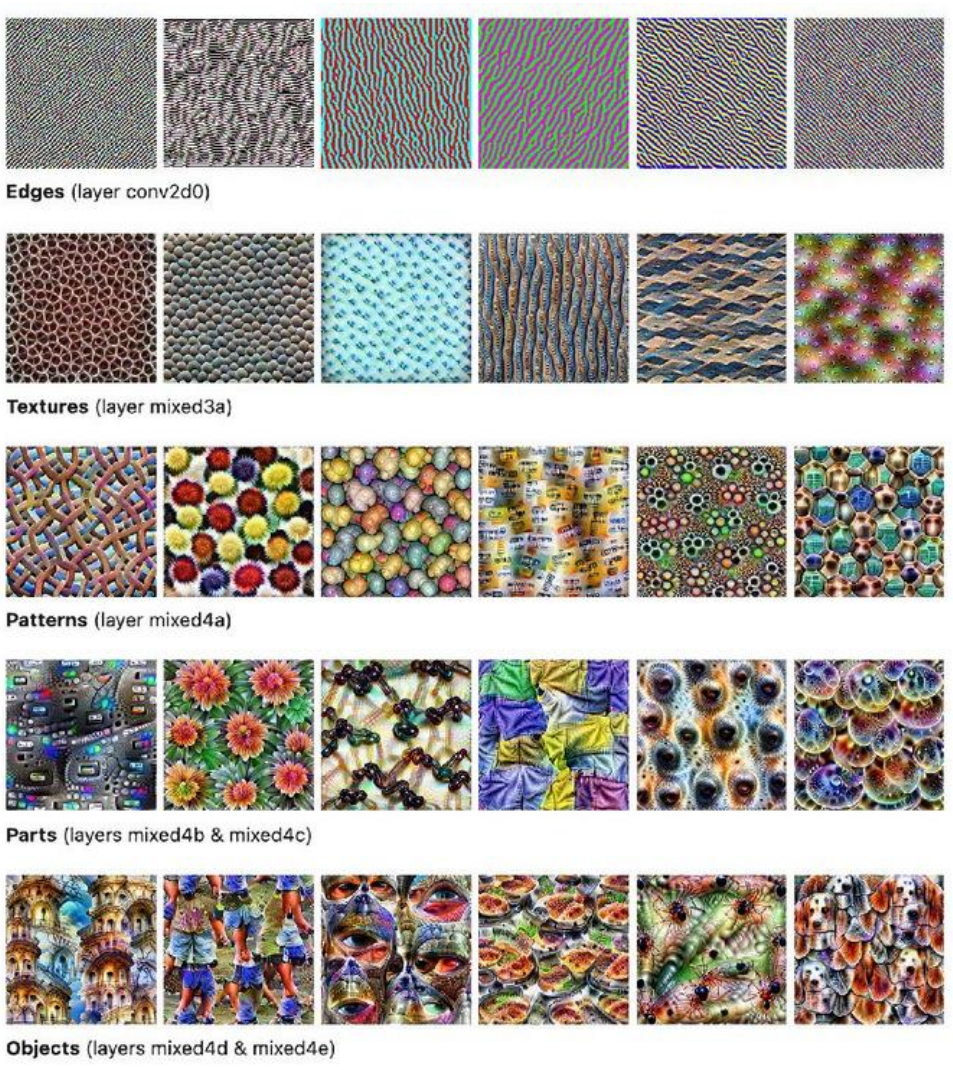
Convolutional Neural Network (CNN)



With CNNs, the kernel values are learned as model parameters

LearnedFilters(aka Convolution Kernels)

ht t ps : //distill.pub/2017/f eat u re-visualization/



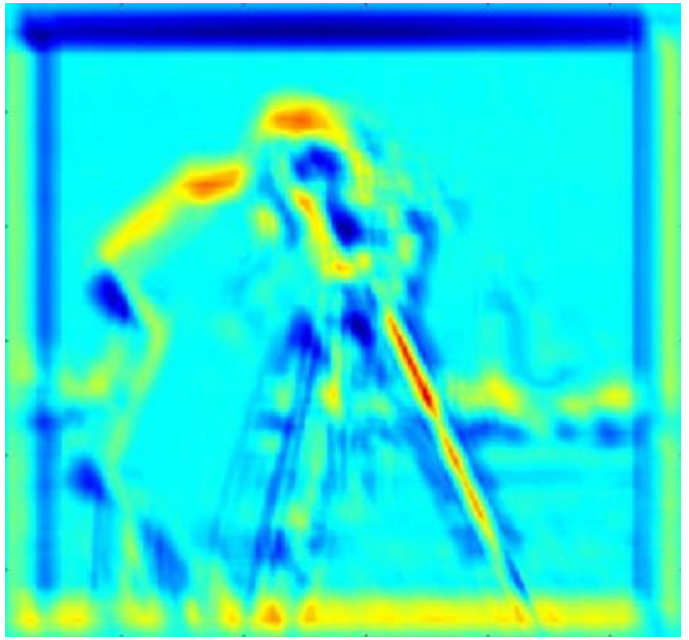
Convolution in 2D –Example

Input image

*

Convolution kernel

=



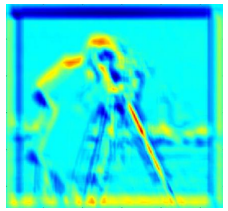
Response map

Convolution as a Fully-Connected Network

Input

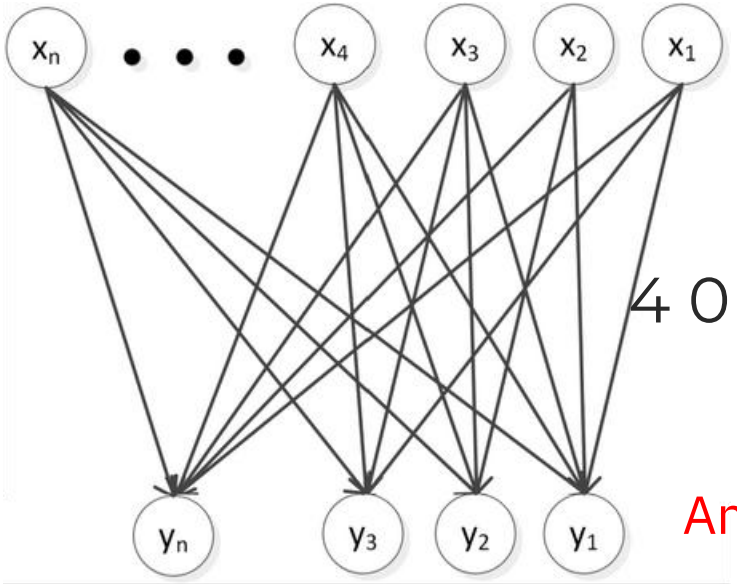
(image)

Output



(response map)

Input: all pixels



Output: kernel responses

Not efficient!

200×200 image requires

$40,000 \times n$ parameters

(where n is size of kernel)

And it may learn different kernels for different pixel positions

➡ Not translation invariant

Convolutional Neural Layer

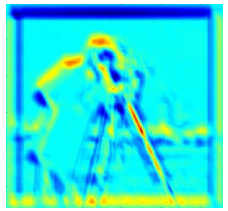
Input

(image)

Weighted sum

$W\ x$

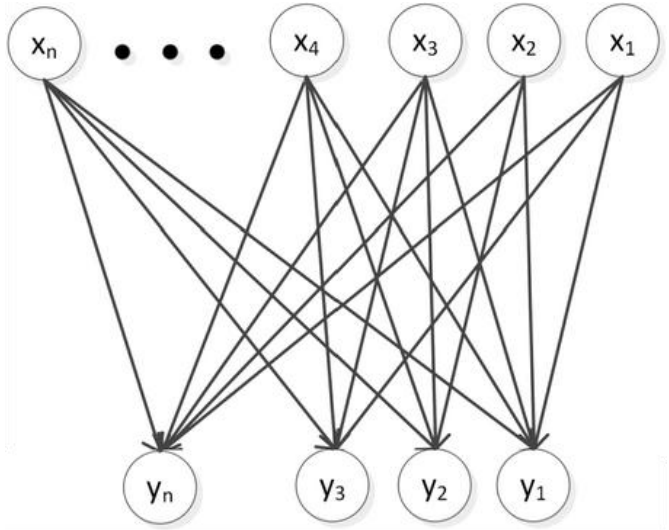
Output



(response map)

$y = W\ x$

Input: all pixels



Output: kernel responses

Example with 1D kernel:

w_1	w_2	w_3

Convolution kernel

Convolutional Neural Layer

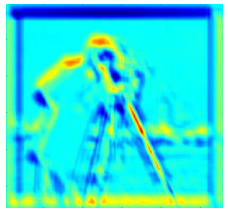
Input

(image)

Weighted sum

$W\ x$

Output

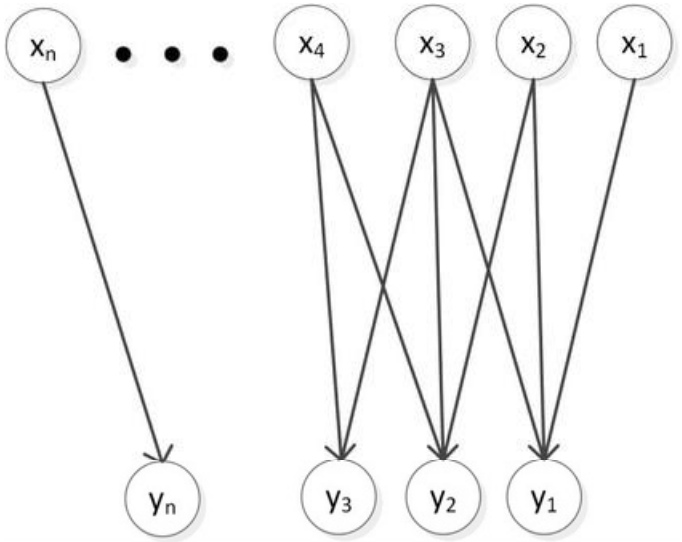


(response map)

$y = W\ x$

Modification 1: Sliding window –Only apply the kernel to a small region

Input: all pixels



Output: kernel responses

Example with 1D kernel:

w_1	w_2	w_3

Convolutional Neural Layer

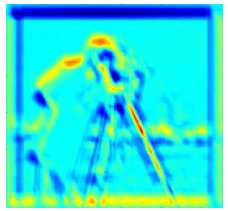
Input

(image)

Weighted sum

$W x$

Output

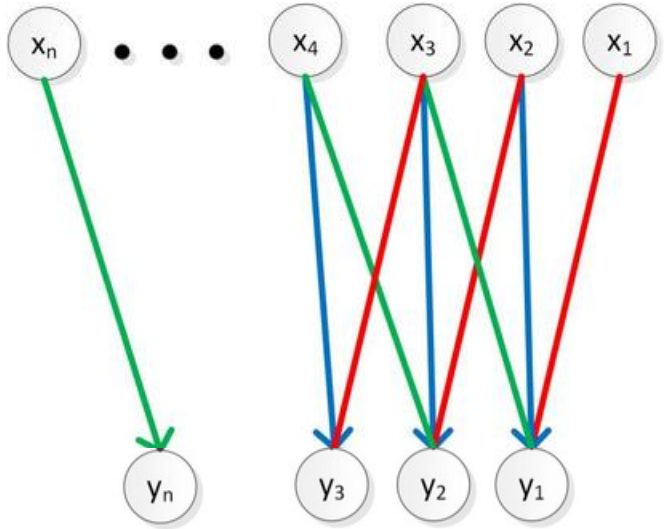


(response map)

$y = W\ x$

Modification 2: Same kernel applied to all sliding windows

Input: all pixels



Output: kernel responses

Example with 1D kernel:

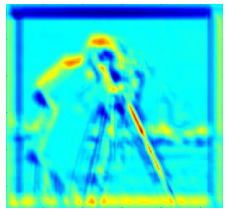
Convolutional Neural Layer

Modification 2: Same kernel applied to all sliding windows

Input

(image)

Output



(response map)

$$y = W\ x$$

$$W = \begin{pmatrix} w_1 & w_2 & w_3 & & 0 & 0 & 0 \\ 0 & w_1 & w_2 & \cdots & 0 & 0 & 0 \\ 0 & 0 & w_1 & & 0 & 0 & 0 \\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & & w_3 & 0 & 0 \\ 0 & 0 & 0 & \cdots & w_2 & w_3 & 0 \\ 0 & 0 & 0 & & w_1 & w_2 & w_3 \end{pmatrix}$$

Example with 1D kernel:

Can be implemented efficiently on GPUs

W will be 3D: 3rd dimension allows for multiple kernels

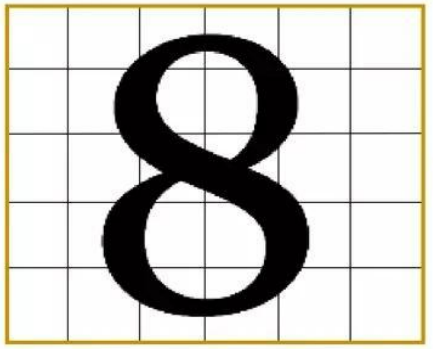
What is a Convolution Neural Network?

CNN is a feed forward neural network that is generally used to analyze visual images by processing data with grid like topology. A CNN is also known as a "***ConvNet***"

Convolution operation forms the basis of any Convolution Neural Network

In CNN, every image is represented in the form of arrays of pixel values

Real Image of the digit 8



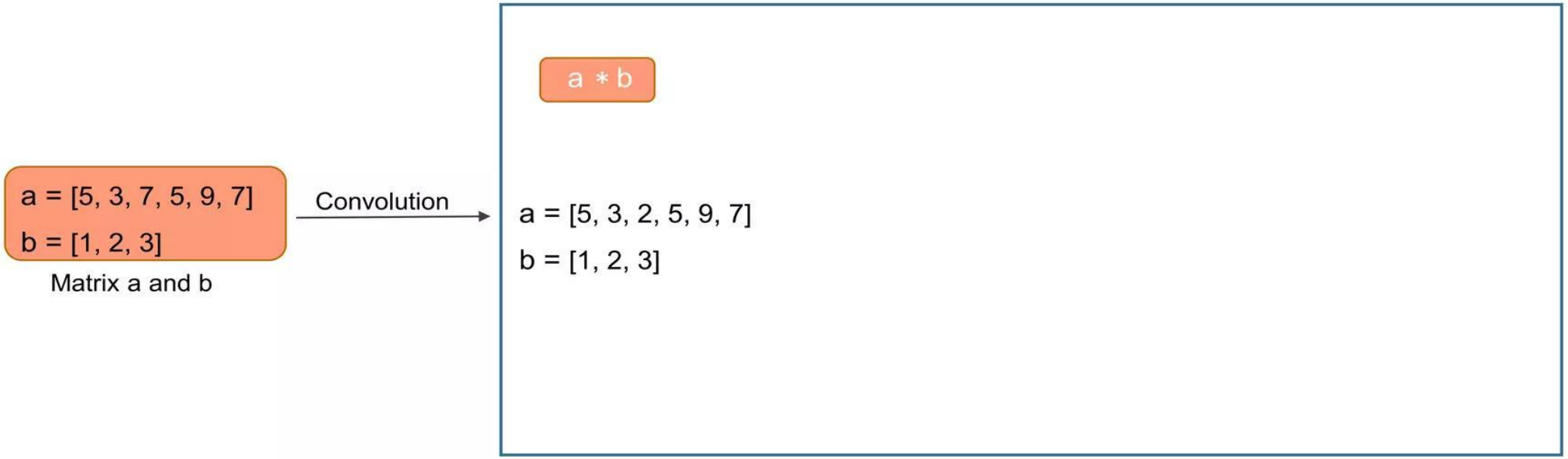
Represented in the form of an array

0	0	1	1	0	0
0	1	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	0	1	1	0	0

Digit 8 represented in the form of pixels of 0's and 1's

What is a Convolution Neural Network?

Let's understand the convolution operation using 2 matrices a and b of 1 dimension



What is a Convolution Neural Network?

Let's understand the convolution operation using 2 matrices a and b of 1 dimension

a = [5, 3, 7, 5, 9, 7]

b = [1, 2, 3]

Matrix a and b

Convolution

a * b

Multiply the arrays element wise

[5, 6, 6]

Sum the product

17

a = [5, 3, 2, 5, 9, 7]

b = [1, 2, 3]

a * b = [17,]

What is a Convolution Neural Network?

Let's understand the convolution operation using 2 matrices a and b of 1 dimension

a = [5, 3, 7, 5, 9, 7]

b = [1, 2, 3]

Matrix a and b

Convolution

a * b

Multiply the arrays element wise

Sum the product

a = [5, 3, 2, 5, 9, 7]

b = [1, 2, 3]

[5, 6, 6]

[3, 4, 15]

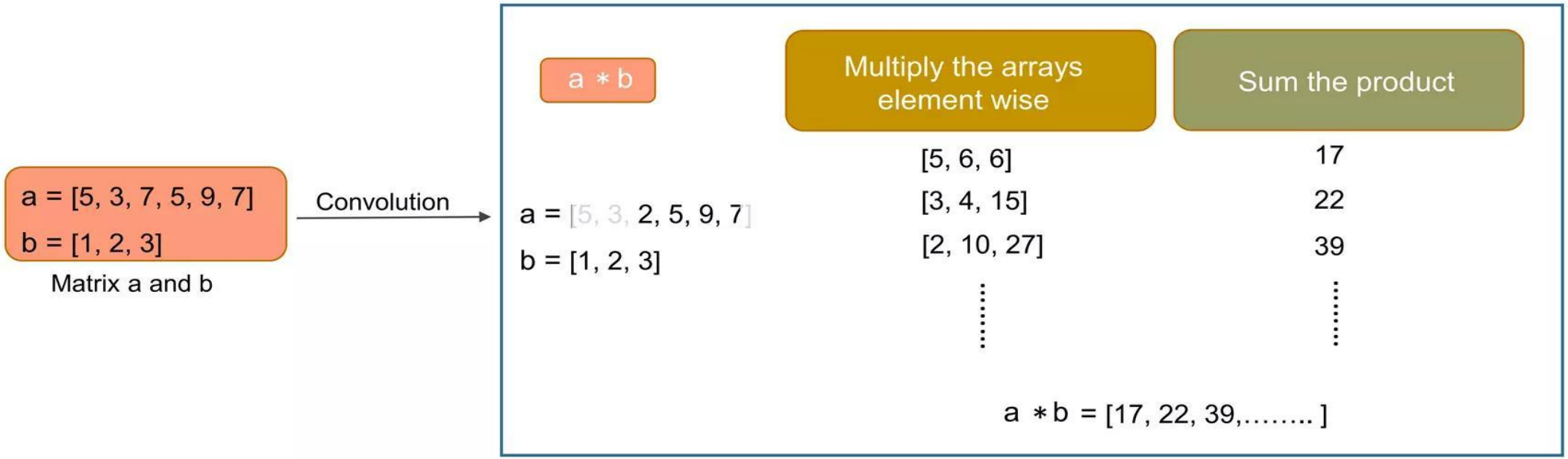
17

22

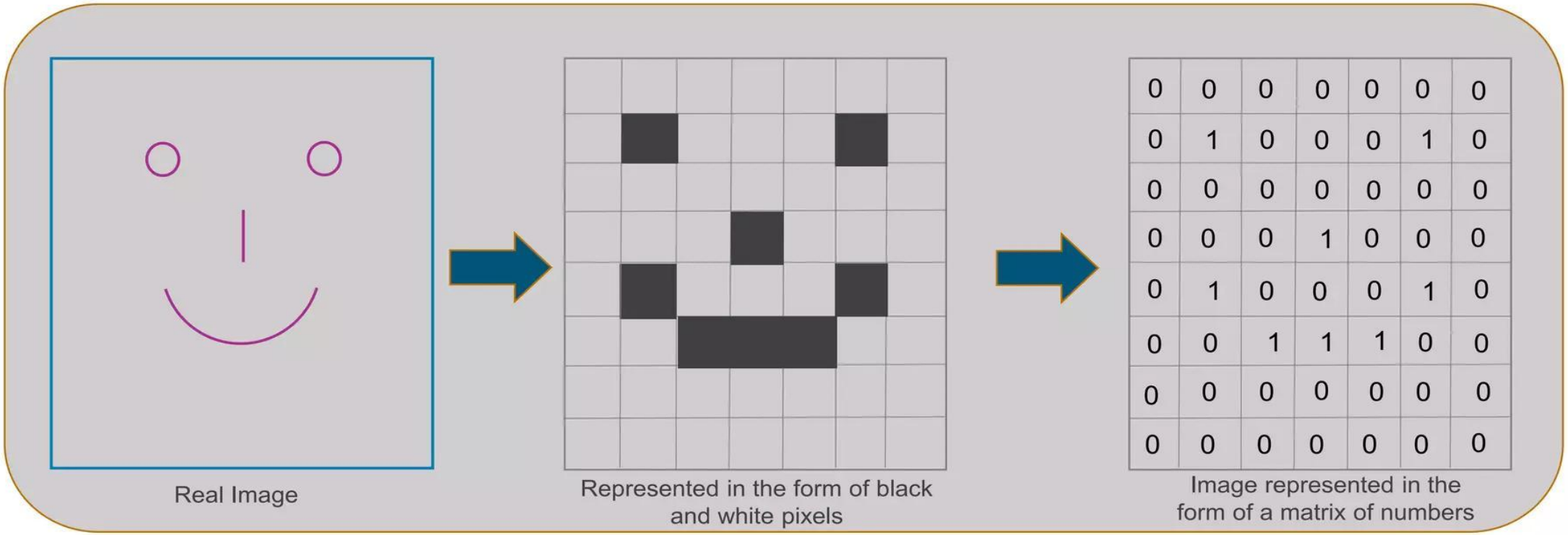
a * b = [17, 22]

What is a Convolution Neural Network?

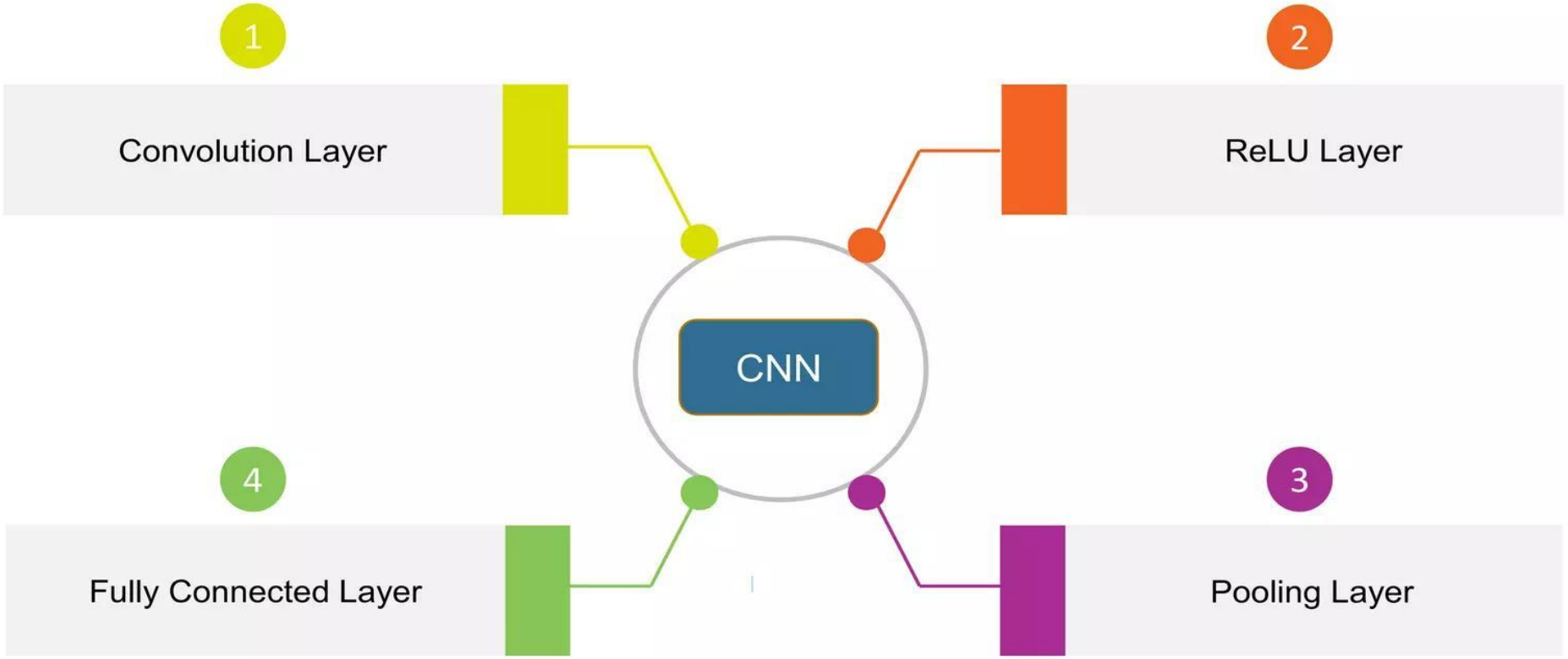
Let's understand the convolution operation using 2 matrices a and b of 1 dimension



How CNN recognizes images?



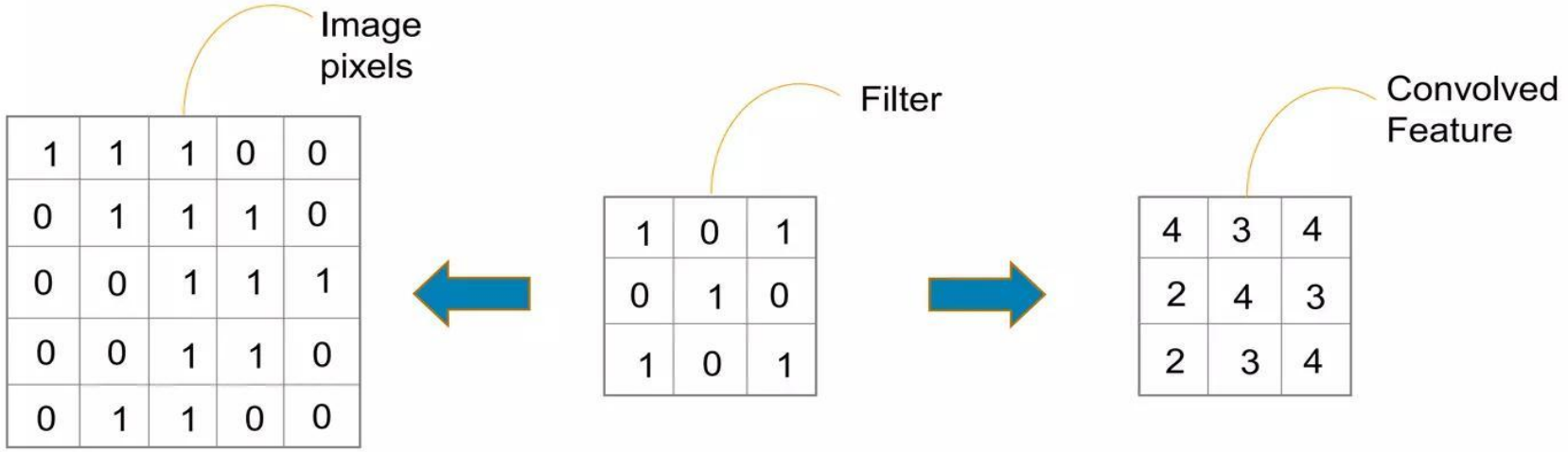
Layers in Convolution Neural Network



Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 × 5 image whose pixel values are only 0 and 1

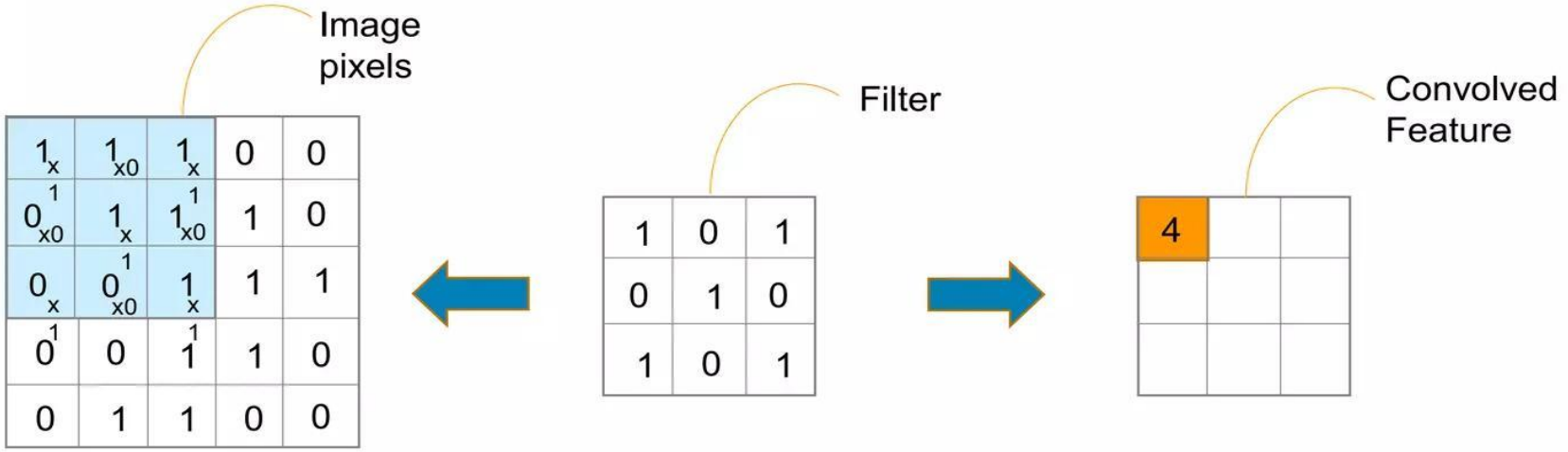


Sliding the filter matrix over the image and computing the dot product to detect patterns

Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 5 image whose pixel values are only 0 and 1

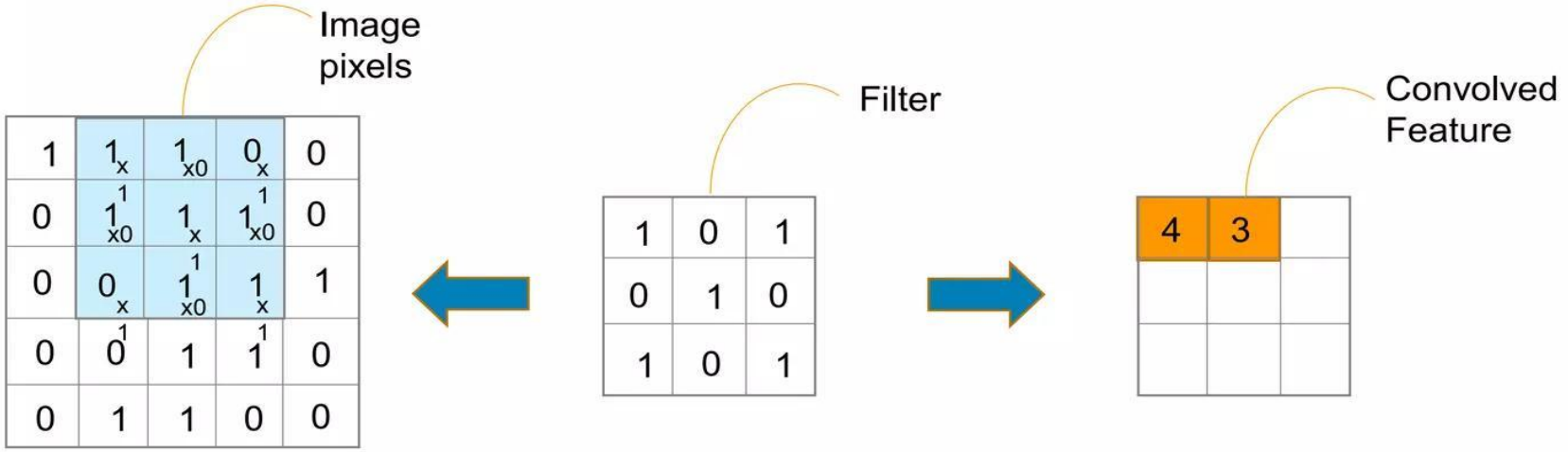


Sliding the filter matrix over the image and computing the dot product to detect patterns

Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 5 image whose pixel values are only 0 and 1

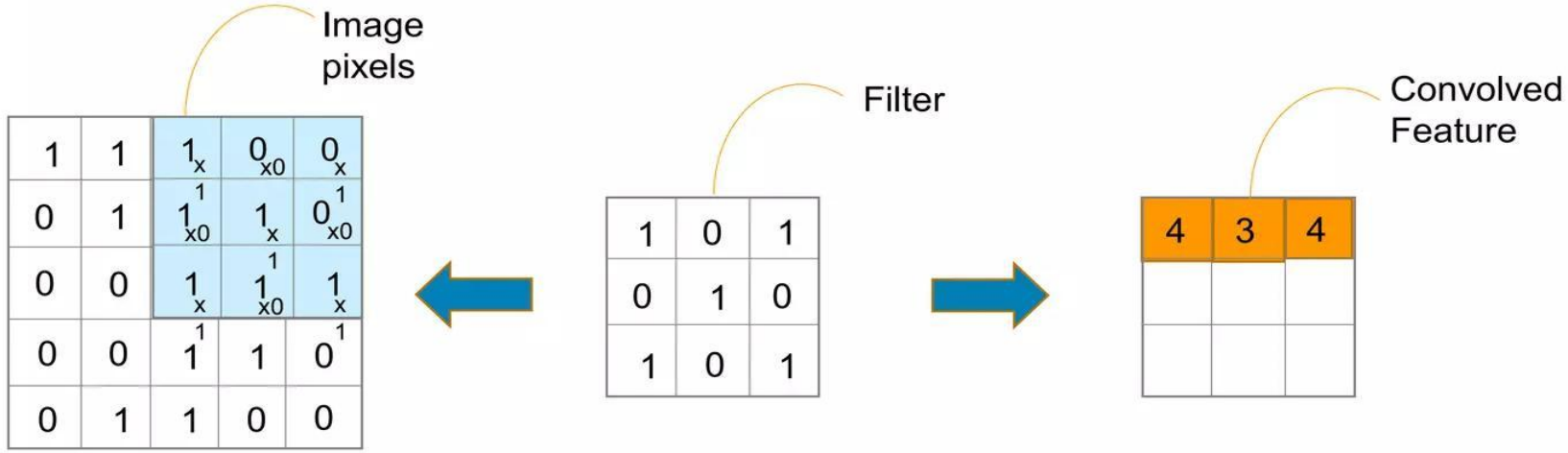


Sliding the filter matrix over the image and computing the dot product to detect patterns

Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 5 image whose pixel values are only 0 and 1

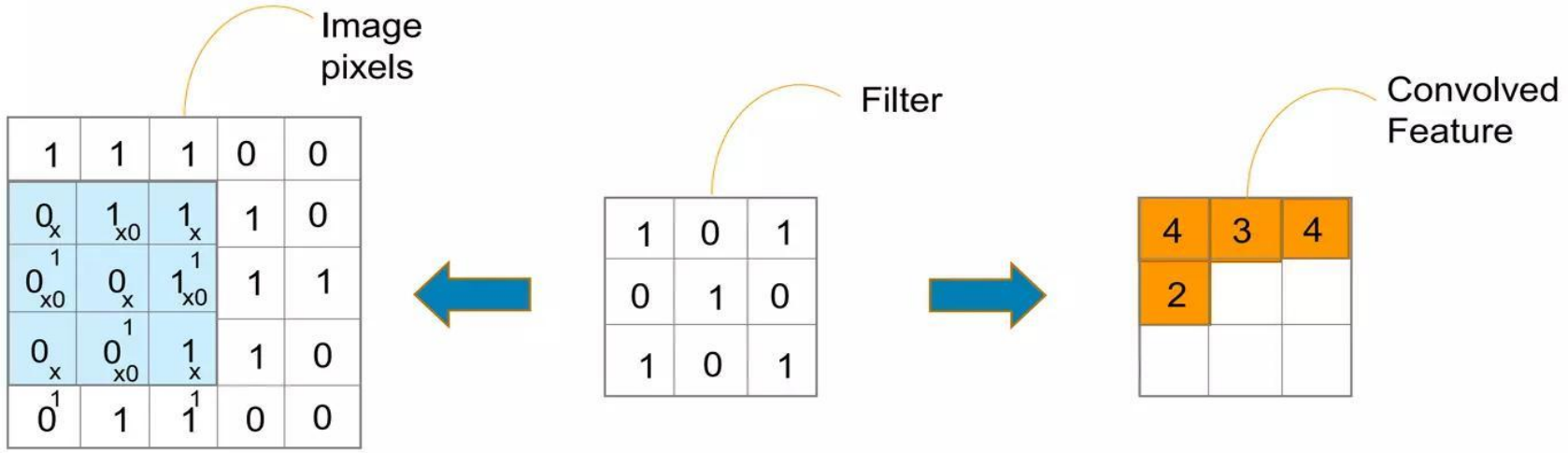


Sliding the filter matrix over the image and computing the dot product to detect patterns

Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 5 image whose pixel values are only 0 and 1

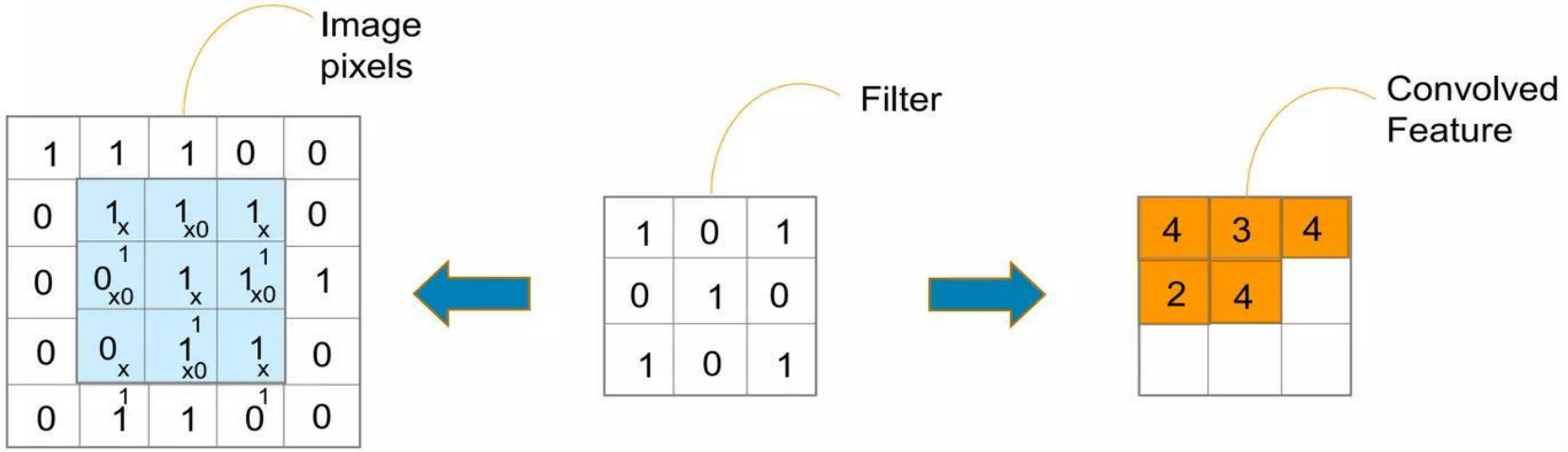


Sliding the filter matrix over the image and computing the dot product to detect patterns

Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 5 image whose pixel values are only 0 and 1

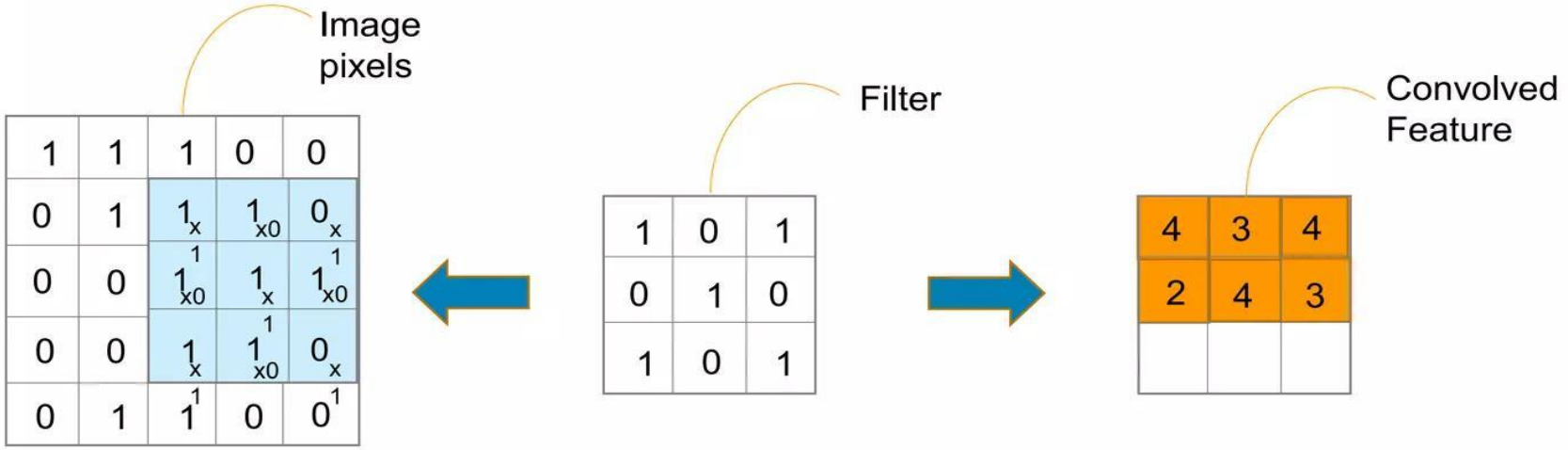


Sliding the filter matrix over the image and computing the dot product to detect patterns

Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 5 image whose pixel values are only 0 and 1

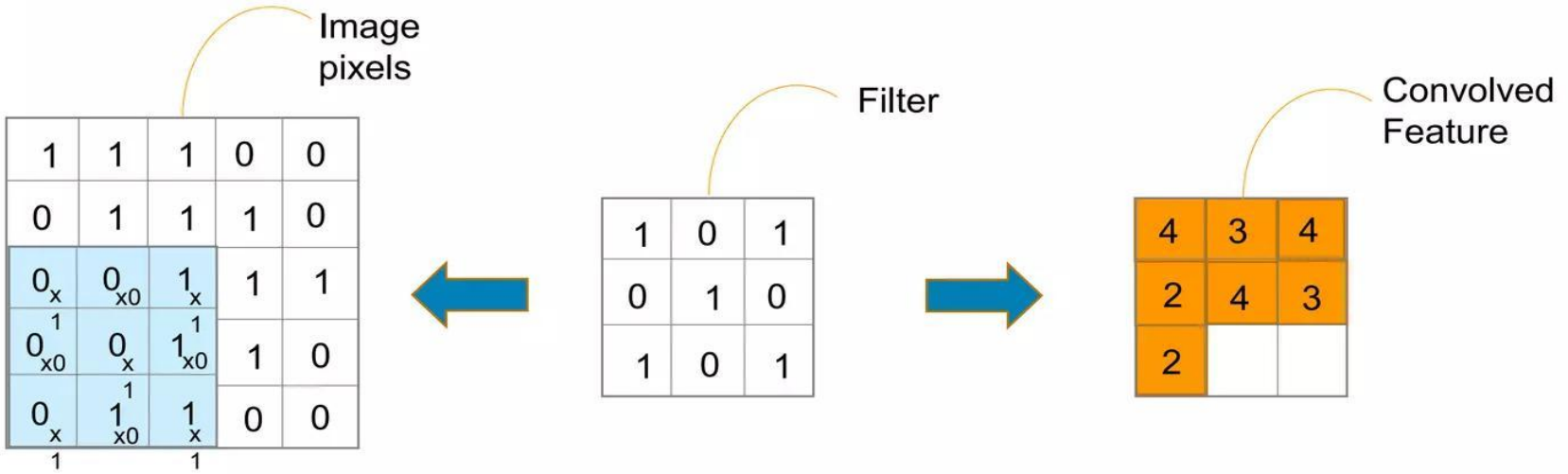


Sliding the filter matrix over the image and computing the dot product to detect patterns

Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 × 5 image whose pixel values are only 0 and 1

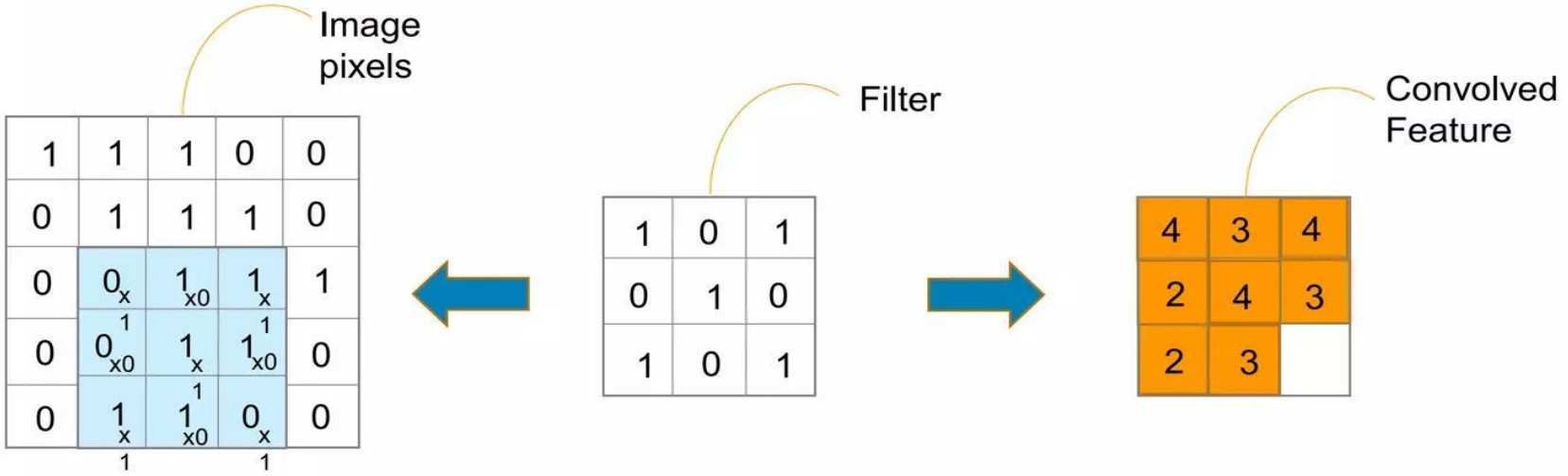


Sliding the filter matrix over the image and computing the dot product to detect patterns

Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 5 image whose pixel values are only 0 and 1

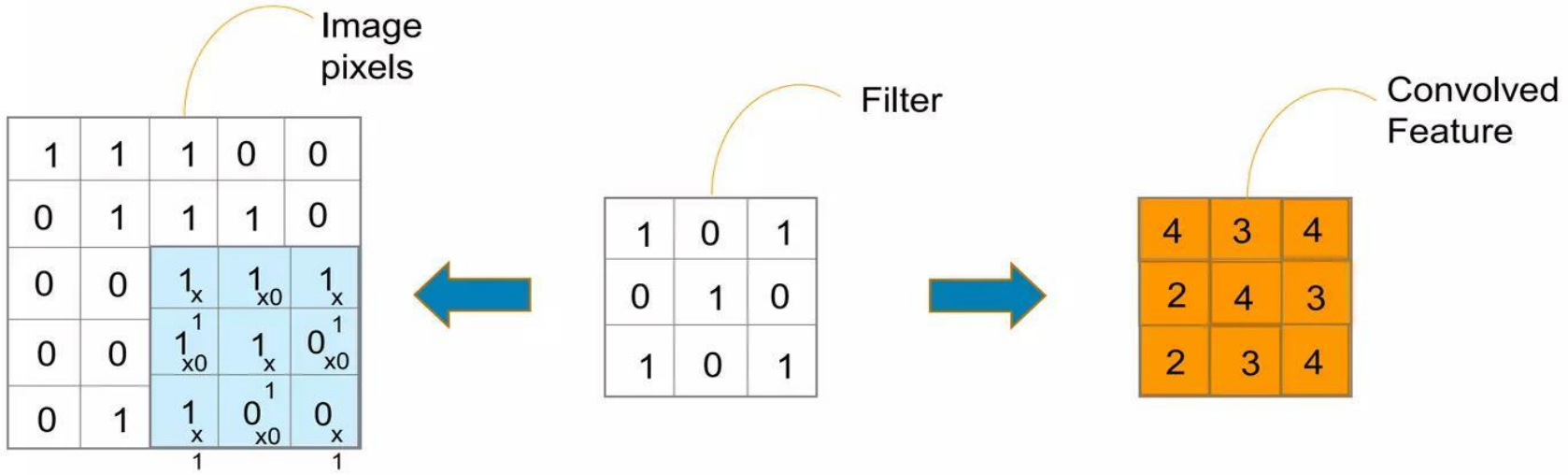


Sliding the filter matrix over the image and computing the dot product to detect patterns

Convolution Layer

A Convolution Layer has a number of filters that perform convolution operation

Every image is considered as a matrix of pixel values.
Consider the following 5 5 image whose pixel values are only 0 and 1



Sliding the filter matrix over the image and computing the dot product to detect patterns

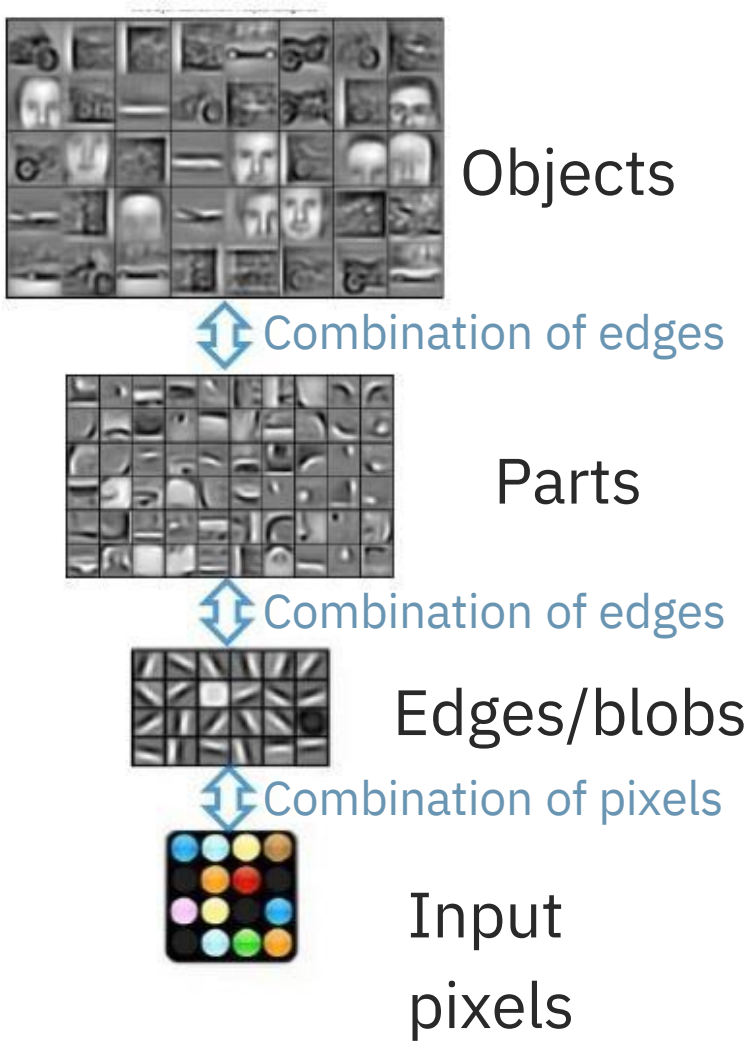
Convolutional Neural Network

Multiple convolutional layers

➡ Allows the network to learn combinations of sub-parts, to increase complexity

but how to encourage abstraction and summarization?

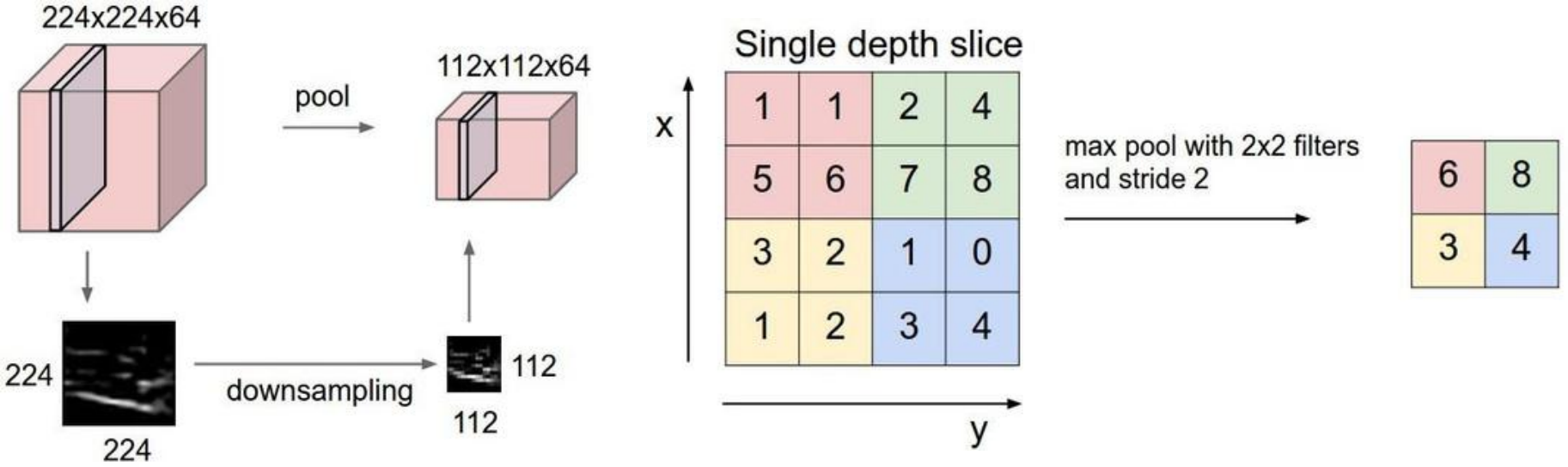
Answer: Pooling layers



Pooling Layer

Response map subsampling:

Allows summarization of the responses

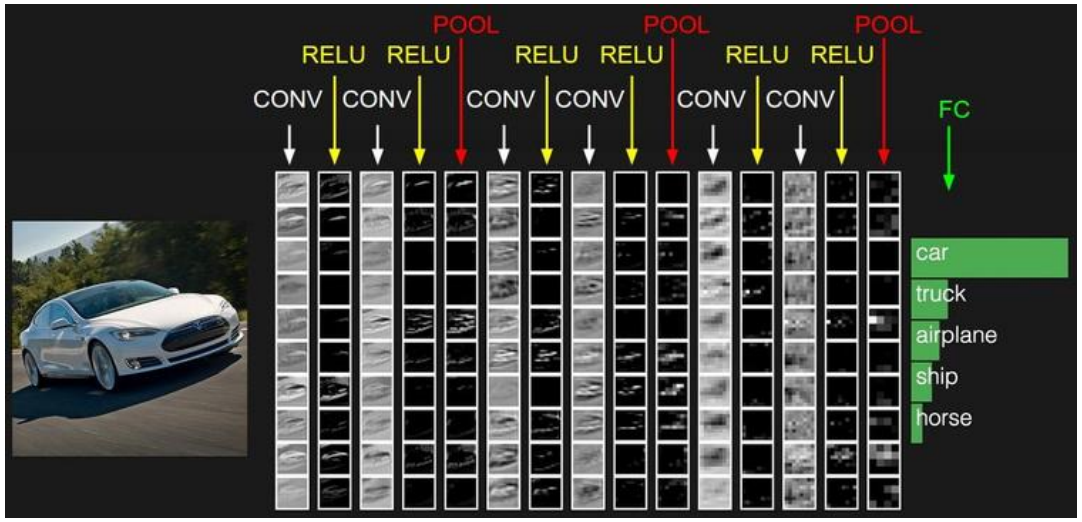


Common architectures

Repeat several times:

- Start with a convolutional layer
  Followed by non-linear activation and pooling
-

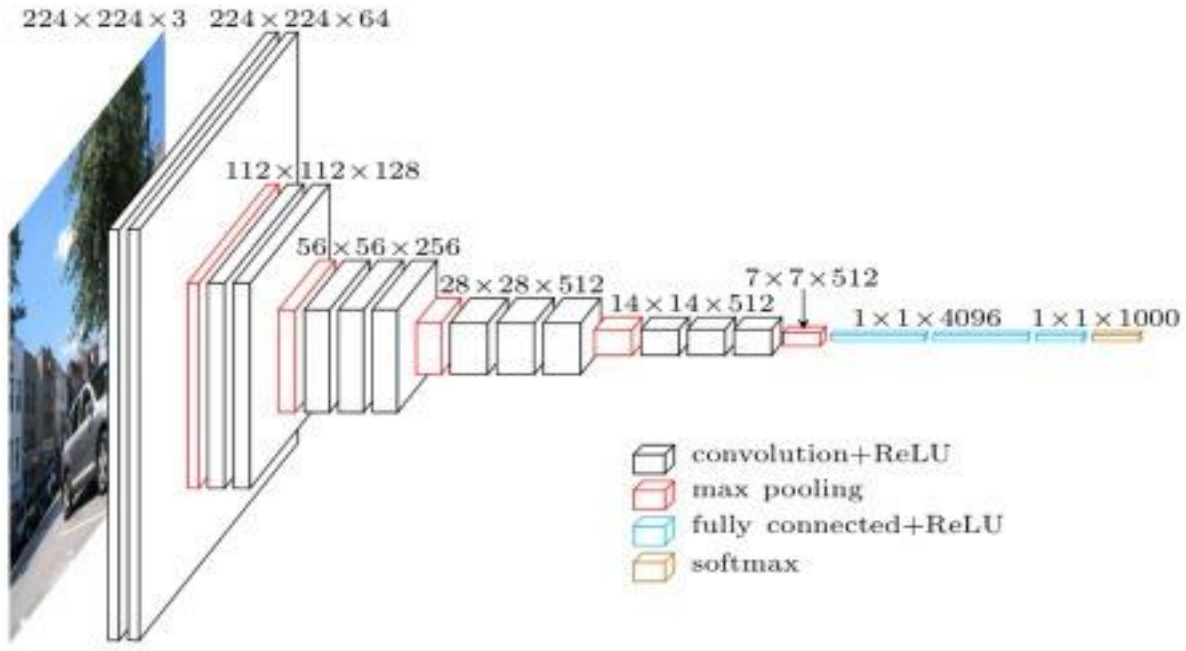
End with a fully connected (MLP) layer



Example: VGGNetmodel

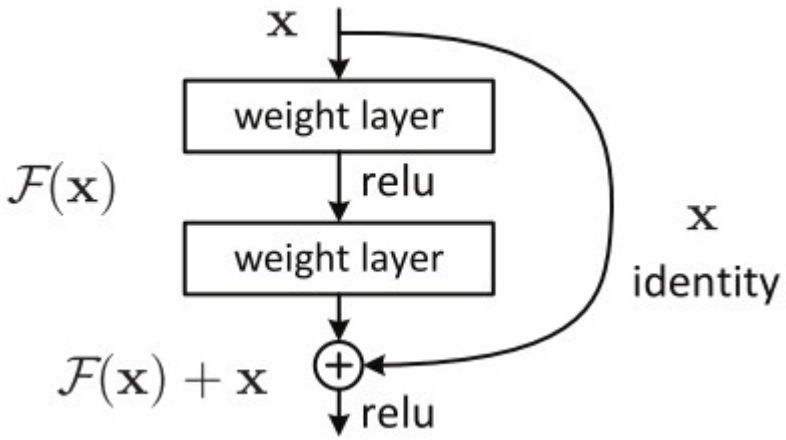
Used for object classification task

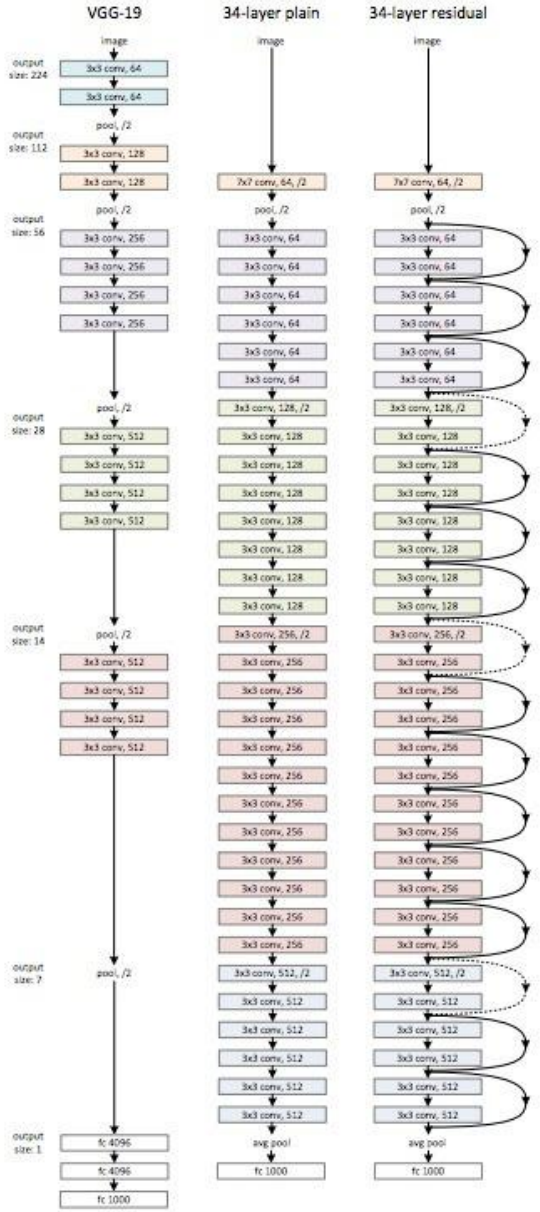
- 1000-way classification
- task 138 million parameters



Residual Networks (ResNet)

Adding residual connections





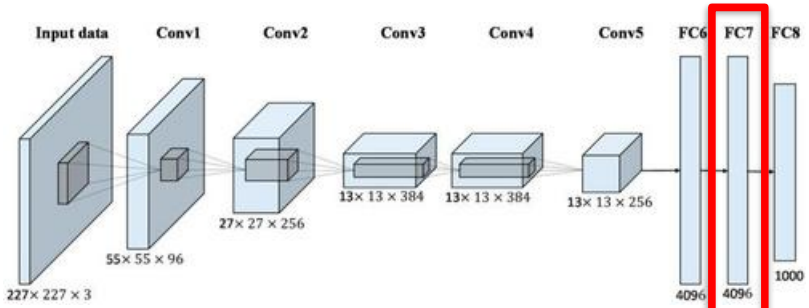
ResNet(He et al., 2015)

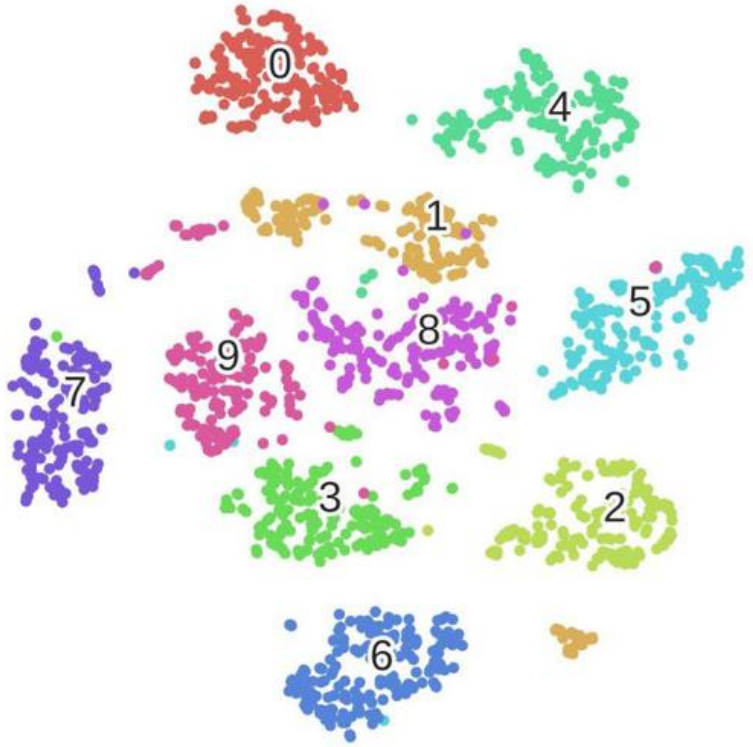
- Up to 152 layers!

Visualizing CNNs

Visualizing the Last CNN Layer: t-sne

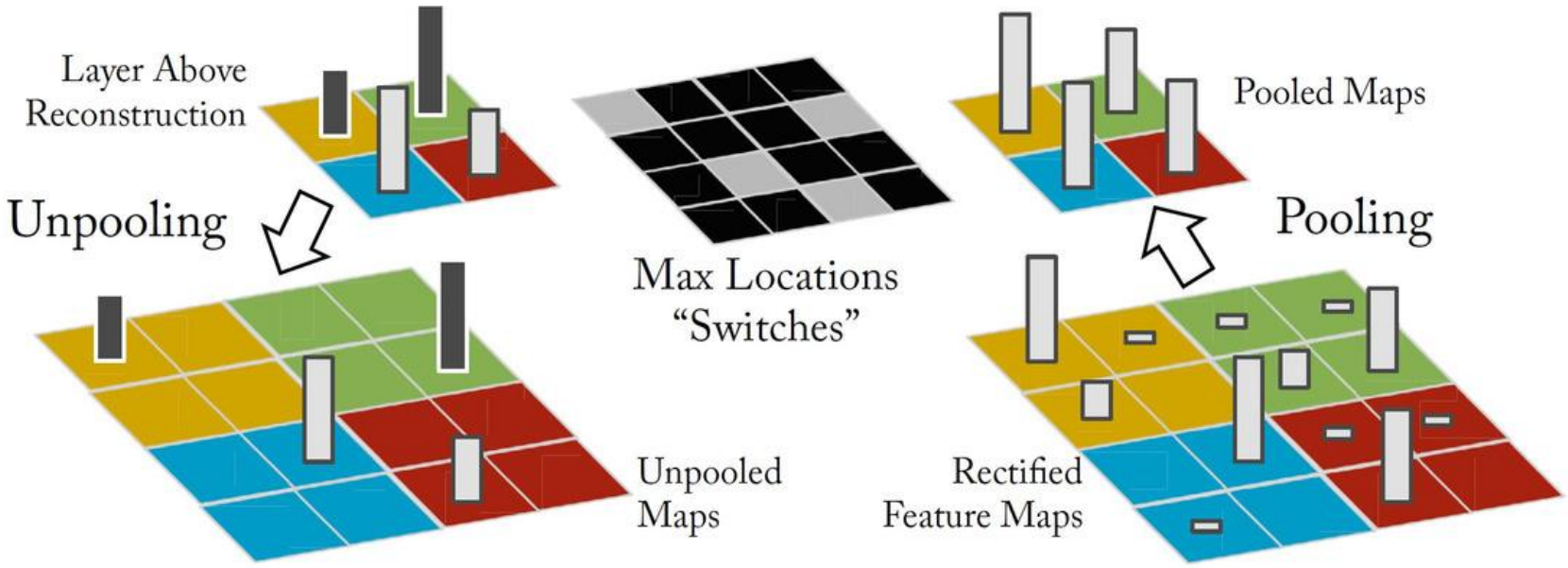
Alex Net



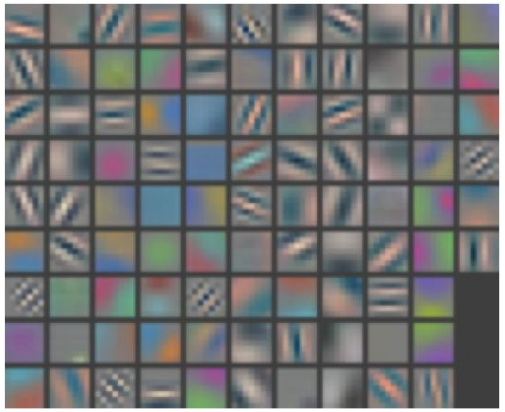


Embed high dimensional data points (i.e.feature codes) so that pairwise distances are conserved in local neighborhoods.

Deconvolution



Visualizing & Understanding Conv. Nets



- What Makes Convnets “Tick”?
- What happens in hidden units?
- Layer 1: easy to visualize
- Deeper layers: just a bunch of numbers? Or something more meaningful?
- Do convnets use context or actually model target classes

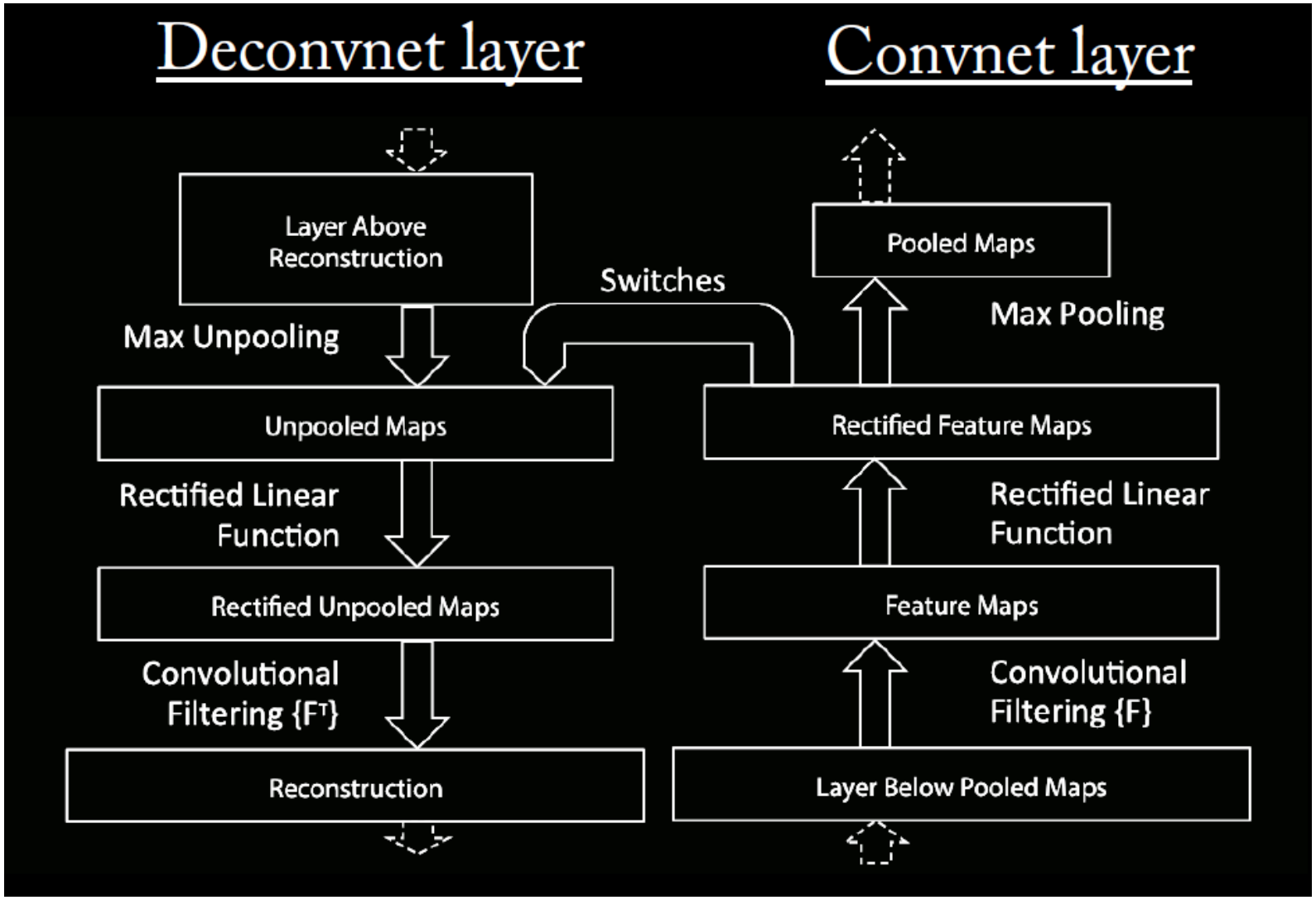
Introducing: Visualizing & Understanding Conv. Nets

- Zeiler & Fergus, 2013
- ***Goal***: Try to visualize the “black box” hidden units, gain insights
- ***Hope*****:** Use conclusions to improve performance
- ***Idea***: “Deconvolutional” neural net

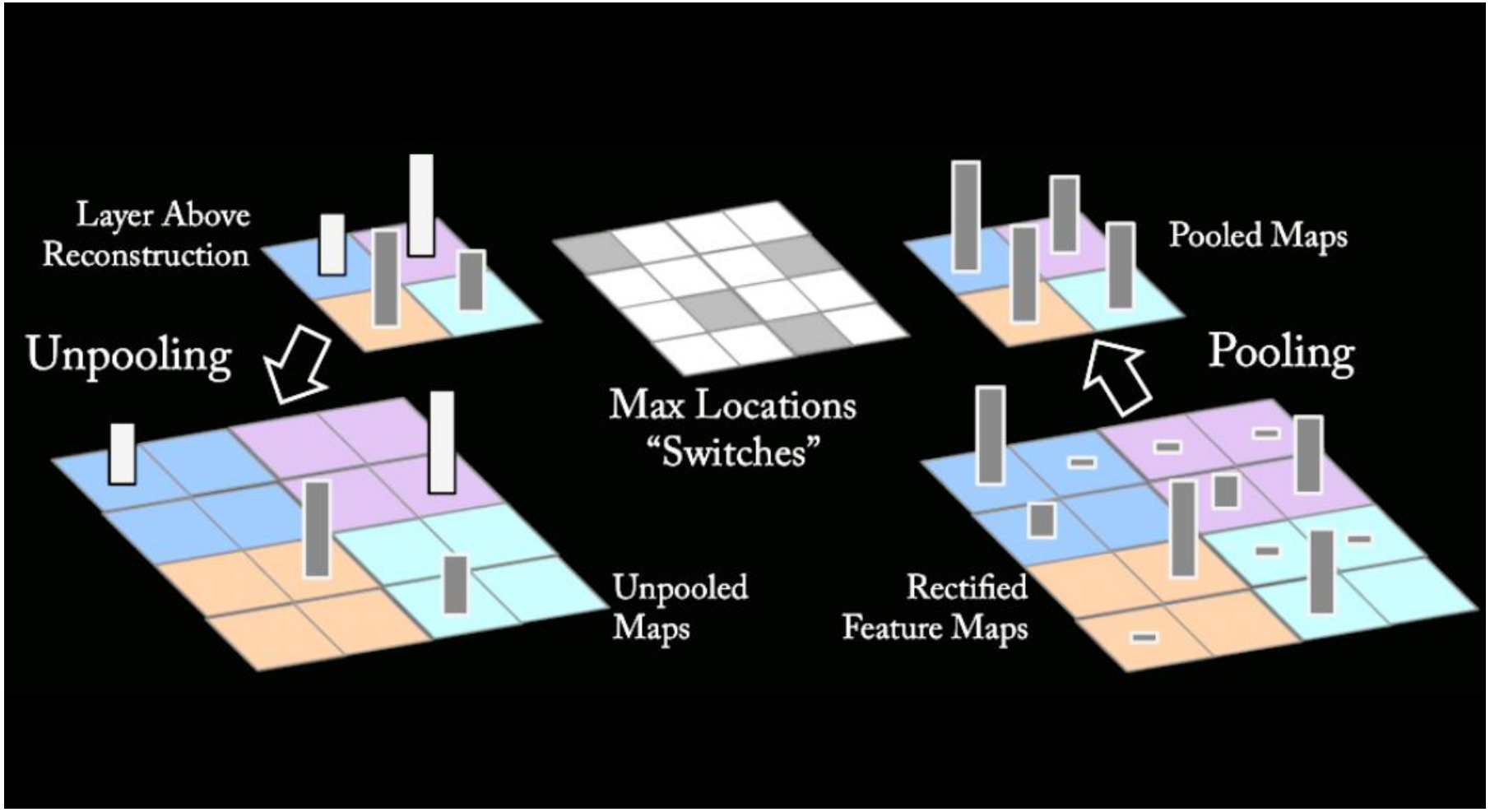
Deconvolutional Nets

- Originally suggested for unsupervised feature learning : construct a convolutional net, cost function is image reconstruction error
- Used here to find what stimuli causes strongest responses in hidden units
- Run many images through net → find strongest unit activations in each layer → visualize by “reversing” net operation

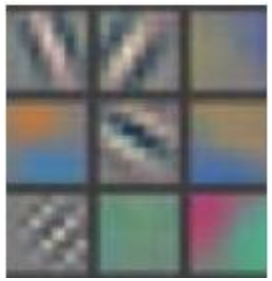
Reversing a convent

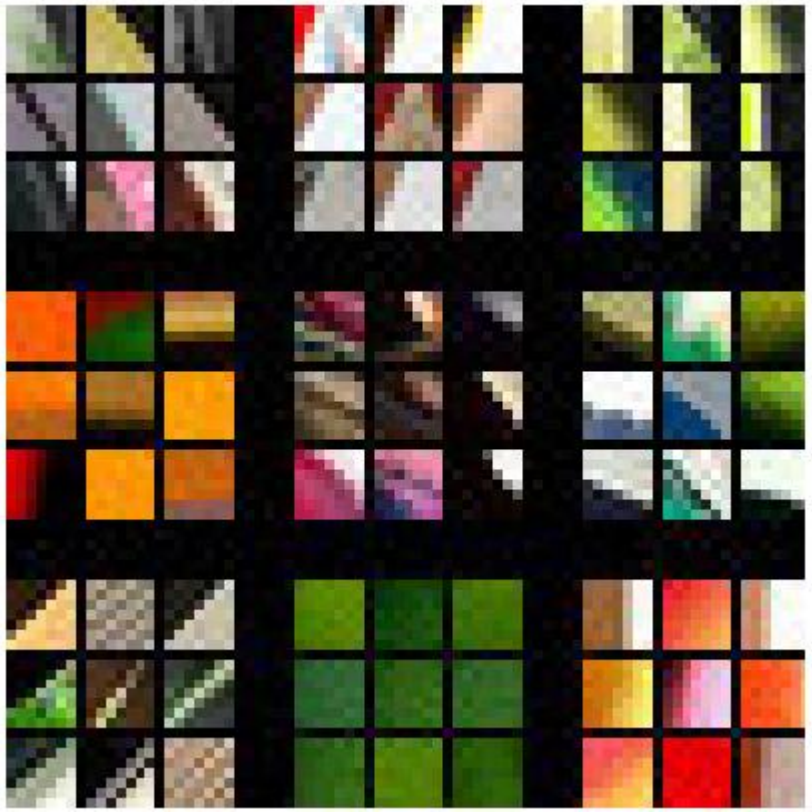


“Unpooling”

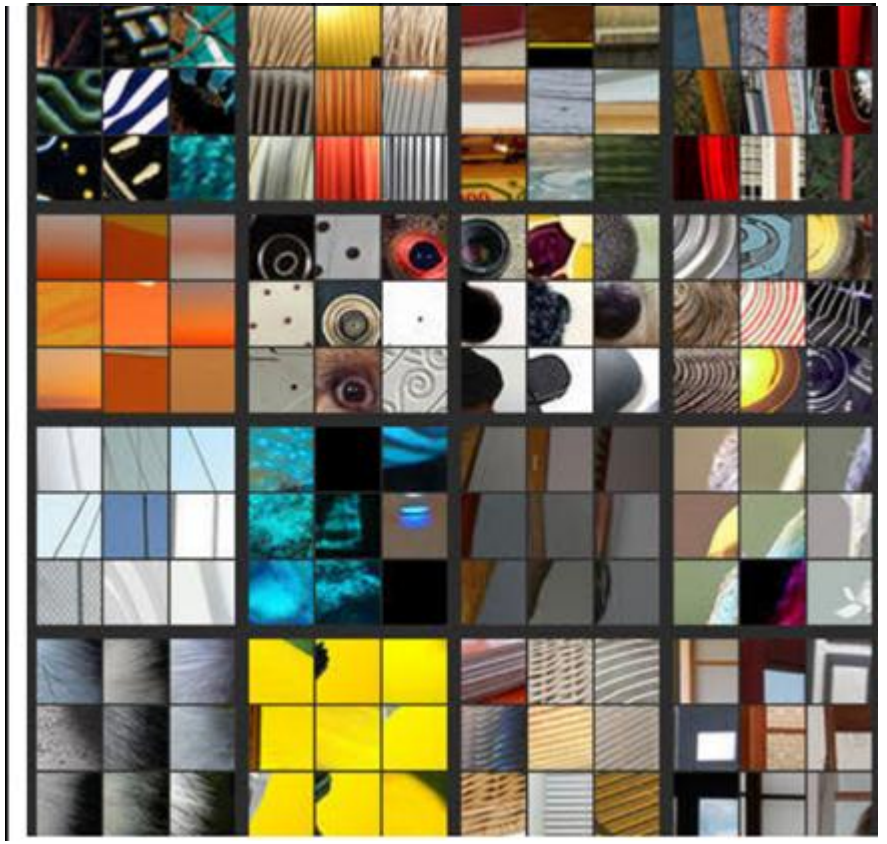

Layer Above
Reconstruction
Unpooling
Max Locations
“Switches”
Pooled Maps
Pooling
Unpooled
Maps
Rectified
Feature Maps

Layer 1:

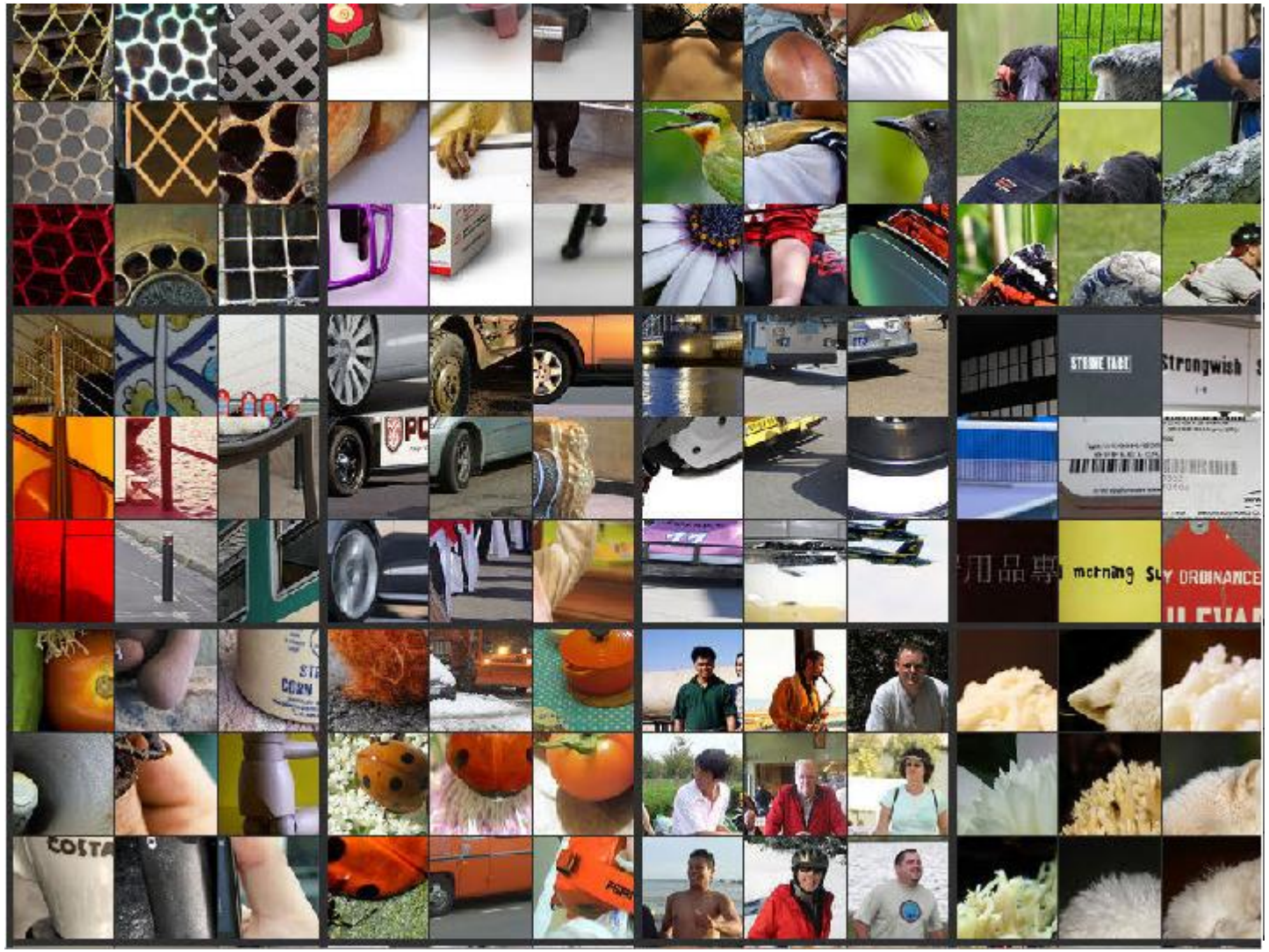




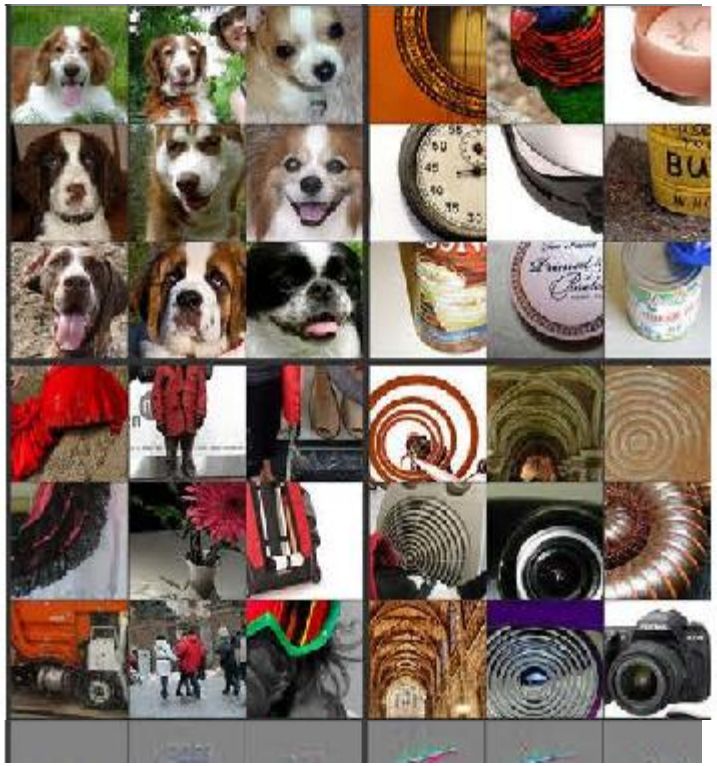
Hidden Layer Visualizations: layer 2



Hidden Layer Visualizations: Layer 3

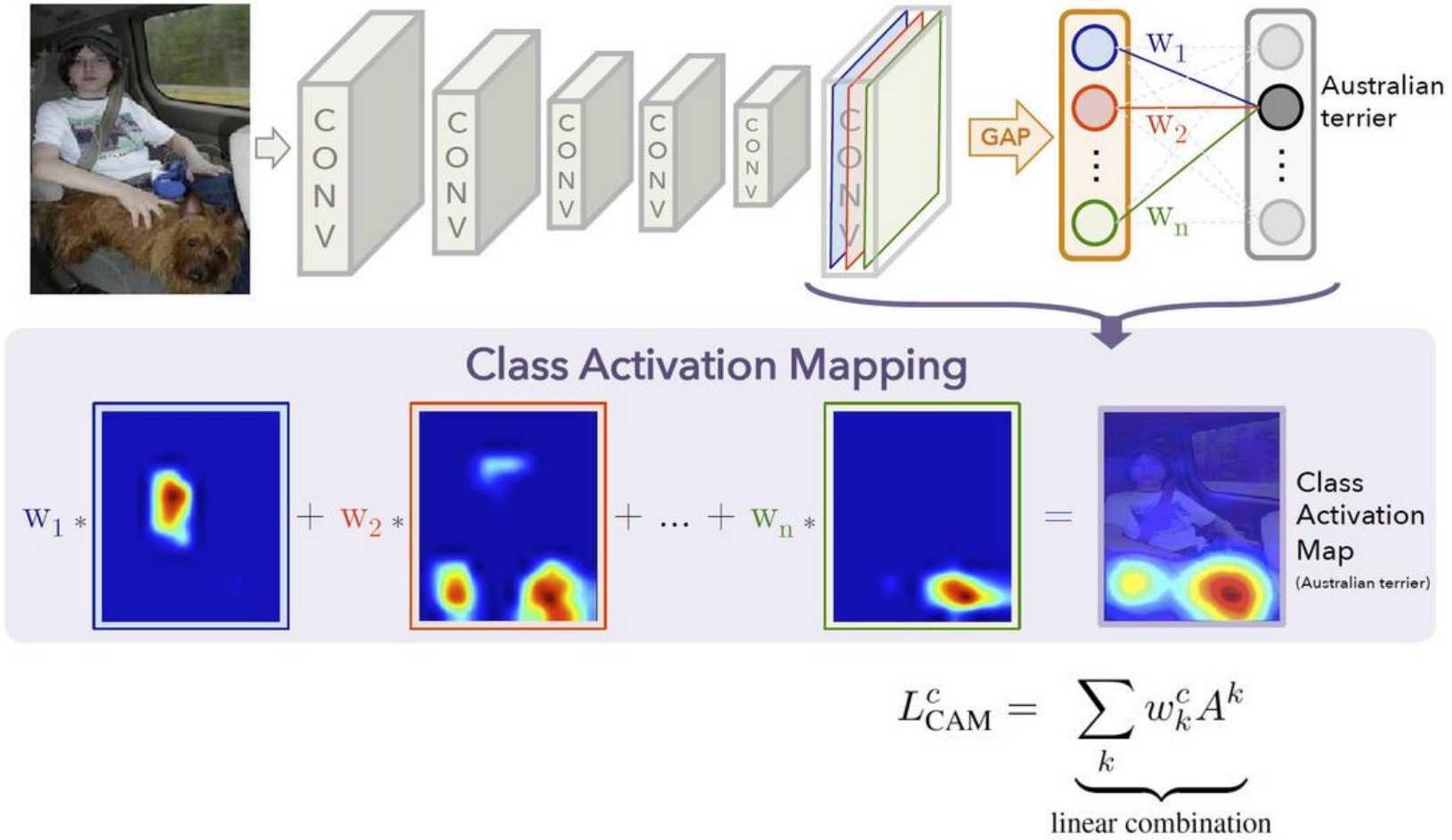


Hidden Layer Visualizations: Layer 4



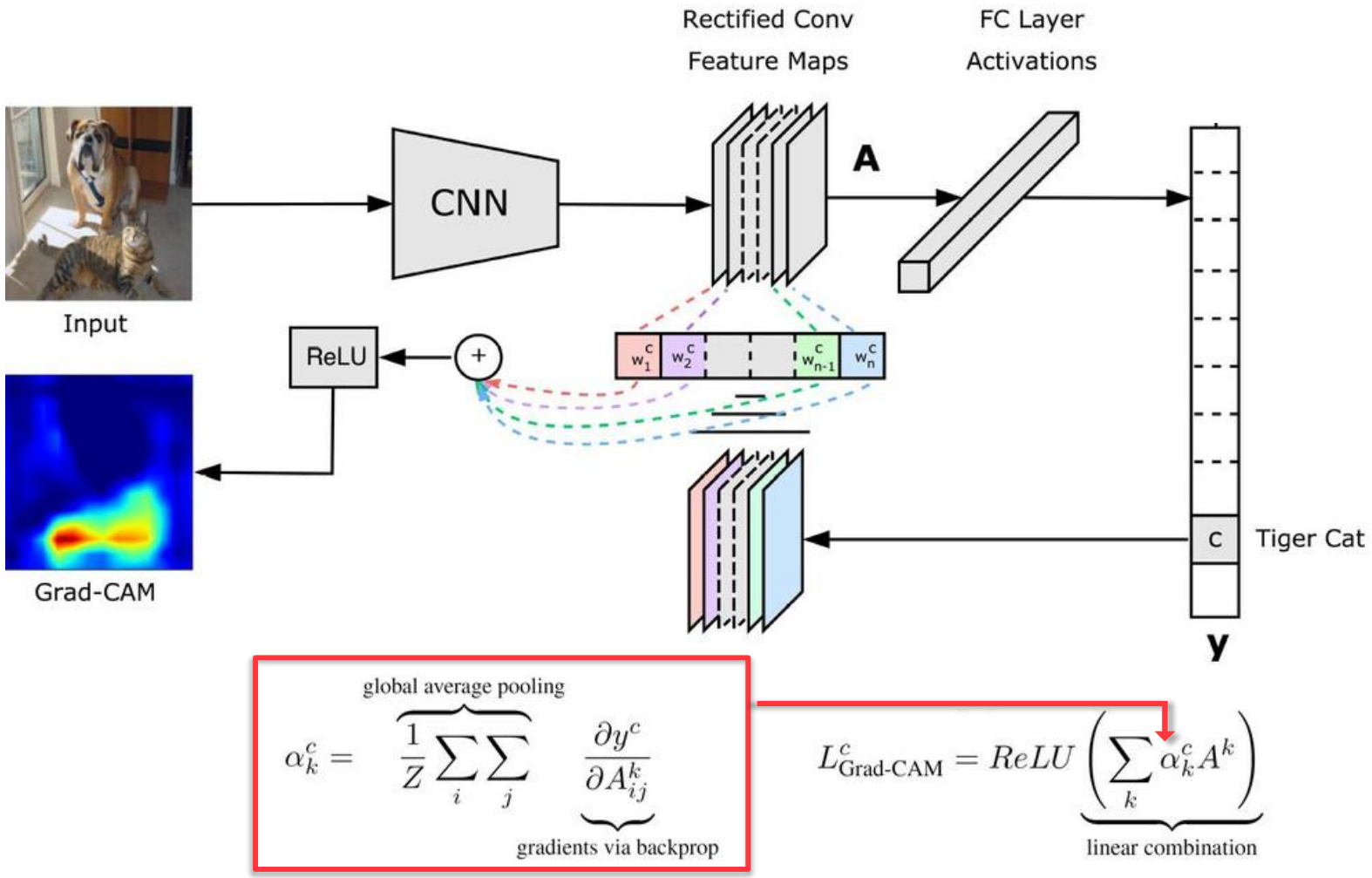
Hidden Layer Visualizations: Layer 5

CAM: Class Activation Mapping [CVPR 2016]



$$L^c_{\text{CAM}} = \underbrace{\sum_k w^c_k A^k}_{\text{linear combination}}$$

Grad-CAM [ICCV 2017]



$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

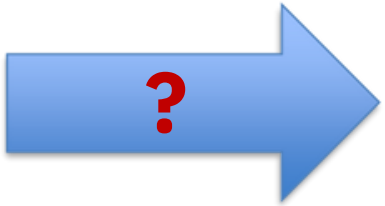
$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

Region-based CNNs

Object recognition

Object Detection (and Segmentation)

Input image



Detected Objects

One option: Sliding window

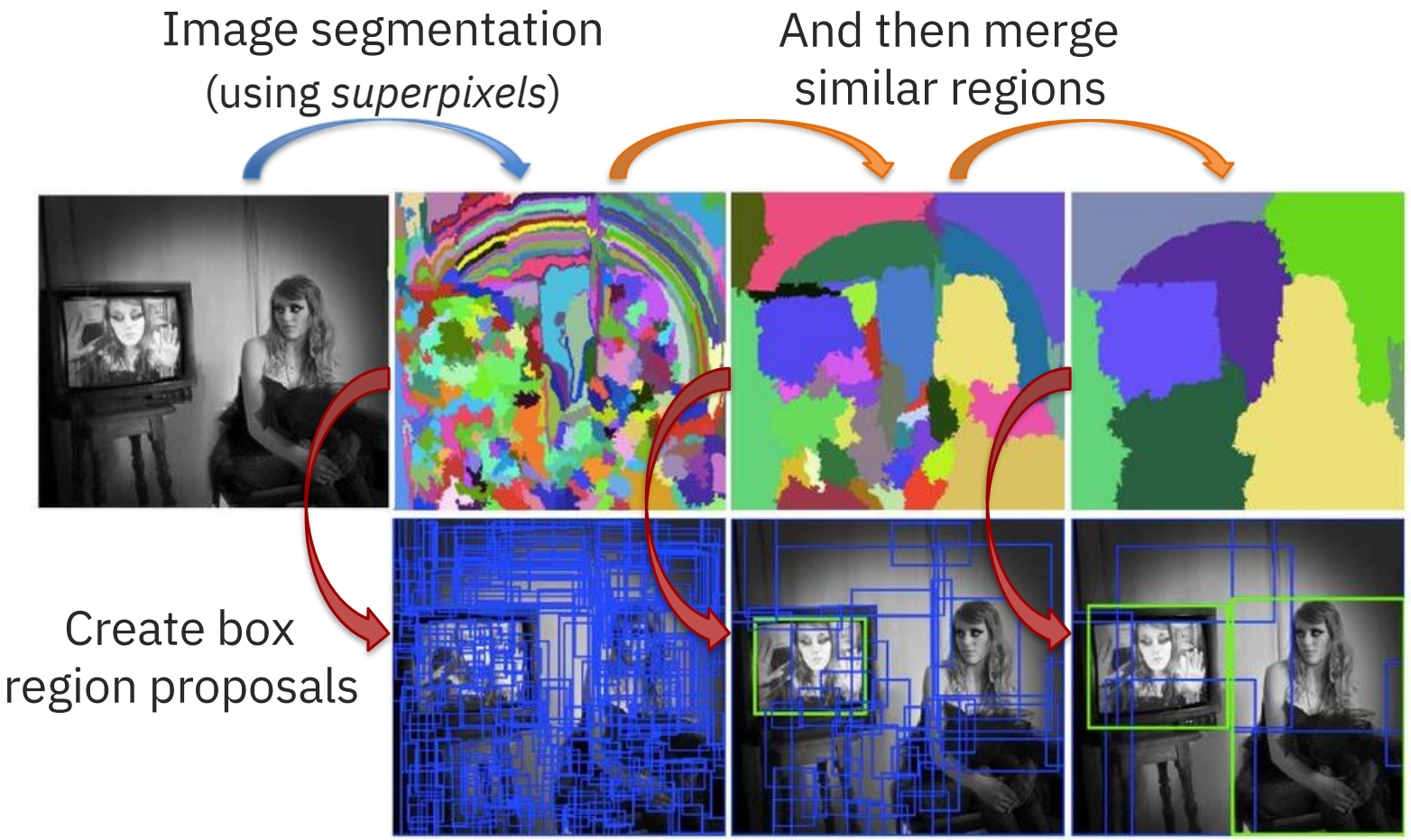
Object Detection (and Segmentation)

Input image Region Proposals Detected Objects

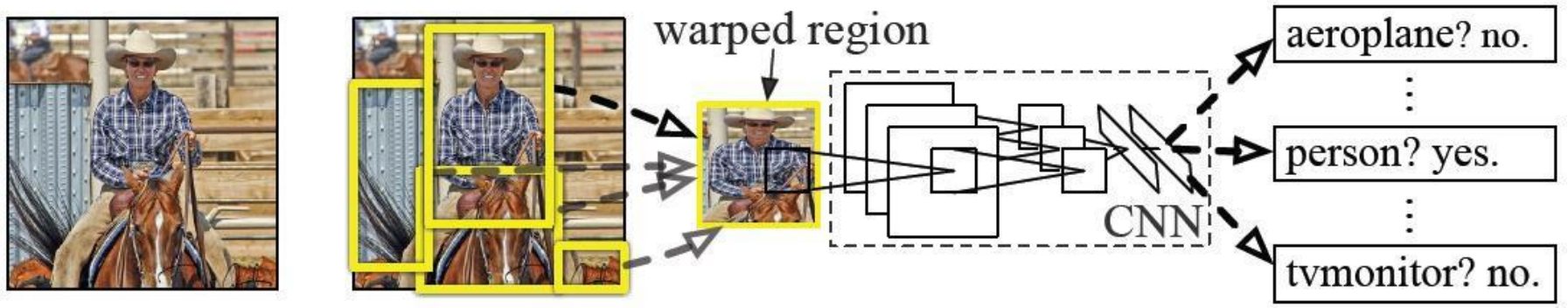
A better option: Start by Identifying hundreds of region proposals and then apply our CNN object detector

How to efficiently identify region proposals?

Selective Search [Uijlings et al., IJCV 2013]



R-CNN [Girshicket al., CVPR 2014]



- Select ~2000 region proposals → Time consuming!
- Warp each region Apply CNN to
- each region → Time consuming!

Fast R-CNN: Applies CNN only once, and then extracts regions

Faster R-CNN: Region selection on the Conv5 response map

Mask R-CNN: Detection and Segmentation

(He et al., 2018)

Sequential Modeling with Convolutional Networks

Modeling Temporal and Sequential Data

How to represent a video sequence?

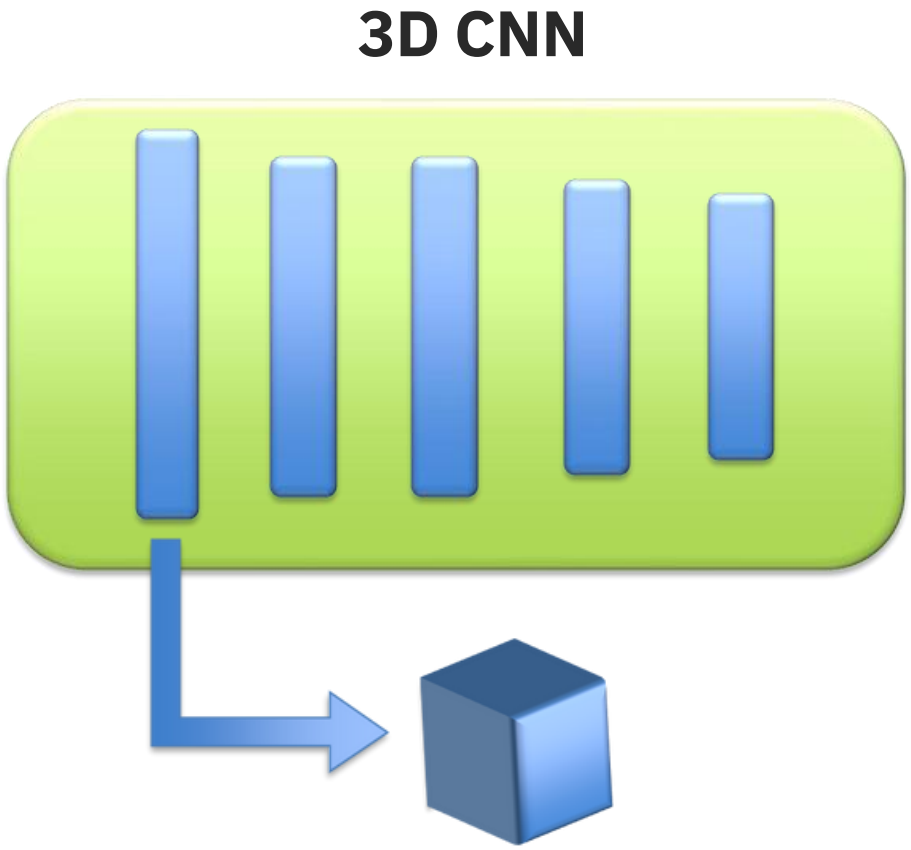
One option: Recurrent Neural Networks

(more about this next week)

3D CNN

Input as a 3D tensor

(stacking video images)



First layer with 3D kernels

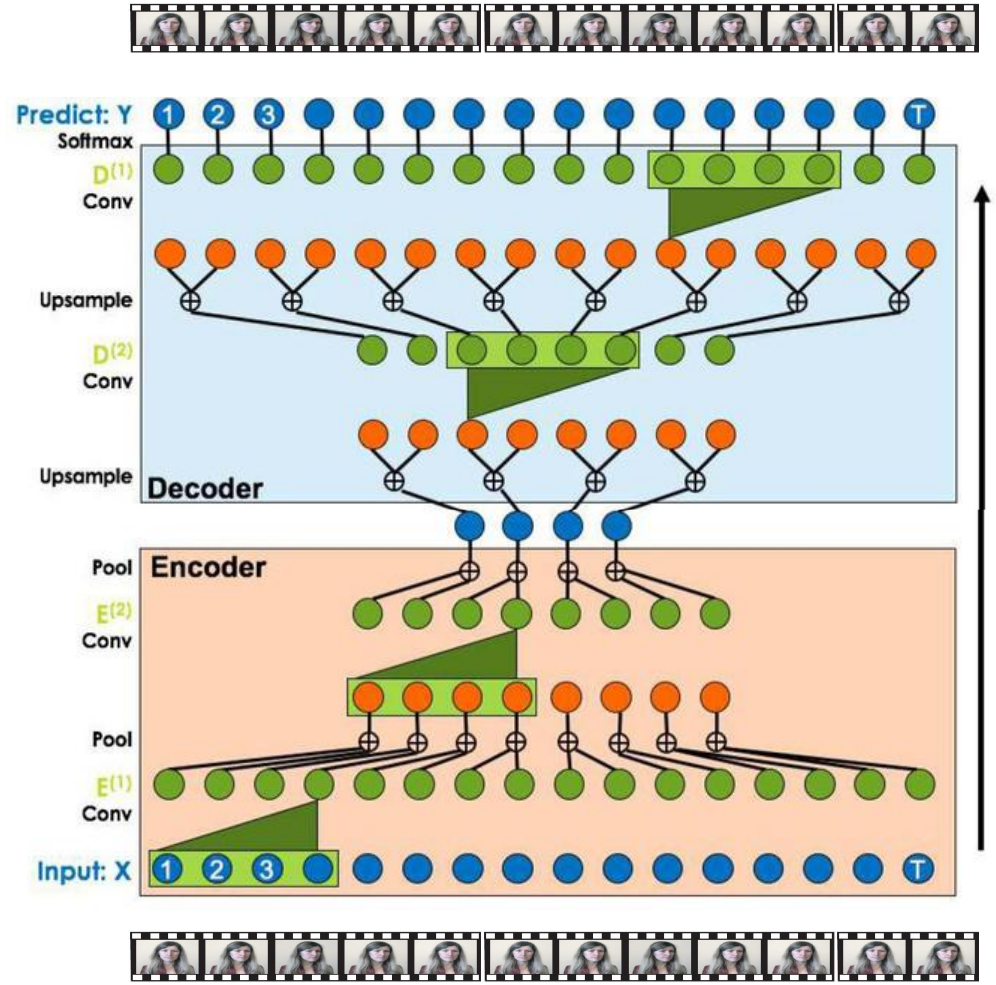
Time-Delay Neural Network

Alexander Waibel, Phoneme Recognition Using Time-Delay Neural Networks, SP87-100, Meeting of the Institute of Electrical, Information and Communication Engineers (IEICE), December,1987,Tokyo, Japan.

Temporal Convolution Network (TCN) [Lea et al., CVPR 2017]

Decoder

Encoder



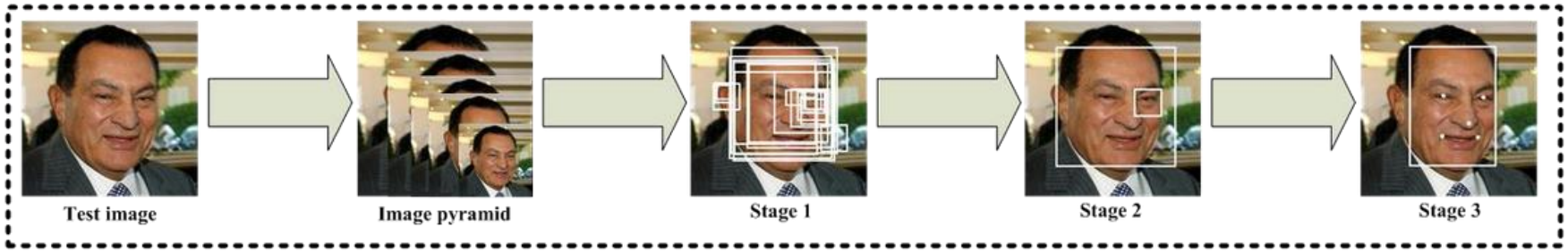
Appendix: Tools for Automatic visual behavior analysis

Automatic analysis of visual behavior

-
  - Face detection
  - Face tracking
    - Facial landmark detecion
- Head pose
- Eye gaze tracking
- Facial expression analysis
- Body pose tracking

Face Detection –Multi-Task CNN

[SPL 2016]



Stage 1: candidate windows are produced through a fast Proposal Network

Stage 2: refine these candidates through a Refinement Network Stage 3:

produces final bounding box and facial landmarks position

Existing software (face detection)

Multi-Task CNN face detector

https://kpzhang93.github.io/MTCNN_face_detection_alignment/index.html

OpenCV (Viola-Jones detector)

dlib(HOG + SVM)

http://dlib.net/

Tree based model (accurate but very slow)

http://www.ics.uci.edu/~xzhu/face/

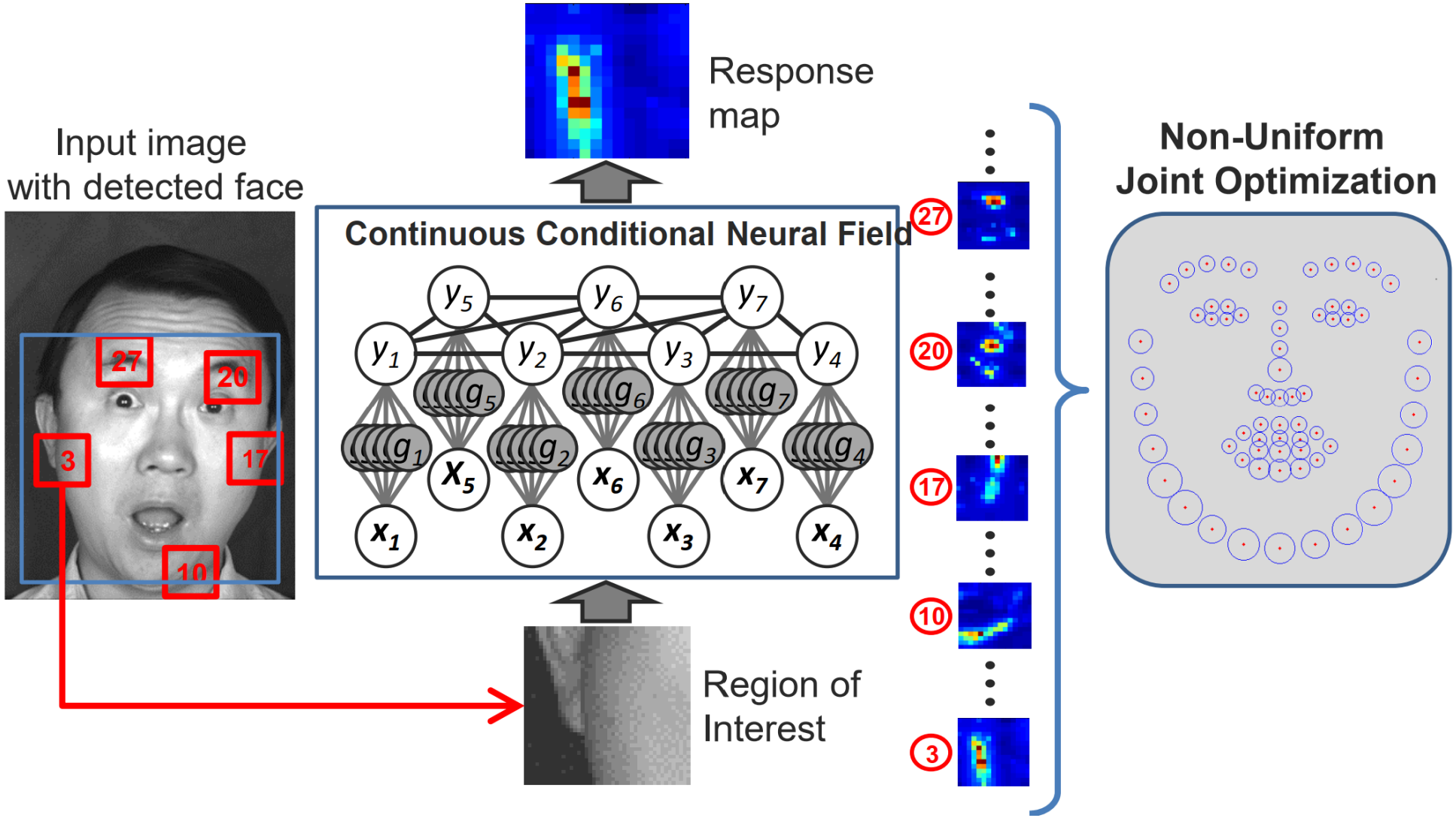
HeadHunter(accurate but slow)

http://markusmathias.bitbucket.org/2014_eccv_face_detection/

NPD

http://www.cbsr.ia.ac.cn/users/scliao/projects/npdface/

Facial Landmarks: Constrained Local Neural Field

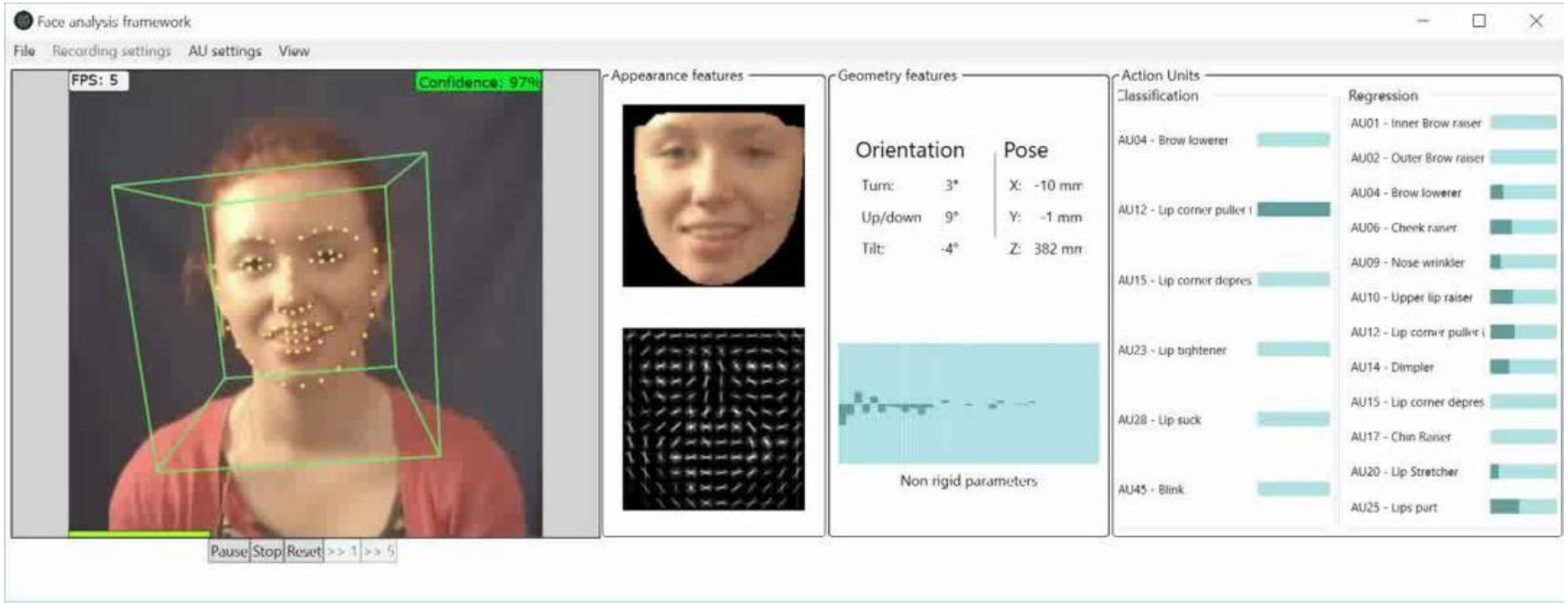


Existing software (facial landmarks)

- OpenFace: facial features
  https://github.com/TadasBaltrusaitis/OpenFace
- Chehraface tracking
  https://sites.google.com/site/chehrahome/
- Menpoproject (good AAM, CLM learning tool)
  http://www.menpo.org/
- IntraFace: Facial attributes, facial expression analysis
  http://www.humansensing.cs.cmu.edu/intraface/
- OKAO Vision: Gaze estimation, facial expression
  http://www.omron.com/ecb/products/mobile/okao03.html (Commercial software)

VisageSDK
http://www.visagetechnologies.com/products/visagesdk/ (Commercial software)

Facial expression analysis



[**OpenFace: an open source facial behavior analysis toolkit**, T. Baltrušaitiset al., 2016]

Existing Software (expression analysis)

OpenFace: Action Units

https://github.com/TadasBaltrusaitis/OpenFace

Shore: facial tracking, smile detection, age and gender detection

http://www.iis.fraunhofer.de/en/bf/bsy/fue/isyst/detektion/

FACET/CERT (EmotientAPI): Facial expression recognition

http://imotionsglobal.com/software/add-on-modules/attention-tool-facet-module-facial-action-coding-system-facs/(Commercial software)

Affdex

http://www.affectiva.com/solutions/apis-sdks/

(commercial software)

Gaze Estimation –Eye, Head and Body

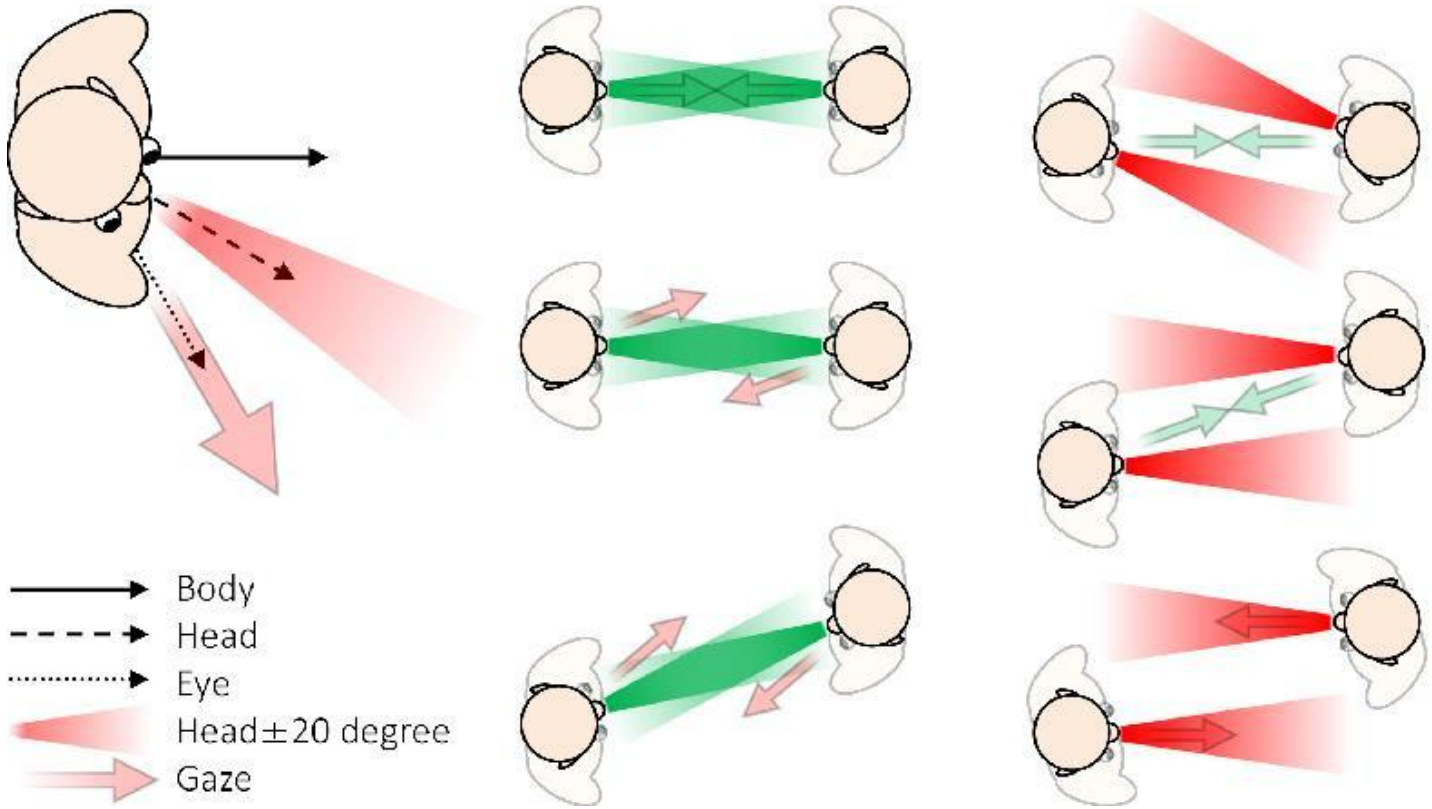
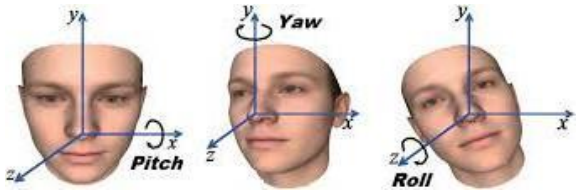


Image from Hachisuet al (2018). FaceLooks: A Smart Headband for Signaling Face-to-Face Behavior.*Sensors*.

Existing Software (head gaze)



OpenFace

https://github.com/TadasBaltrusaitis/OpenFace

Chehraface tracking

https://sites.google.com/site/chehrahome/

Watson: head pose estimation

http://sourceforge.net/projects/watson/

Random forests

http://www.vision.ee.ethz.ch/~gfanelli/head_pose/head_forest.html
(requires a Kinect)

IntraFace

http://www.humansensing.cs.cmu.edu/intraface/

Existing Software (eye gaze)

OpenFace: gaze from a webcam

https://github.com/TadasBaltrusaitis/OpenFace

EyeAPI: eye pupil detection

http://staff.science.uva.nl/~rvalenti/

EyeTab

https://www.cl.cam.ac.uk/research/rainbow/projects/eyetab/

OKAO Vision: Gaze estimation, facial expression

http://www.omron.com/ecb/products/mobile/okao03.html (Commercial software)

Articulated Body Tracking: OpenPose

Existing Software (body tracking)

- OpenPose
  - https://github.com/CMU-Perceptual-Computing-Lab/openpose
- Microsoft Kinect
  - http://www.microsoft.com/en-us/kinectforwindows/
- OpenNI
  - http://openni.org/
- Convolutional Pose Machines
  - https://github.com/shihenw/convolutional-pose-machines-release

Visual Descriptors

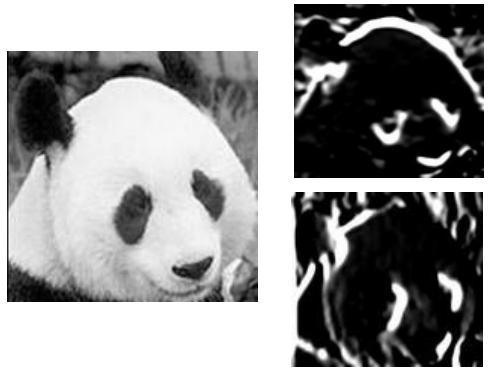
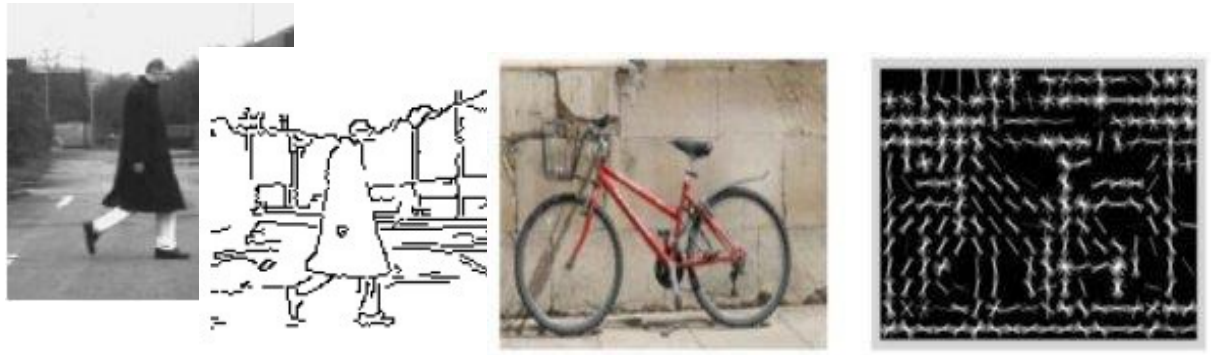


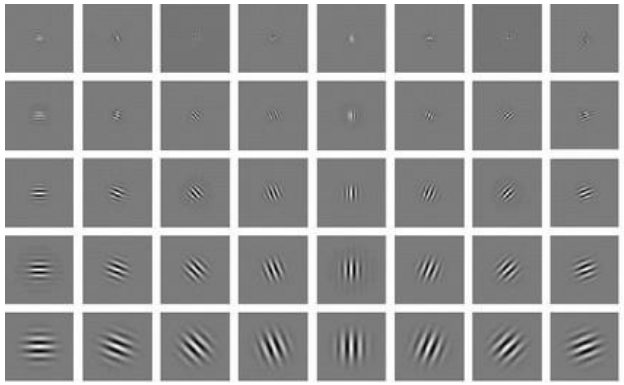
Image gradient



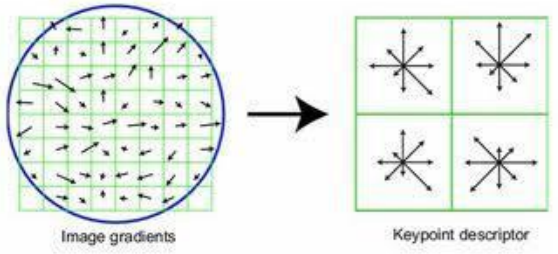
Edge detection

Histograms of Oriented Gradients

Optical Flow



Gabor Jets



SIFT descriptors

Existing Software (visual descriptors)

- OpenCV: optical flow, gradient, Haarfilters…

  SIFT descriptors
  - http://blogs.oregonstate.edu/hess/code/sift/
- dlib – HoG
  - http://dlib.net/
- OpenFace: Aligned HoGfor faces
  - https://github.com/TadasBaltrusaitis/CLM-framework