Convolutional Neural Networks for Computer Vision

Shiyao Wang
sy-wang14@mails.tsinghua.edu.cn
Dept. of Computer Science and Technology
Tsinghua University

Outline

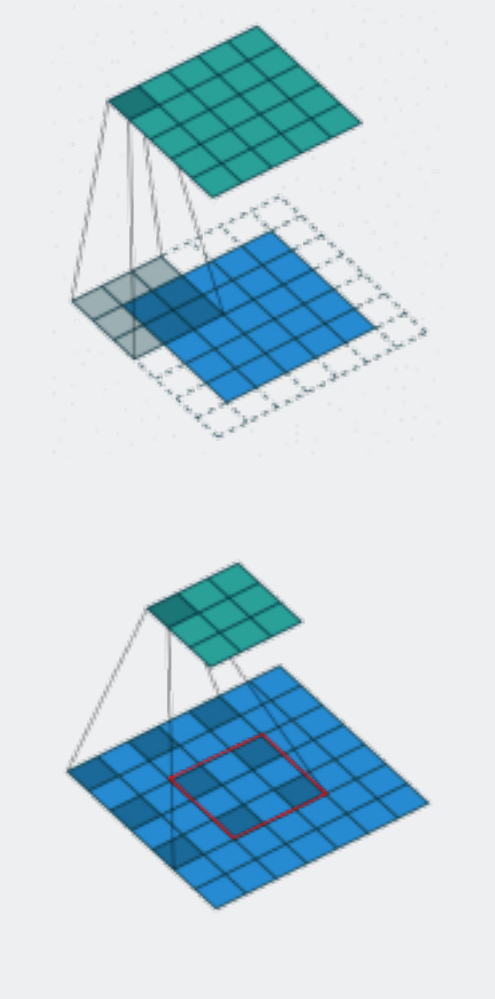
- CNN Techniques
  - Dilated/Deformable 2D Convolution, 3D Convolution
  - ROI Pooling
  - Batch/Layer/Instance/Group Normalization
  - Activation Functions, Dropout, Weight Initialization
- CNN Architectures
  - AlexNet/VGGNet
  - CoogLeNet (Inception v1-v4)
  - Short-cut (Resnet, DenseNet, ResNeXt)
  - AutoML
- Segmentation & Detection
  - Semantic/Instance Segmentation
  - (Video) Object Detection
- Others
  - Image Caption
  - Generative Adversarial Network
  - Style transfer

Outline

- CNN Techniques
  - Dilated/Deformable 2D Convolution, 3D Convolution
  - ROI Pooling
  - Batch/Layer/Instance/Group Normalization
  - Activation Functions, Dropout, Weight Initialization
- CNN Architectures
  - AlexNet/VGGNet
  - CoogLeNet (Inception v1-v4)
  - Short-cut (Resnet, DenseNet, ResNeXt)
  - AutoML
- Segmentation & Detection
  - Semantic/Instance Segmentation
  - (Video) Object Detection
- Others
  - Image Caption
  - Generative Adversarial Network
  - Style transfer

- CNN Techniques - Dilated/Deformable 2D Convolution, 3D Convolution

Dilated networks (Atrous) Convolution is a powerful tool to support exponential expansion of the receptive field as well as control the resolution of feature responses computed.



- **'familiar' Discrete convolution:**

$$(F * k)(\boldsymbol{p}) = \sum_{\boldsymbol{s}+\boldsymbol{t}=\boldsymbol{p}} F(\boldsymbol{s})k(\boldsymbol{t})$$

For each location $\boldsymbol{p}$ on the output, the familiar discrete convolution is applied over the input feature map by using filter k as above.

- **Dilated convolution:**

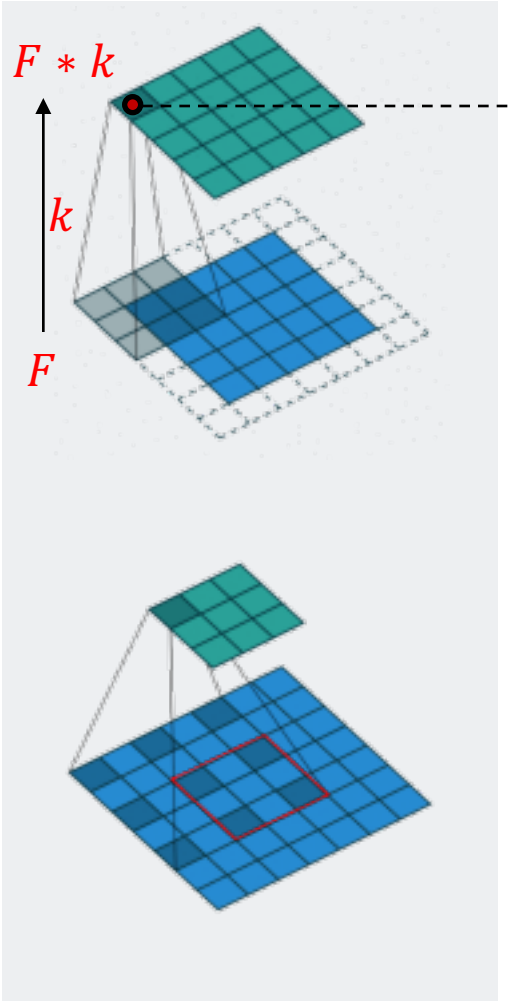
$$(F *_l k)(\boldsymbol{p}) = \sum_{\boldsymbol{s}+l\boldsymbol{t}=p} F(\boldsymbol{s})k(\boldsymbol{t})$$

where $*_l$ is an l-dilated convolution. The discrete convolution $*$ is simply the 1-dilated convolution.

gif source: vdumoulin/conv_arithmetic

- CNN Techniques - Dilated/Deformable 2D Convolution, 3D Convolution

Dilated networks (Atrous) Convolution is a powerful tool to support exponential expansion of the receptive field as well as control the resolution of feature responses computed.



- **'familiar' Discrete convolution:**

$$(F * k)(\boldsymbol{p}) = \sum_{\boldsymbol{s}+\boldsymbol{t}=\boldsymbol{p}} F(\boldsymbol{s})k(\boldsymbol{t})$$

For each location $\boldsymbol{p}$ on the output, the familiar discrete convolution is applied over the input feature map by using filter k as above.

- **Dilated convolution:**

$$(F *_l k)(\boldsymbol{p}) = \sum_{\boldsymbol{s}+l\boldsymbol{t}=\boldsymbol{p}} F(\boldsymbol{s})k(\boldsymbol{t})$$

where $*_l$ is an l-dilated convolution. The discrete convolution $*$ is simply the 1-dilated convolution.

If $\boldsymbol{p}$ is (0, 0), then

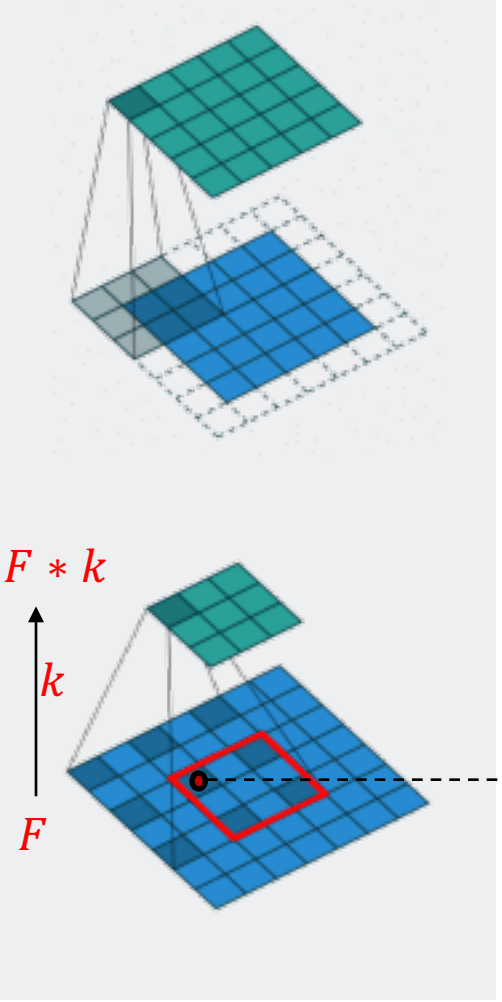
$\boldsymbol{t}$: (-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0), (1, 1)

$\boldsymbol{s}$: (1, 1), (1, 0), (1, -1), (0, 1), (0, 0), (0, -1), (-1,1), (-1, 0), (-1, -1)

gif source: vdumoulin/conv_arithmetic

- CNN Techniques - Dilated/Deformable 2D Convolution, 3D Convolution

Dilated networks (Atrous) Convolution is a powerful tool to support exponential expansion of the receptive field as well as control the resolution of feature responses computed.



- **Discrete convolution:**

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t)$$

For each location p on the output feature map, the discrete convolution is applied over the input feature map by using filter k as above.

If $\boldsymbol{p}$ is (0, 0) and l is 2, then

$\boldsymbol{t}$: (1, 1), (1, 0), (1, -1), (0, 1), (0, 0), (0, -1), (-1,1), (-1, 0), (-1, -1)

$\boldsymbol{s}$: (-2, -2), (-2, 0), (-2, 2), (0, -2), (0, 0), (0, 2), (2, -2), (2, 0), (2, 2)

- **Dilated convolution:**

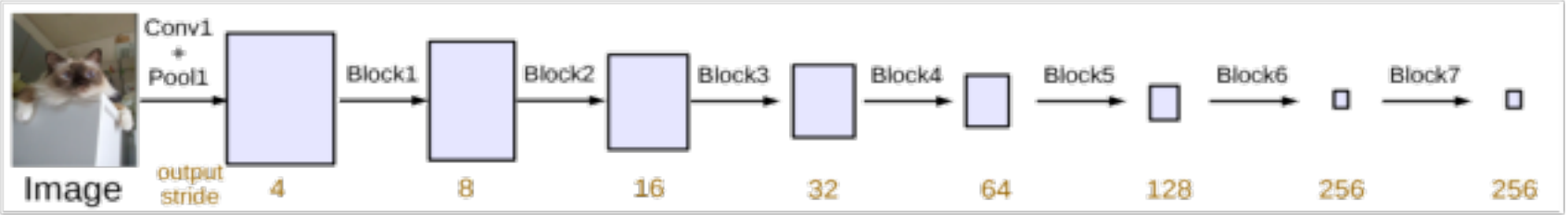
$$(F *_l k)(\boldsymbol{p}) = \sum_{\boldsymbol{s}+l\boldsymbol{t}=p} F(\boldsymbol{s})k(\boldsymbol{t})$$

where $*_l$ is an l-dilated convolution. The discrete convolution $*$ is simply the 1-dilated convolution.

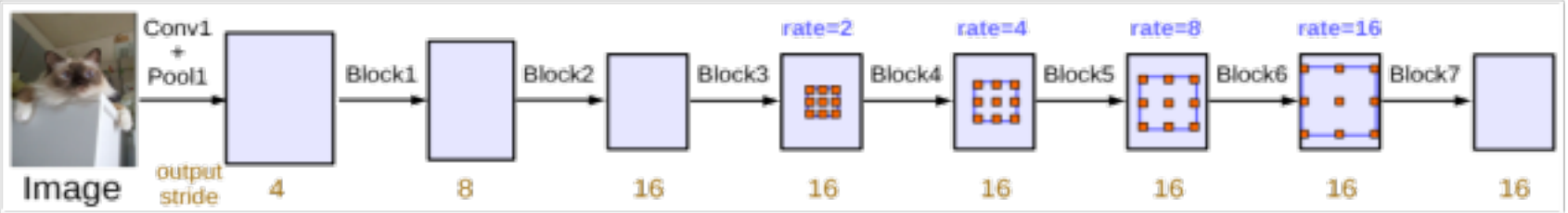
gif source: vdumoulin/conv_arithmetic

- CNN Techniques - Dilated/Deformable 2D Convolution, 3D Convolution

- Atrous convolution for semantic image segmentation



(a) Going deeper without atrous convolution.



Cascaded modules without and with dilated convolution. Dilated convolution with rate > 1 is applied after block3 when output stride = 16.

Advantages

- Support expansion of the receptive field without loss of resolution or coverage (vs. Pooling and Stride Convolutions which have similar concepts but both reduce the resolution).
- Have larger receptive field with same computation and memory costs.
- Generally improve performance!

Chen L C, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation[J]. arXiv preprint arXiv:1706.05587, 2017.

- CNN Techniques - Dilated/Deformable 2D Convolution, 3D Convolution

Deformable Convolution adds 2D offsets to the regular grid sampling locations in the standard convolution. It enables free form deformation of the sampling grid.

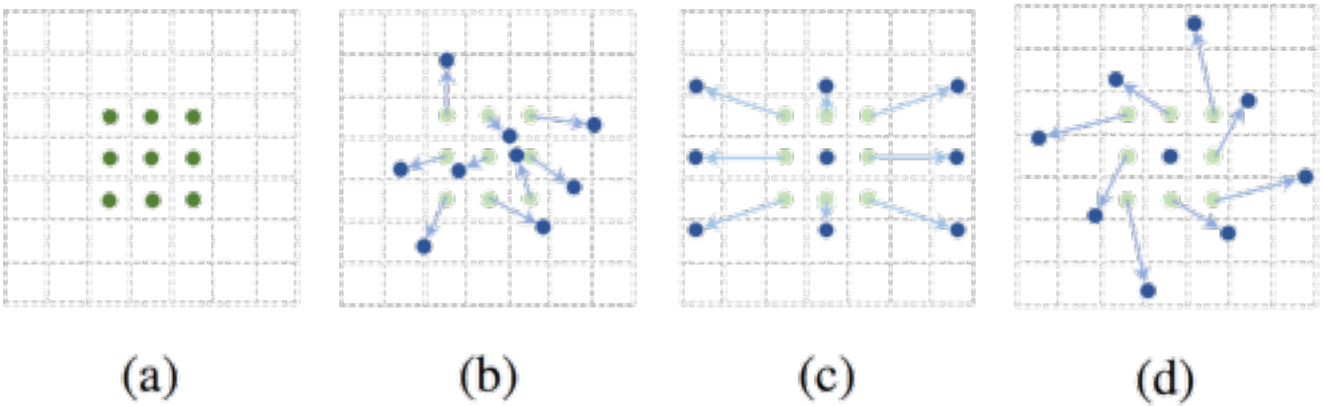
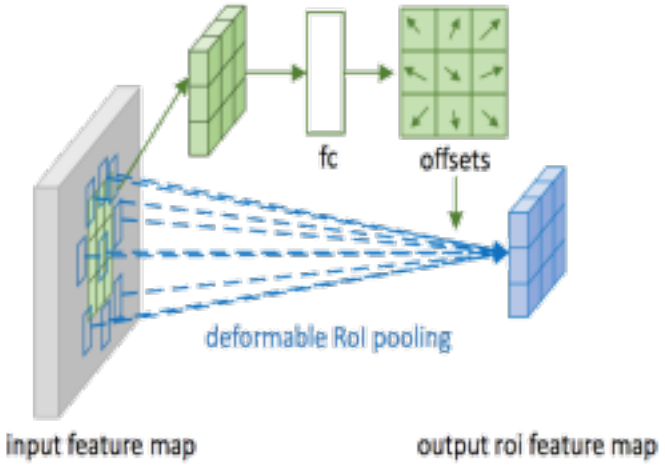


Illustration of the sampling locations in 3 × 3 standard and deformable convolutions.

- (a) Regular sampling grid (green points) of standard convolution.
- (b) Deformed sampling locations (dark blue points) with augmented offsets (light blue arrows) in deformable convolution.
- (c) and (d) are special cases of (b), showing that the deformable convolution generalizes various transformations for scale, (anisotropic) aspect ratio and rotation.



Dai, Jifeng et al. "Deformable Convolutional Networks." 2017 IEEE International Conference on Computer Vision (ICCV) (2017): 764-773.

- CNN Techniques - Dilated/Deformable 2D Convolution, 3D Convolution

- Deformable Convolutional Networks

Each image triplet shows the sampling locations (9^3 = 729 red points in each image) in three levels of 3 × 3 deformable filters for three activation units (green points) on the background (left), a small object (middle), and a large object (right), respectively.

The grid $\mathcal{R}$ defines the receptive field size and dilation:

$$\mathcal{R} = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\}$$

defines a 3 × 3 kernel with dilation 1.

For each location $\boldsymbol{p}$ on the output feature map y,

$$y(\boldsymbol{p}) = \sum_{\boldsymbol{p_n}\in\mathcal{R}} k(\boldsymbol{p_n}) \cdot x(\boldsymbol{p}+\boldsymbol{p_n})$$

where enumerates the locations in $\mathcal{R}$.

In deformable convolution, the regular grid $\mathcal{R}$ is augmented with offsets $\Delta\boldsymbol{p_n}$

$$y(\boldsymbol{p}) = \sum_{\boldsymbol{p_n}\in\mathcal{R}} k(\boldsymbol{p_n}) \cdot x(\boldsymbol{p}+\boldsymbol{p_n}+\Delta\boldsymbol{p_n})$$

As the offset $\Delta\boldsymbol{p_n}$ is typically fractional, it is implemented via bilinear interpolation as

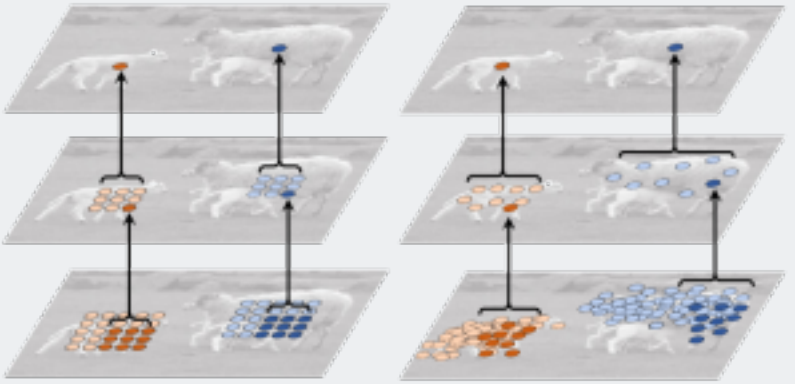
$$x(\boldsymbol{p}) = \sum_{\boldsymbol{q}} G(\boldsymbol{q},\boldsymbol{p}) \cdot x(\boldsymbol{q})$$

Dai, Jifeng et al. "Deformable Convolutional Networks." 2017 IEEE International Conference on Computer Vision (ICCV) (2017): 764-773.

- CNN Techniques - Dilated/Deformable 2D Convolution, 3D Convolution

- Deformable Convolutional Networks

Each image triplet shows the sampling locations (9^3 = 729 red points in each image) in three levels of 3 × 3 deformable filters for three activation units (green points) on the background (left), a small object (middle), and a large object (right), respectively.



In deformable convolution, the regular grid $\mathcal{R}$ is augmented with offsets $\Delta \boldsymbol{p_n}$

$$y(\boldsymbol{p}) = \sum_{\boldsymbol{p_n} \in \mathcal{R}} k(\boldsymbol{p_n}) \cdot x(\boldsymbol{p} + \boldsymbol{p_n} + \Delta \boldsymbol{p_n})$$

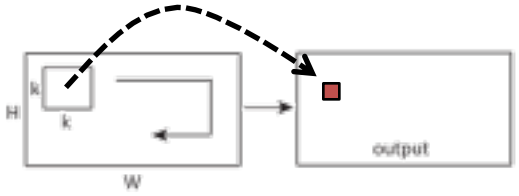
As the offset $\Delta \boldsymbol{p_n}$ is typically fractional, it is implemented via bilinear interpolation as

$$x(\boldsymbol{p}) = \sum_{\boldsymbol{q}} G(\boldsymbol{q}, \boldsymbol{p}) \cdot x(\boldsymbol{q})$$

Dai, Jifeng et al. "Deformable Convolutional Networks." 2017 IEEE International Conference on Computer Vision (ICCV) (2017): 764-773.

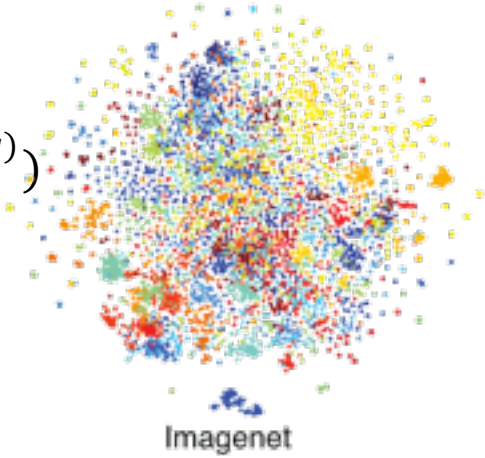
- CNN Techniques - Dilated/Deformable 2D Convolution, 3D Convolution

3D Convolution is a a simple, yet effective approach for spatiotemporal feature learning method.

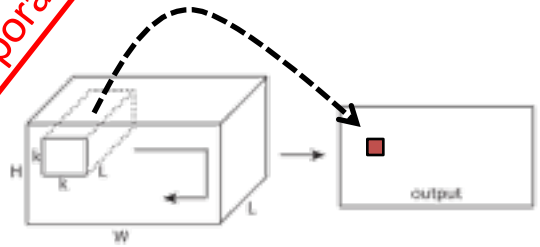


2D convolution

$$v^{xy} = f\left(\sum_{c}\sum_{p=-\Delta p}^{\Delta p}\sum_{q=-\Delta q}^{\Delta q} w_c^{pq}\, v_c^{(x+p)(y+q)}\right)$$

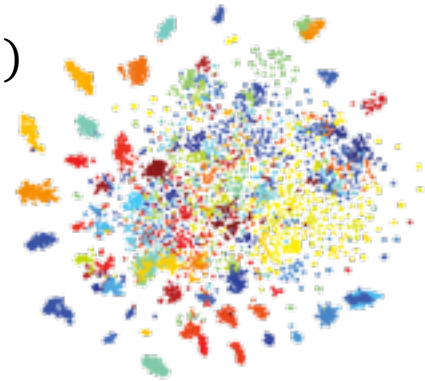


temporal

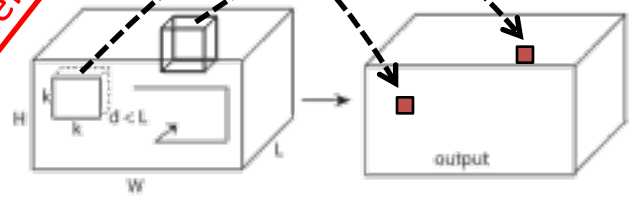


2D convolution on mutiple frames

$$v^{xy} = f\left(\sum_{c}^{C\times L}\sum_{p=-\Delta p}^{\Delta p}\sum_{q=-\Delta q}^{\Delta q} w_c^{pq}\, v_c^{(x+p)(y+q)}\right)$$



temporal



3D convolution

$$v_t^{xy} = f\left(\sum_{c}\sum_{t}^{T}\sum_{p=-\Delta p}^{\Delta p}\sum_{q=-\Delta q}^{\Delta q} w_{ct}^{pq}\, v_{ct}^{(x+p)(y+q)}\right)$$

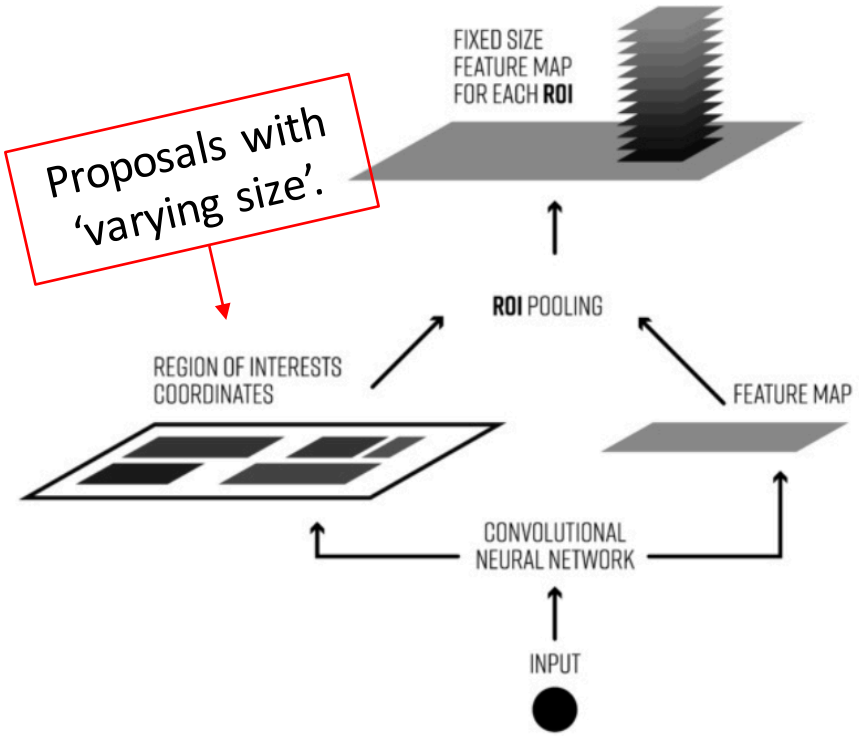
C3D

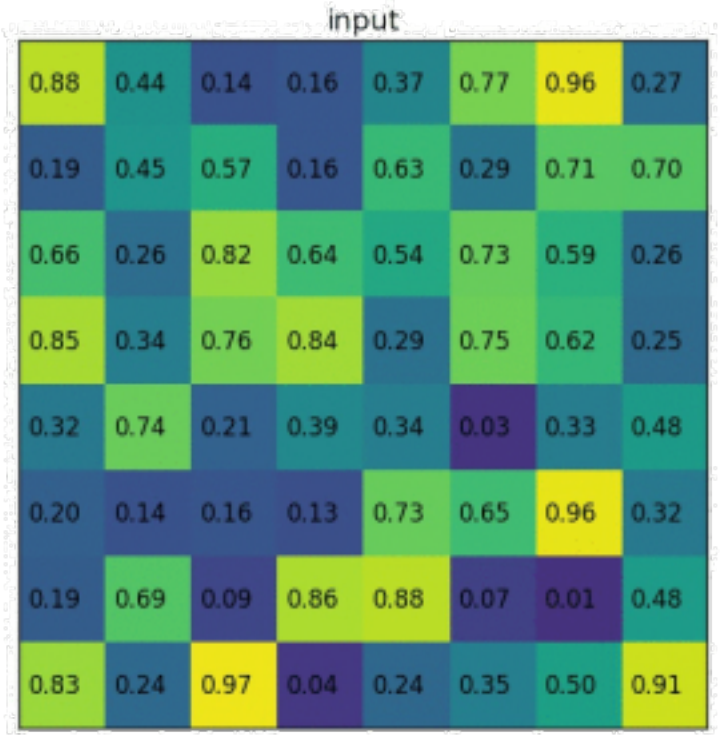
where *T* is the temporal depth (e.g., *T=3*)

Tran D, Bourdev L, Fergus R, et al. Learning spatiotemporal features with 3d convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2015: 4489-4497.

- CNN Techniques – ROI Pooling

Region of interest pooling (also known as RoI pooling) adopts max pooling to convert the features inside any valid region of interest into a small feature map with a fixed spatial extent. **It is an operation widely used in object detection tasks.**





① Dividing the region proposal into equal-sized sections
② Finding the largest value in each section
③ Copying these max values to the output buffer

Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

- CNN Techniques - Batch/Layer/Instance/Group Normalization
  - Reducing **Internal Covariate Shift**

- ICS: the change in the distribution of network activations due to the change in network parameters during training.
  - Intuition: normalize minibatch activations - **Batch Norm**

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

Forward

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

$$\frac{\partial \ell}{\partial \widehat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \gamma$$

$$\frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2}(\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2}$$

$$\frac{\partial \ell}{\partial \mu_{\mathcal{B}}} = \left(\sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}\right) + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_{\mathcal{B}})}{m}$$

$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}$$

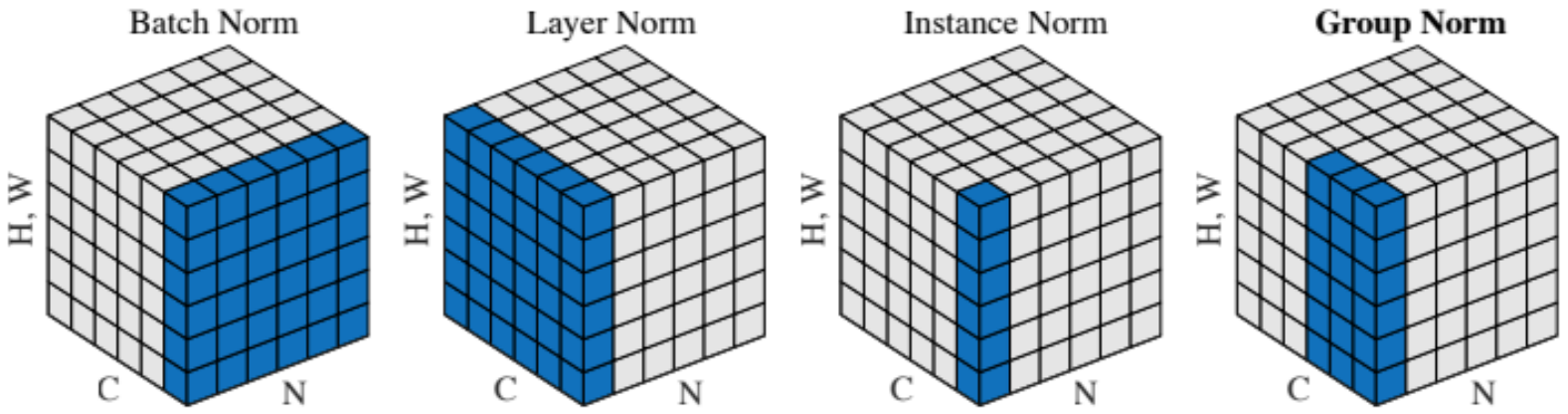
$$\frac{\partial \ell}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i} \cdot \widehat{x}_i$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i}$$

backward

Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[C]//ICML, 2015: 448-456

- CNN Techniques - Batch/Layer/Instance/Group Normalization



Batch Norm

$$\mathcal{S}_i = \{k \mid k_C = i_C\}$$

Instance Norm

$$\mathcal{S}_i = \{k \mid k_N = i_N, k_C = i_C\}$$

Layer Norm

$$\mathcal{S}_i = \{k \mid k_N = i_N\}$$

Group Norm

$$\mathcal{S}_i = \{k \mid k_N = i_N, \lfloor \frac{k_C}{C/G} \rfloor = \lfloor \frac{i_C}{C/G} \rfloor\}$$

Wu Y, He K. Group normalization[J]. arXiv preprint arXiv:1803.08494, 2018.

- CNN Techniques - Activation Functions, Dropout, Weight Initialization

Activation Function is used to map the output of neural network in between 0 to 1 or -1 to 1 etc. (depending upon the function).

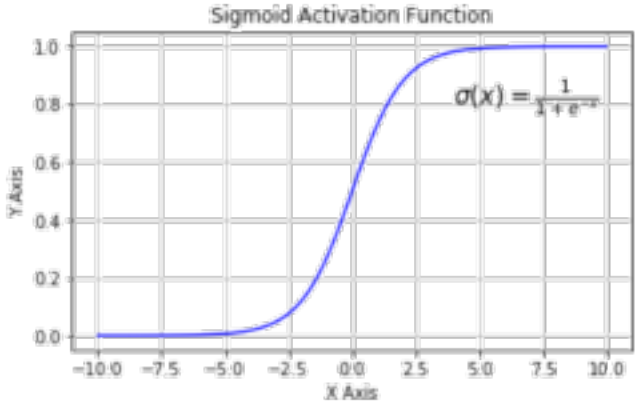


Figure: Sigmoid Activation Function

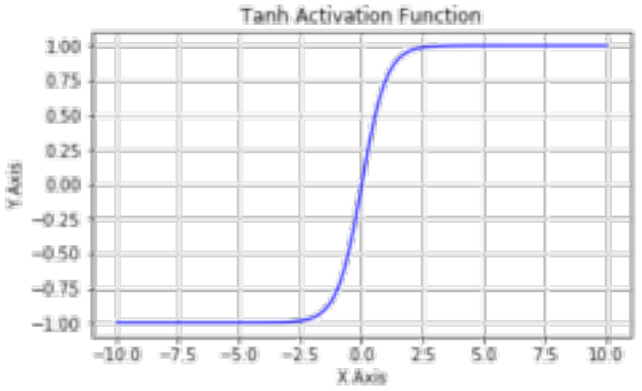


Figure: Tanh Activation Function

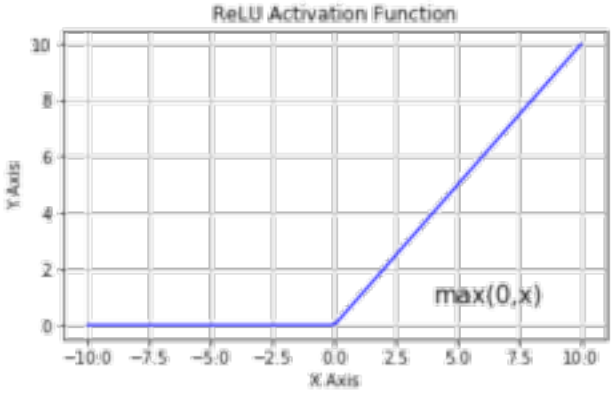


Figure: ReLU Activation Function

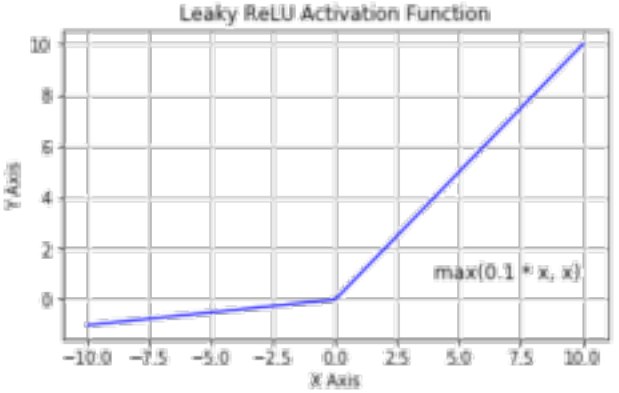


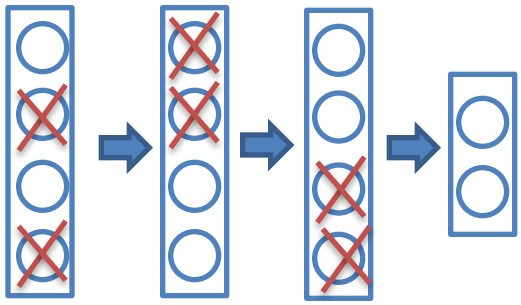
Figure : Leaky ReLU activation function

ReLU is less computationally expensive than *sigmoid* or *tanh* because it involves simpler mathematical operations.

- CNN Techniques - Activation Functions, Dropout, Weight Initialization

Dropout:

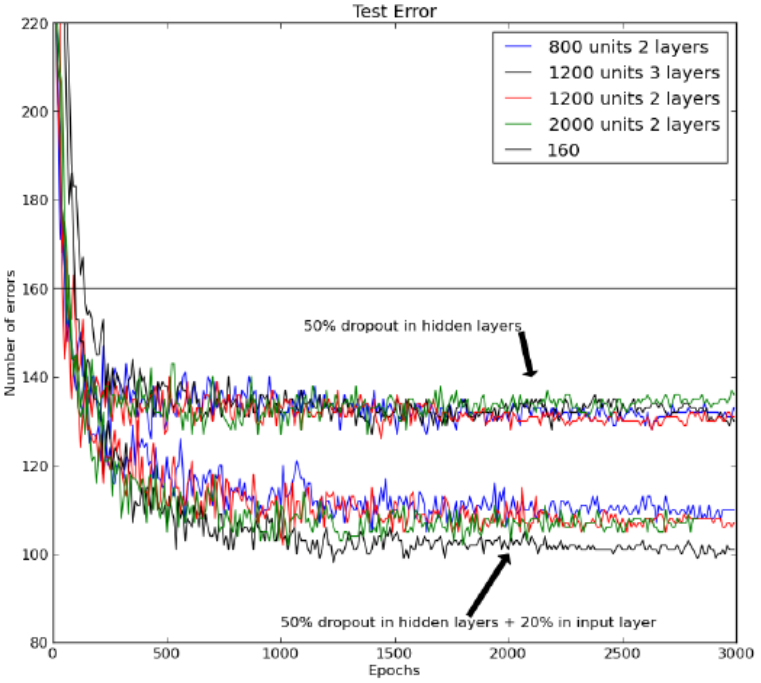
On each presentation of each training case, each hidden unit is randomly omitted (**zero its output**) from the network with a probability (e.g., 0.5)



Advantage:

- A hidden unit cannot rely on other hidden units being present, therefore we prevent complex co-adaptations of the neurons on the training data
- It trains a huge number of different networks in a reasonable time, then average their predictions (ensemble)

Results on MNIST



The lower set of lines also use 20% dropout for the input layer. The best previously published result using backpropagation without pre-training or weight-sharing or enhancements of the training set is shown as a horizontal line.

- CNN Techniques - Activation Functions, Dropout, Weight Initialization

- Weights Initialization

- Want a better start point for the network
- Intuition
  - want the variance of the signal going forward in the network to remain the same
- Xavier initialization [1]

$$\forall i, \quad n_i Var[W^i] = 1$$

$$\forall i, \quad n_{i+1} Var[W^i] = 1$$

$$\forall i, \quad Var[W^i] = \frac{2}{n_i + n_{i+1}}$$

X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks,"

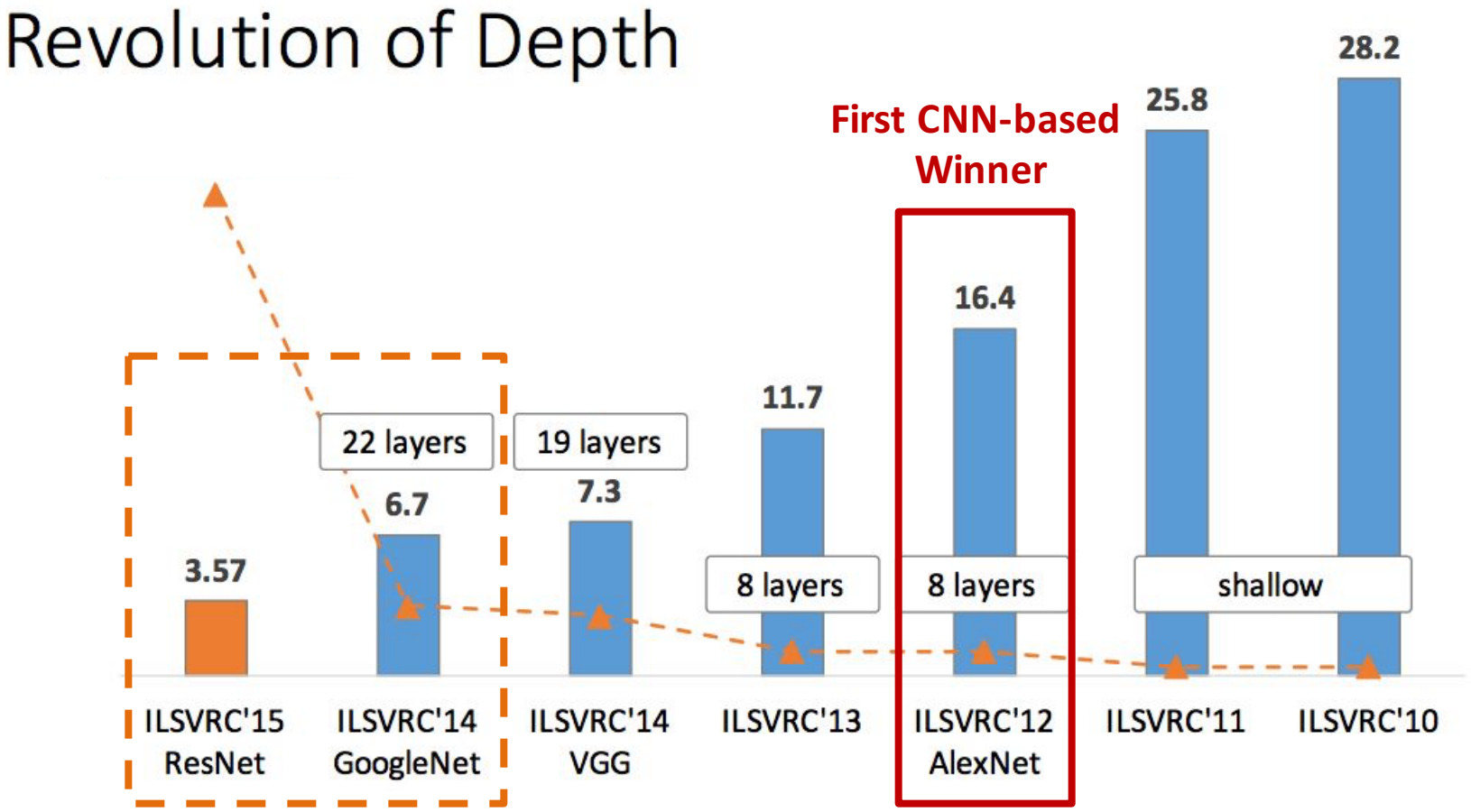
Outline

- CNN Techniques
  - Dilated/Deformable 2D Convolution, 3D Convolution
  - ROI Pooling
  - Batch/Layer/Instance/Group Normalization
  - Activation Functions, Dropout, Weight Initialization
- CNN Architectures
  - AlexNet/VGGNet
  - CoogLeNet (Inception v1-v4)
  - Short-cut (Resnet, DenseNet, ResNeXt)
  - AutoML
- Segmentation & Detection
  - Semantic/Instance Segmentation
  - (Video) Object Detection
- Others
  - Image Caption
  - Generative Adversarial Network
  - Style transfer

- CNN Architectures - AlexNet/VGGNet

- ImageNet Large Scale Visual Recognition Challenge

  - 1000 categories
  - 1.2 M images for training
  - 50K for validation 100K for testing

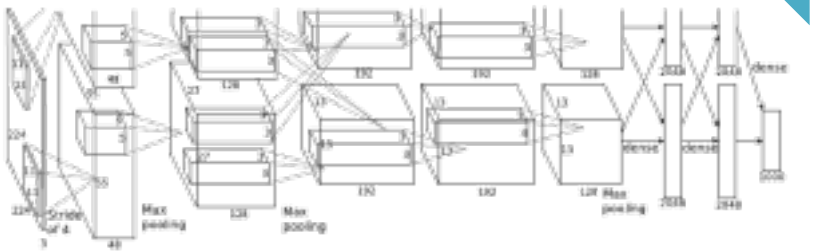


- CNN Architectures - AlexNet/VGGNet

- ImageNet Large Scale Visual Recognition Challenge

- 1000 categories
- 1.2 M images for training
- 50K for validation 100K for testing

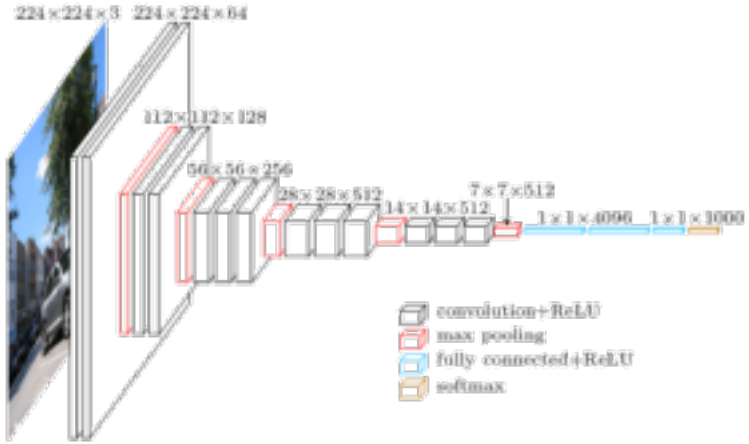
AlexNet [1]



Details/Retrospectives:

- First use of ReLU
- Use Norm layers
- Heavy data augmentation
- Dropout 0.5
- Batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when via accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

VGGNet [2]



Details:

- ILSVRC'14 *2*nd in classification
- *1*st in localization
- 3*3 filters are extensively used
- No Local Response Normalization (LRN)
- Use ensembles for best results

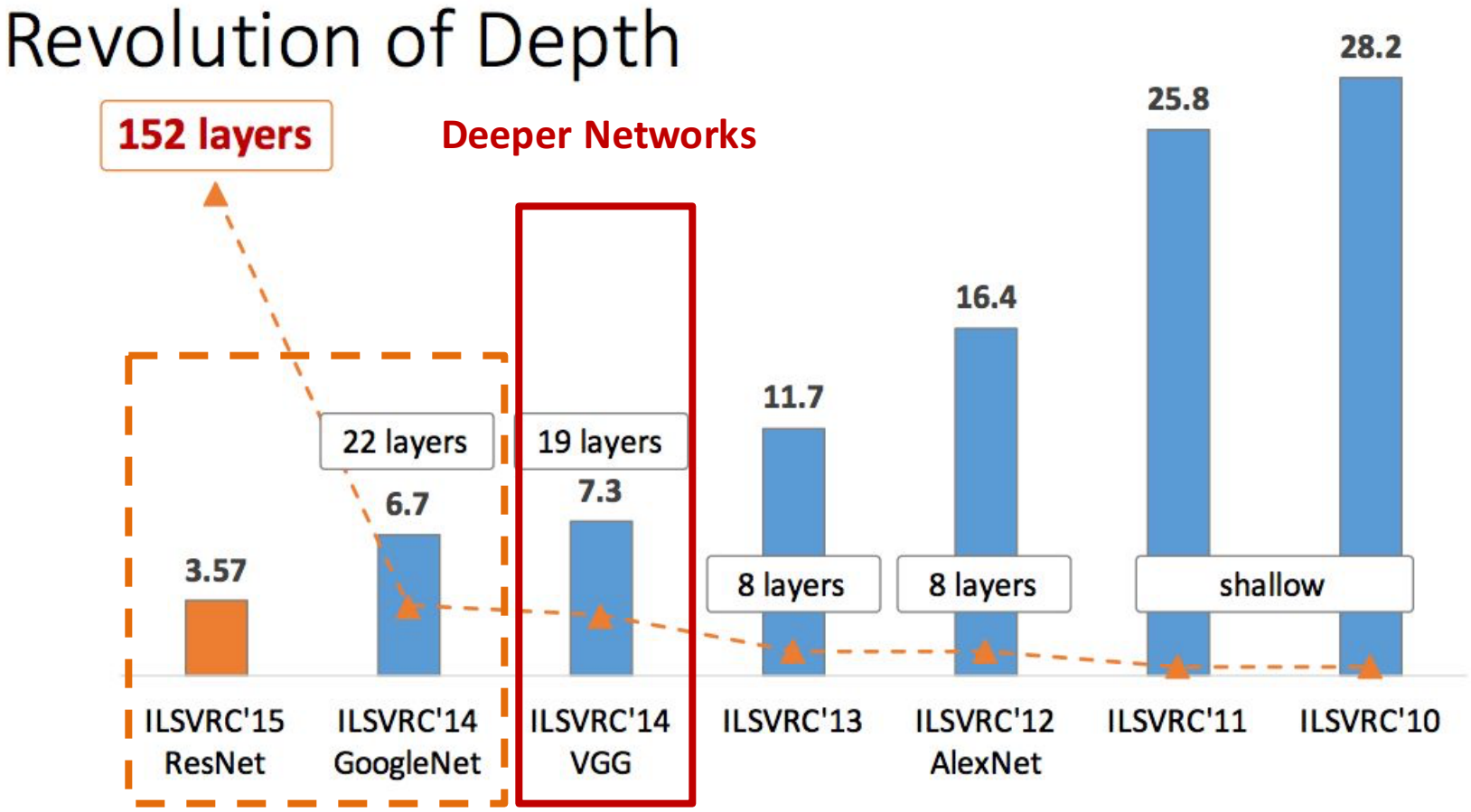
[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[2] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

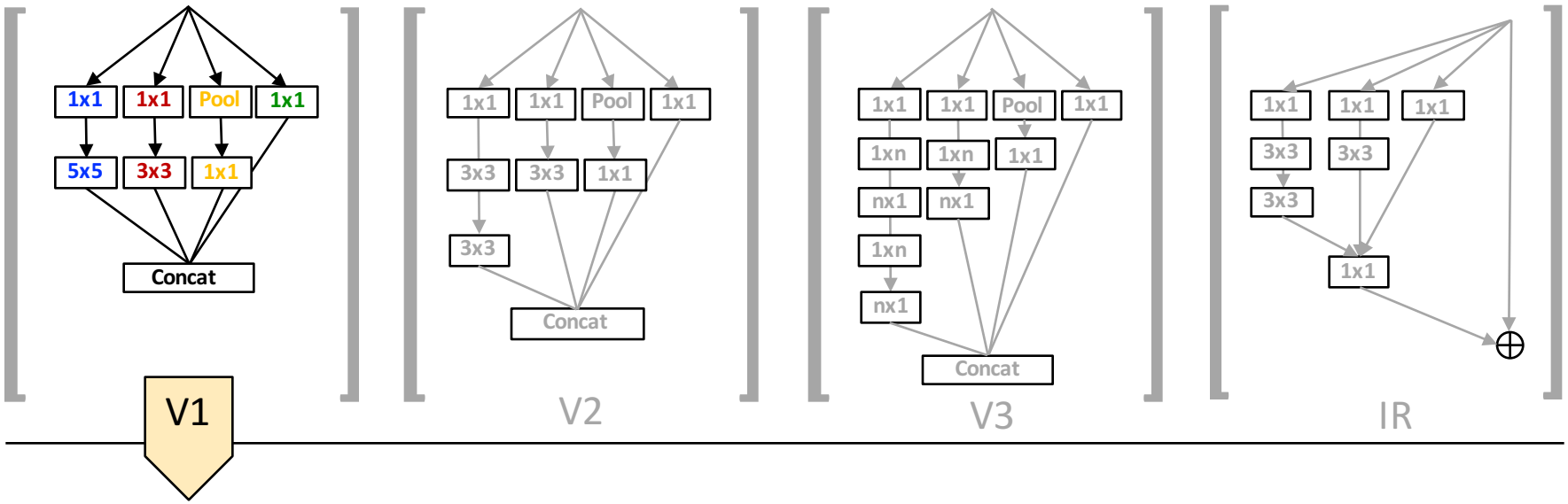
- CNN Architectures - AlexNet/VGGNet

- ImageNet Large Scale Visual Recognition Challenge

  - 1000 categories
  - 1.2 M images for training
  - 50K for validation 100K for testing

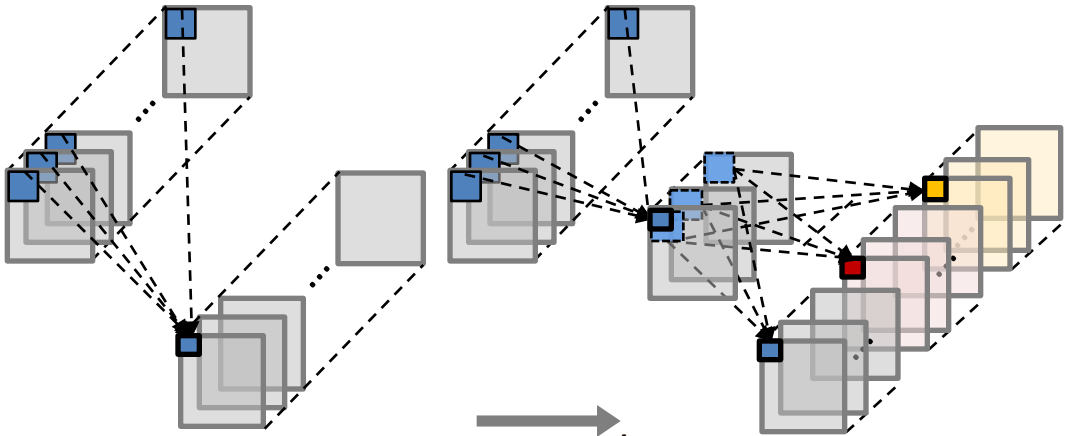


- CNN Architectures – GoogLeNet (Inception V1 - V4)



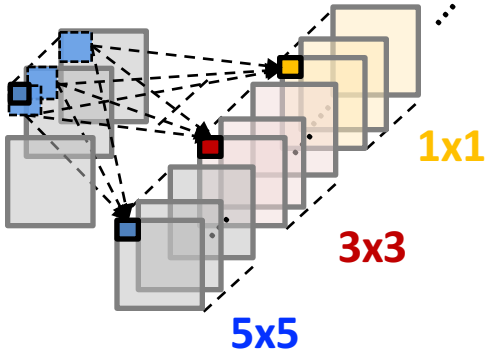
Increasing the depth and width of the network while keeping the computational budget constant.

Hebbian Principle



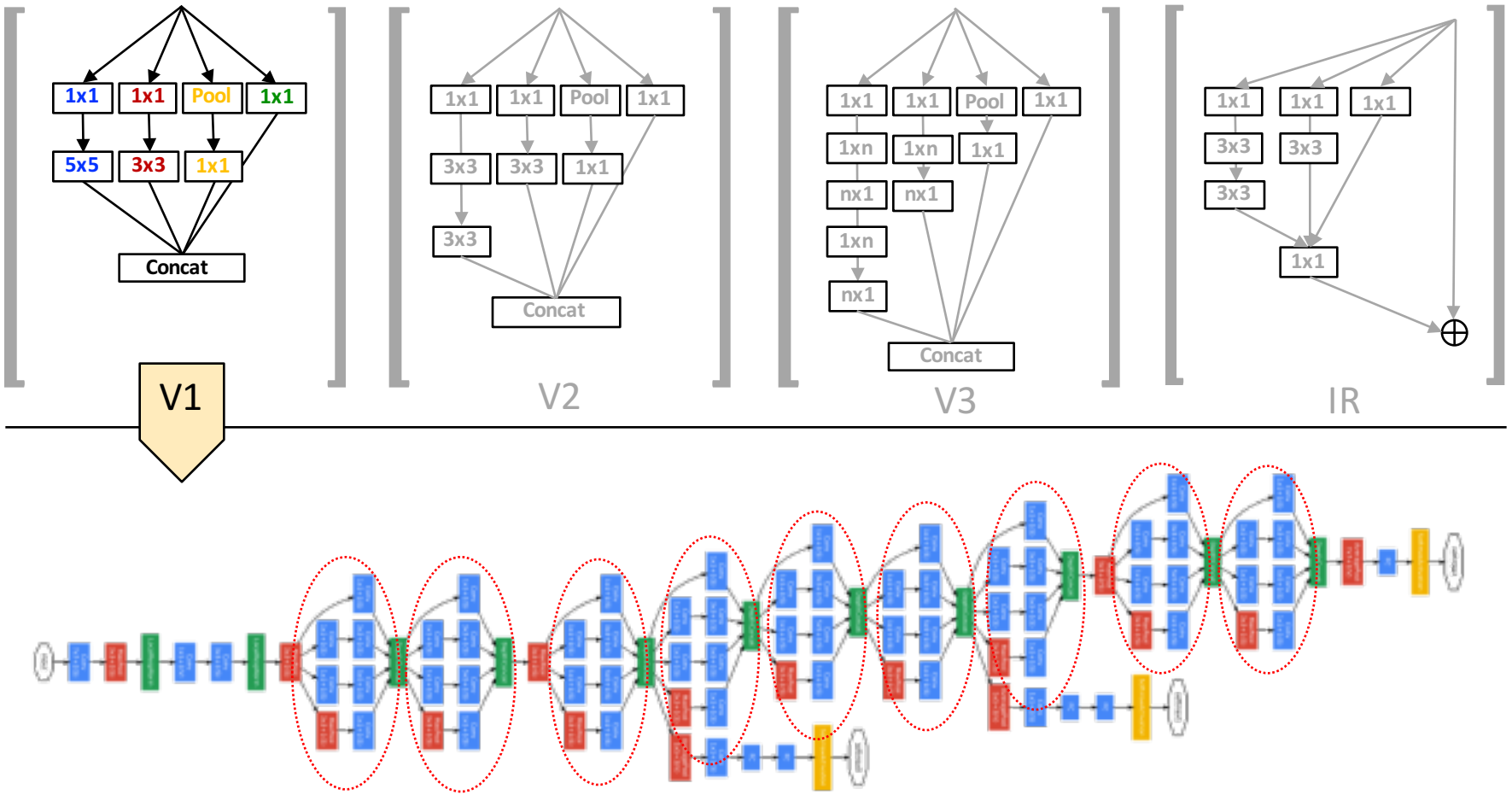
$n \times m > n \times p + p \times m, \quad when \quad p << n$

Multi-scale Processing



Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

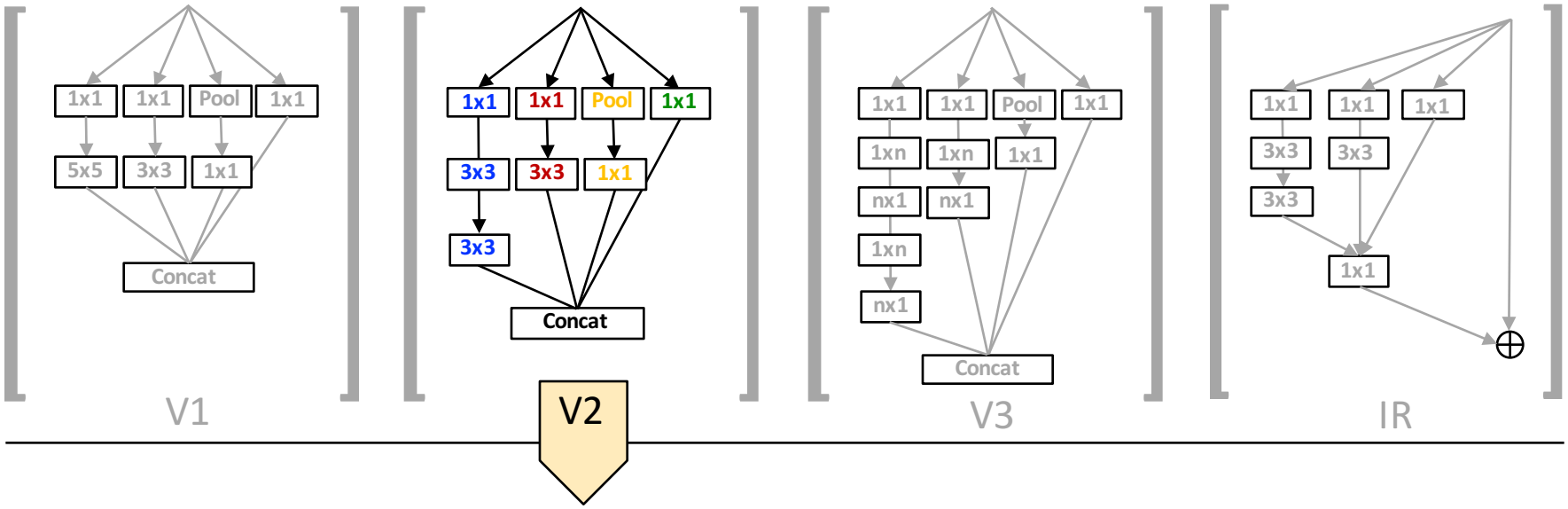
- CNN Architectures – GoogLeNet (Inception V1 - V4)



- 9 inception modules;
- Two auxiliary classifiers connected to intermediate layers are used to increase the gradient signal for BP algorithm.

Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

- CNN Architectures – GoogLeNet (Inception V1 - V4)

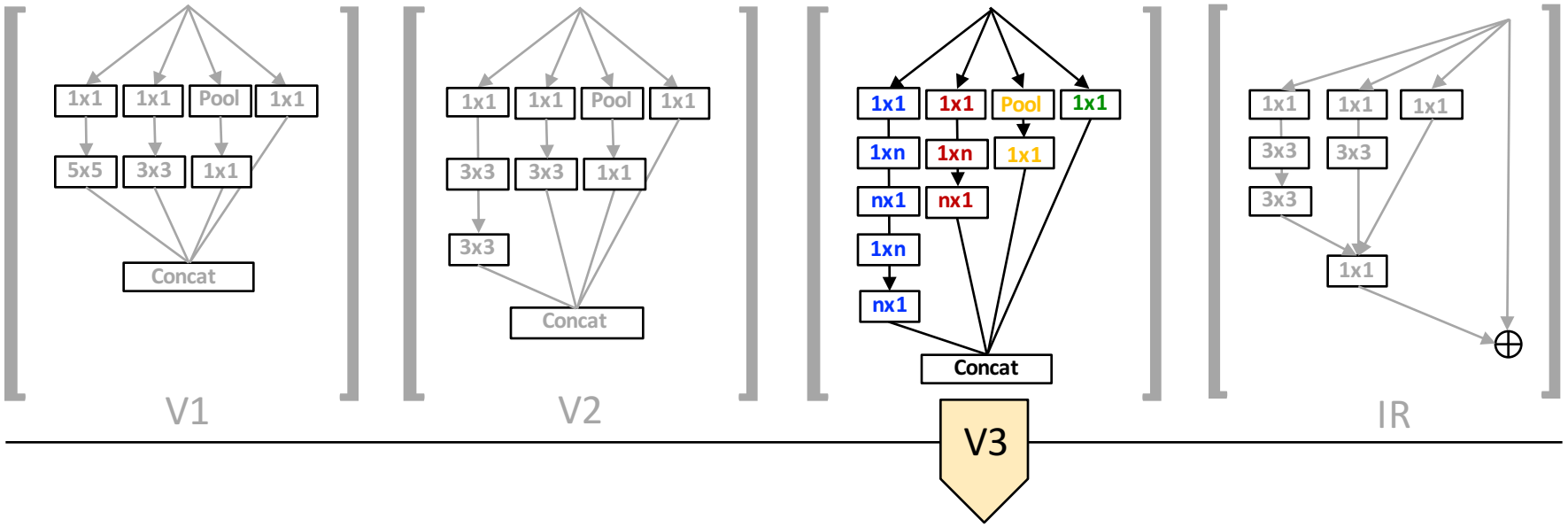


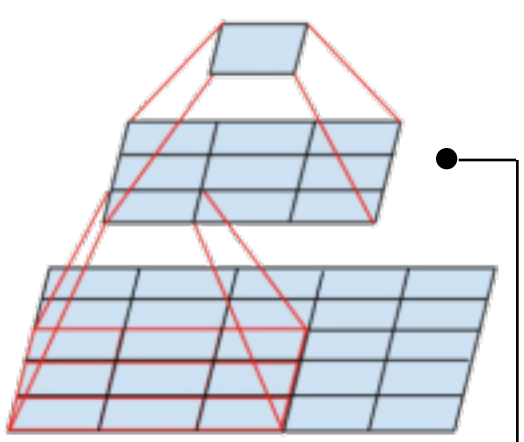
Model	Top-1 error	Top-5 error
GoogLeNet ensemble	-	6.67%
Deep Image low-res	-	7.96%
Deep Image high-res	24.88	7.42%
Deep Image ensemble	-	5.98%
BN-Inception single crop	25.2%	7.82%
BN-Inception multicrop	21.99%	5.82%
BN-Inception ensemble	20.1%	**4.9%***

- BN-inception;
- The 5×5 is replaced by two consecutive 3×3 convolutional layers;
- stride-2 convolution/pooling layers are employed before the concatenation in the modules 3c, 4e;

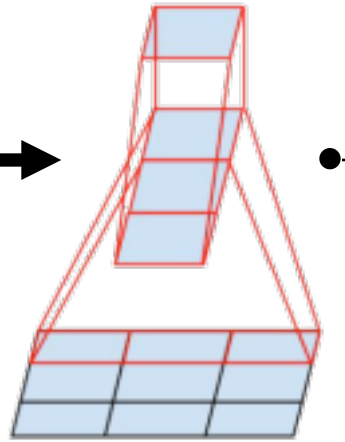
Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[C]//ICML, 2015: 448-456

- CNN Architectures – GoogLeNet (Inception V1 - V4)





5x5 → stack 3x3s

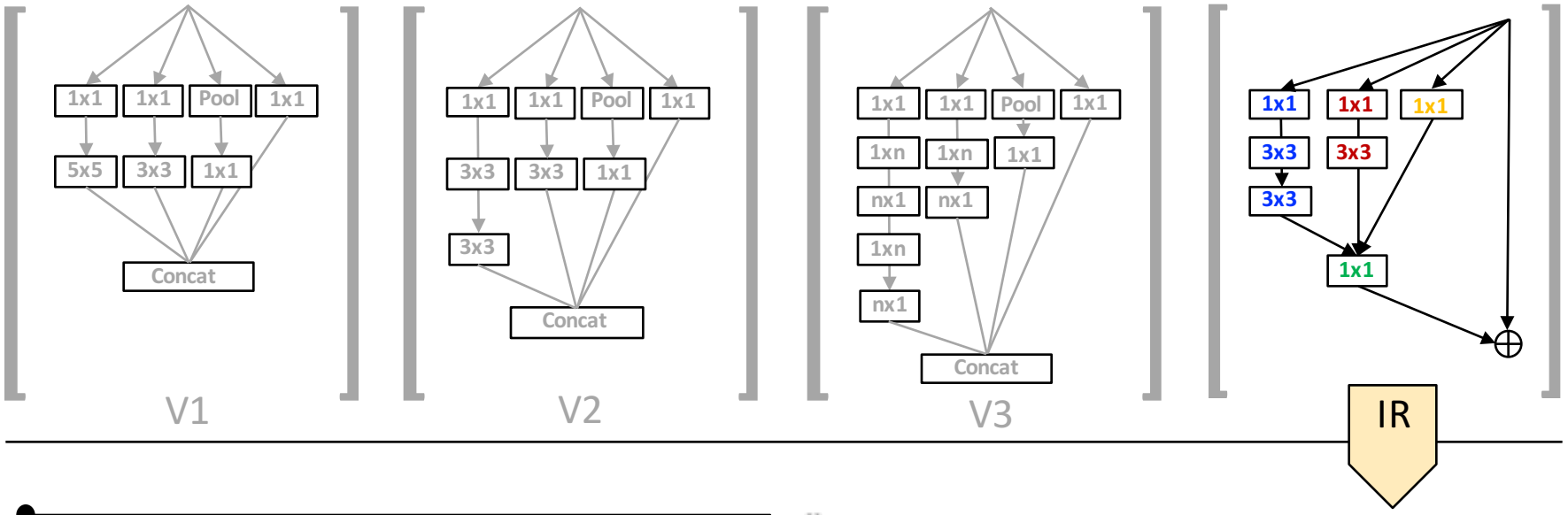


3x3 → 1x3 + 3x1

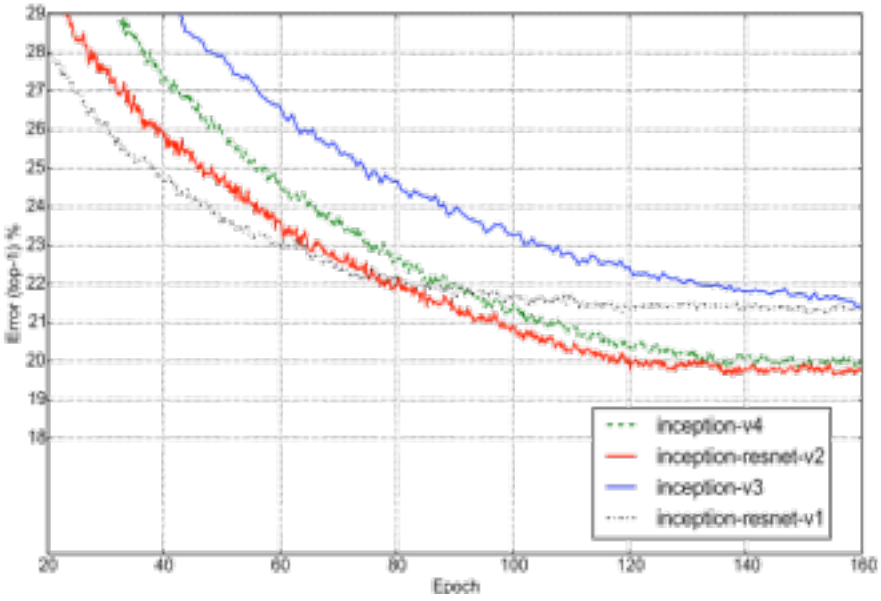
- RMSProp Optimizer.
- Factorized 7x7 convolutions.
- BatchNorm in the Auxiliary Classifiers.
- Label Smoothing (A type of regularizing component added to the loss formula that prevents the network from becoming too confident about a class.

Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.

- CNN Architectures – GoogLeNet (Inception V1 - V4)

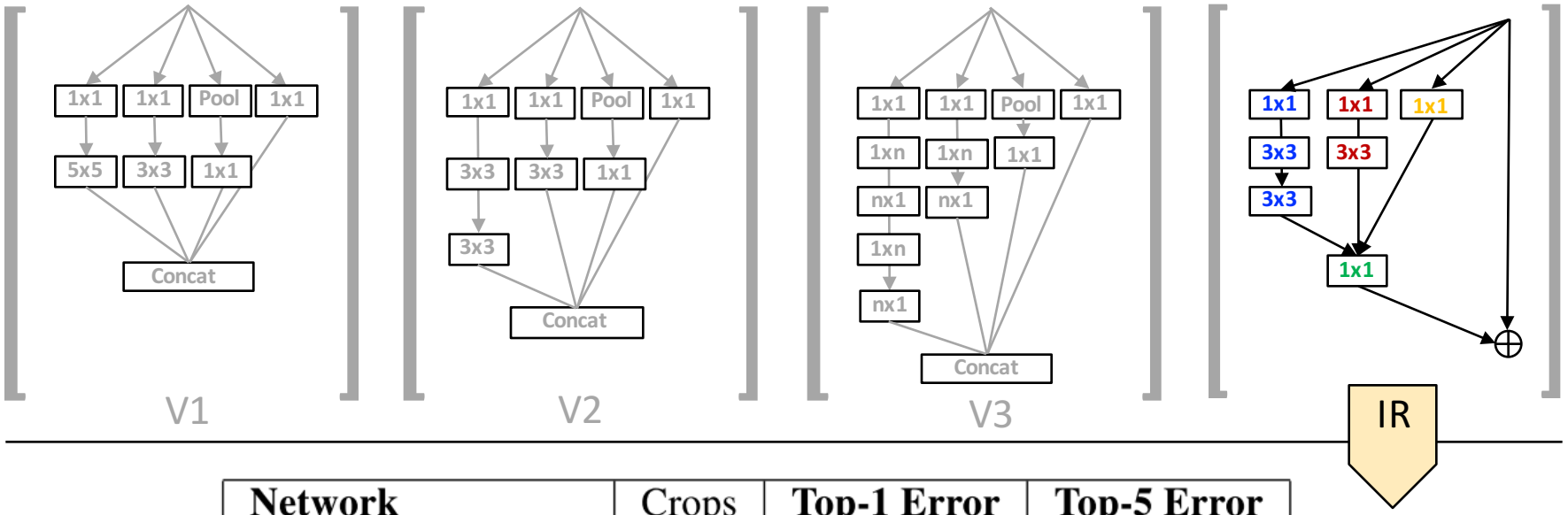


For inception V4,
the modules are mode more **uniform** than inception V3;
Deeper and wider;
For IR (Inception ResNet),
residual connections are introduced that add the output of the convolution operation of the inception module, to the input.



Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]//AAAI. 2017, 4: 12.

- CNN Architectures – GoogLeNet (Inception V1 - V4)



Network	Crops	Top-1 Error	Top-5 Error
ResNet-151 [5]	dense	19.4%	4.5%
Inception-v3 [15]	144	18.9%	4.3%
Inception-ResNet-v1	144	18.8%	4.3%
Inception-v4	144	17.7%	3.8%
Inception-ResNet-v2	144	17.8%	3.7%

144 crops evaluations - single model experimental results. Reported on the all 50000 images of the validation set of ILSVRC 2012.Network

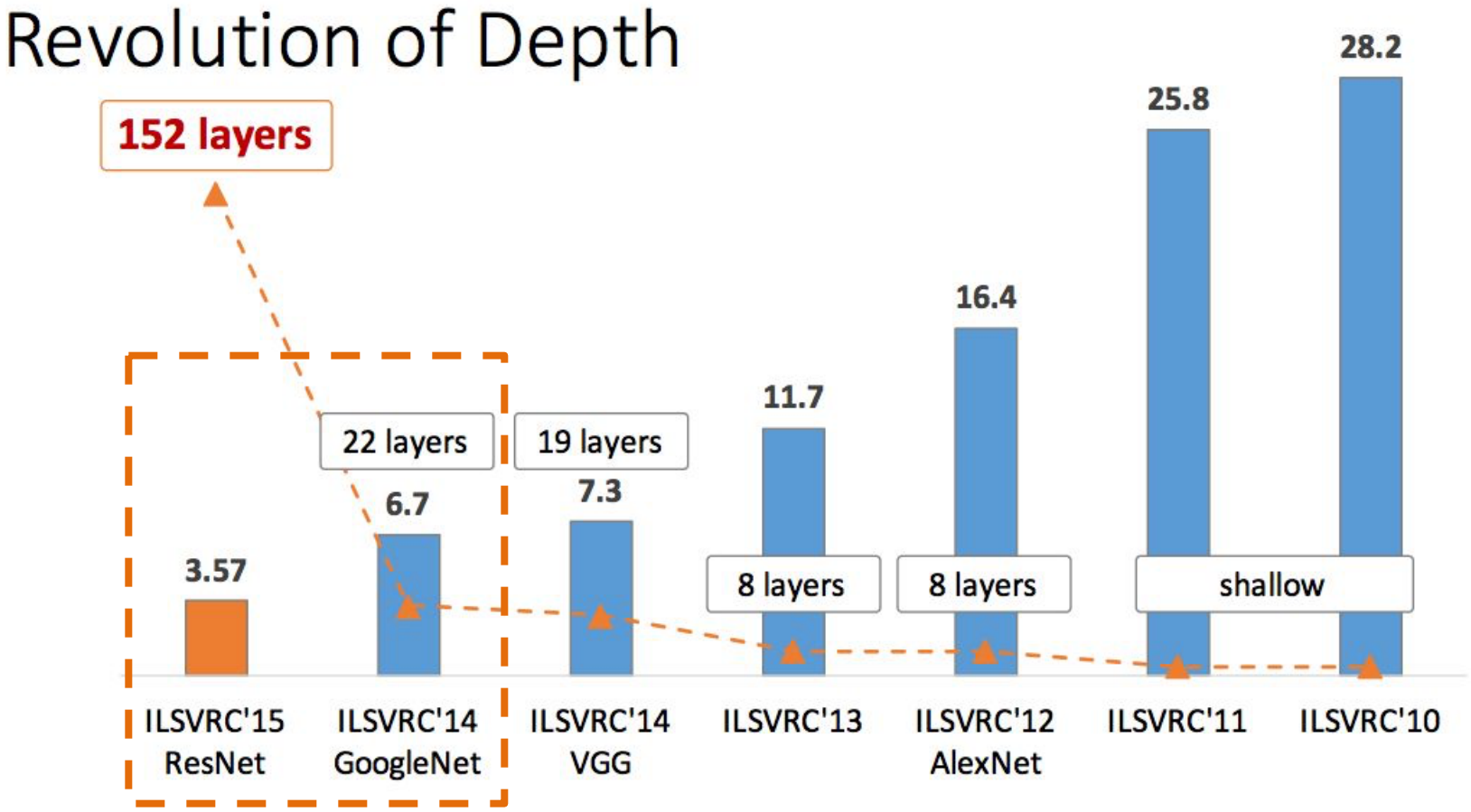
- It was found that Inception-ResNet models were able to achieve higher accuracies at a lower epoch.

Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]//AAAI. 2017, 4: 12.

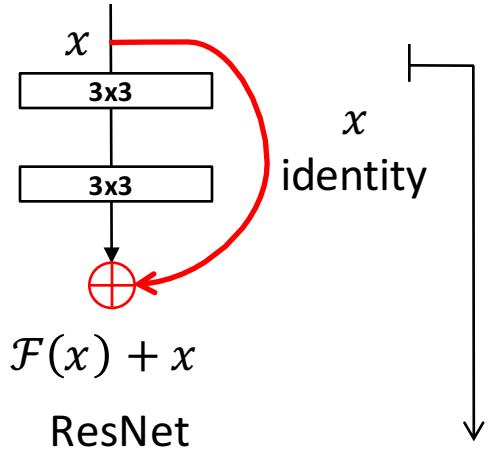
- CNN Architectures - AlexNet/VGGNet

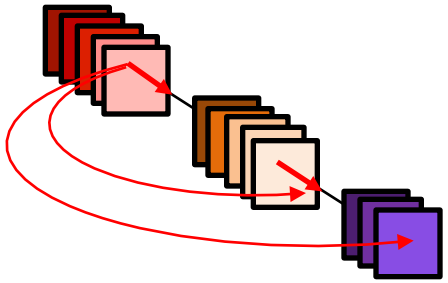
- ImageNet Large Scale Visual Recognition Challenge

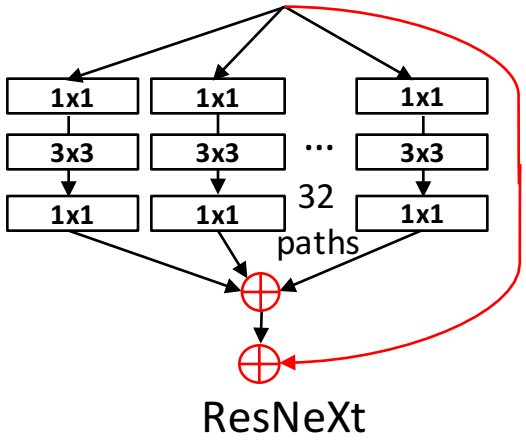
  - 1000 categories
  - 1.2 M images for training
  - 50K for validation 100K for testing



- CNN Architectures - Short-cut (Resnet, DenseNet, ResNeXt…)

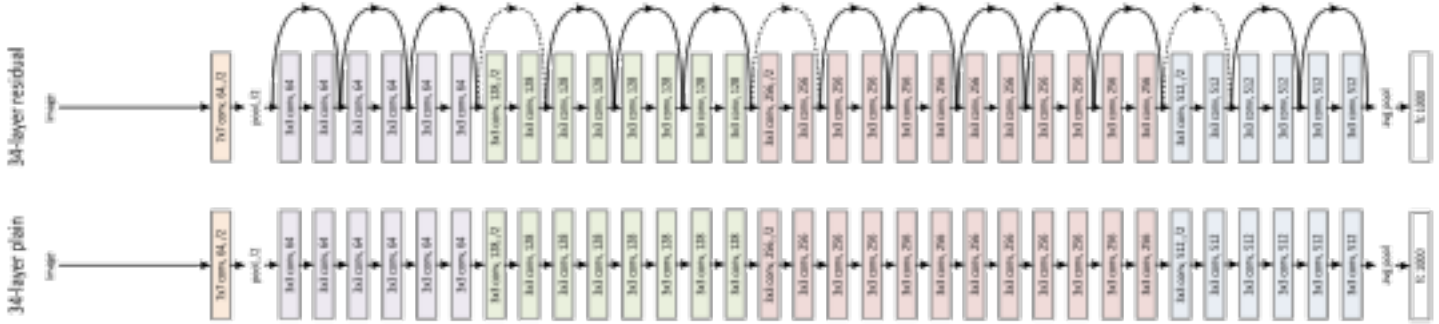






ResNet

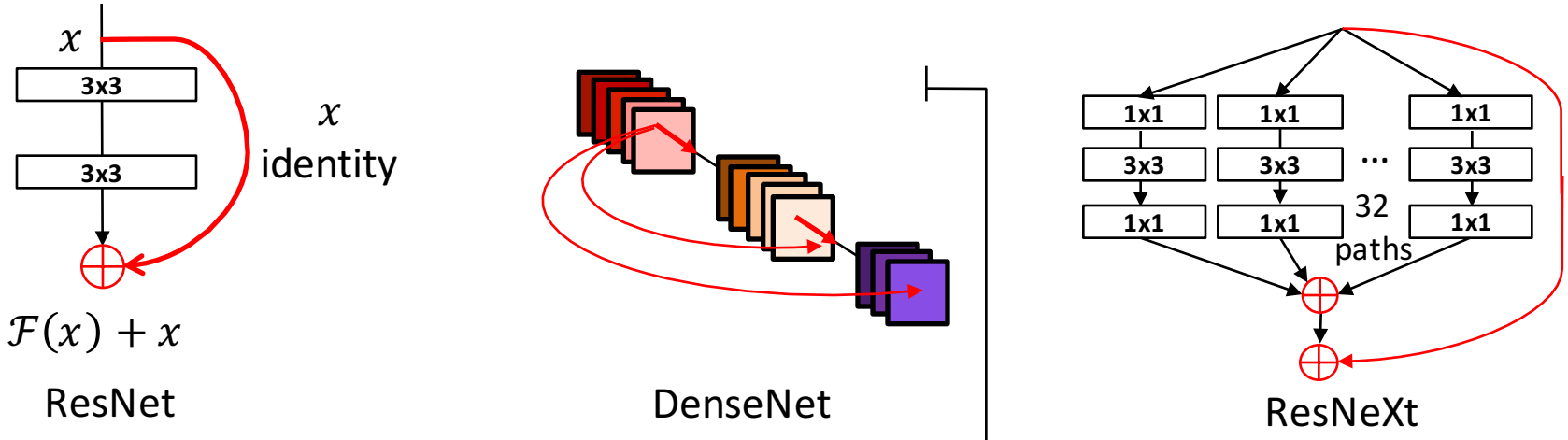
The residual learning framework eases the training of the deeper networks, and enables them to be substantially deeper — leading to improved performance.



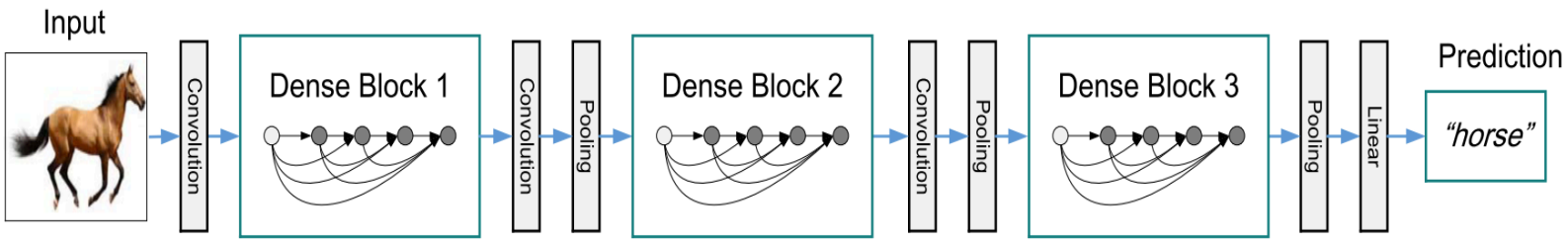
- Won the 1st place on the ILSVRC 2015 classification, ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation;
- The ImageNet dataset, up to 152 layers—8× deeper than VGG nets;
- On CIFAR-10 with 100 and 1000 layers;

He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

- CNN Architectures - Short-cut (Resnet, DenseNet, ResNeXt…)



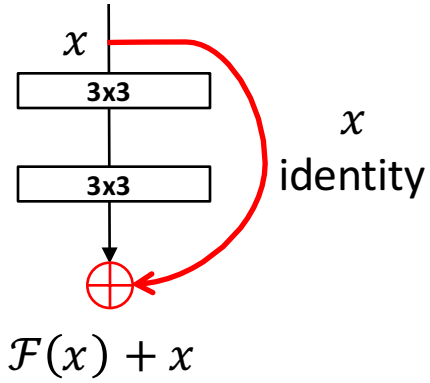
DenseNet



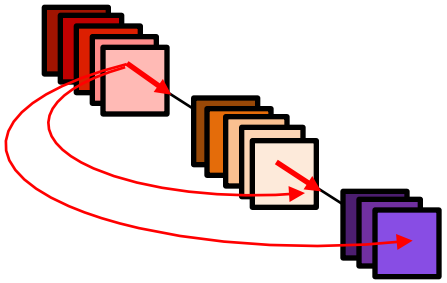
- Dense blocks where each layer is connected to every other layer in feedforward fashion;
- Alleviates vanishing gradient, strengthens feature propagation, encourages feature reuse.

Huang G, Liu Z, Van Der Maaten L, et al. Densely Connected Convolutional Networks[C]//CVPR. 2017, 1(2): 3.

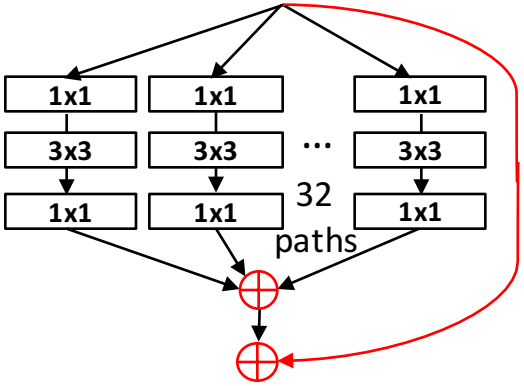
- CNN Architectures - Short-cut (Resnet, DenseNet, ResNeXt…)



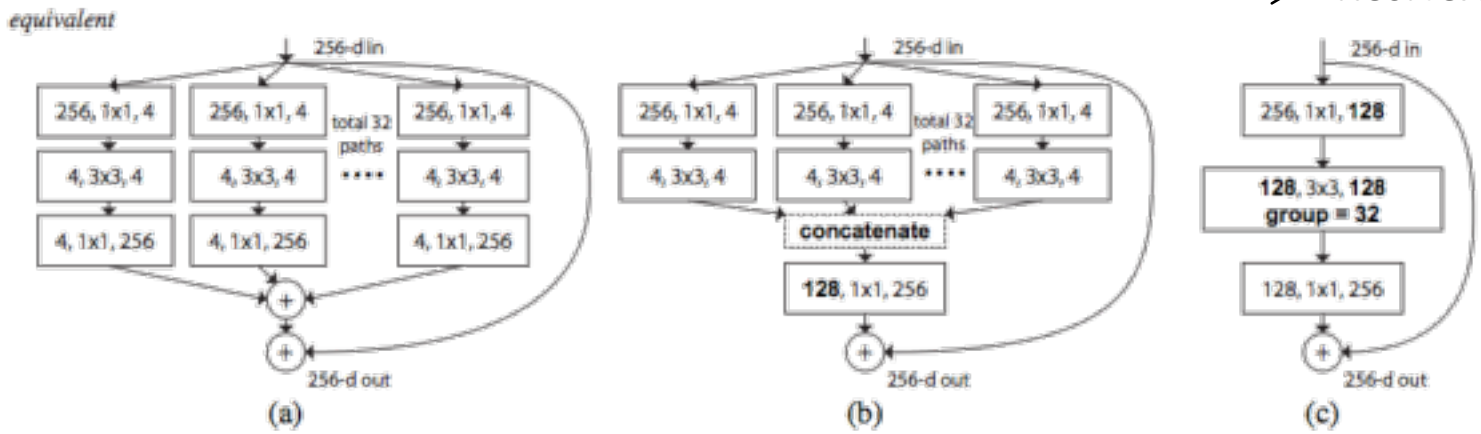
ResNet



DenseNet



ResNeXt

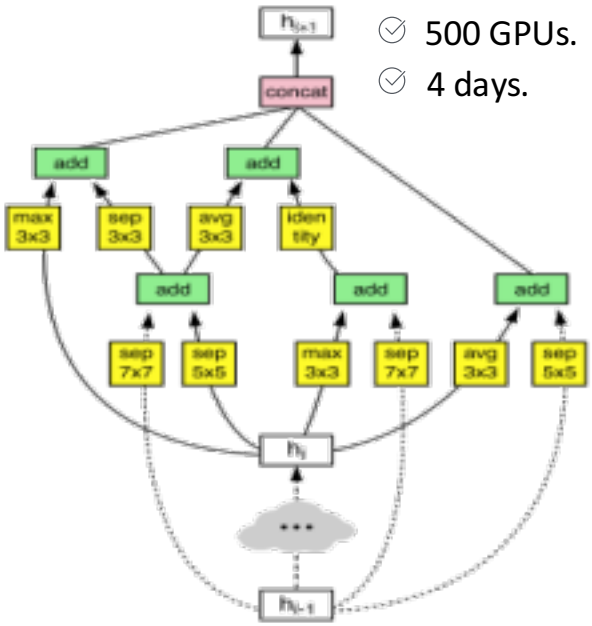


- Also from creators of ResNet
- Increases width of residual block through multiple parallel pathways ("cardinality")
- Parallel pathways similar in spirit to Inception module

Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks[C]//Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, 2017: 5987-5995.

- CNN Architectures - AutoML

- NasNet



- 500 GPUs.
- 4 days.

Reduction Cell

- Achieves a new state-of-the-art performance for Cifar-10 (97.6%)
- Transfer to large-scale classification dataset **ImageNet** and object detection dataset **COCO**, exceeding state-of-the-art performance.

On CIFAR-10, a NASNet found by this method achieves 2.4% error rate, which is state-of-the-art.

model	depth	# params	error rate (%)
DenseNet ($L = 40, k = 12$) [26]	40	1.0M	5.24
DenseNet($L = 100, k = 12$) [26]	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) [26]	100	27.2M	3.74
DenseNet-BC ($L = 100, k = 40$) [26]	190	25.6M	3.46
Shake-Shake 26 2x32d [18]	26	2.9M	3.55
Shake-Shake 26 2x96d [18]	26	26.2M	2.86
Shake-Shake 26 2x96d + cutout [12]	26	26.2M	2.56
NAS v3 [71]	39	7.1M	4.47
NAS v3 [71]	39	37.4M	3.65
NASNet-A (6 @ 768)	-	3.3M	3.41
NASNet-A (6 @ 768) + cutout	-	3.3M	2.65
NASNet-A (7 @ 2304)	-	27.6M	2.97
NASNet-A (7 @ 2304) + cutout	-	27.6M	2.40
NASNet-B (4 @ 1152)	-	2.6M	3.73
NASNet-C (4 @ 640)	-	3.1M	3.59

NASNet-A (6 @ 4032) operating on images of the same spatial resolution(800 × 800) achieves mAP = 40.7%, exceeding equivalent object detection systems based off lesser performing image featurization (i.e. Inception-ResNet-v2) by 4.0%

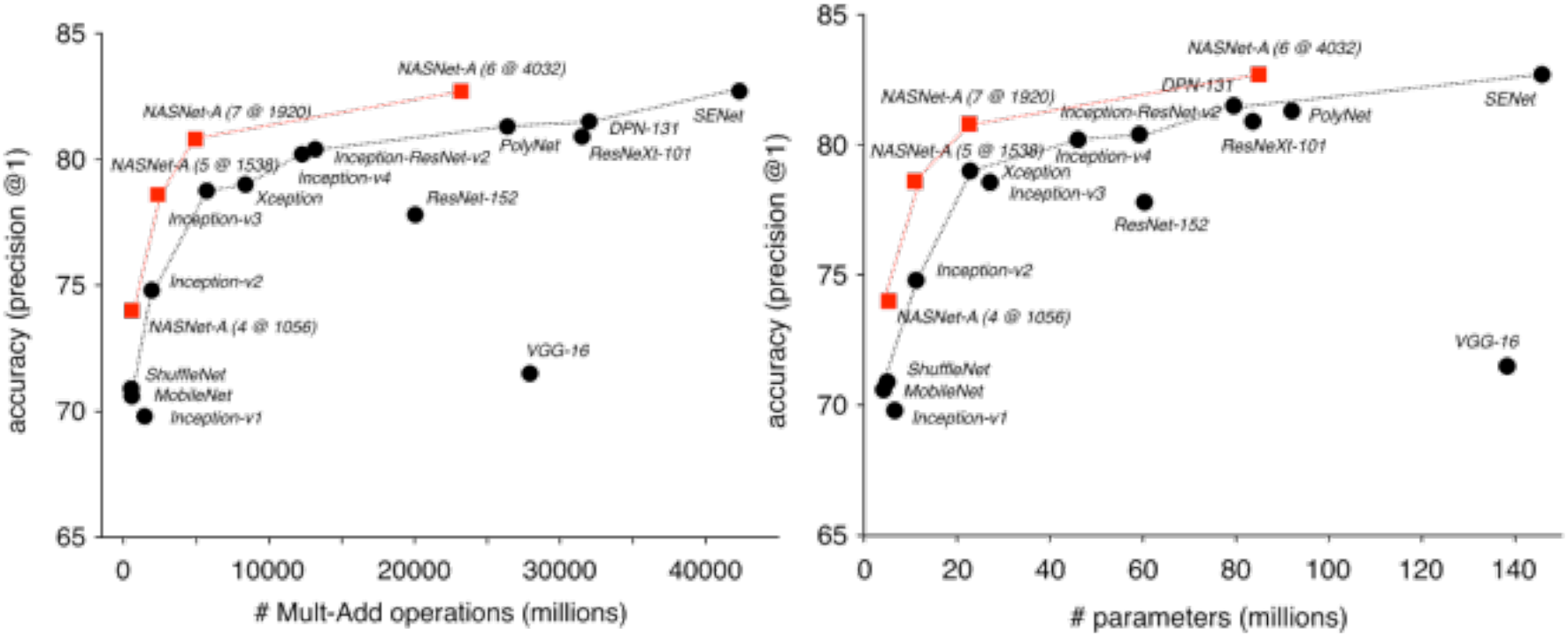
Model	resolution	mAP (mini-val)	mAP (test-dev)
MobileNet-224 [24]	600 × 600	19.8%	-
ShuffleNet (2x) [70]	600 × 600	24.5%†	-
NASNet-A (4 @ 1056)	600 × 600	**29.6%**	-
ResNet-101-FPN [36]	800 (short side)	-	36.2%
Inception-ResNet-v2 (G-RMI) [28]	600 × 600	35.7%	35.6%
Inception-ResNet-v2 (TDM) [52]	600 × 1000	37.3%	36.8%
NASNet-A (6 @ 4032)	800 × 800	41.3%	40.7%
NASNet-A (6 @ 4032)	1200 × 1200	**43.2%**	**43.1%**
ResNet-101-FPN (RetinaNet) [37]	800 (short side)	-	39.1%

Zoph B, Vasudevan V, Shlens J, et al. Learning transferable architectures for scalable image recognition[J]. arXiv preprint arXiv:1707.07012, 2017, 2(6).

- CNN Architectures - AutoML

  - NasNet

Merely transfer the architectures from CIFAR-10 but train all ImageNet models weights from scratch.

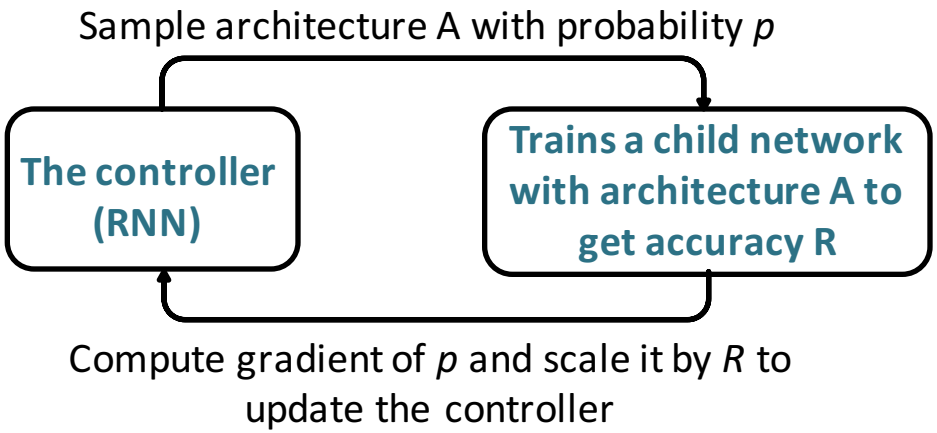


The largest model (NASNet-A(6 @ 4032)) achieves a new state-of-the-art performance for ImageNet (82.7%) based on single, non-ensembled predictions, surpassing previous best published result by ~1.2%(DPN).

Zoph B, Vasudevan V, Shlens J, et al. Learning transferable architectures for scalable image recognition[J]. arXiv preprint arXiv:1707.07012, 2017, 2(6).

- CNN Architectures - AutoML

  - NasNet



Controller

A controller recurrent neural network (LSTM) samples child networks with different architectures.

Child Network

The child networks are trained to convergence to obtain some accuracy on a held-out validation set.

Reward

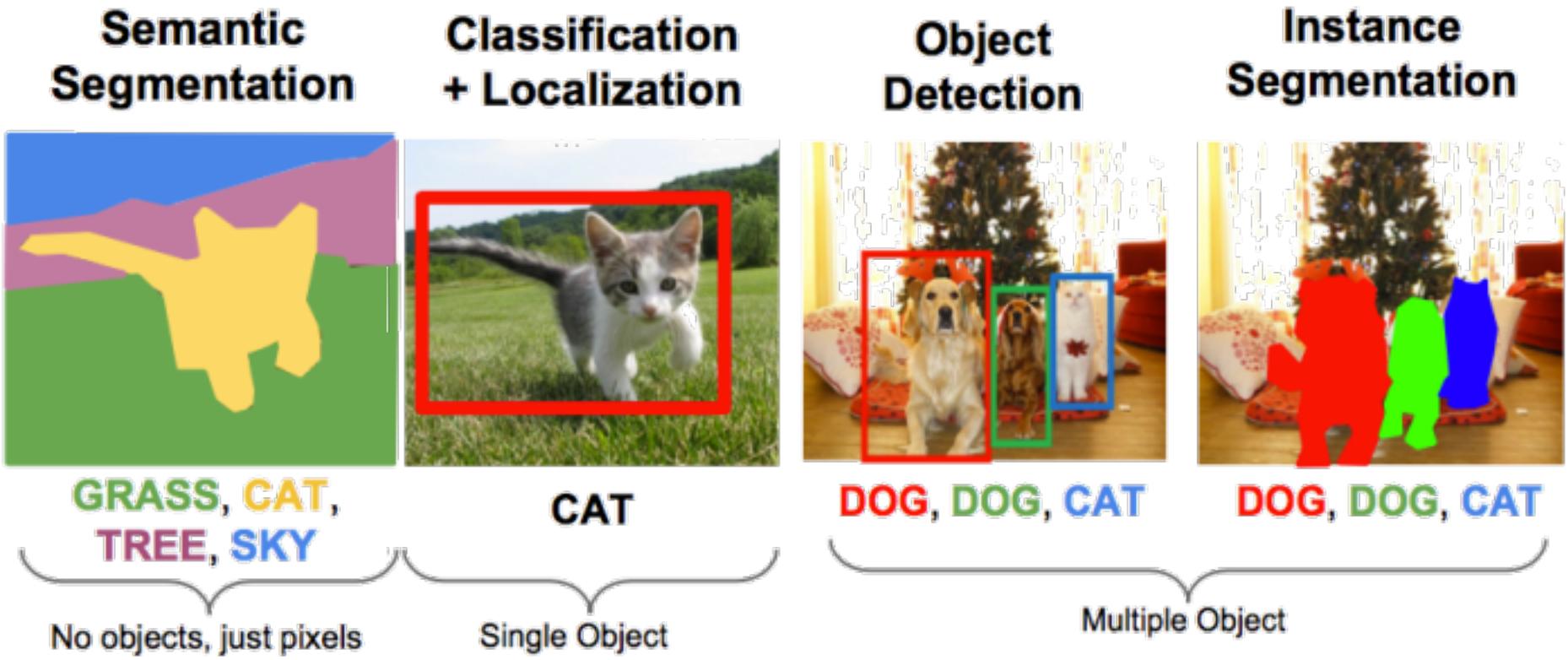
The resulting accuracies are used to update the controller so that the controller will generate better architectures over time.

Zoph B, Vasudevan V, Shlens J, et al. Learning transferable architectures for scalable image recognition[J]. arXiv preprint arXiv:1707.07012, 2017, 2(6).

Outline

- CNN Techniques
  - Dilated/Deformable 2D Convolution, 3D Convolution
  - ROI Pooling
  - Batch/Layer/Instance/Group Normalization
  - Activation Functions, Dropout, Weight Initialization
- CNN Architectures
  - AlexNet/VGGNet
  - CoogLeNet (Inception v1-v4)
  - Short-cut (Resnet, DenseNet, ResNeXt)
  - AutoML
- Segmentation & Detection
  - Semantic/Instance Segmentation
  - (Video) Object Detection
- Others
  - Image Caption
  - Generative Adversarial Network
  - Style transfer

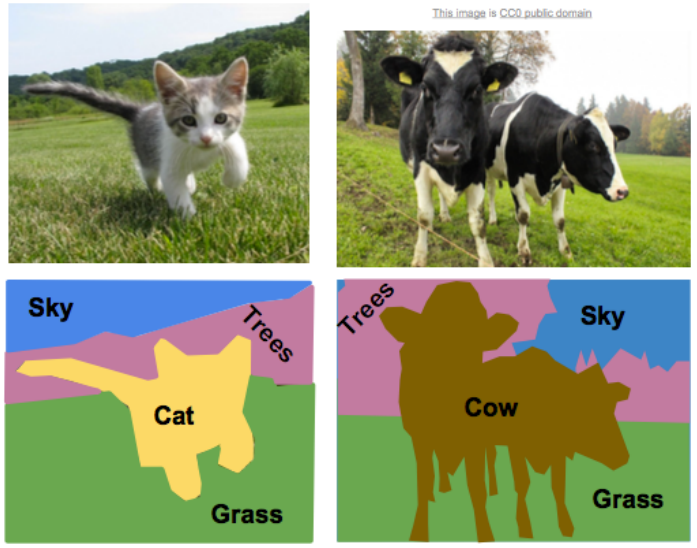
- Segmentation & Detection



- Segmentation & Detection - Semantic/Instance Segmentation

Semantic Segmentation is to assign a semantic label, such as “cat”, “dog”, “person”, “sky”, …, to each pixel in an image.

- Input: An image $\mathcal{X} = \begin{bmatrix} x_{11} & \cdots & x_{1w} \\ \vdots & \ddots & \vdots \\ x_{h1} & \cdots & x_{hw} \end{bmatrix}$, where w is the width and h is the height of the given image.
- Output: Segmentation mask $\mathcal{M} = \begin{bmatrix} m_{11} & \cdots & m_{1w} \\ \vdots & \ddots & \vdots \\ m_{h1} & \cdots & m_{hw} \end{bmatrix}$, each pixel $m_{\mathrm{ij}} \in \mathcal{L}$, where $\mathcal{L}$ is the label space.

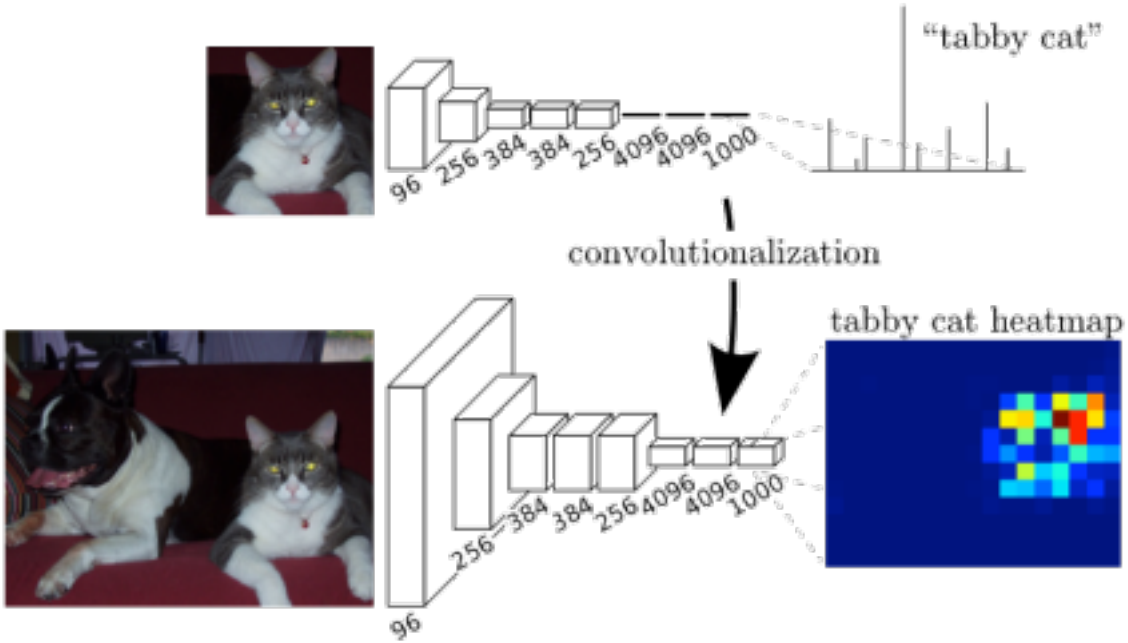


Recommendation

- ✓ FCN
- ✓ SegNet
- ✓ Dilated Convolutions
- ✓ DeepLab v1 & v2
- ✓ RefineNet
- ✓ PSPNet
- ✓ Large Kernel Matters
- ✓ DeepLab v3

- Segmentation & Detection - Semantic Segmentation

Fully Convolutional Network is the first work that introduces ANNFCN to image segmentation area. The main insight is the replacement of fully connected layer by fully convolutional layer.

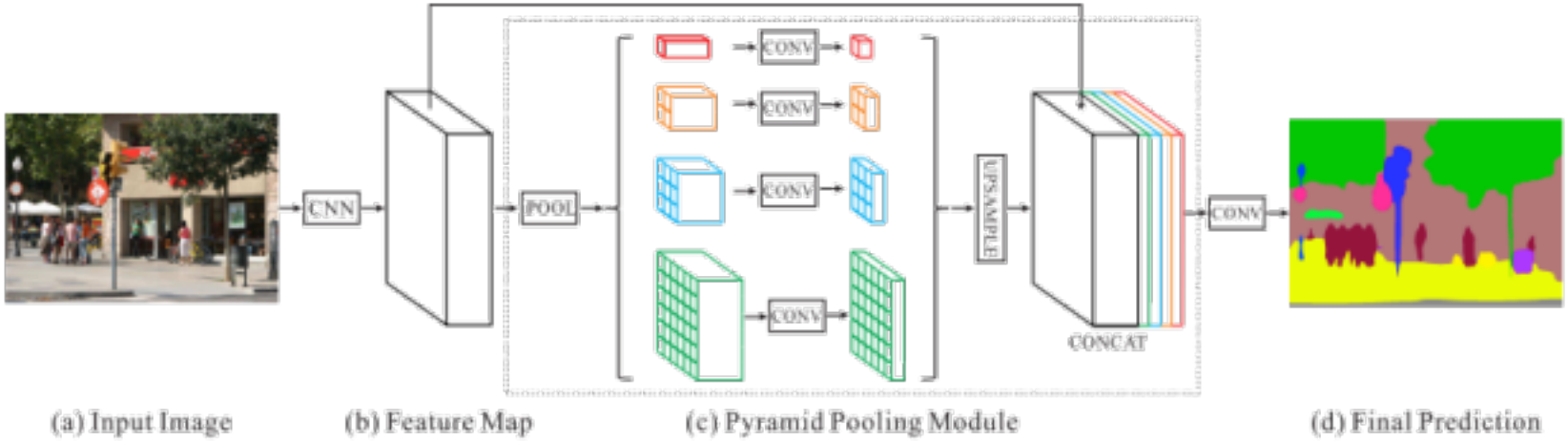


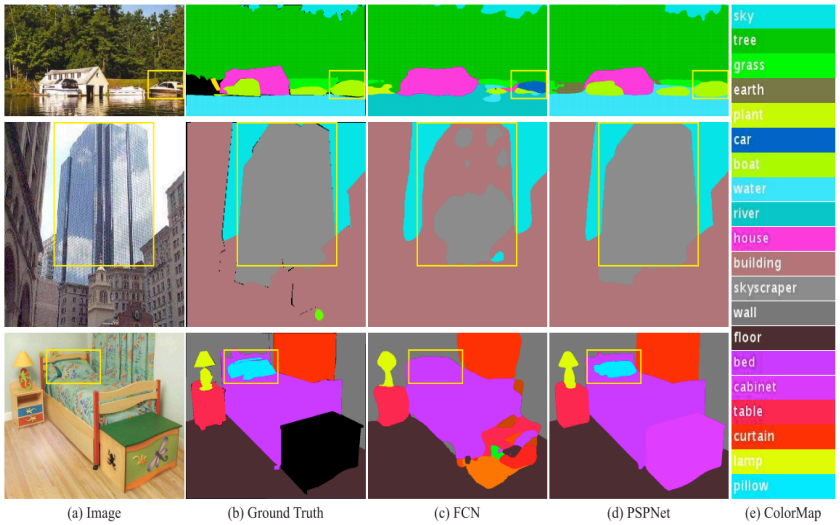
- Popularize the use of end-to-end CNNs for semantic segmentation
- Re-purpose ImageNet pretrained models for segmentation
- Replace upsampling by *deconvolutional* layers
- Introduce skip connections to improve the coarseness of upsampling

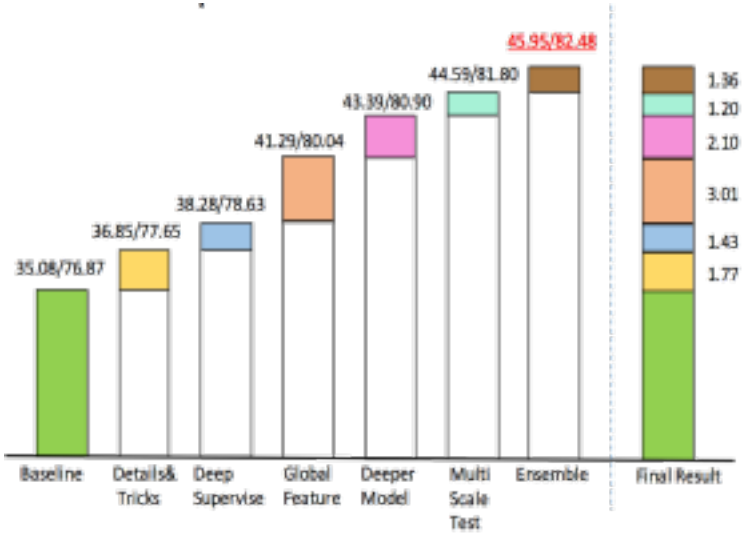
Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation.

- Segmentation & Detection - Semantic Segmentation

Pyramid Scene Parsing Network exploits the capability of global context information by different-region based context aggregation through pyramid pooling module.



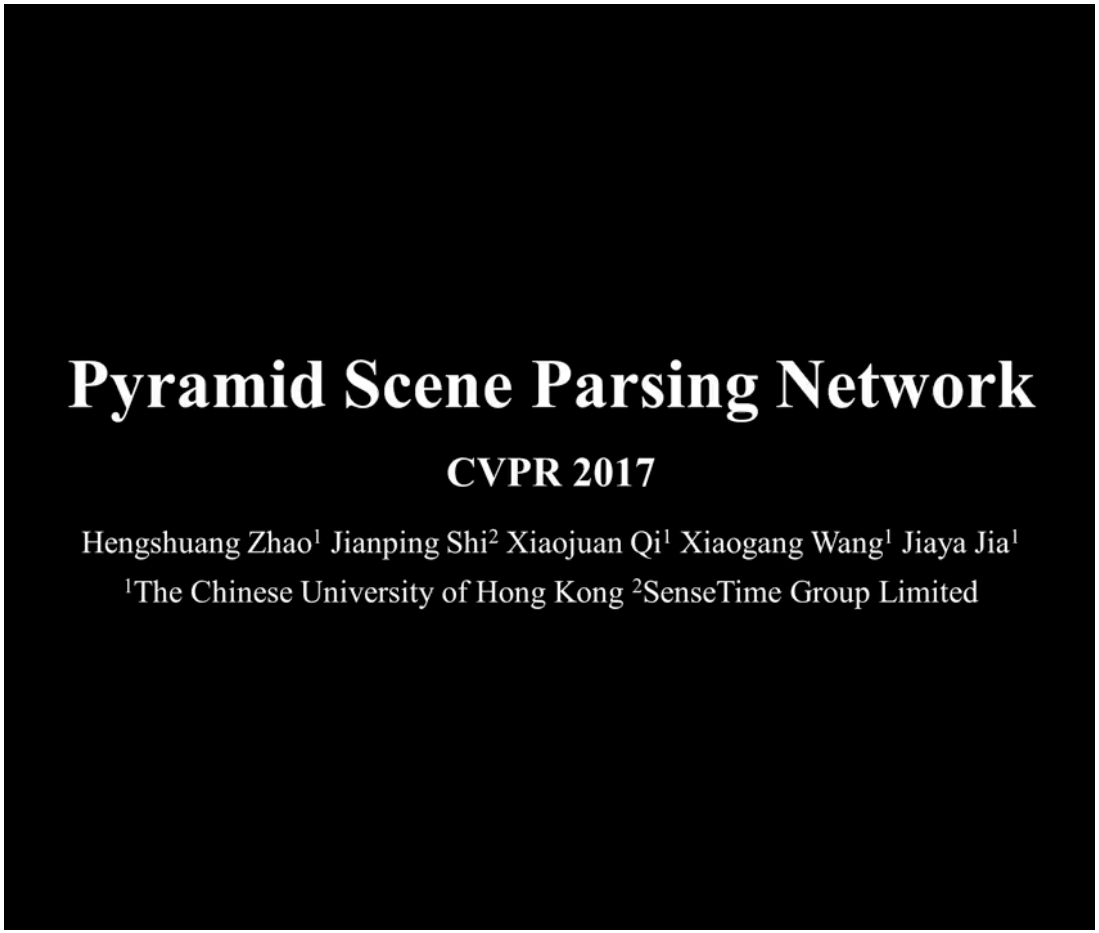




[1] Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network[C]//IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2017: 2881-2890.

- Segmentation & Detection - Semantic Segmentation

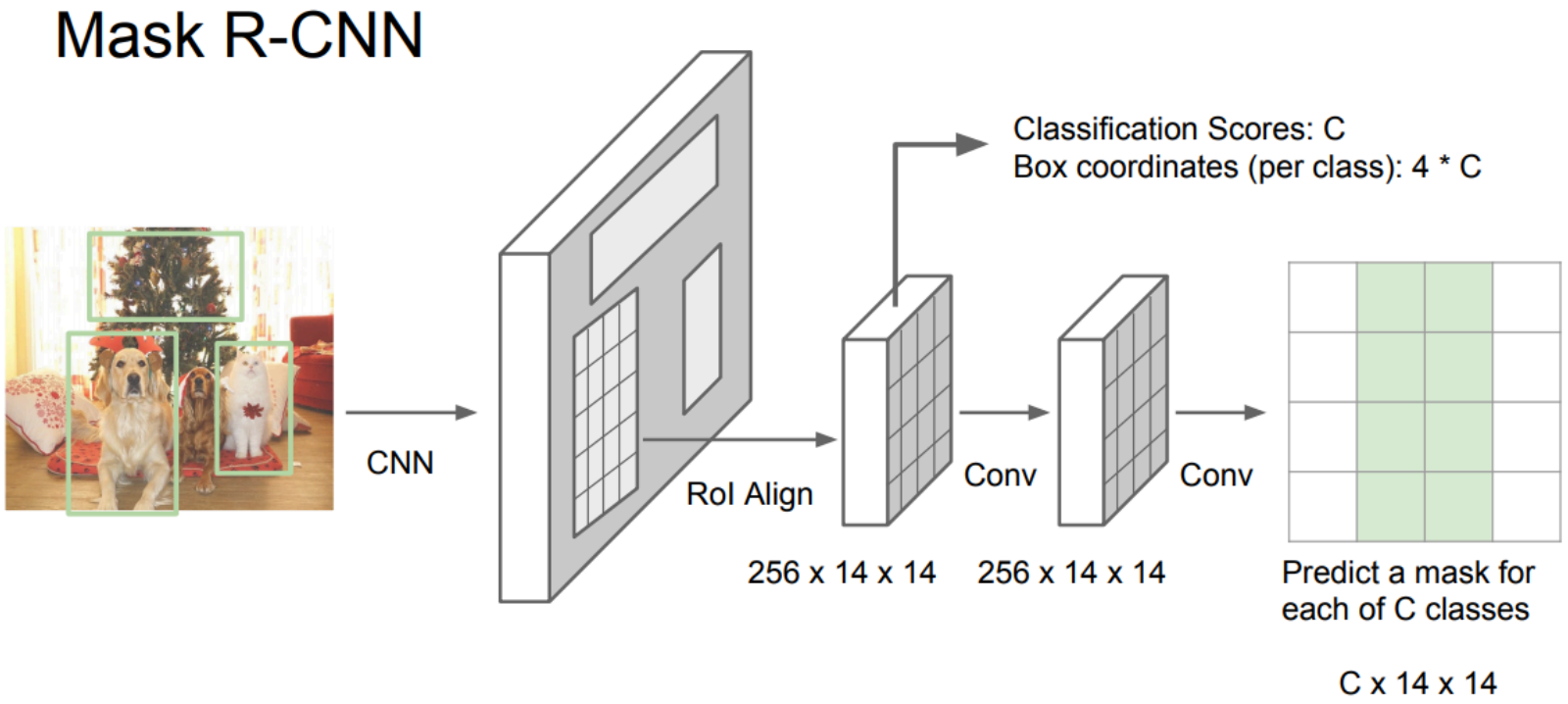
Pyramid Scene Parsing Network exploits the capability of global context information by different-region based context aggregation through pyramid pooling module.



[1] Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network[C]//IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2017: 2881-2890.

- Segmentation & Detection - Instance Segmentation

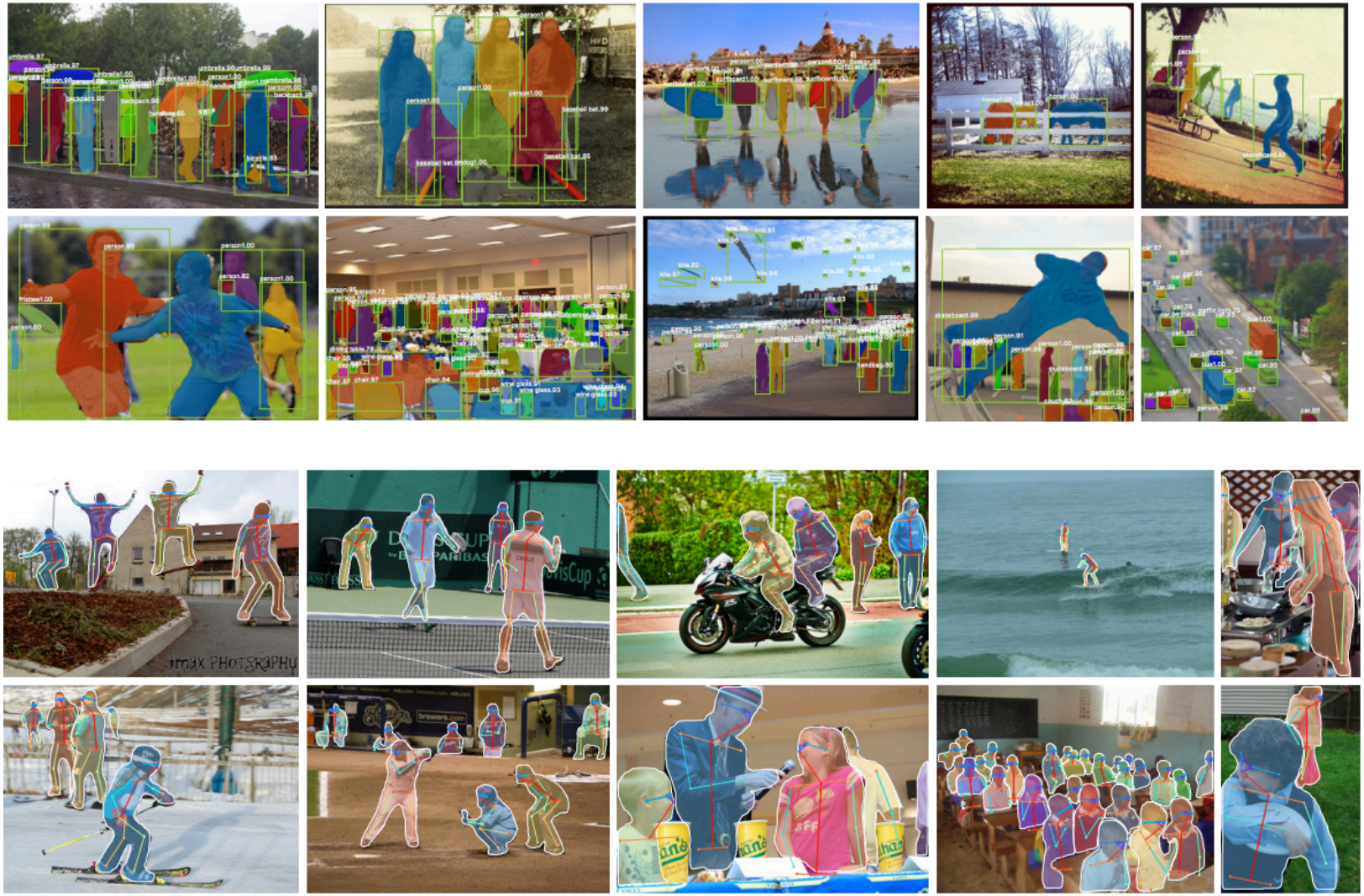
Mask R-CNN efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance.



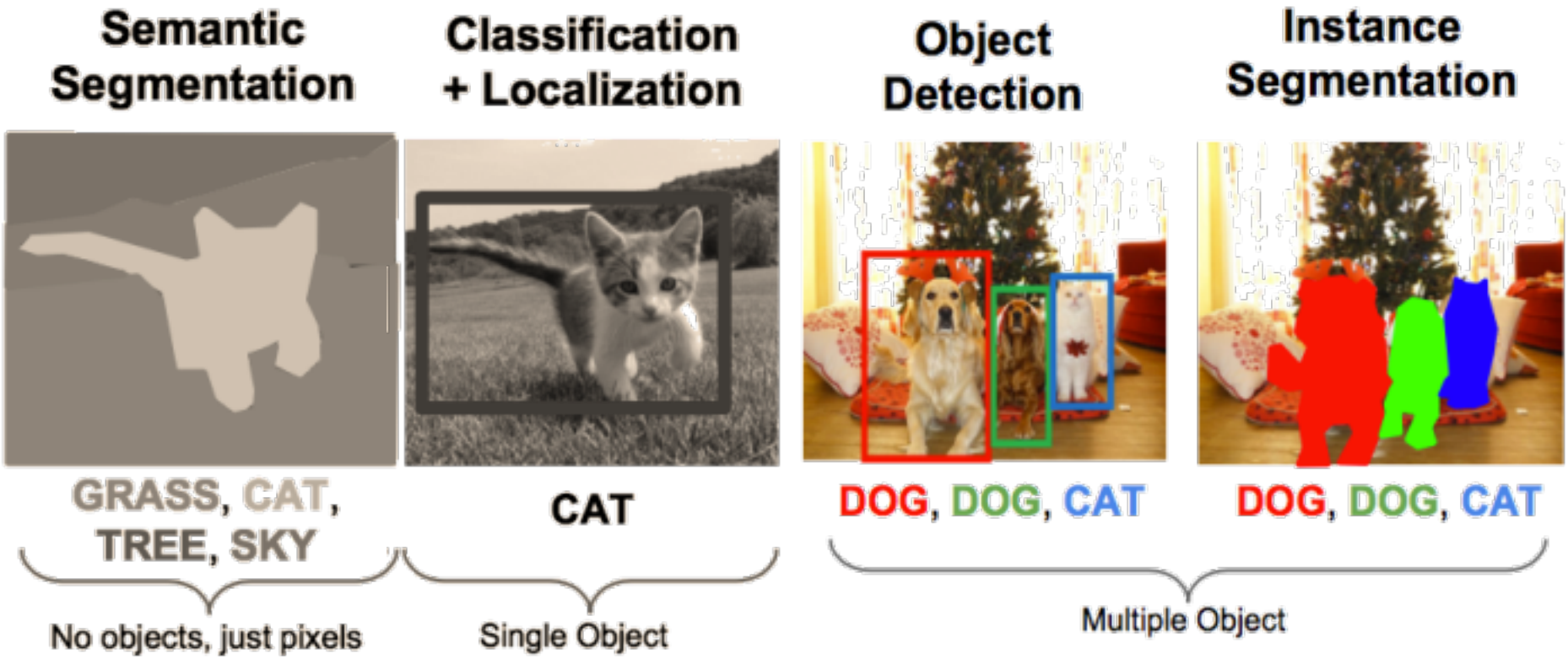
He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017: 2980-2988.

- Segmentation & Detection - Instance Segmentation

  - Mask R-CNN

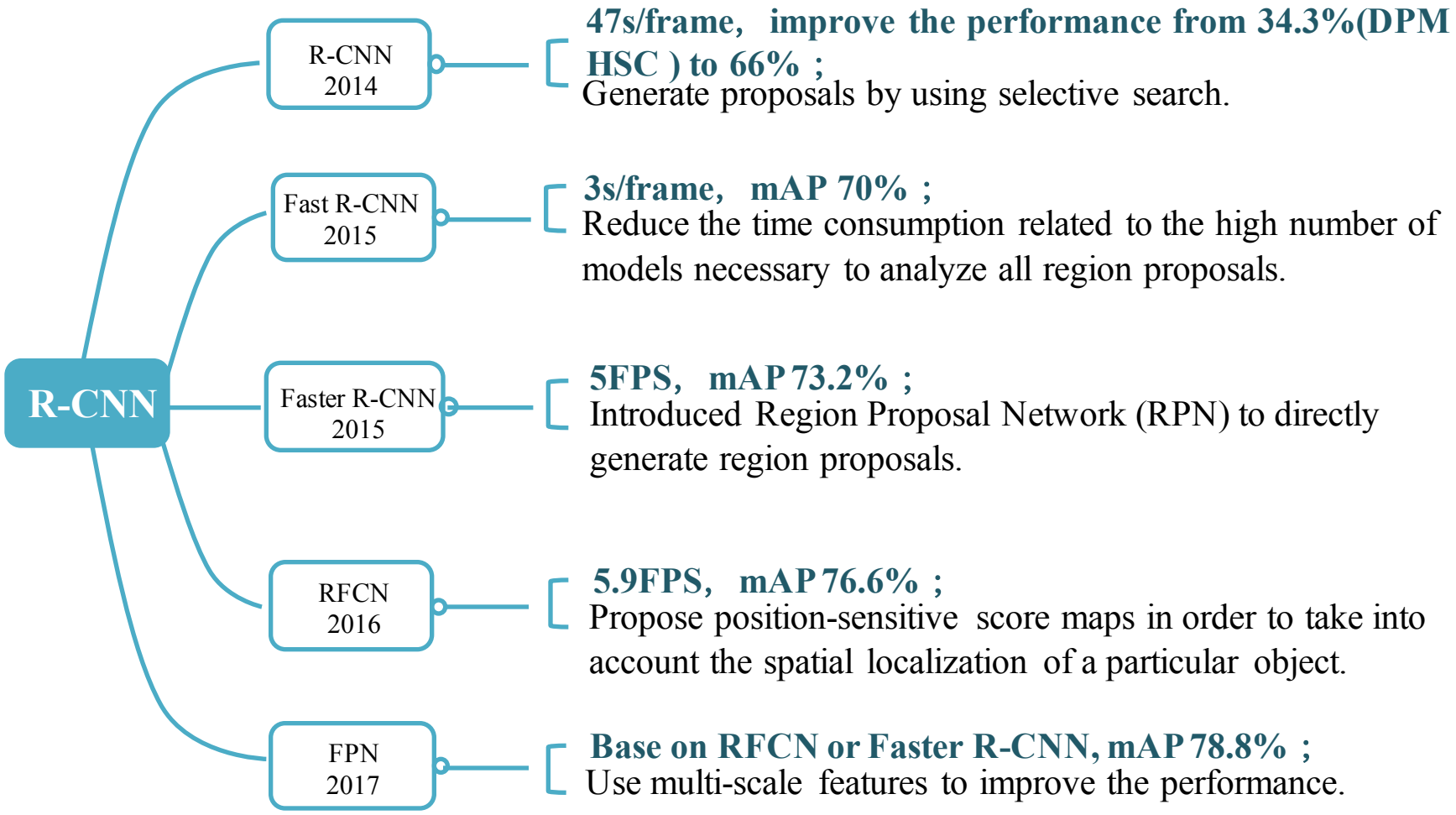


- Segmentation & Detection – (Video) Object Detection



- Segmentation & Detection – Object Detection

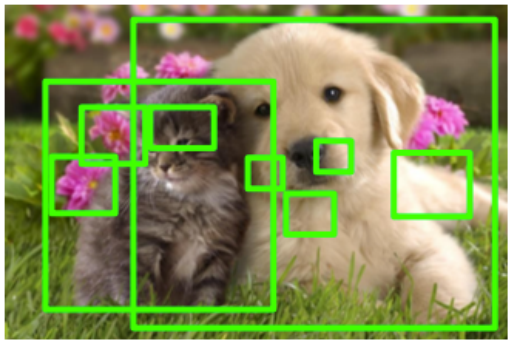
Detection – VOC2017

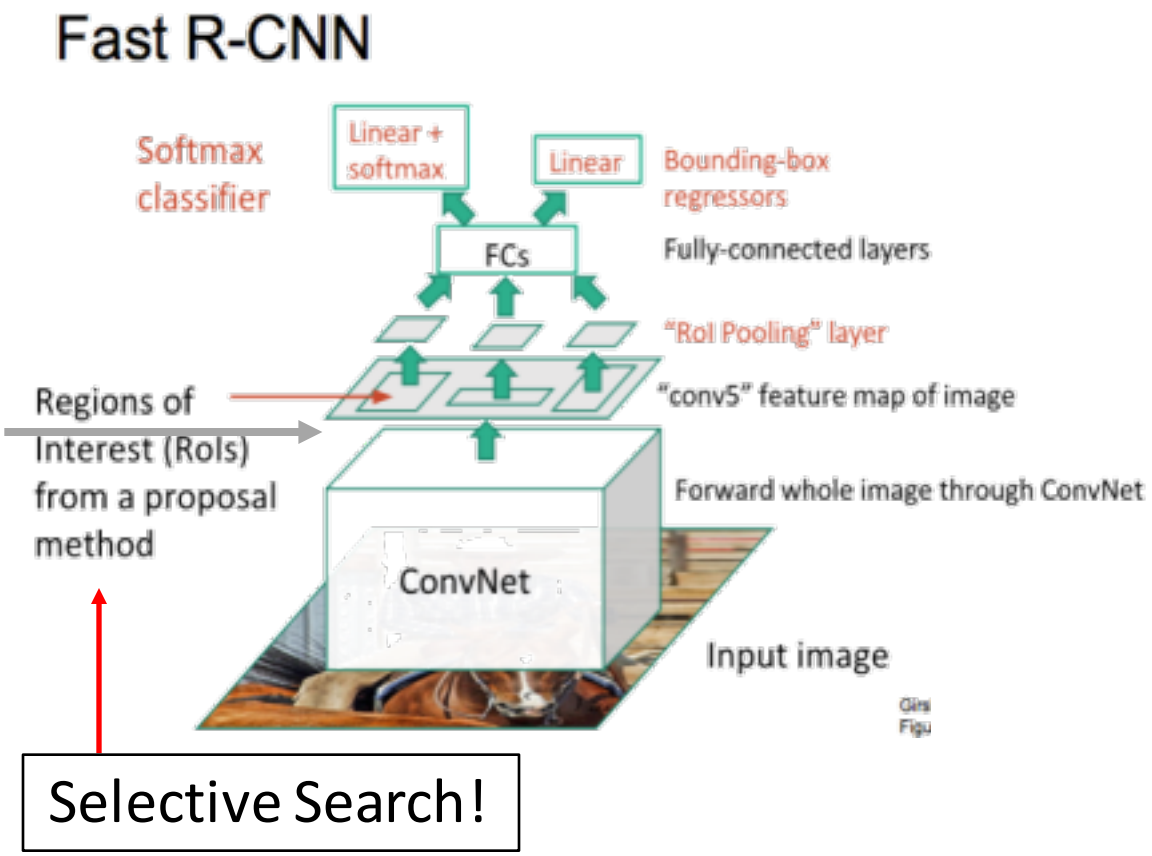


- Segmentation & Detection - Object Detection

Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. F

Proposals





Selective Search!

- Find "blobby" image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 1000 region proposals in a few seconds on CPU

Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

- Segmentation & Detection - Object Detection

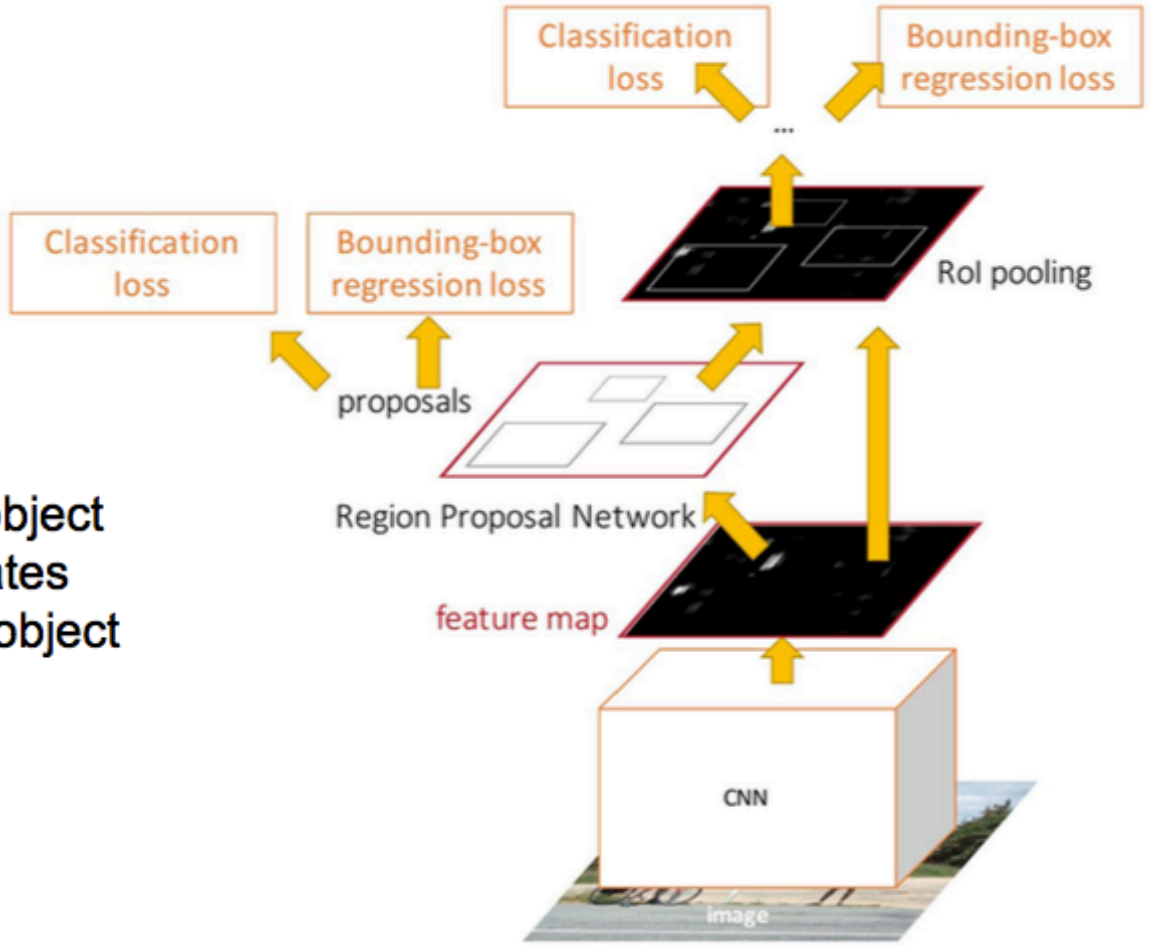
Faster R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

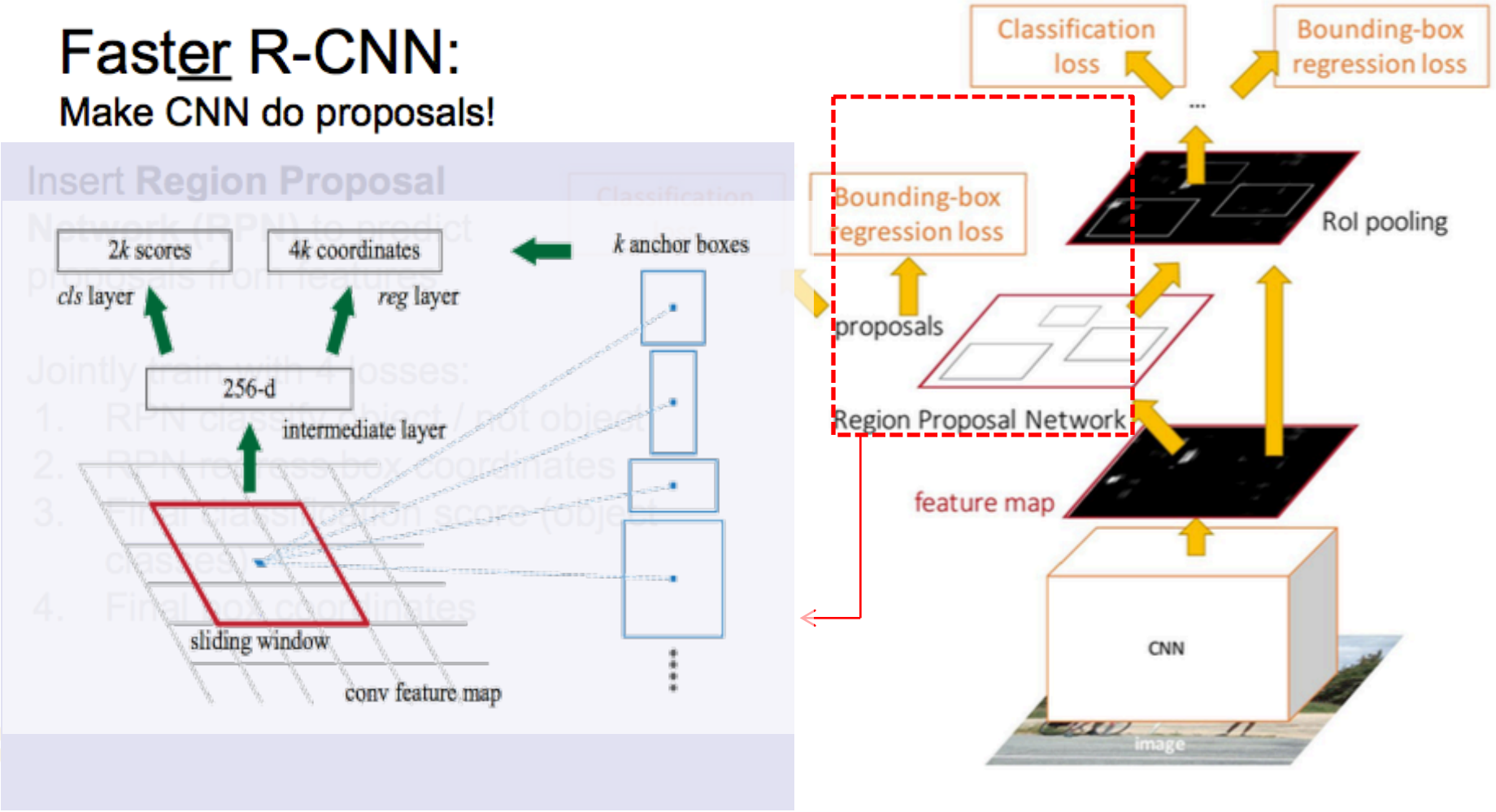
Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.

- Segmentation & Detection - Object Detection

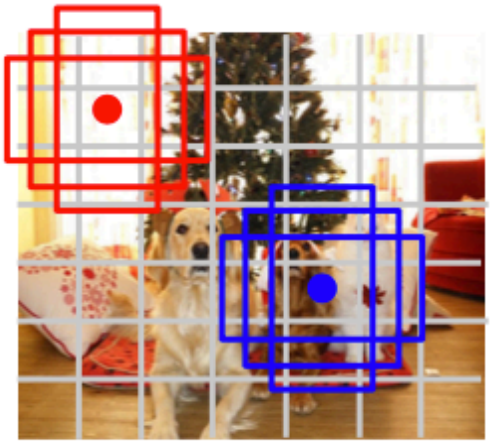


Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.

- Segmentation & Detection - Object Detection

Detection without Proposals: YOLO / SSD

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

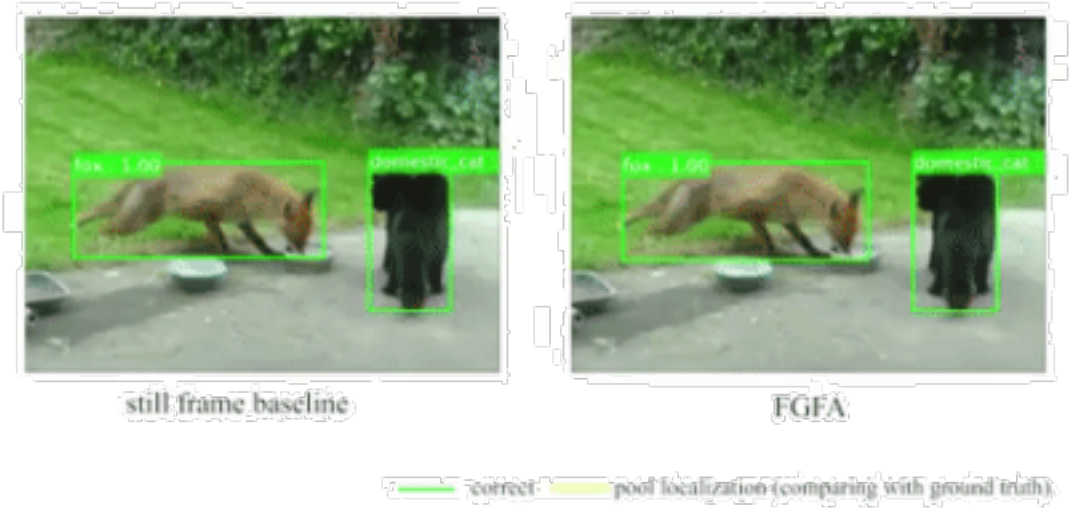
Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

Output:
7 x 7 x (5 * B + C)

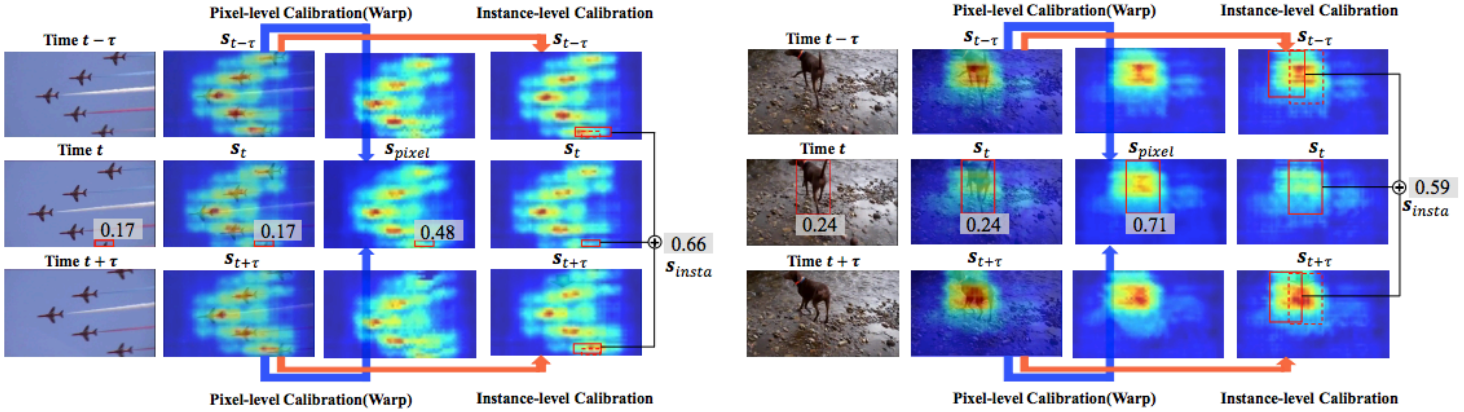
[1] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
[2] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.

- Segmentation & Detection – Video Object Detection

Flow-Guided Feature Aggregation for Video Object Detection



Fully Motion-Aware Network for Video Object Detection



(a) Occluded airplane (b) Non-rigid motion

[1] Zhu X, Wang Y, Dai J, et al. Flow-guided feature aggregation for video object detection[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017, 3.
[2] Wang S, Zhou Y, Yan J, et al. Fully Motion-Aware Network for Video Object Detection[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 542-557.

Outline

- CNN Techniques
  - Dilated/Deformable 2D Convolution, 3D Convolution
  - ROI Pooling
  - Batch/Layer/Instance/Group Normalization
  - Activation Functions, Dropout, Weight Initialization
- CNN Architectures
  - AlexNet/VGGNet
  - CoogLeNet (Inception v1-v4)
  - Short-cut (Resnet, DenseNet, ResNeXt)
  - AutoML
- Segmentation & Detection
  - Semantic/Instance Segmentation
  - (Video) Object Detection
- Others
  - Image Caption
  - Generative Adversarial Network
  - Style transfer

- From Image/Video to Language Generation

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

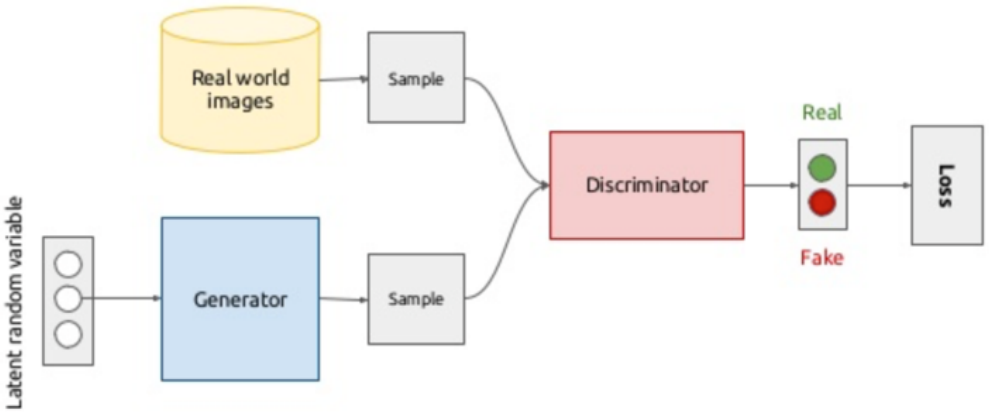
A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

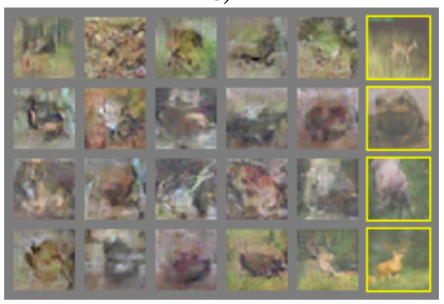
Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention[J]. arXiv preprint arXiv:1502.03044, 2015, 2(3): 5.

- Generative Adversarial Network (GAN)

- intuition: find a generative model that makes samples generated from random codes look like the training data



b)



d)

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets.

- A Neural Algorithm of Artistic Style

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.