Popular DNNs and Applications

Chia-Chi Tsai (蔡家齊)
cctsai@gs.ncku.edu.tw
AI System Lab
Department of Electrical Engineering
National Cheng Kung University

Outline

- Example 101 – LeNet, AlexNet
- Popular DNNs
- Popular Applications and Case Studies

Outline

- Example 101 – LeNet, AlexNet
- Popular DNNs
- Popular Applications and Case Studies

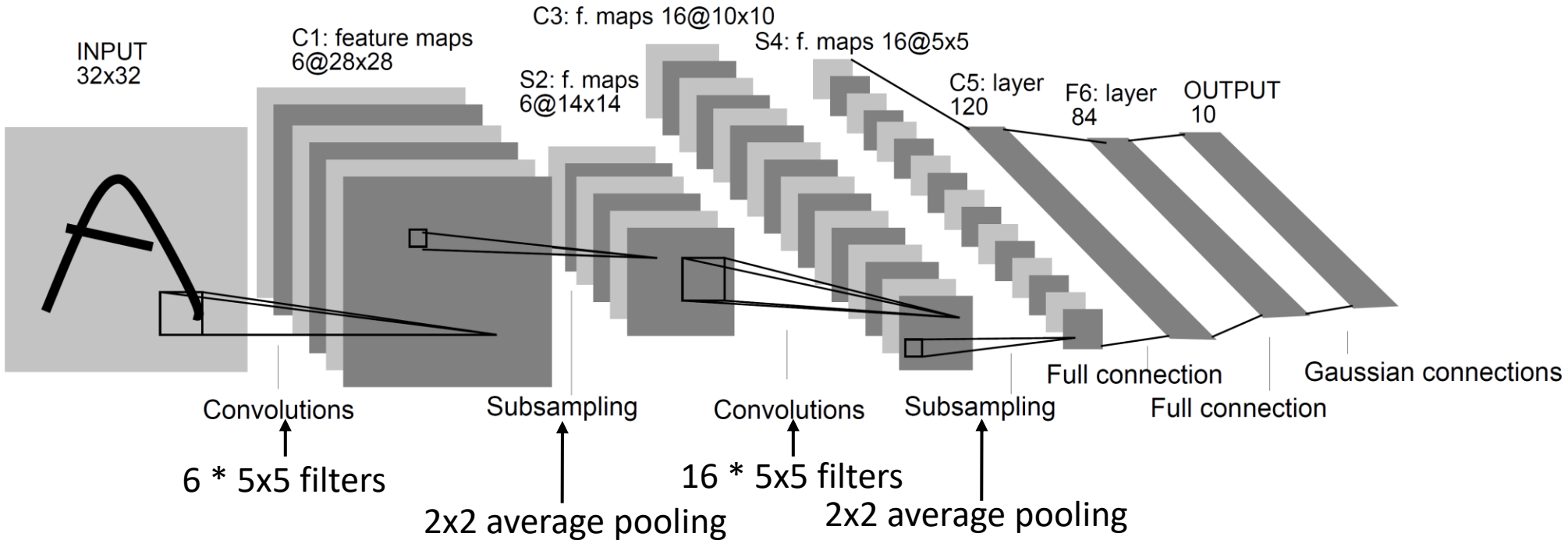
MNIST Dataset

- Handwritten digit database
- 28x28 pixels
- Grayscale
- 10 classes
- 60,000 Training
- 10,000 Testing

LeNet-5 Architecture

- CONV Layers: 2
- Fully Connected Layers: 2
- Sigmoid used for non-linearity

Weights: 60k
MACs: 341k

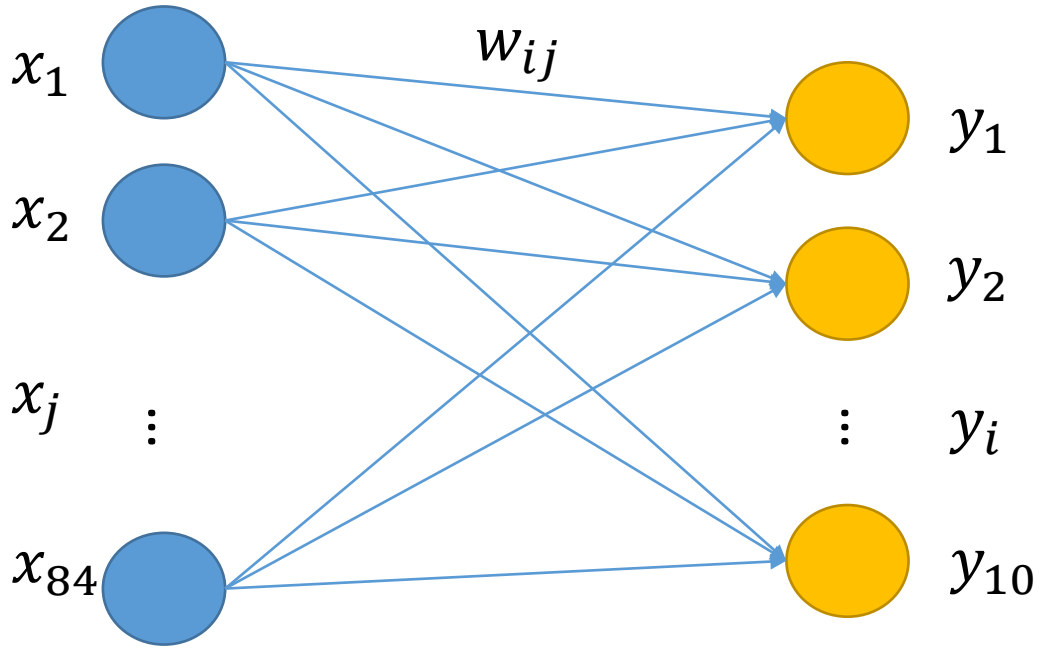


Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

LeNet-5 Output Layer

- Gaussian connection with Euclidean Radial Basis Function units(RBF)

$$y_i = \sum_j (x_i - w_{ij})^2$$



LeNet-5 Loss Function

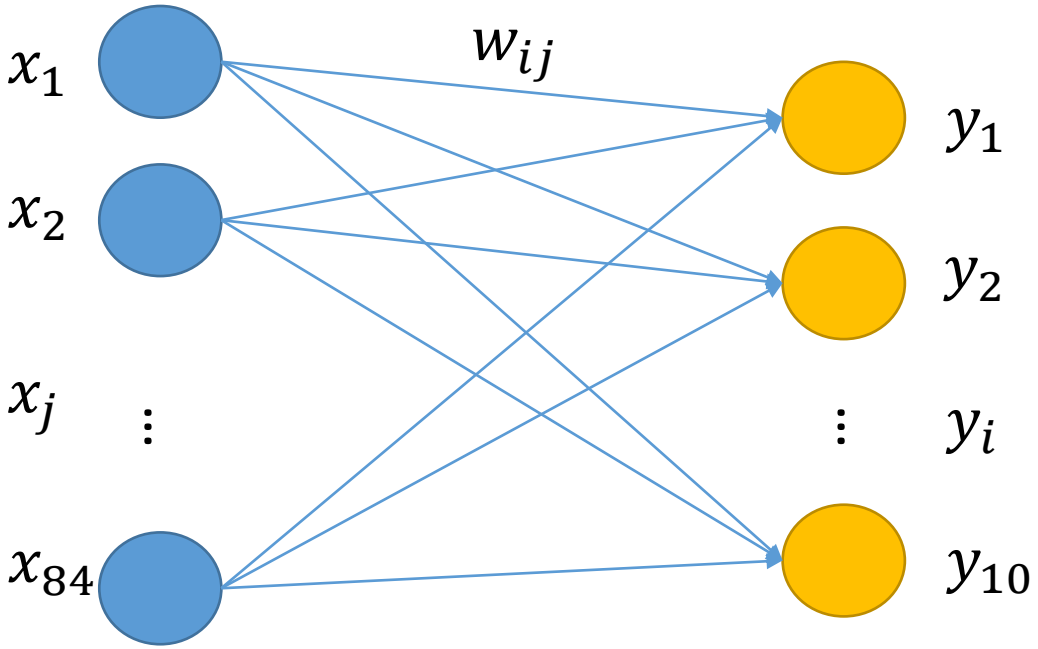
- Loss function
  - Gaussian connection with Euclidean Radial Basis Function units(RBF)

$$\text{minimize } E(W) = \frac{1}{P}\sum_{p=1}^{P}\left(y_{D_p}(Z^p, W) + \log\left(e^{-j} + \sum_i e^{-y_i(Z^p, W)}\right)\right)$$

Pull-down correct class

prevents class score too large

Pull-up background class



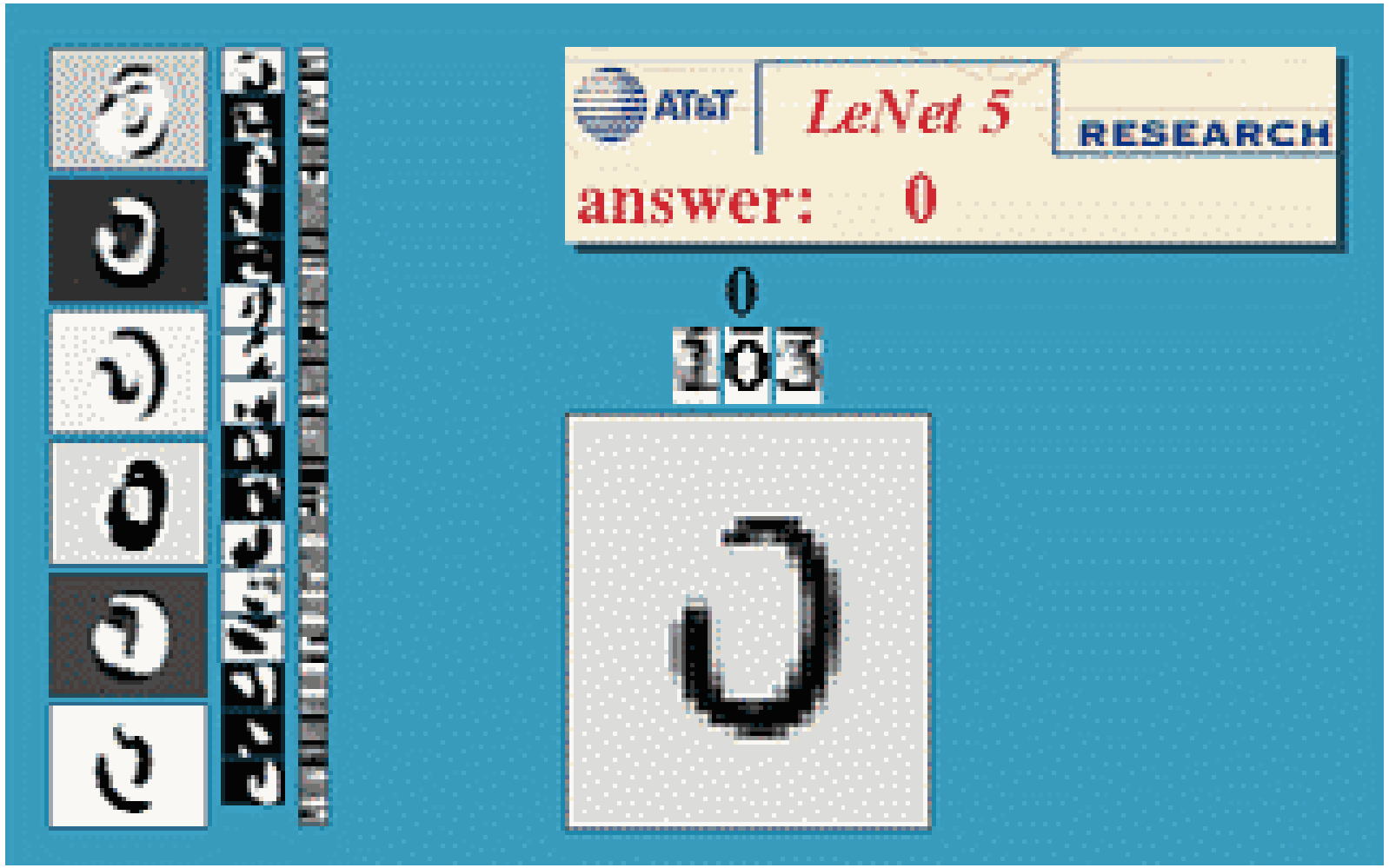
D_p: correct class of input pattern

P: $samples$

Z^p: input pattern

j: constant

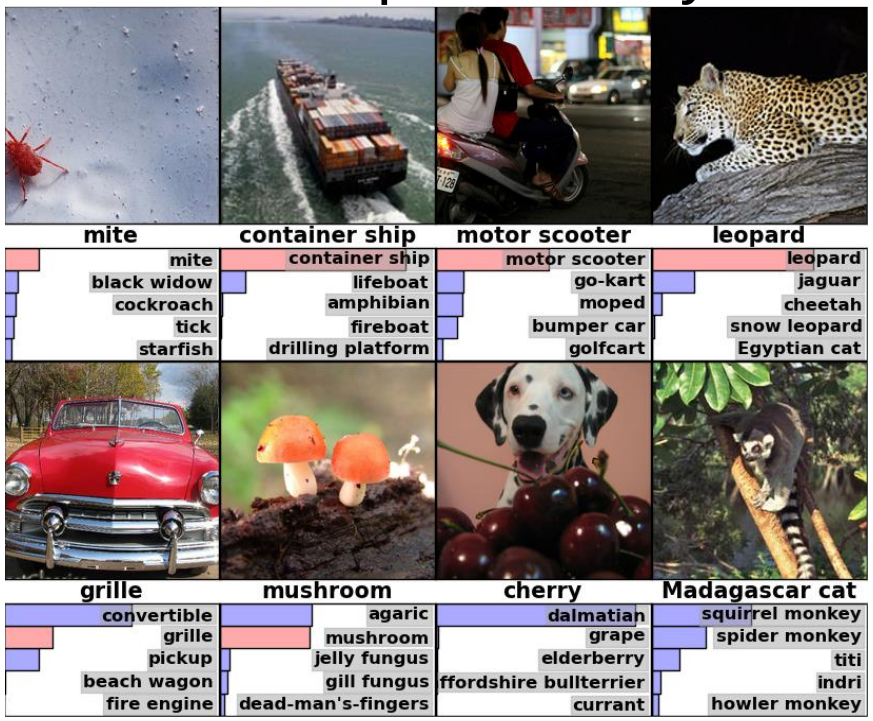
LeNet-5 Example

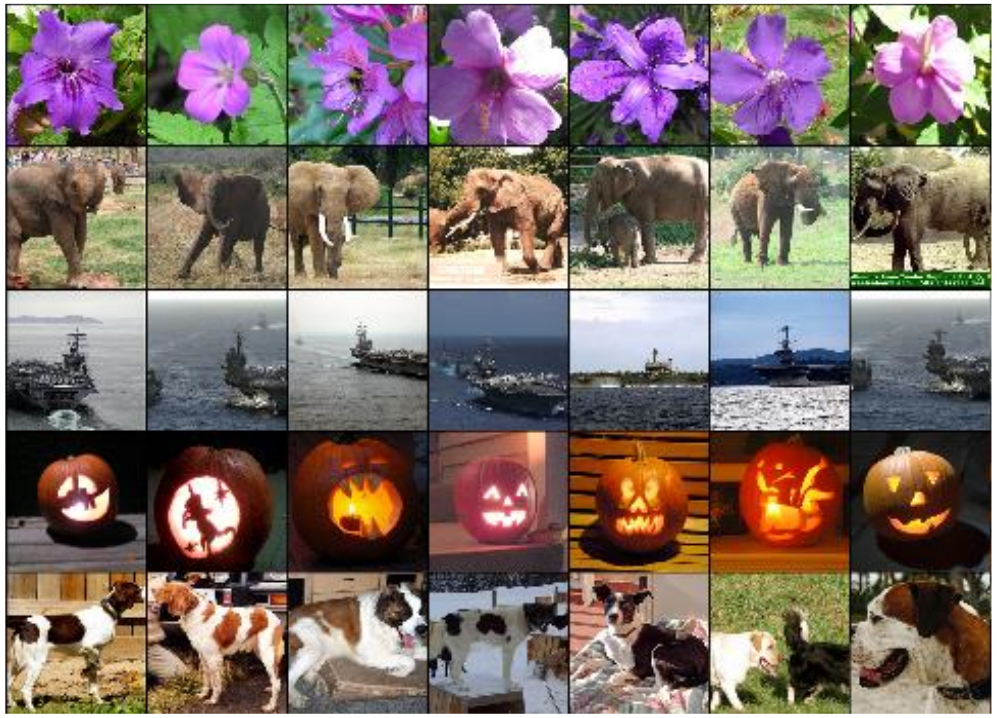


http://yann.lecun.com/exdb/lenet/

ImageNet Dataset

- ImageNet LSVRC-2010 with 1.2 million images
- Target - Top-1 and top-5 error rate:
  - The fraction of images for which the correct label is not among the 1/5 labels considered most probable by the model.

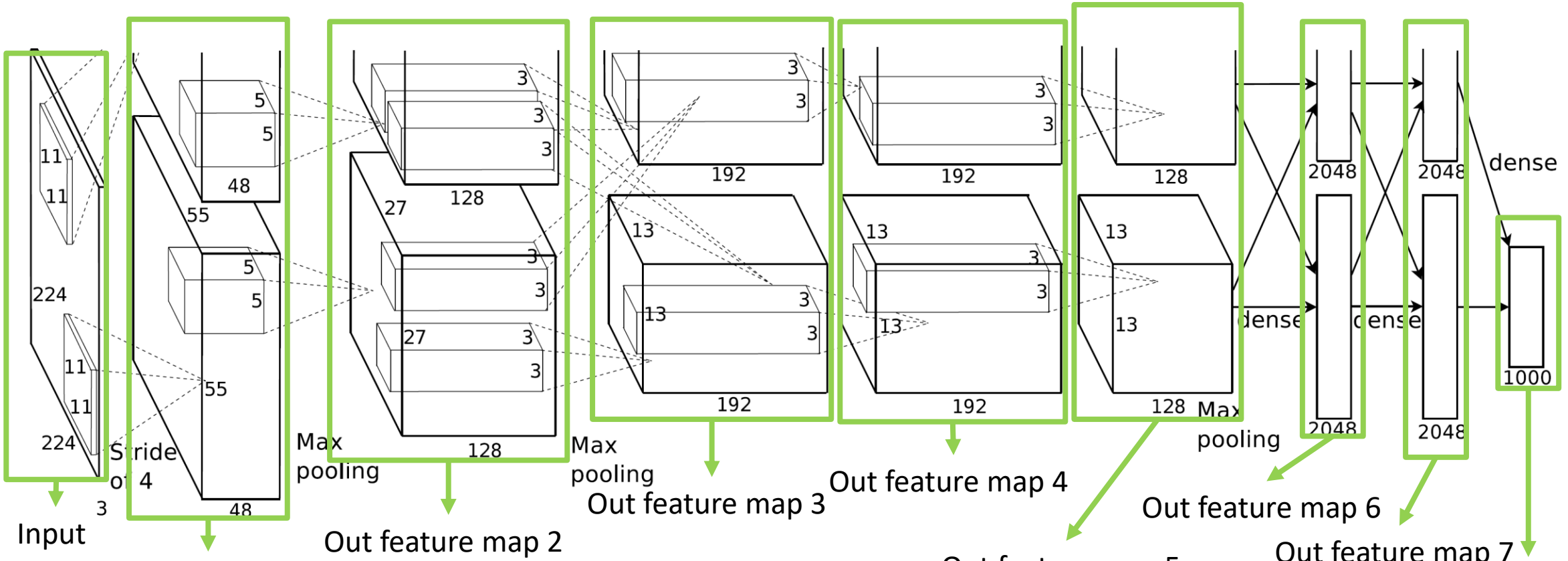




AlexNet Architecture

ILSCVR12 Winner !

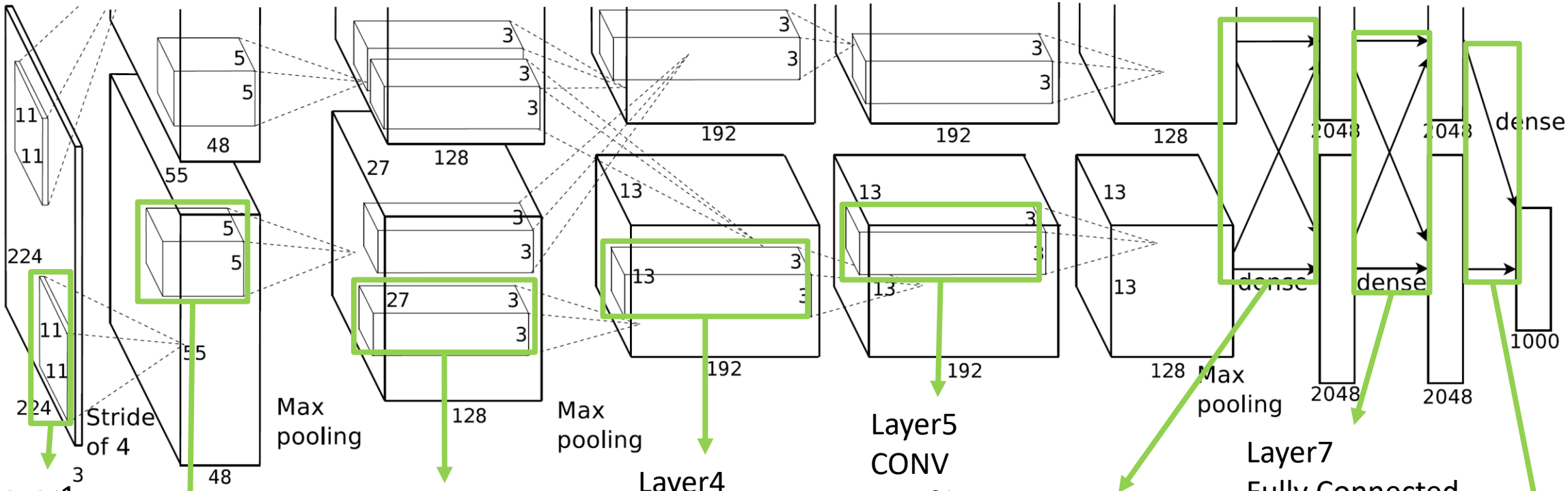
Weights: 61M
MACs: 724M



CONV Layers: 5
Fully Connected Layers: 3
ReLU used for non-linearity

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

AlexNet Architecture



Layer1
CONV
96 filters
11x11x3

Layer2
CONV
256filters
5x5x48

Layer3
CONV
384 filters
3x3x256

Layer4
CONV
384 filters
3x3x192

Layer5
CONV
384 filters
3x3x192

Layer6
Fully Connected
4096 filters
13x13x256

Layer7
Fully Connected
4096 filters
4096x1

Layer8
Fully Connected
1000 filters
4096x1

Large Sizes with Varying Shapes

AlexNet Convolutional Layer Configurations

Layer	Filter Size (RxS)	# of Filters (M)	# of Channels (C)	Outpur fmap Size (ExF)	Stride
1	11x11	96	3	55x55	4
2	5x5	256	48	27x27	1
3	3x3	384	256	13x13	1
4	3x3	384	192	13x13	1
5	3x3	256	192	13x13	1

weights bias

$$\#\ of\ params\ of\ CONV = \boxed{R \times S \times C \times M} + \boxed{M}$$

$$\#\ of\ MACs\ of\ CONV = (R \times S \times C + \boxed{1}) \times E \times F \times \mathrm{M}$$

bias add

Layer 1

$11 \times 11 \times 3 \times 96 + 96 = 34944\ parmas$

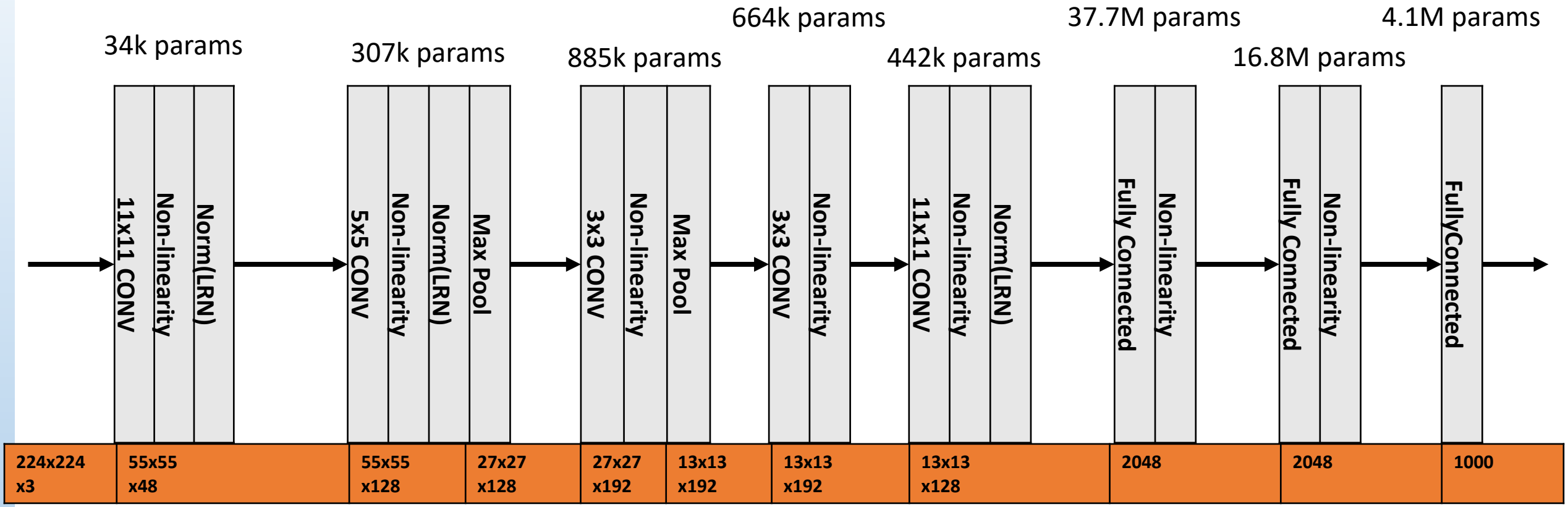
$(11 \times 11 \times 3 + 1) \times 55 \times 55 \times 96 = 105705600\ MACs$

Layer 2

$5 \times 5 \times 48 \times 256 + 256 = 307456 parmas$

$(5 \times 5 \times 48 + 1) \times 27 \times 27 \times 256 = 224135424\ MACs$

AlexNet Dataflow (Simplified)



Local Response Normalization (LRN)

- Help generalization

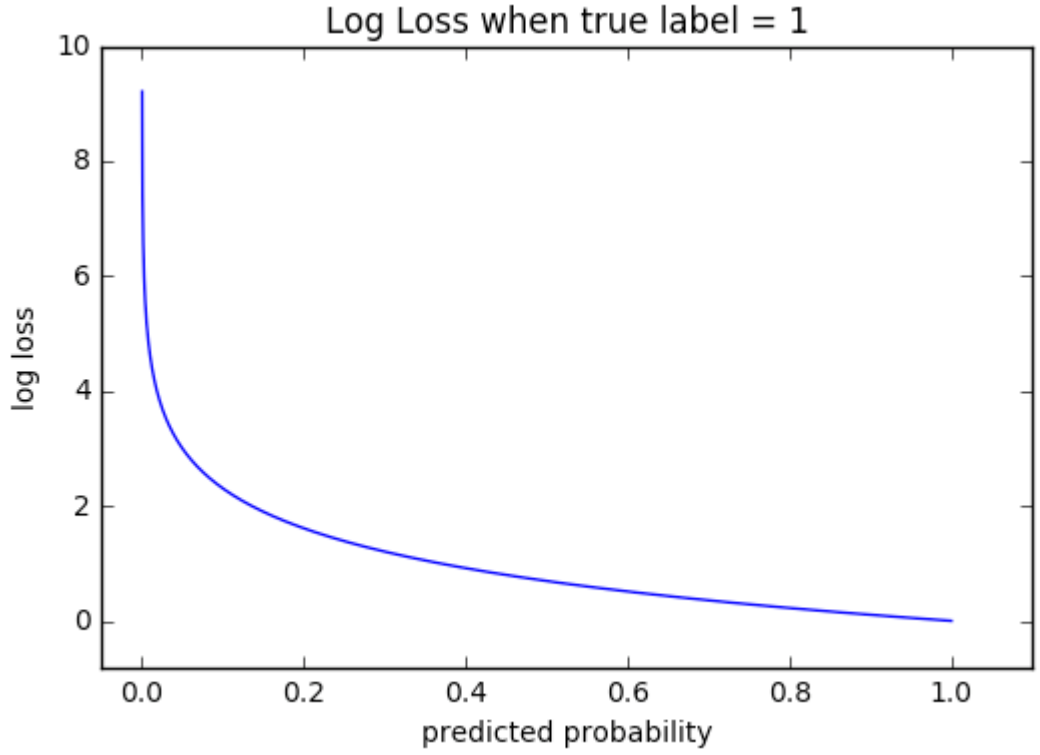
$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

  - $\mathrm{n} = 5, K = 2, \alpha = 10^{-4}, \beta = 0.75$ => heuristic
  - N: # of channels, a: output of CONV at location x,y

- Limited improvement in modern DNNs
  - Seldom used recently

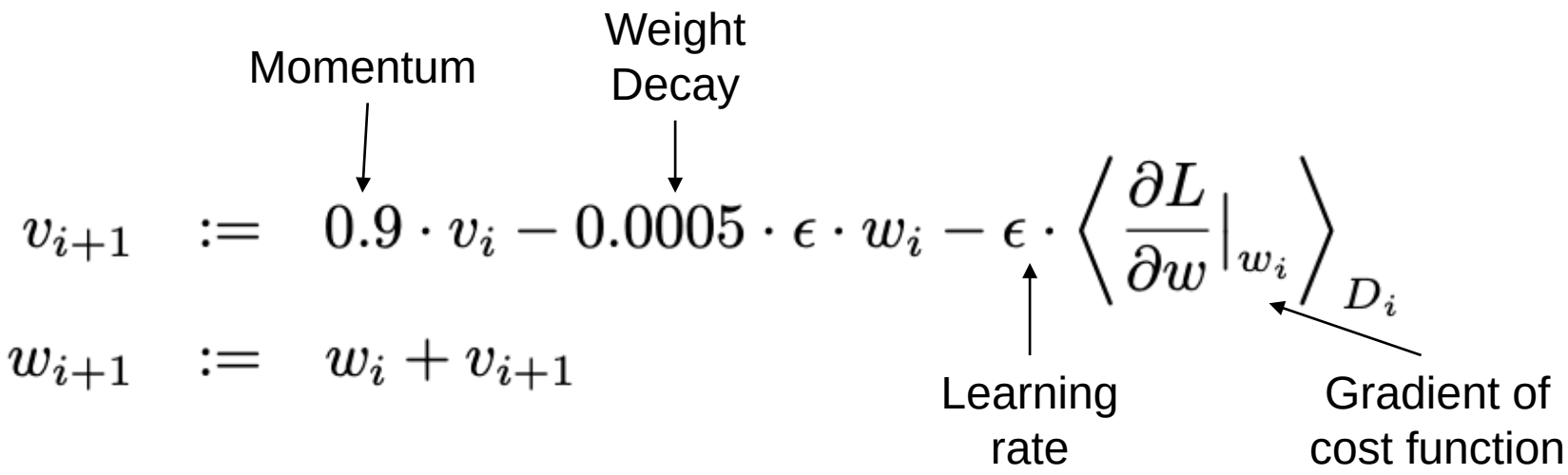
AlexNet Cost(Loss) Function

- Minimize the cross-entropy loss function
- Measures the performance of a classification model whose output is a probability value between 0 and 1
- $CF = -\frac{1}{N}(\sum_{i-1}^{N} y_i \cdot \log(\hat{y}_i))$



AlexNet Optimization Method

- Stochastic Gradient Descent
  - Batch size = 128
  - Update rule:

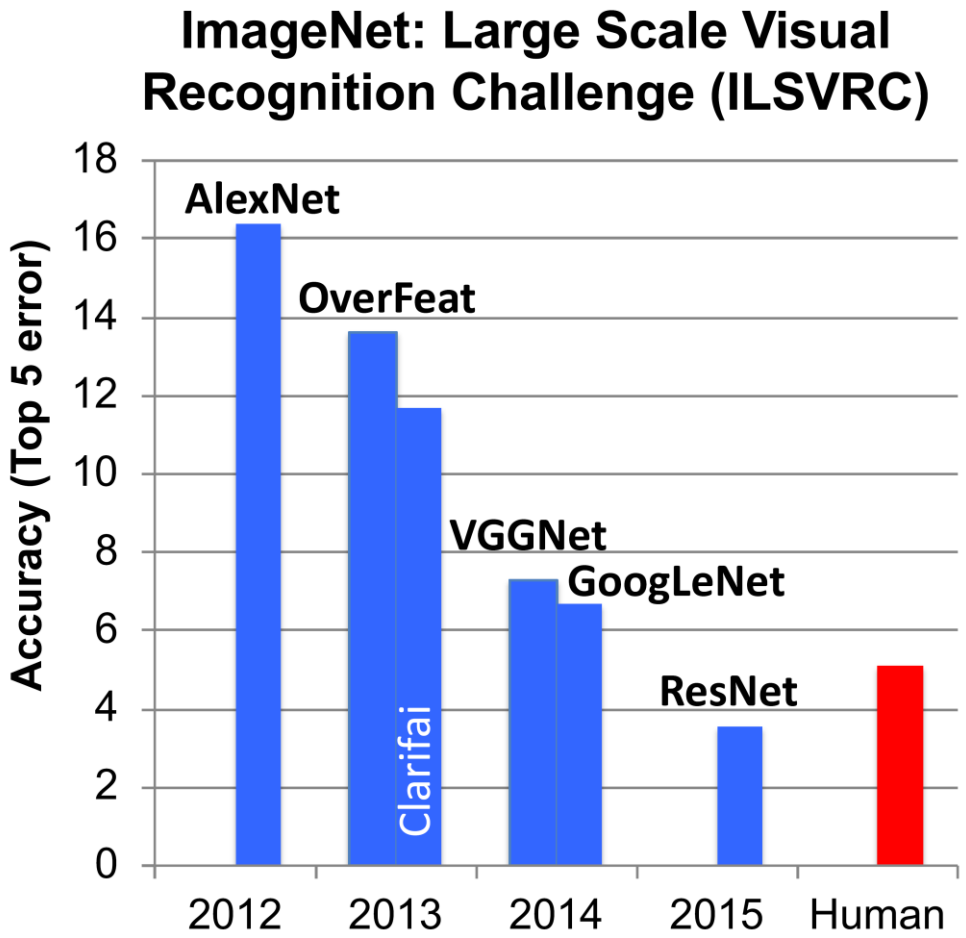


Outline

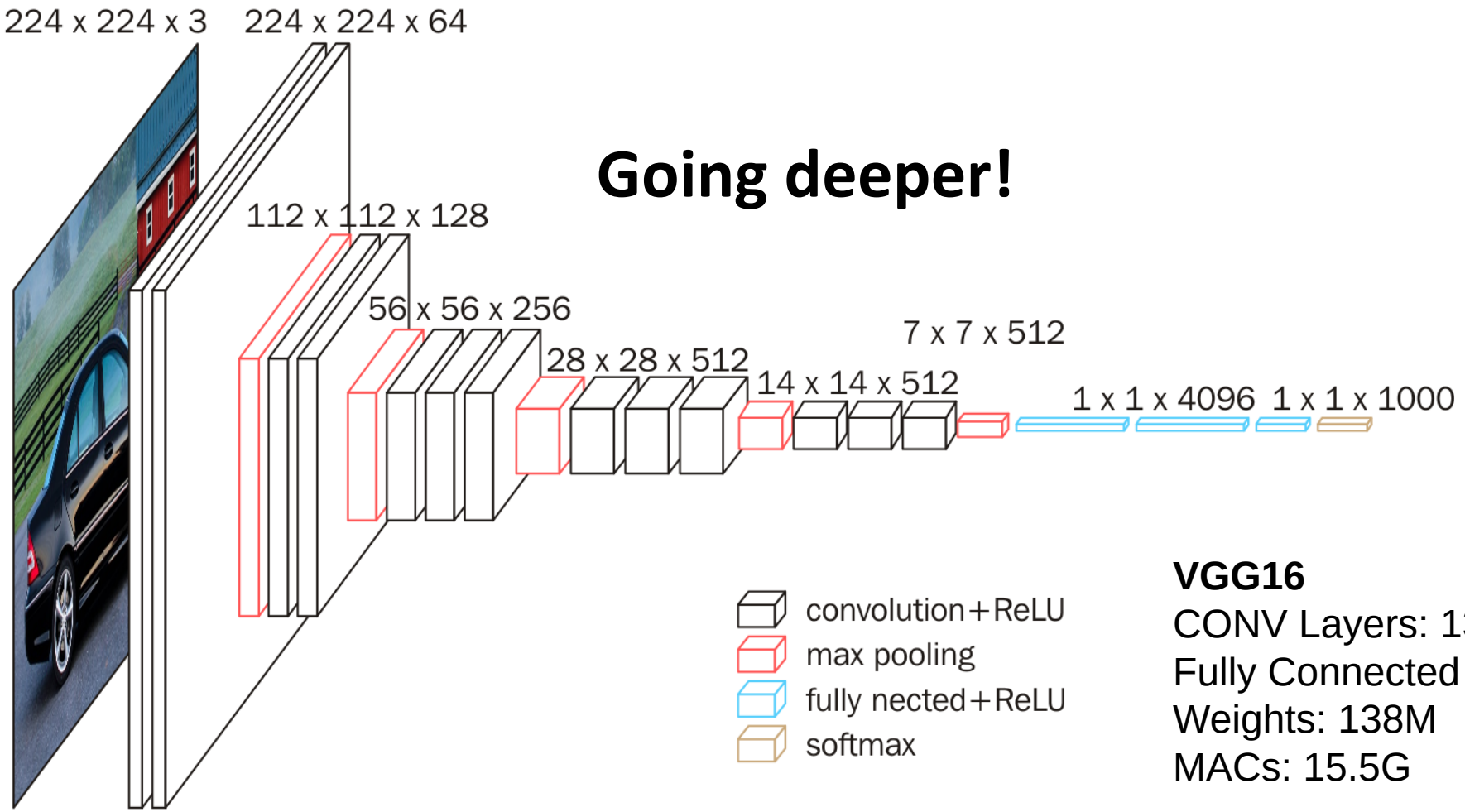
- Example 101 – LeNet, AlexNet
- Popular DNNs
- Popular Applications and Case Studies

Popular DNNs

- LeNet (1998)
- AlexNet (2012)
- OverFeat (2013)
- VGGNet (2014)
- GoogleNet (2014)
- ResNet (2015)



VGGNet



Going deeper!

VGG16
CONV Layers: 13
Fully Connected Layers: 3
Weights: 138M
MACs: 15.5G

K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, May 2015.

VGGNet Series

- # of params (millions)
  - A/A-LRN: 133
  - B: 133
  - C: 134
  - D: 138
  - E: 144
- # of params dominant by fully connected layer
  - $7 \times 7 \times 512 \times 4096 + 4096 \times 4096 + 4096 \times 1000 \approx 123\ M$

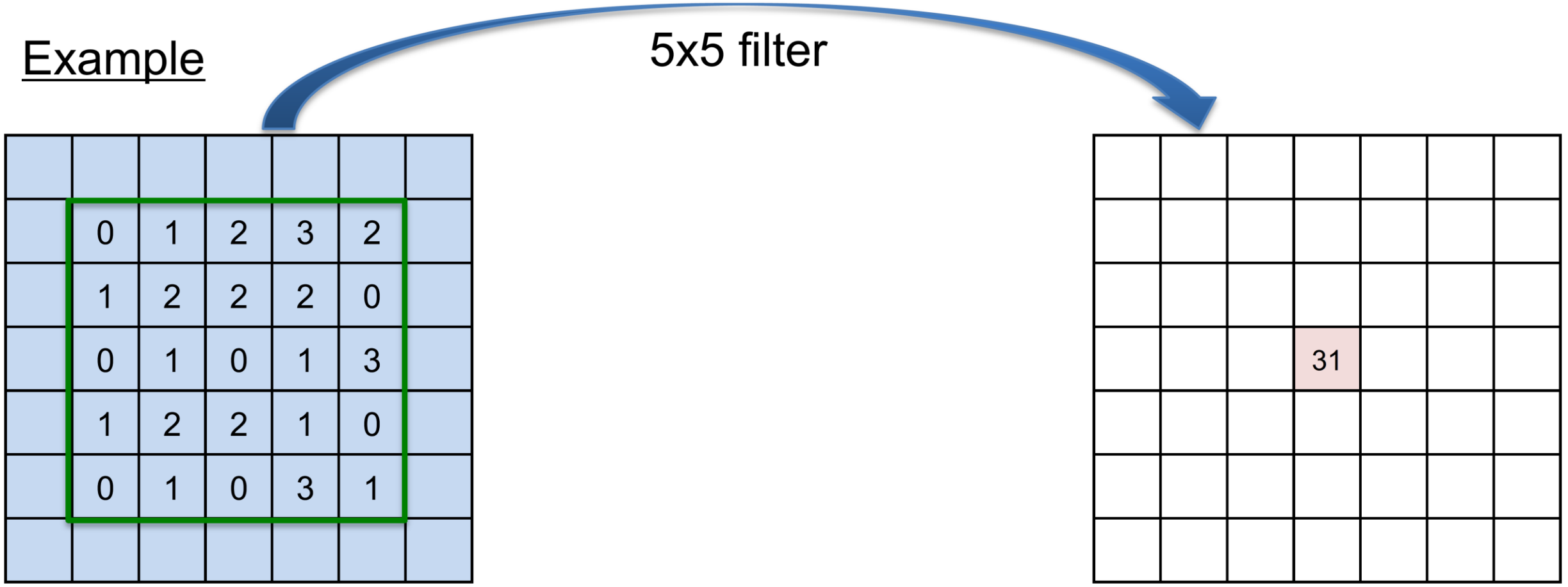
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Smaller CONV Filter

- Deeper network → more weight, more computation
- Use stack of smaller filter (3x3) to cover the same receptive field
  - Fewer parameters
- Non-linear activation inserted between each filter
  - More non-linearity
- [5x5x1 CONV] → [3x3x1 CONV] + [3x3x1 CONV]
  - # of params: 25 → 2x9 = 18
- [7x7x1 CONV] → [3x3x1 CONV] + [3x3x1 CONV] + [3x3x1 CONV]
  - # of params: 49 → 3x9 = 27

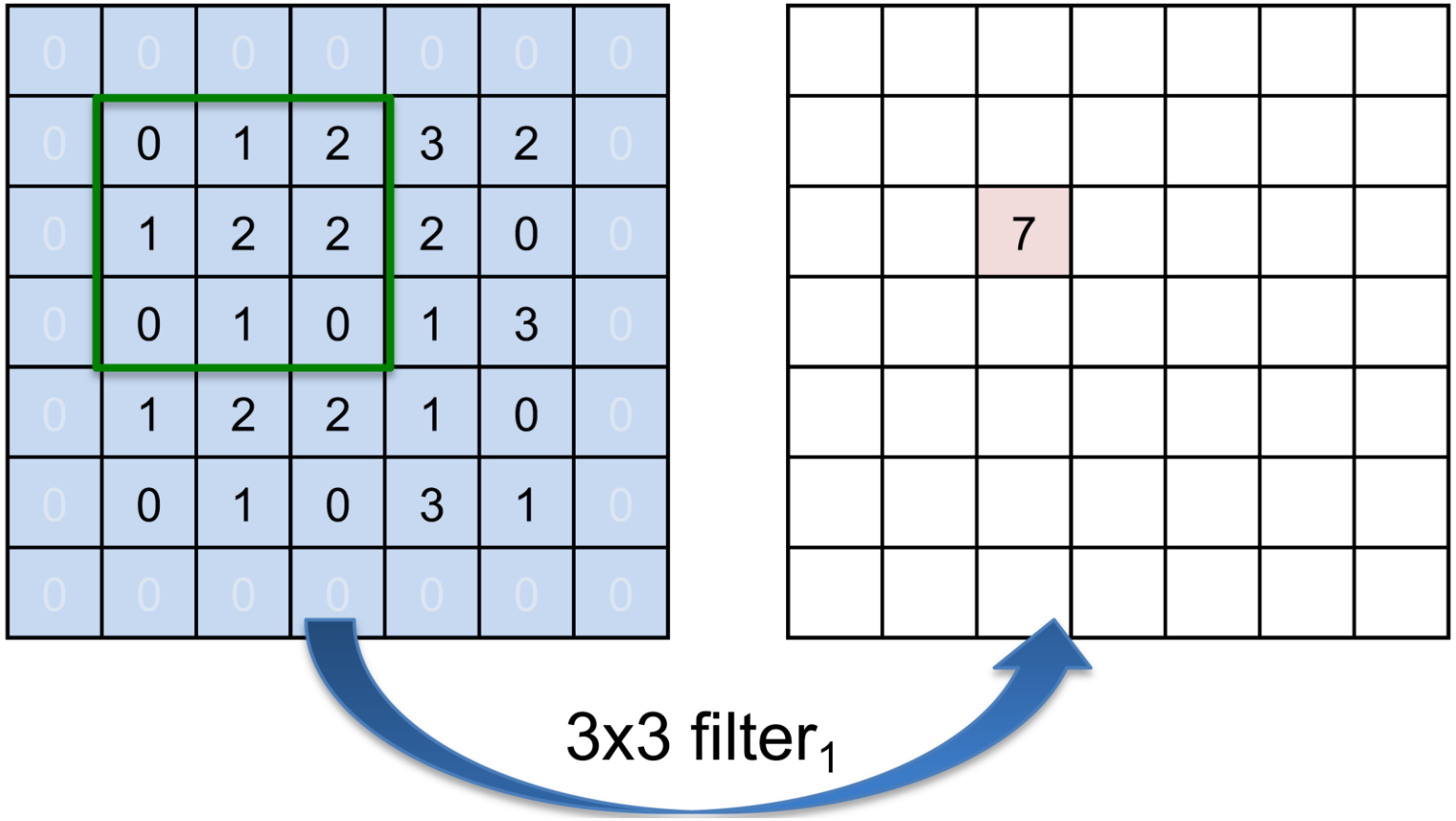
Receptive Field

Example



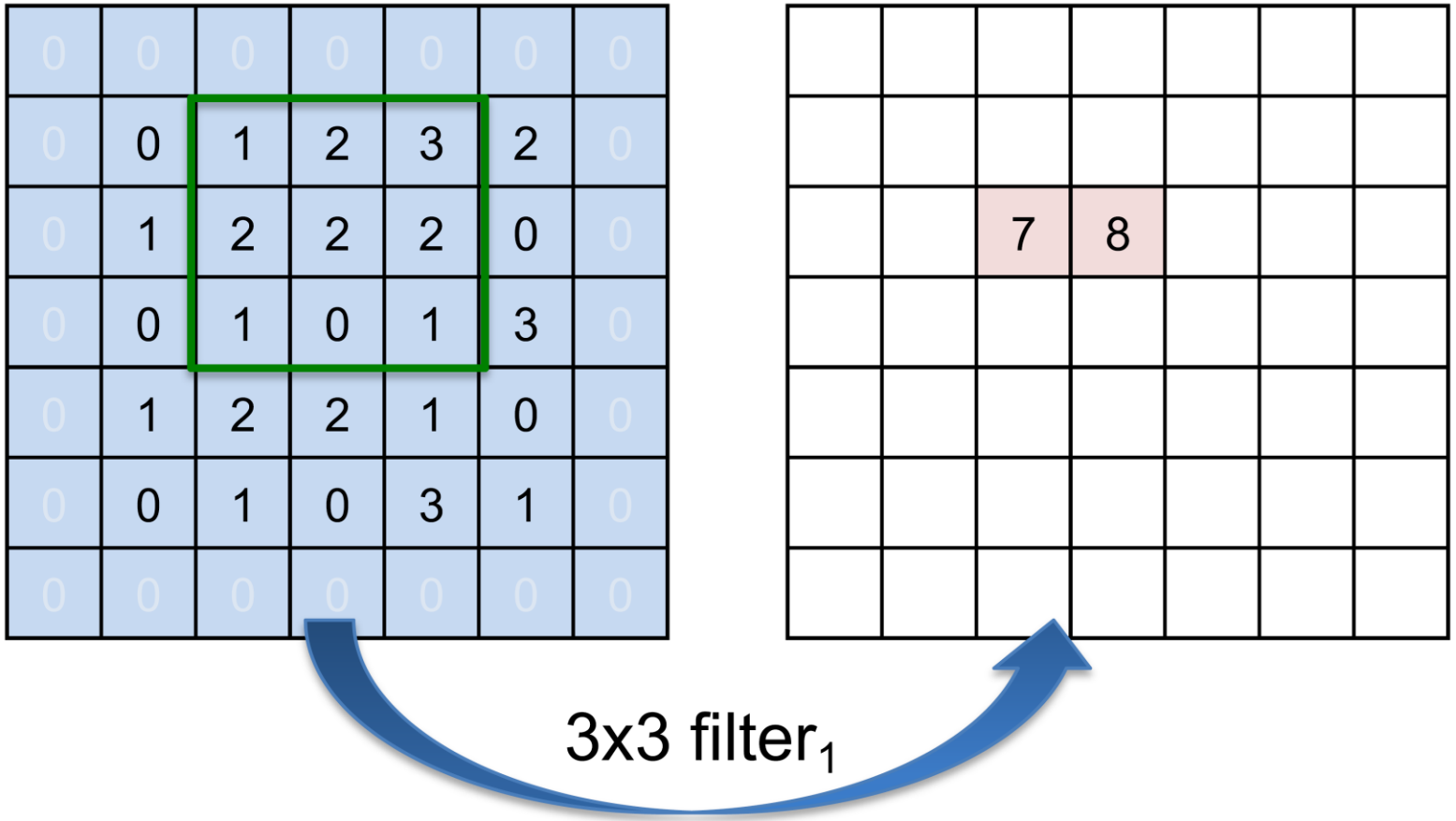
Receptive Field

Example



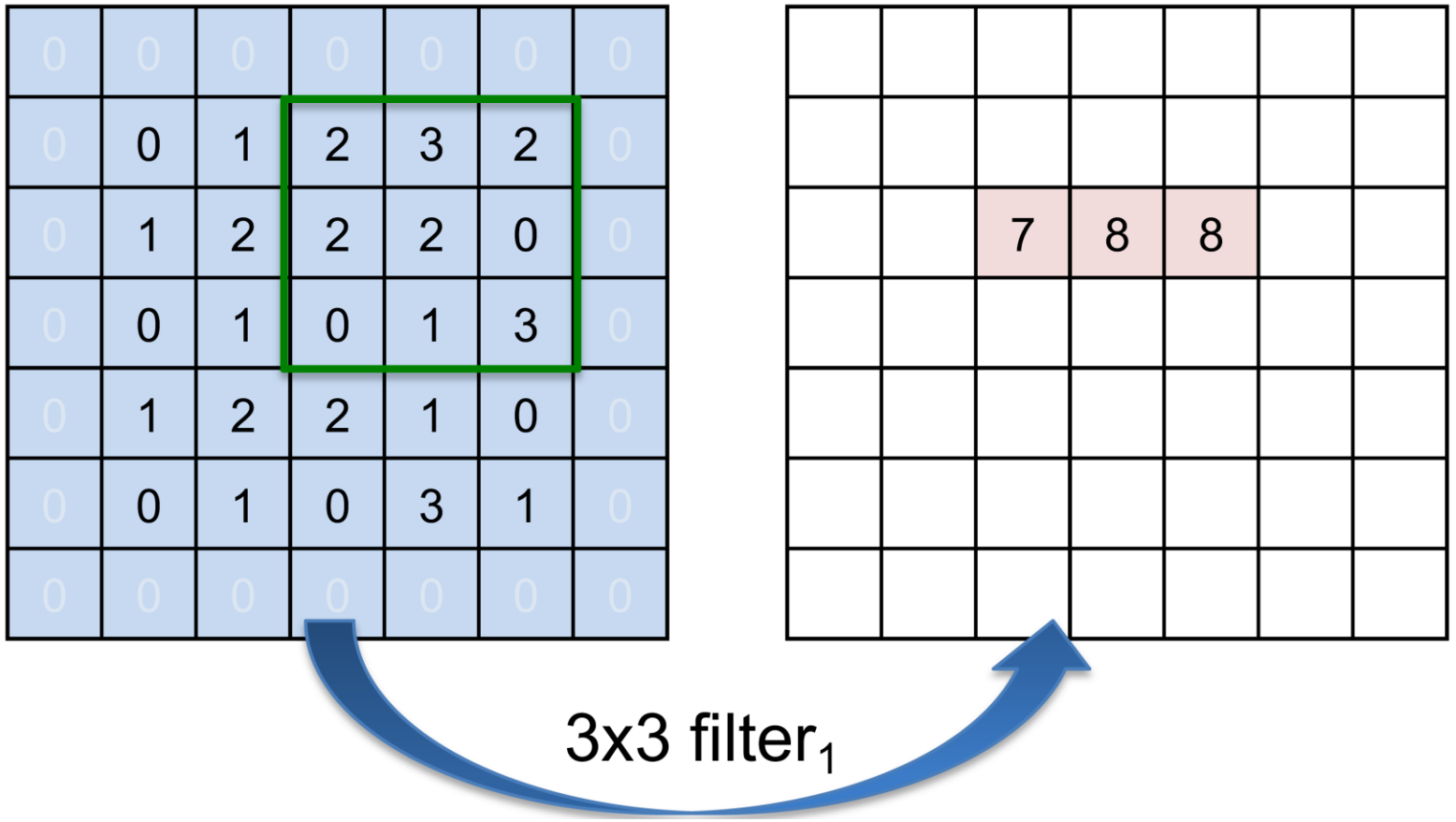
Receptive Field

Example



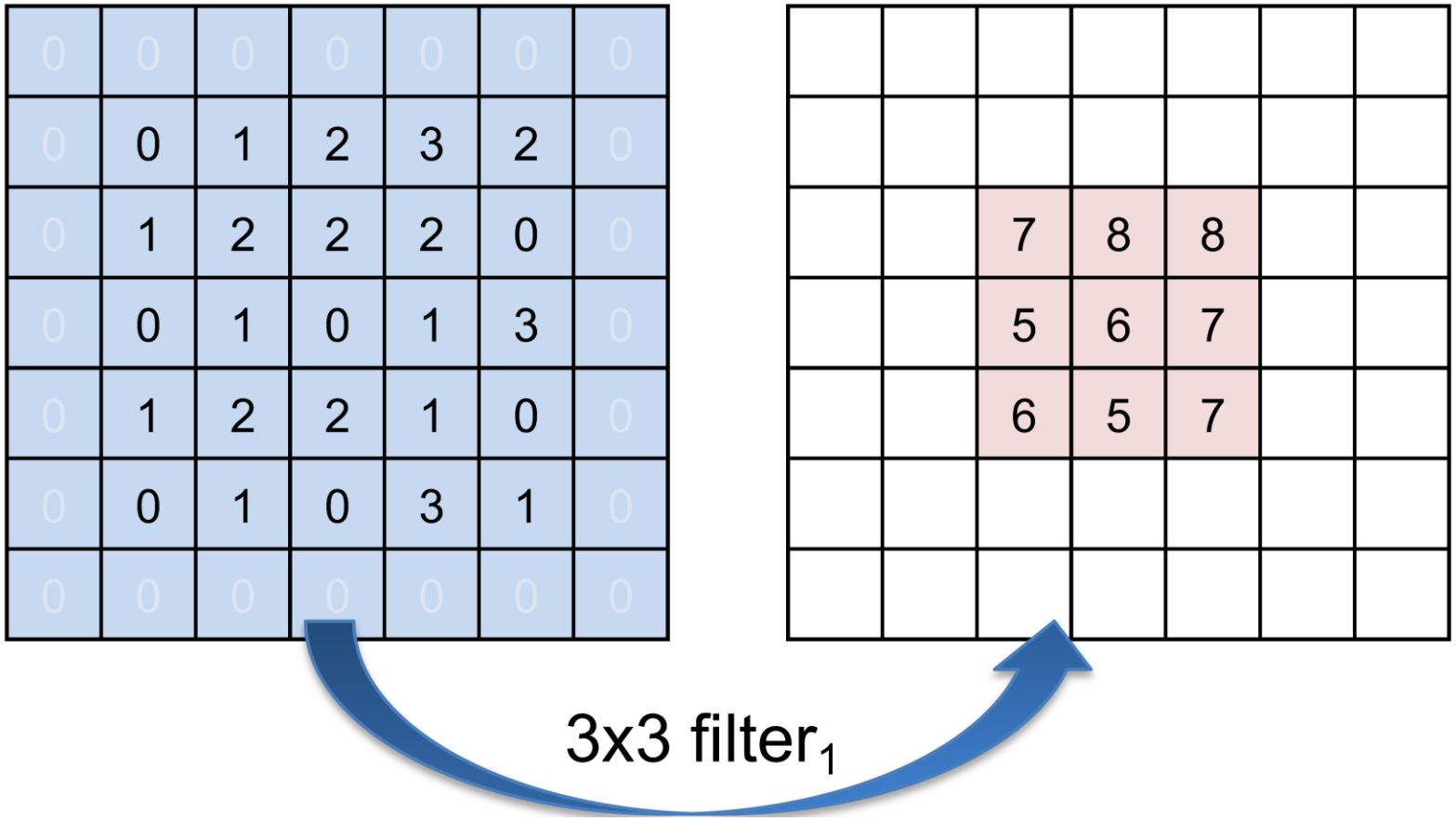
Receptive Field

Example



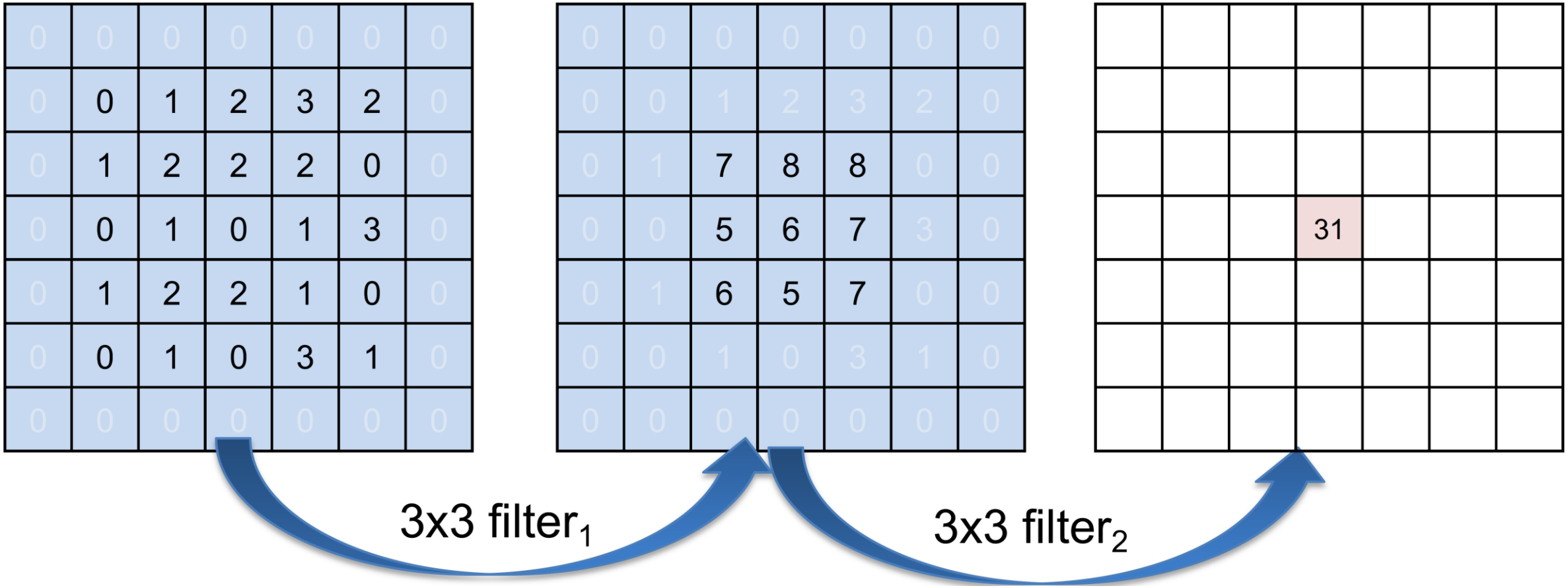
Receptive Field

Example



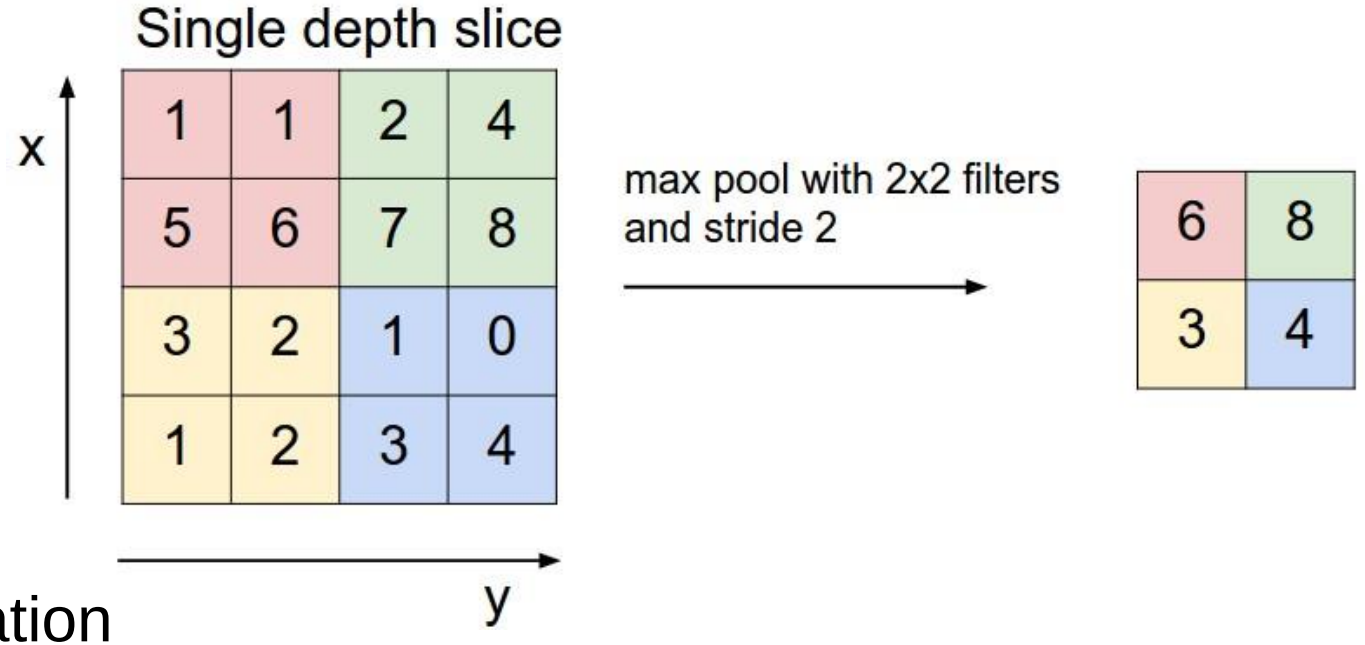
Receptive Field

Example: 5x5 filter (25 weights) → two 3x3 filters (18 weights)



2x2 Pooling

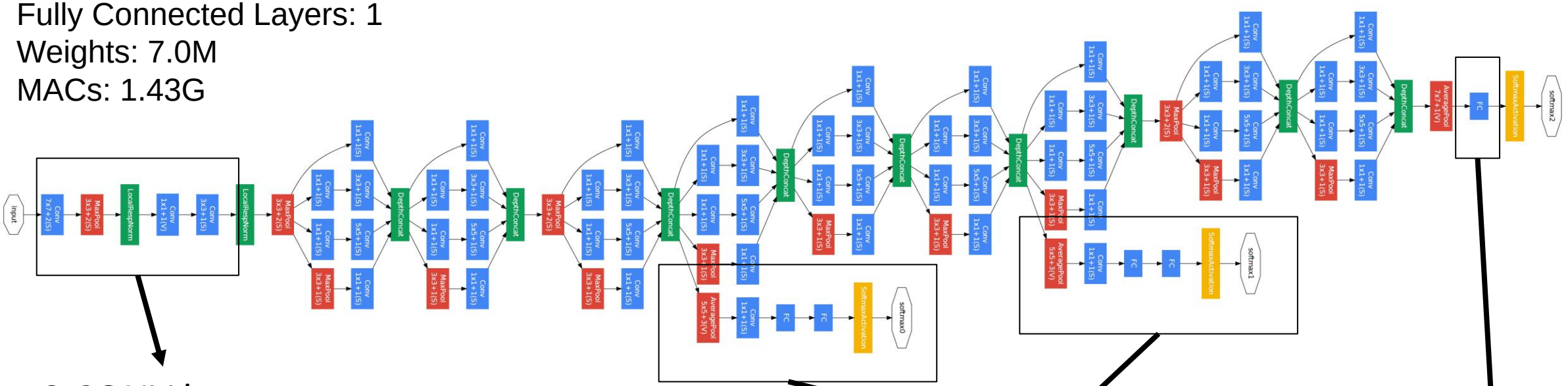
- Alexnet
  - Overlapped Pooling
  - 3x3, Stride 2
- VGGNet
  - Non-overlapped Pooling
  - 2x2, Stride 2
  - Smaller Pooling get more information
  - Proven to be more effective



- 2x2 non overlapped pooling has been widely used in modern DNNs

GoogLeNet/Inception (v1) ILSVRC14 Winner

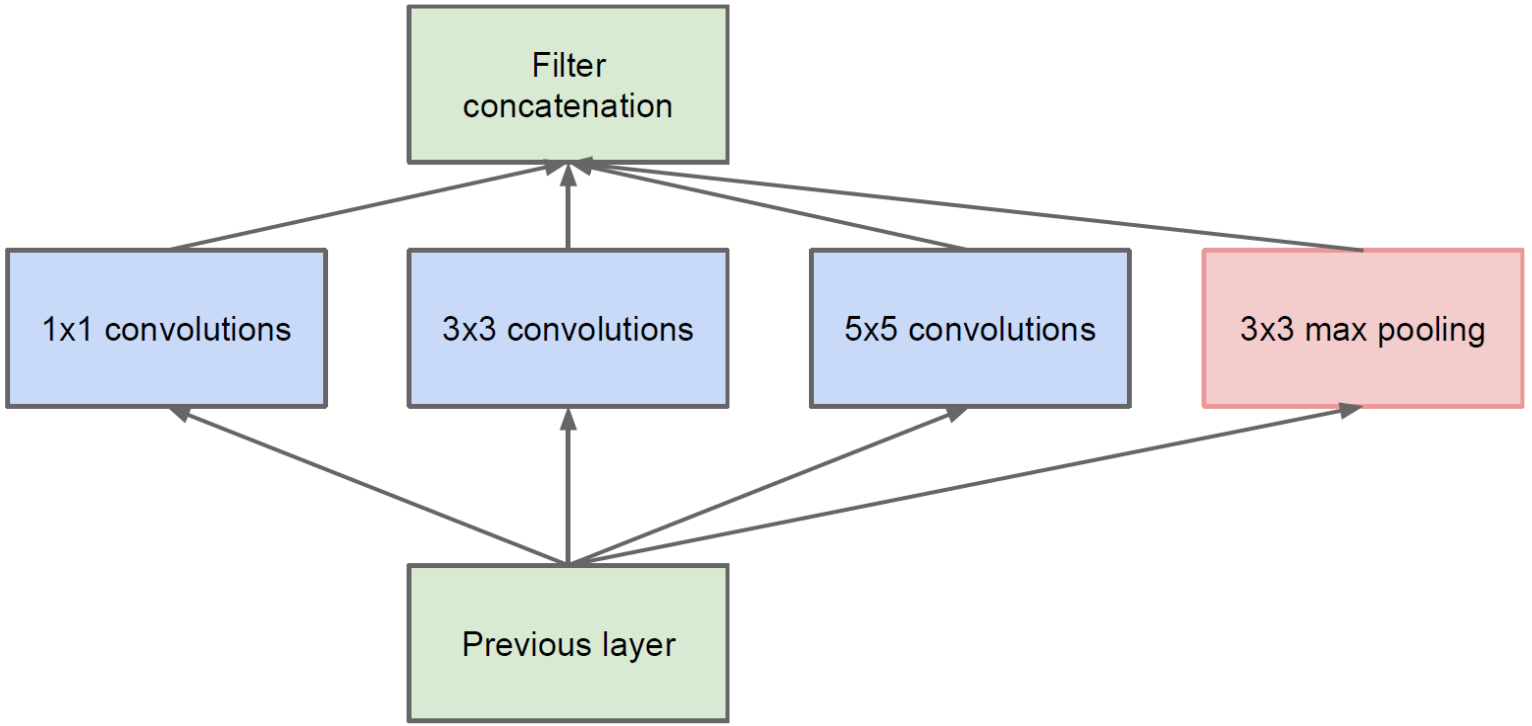
CONV Layers: 22 (depth), 57 (total)
Fully Connected Layers: 1
Weights: 7.0M
MACs: 1.43G



Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

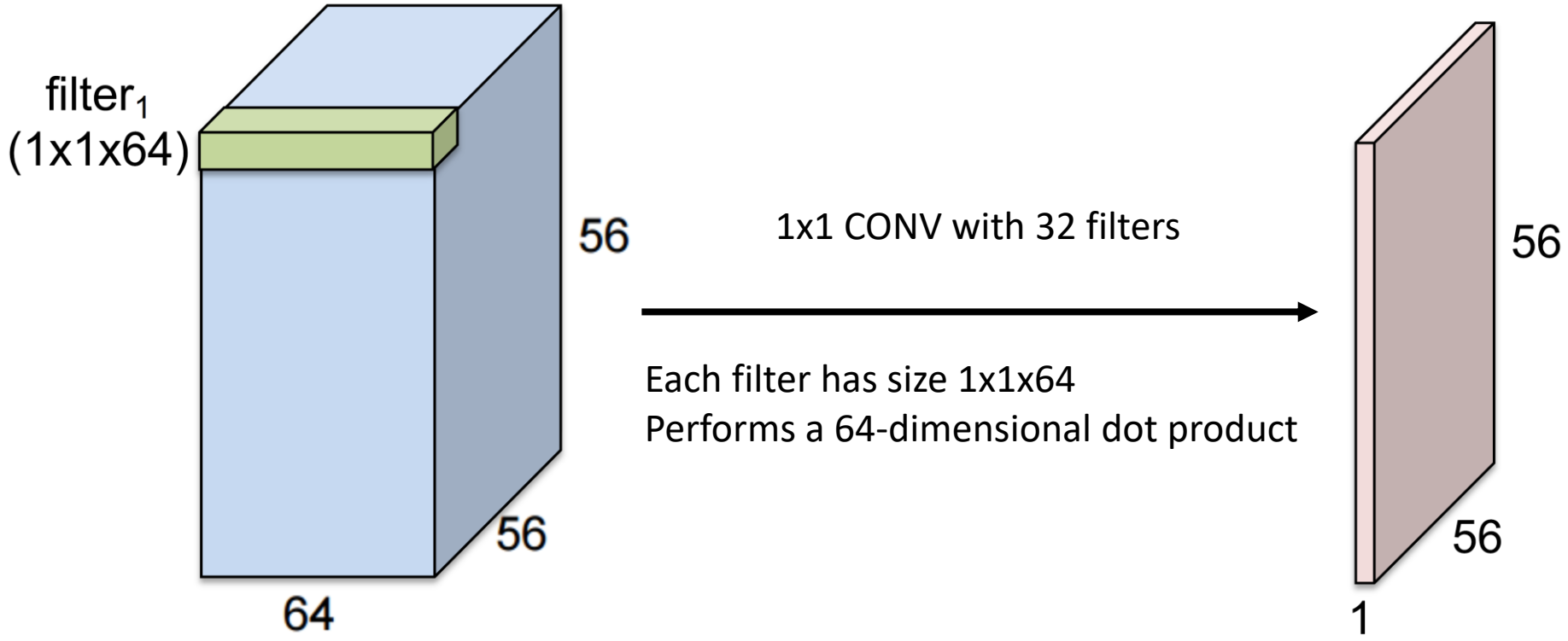
Inception Module

- Parallel filters of different size
  - Processing image at different scales



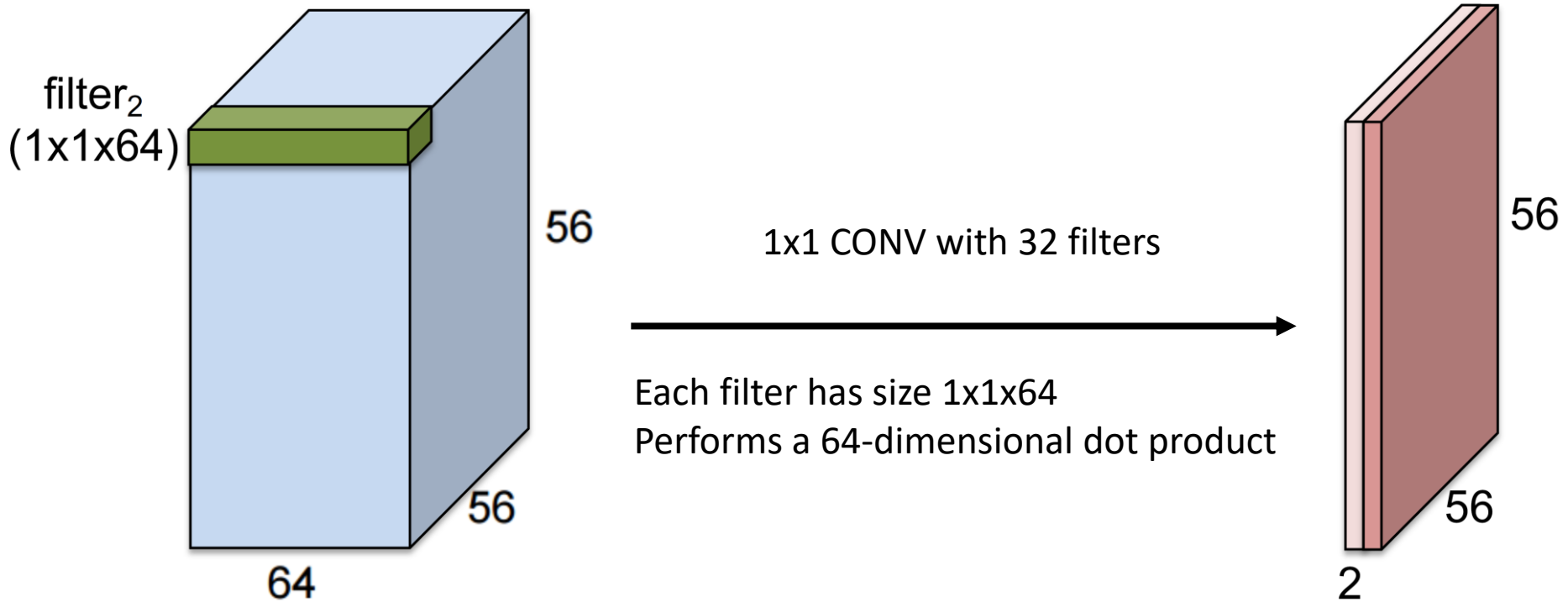
1x1 Dimension Reductions(Bottleneck)

- 1x1 filter
  - Capture cross-channel correlation, but no spatial correlation.
  - Can be used to reduce the number of channels in next layer (bottleneck)



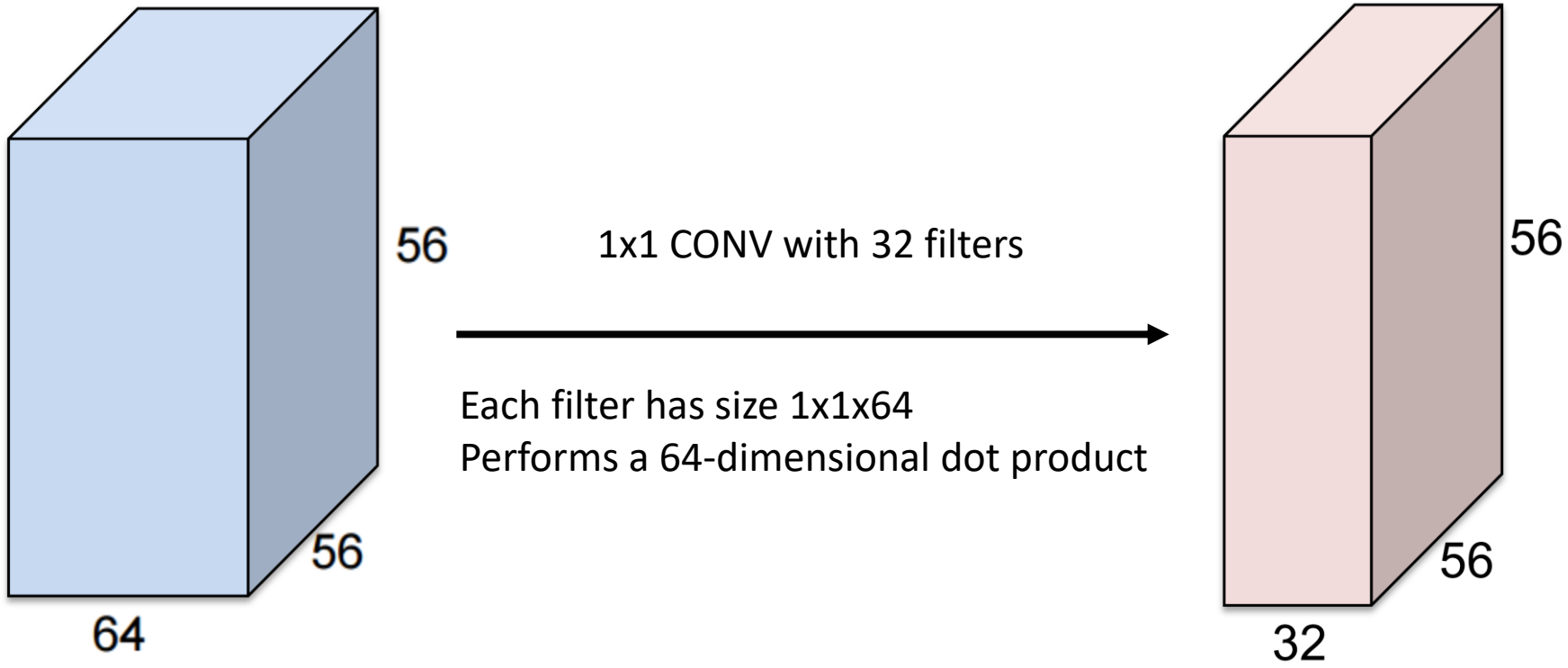
1x1 Dimension Reductions(Bottleneck)

- 1x1 filter
  - Capture cross-channel correlation, but no spatial correlation.
  - Can be used to reduce the number of channels in next layer (bottleneck)



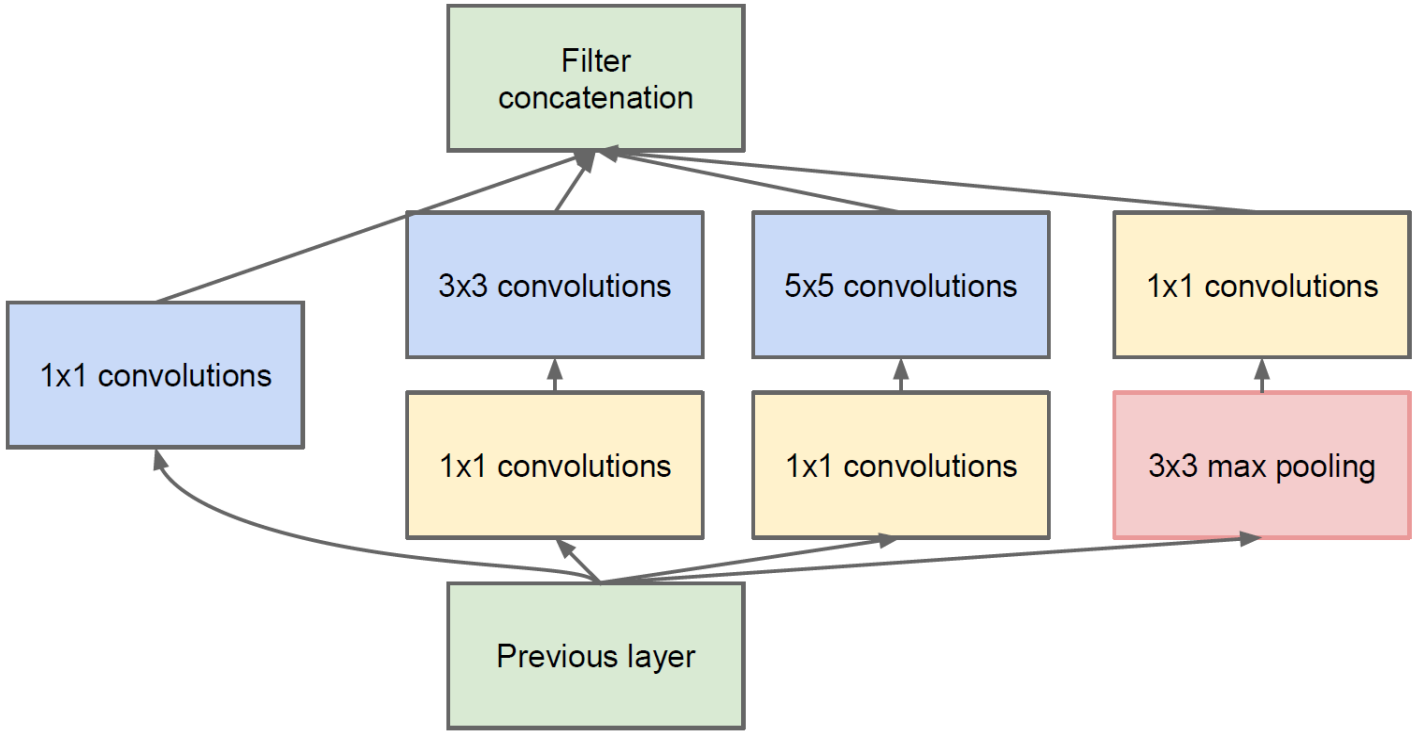
1x1 Dimension Reductions(Bottleneck)

- 1x1 filter
  - Capture cross-channel correlation, but no spatial correlation.
  - Can be used to reduce the number of channels in next layer (bottleneck)



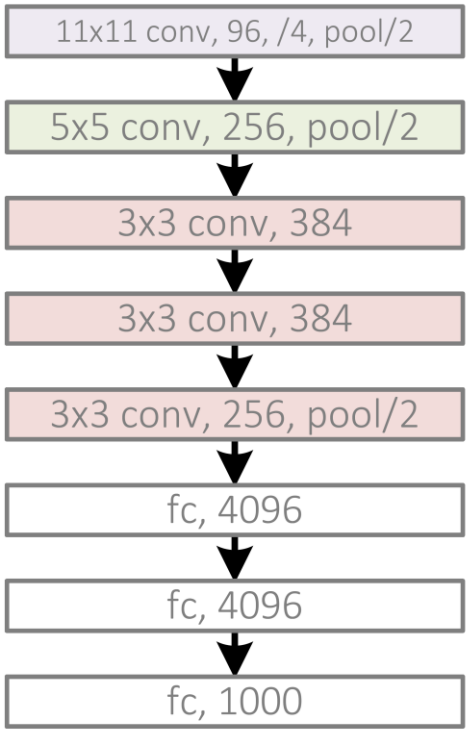
Inception Module with Bottleneck

- Apply bottleneck before ‘large’ convolution filters.
  - Reduce weights and computation
  - Number of multiplications reduced from 854M → 358M



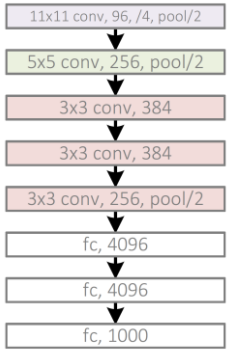
Revolution of Depth

AlexNet, 8 layers

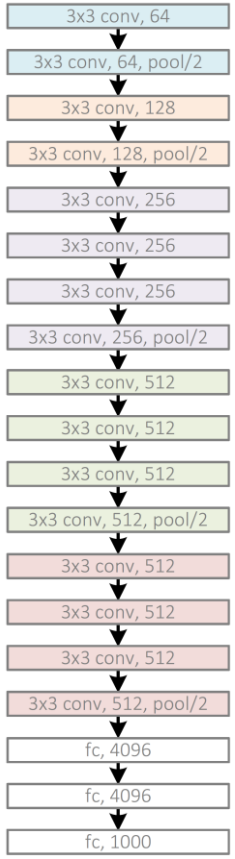


Revolution of Depth

AlexNet
8 layers



VGGNet
19 layers



GoogLeNet
22 layers

Revolution of Depth

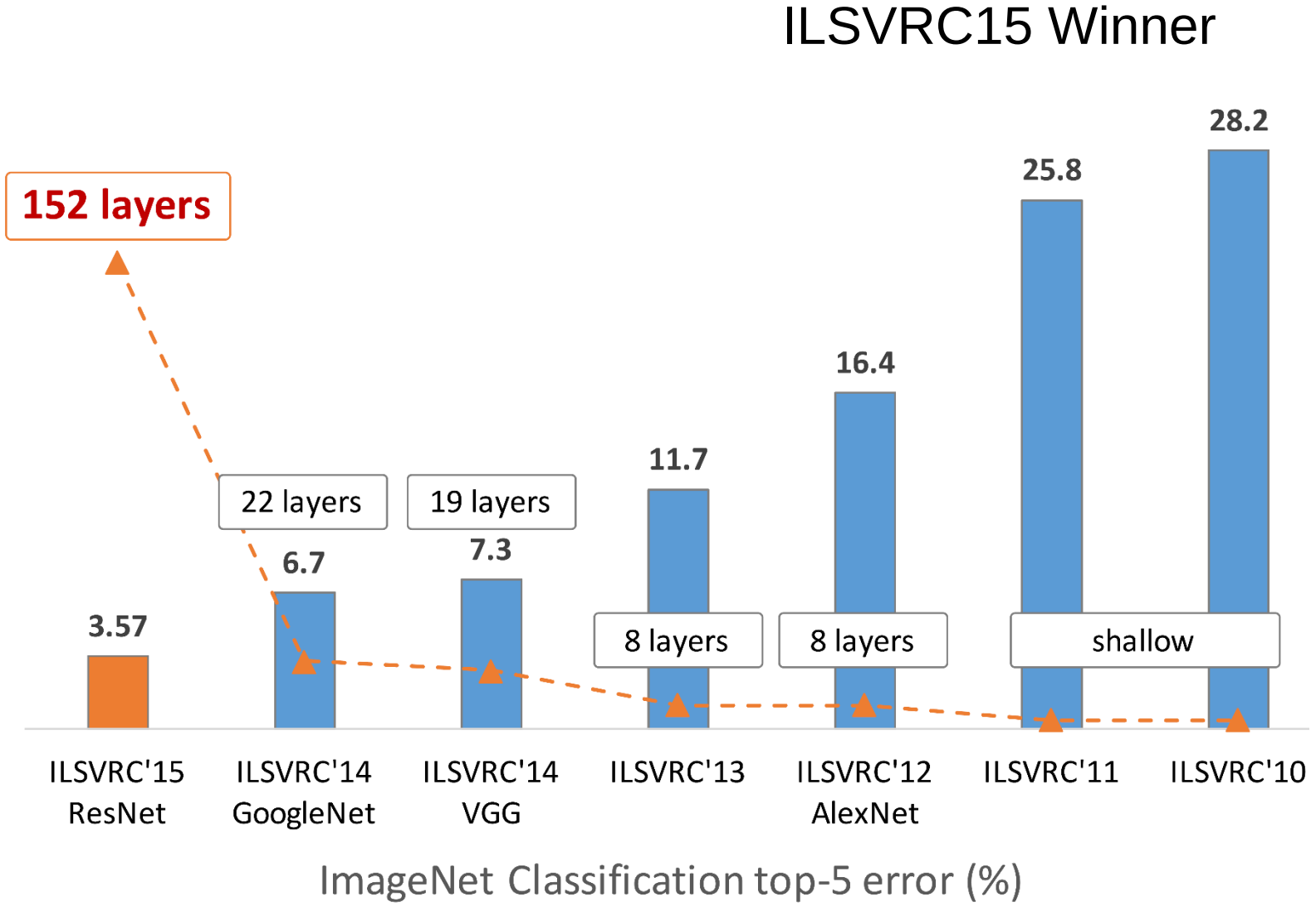
AlexNet
8 layers

VGGNet
19 layers

GoogLeNet
22 layers

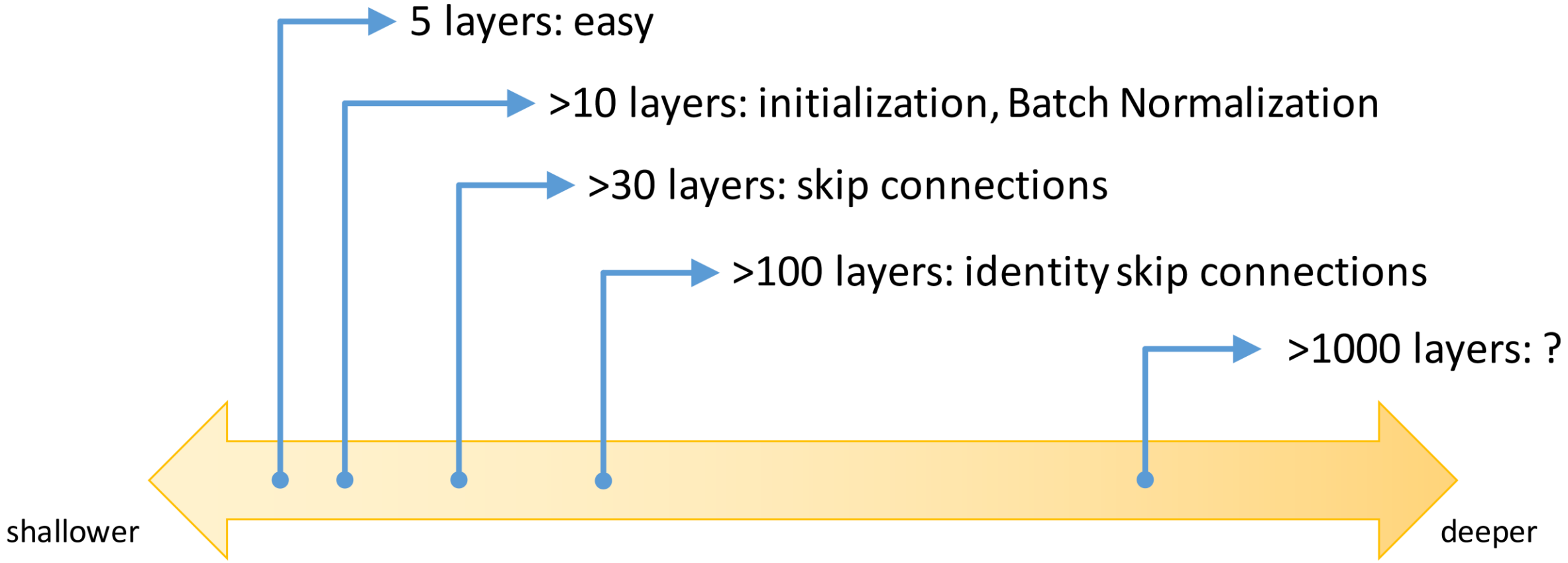
ResNet
152 layers

ResNet – Beyond Human Level!



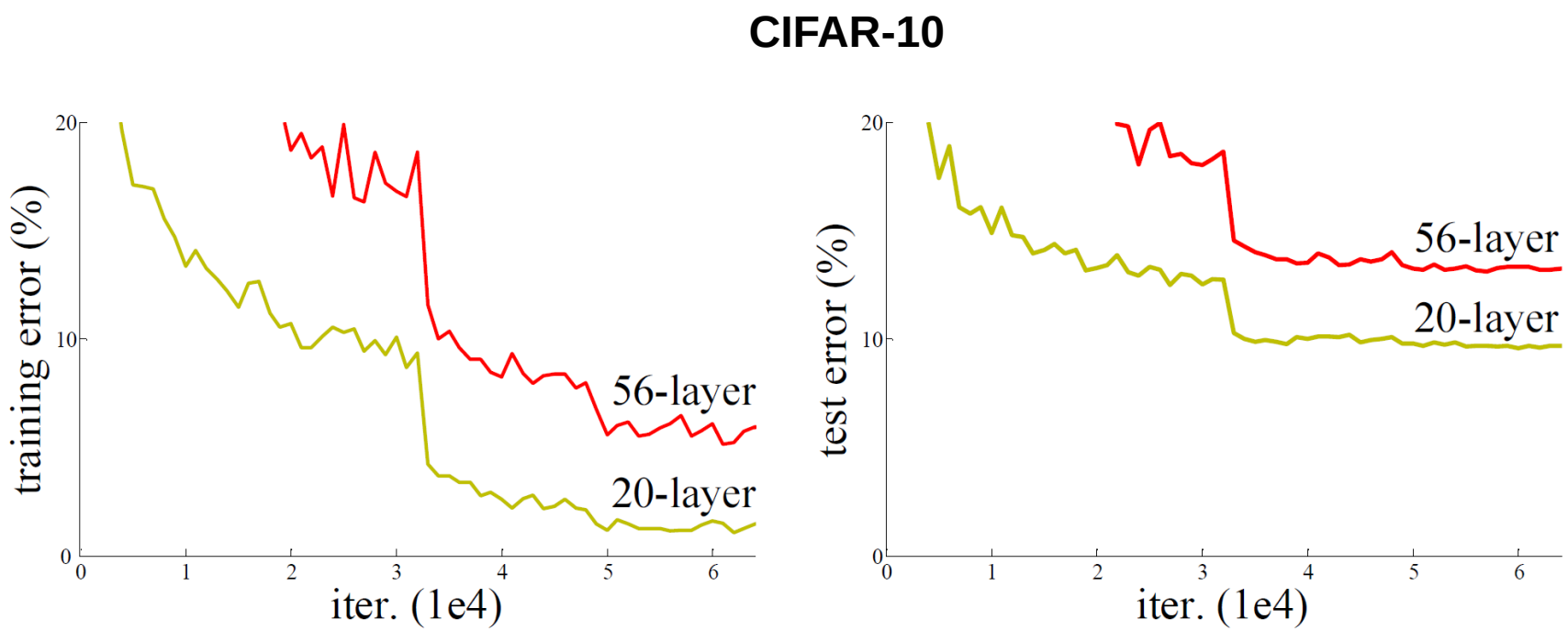
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Spectrum of Depth



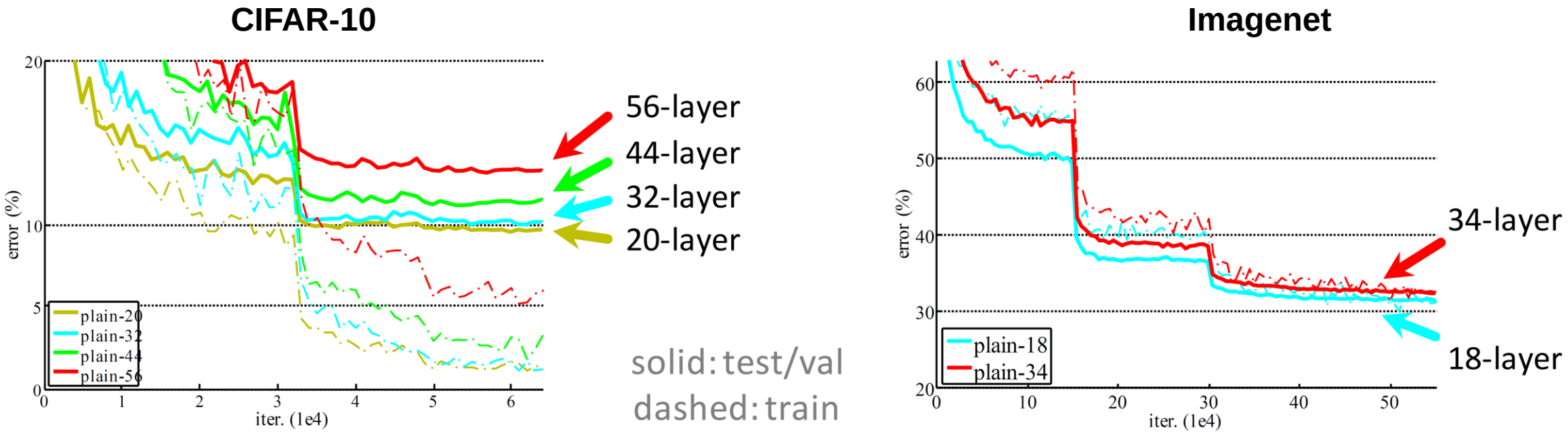
Going Deeper - Simply Stacking Layers?

- Stacking 3x3 CONV layers
- 56-layer net has higher training error and test error than 20-layer net



Going Deeper - Simply Stacking Layers?

- Overly deep DNNs have **higher training error**
- A general phenomenon, observed in many datasets



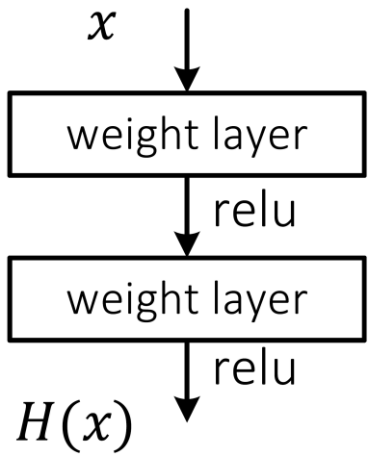
Going Deeper

- Deeper network
  - Richer solution space
  - Should not have **higher training error**
- Why higher training error?
  - **Optimization difficulties**
  - Vanishing gradient
  - Solvers cannot find the solution when going deeper

Deep Residual Learning

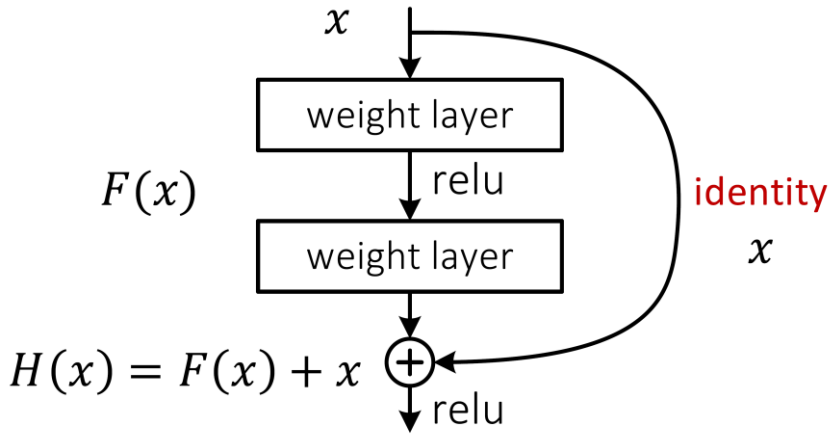
Plain net

$H(x)$ is any desired mapping



Hope the 2 weight layers fit $H(x)$

Residual



Hope the 2 weight layers fit $F(x)$

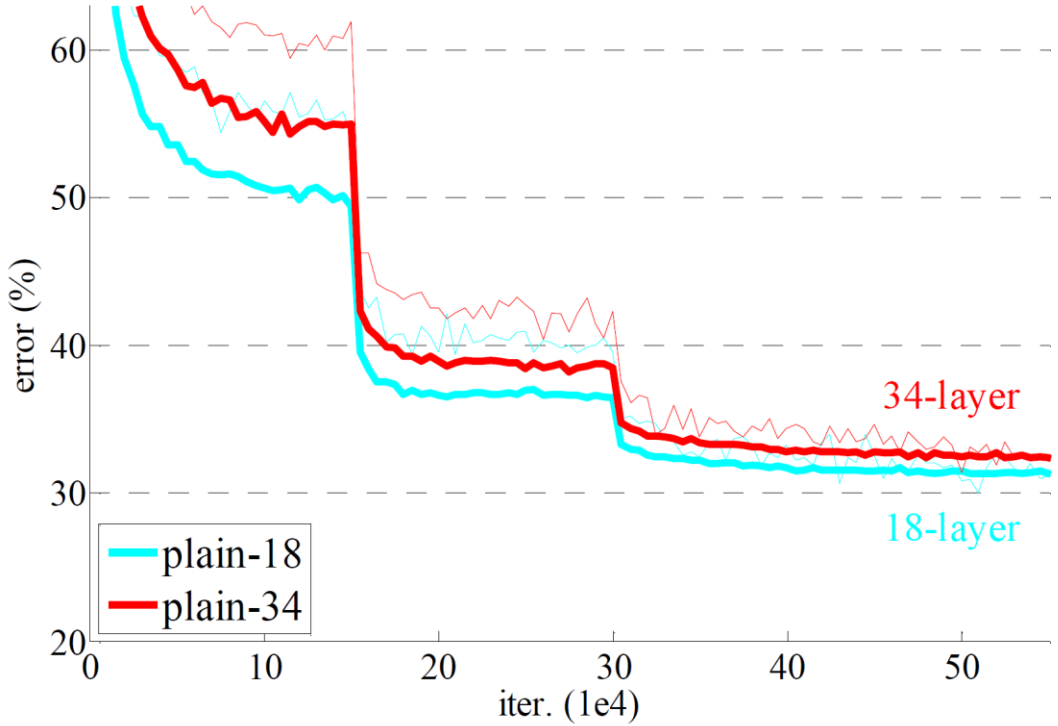
Let $H\ x = F\ x + x$

$F(x)$ is a residual mapping w.r.t. identity

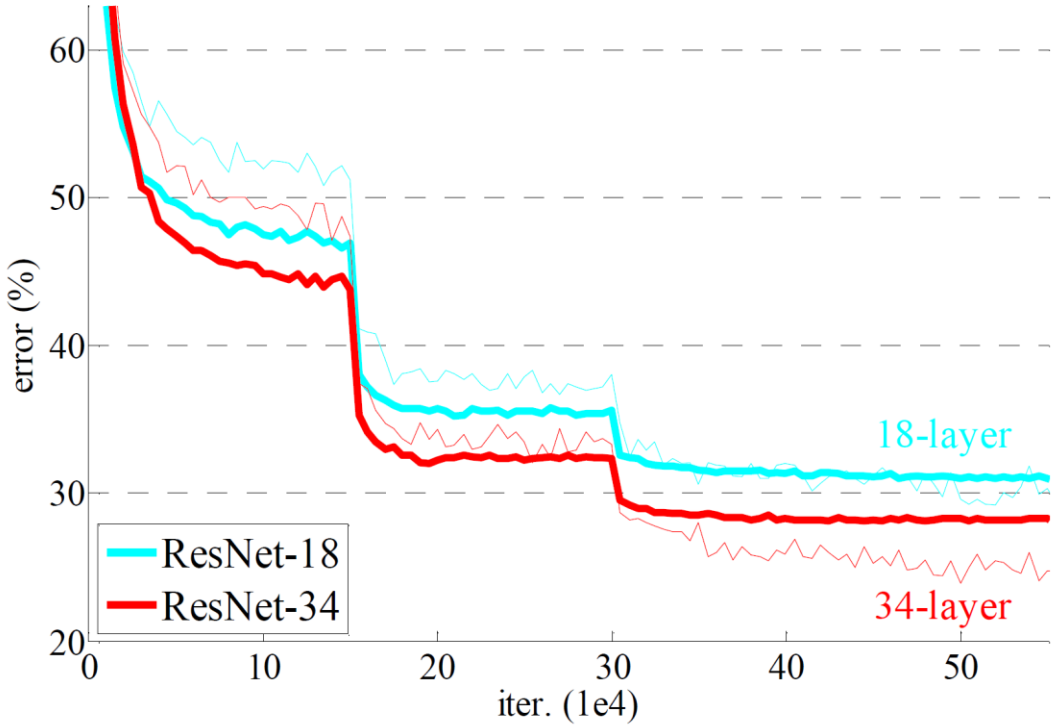
- If identity were optimal, easy to set weights as 0
- If optimal mapping is closer to identity, easier to find small fluctuations

Going Deeper with Shortcut

Without shortcut

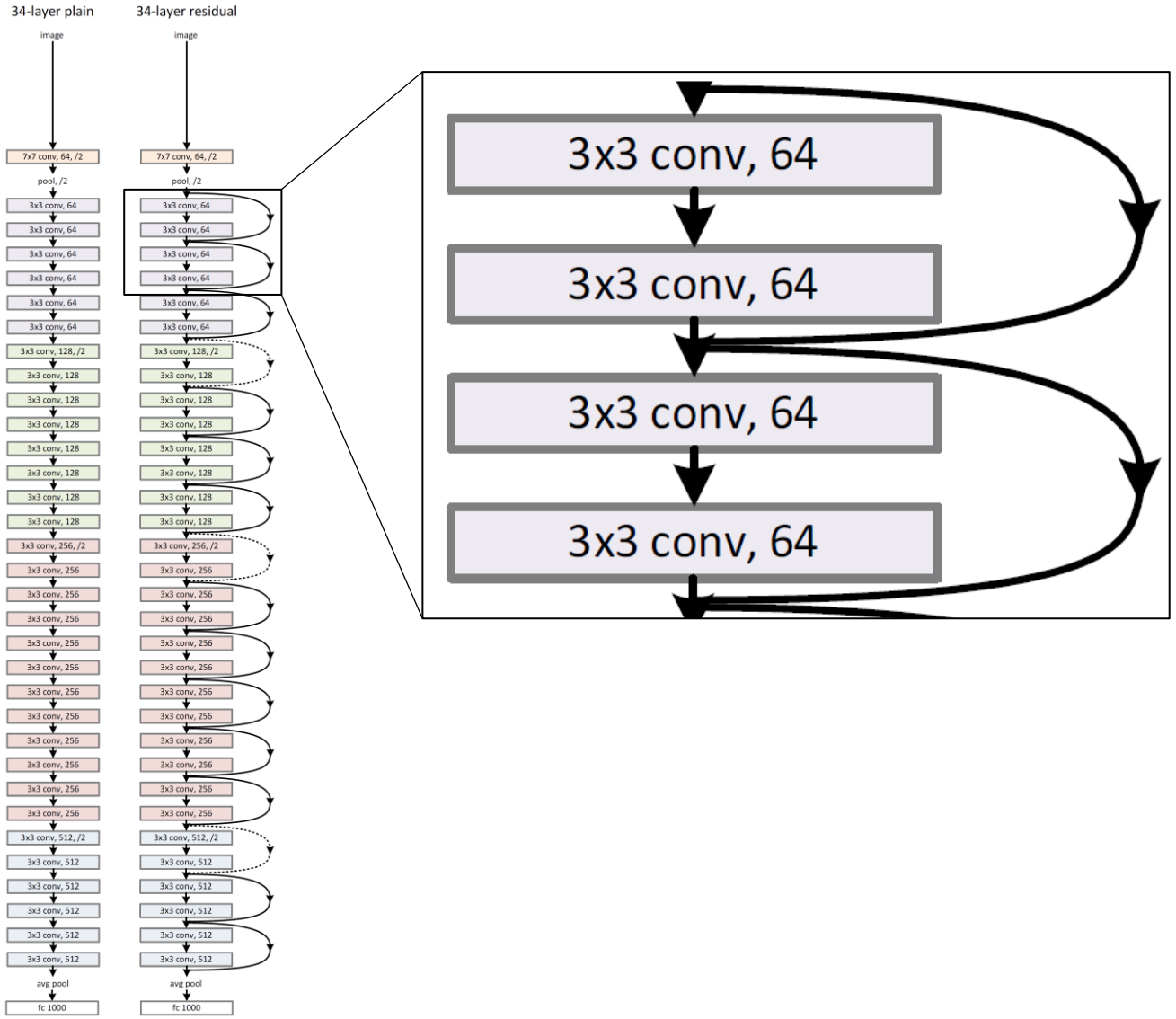


With shortcut



Thin curves denote training error, and bold curves denote validation error.

ResNet - Architecture

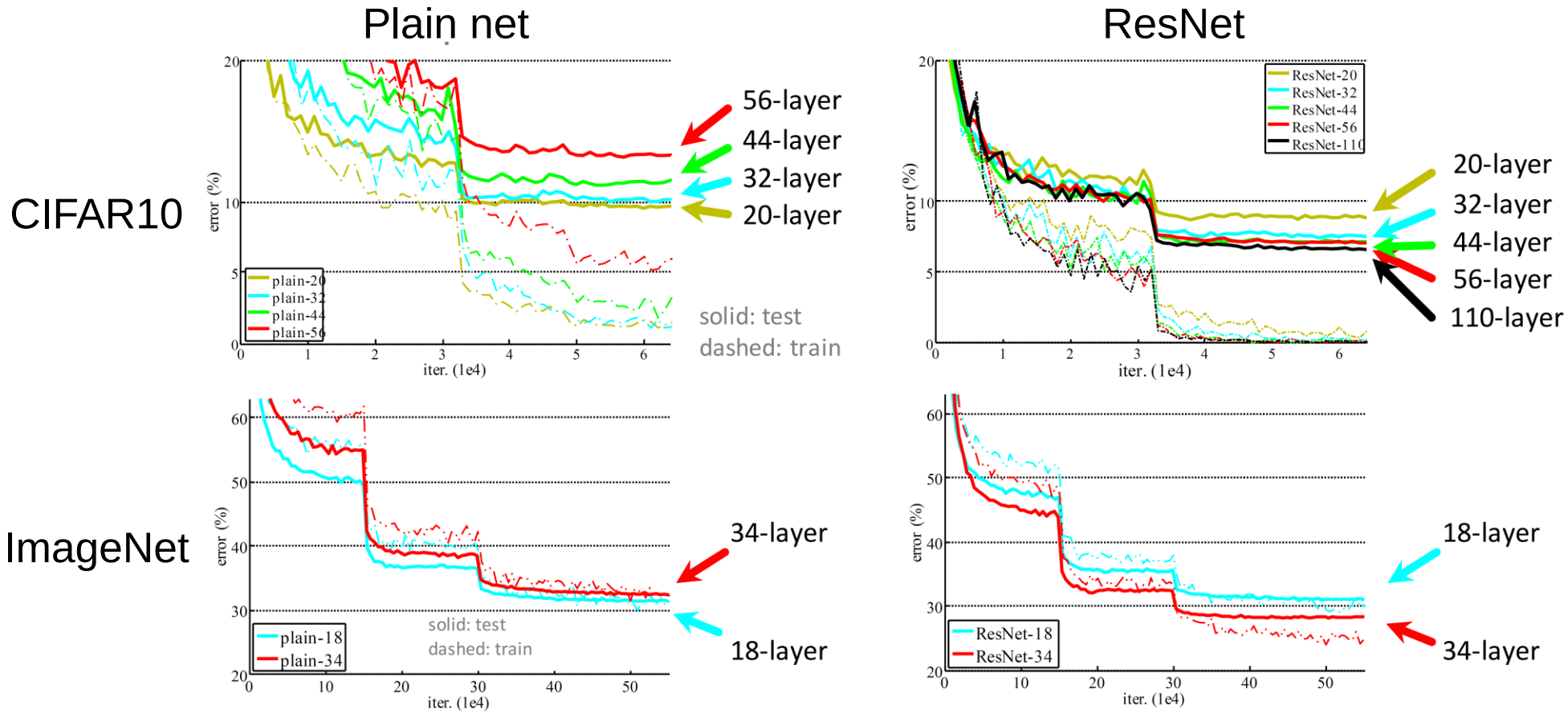


- All 3x3 CONV
- spatial size /2 => # filters x2
  - same complexity per layer
- Simple design; **just deep!**

- Can be **trained from scratch**

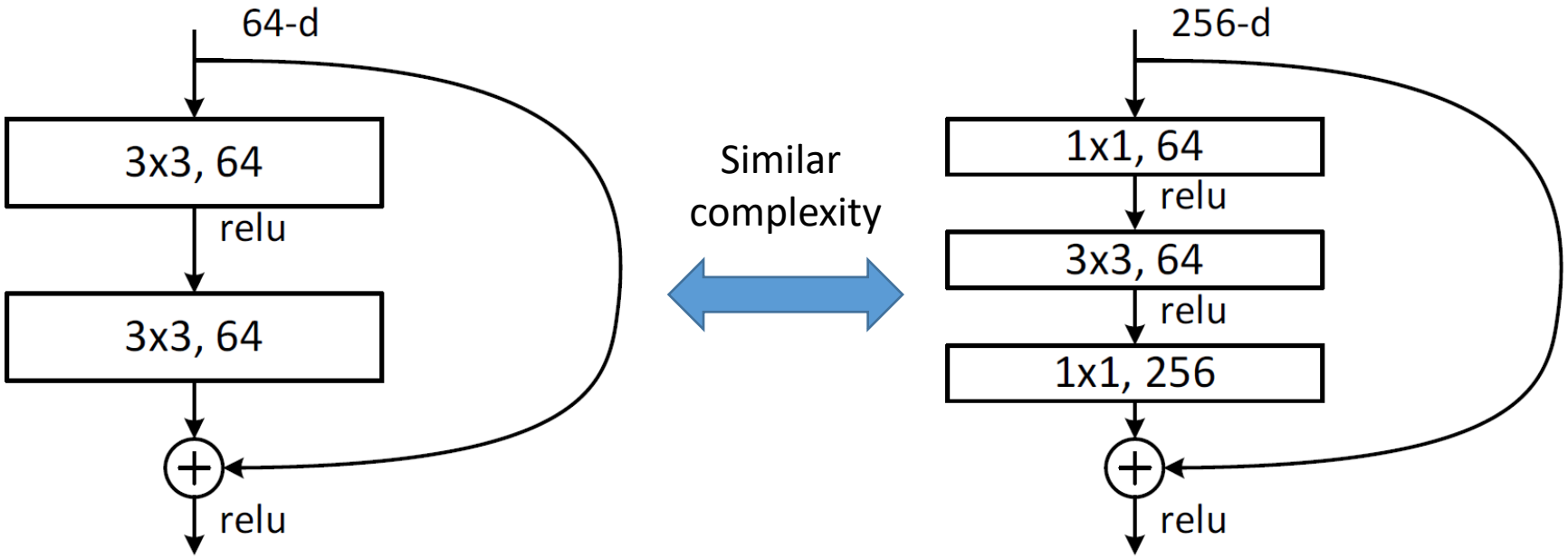
Going Deeper with ResNet – Examples

- Deeper ResNets have lower training error, and also lower test error



ResNet: Bottleneck

- Apply 1x1 bottleneck to reduce computation and size
- Also makes network deeper (ResNet-34 → ResNet-50)



ResNet Series

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Summary

Metrics	LeNet-5	AlexNet	VGG-16	GoogLeNet(v1)	ResNet-50
Top-5 error	n/a	16.4	7.4	6.7	5.3
Input Size	28x28	227x227	224x224	224x224	224x224
# of CONV Layers	**2**	**5**	**16**	**21(depth)**	**49**
Filter Sizes	5	3,5,11	3	1,3,5,7	1,3,7
# of Channels	1,6	3-256	3-512	3-1024	3-2048
# of Filters	6,16	96-384	64-512	64-384	64-2048
Stride	1	1,4	1	1,2	1,2
# of Weights	2.6k	2.3M	14.7M	6.0M	23.5M
# of MACs	283k	666M	15.3G	1.43G	3.86G
# of FC layers	**2**	**3**	**3**	**1**	**1**
# of Weights	58k	58.6M	124M	1M	2M
# of MACs	58k	58.6M	124M	1M	2M
Total Weights	**60k**	**61M**	**138M**	**7M**	**25.5M**
Total MACs	**341k**	**724M**	**15.5G**	**1.43G**	**3.9G**

Pitfalls

- Fewer number of operation → Higher throughput
  - Number of operations doesn't directly translate to throughput
  - Just provide a rough complexity estimation
  - Different network architecture → different throughput
- Fewer weights → Lower power/energy consumption
  - Number of weights and operations doesn't directly translate to power/energy consumption
  - Just provides an indication of **storage cost** for inference
  - Different spatial access → different bandwidth/energy
- **Understanding the underlying hardware is important for evaluating the impact of these "efficient" DNN models**

Outline

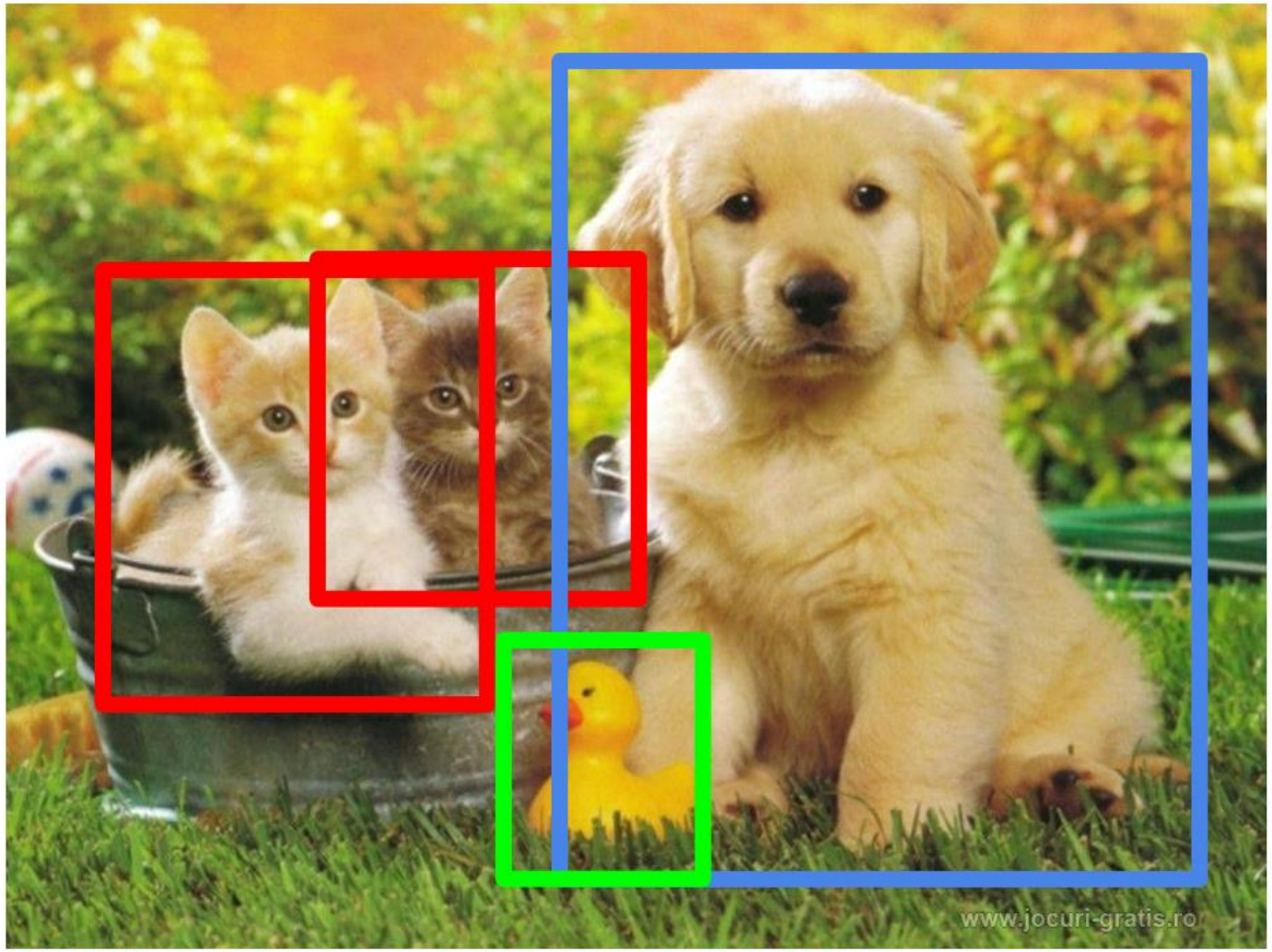
- Example 101 – LeNet, AlexNet
- Popular DNNs
- Popular Applications and Case Studies

Beyond Classification

- Detection
- Segmentation
- Regression
- Pose Estimation
- Matching Patches
- Tracking
- Synthesis
- Style transfer
- Network Architecture Search
- Health Care
- Auto Pilot
- Natural Language Processing
- Electronic Control Unit
- Smart Sport
- Gaming
- Numerical Analysis
- Automated Optical Inspection

to name a few…

Object Detection



Treat as Regression Problem

- DOG (x,y,w,h)
- CAT (x,y,w,h)
- CAT (x,y,w,h)
- DUCK (x,y,w,h)

- 16 numbers

Treat as Regression Problem

- DOG (x,y,w,h)
- CAT (x,y,w,h)

- 8 numbers

Treat as Regression

- Need variable number of output for different image

Treat as Classification Problem

CAT? NO

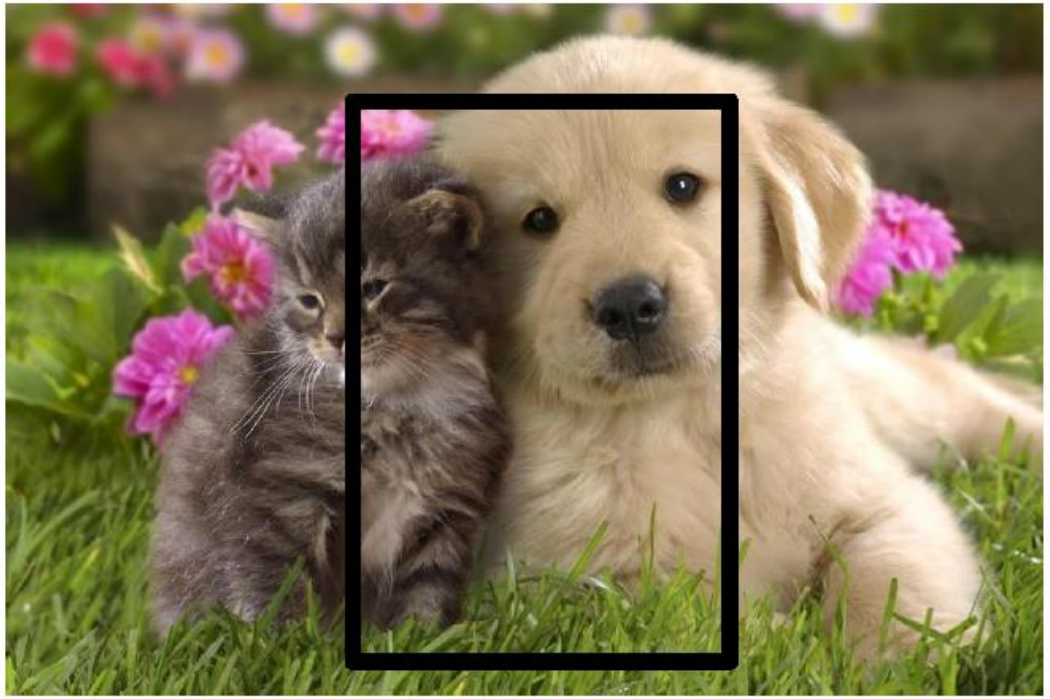
DOG? NO

Treat as Classification Problem

CAT? YES!

DOG? NO

Treat as Classification Problem



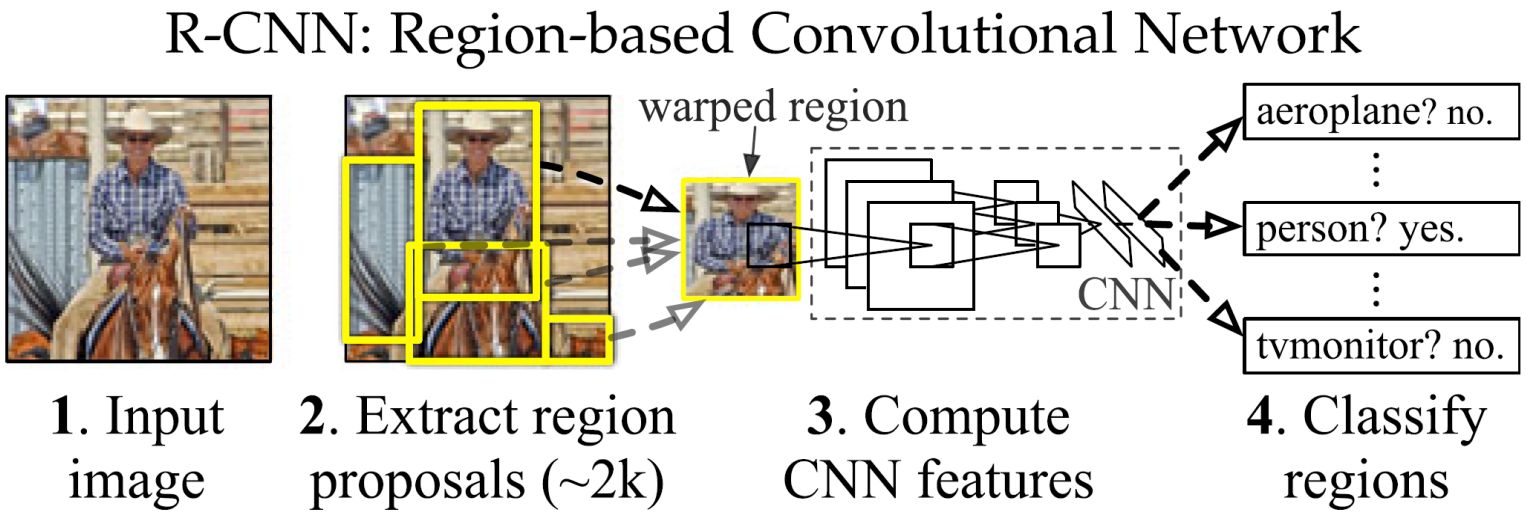
CAT? NO

DOG? NO

Treat as Classification Problem

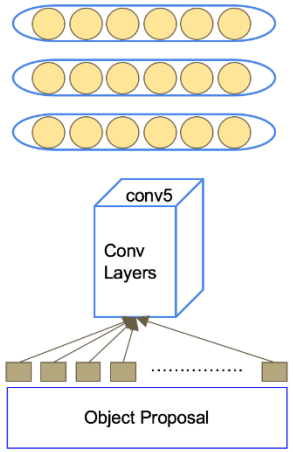
- Need to test many positions and scales
  - Sliding Window
- Need a fast classifier
  - CNN is too slow

R-CNN

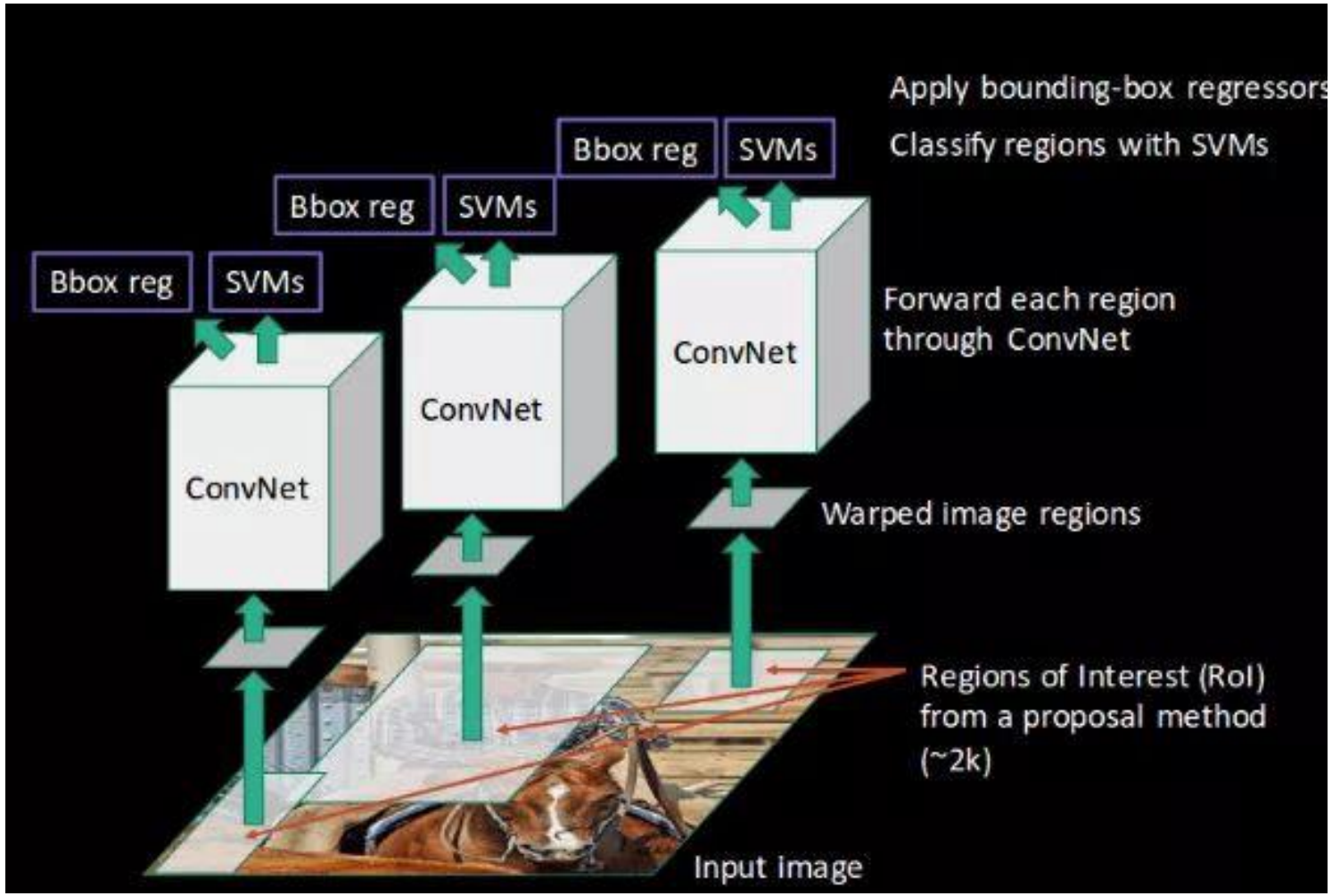


- (1) Use Selective Search to get the region proposals

- (2) For each proposal, use CNN to get the features(from the fc7 of AlexNet)
- (3) Input the feature to all of the classs-specified SVM to predict scores
- (4) Rank the scores to get the final solution



R-CNN



R. Girshick, J. Donahue, T. Darrell and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142-158, 1 Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.

Fast R-CNN

- Traditional CNN needs input be fixed size images(Alexnet 224x224)
  - Resizing image may result in loss of information

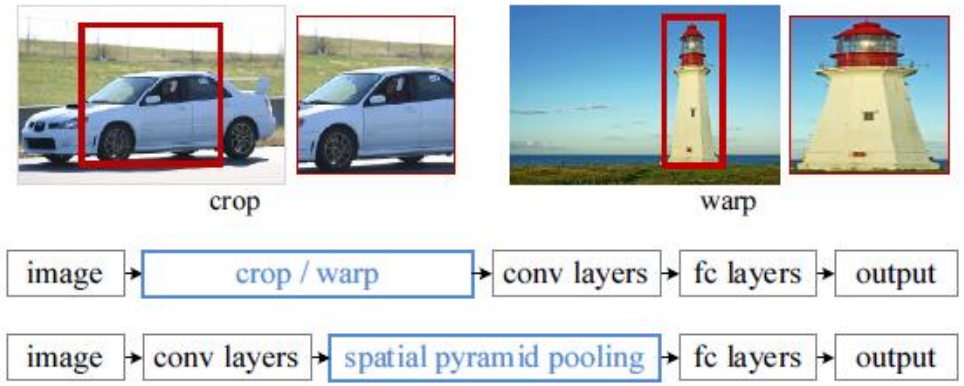
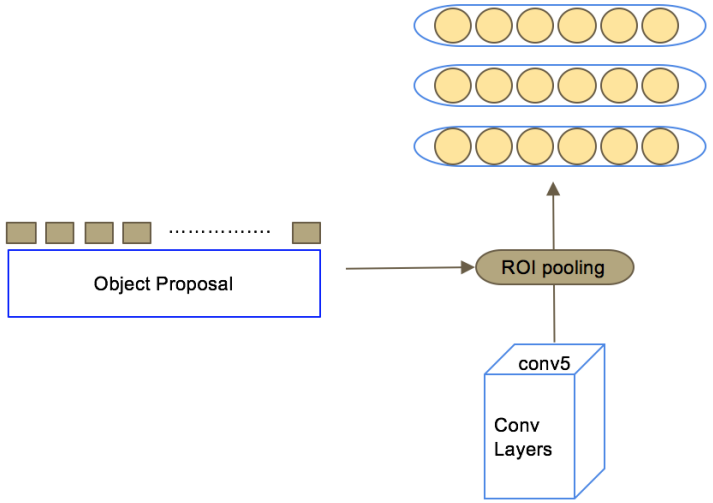
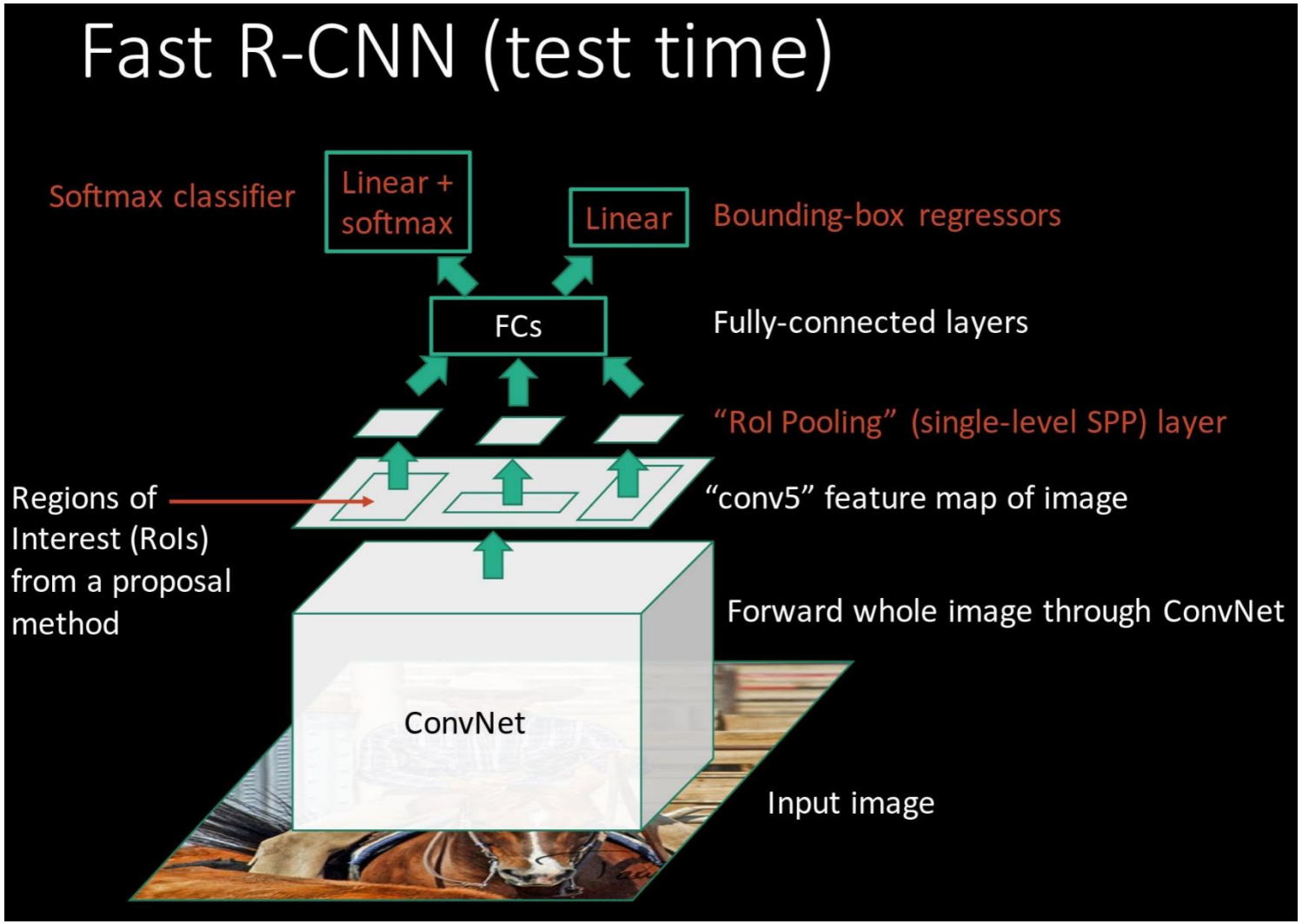


Figure 1: Top: cropping or warping to fit a fixed size. Middle: a conventional deep convolutional network structure. Bottom: our spatial pyramid pooling network structure.



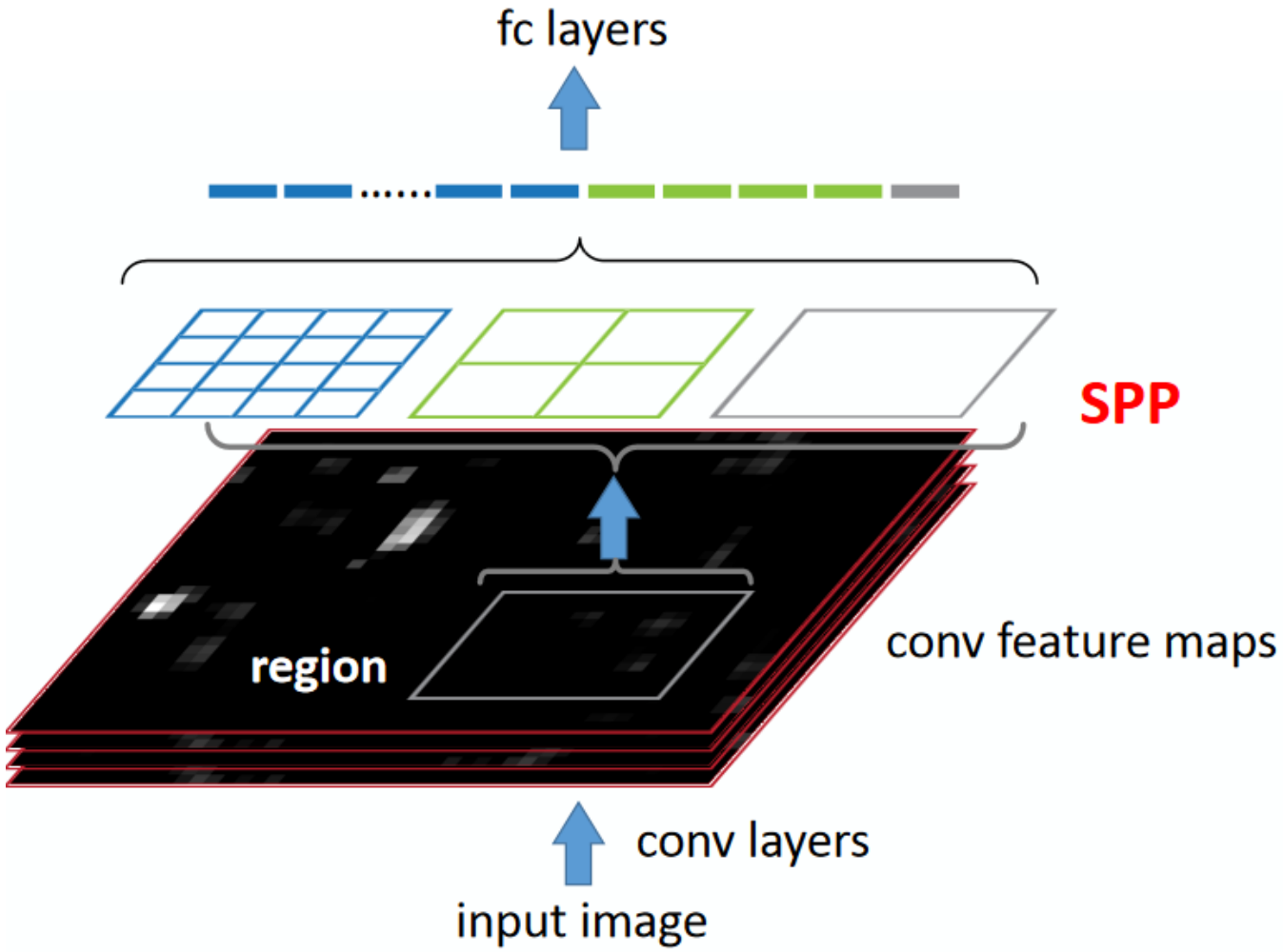
R. Girshick, "Fast R-CNN," in IEEE International Conference on Computer Vision (ICCV), 2015

Fast R-CNN



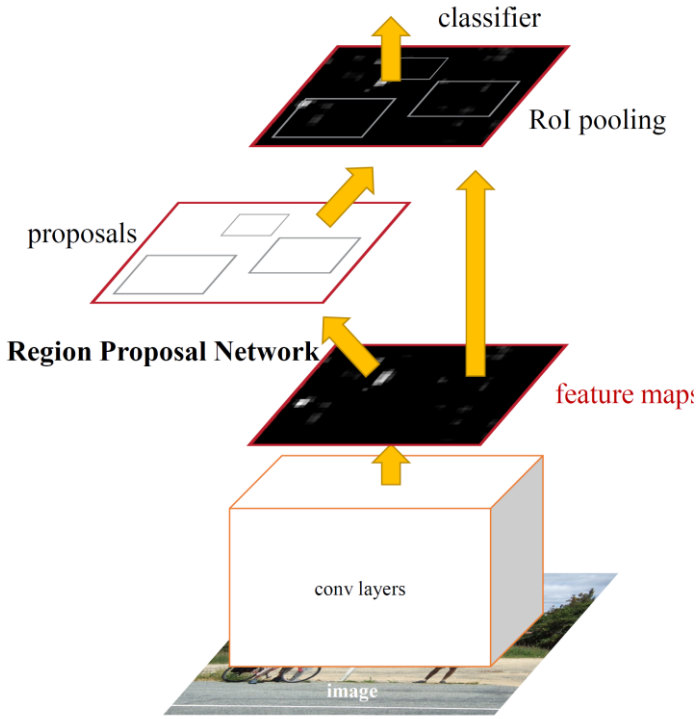
R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

RoI Pooling Layer

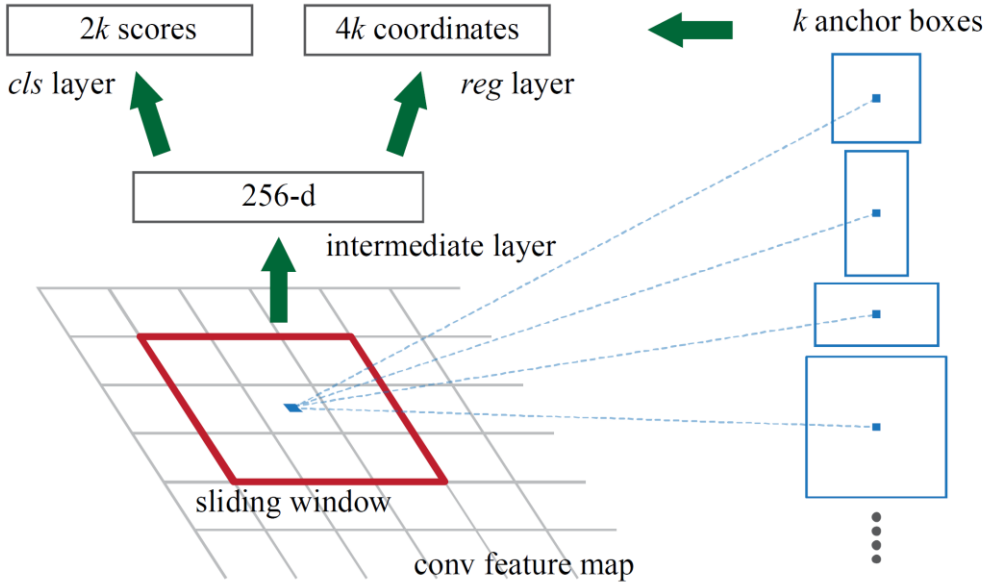


Faster R-CNN

- Replace object proposal(select search) with CNN layer(RPN)
  - Whole end-to-end CNN network

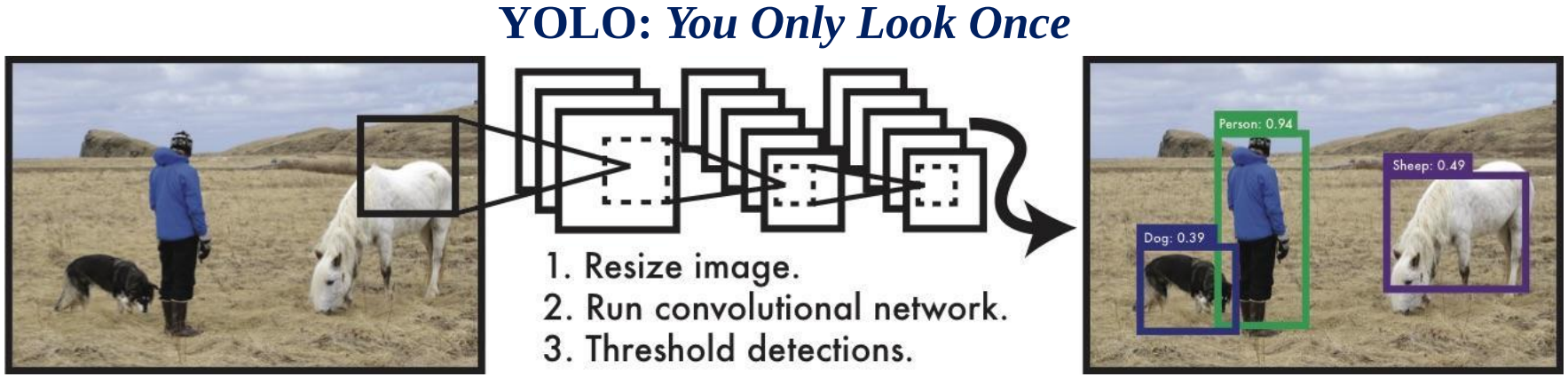


Region Proposal Network (RPN)



YOLO – You Only Look Once

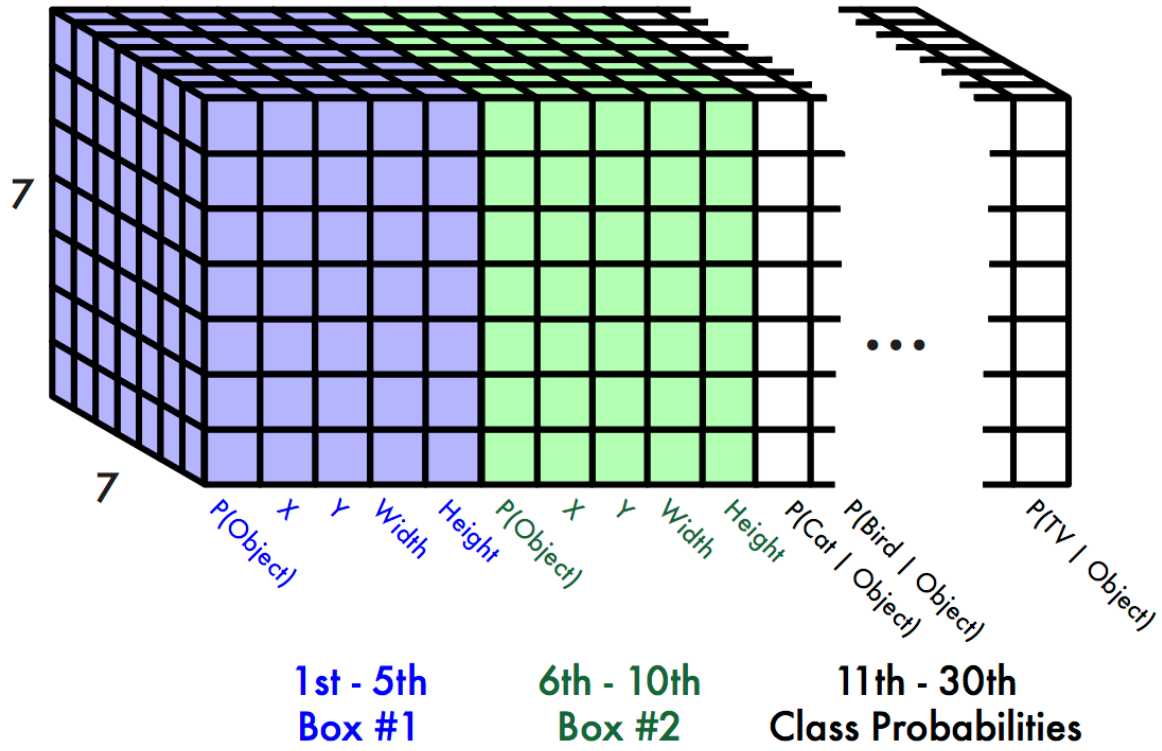
- R-CNN series do object proposal then classification
  - Two-stage detector
  - Too slow and not natural
- YOLO
  - One-stage detector
  - Get object location and class at the same time



Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

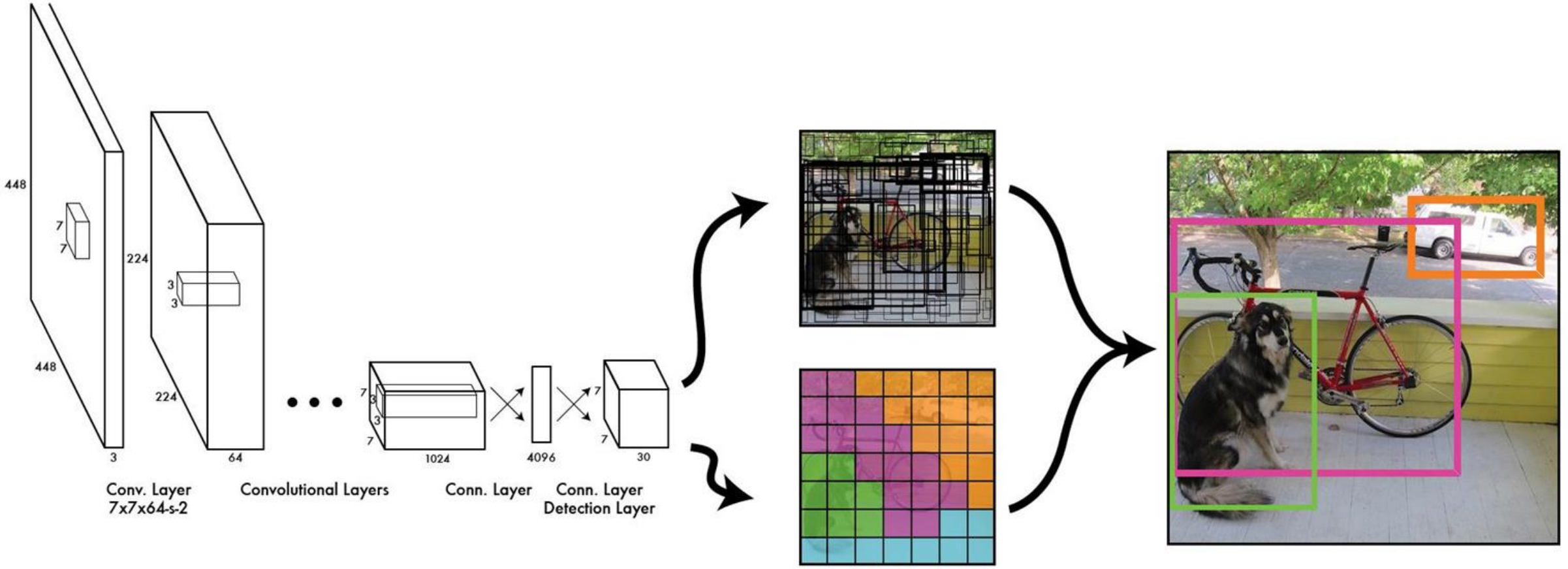
Output Tensor

- Each cell predicts:
  - For each bounding box:
    - 4 coordinates (x, y, w, h)
    - 1 confidence value
  - Some number of class probabilities

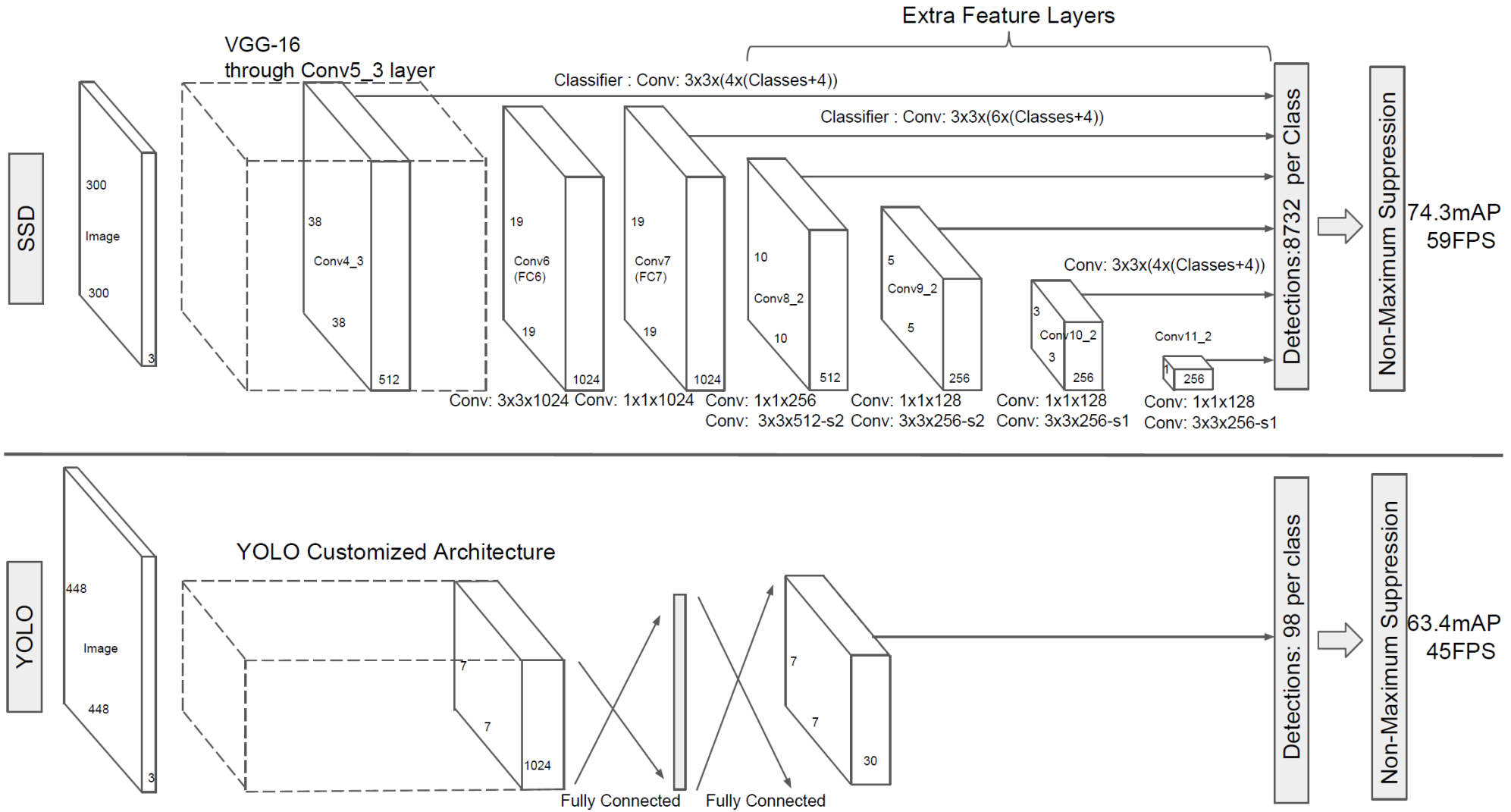


- For Pascal VOC:
  - 7x7 grid
  - 2 bounding boxes / cell
  - 20 classes
  - 7 x 7 x (2 x 5 + 20) = 7 x 7 x 30 tensor = 1470 outputs

YOLO Detection Flow



SSD – Detect at Multi-scale



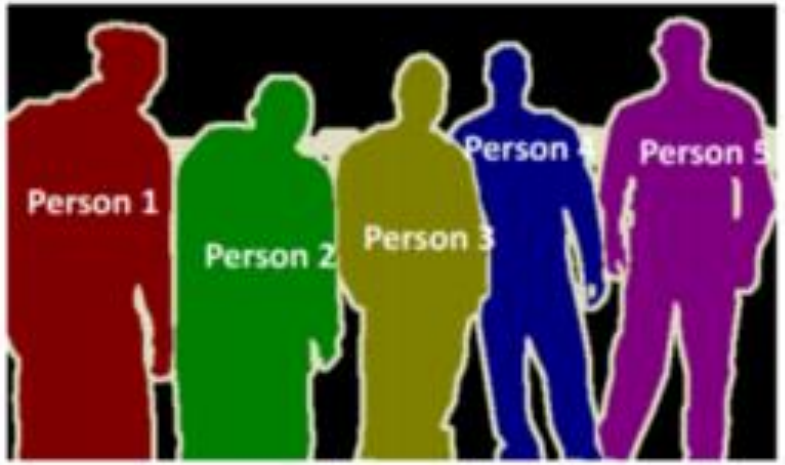
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

Image Segmentation

- Pixel level

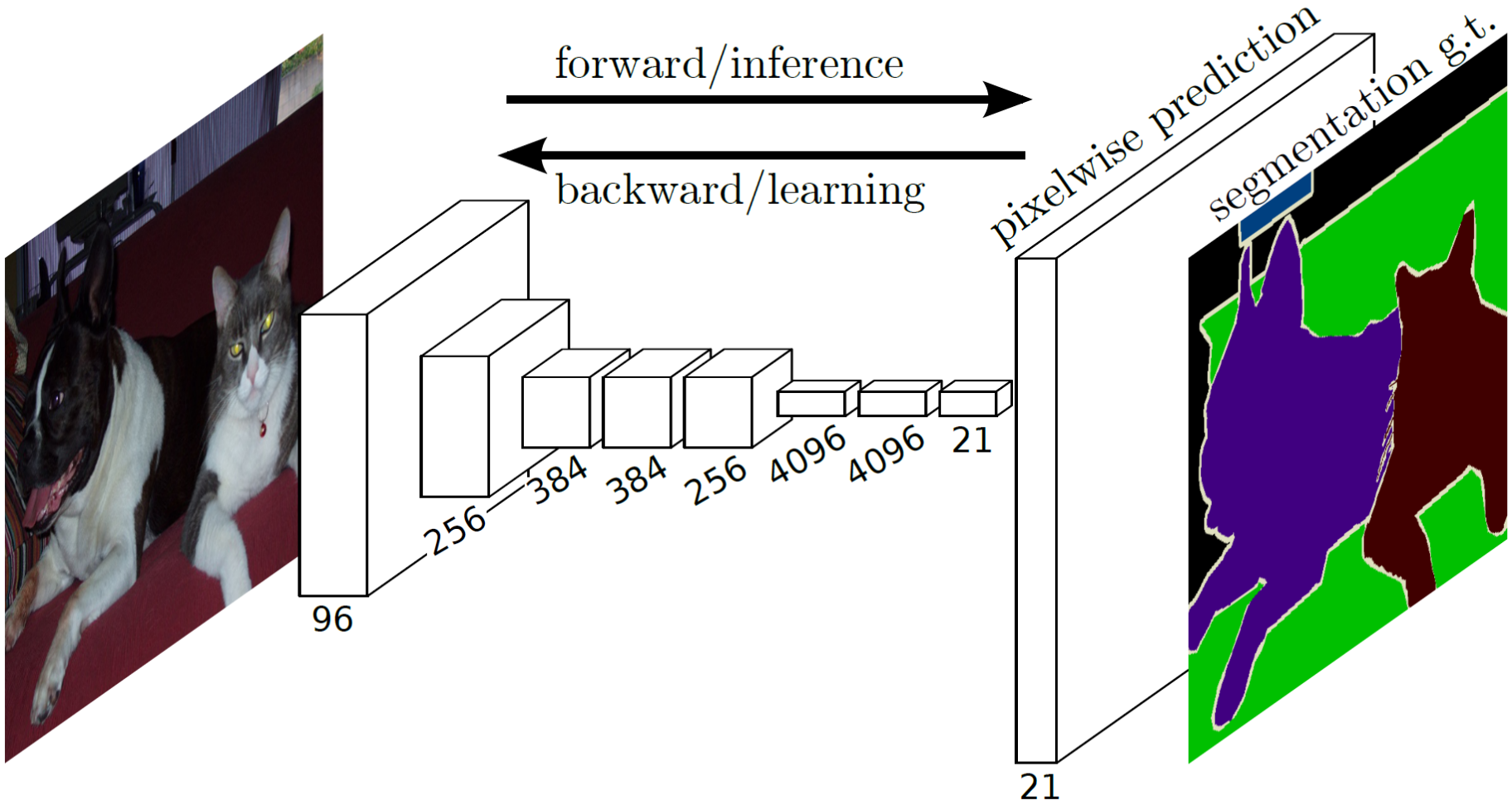
Object Detection

Semantic Segmentation



Instance Segmentation

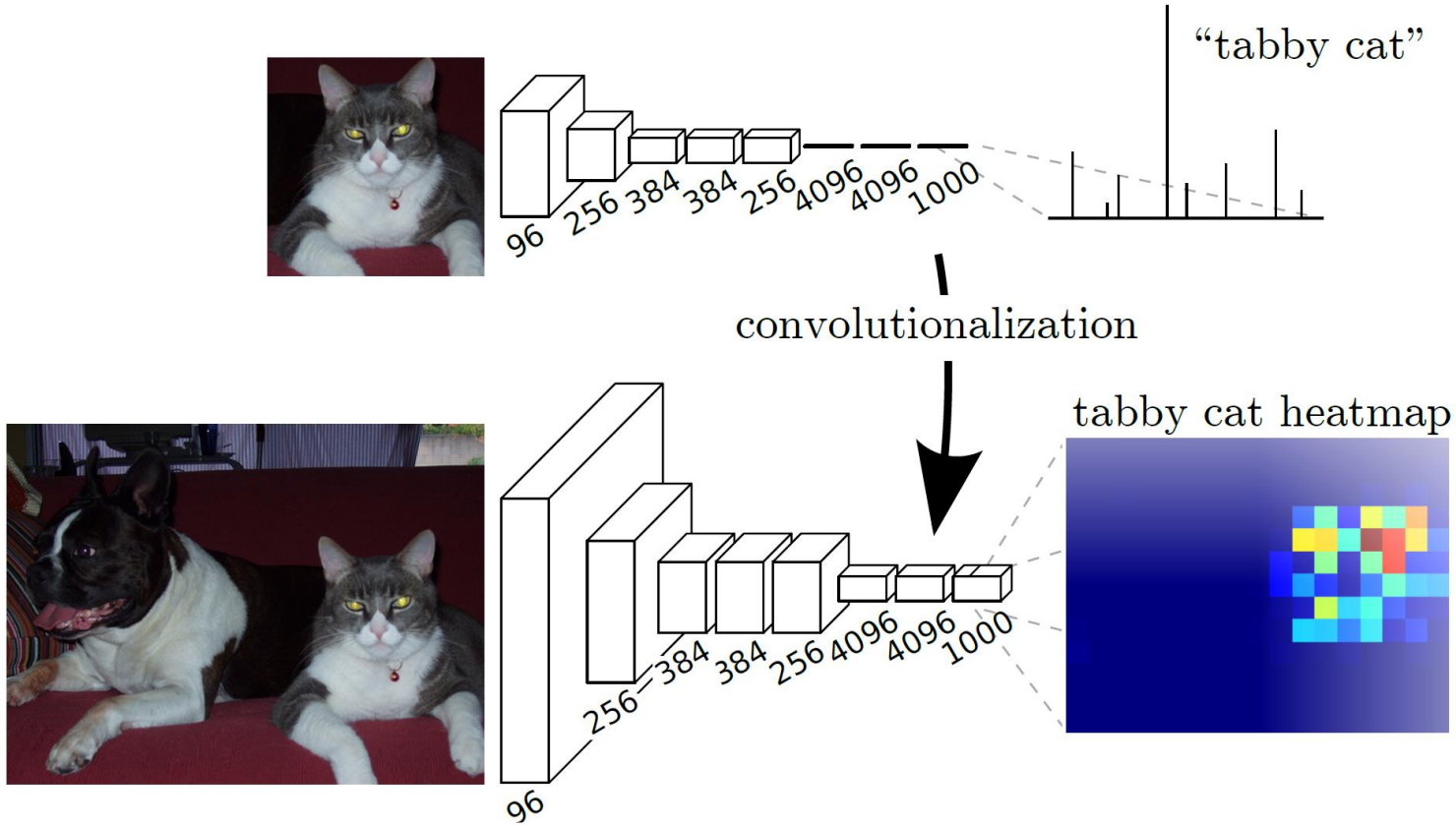
Fully Convolutional Networks for Semantic Segmentation



Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

Producing a Heatmap

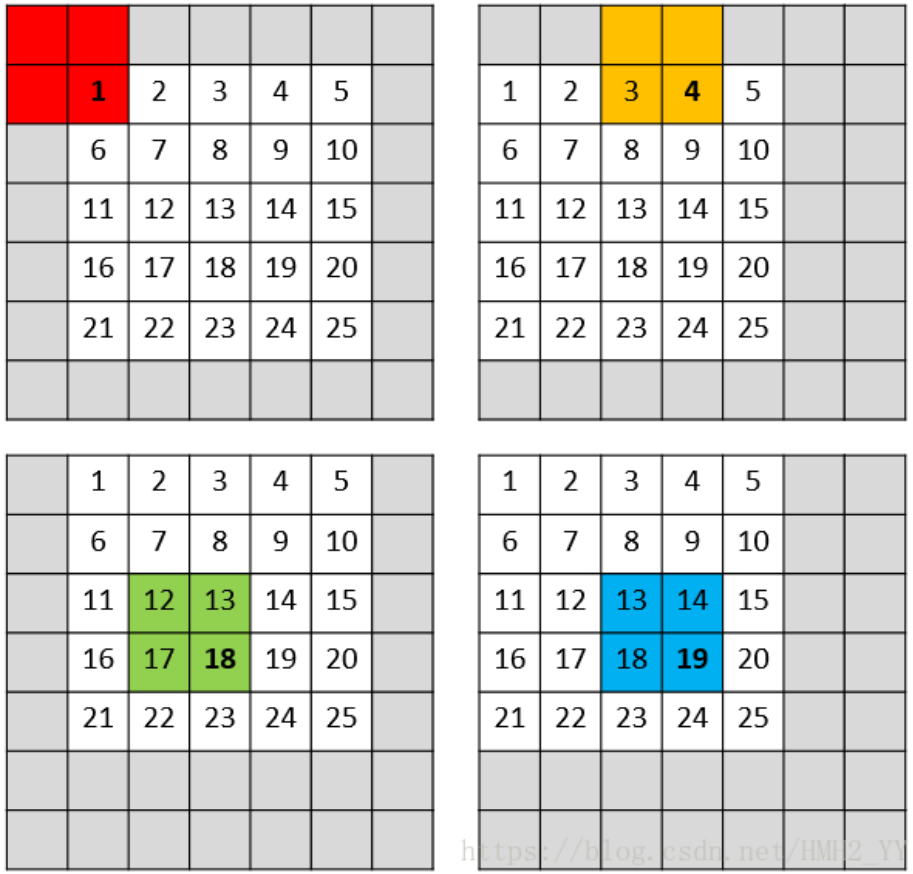
- Transforming fully connected layers into convolution layers
  - Output a heatmap

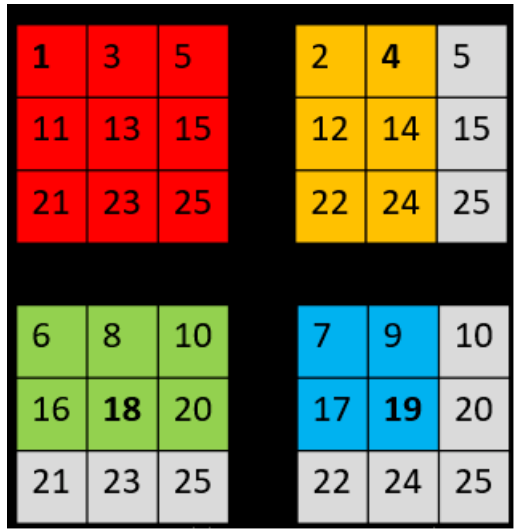


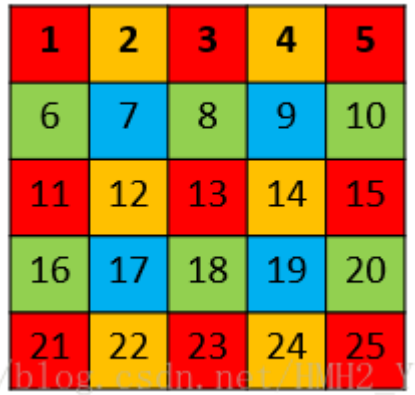
Dense Prediction - Shift-and Stitch

- Shift-and Stitch

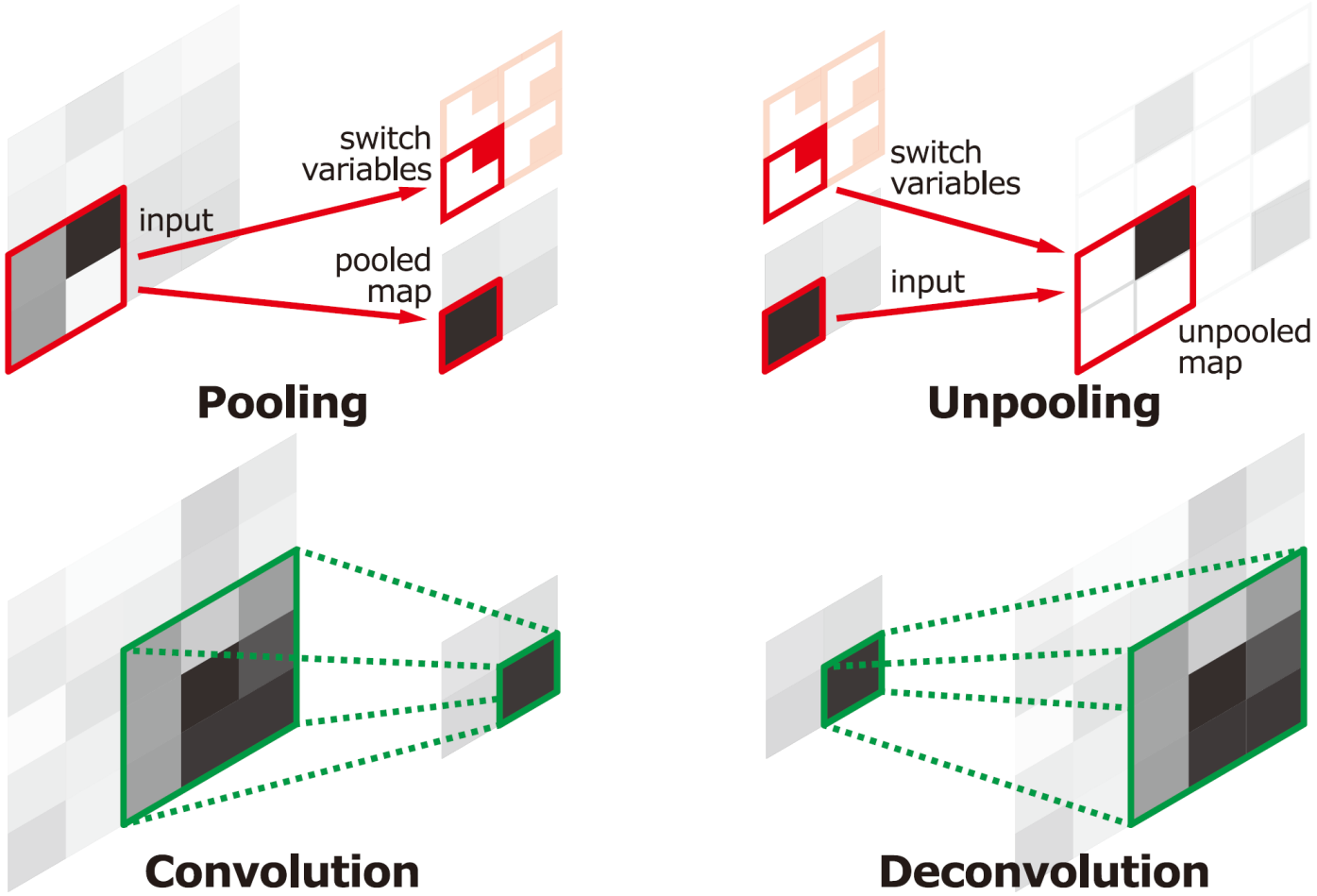
(0,0)	(1,0)
(0,1)	(1,1)





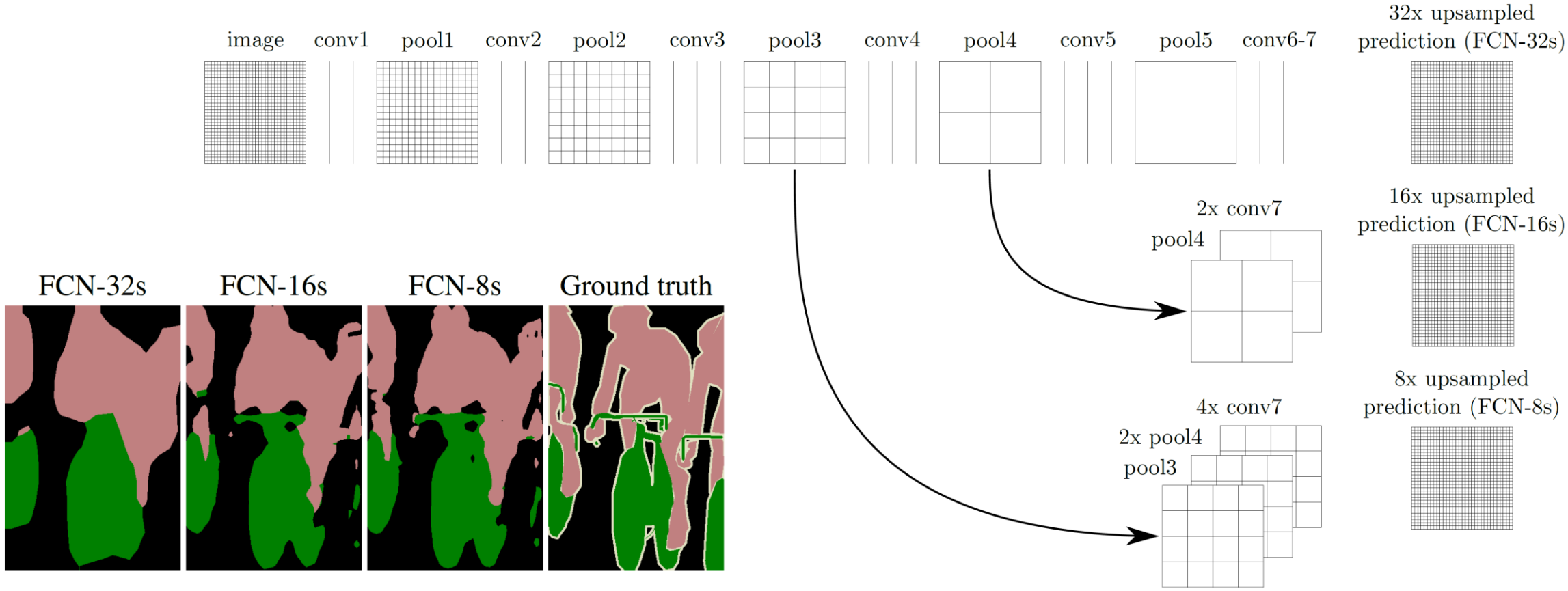


Dense Prediction - Deconvolution and Unpooling



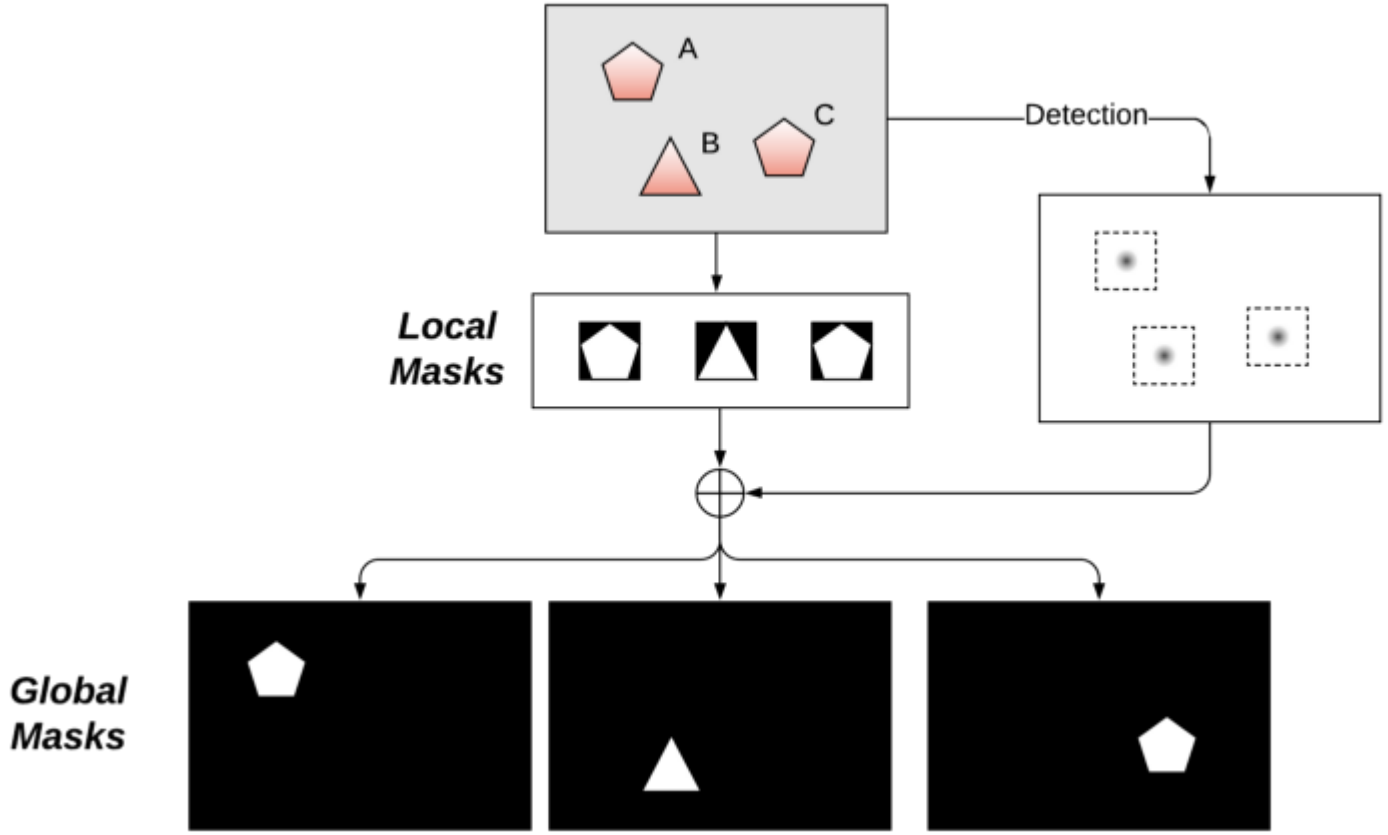
Noh, H., Hong, S., & Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1520-1528).

Combine Coarse and Fine Feature Map



From Semantic to Instance Segmentation

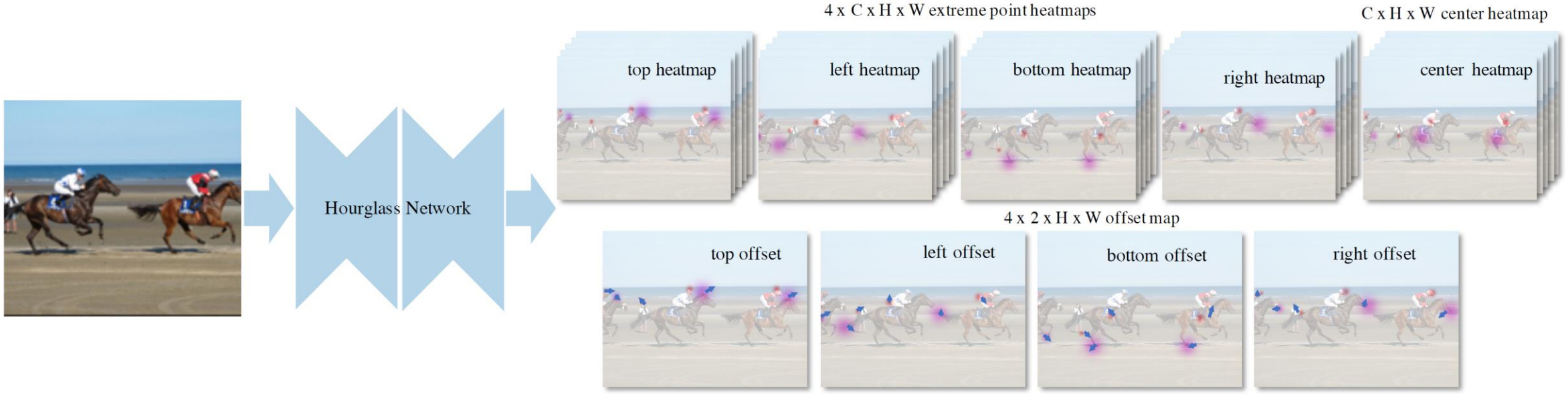
- Local mask or global mask
- Representation or parameterization



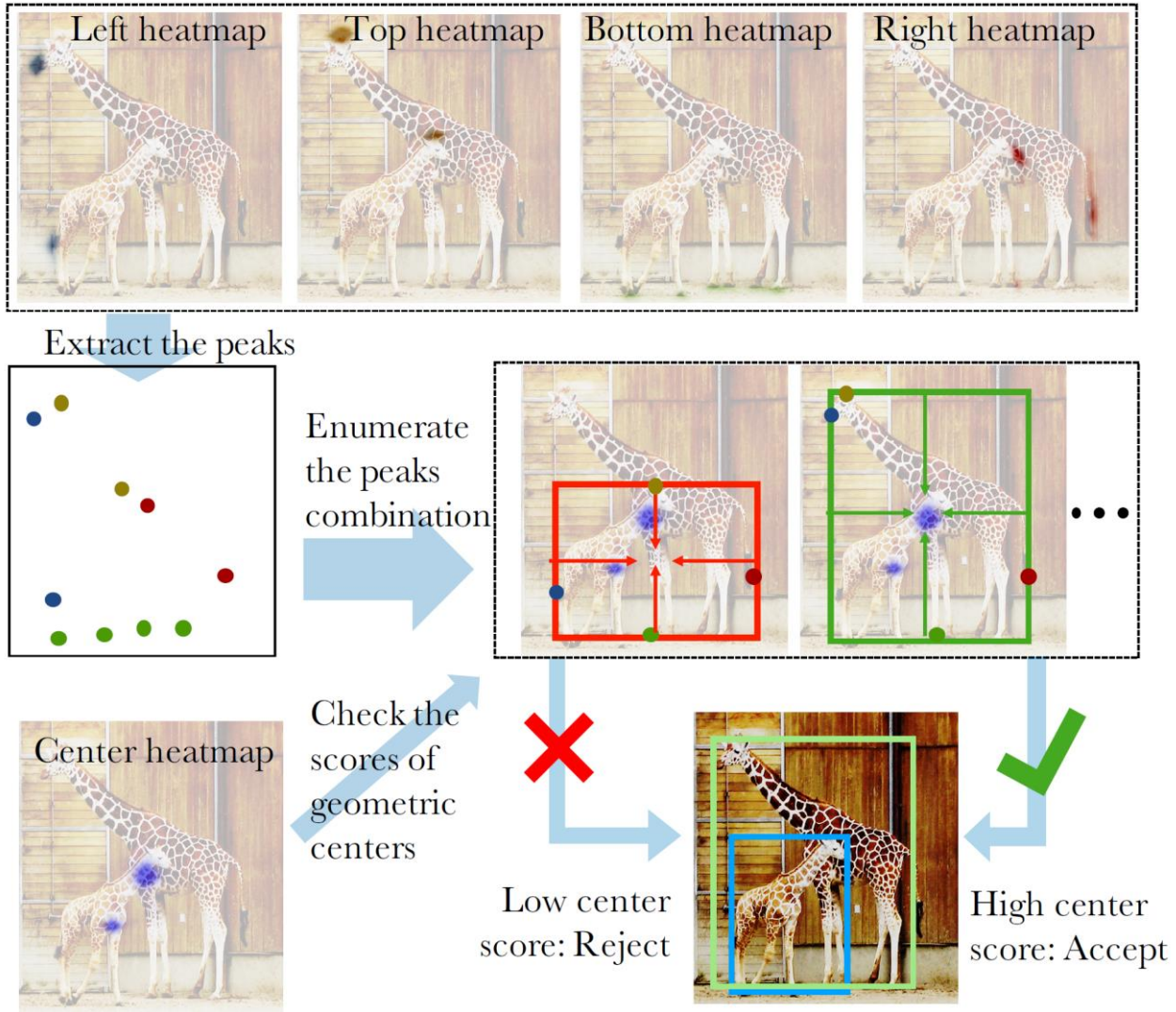
Contours with Explicit Encoding

Zhou, X., Zhuo, J., & Krahenbuhl, P. (2019). Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 850-859).

Contours with Explicit Encoding

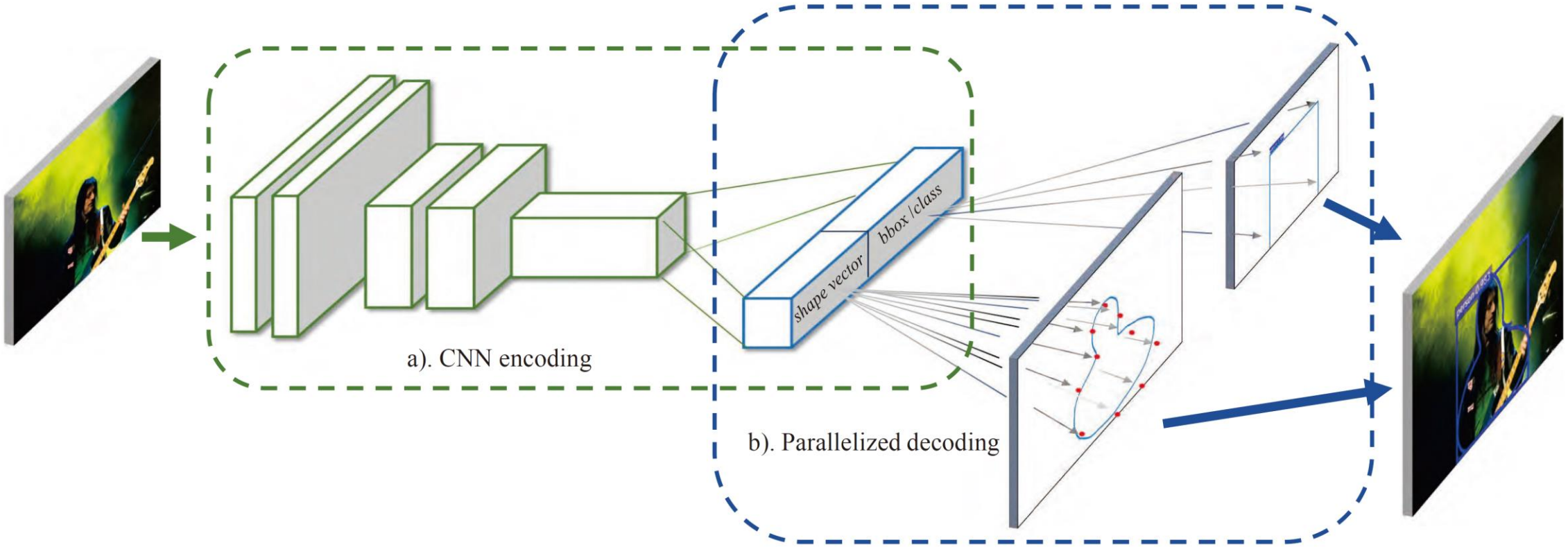


Contours with Explicit Encoding



Explicit Shape Encoding for Real-time Instance Segmentation

- Designs an inner center radius shape signature for each instance and fits it with Chebyshev polynomials

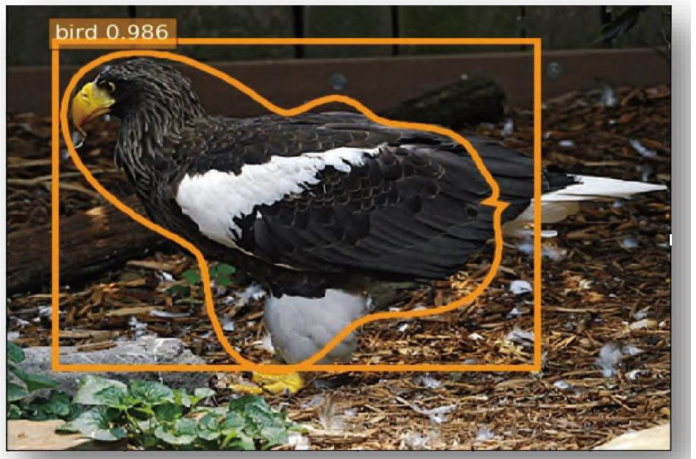


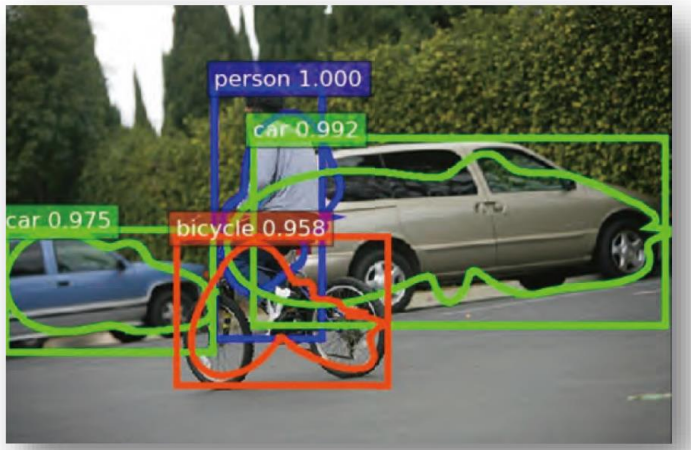
Xu, W., Wang, H., Qi, F., & Lu, C. (2019). Explicit shape encoding for real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5168-5177).

Explicit Shape Encoding for Real-time Instance Segmentation

Original Image

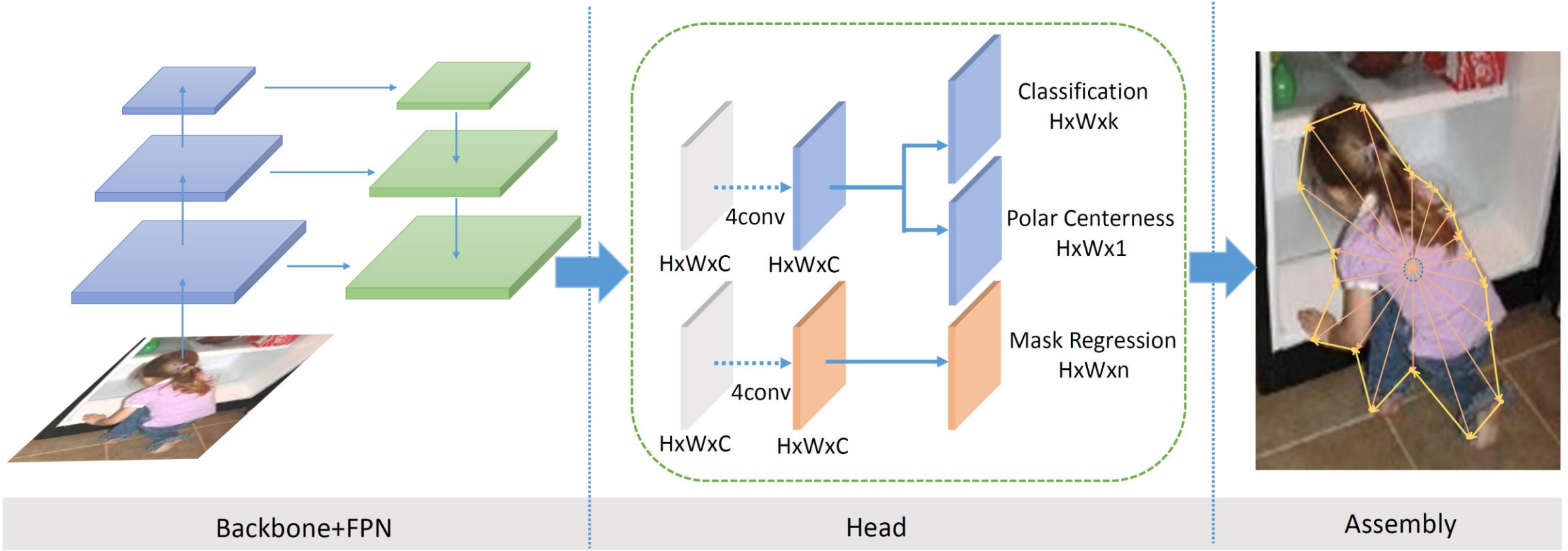
Predicted Shape





PolarMask

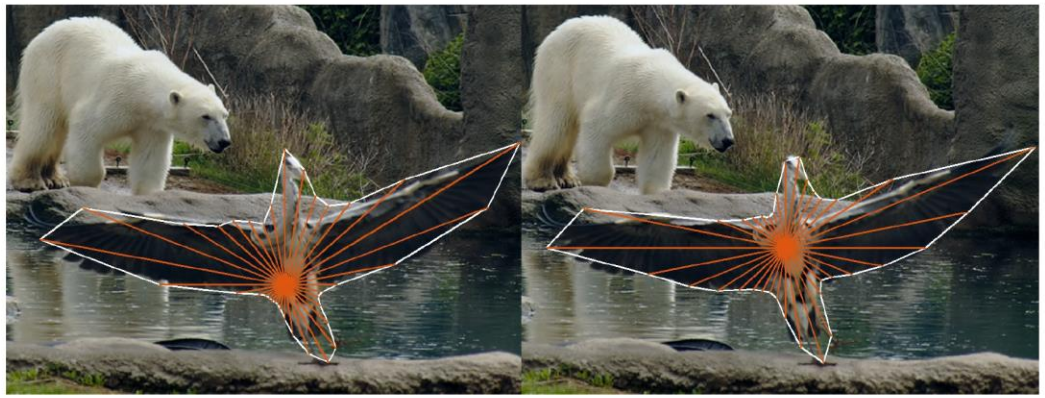
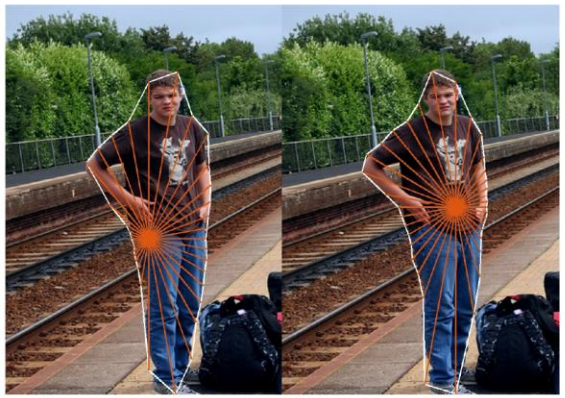
- Utilizes rays at constant angle intervals from the center to describe the

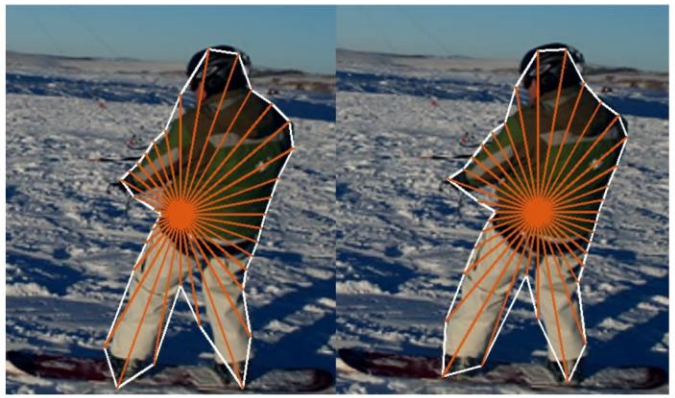


Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., ... & Luo, P. (2020). Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12193-12202).

Cartesian Centerness vs. Polar Centerness

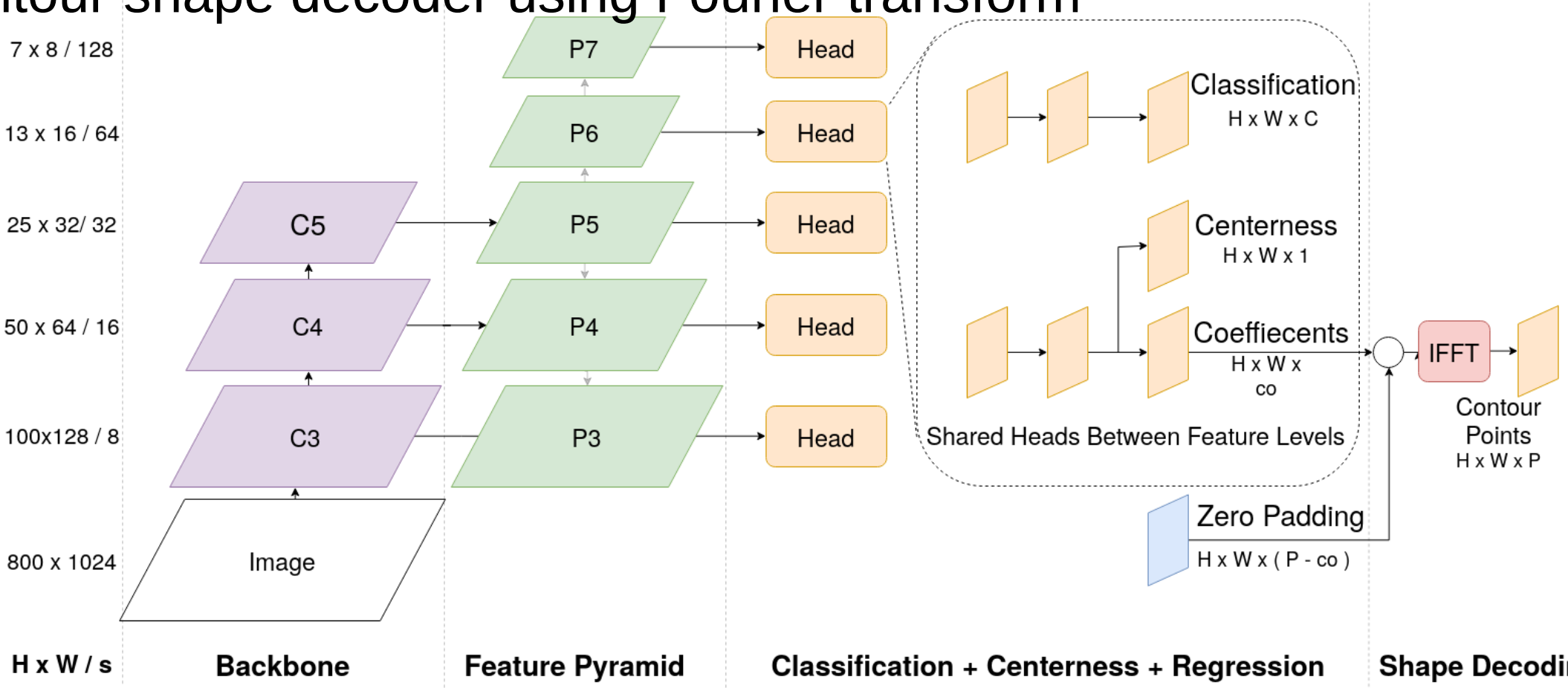




FourierNet

- Contour shape decoder using Fourier transform



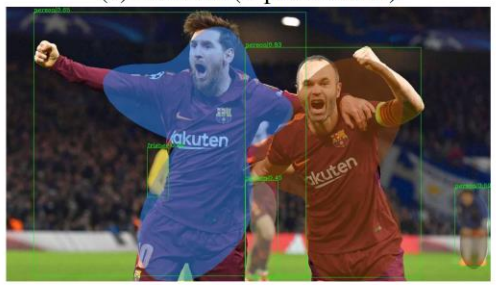
Riaz, H. U. M., Benbarka, N., & Zell, A. (2021, January). FourierNet: Compact mask representation for instance segmentation using differentiable shape decoders. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 7833-7840). IEEE.

FourierNet

(a) 2 coeff. (4 parameters.)

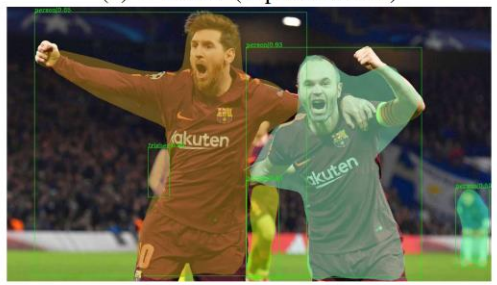
(b) 3 coeff. (6 parameters.)

(c) 4 coeff. (8 parameters.)

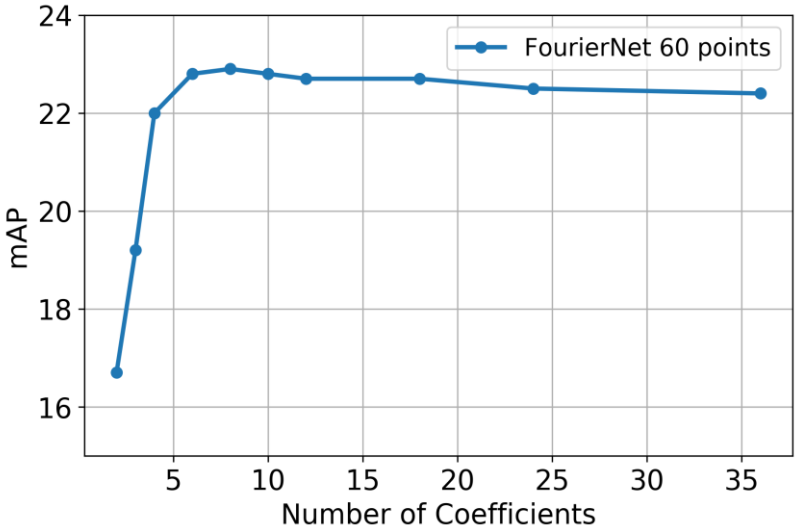


(d) 6 coeff. (12 parameters.)

(e) 12 coeff. (24 parameters.)



(f) 36 coeff. (72 parameters.)



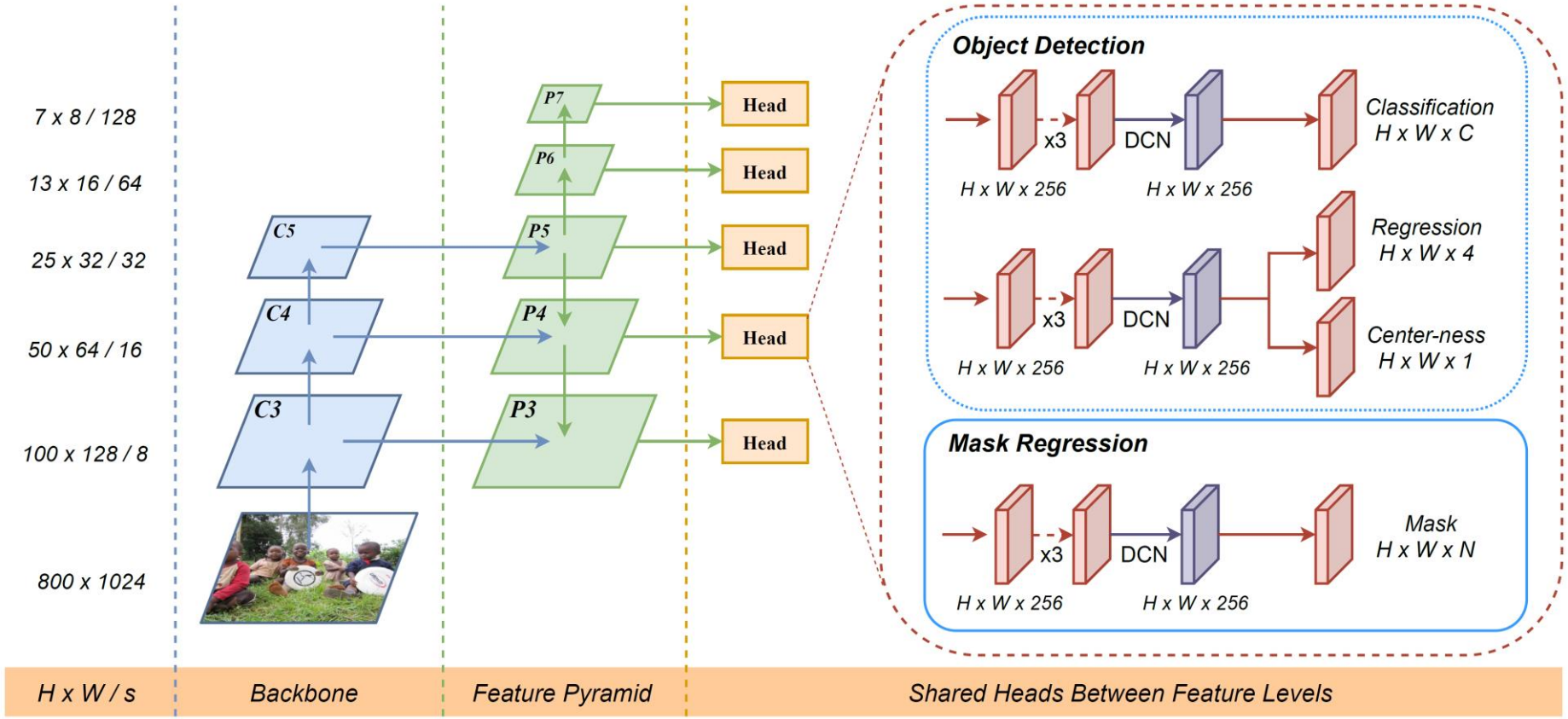
Contour-based vs. Mask-based Segmentation

(a) Contour-Based

(b) Mask-Based

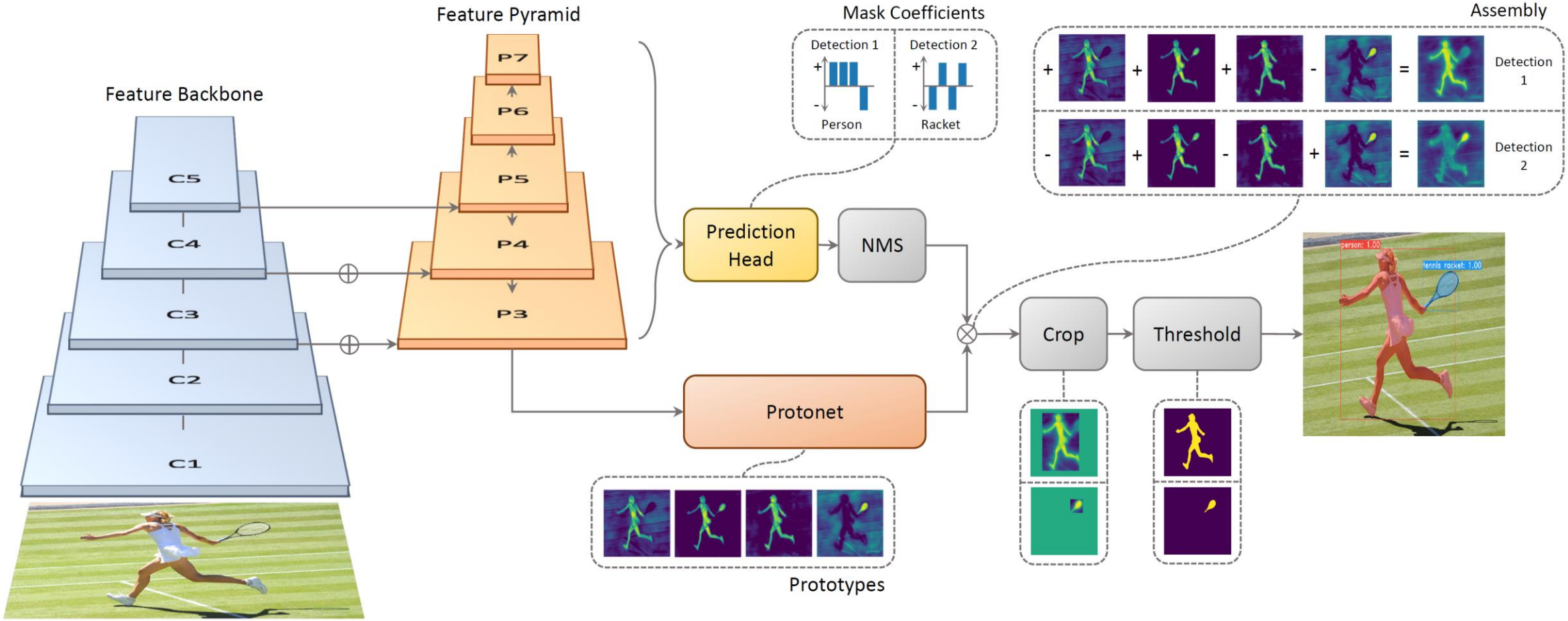
MEInst - Mask Encoding

- Distills the mask into a compact and fixed dimensional representation



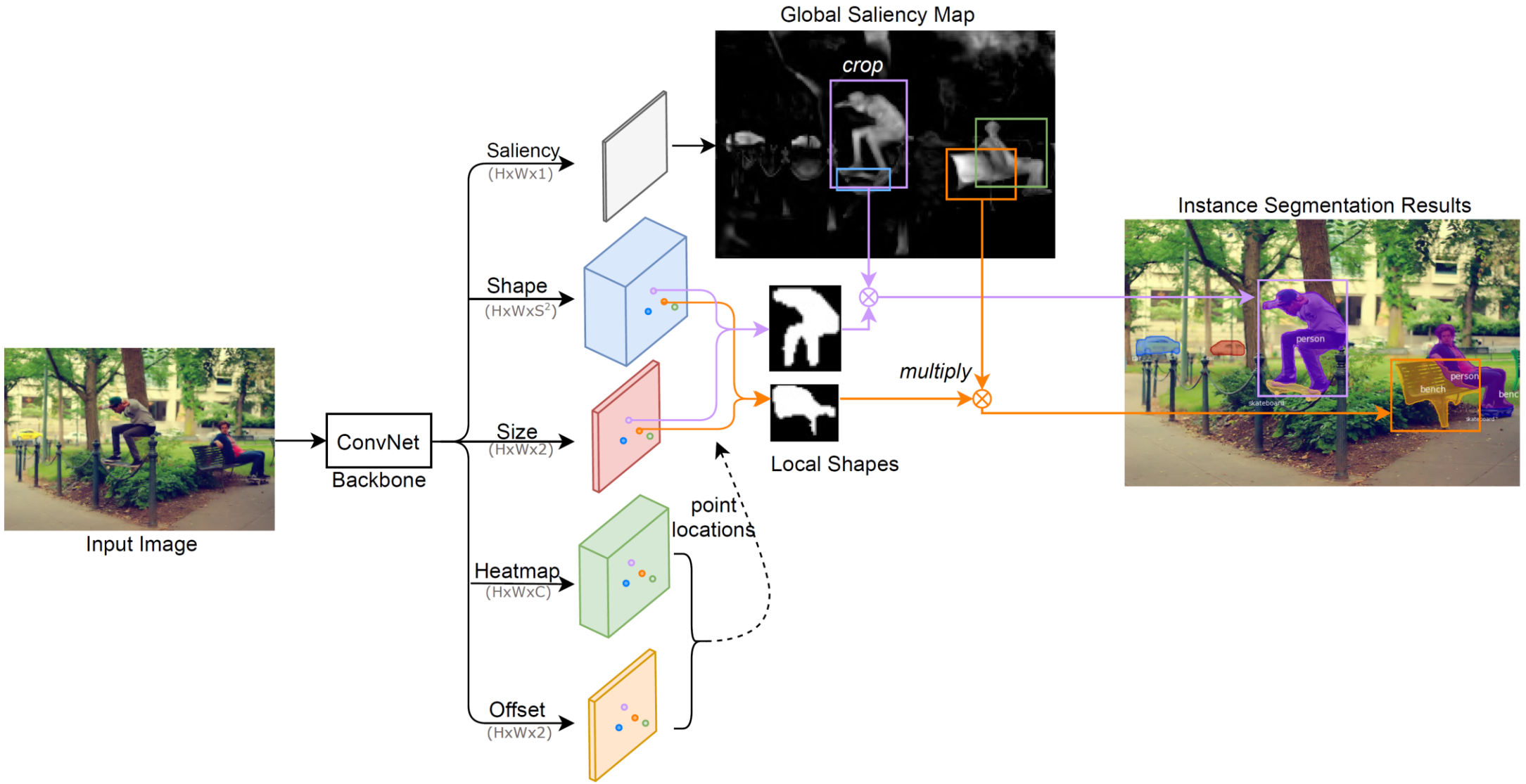
Zhang, R., Tian, Z., Shen, C., You, M., & Yan, Y. (2020). Mask encoding for single shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10226-10235).

YOLACT - Global-mask-Based Segmentation



Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9157-9166).

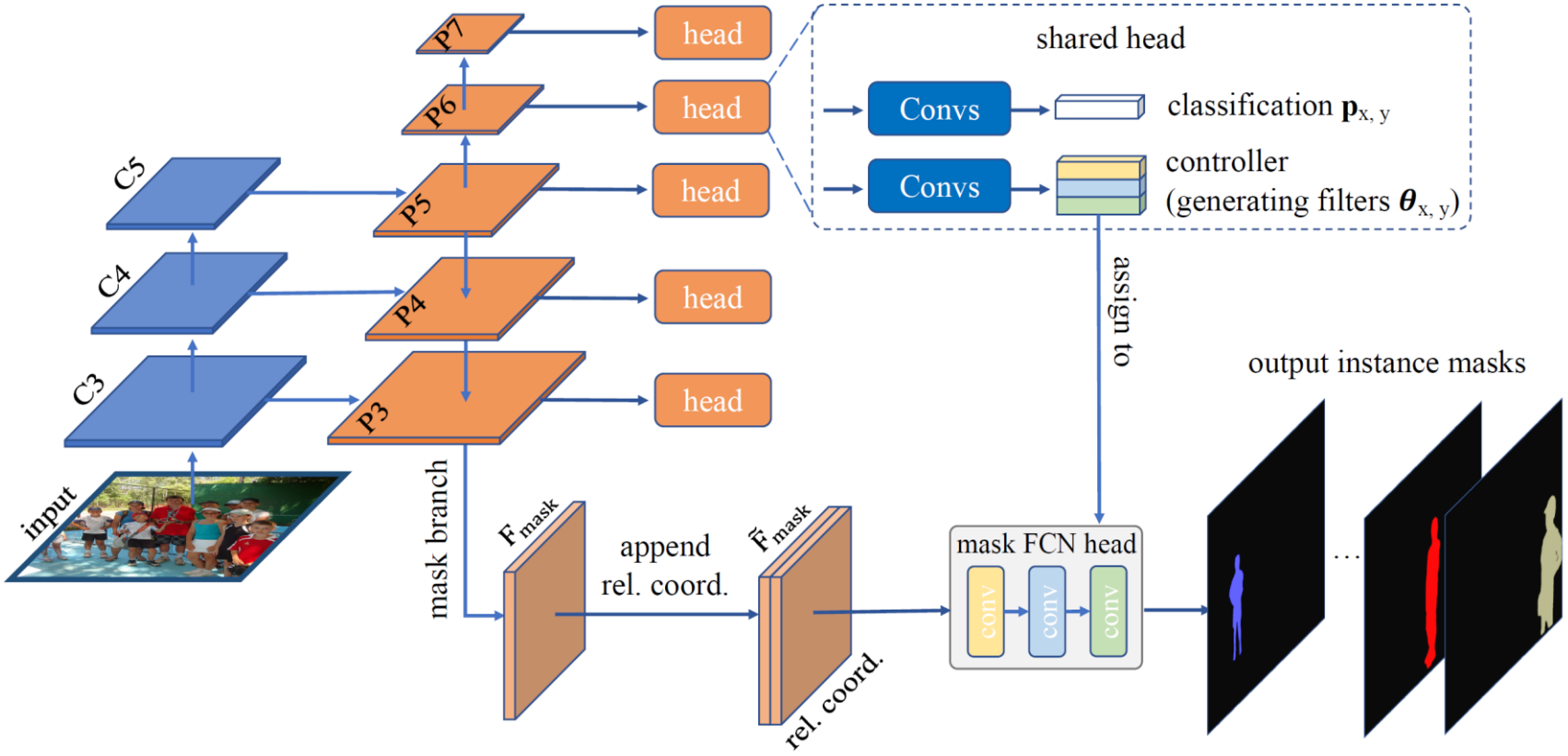
CenterMask - Combine Global and Local



Lee, Y., & Park, J. (2020). Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13906-13915).

CondInst

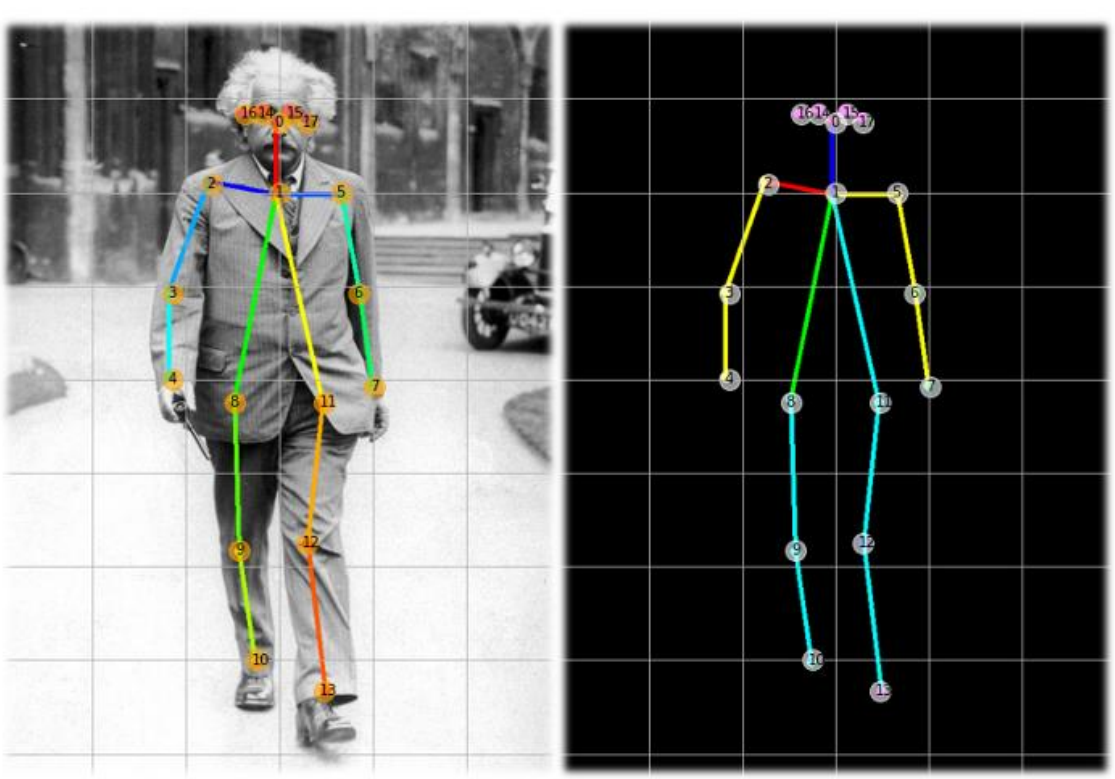
- Completely removes any dependency on bounding boxes



Tian, Z., Shen, C., & Chen, H. (2020, August). Conditional convolutions for instance segmentation. In *European Conference on Computer Vision* (pp. 282-298). Springer, Cham.

Pose Estimation

CNN for Regression

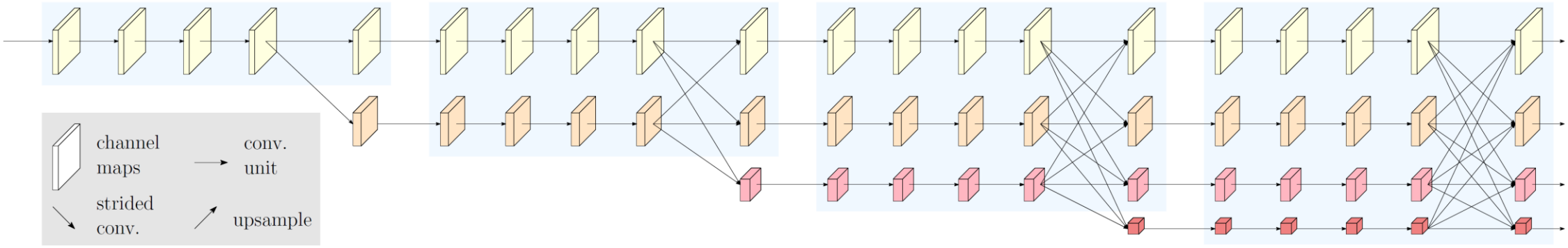


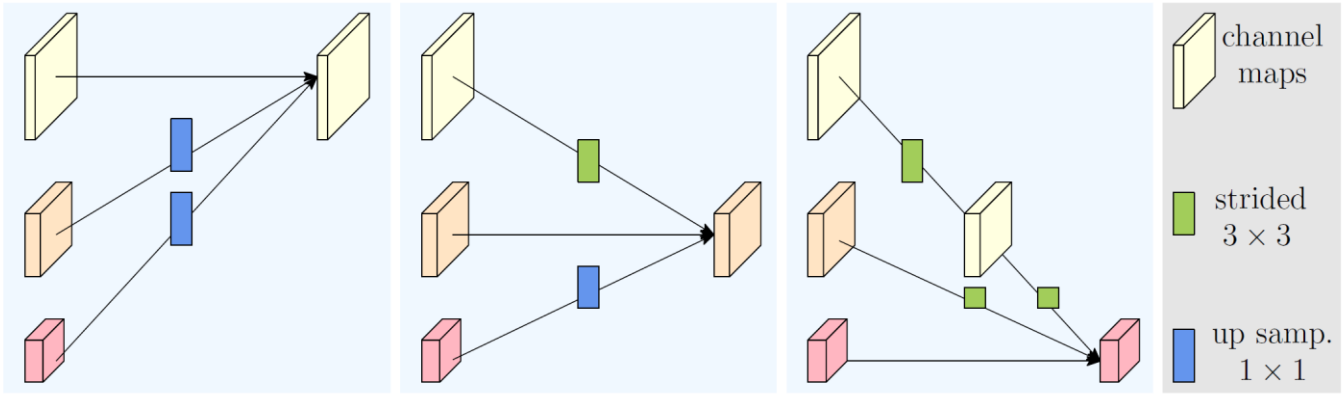
Pose Estimation

- Top-down
  - Detect then single person pose estimation
  - Complexity proportional to number of people
  - Accuracy affected by detector performance
- Bottom-up
  - Pose estimate then clustering
  - Complexity not directly proportional to number of people
  - Real-time speed

HRNet

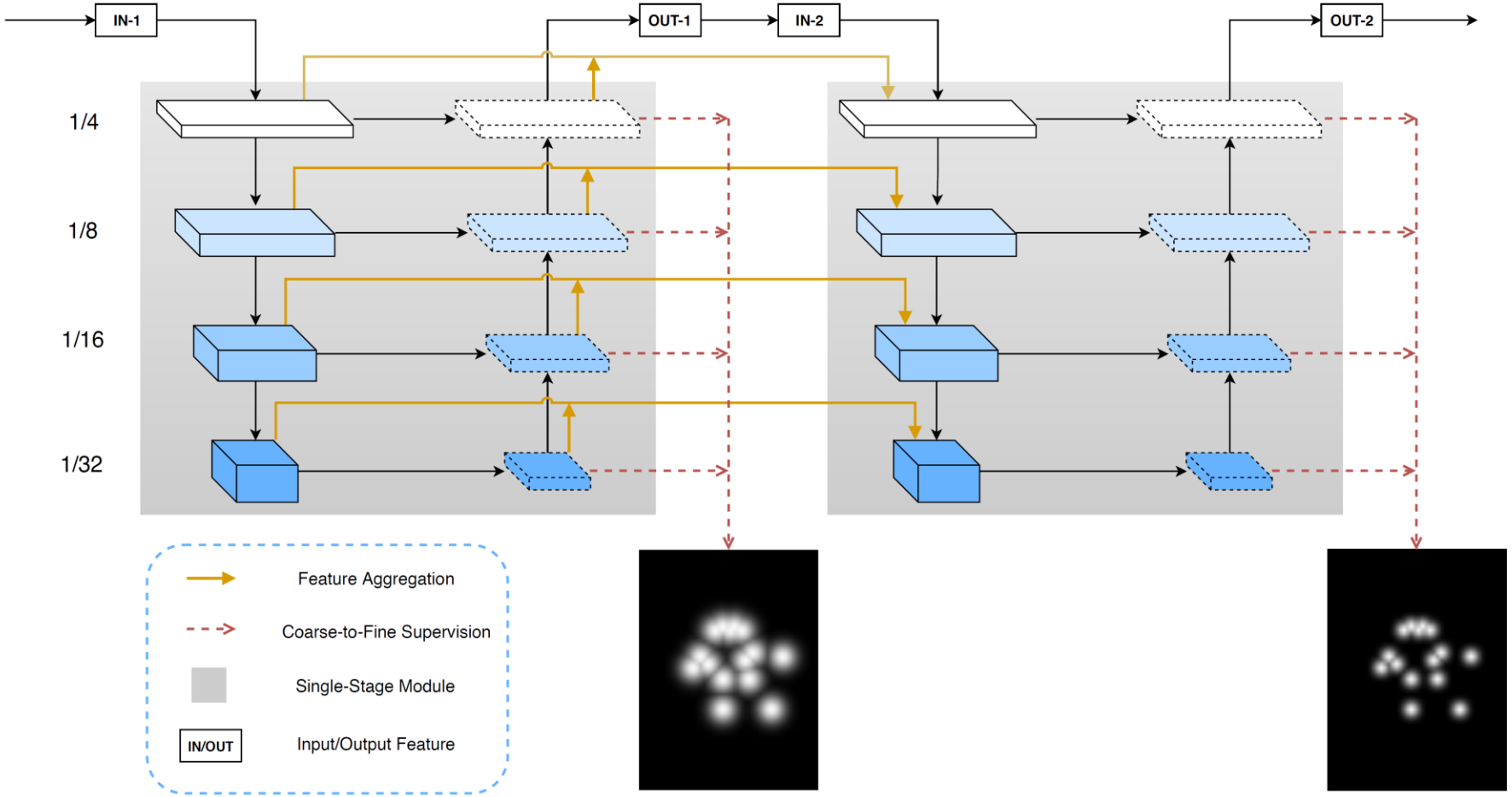
- Parallel feature extraction from different resolution
- Feature aggregation at each stage





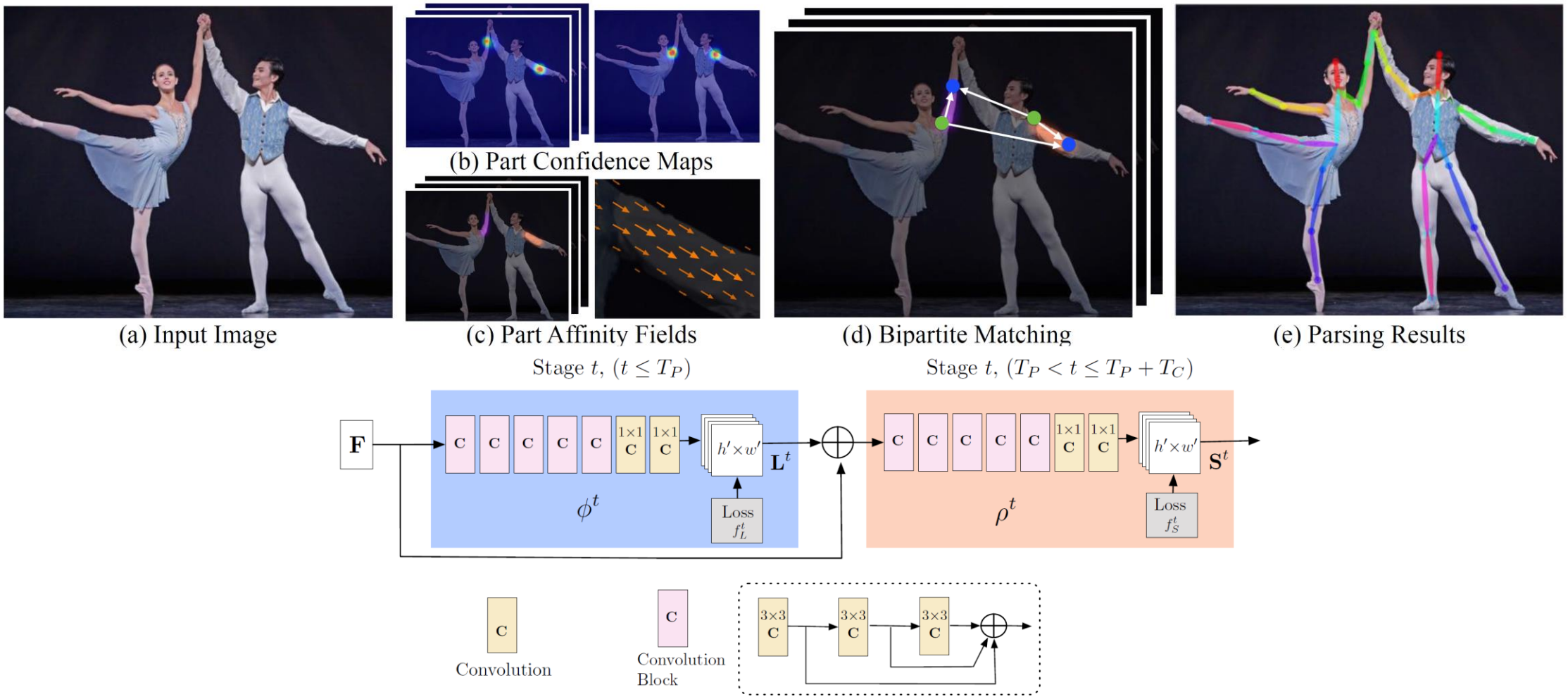
Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., ... & Xiao, B. (2020). Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, *43*(10), 3349-3364.

MSPN - Coarse-to-fine Supervision



Li, W., Wang, Z., Yin, B., Peng, Q., Du, Y., Xiao, T., ... & Sun, J. (2019). Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*.

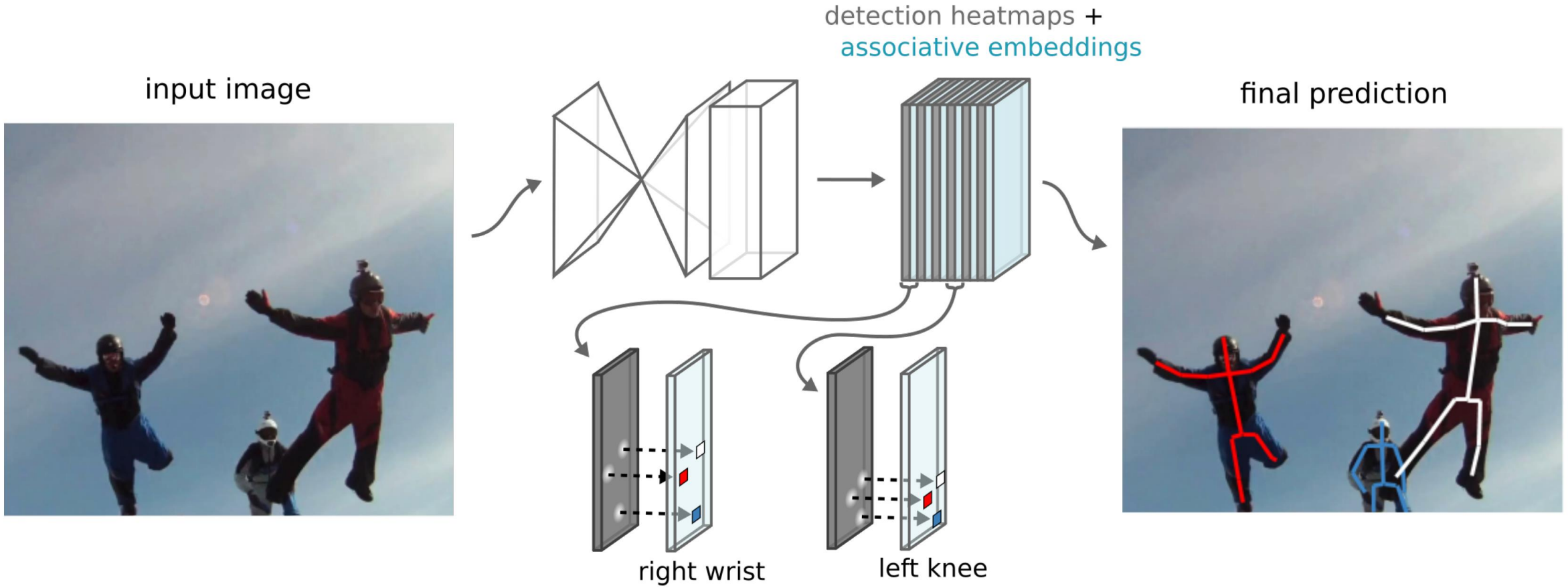
OpenPose – Part and Affinity



Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7291-7299).

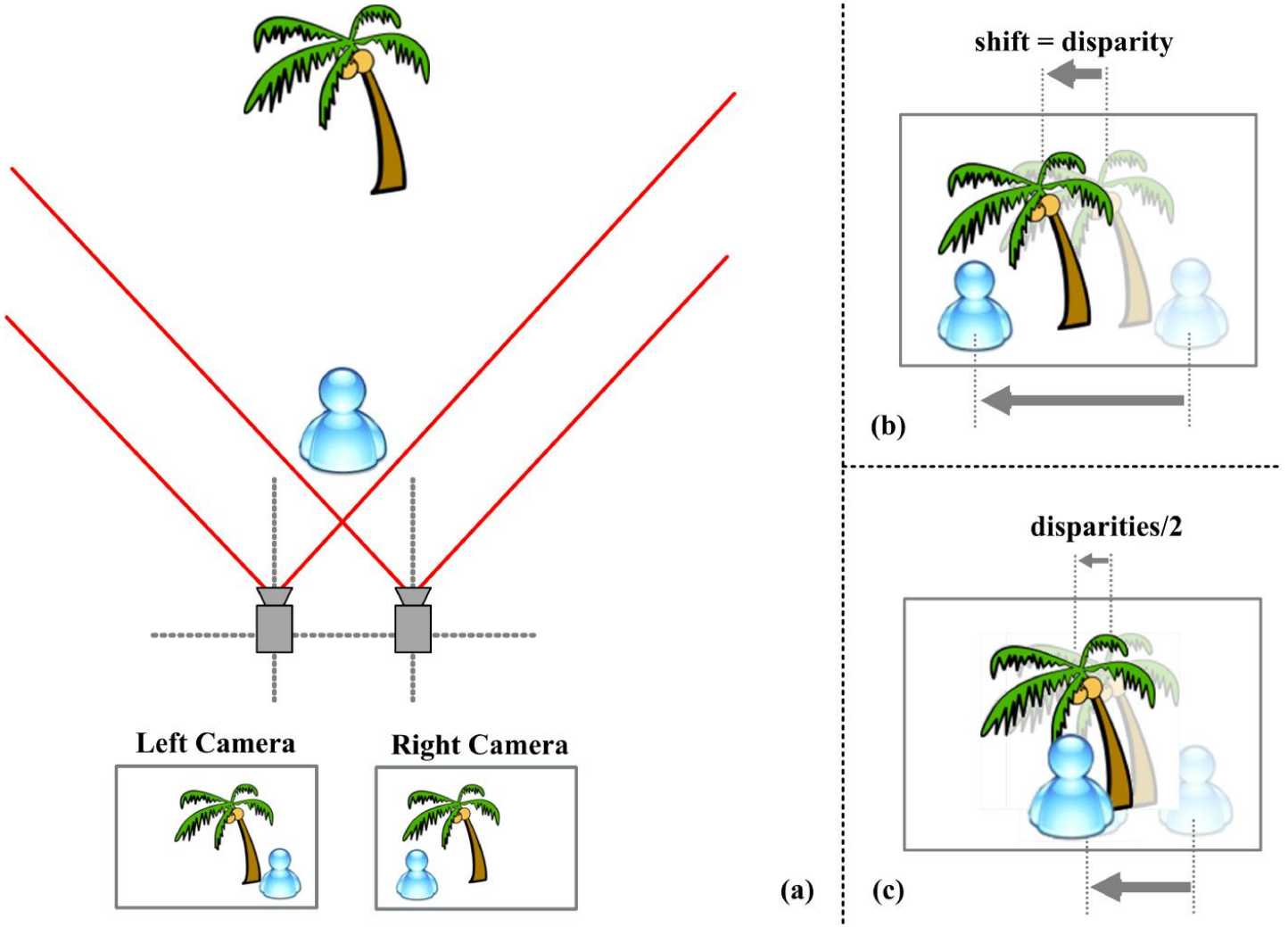
Associative Embedding

- Detection and grouping at the same time

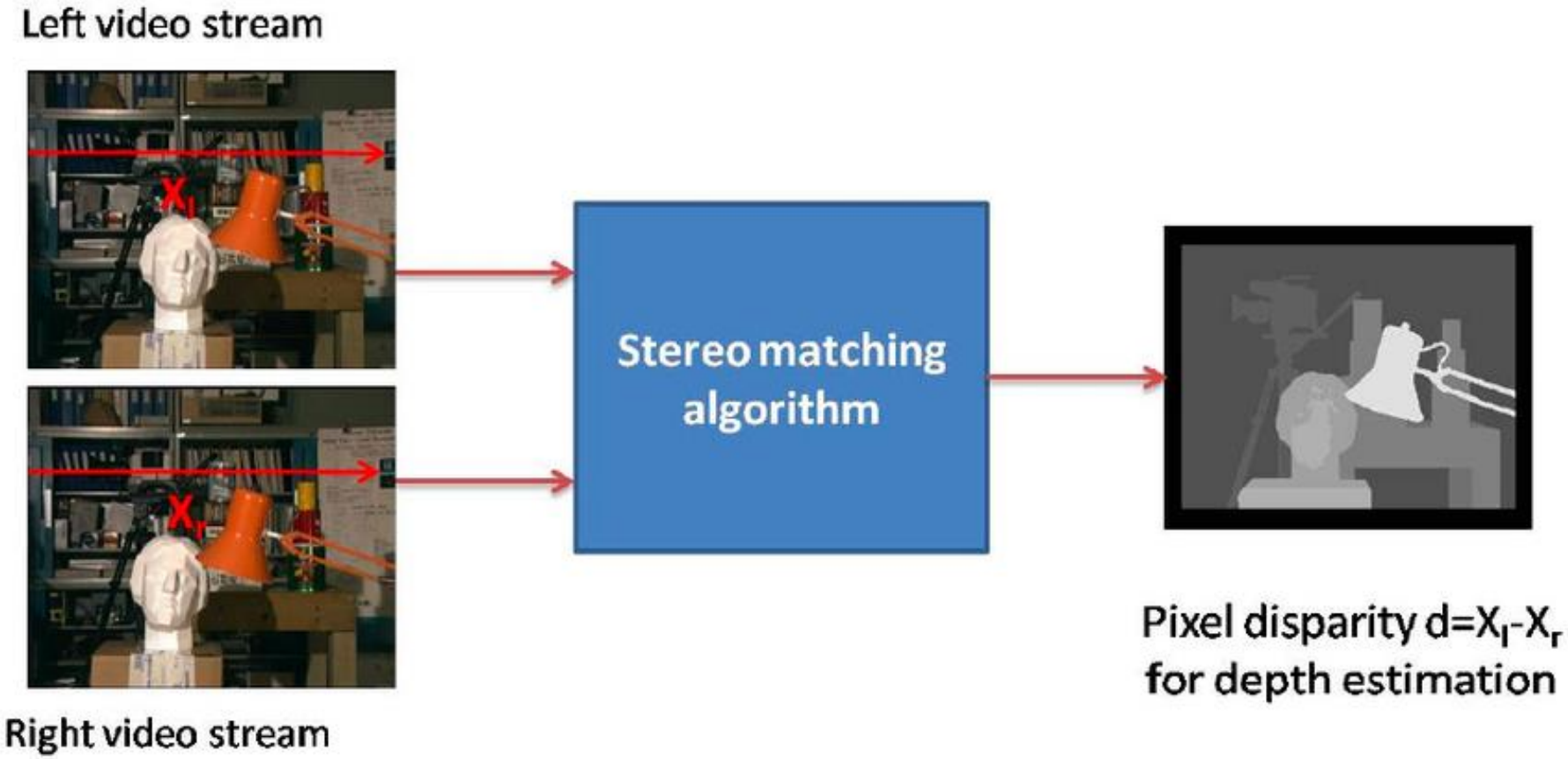


Newell, A., Huang, Z., & Deng, J. (2017). Associative embedding: End-to-end learning for joint detection and grouping. *Advances in neural information processing systems*, *30*.

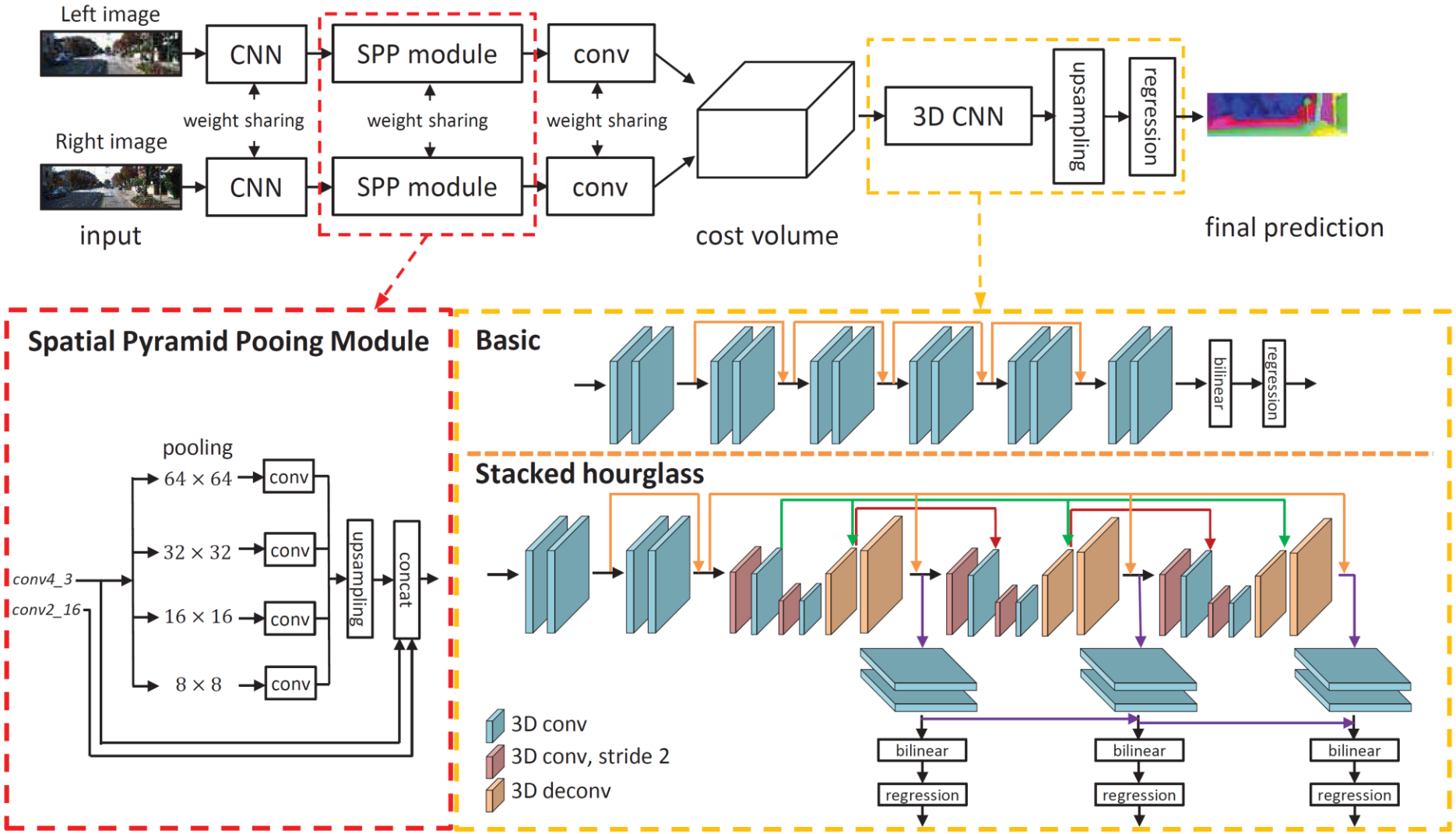
Stereo Matching



Stereo Matching

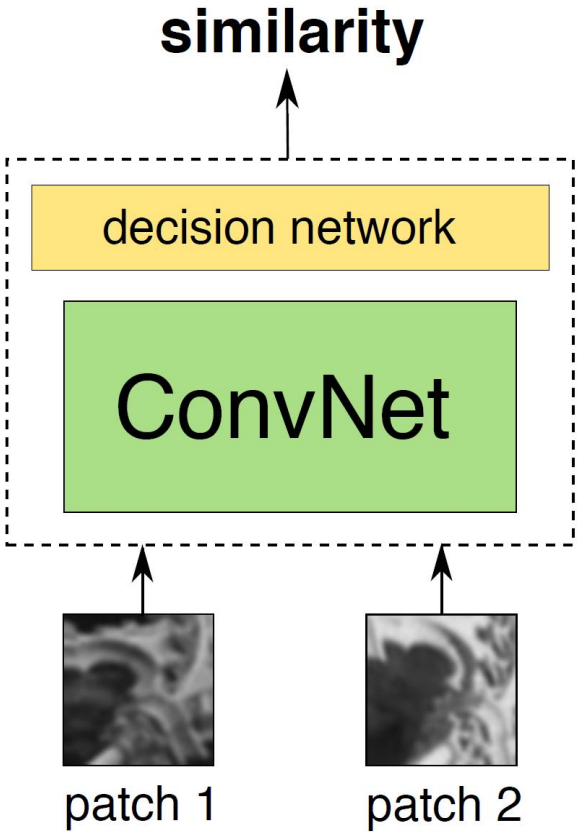


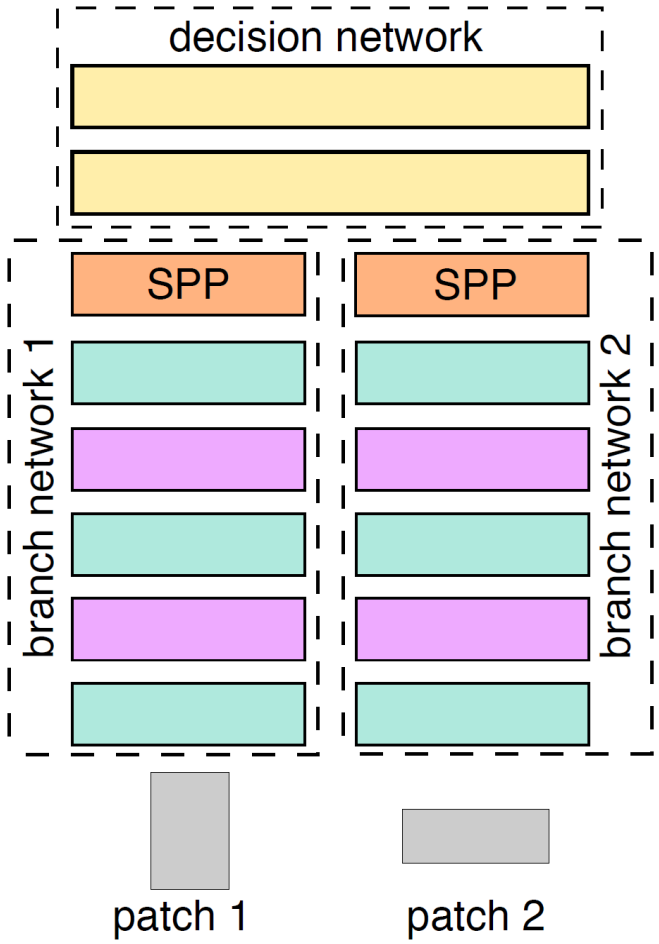
Pyramid Stereo Matching Network



Chang, J. R., & Chen, Y. S. (2018). Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5410-5418).

Compare Image Patches

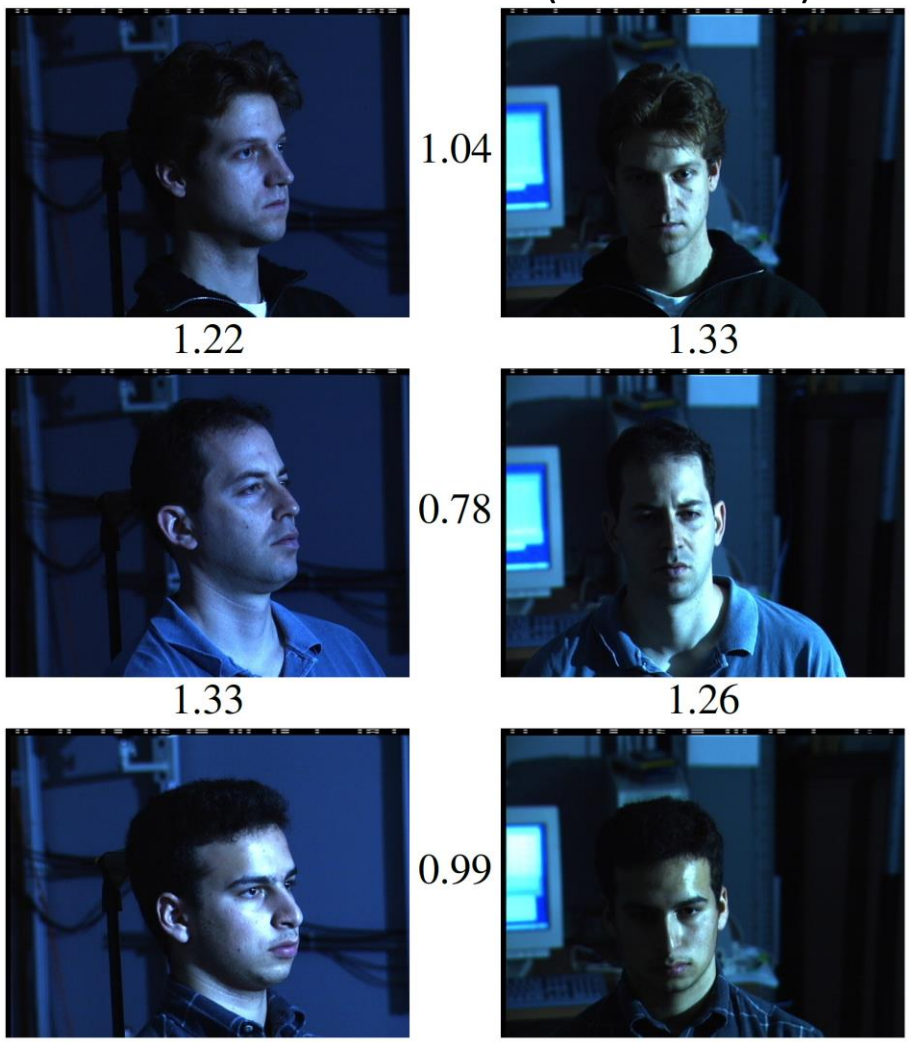


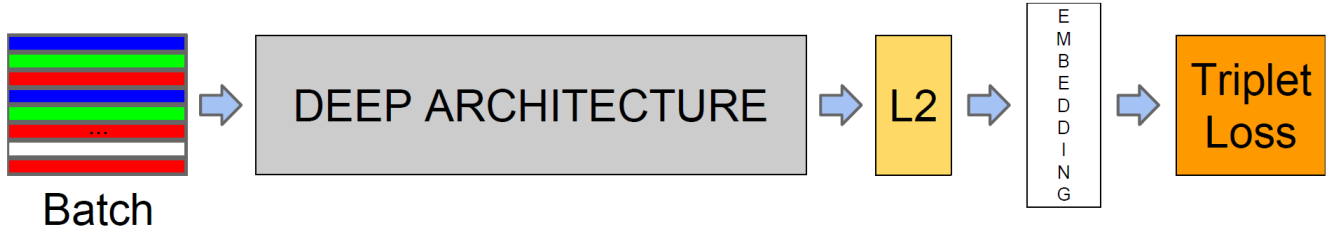


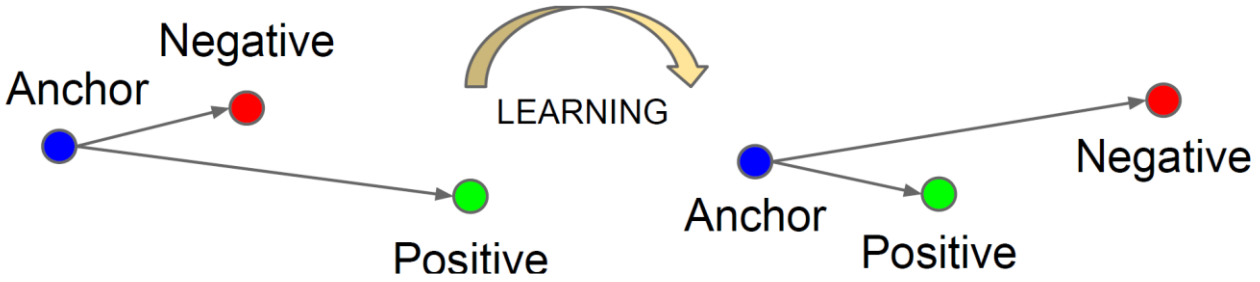
Zagoruyko, S., & Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4353-4361).

FaceNet

Similarity between faces
smaller is more similar(0 is identical)

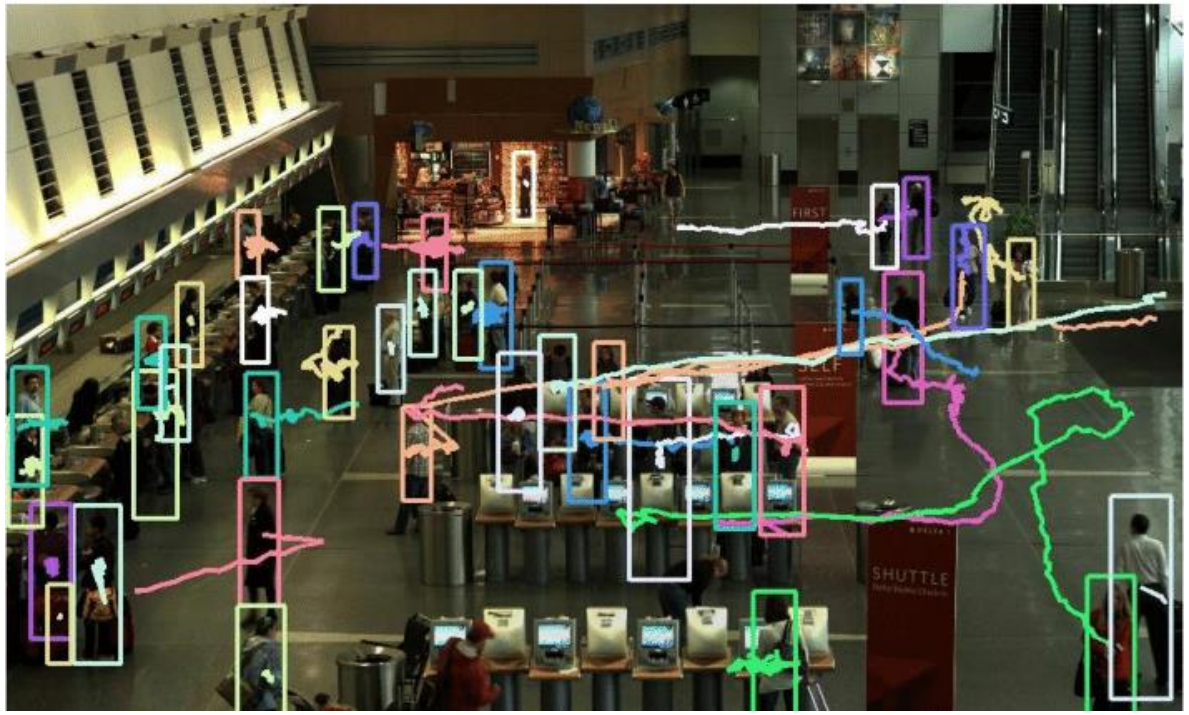




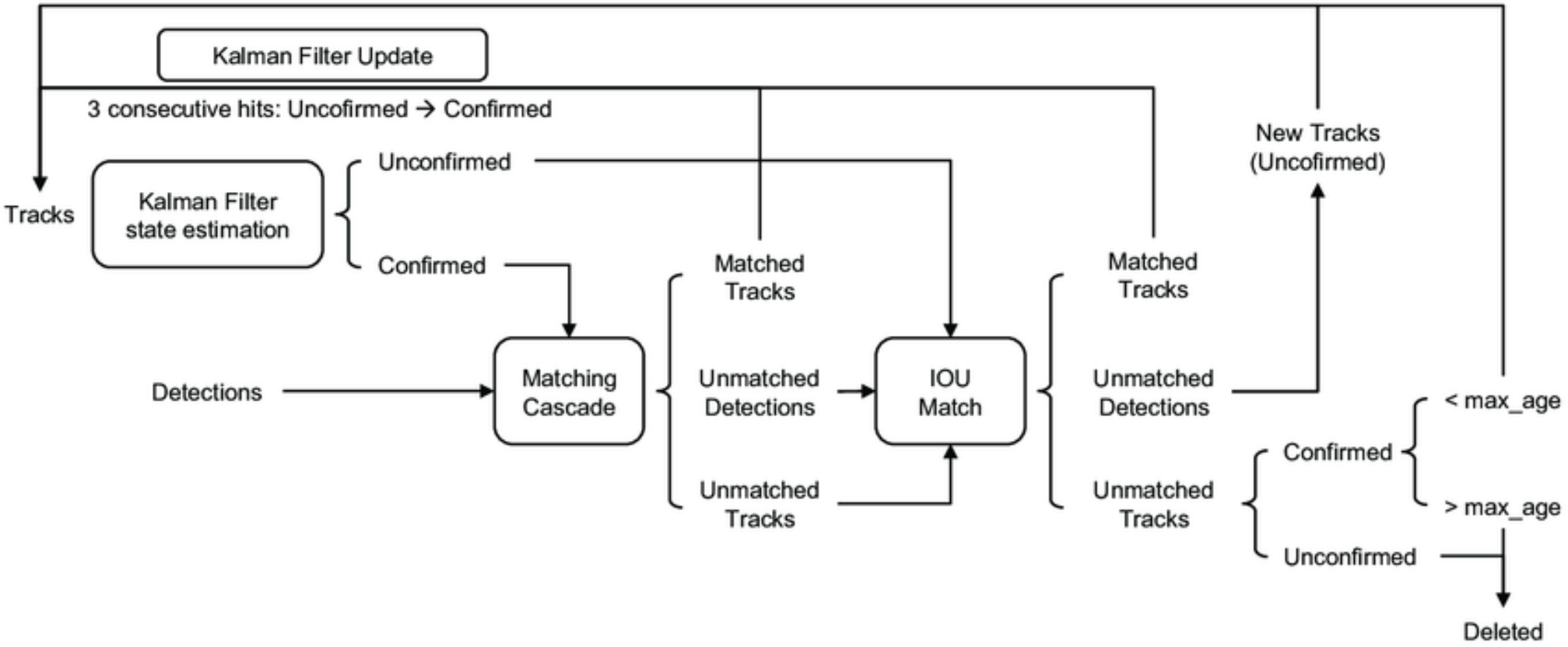


Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).

Tracking



DeepSORT



Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)* (pp. 3645-3649). IEEE.

DeepSORT

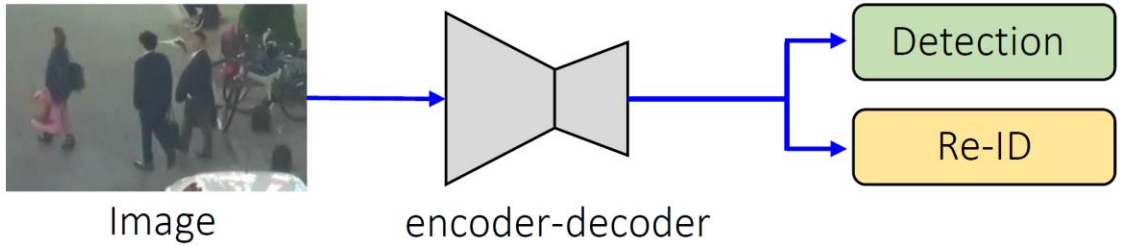
Listing 1 Matching Cascade

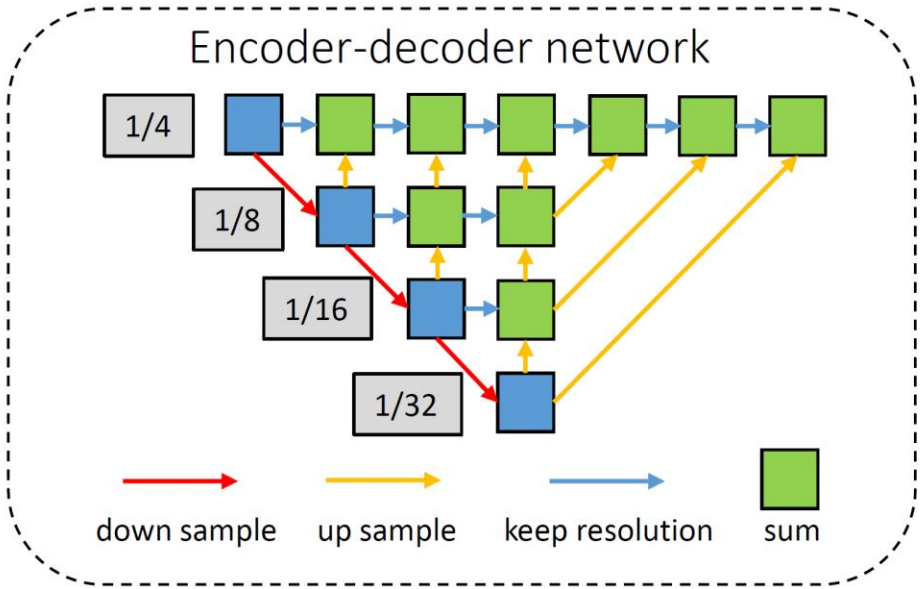
Input: Track indices $\mathcal{T} = \{1, \ldots, N\}$, Detection indices $\mathcal{D} = \{1, \ldots, M\}$, Maximum age $A_{\max}$

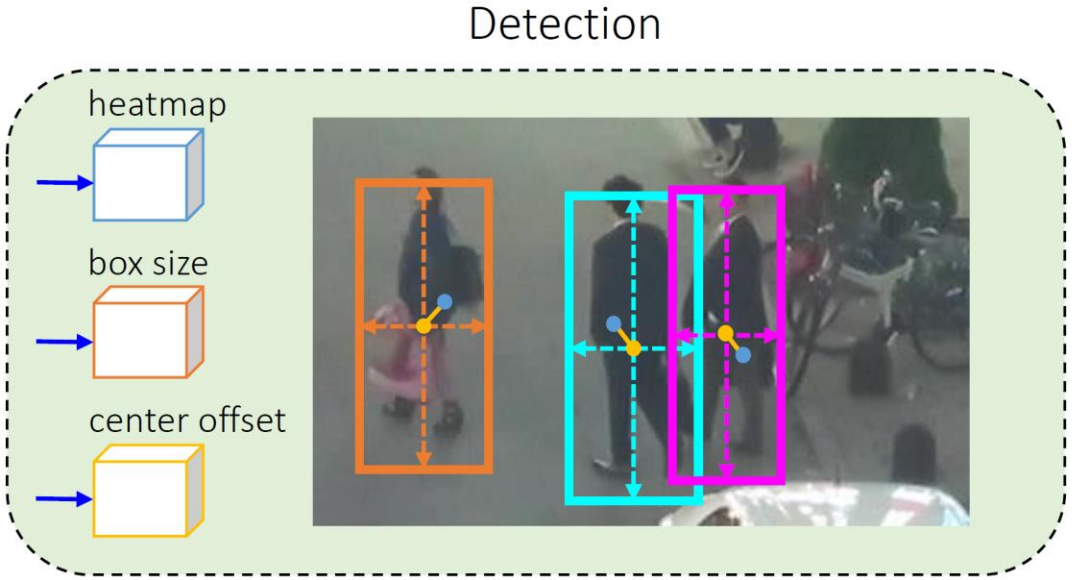
Compute cost matrix $\boldsymbol{C} = [c_{i,j}]$ using Eq. 5
Compute gate matrix $\boldsymbol{B} = [b_{i,j}]$ using Eq. 6
Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
Initialize set of unmatched detections $\mathcal{U} \leftarrow \mathcal{D}$
for $n \in \{1, \ldots, A_{\max}\}$ **do**
    Select tracks by age $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
    $[x_{i,j}] \leftarrow \text{min\_cost\_matching}(\boldsymbol{C}, \mathcal{T}_n, \mathcal{U})$
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
end for
return $\mathcal{M}, \mathcal{U}$

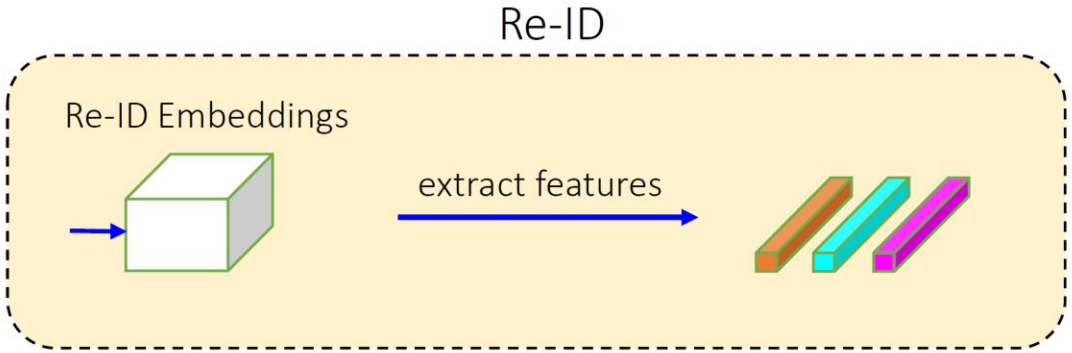
Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and ℓ_2 normalization		128

FairMOT



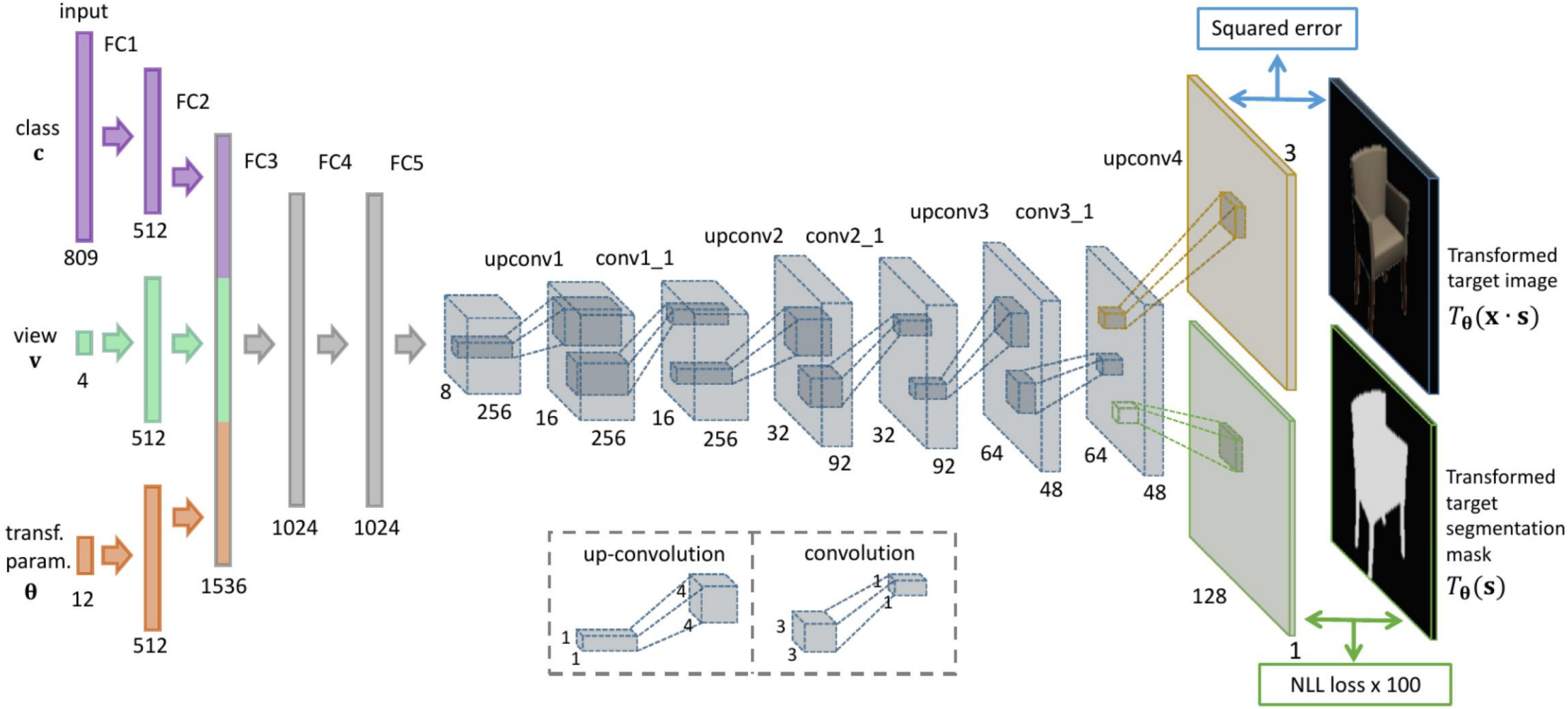






Zhang, Y., Wang, C., Wang, X., Zeng, W., & Liu, W. (2021). Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, *129*(11), 3069-3087.

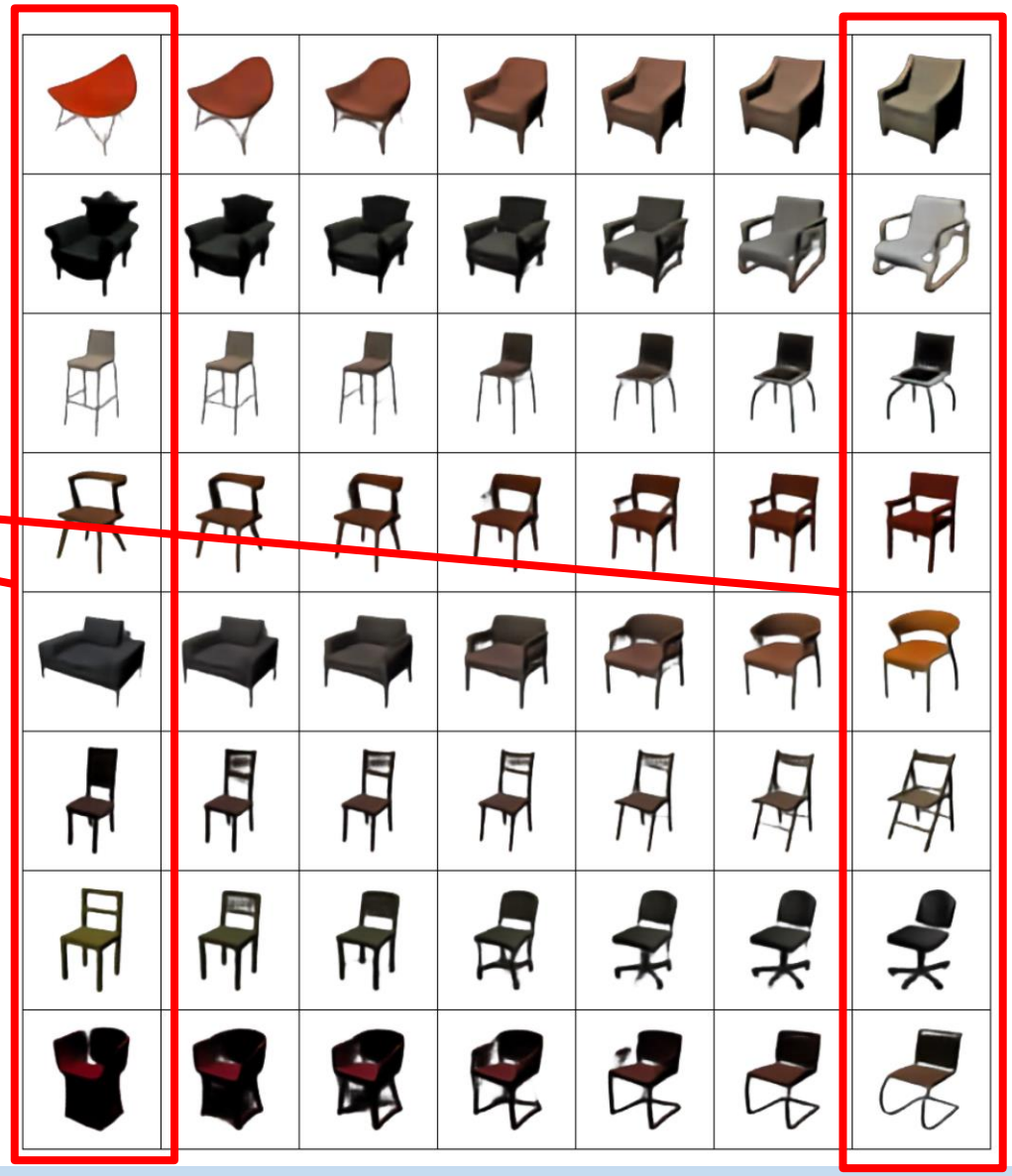
CNN for Image Generation



A. Dosovitskiy, J. T. Springenberg and T. Brox, "Learning to generate chairs with convolutional neural networks," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1538-1546, doi: 10.1109/CVPR.2015.7298761.

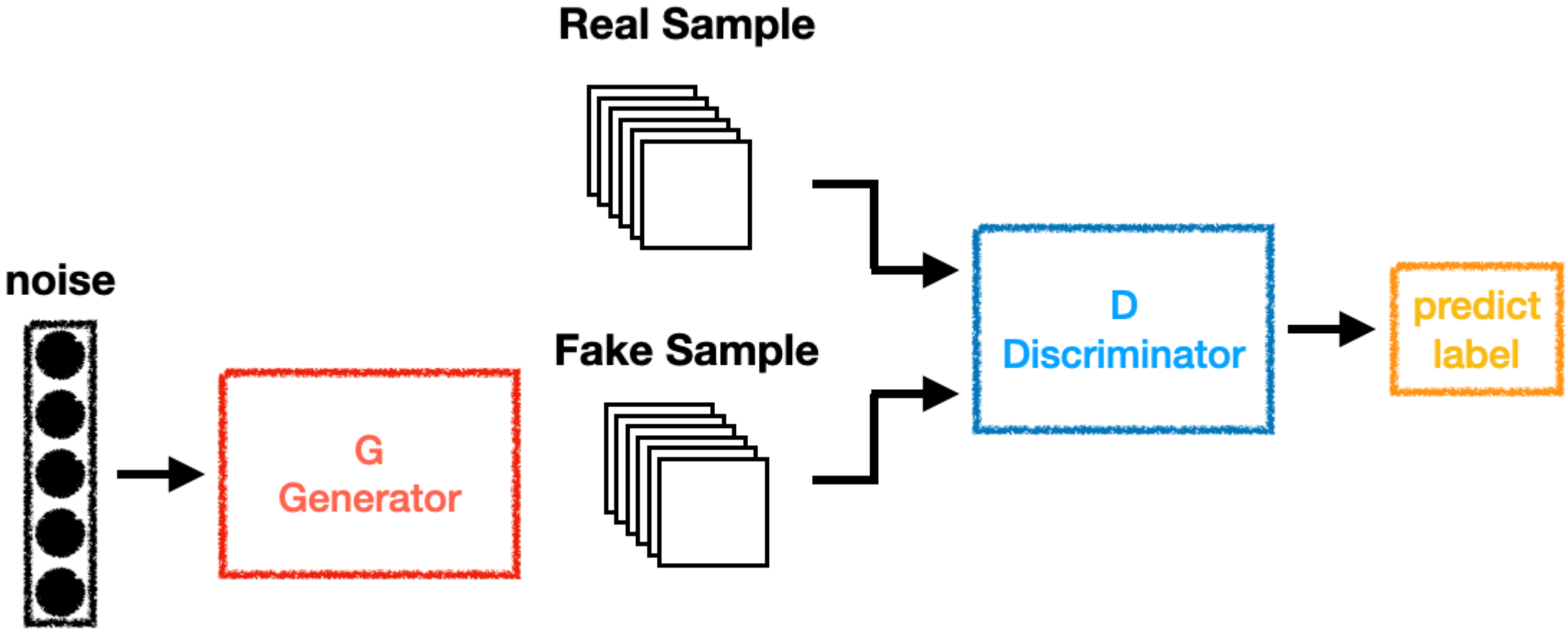
Chair Morphing

Exist in Training data



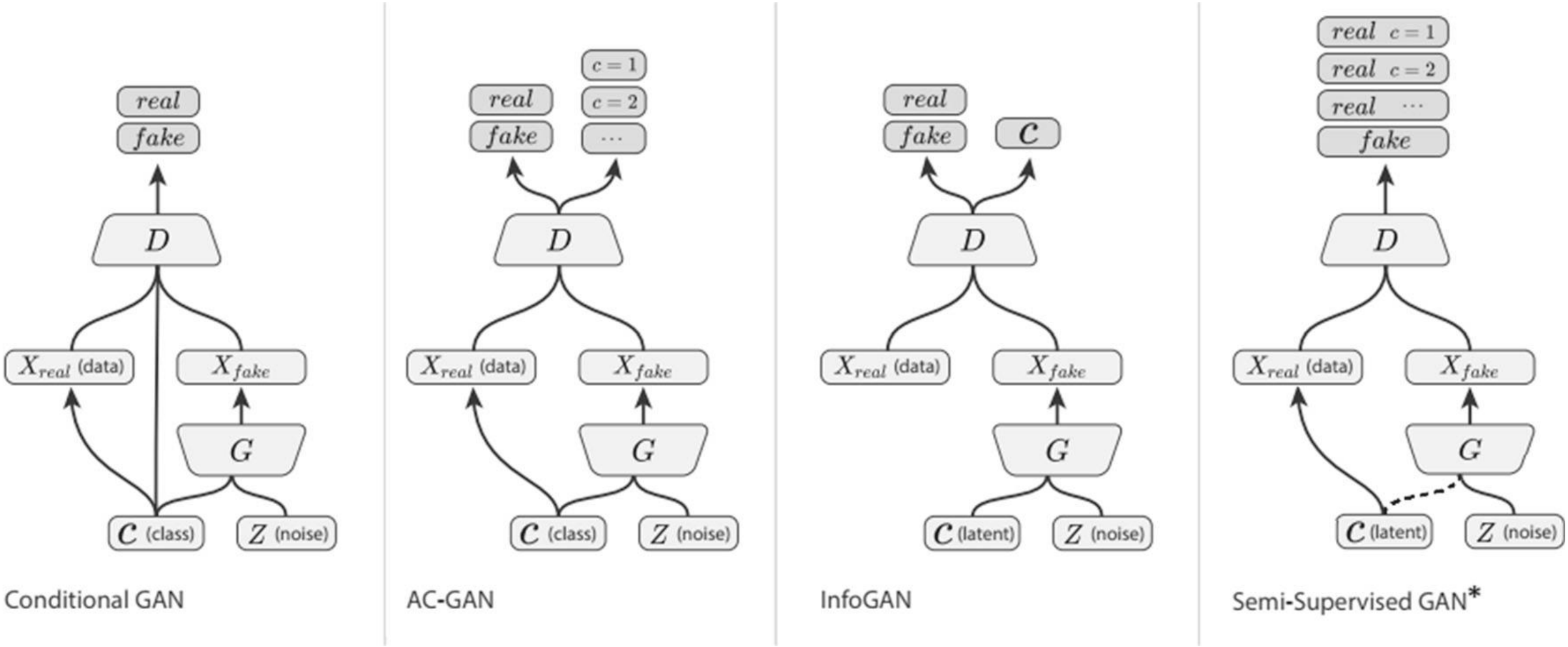
More REAL Image Generation

Not just discriminate, learn the distribution



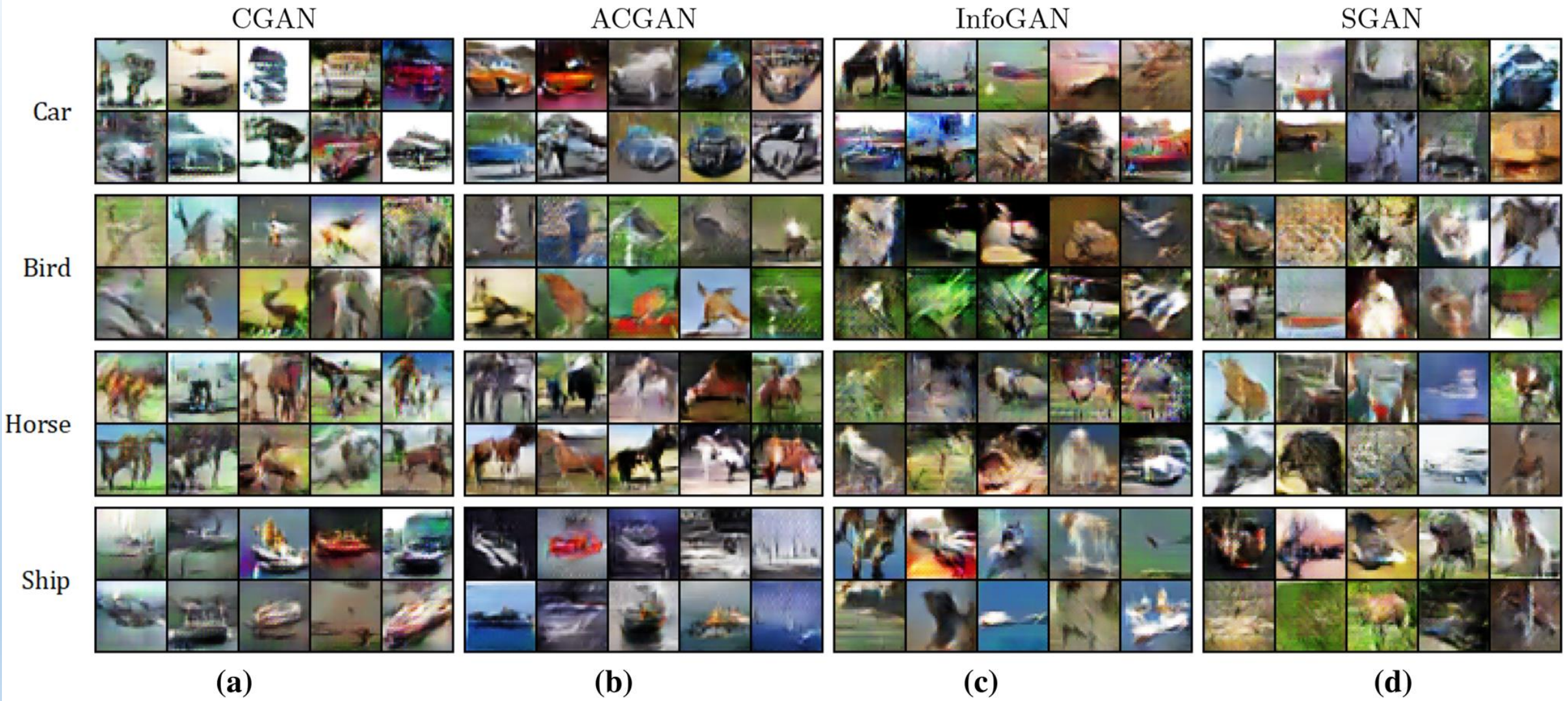
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Control the Generated Image



Benny, Y., Galanti, T., Benaim, S., & Wolf, L. (2021). Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, *129*(5), 1712-1731.

Control the Generated Image



Super Resolution

- 4x up-scaling

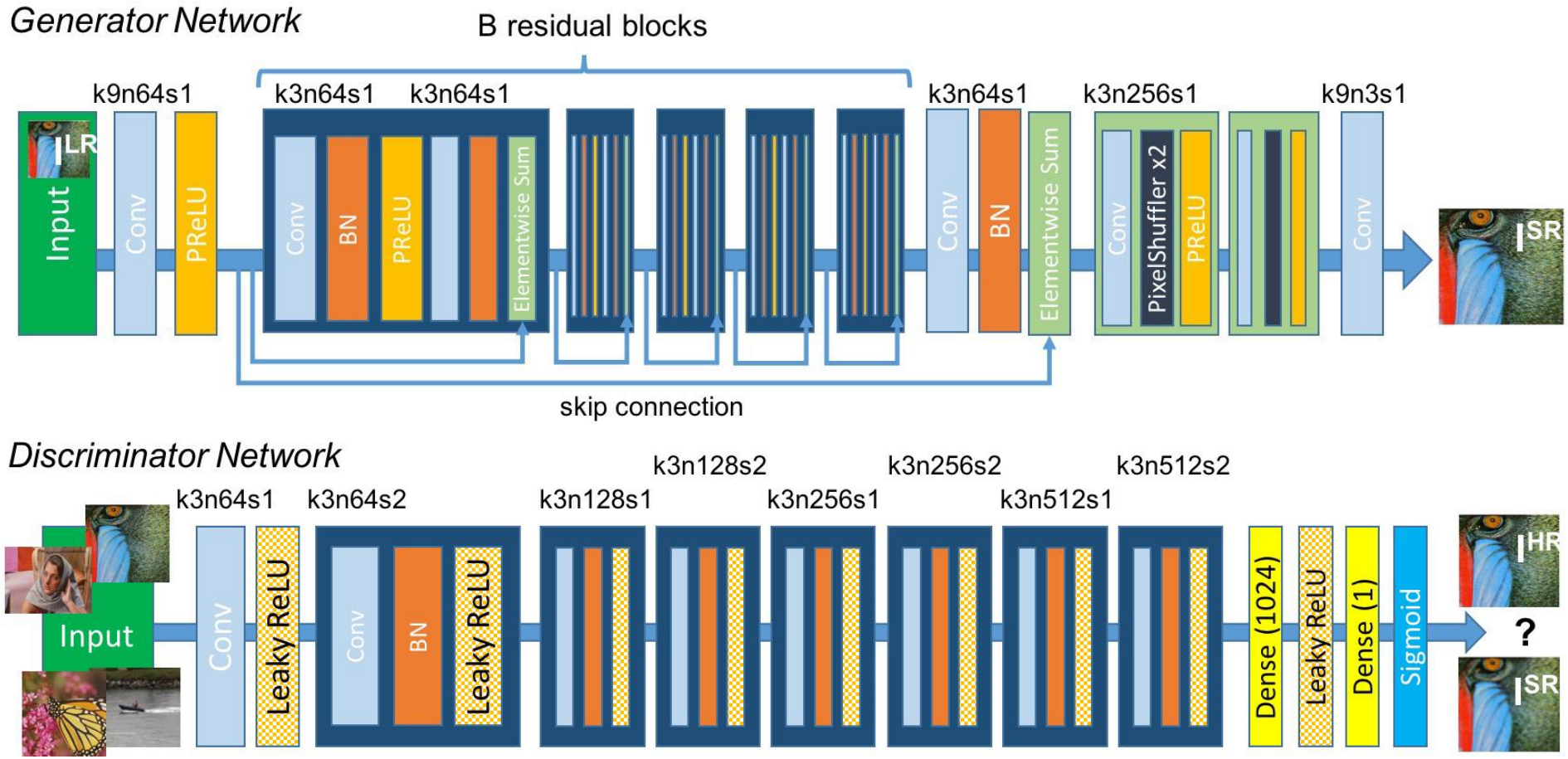
bicubic
(21.59dB/0.6423)

SRResNet
(23.53dB/0.7832)

SRGAN
(21.15dB/0.6868)

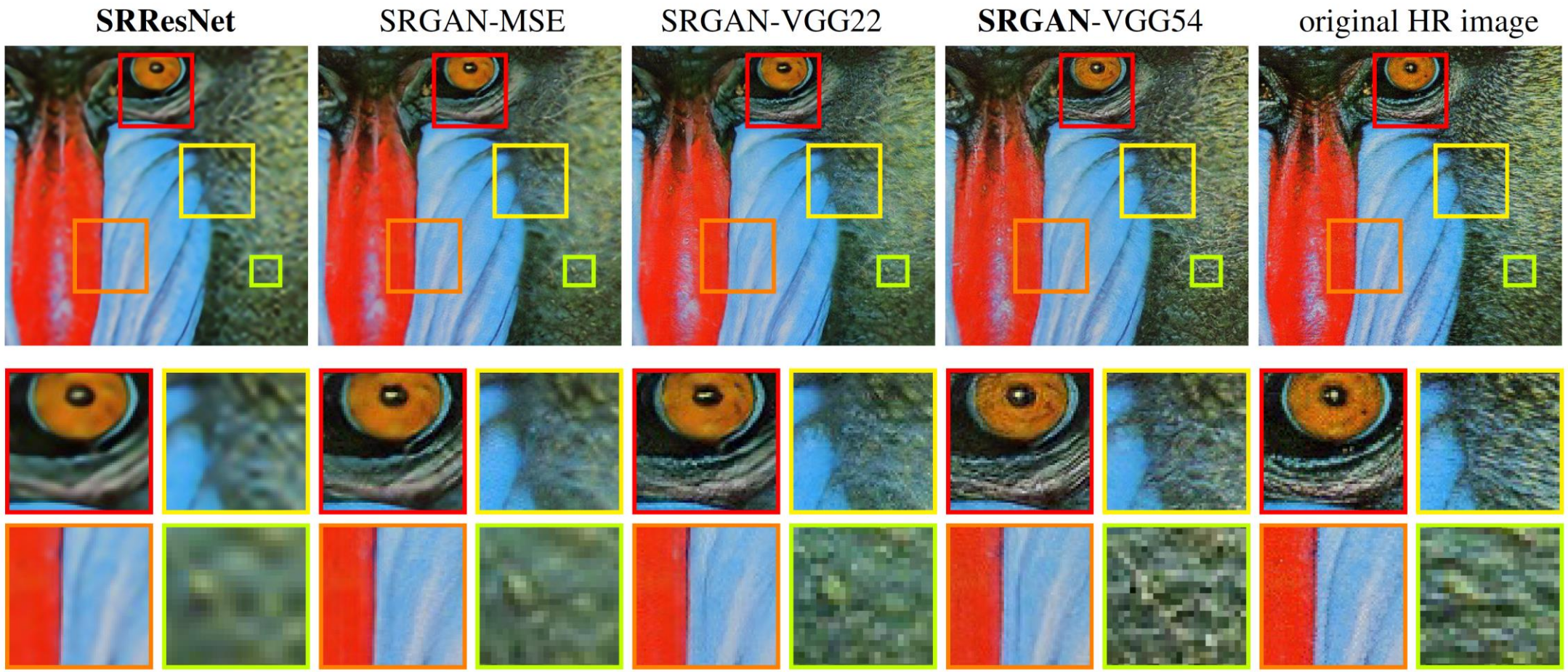
original

SRGAN

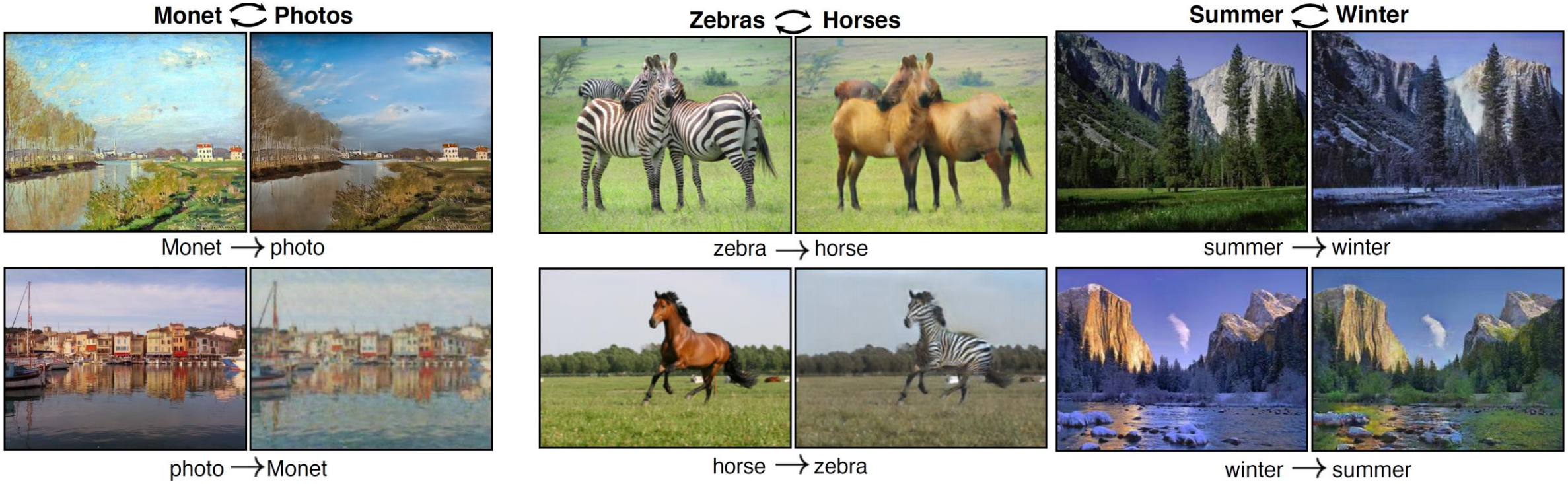


Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690).

SRGAN

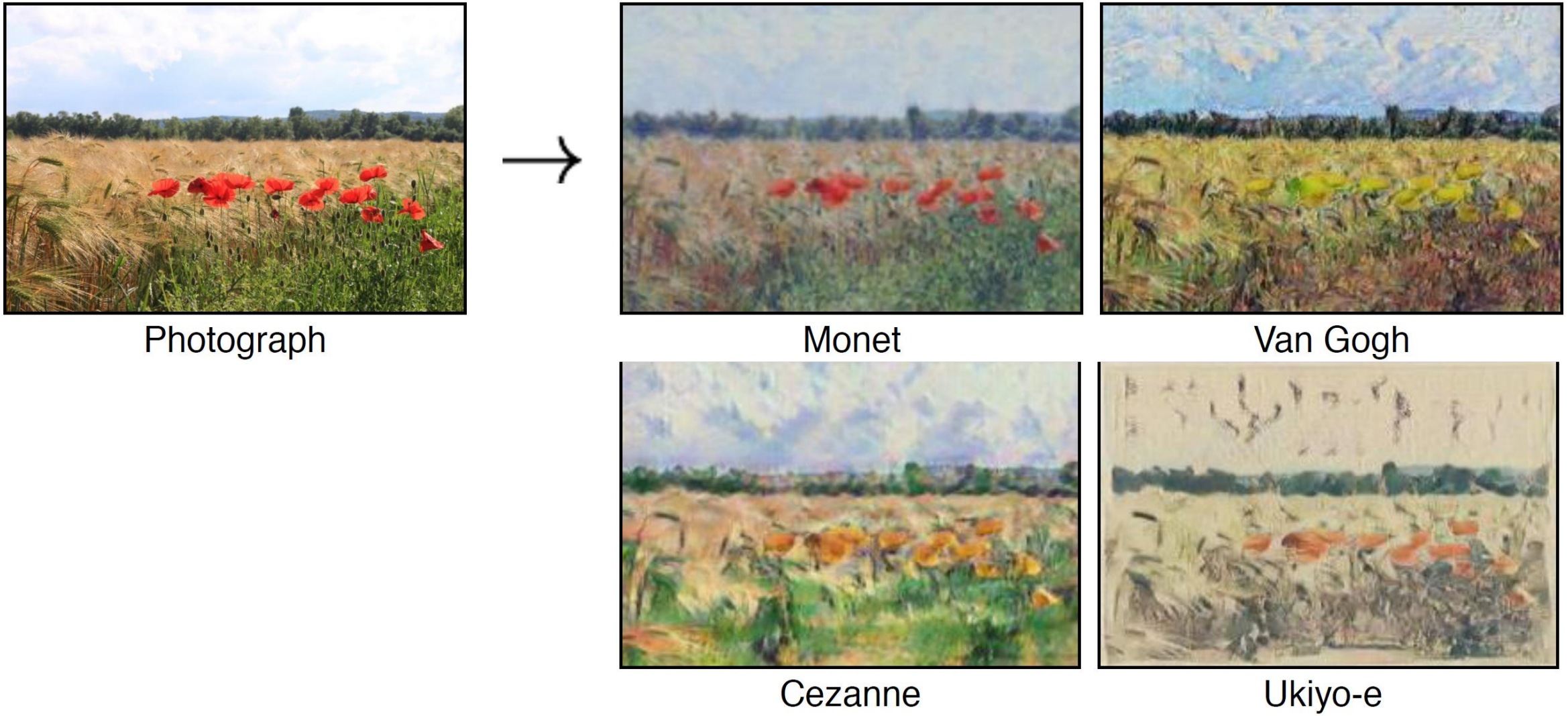

SRResNet
SRGAN-MSE
SRGAN-VGG22
SRGAN-VGG54
original HR image

Style Transfer



Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

Style Transfer



Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

CycleGAN

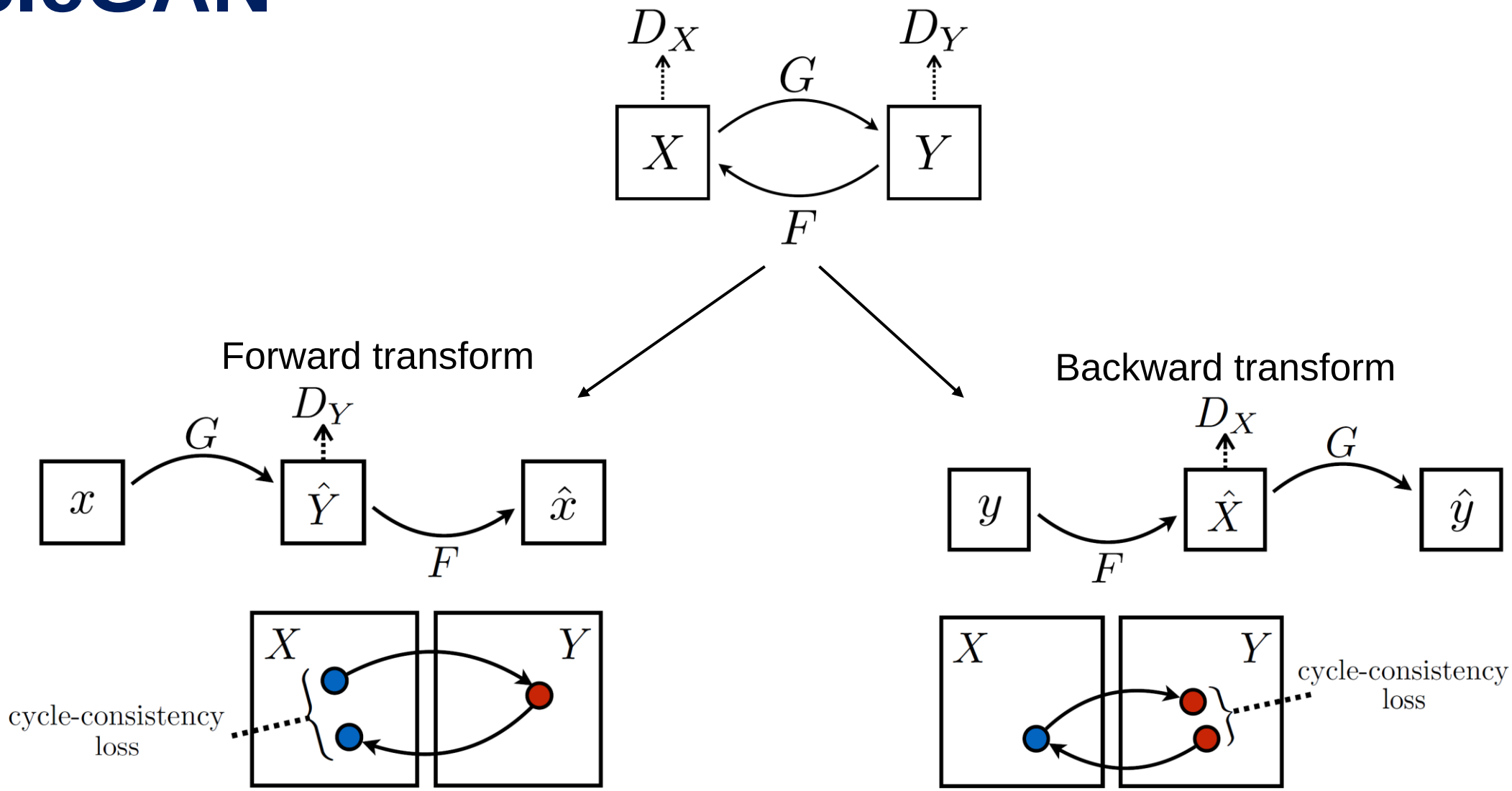
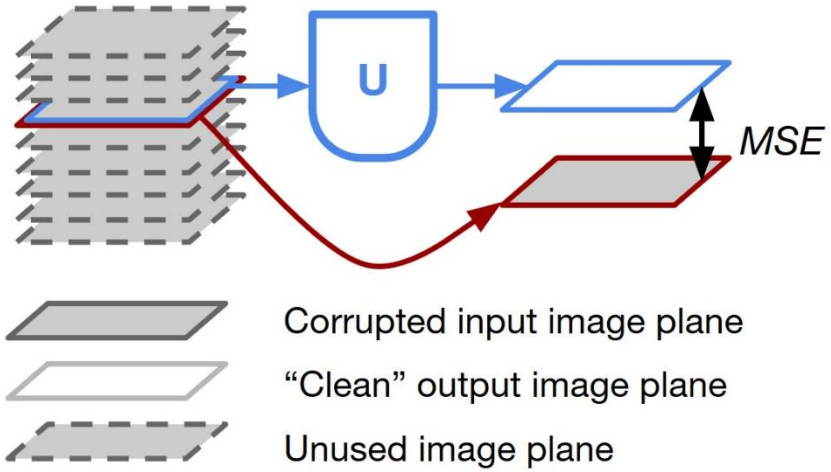
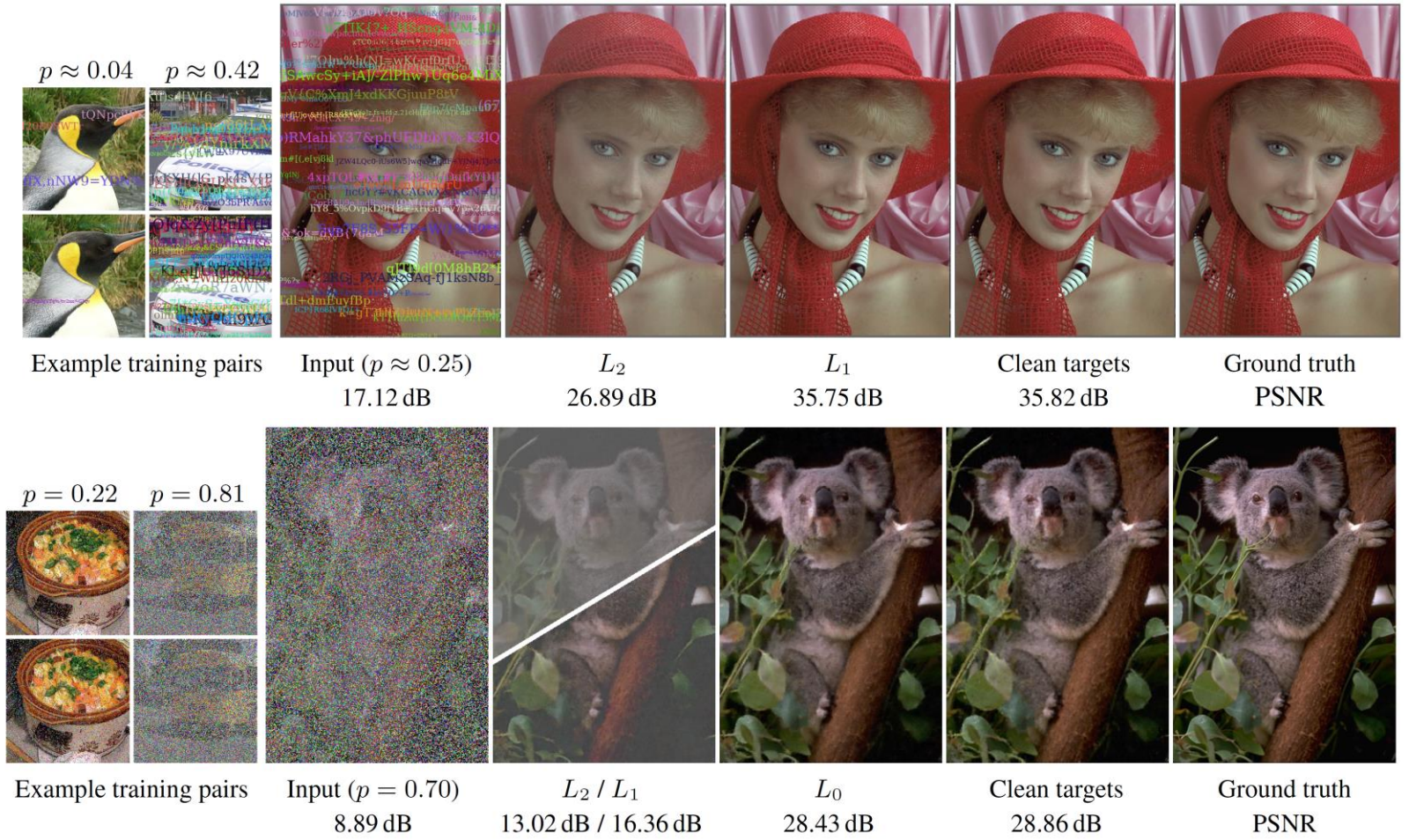

D_X
G
D_Y
X
Y
F
Forward transform
D_Y
G
x
$\hat{Y}$
$\hat{x}$
F
X
Y
cycle-consistency loss
Backward transform
D_X
G
y
$\hat{X}$
$\hat{y}$
F
X
Y
cycle-consistency loss

Image Restoration

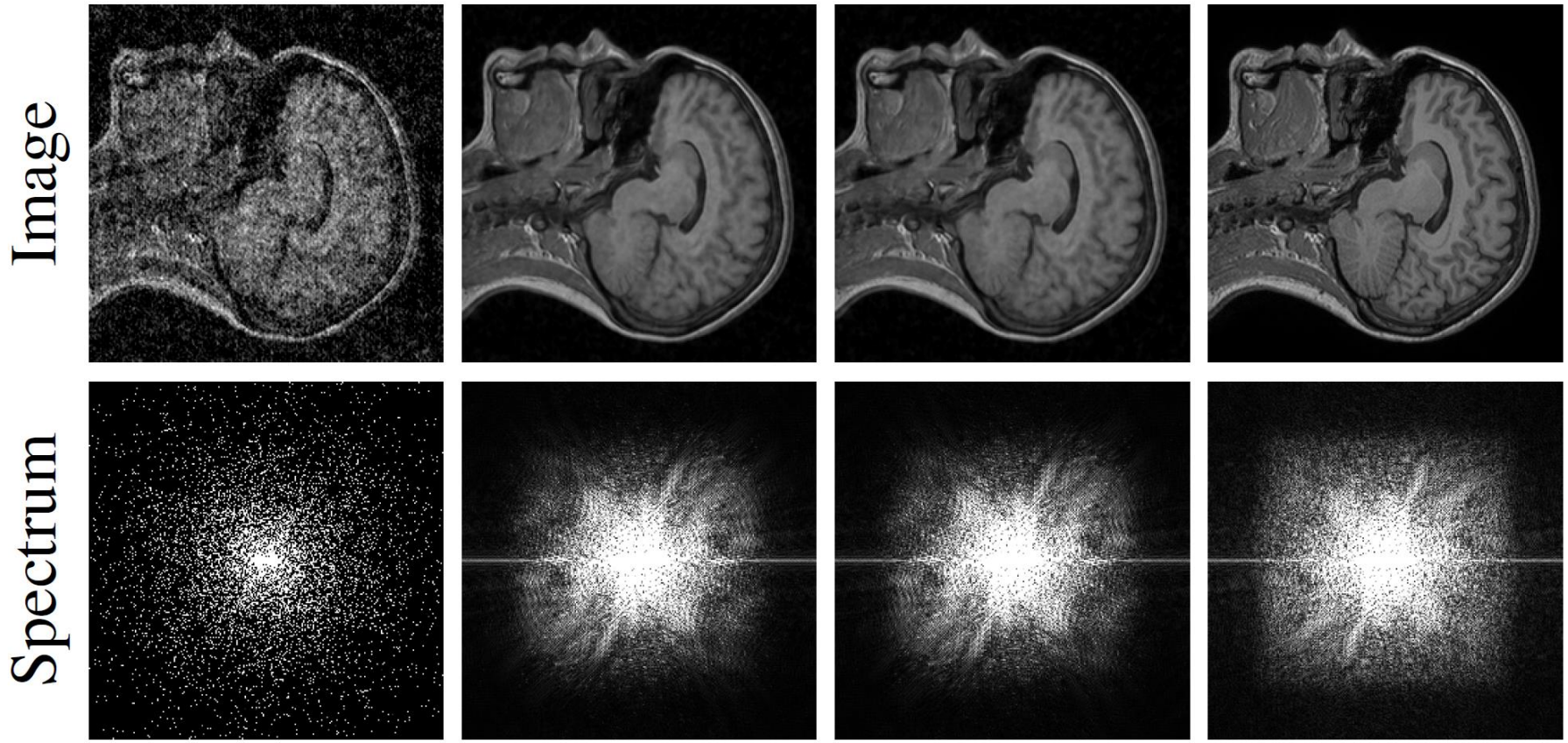
Noise2Noise





Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., & Aila, T. (2018). Noise2Noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*.

MRI Reconstruction



(a) Input 18.93 dB

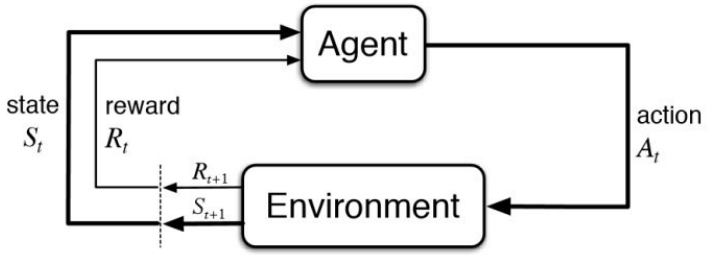
(b) Noisy trg. 29.77 dB

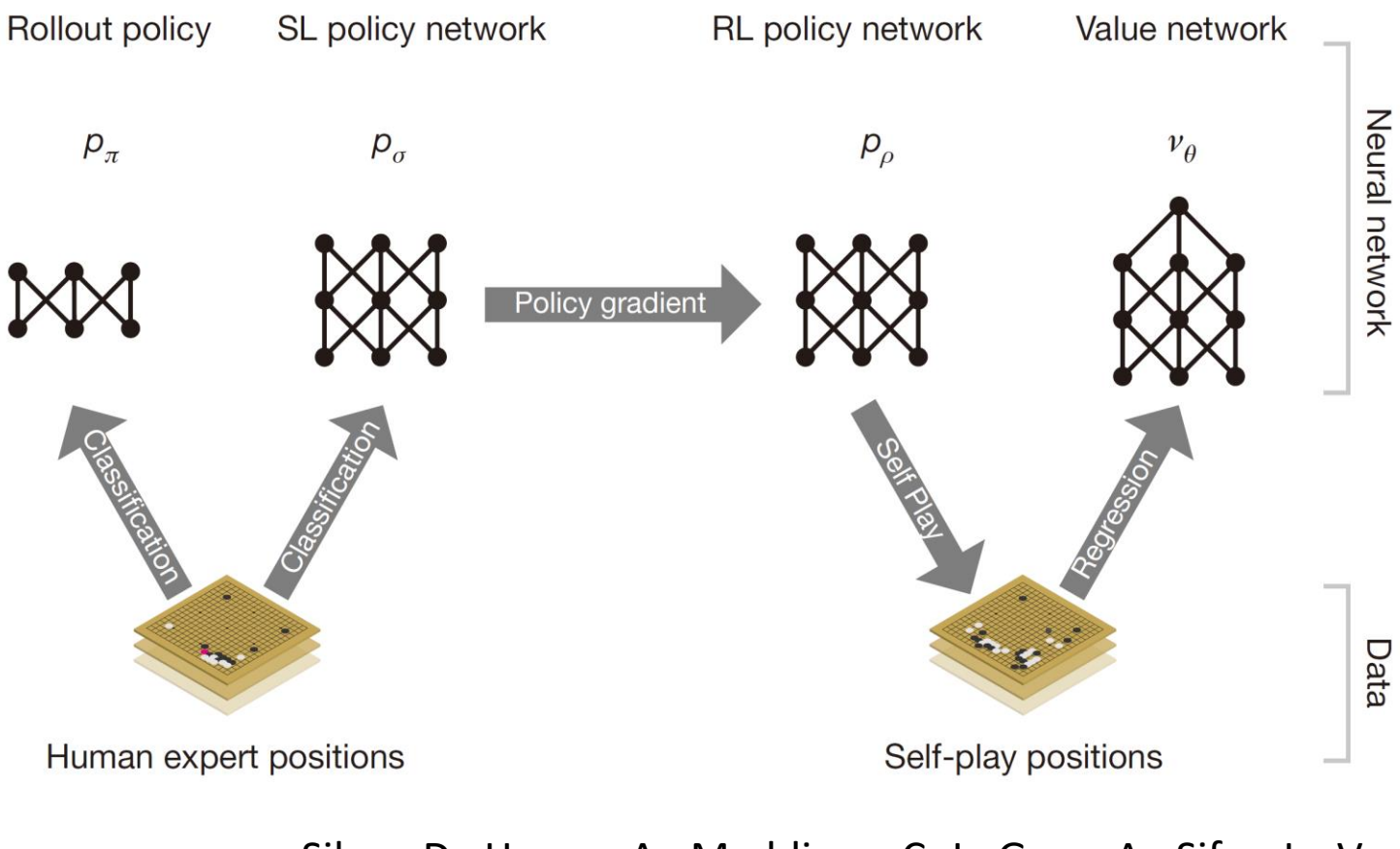
(c) Clean trg. 29.81 dB

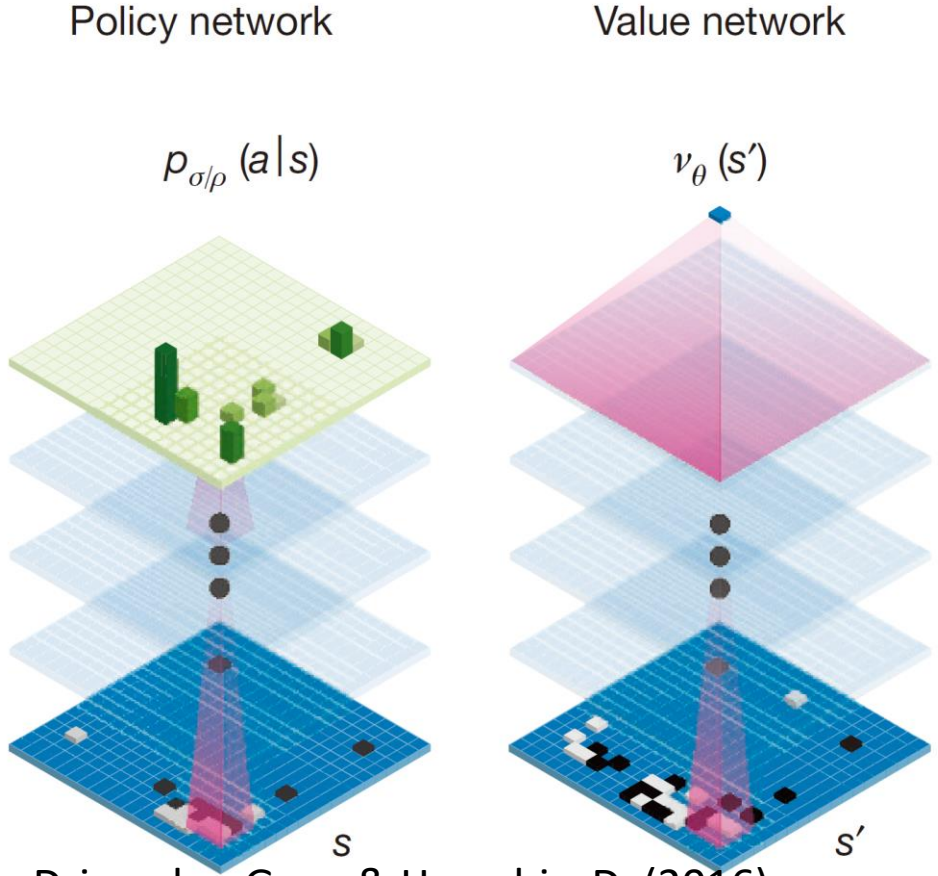
(d) Reference

AlphaGo – Exceed Human Expert

AlphaGo - Mastering the Game of Go

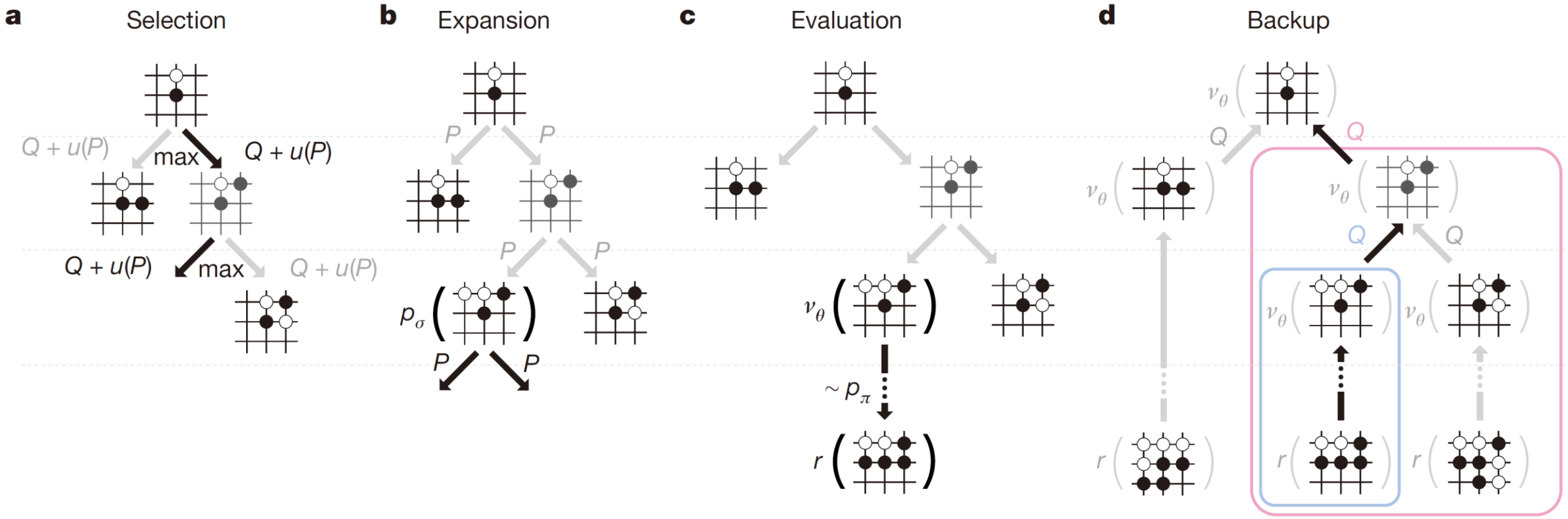






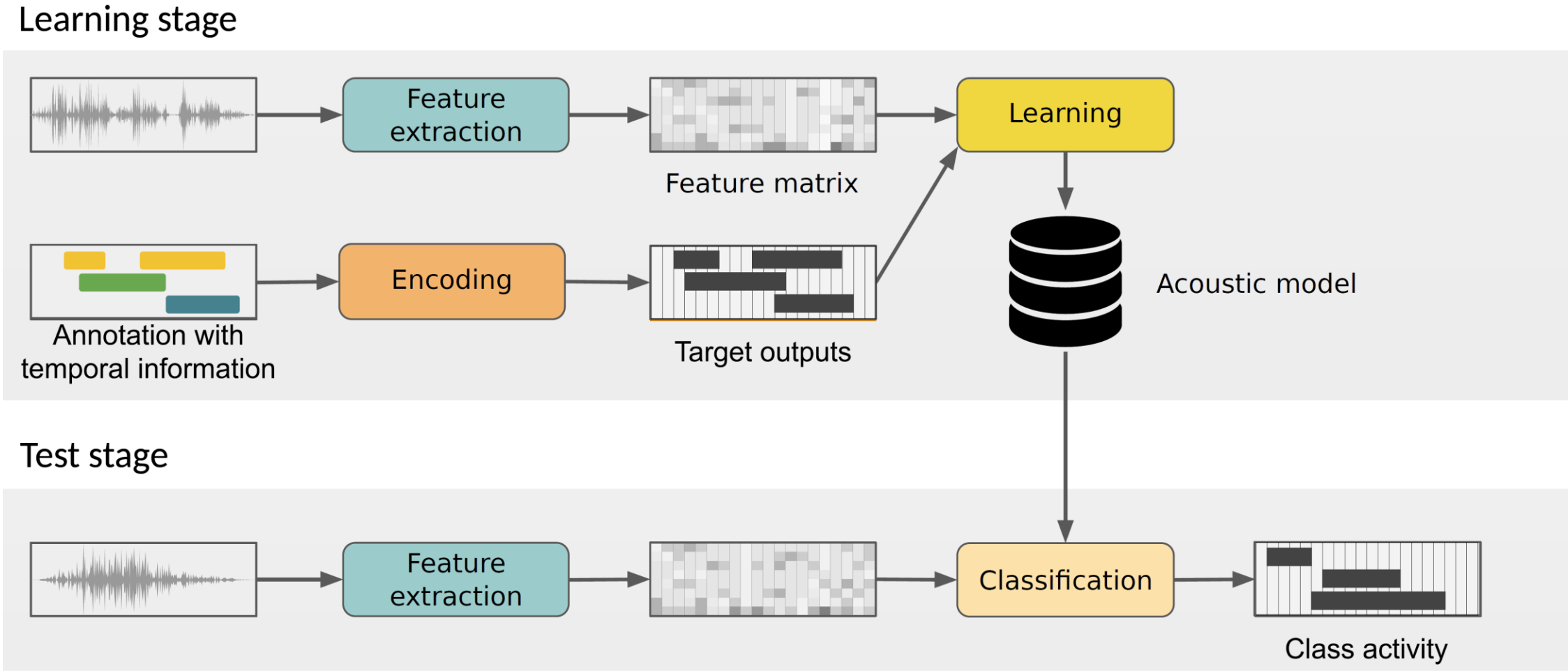
Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, *529*(7587), 484-489.

Monte Carlo Tree Search in AlphaGo



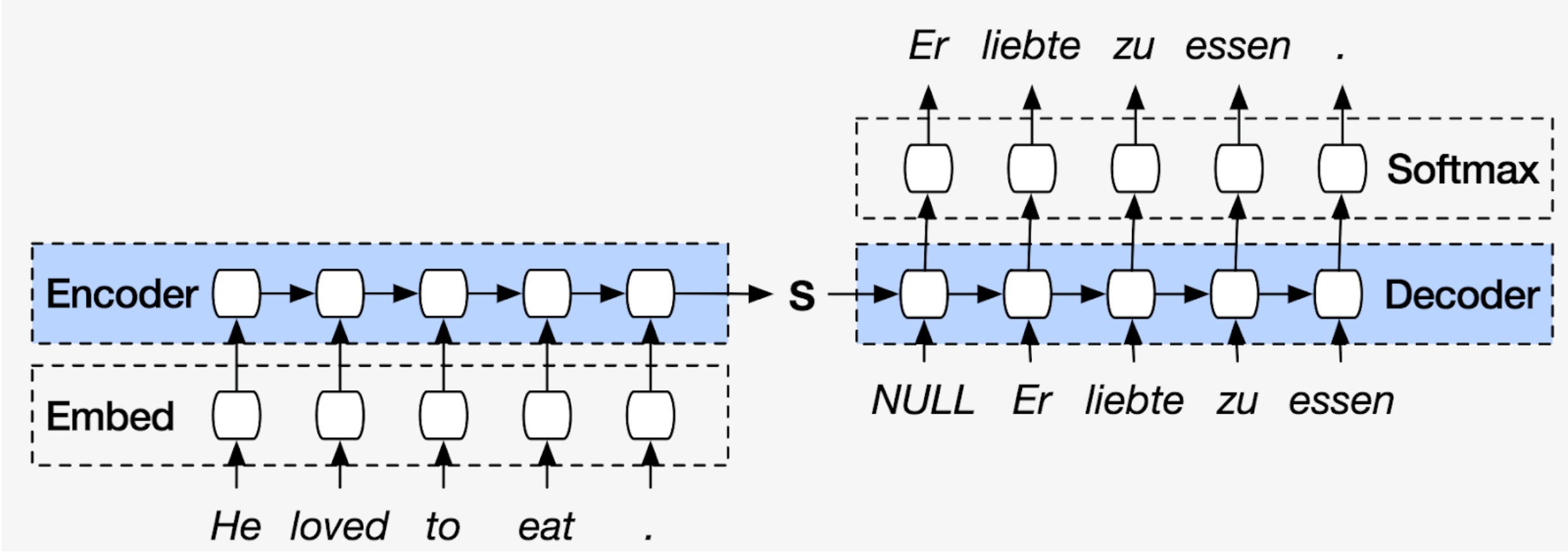
Sound Event Detection

Sound Event Detection

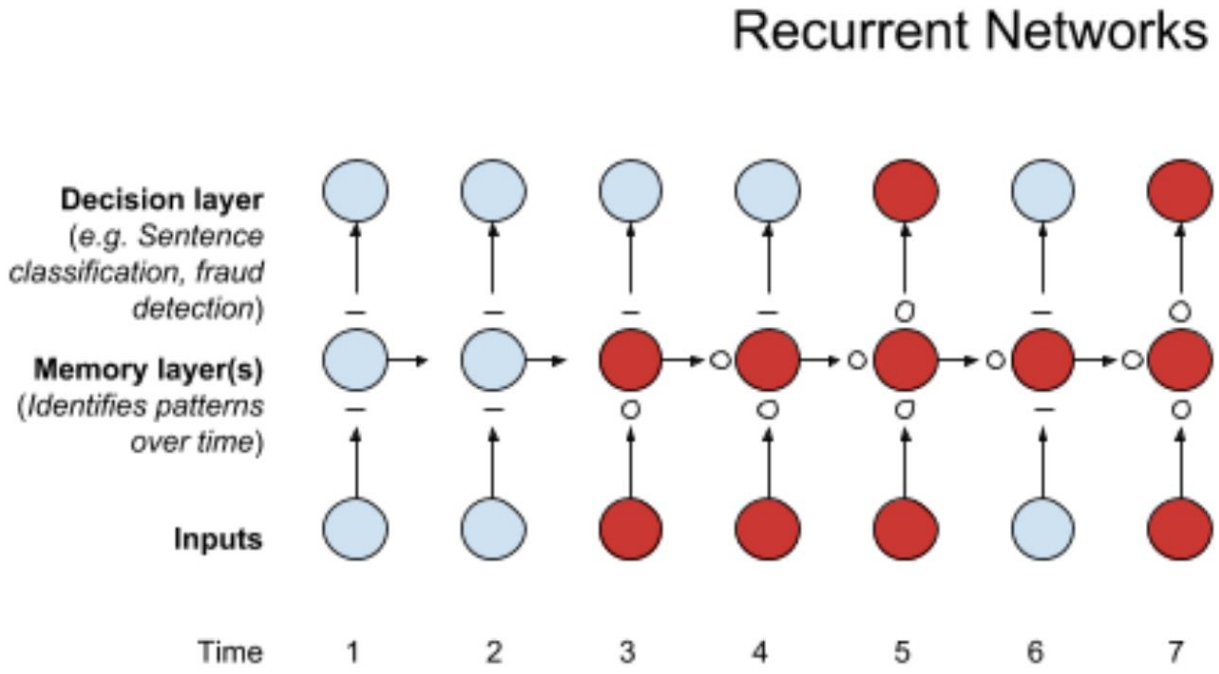


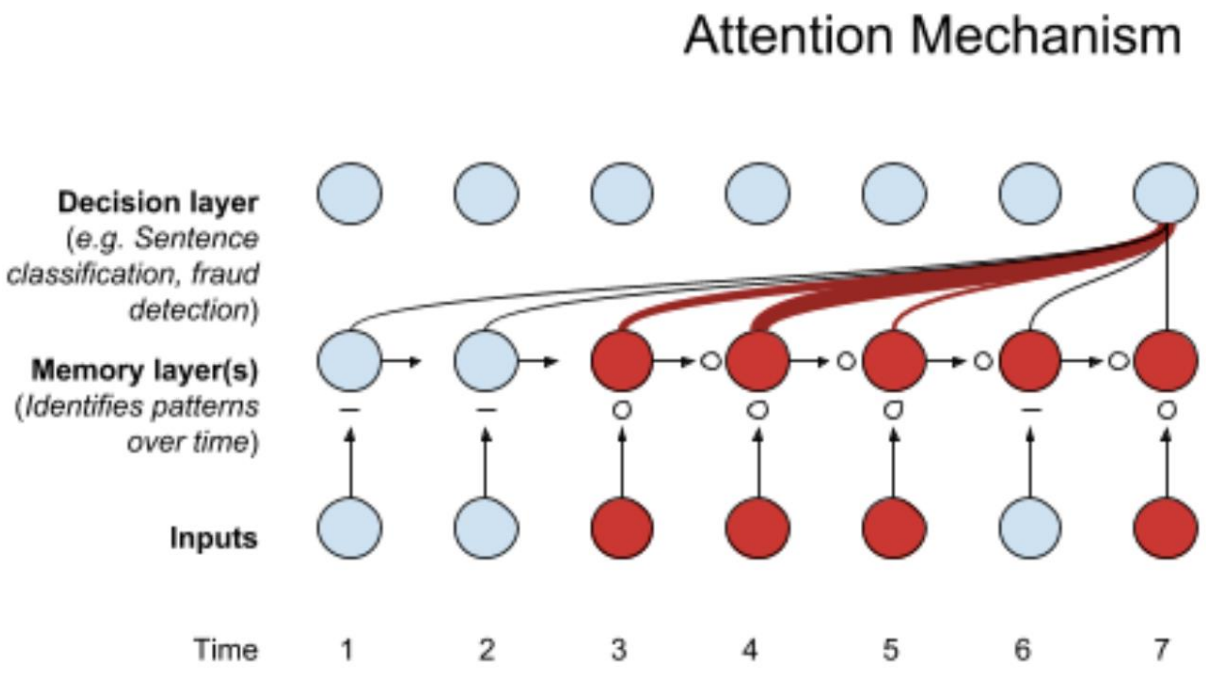
Mesaros, A., Heittola, T., Virtanen, T., & Plumbley, M. D. (2021). Sound event detection: A tutorial. *IEEE Signal Processing Magazine*, *38*(5), 67-83.

Machine Translation

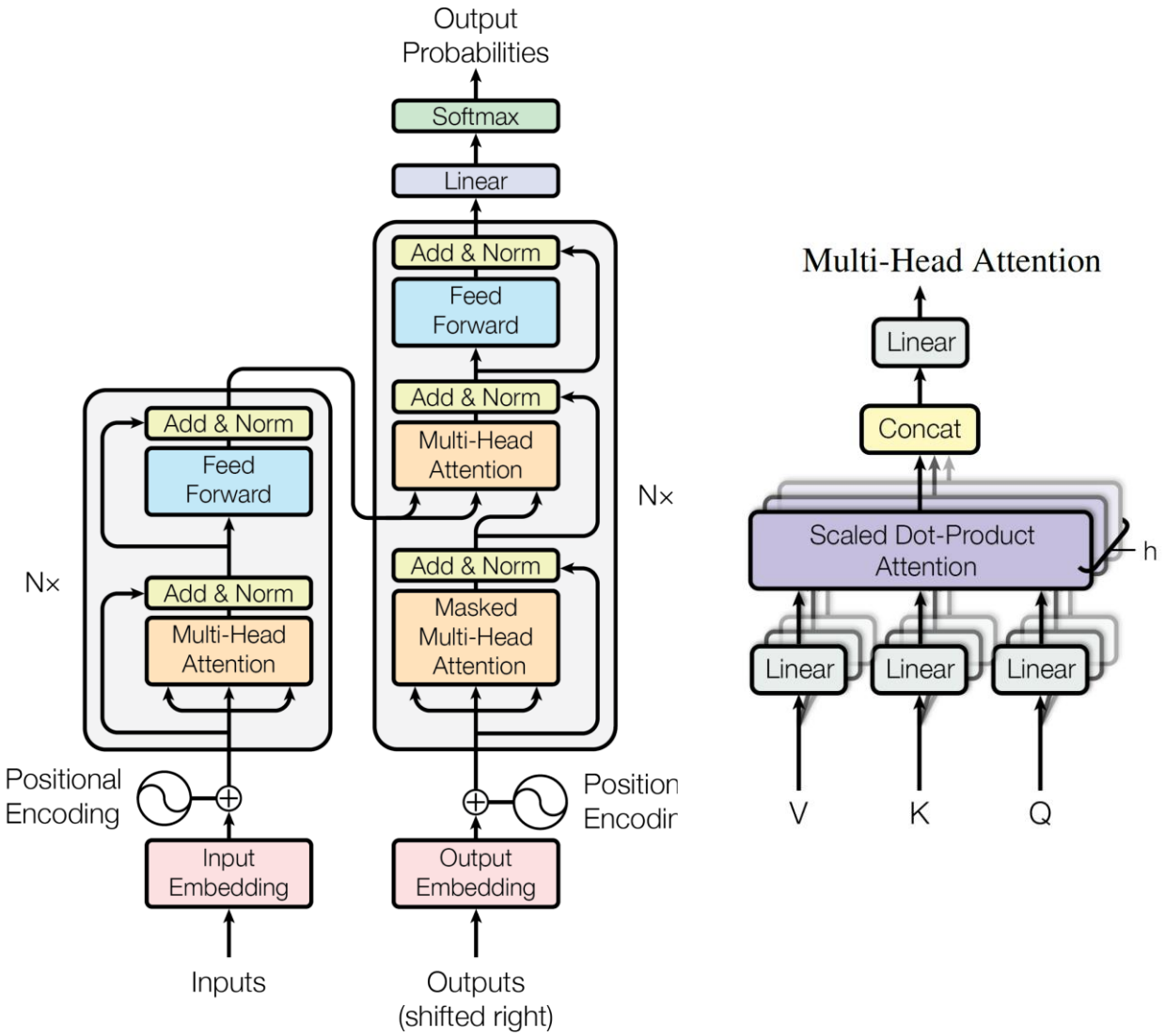


RNN vs. Attention





Transformer



Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Speech Recognition

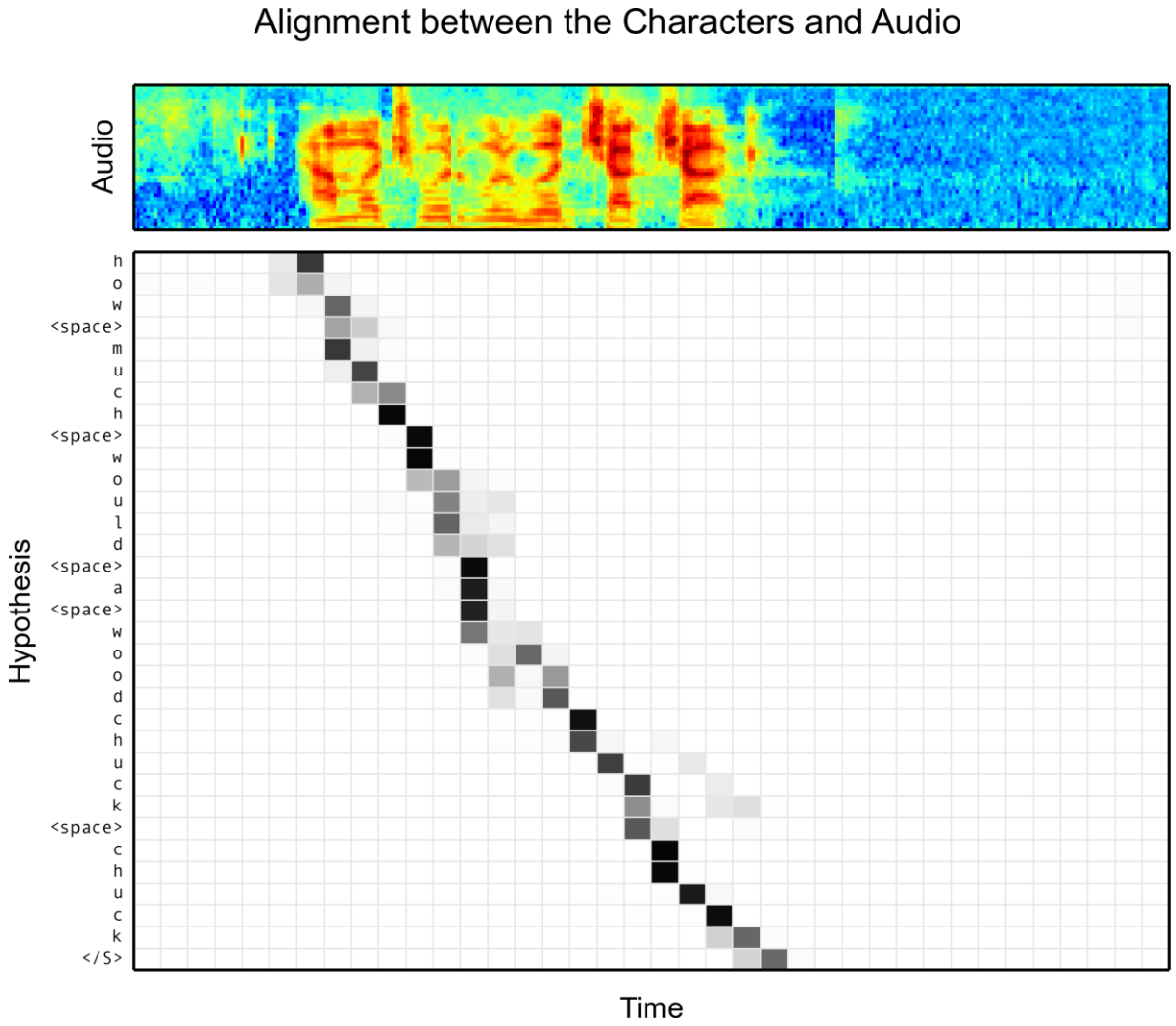


Figure 1 visualizes LAS with these two components. We provide more details of these components in the following sections.

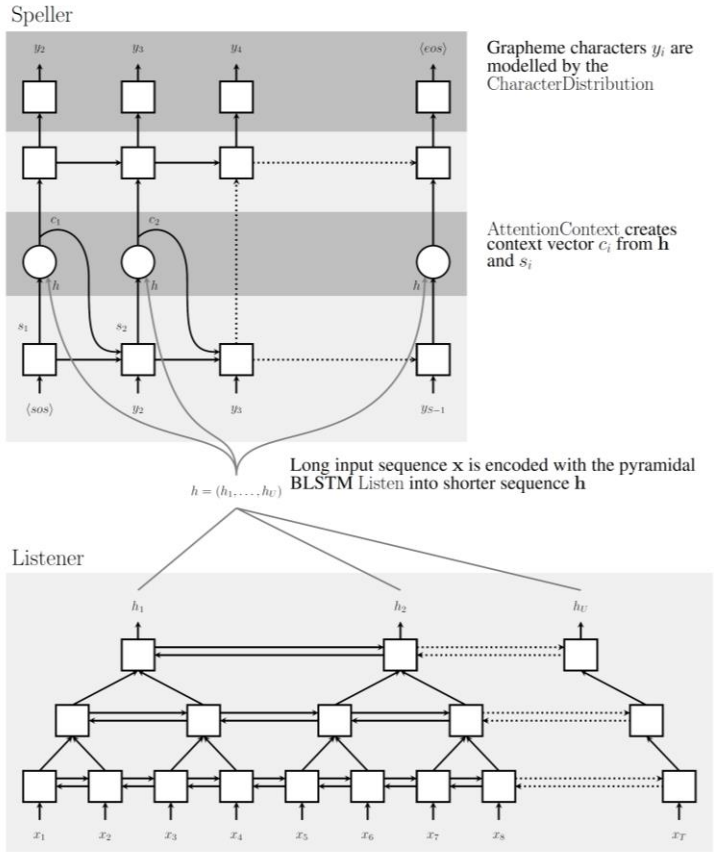


Figure 1: Listen, Attend and Spell (LAS) model: the listener is a pyramidal BLSTM encoding our input sequence $\mathbf{x}$ into high level features $\mathbf{h}$, the speller is an attention-based decoder generating the $\mathbf{y}$ characters from $\mathbf{h}$.

Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016, March). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4960-4964). IEEE.

Language Modeling

Web search engine / ...

I saw a cat

I saw a cat on the chair
I saw a cat running after a dog
I saw a cat in my dream
I saw a cat book

Translation service / mail agent / ...

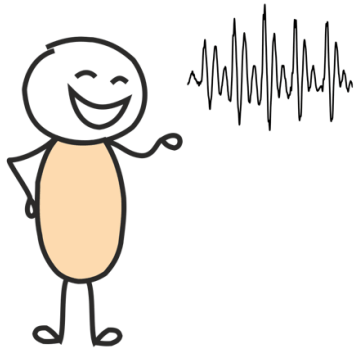
I saw a ca

car ↵

Keyboard / mail agent / ...

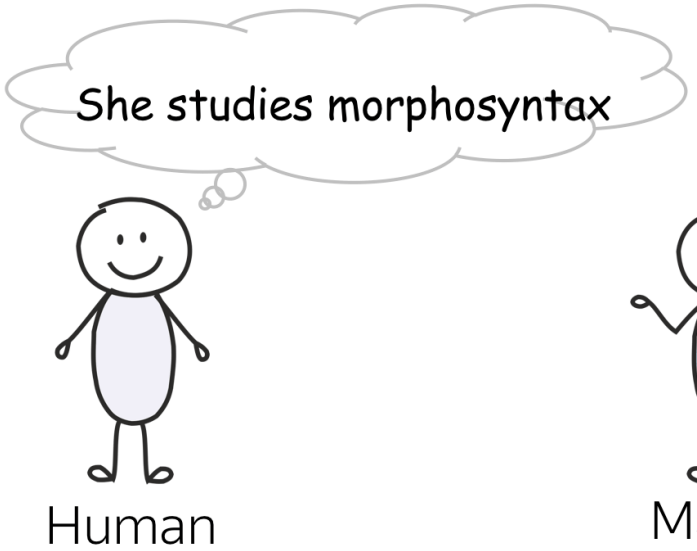
I saw a catt

cat
car

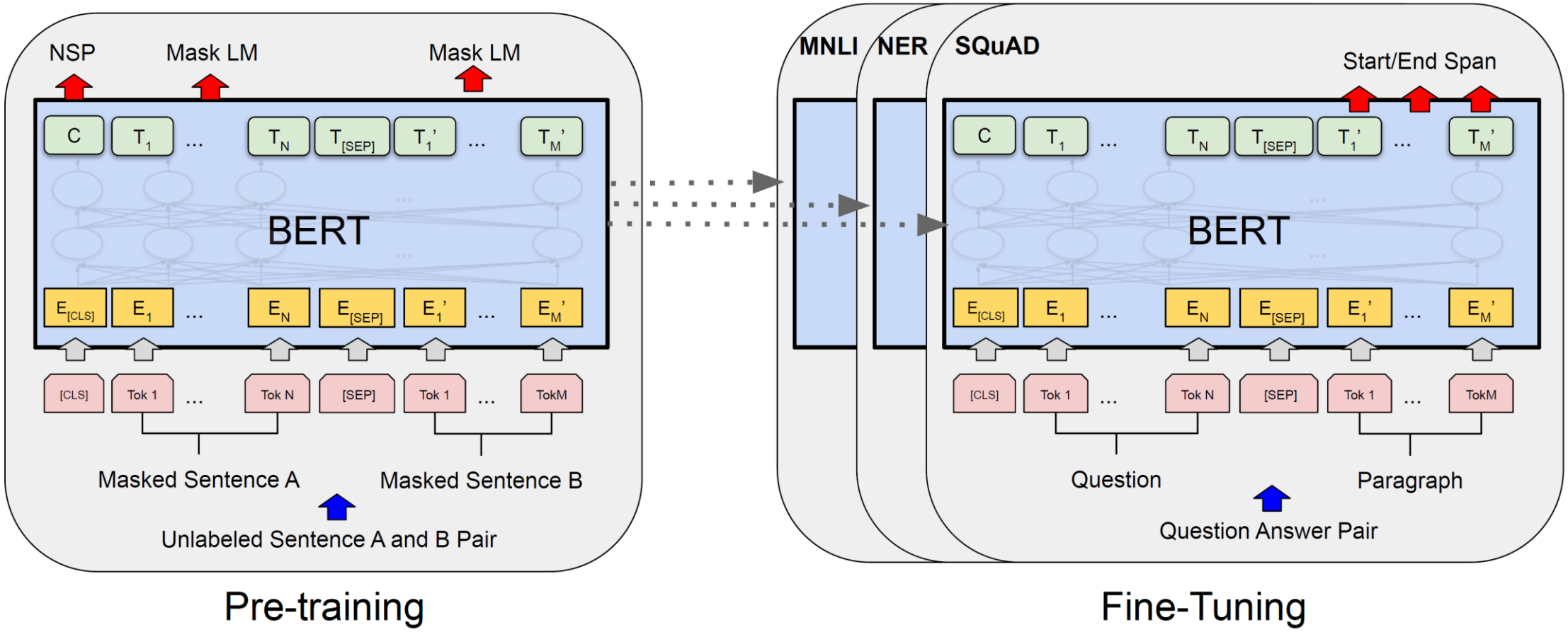


Similarly sounding options

She studies morphosyntax
She studies more faux syntax
She studies morph or syntax
....



BERT



Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

The AI invasion

- Robots
- Public/Personal Services
- Virtual Reality
- Augmented Reality
- Metaverse
- ChatGPU

More complex applications,
demands more power, energy, computing resources
→AI accelerators!