28.03.2020

Statistical Methods in AI (CSE/ECE 471)

Transfer Learning, Landscape of CNNs

Ravi Kiran (ravi.kiran@iiit.ac.in)

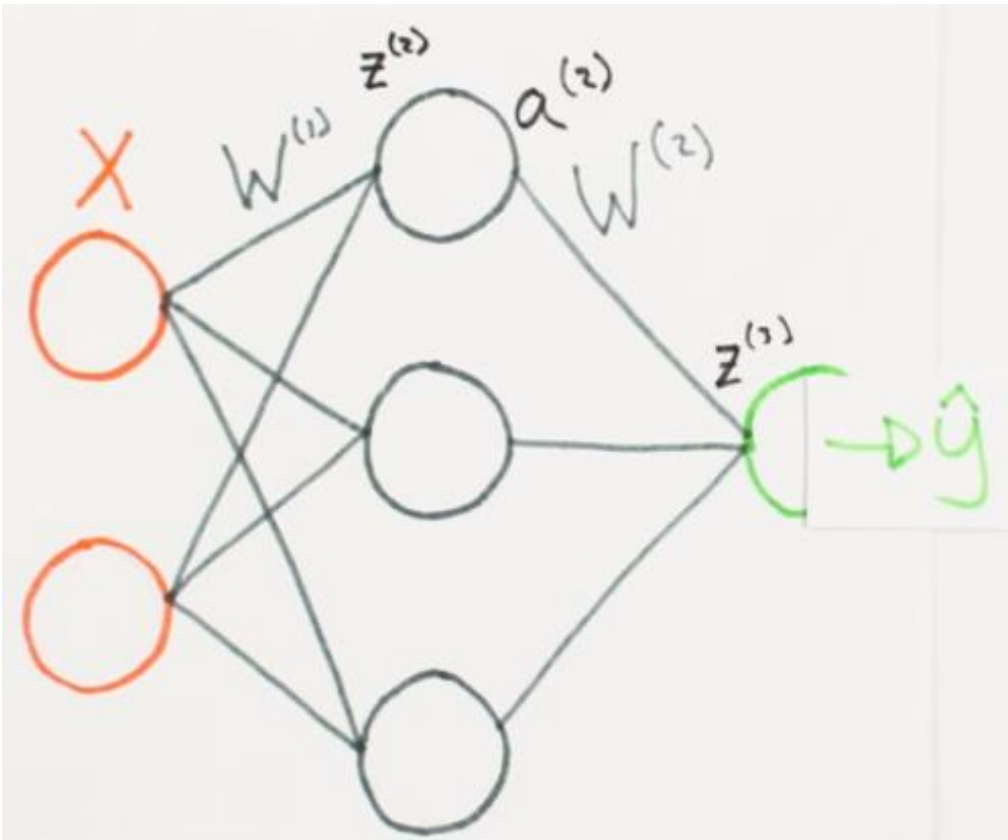
Vineet Gandhi (v.gandhi@iiit.ac.in)

Center for Visual Information Technology (CVIT)
IIIT Hyderabad

Multi-Neuron Networks

$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix}$$



$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

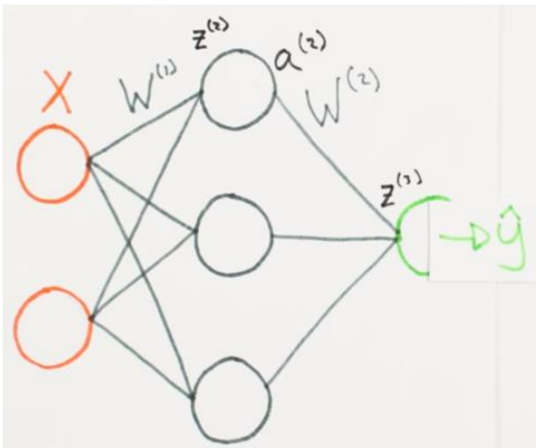
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - \hat{y})^2 \quad (5)$$

$$\boxed{J = \sum \frac{1}{2}\left(y - f\left(f\left(XW^{(1)}\right)W^{(2)}\right)\right)^2}$$

Multi-Neuron Networks :: Backpropagation



$$J = \sum \frac{1}{2}\left(y - f\left(f\left(XW^{(1)}\right)W^{(2)}\right)\right)^2$$

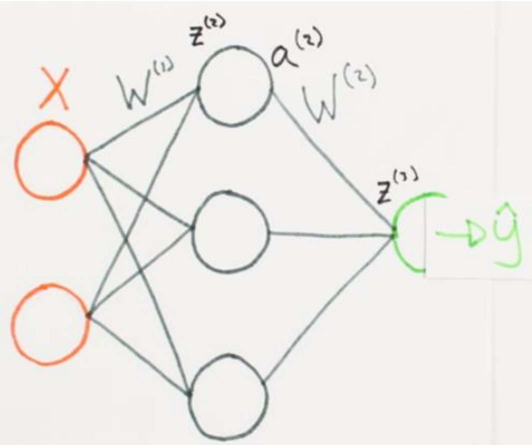
HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$

$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J / \partial W_{11}^{(2)} \\ \partial J / \partial W_{21}^{(2)} \\ \partial J / \partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation



$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J / \partial W_{11}^{(2)} \\ \partial J / \partial W_{21}^{(2)} \\ \partial J / \partial W_{31}^{(2)} \end{bmatrix}$$

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$J = \sum \frac{1}{2}\left(y - f\left(f\left(XW^{(1)}\right)W^{(2)}\right)\right)^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$

$$\frac{\partial J}{\partial W^{(2)}} = (a^{(2)})^T \delta^{(3)}$$

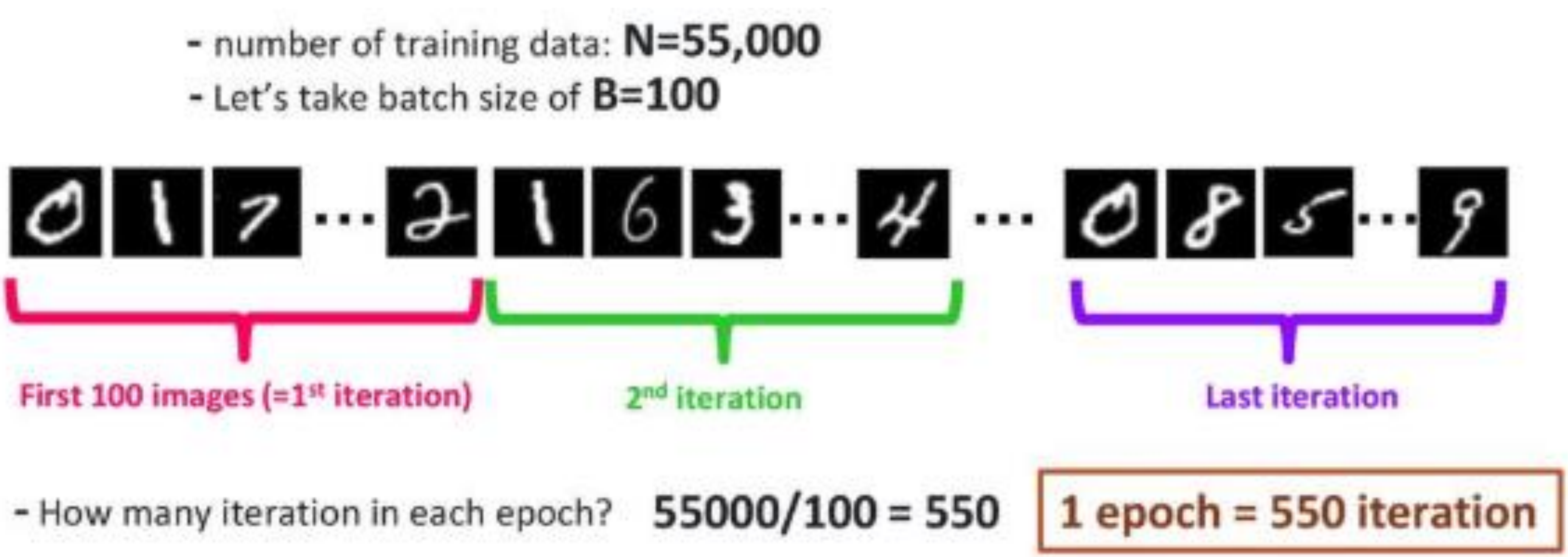
$$\delta^{(3)} = -(y - \hat{y}) f'(z^{(3)})$$

$$\frac{\partial J}{\partial W^{(1)}} = X^T \delta^{(2)}$$

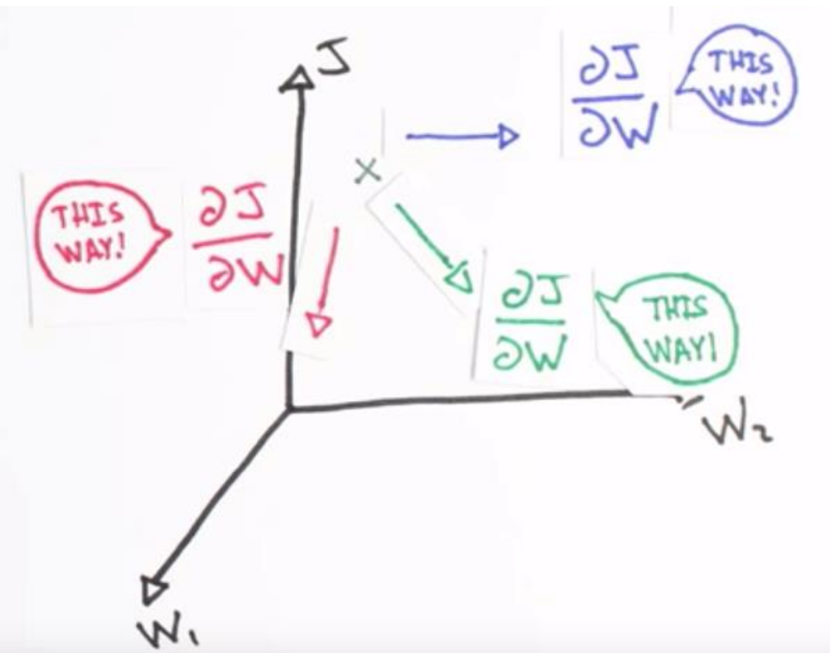
$$\delta^{(2)} = \delta^{(3)} (W^{(2)})^T f'(z^{(2)})$$

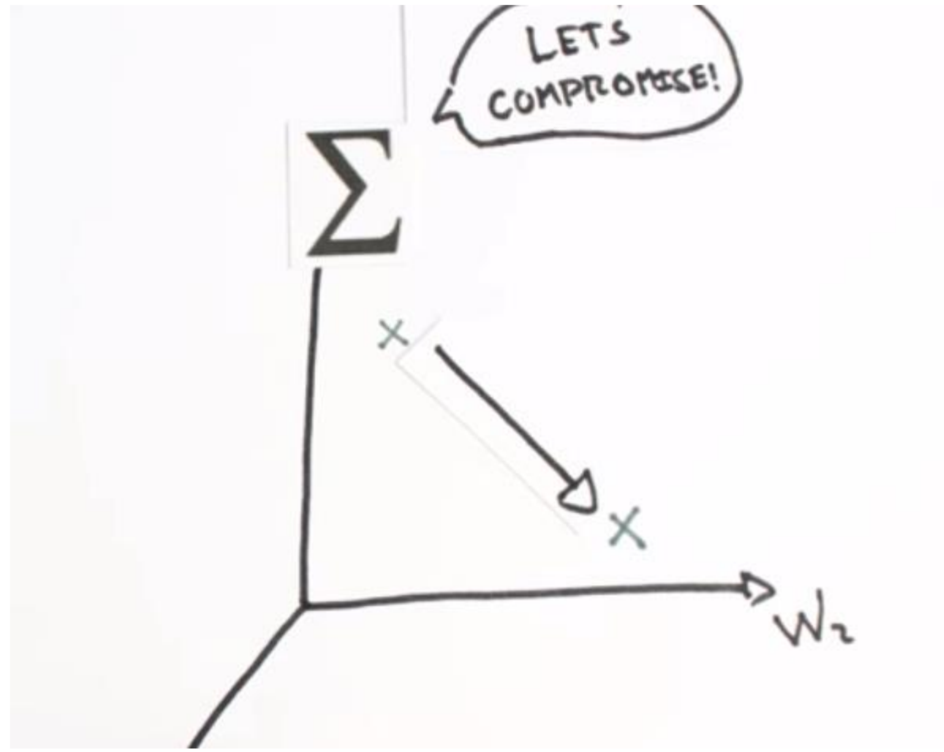
Epoch, Mini-batch, Iteration

- Mini-batch : Size of sample group being used to update weights
- Epoch = One full forward & backward pass over entire training data
- Iteration = # of forward and backward passes at mini-batch level

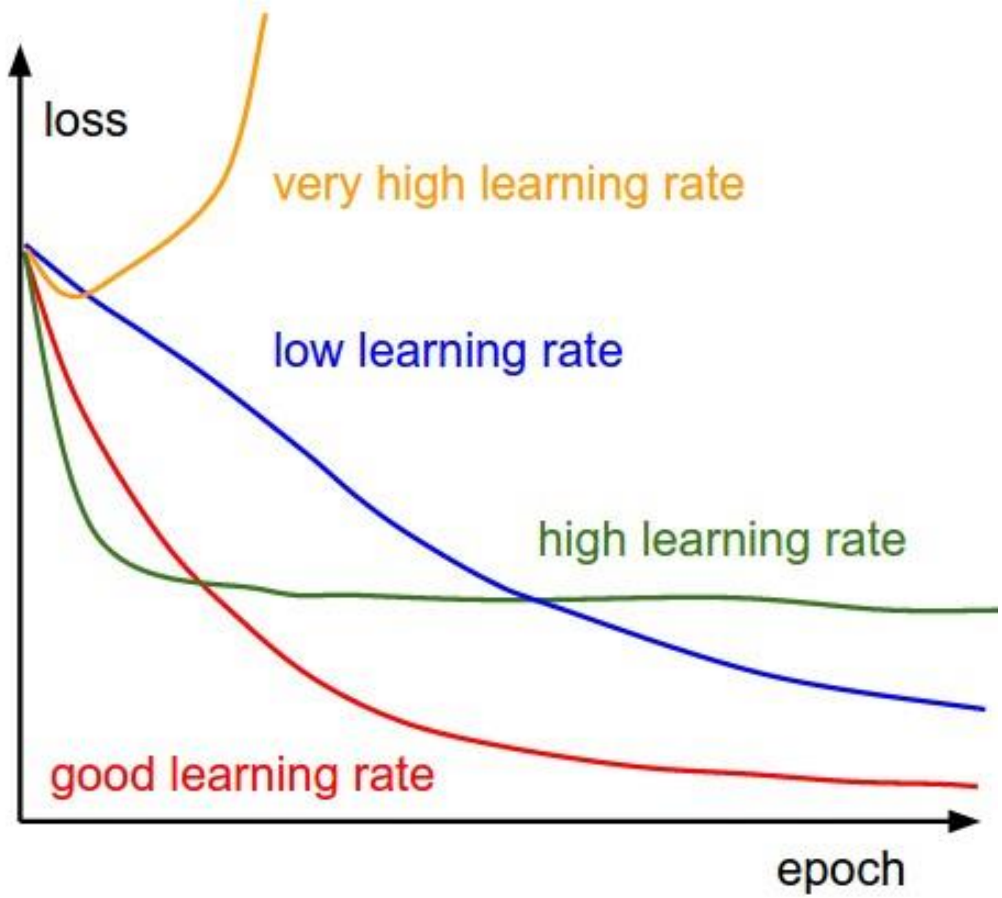


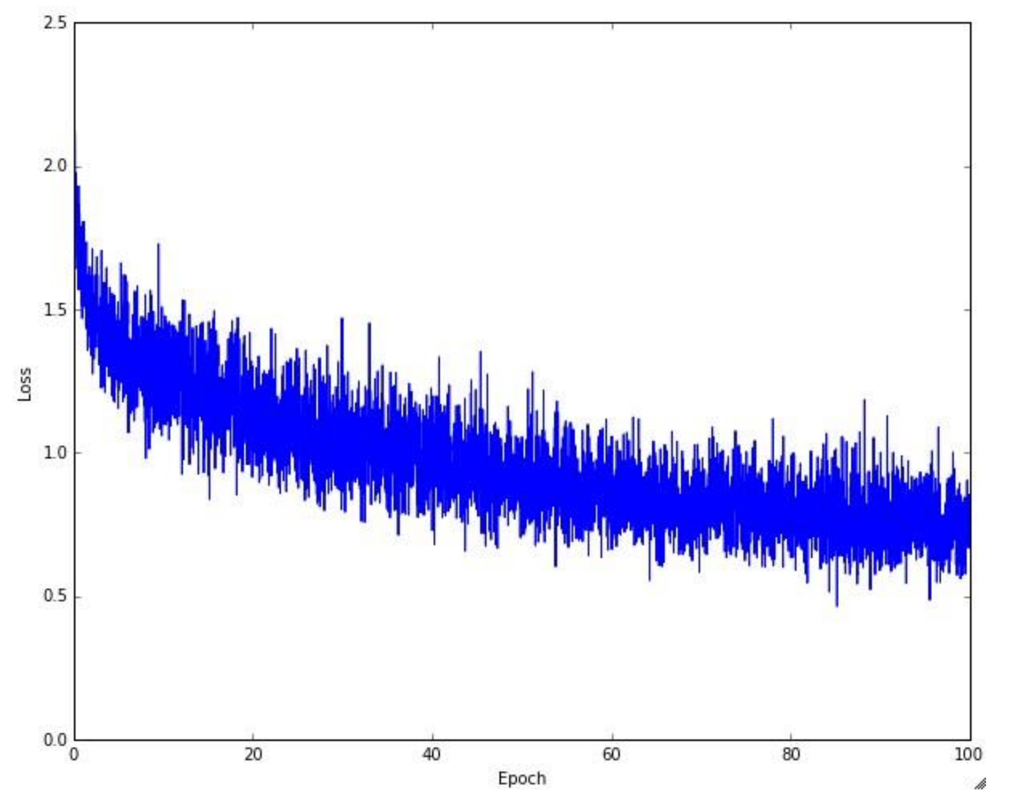
Mini-batch gradient descent



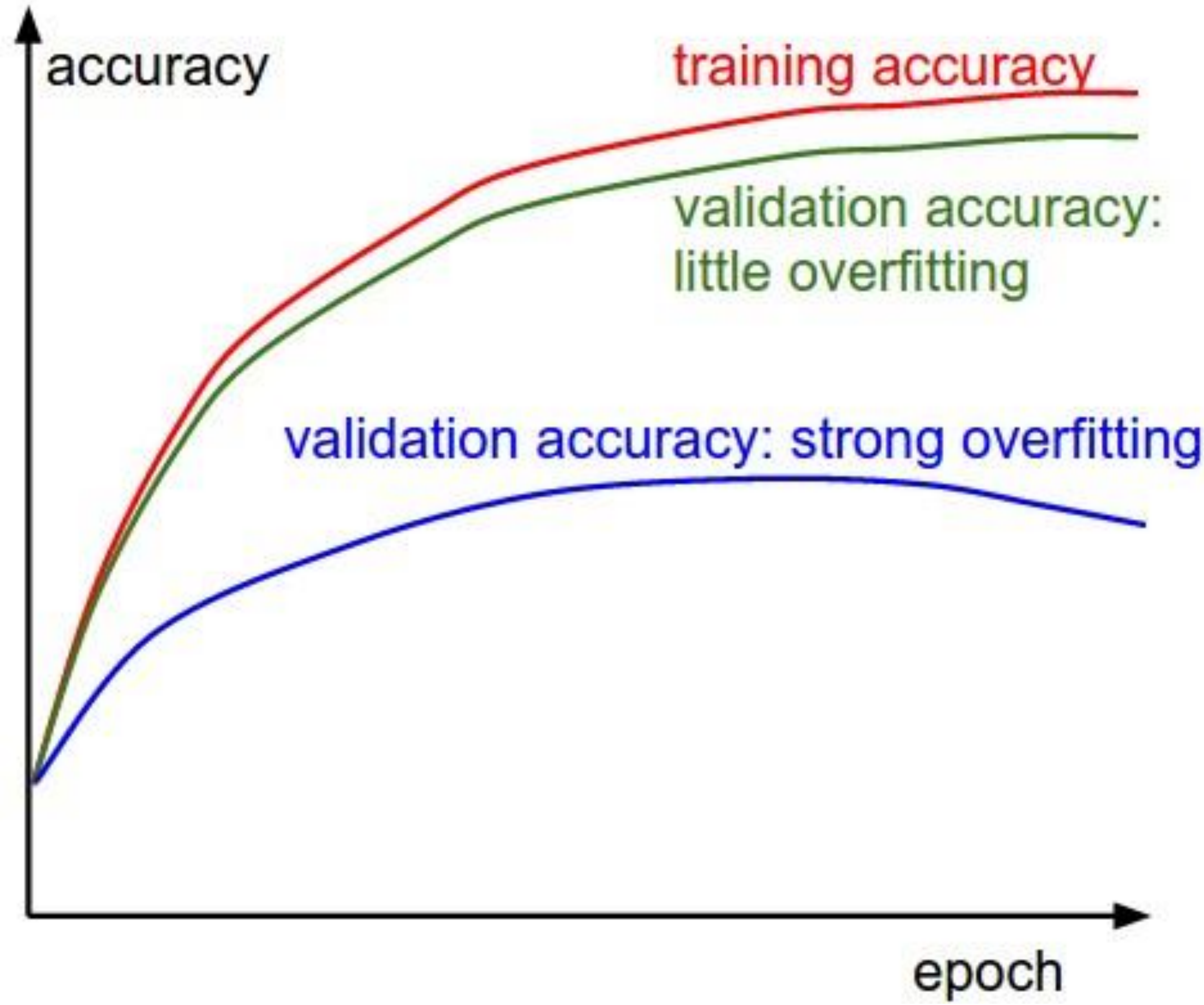


Training – setting learning rate

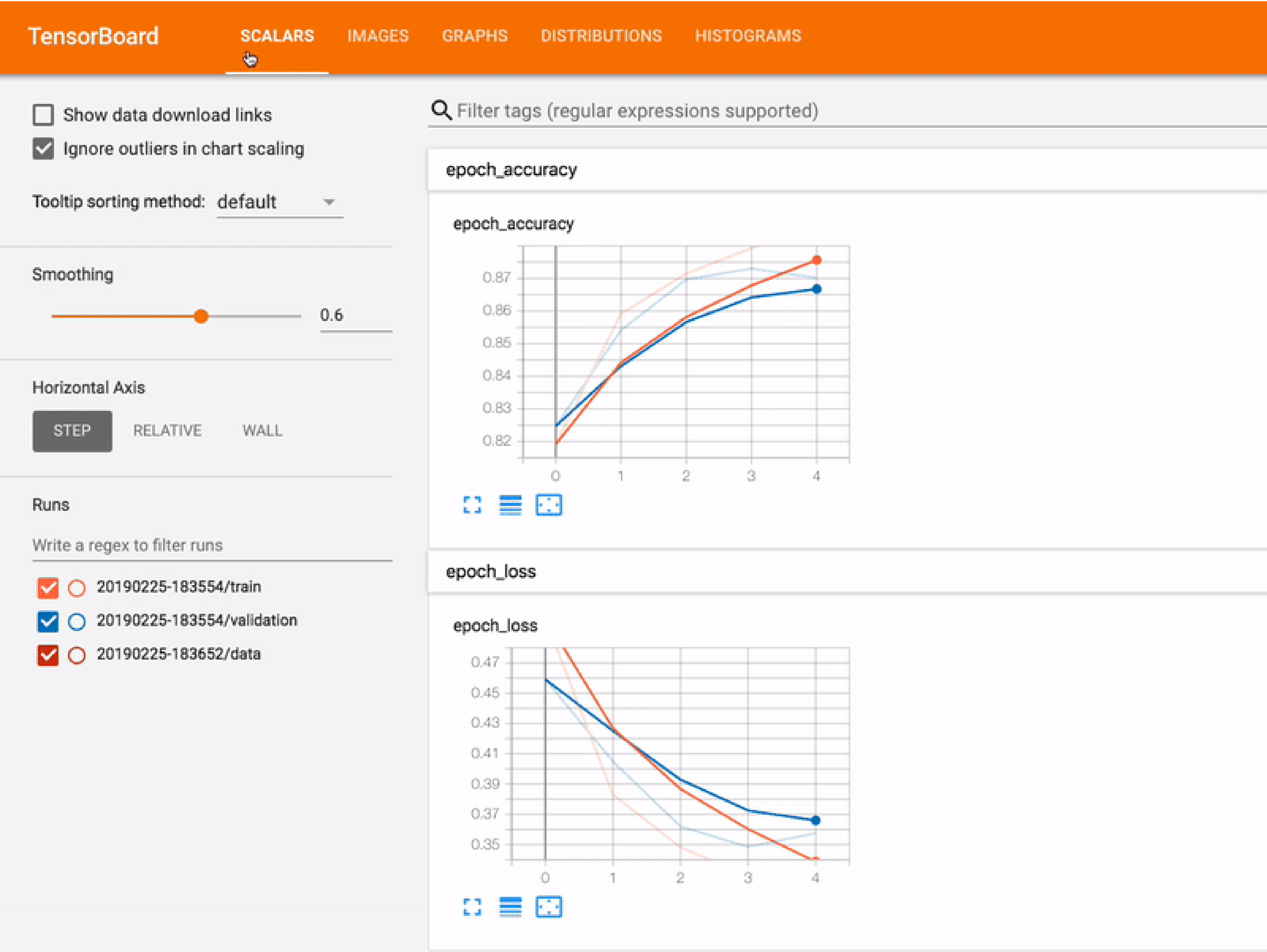




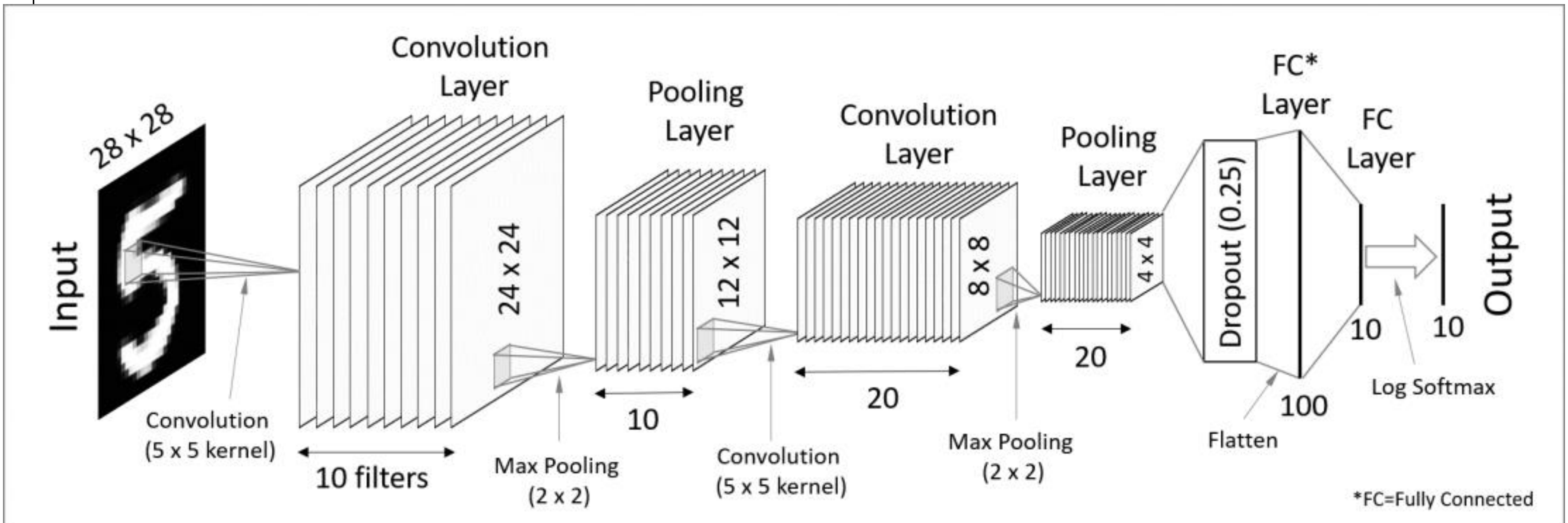
Monitoring



accuracy
training accuracy
validation accuracy:
little overfitting
validation accuracy: strong overfitting
epoch

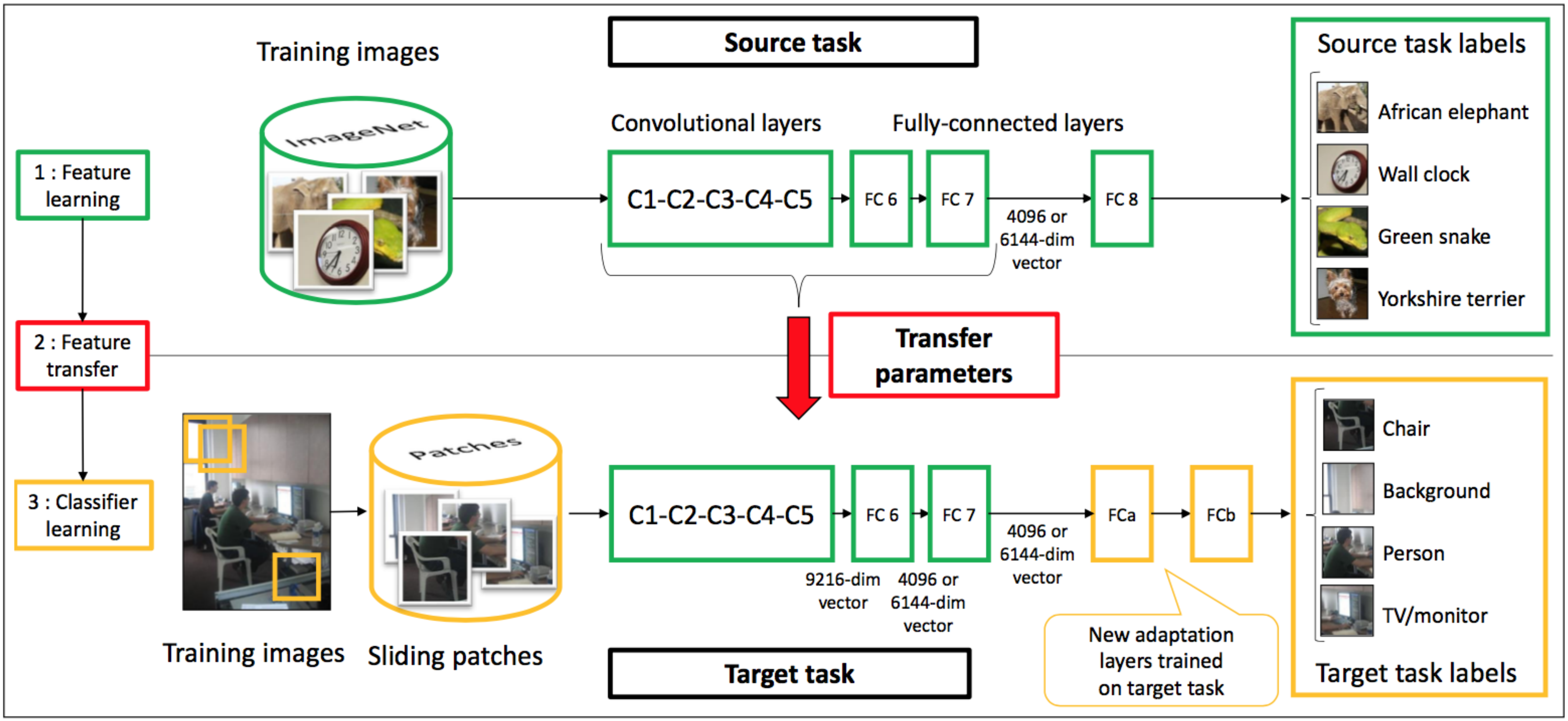

TensorBoard
SCALARS
IMAGES
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
20190225-183554/train
20190225-183554/validation
20190225-183652/data
Filter tags (regular expressions supported)
epoch_accuracy
epoch_accuracy
0.87
0.86
0.85
0.84
0.83
0.82
0
1
2
3
4
epoch_loss
epoch_loss
0.47
0.45
0.43
0.41
0.39
0.37
0.35
0
1
2
3
4

Recall: CNN

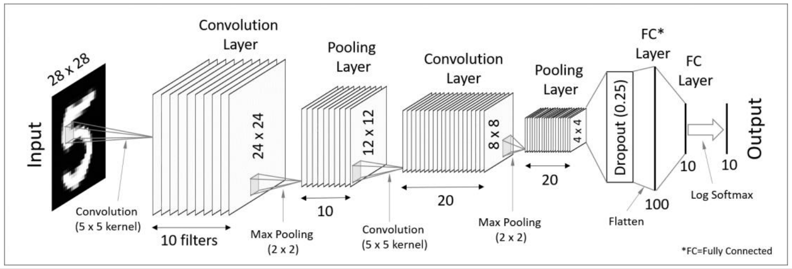

Input
28 x 28
Convolution (5 x 5 kernel)
Convolution Layer
24 x 24
10 filters
Max Pooling (2 x 2)
Pooling Layer
12 x 12
10
Convolution (5 x 5 kernel)
Convolution Layer
8 x 8
20
Max Pooling (2 x 2)
Pooling Layer
4 x 4
20
Dropout (0.25)
Flatten
FC* Layer
100
FC Layer
10
Log Softmax
10
Output
*FC=Fully Connected

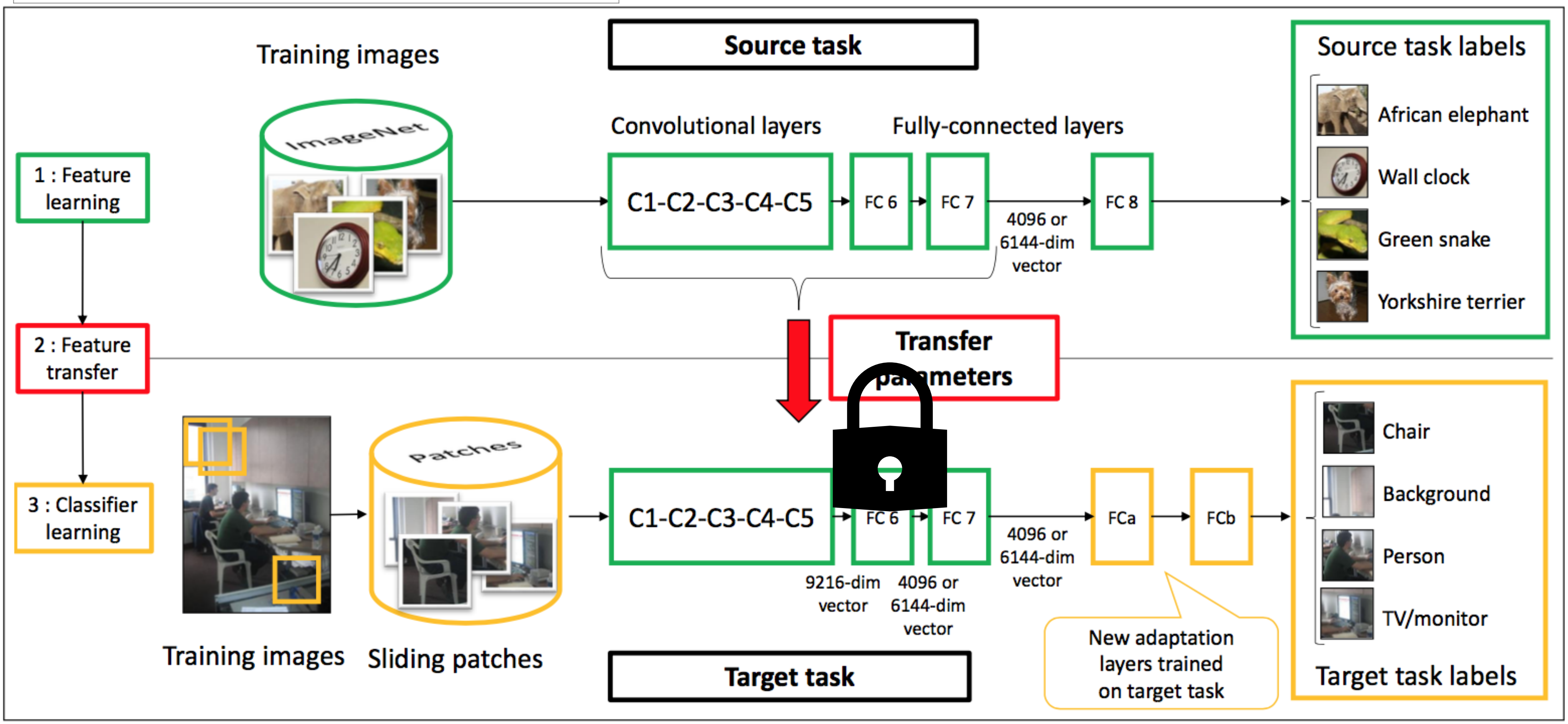
Transfer Learning

Transfer Learning



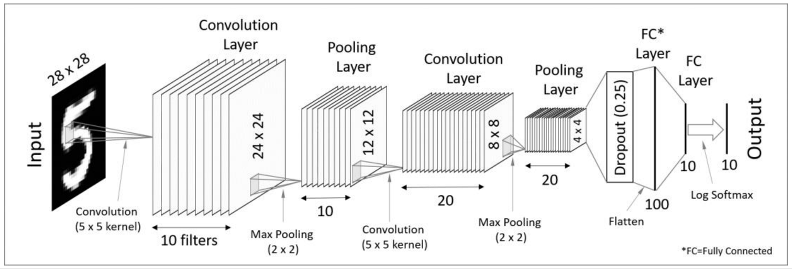
Transfer Learning : Approach-1

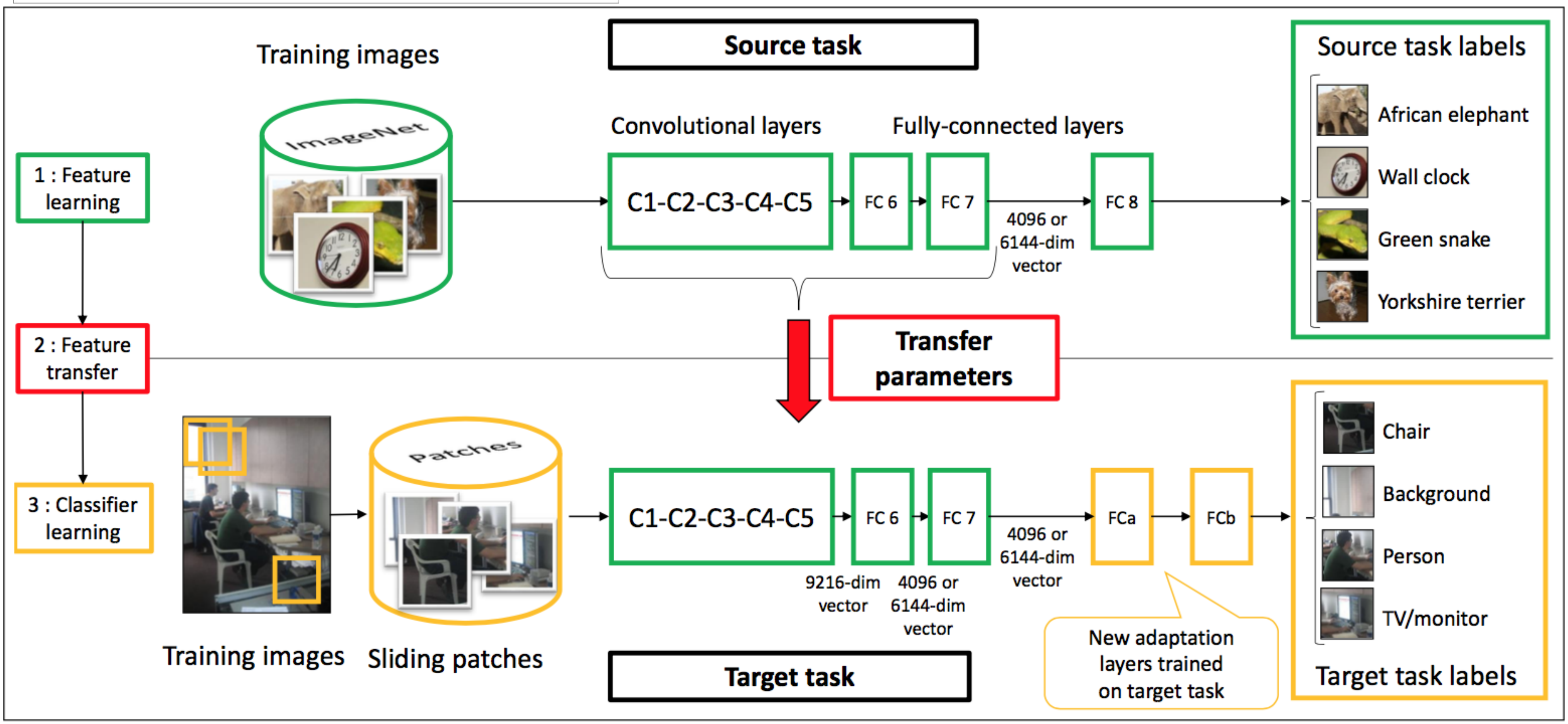




- Learn only weights for newly added layers.
- Ideal when 'new domain' data is small in quantity

Transfer Learning : Approach-2



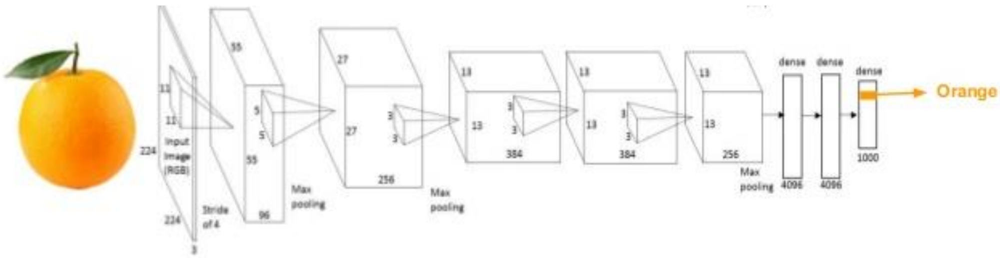


- LR for new layer weights = 10 * source_lr (for bias, 20 * source_lr)
- Ideal when 'new domain' data is reasonably large or domain shift is significant

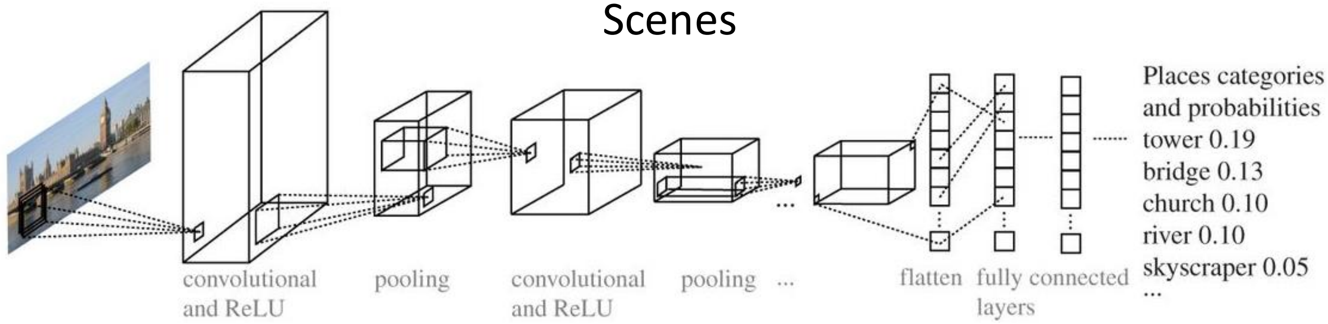
Landscape of CNNs : Applications and Architectures

Image → Label

Objects



Scenes



Faces

DeepFace Architecture

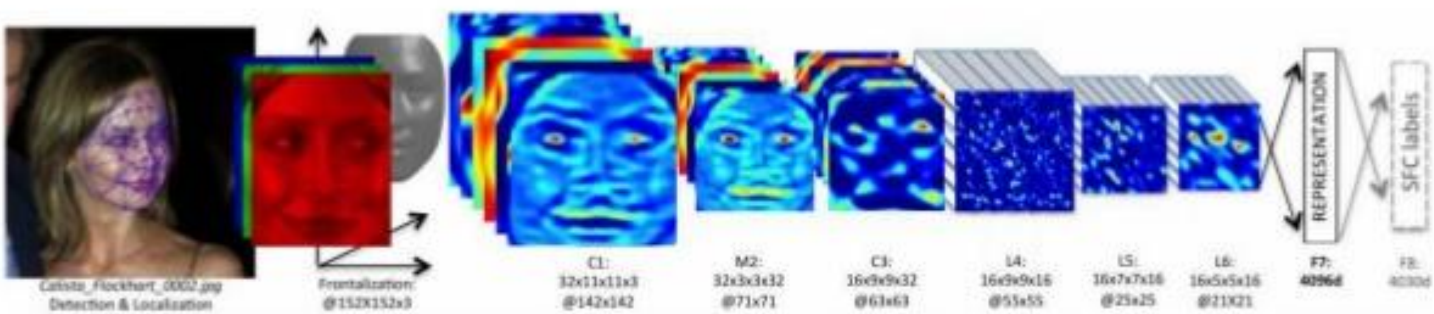
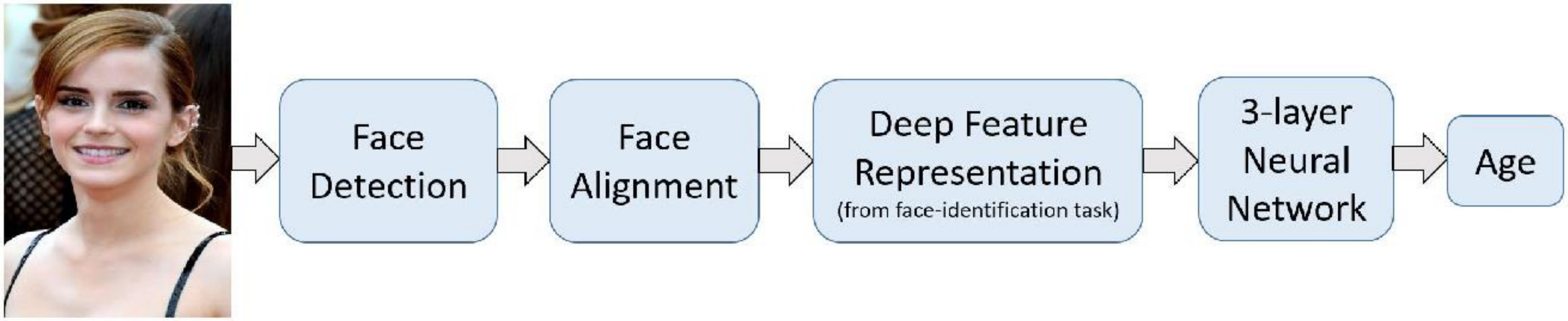
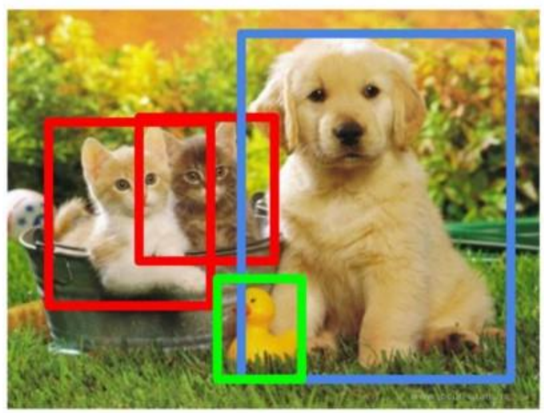


Image → Number



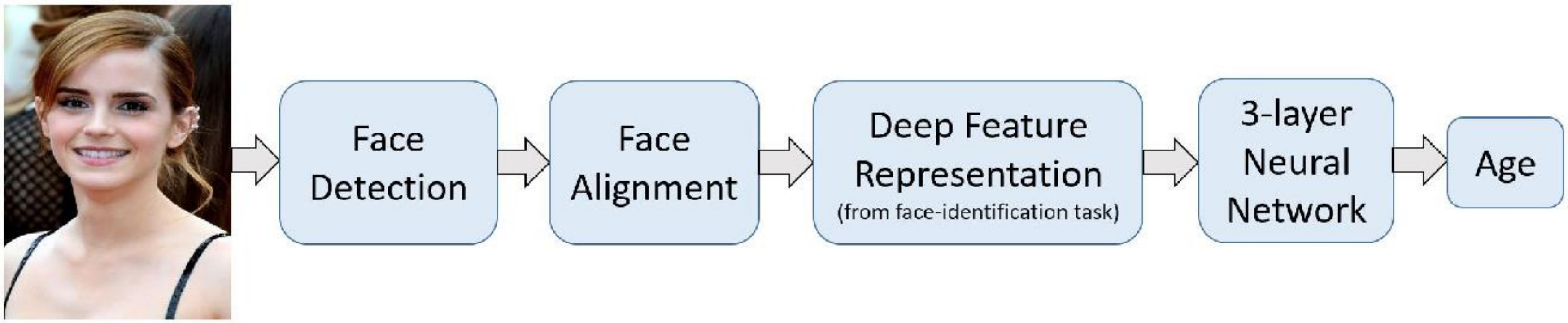
Object Detection



CAT, DOG, DUCK

Image → Number

Age Estimation



Object detection

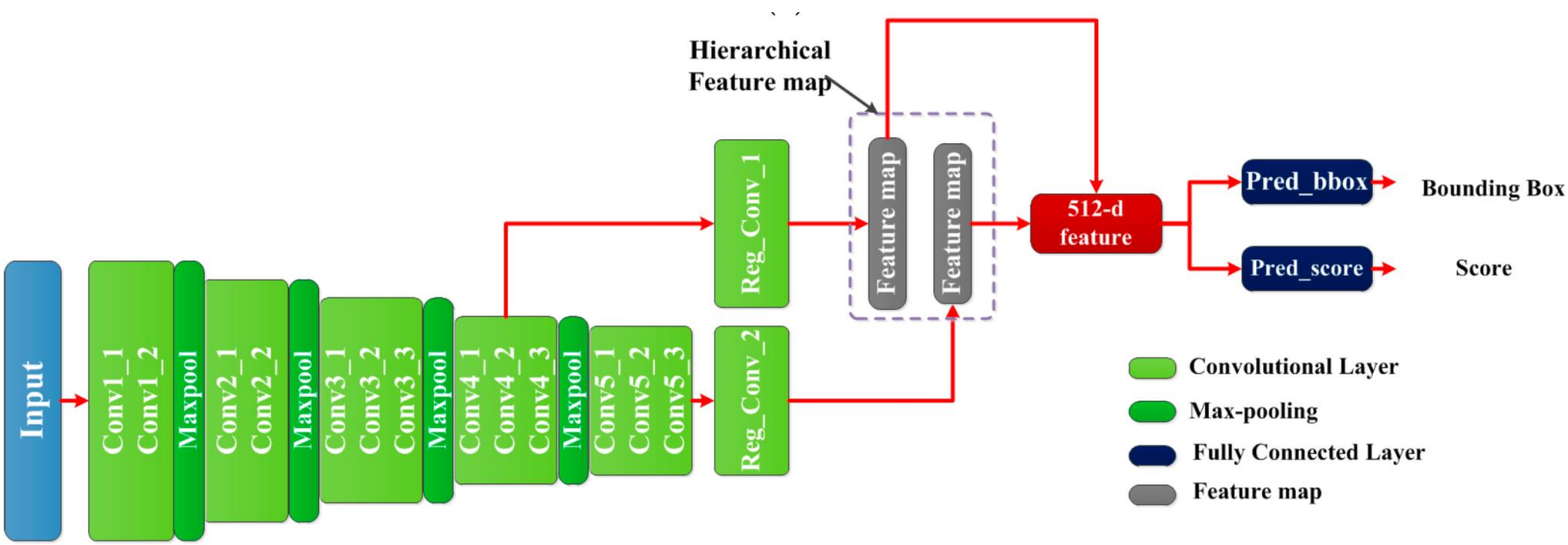
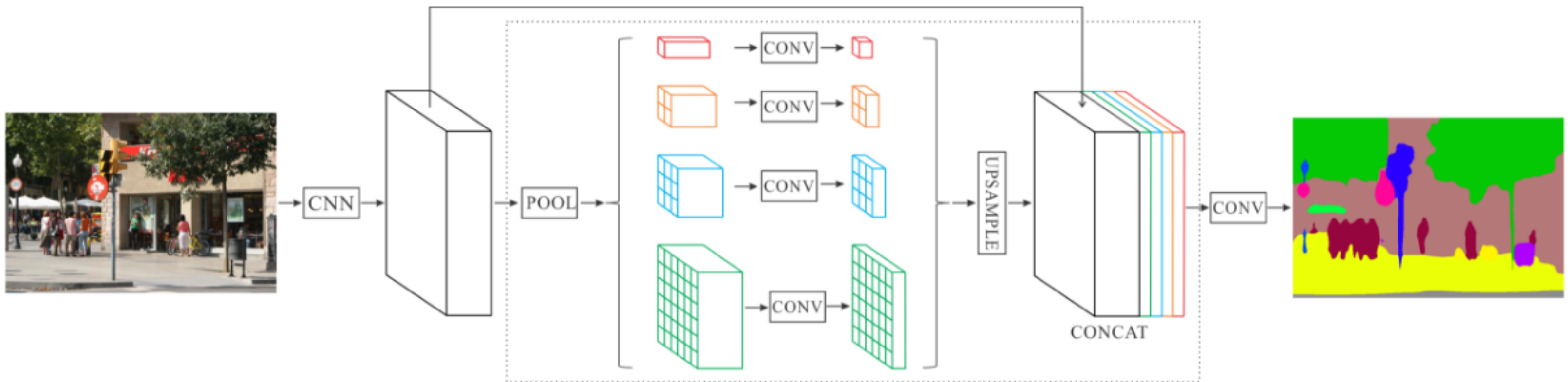


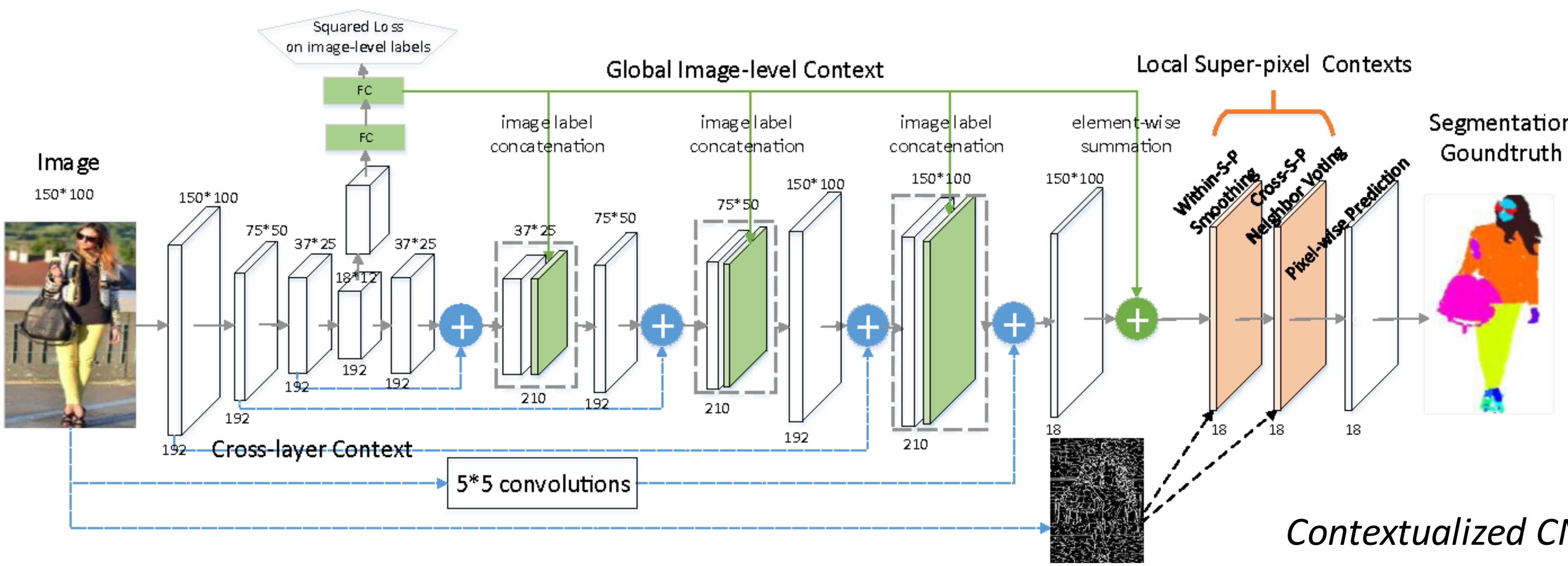
Image → Label Image

Scene Parsing



PSPNet

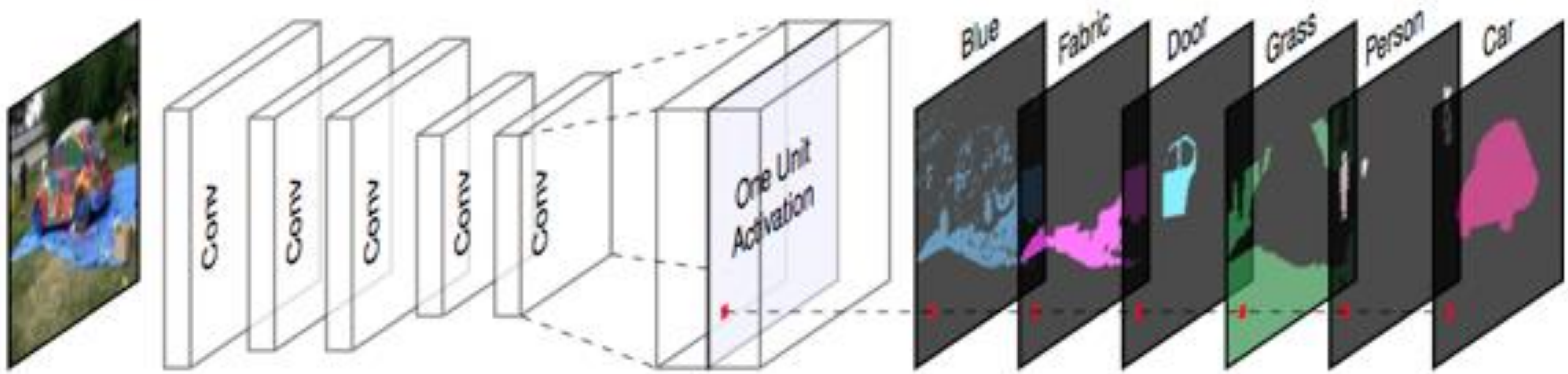
Object Parsing



Contextualized CNN

Semantic segmentation

Semantic segmentation

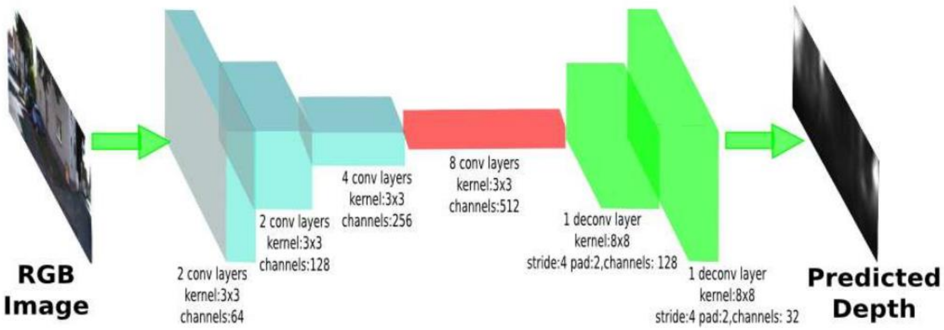


http://netdissect.csail.mit.edu/thumb/slides-thumbnail.png

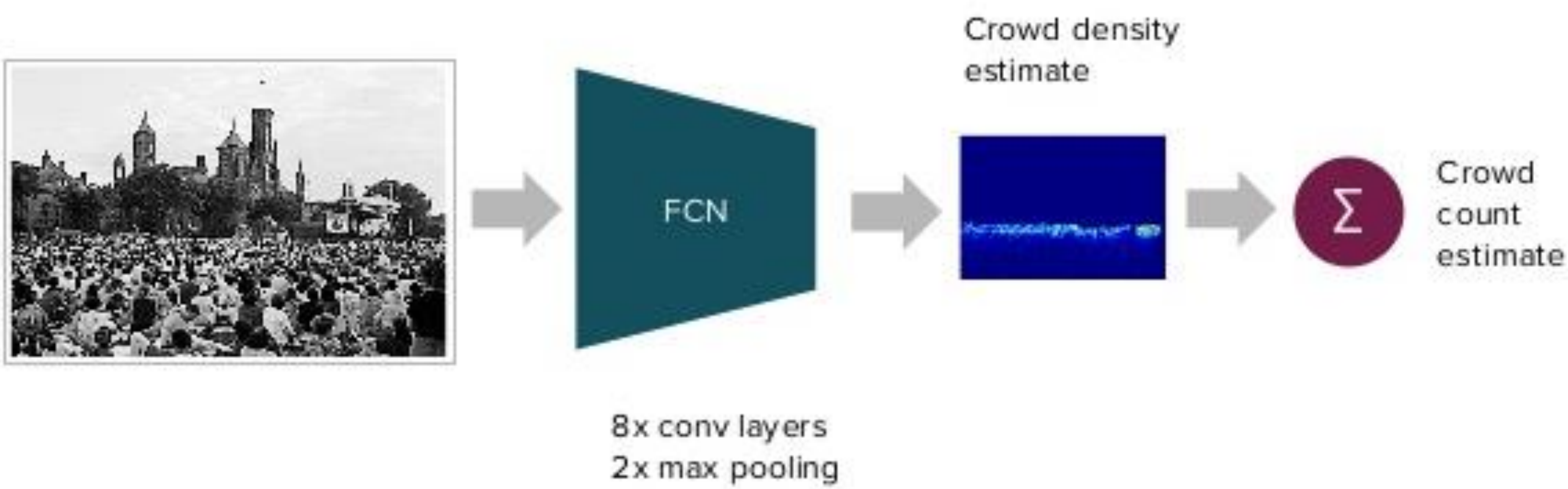
https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/image_segmentation.html

Image → Image

Depth Estimation



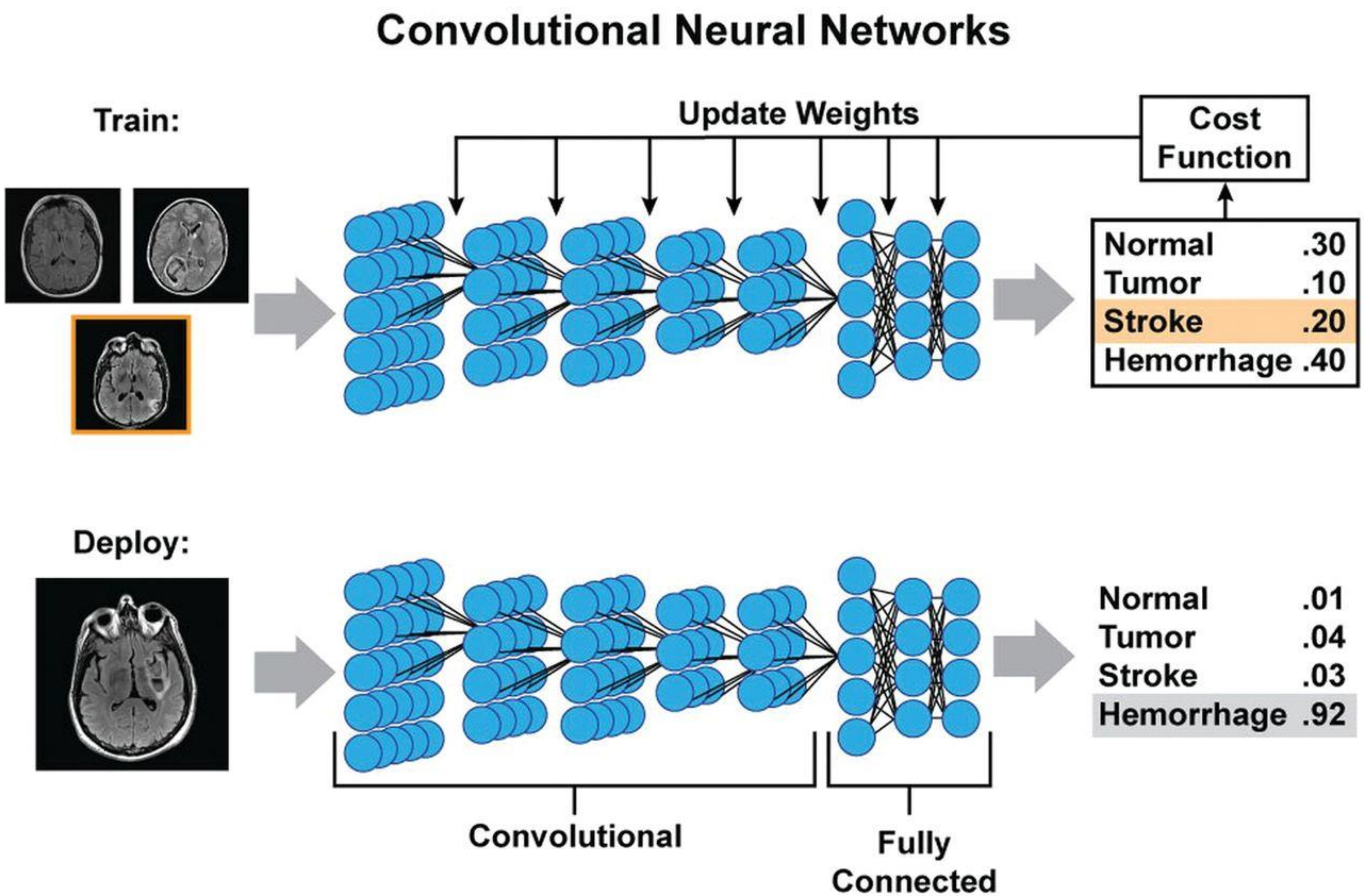
Crowd Counting



Marsden et al. **Fully convolutional crowd counting on highly congested scenes.** VISAPP 2017
https://arxiv.org/abs/1612.00220

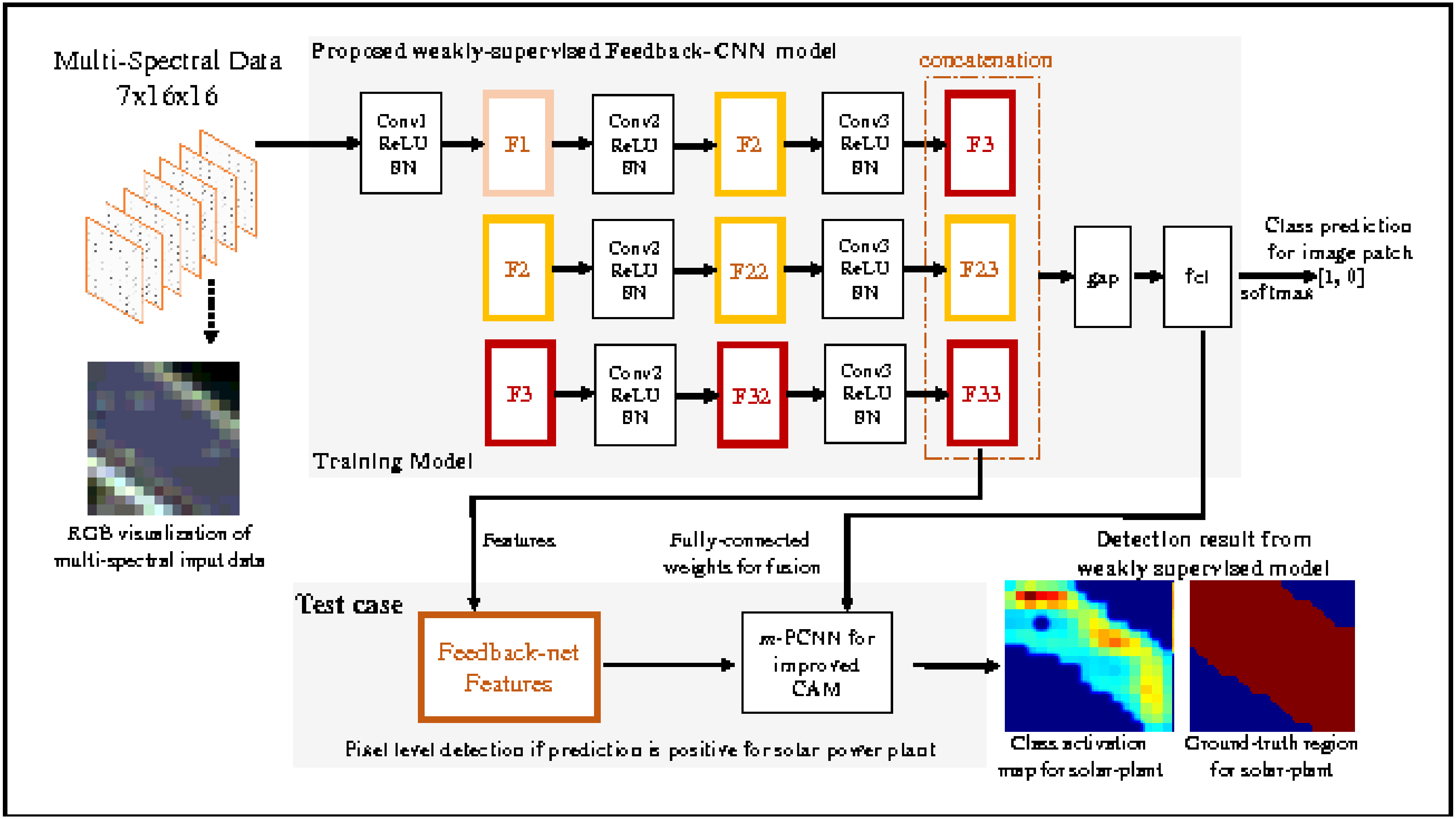
Multi-channel input

NeuroRadiology (fMRI)



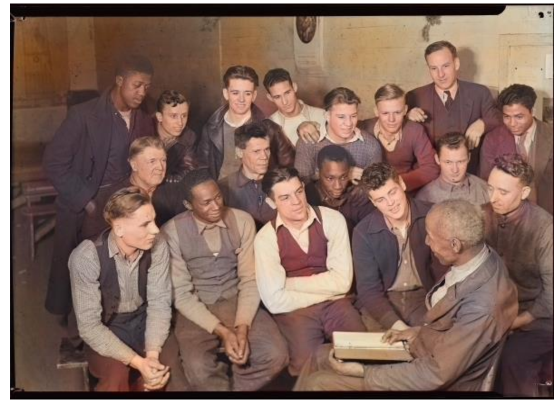
Multi-channel input

Solar-power plant detection from multi-spectral data



Multi-branch input

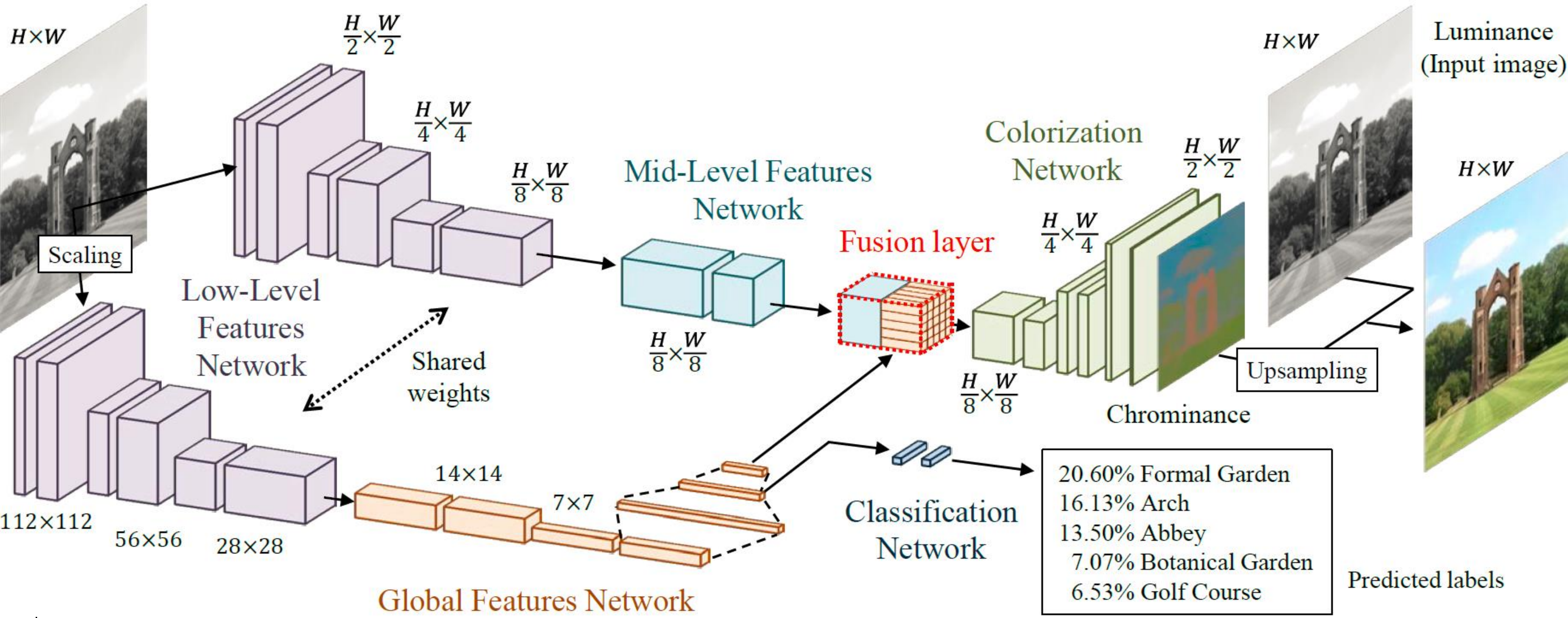
Image Colorization



https://algorithmia.com/algorithms/deeplearning/ColorfulImageColorization

Multi-branch input

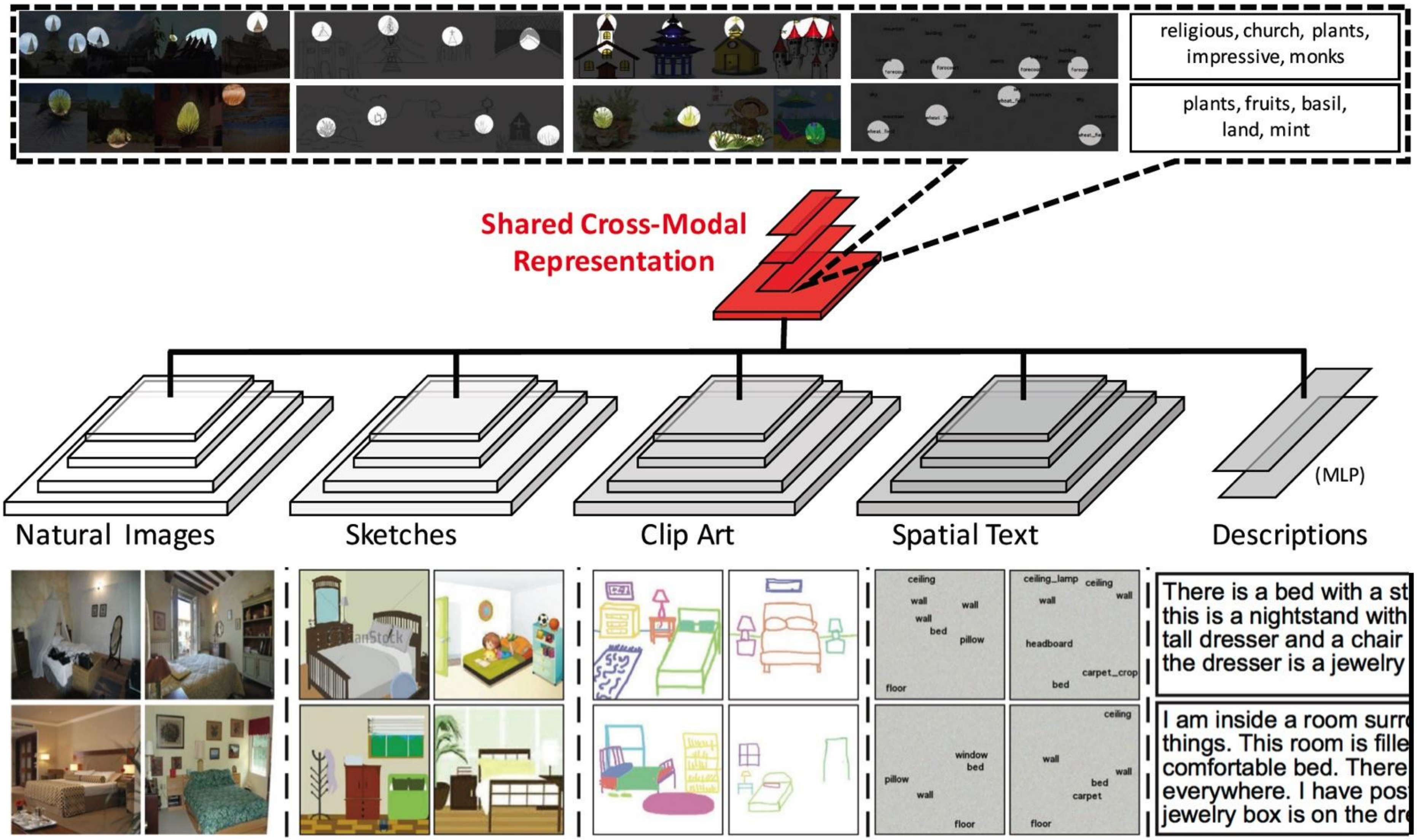
Image Colorization



http://hi.cs.waseda.ac.jp/~iizuka/projects/colorization/en/

Multi-branch input

Cross-modal Scene Classification



http://projects.csail.mit.edu/cmplaces/

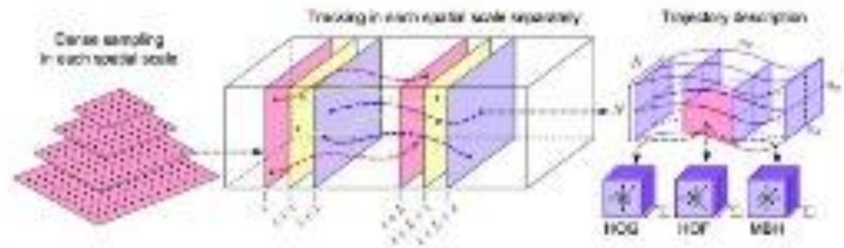
Multi-branch input

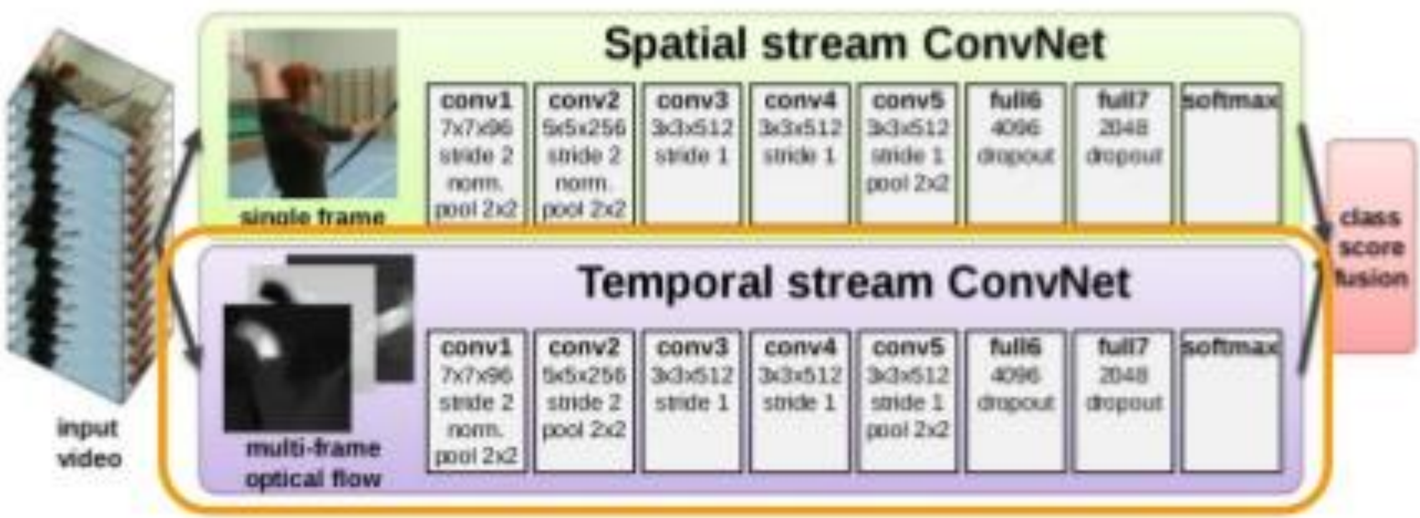
Activity Recognition

Recognition: Two stream

Two CNNs in paralel:
- One for RGB images
- One for Optical flow (hand-crafted features)

Fusion after the softmax layer

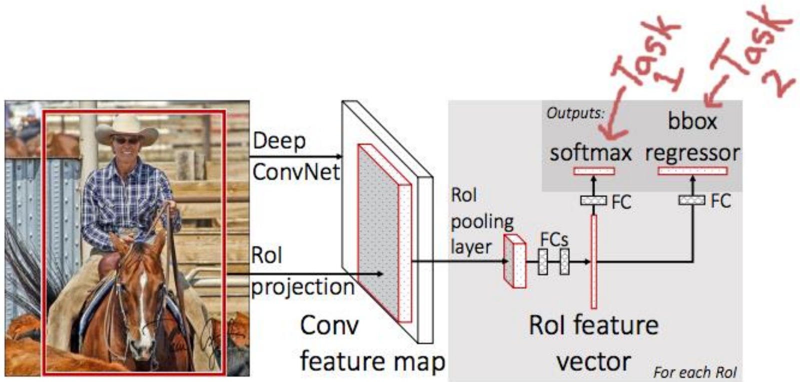




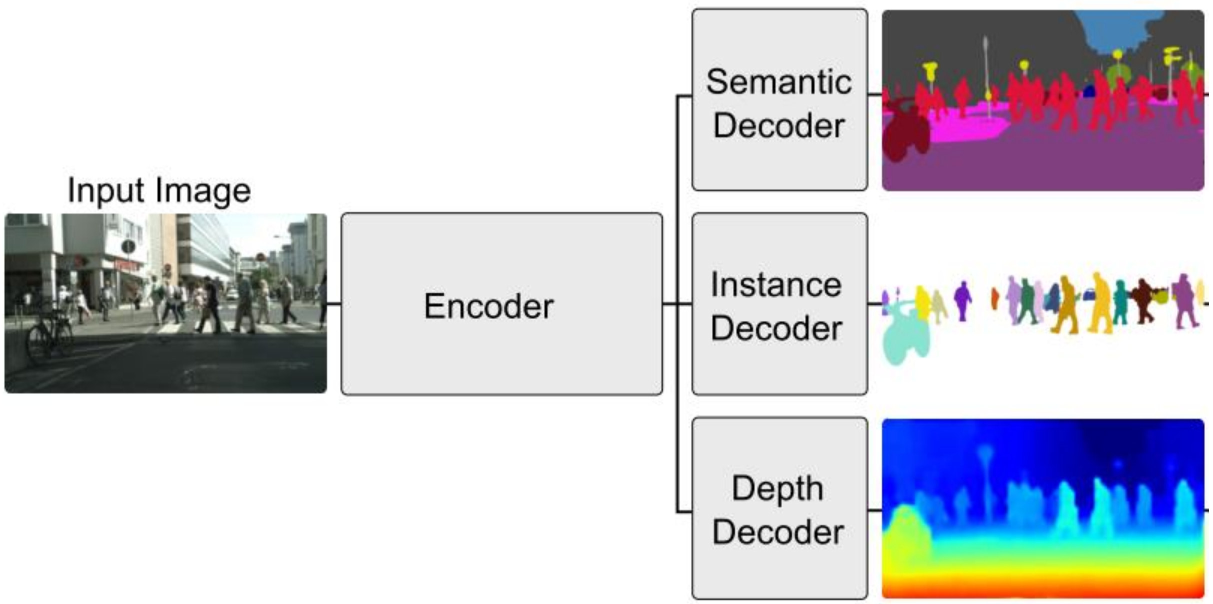
Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." NIPS 2014.

Multi-branch output

Object Detection, Classification



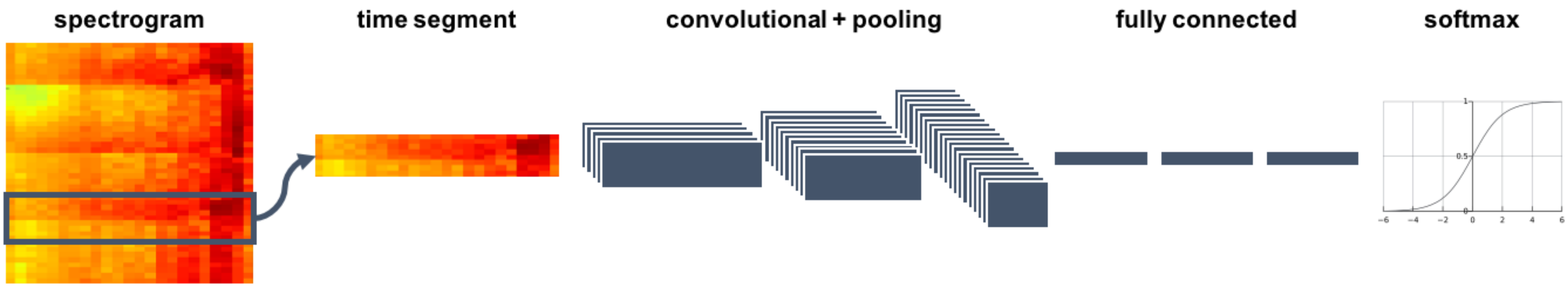
Scene Parsing



http://ruder.io/multi-task/

Image CNNs for non-image data

Audio Beat Detection



http://tommymullaney.com/projects/rhythm-games-neural-networks