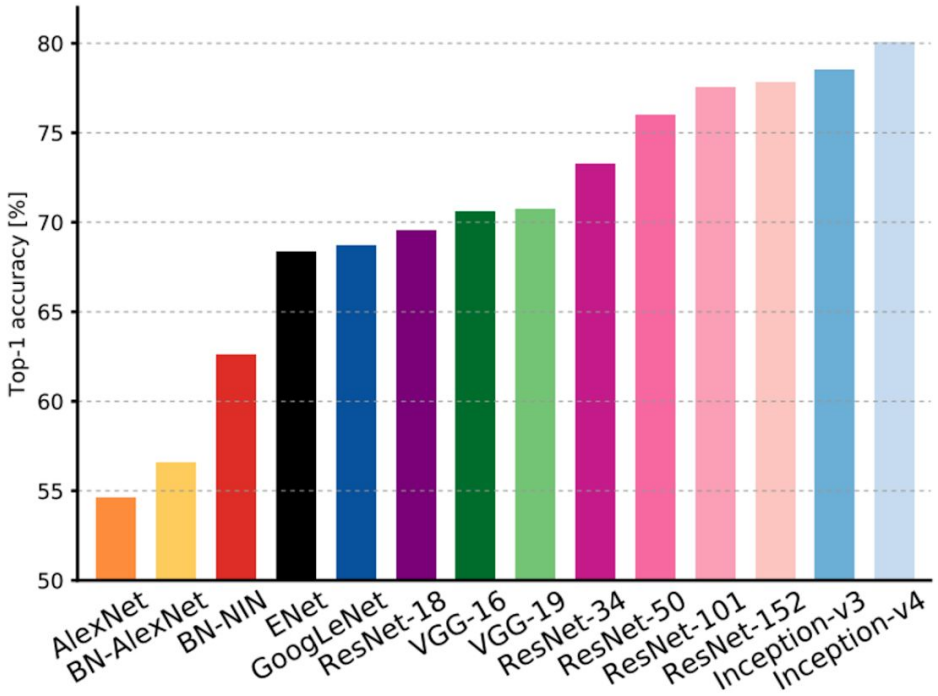
Final Note on Classical Architecture

Classical Architecture

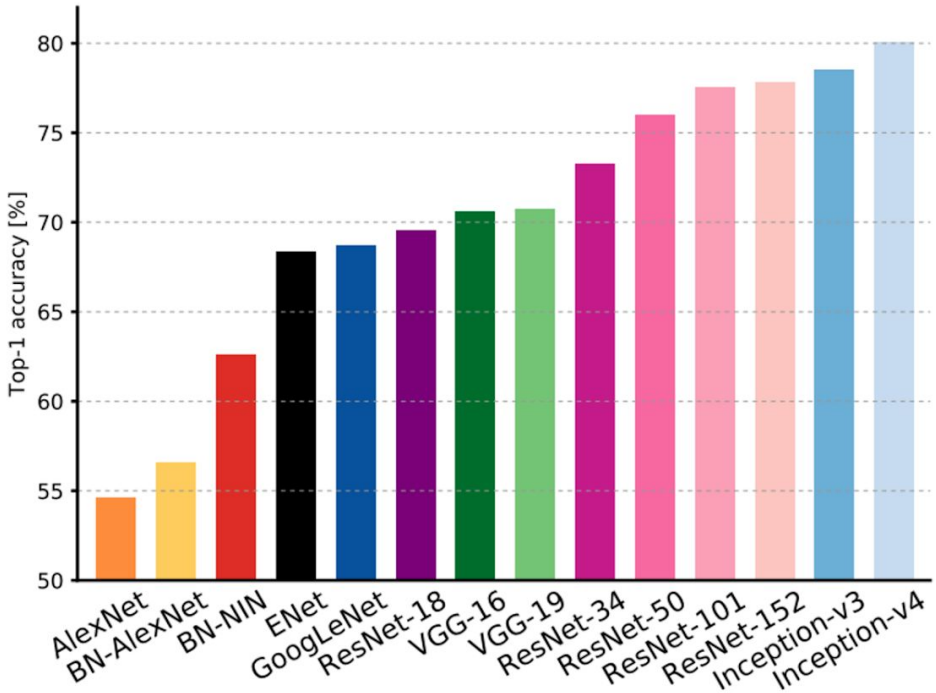
Architecture	Year	Layers	Key Innovations	Parameters	Accuracy (ImageNet)	Researchers
AlexNet	2012	8	ReLU, LRN	62 million	57.2%	Alex Krizhevsky et al.
VGGNet	2014	16-19	3x3 convolution filters, Deep architecture	138-144 million	74.4%	Karen Simonyan and Andrew Zisserman
Inception Net	2014	22-42	Inception modules, Auxiliary classifiers, Batch normalization	4-12 million	74.8%	Szegedy et al.
ResNet	2015	50-152	Residual connections, Shortcut connections	25.6-60 million	75.3%	He et al.

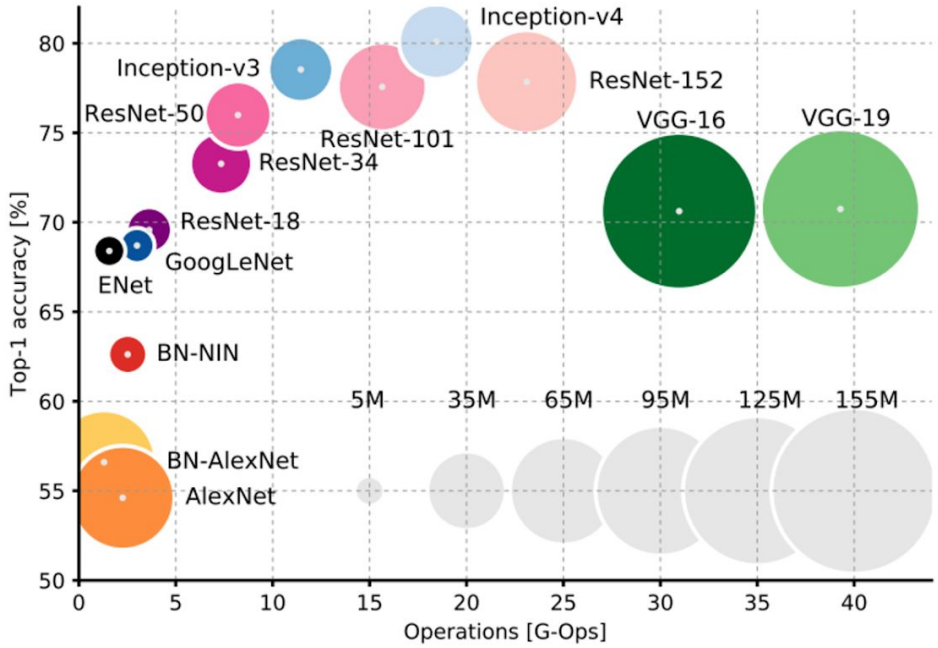
Comparing Complexity



Top1 vs. network. Single-crop top-1 validation accuracies for top scoring single-model architectures

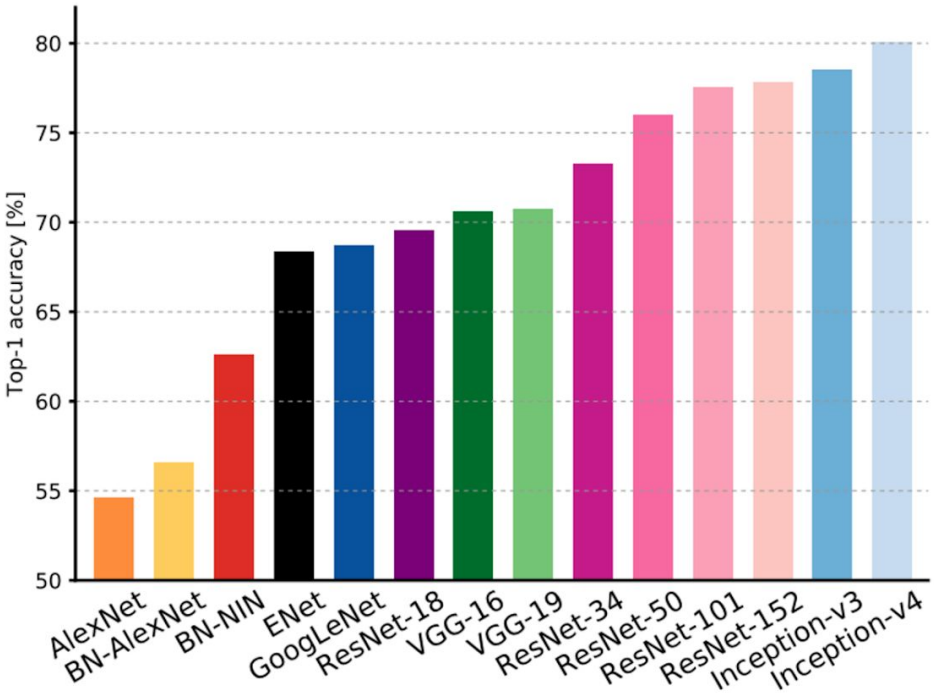
Comparing Complexity

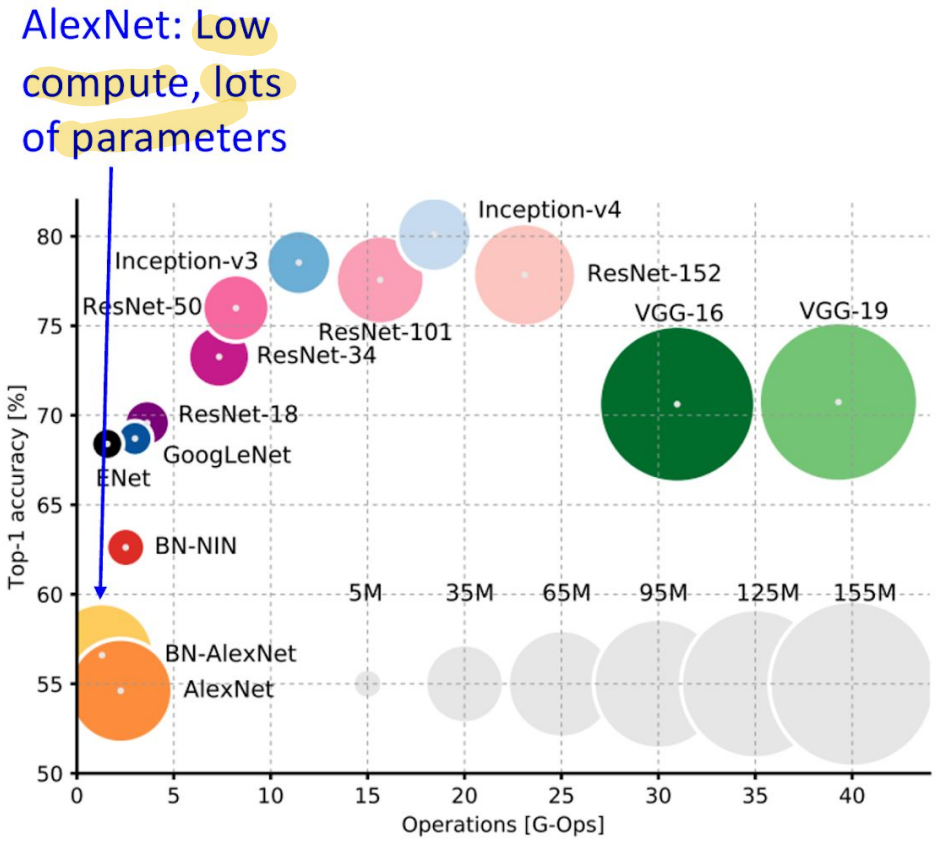




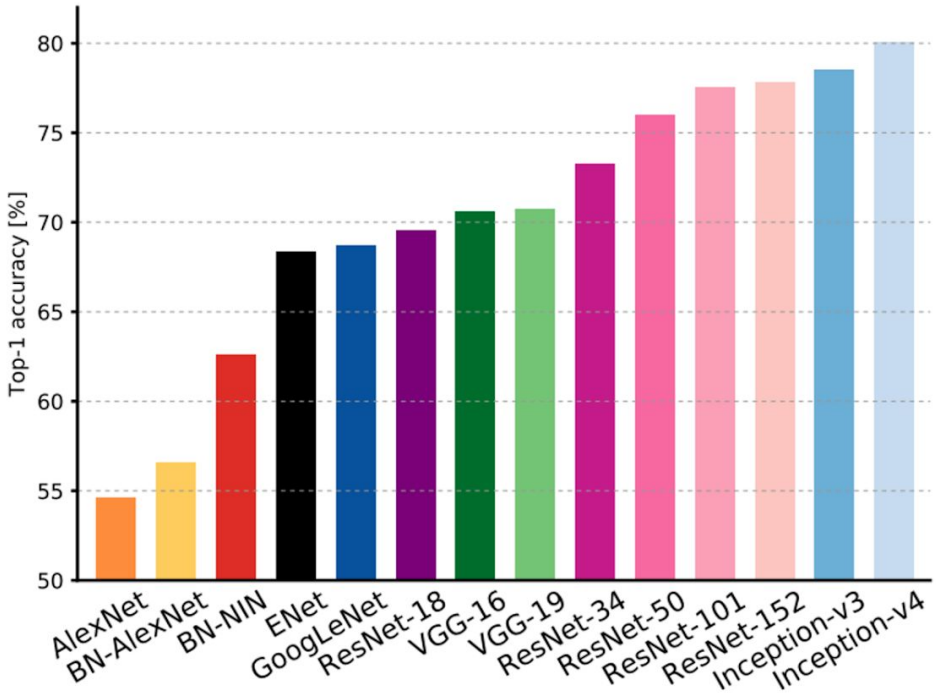
Top1 vs. operations, size ∝ parameters. Top-1 one-crop accuracy versus amount of operations required for a single forward pass. The size of the blobs is proportional to the number of network parameters

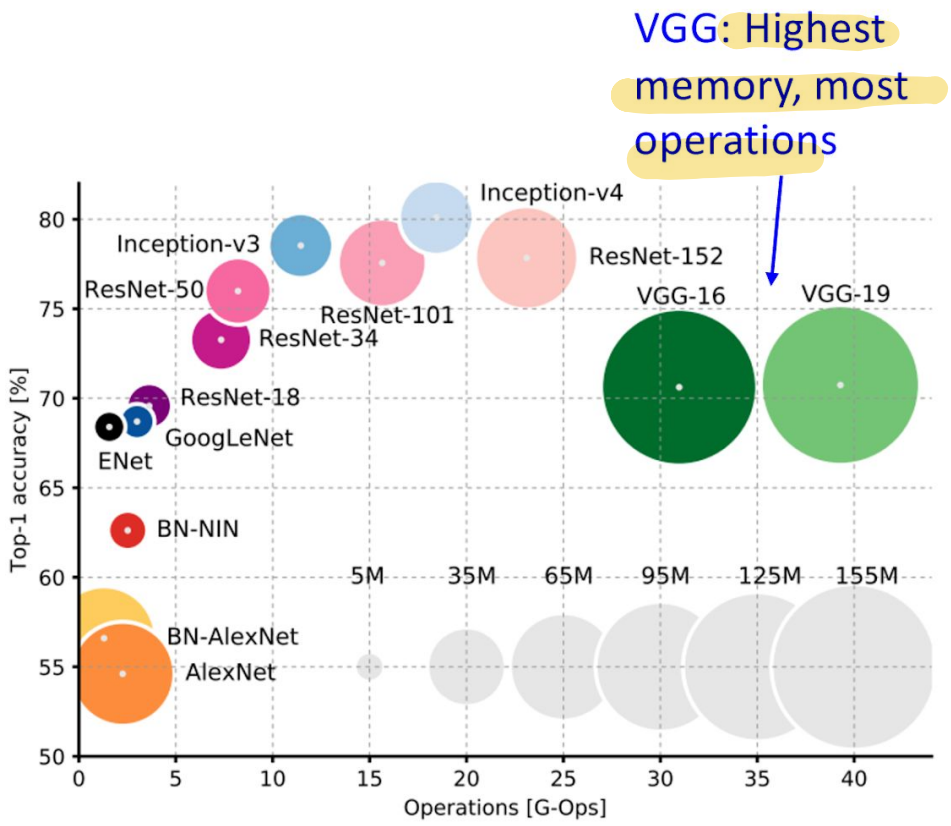
Comparing Complexity



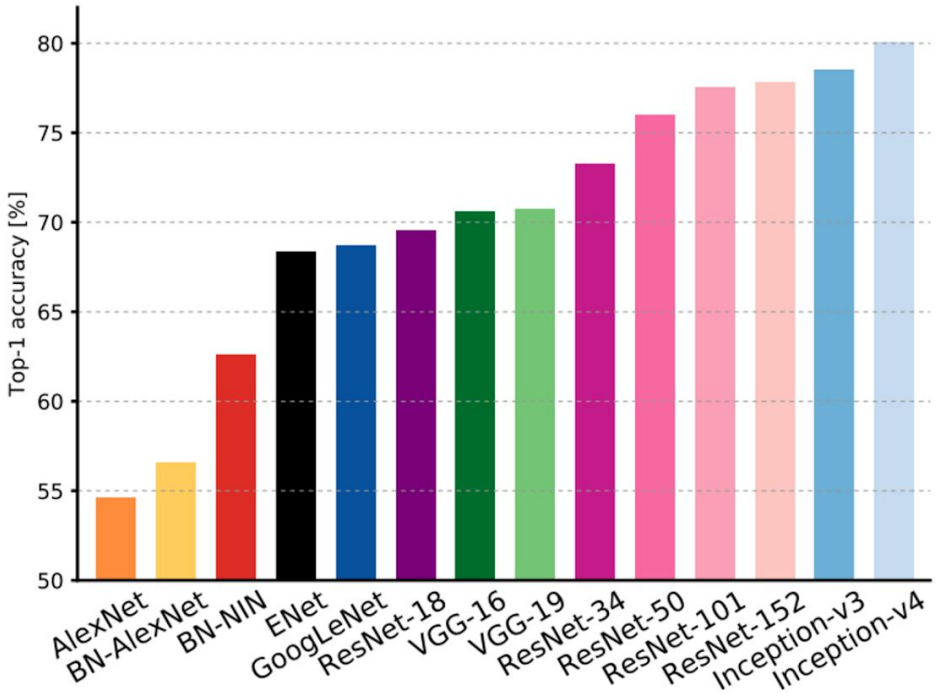


Comparing Complexity

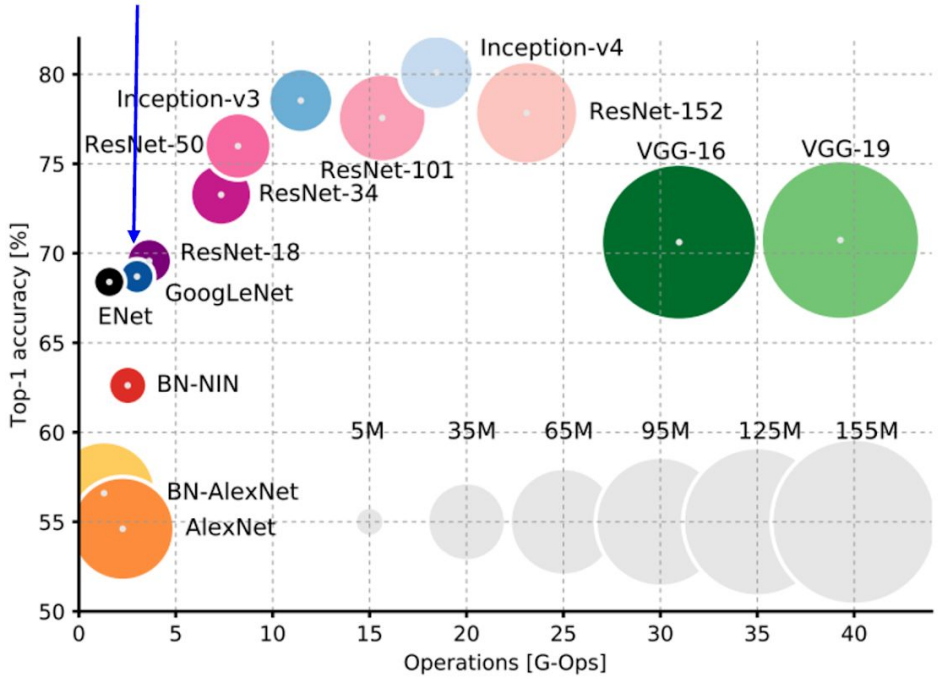




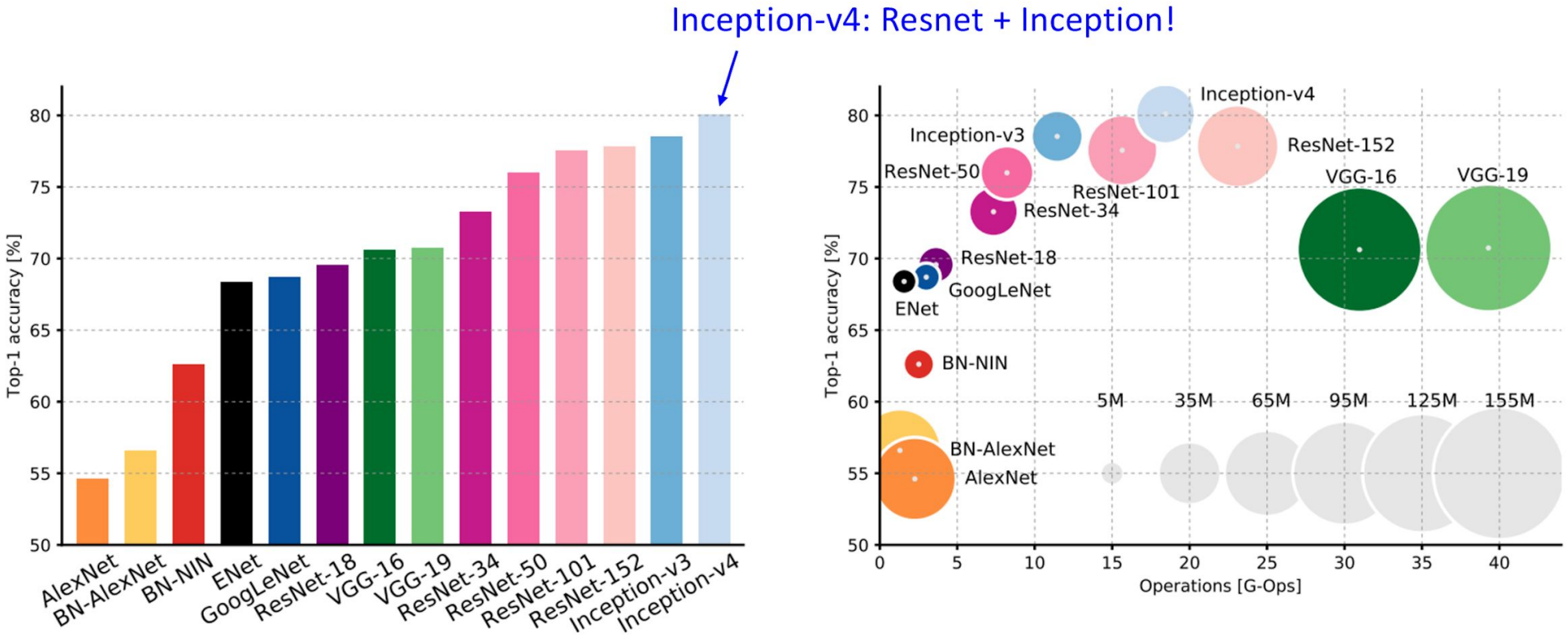
Comparing Complexity



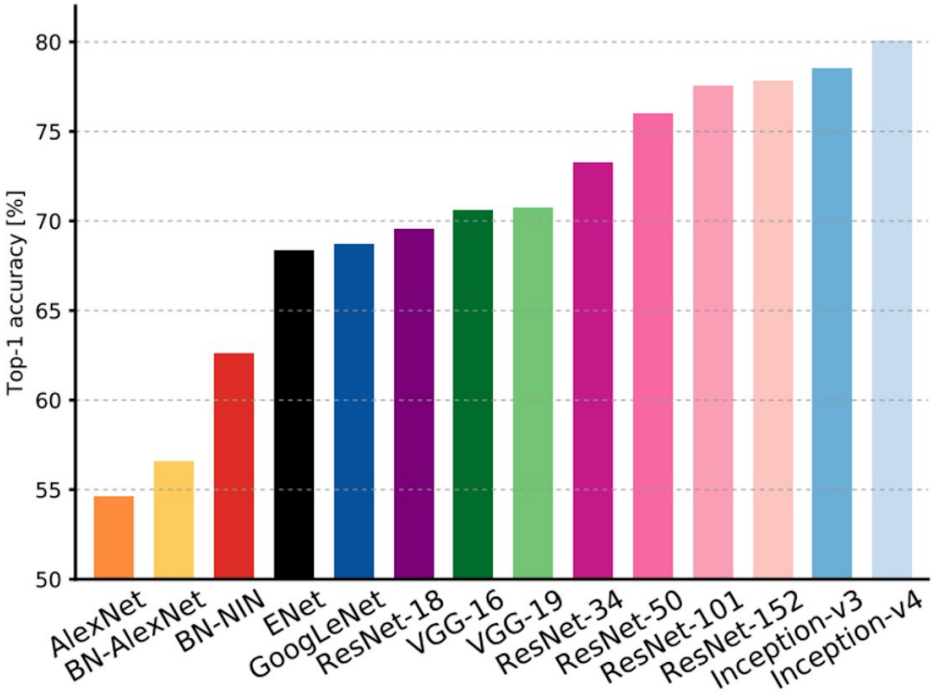
GoogLeNet:
Very efficient!

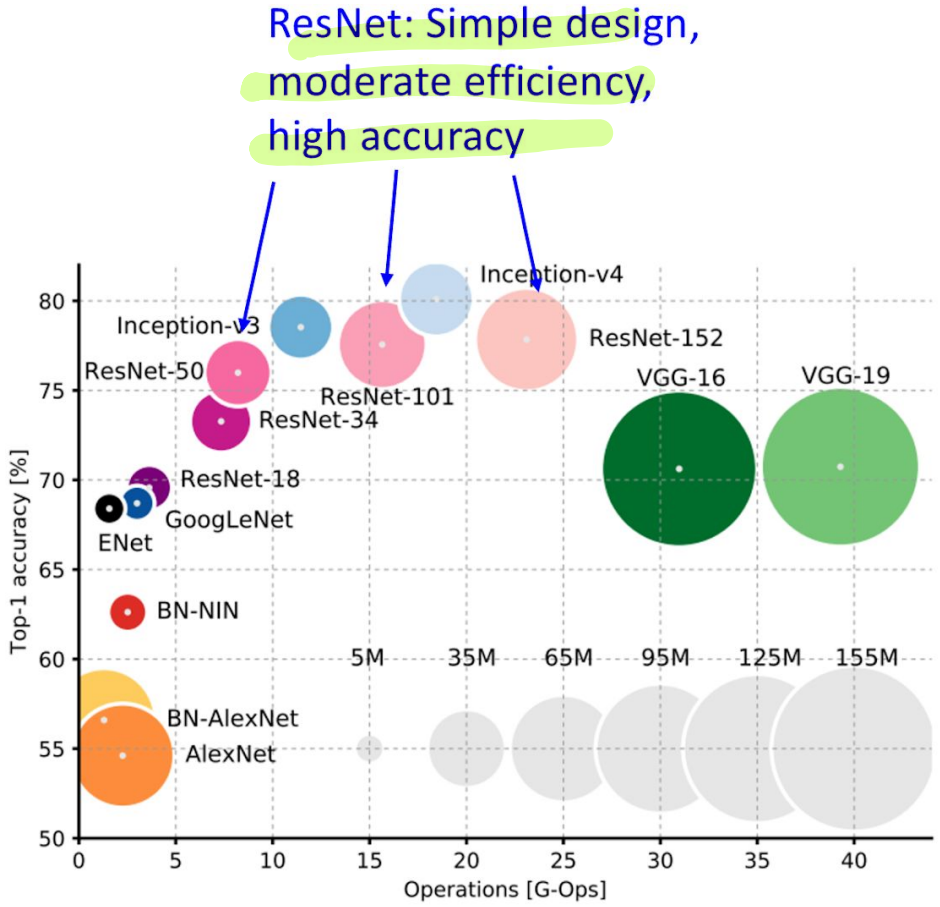


Comparing Complexity

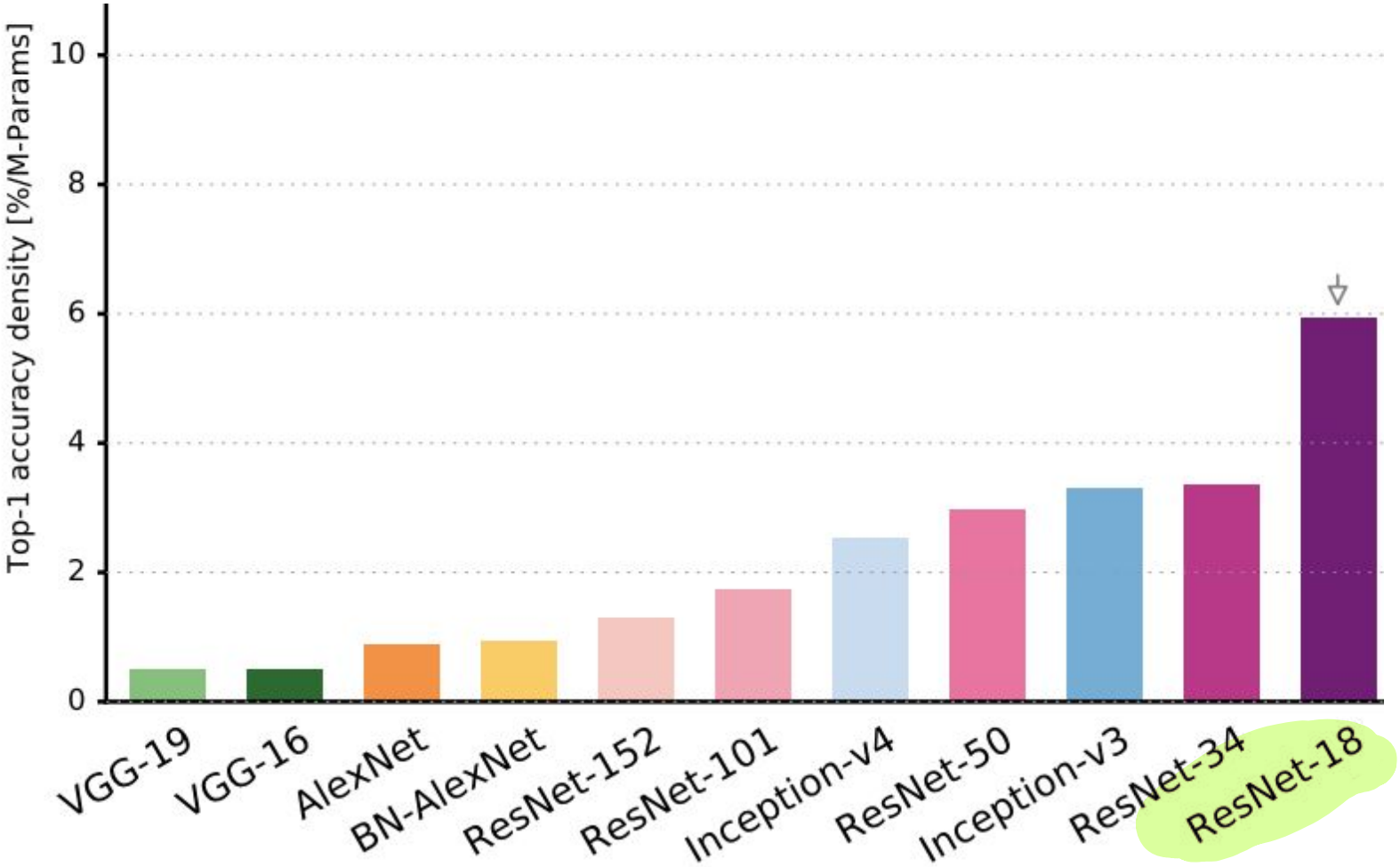


Comparing Complexity





Accuracy per parameter vs. network



Accuracy per parameter vs. network. Information density (accuracy per parameters) is an efficiency metric that highlight that capacity of a specific architecture to better utilise its parametric space.

References

- Canziani, Alfredo, Adam Paszke, and Eugenio Culurciello. "An analysis of deep neural network models for practical applications." (2016).

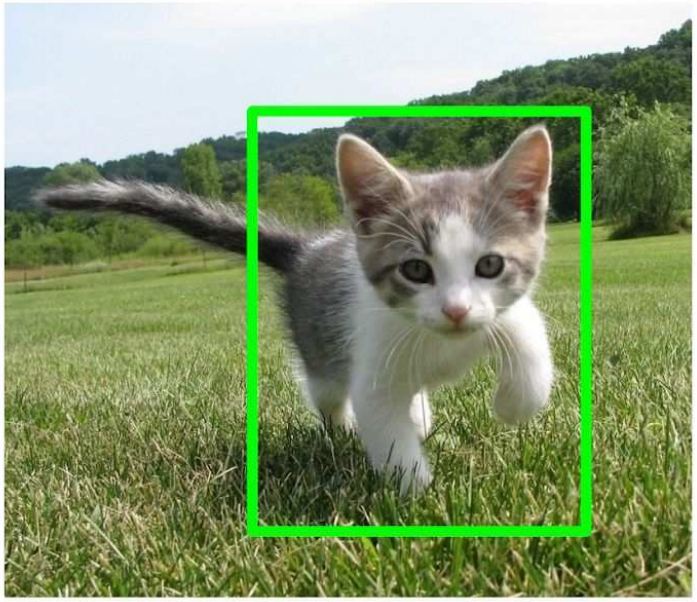
Object Recognition and Face Recognition

Image Classification

(assume given a set of possible labels)
{dog, cat, truck, plane, ...}

cat

Object Detection and Localization



Input Image
(e.g. 3 x 640 x 480)

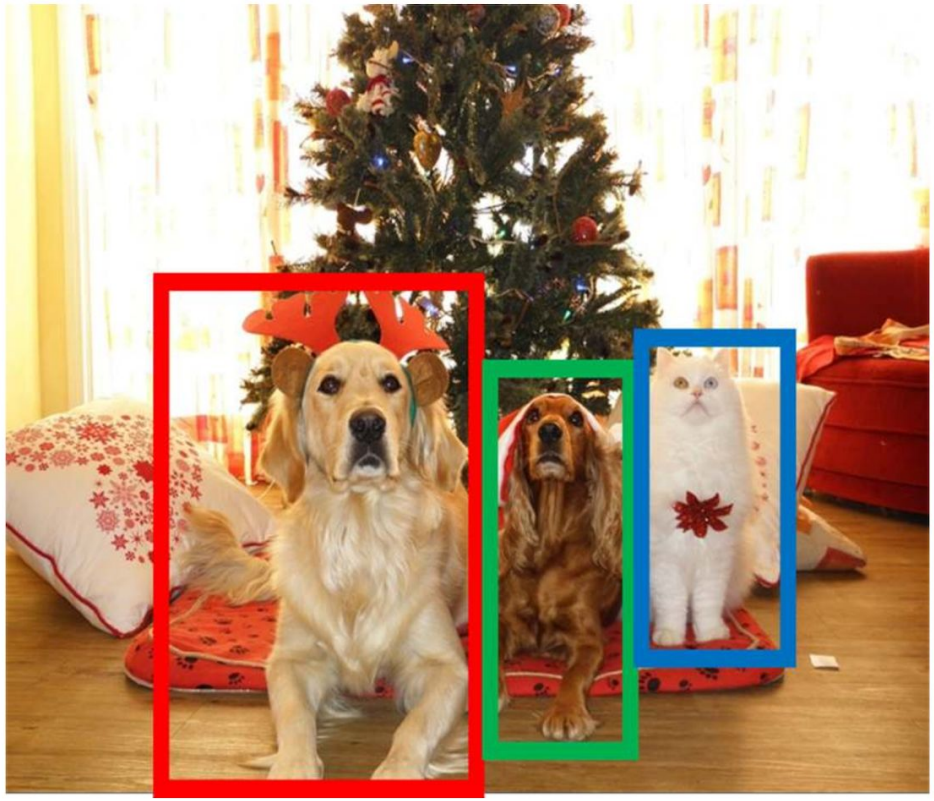
Can we relate Object Detection and Classification?

Object Detection: Task Definition

Input: Single RGB Image

Output: A set of detected objects;
For each object predict:

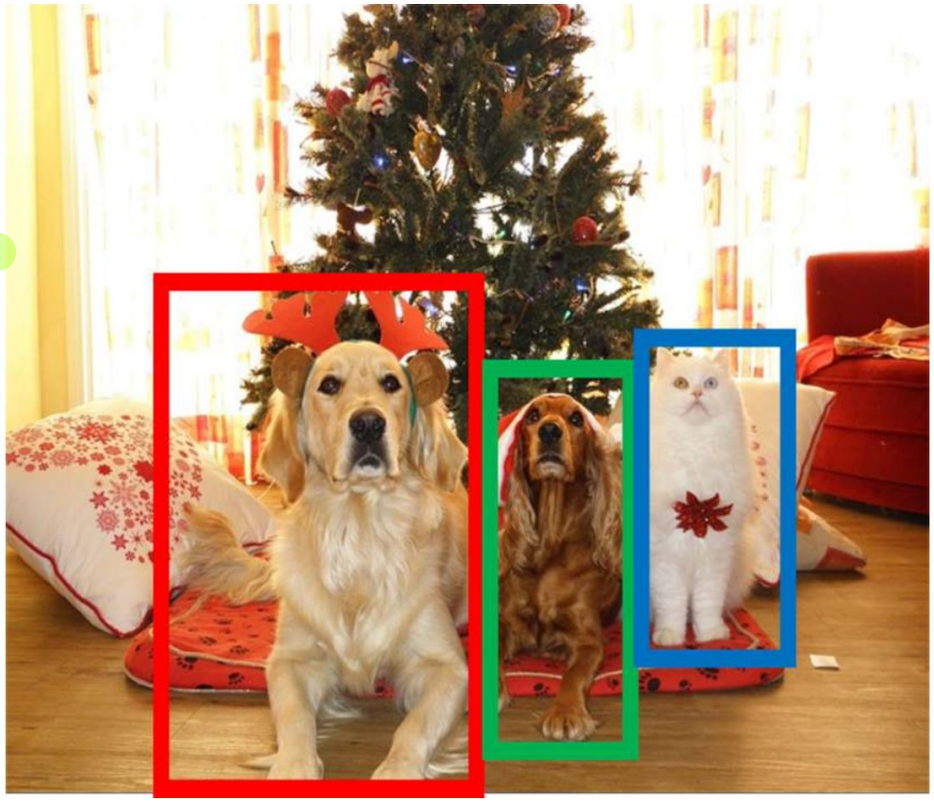
1. Category label (from fixed, known set of categories)
2. Bounding box (four numbers: x, y, width, height)



category and bounding box
i.e x, y, width, height

Object Detection: Challenges

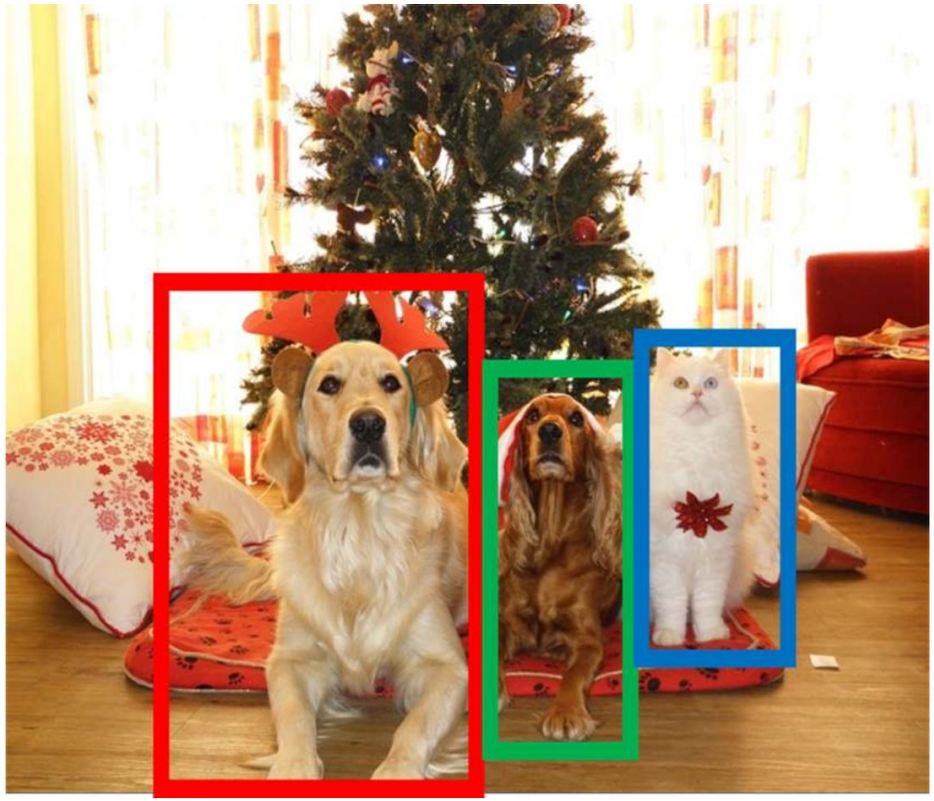
- **Multiple outputs**: Need to output variable numbers of objects per image
- **Multiple types of output**: Need to predict "what" (category label) as well as "where" (bounding box)
- **Large images**: Classification works at 224x224; need higher resolution for detection, often ~800x600



Object Detection: Challenges

- **Multiple outputs**: Need to output variable numbers of objects per image
- **Multiple types of output**: Need to predict ”what” (category label) as well as “where” (bounding box)
- **Large images**: Classification works at 224x224; need higher resolution for detection, often ~800x600

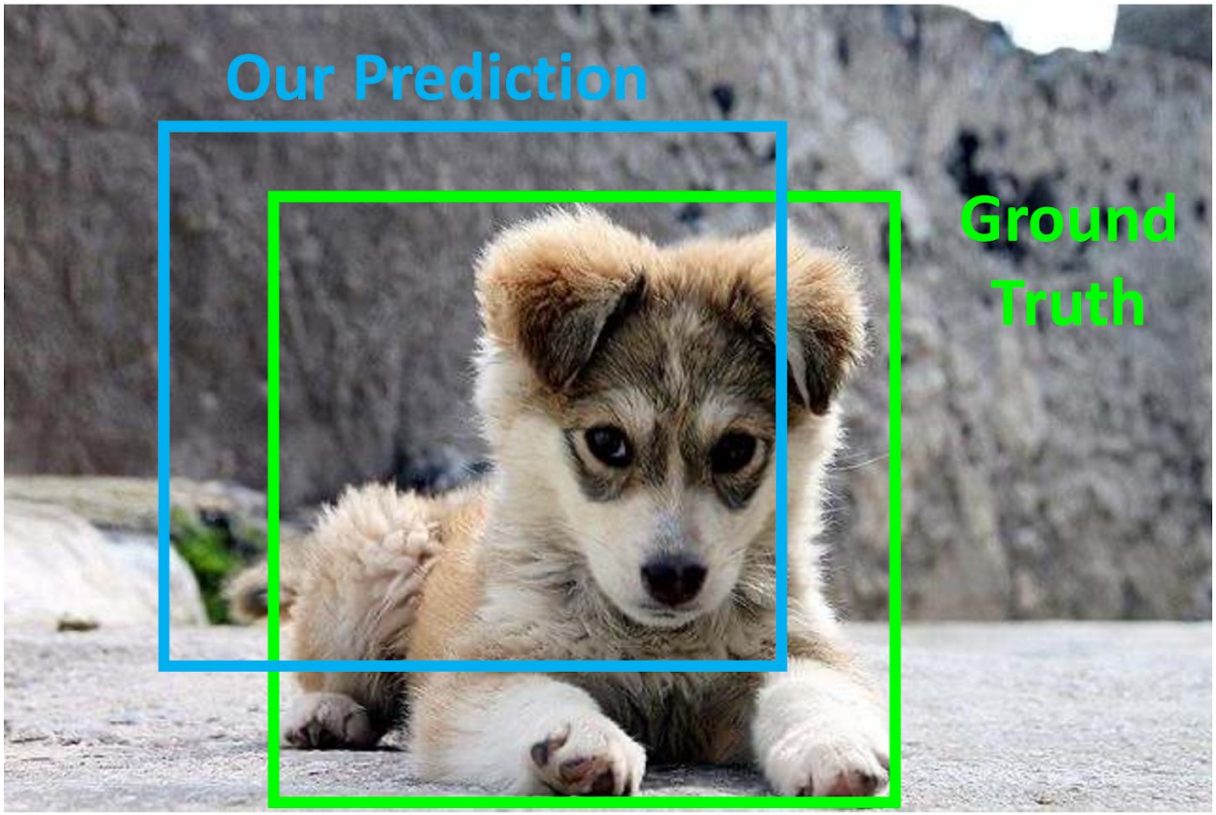
How to verify if the output is correct?



Comparing Boxes: Intersection over Union (IoU)

Jaccard similarity

How can we compare our prediction to the ground-truth box?

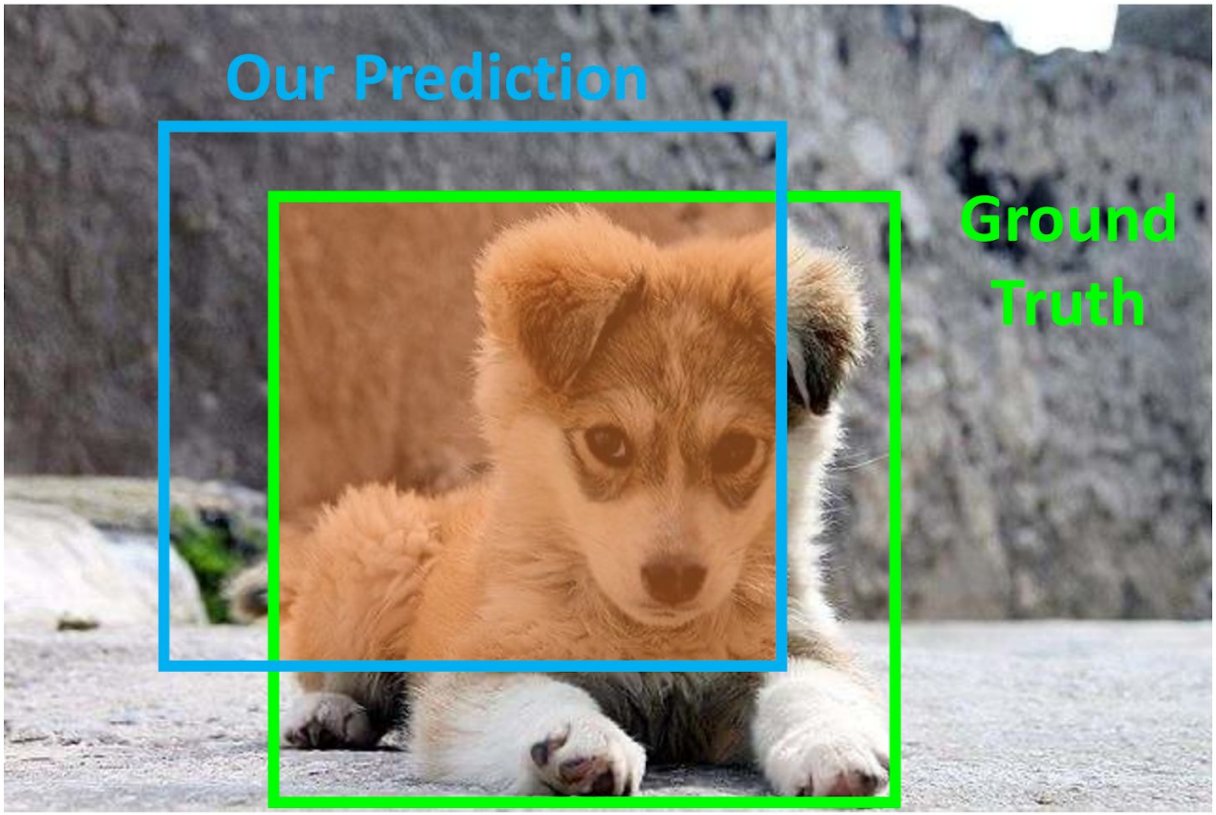


Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU) (Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{\textit{Area of Intersection}}{\textit{Area of Union}}$$

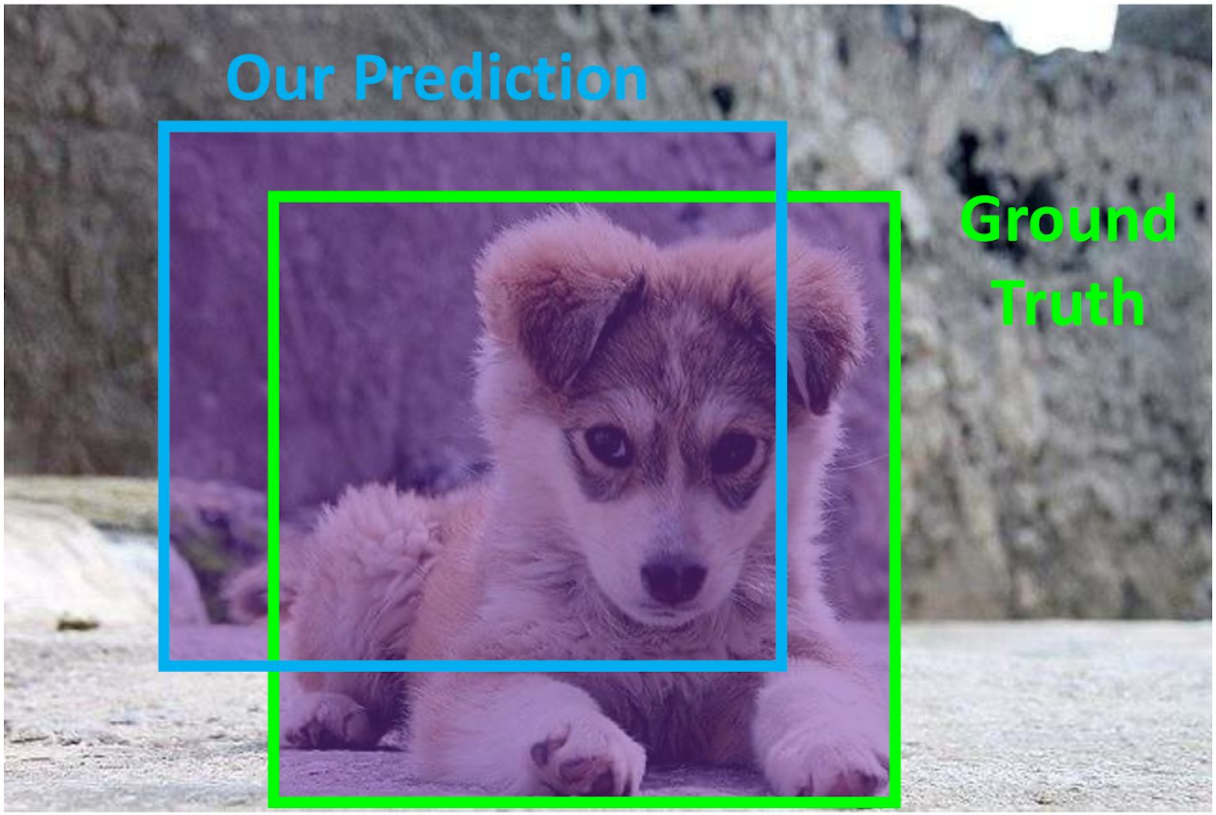


Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU) (Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{Area\ of\ Intersection}{Area\ of\ Union}$$



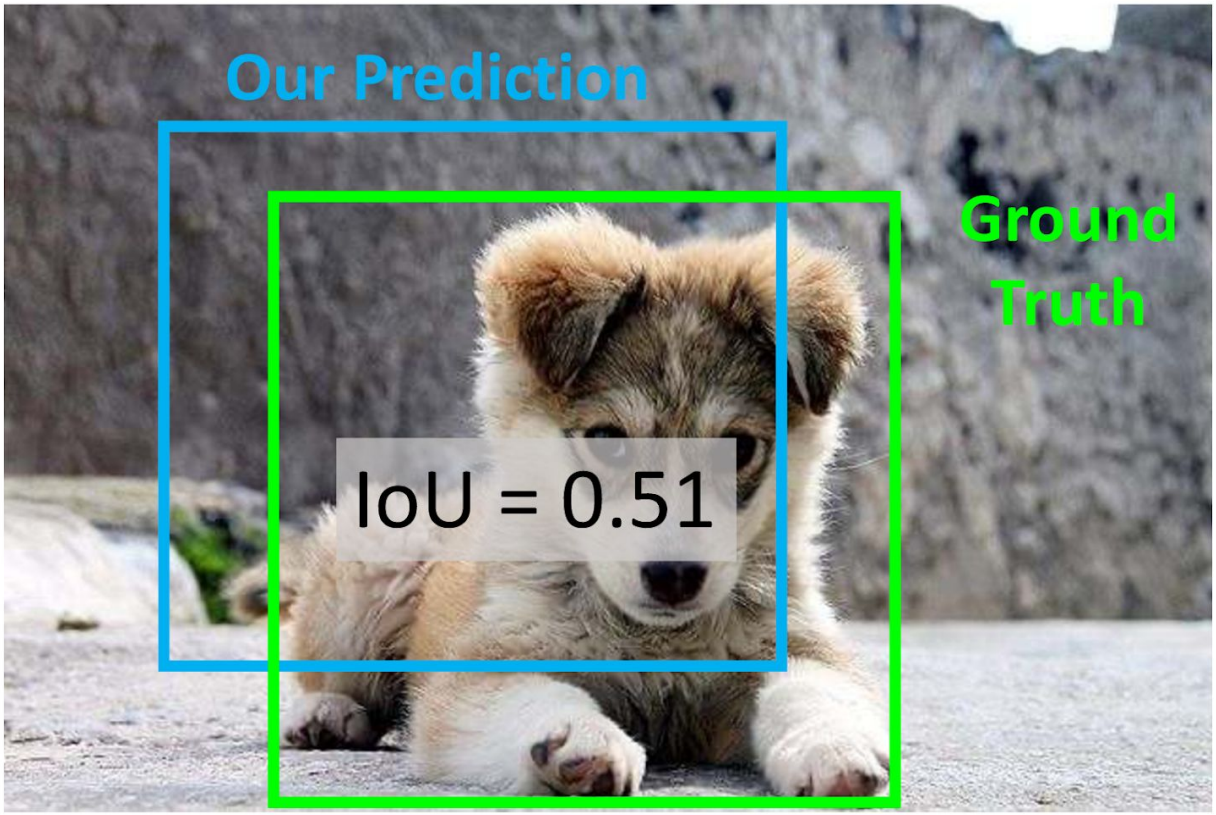
Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU) (Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{Area\ of\ Intersection}{Area\ of\ Union}$$

IoU > 0.5 is "decent"



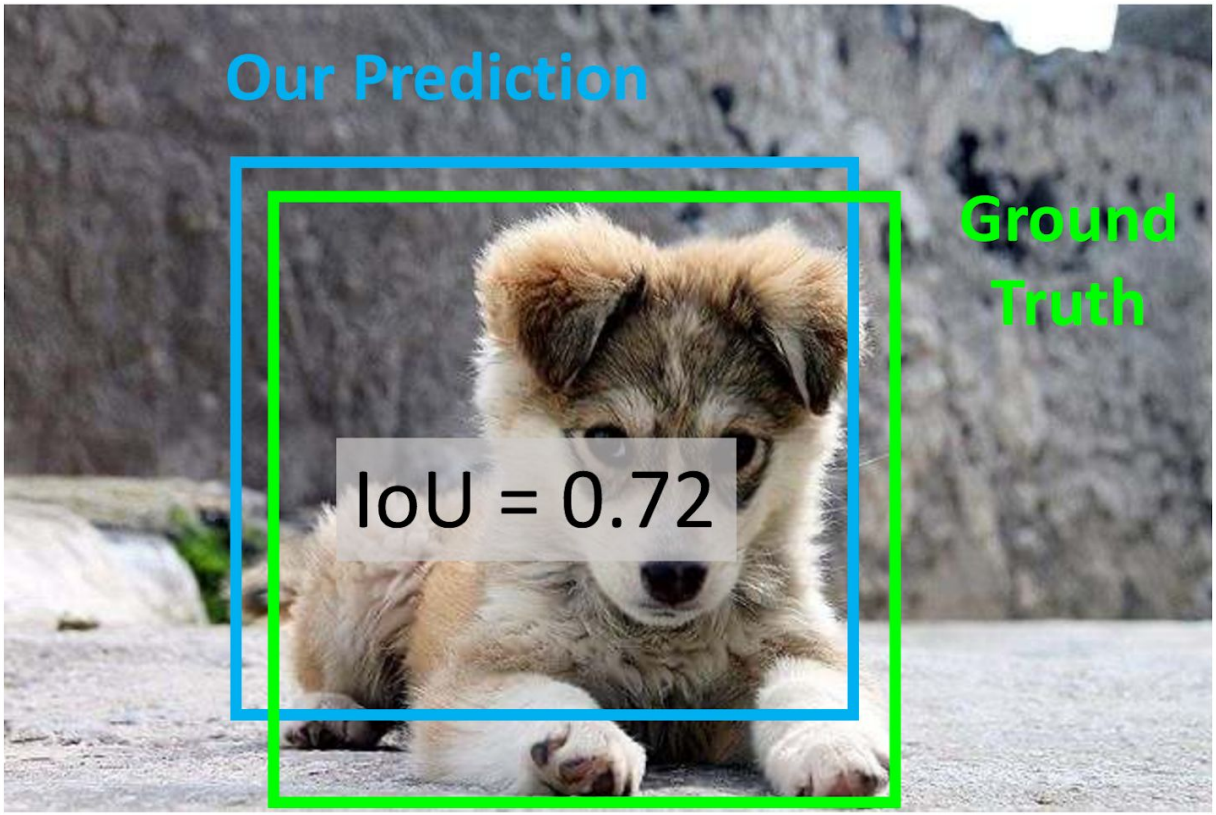
Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU) (Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{\textit{Area of Intersection}}{\textit{Area of Union}}$$

IoU > 0.5 is "decent",
IoU > 0.7 is "pretty good",



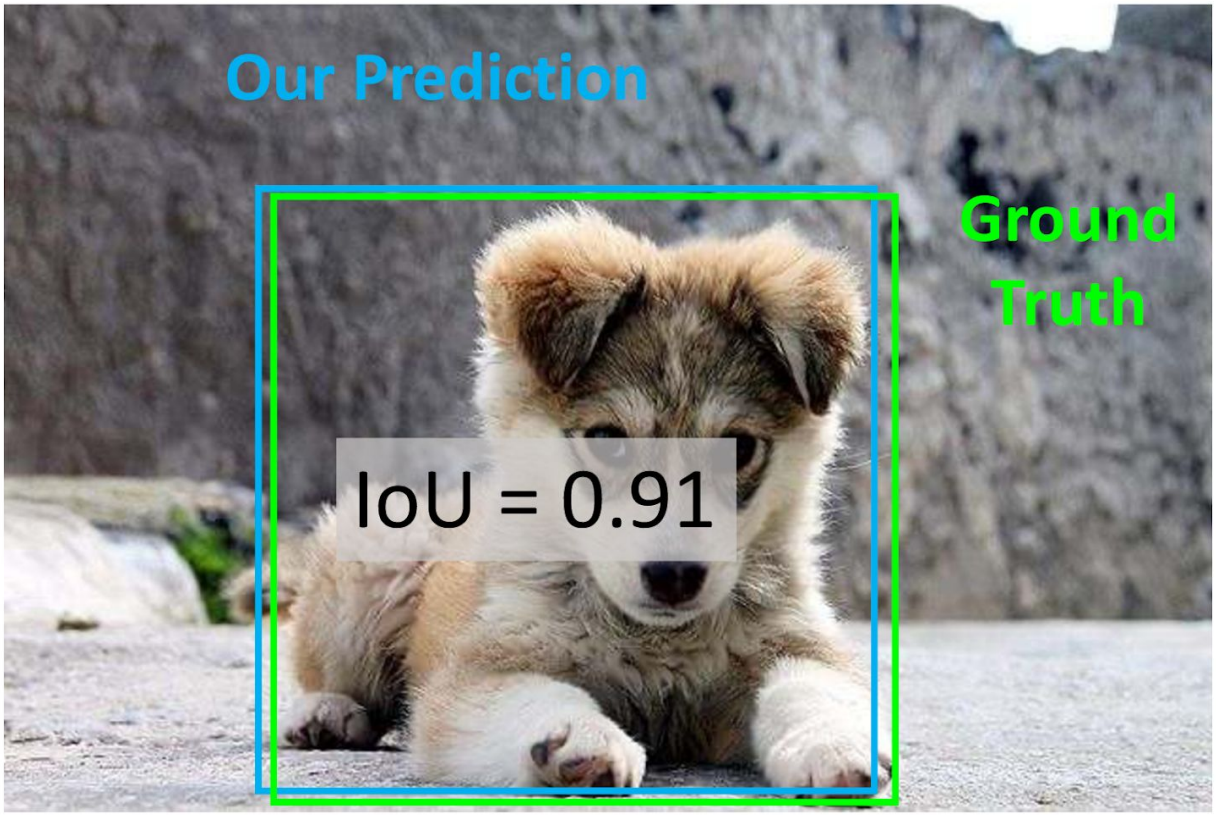
Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU) (Also called "Jaccard similarity" or "Jaccard index"):

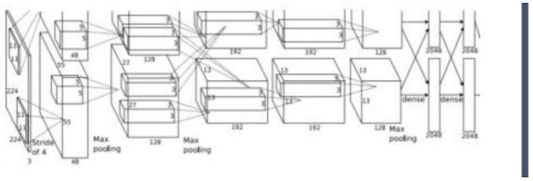
$$\frac{\textit{Area of Intersection}}{\textit{Area of Union}}$$

IoU > 0.5 is "decent",
IoU > 0.7 is "pretty good",
IoU > 0.9 is "almost perfect"



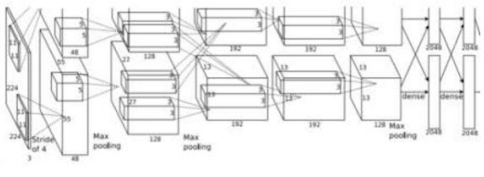
Detecting a single object

Detecting a single object



Vector:
4096

Detecting a single object



Vector:
4096

Fully Connected:
4096 to 1000

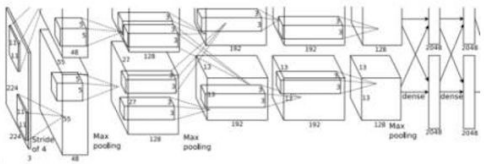
“What”

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label:
Cat

Softmax Loss

Detecting a single object



Treat localization as a regression problem!

Vector:
4096

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax Loss

Correct label:
Cat

"What"

Fully Connected:
4096 to 4

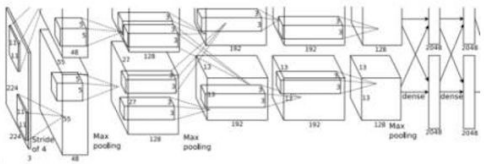
Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x', y', w', h')

"Where"

Detecting a single object



Treat localization as a regression problem!

Vector:
4096

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

"What"

Correct label:
Cat

Softmax Loss

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

"Where"

L2 Loss

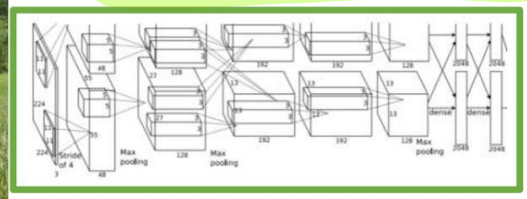
Correct box:
(x', y', w', h')

Multitask Loss

Weighted Sum → **Loss**

$$L = L_{cls} + \lambda L_{reg}$$

Detecting a single object



Often pretrained on ImageNet (Transfer learning)

Treat localization as a regression problem!

Vector: 4096

Fully Connected: 4096 to 1000

Fully Connected: 4096 to 4

"What"

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Box Coordinates
(x, y, w, h)

"Where"

Correct label:
Cat

Softmax Loss

L2 Loss

Correct box:
(x', y', w', h')

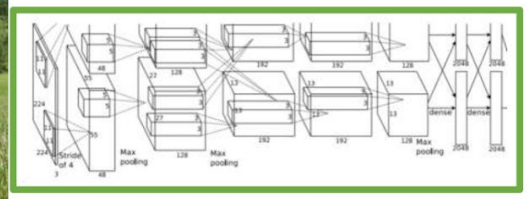
Multitask Loss

Weighted Sum → **Loss**

$$L = L_{cls} + \lambda L_{reg}$$

Detecting a single object

Often pretrained on ImageNet (Transfer learning)



Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

"What"

Correct label: Cat

Softmax Loss

Multitask Loss

Weighted Sum → **Loss**

$$L = L_{cls} + \lambda L_{reg}$$

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

L2 Loss

Correct box: (x', y', w', h')

"Where"

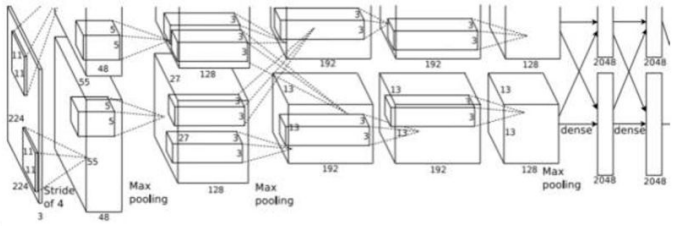
Treat localization as a regression problem!

Problem: Images can have more than one object!

Detecting Multiple Objects

Detecting Multiple Objects

Need different numbers of outputs per image

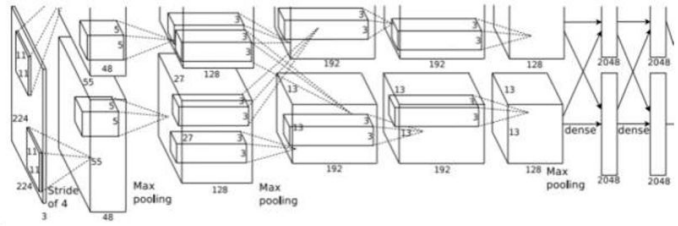


CAT: (x, y, w, h)

4 numbers

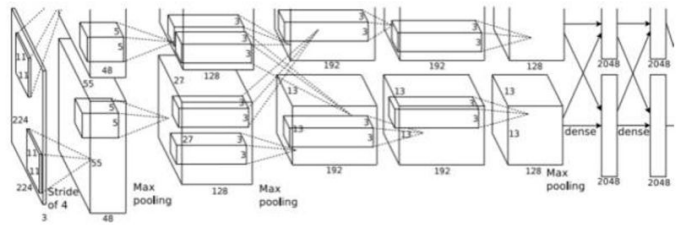
Detecting Multiple Objects

Need different numbers of outputs per image



CAT: (x, y, w, h)

4 numbers

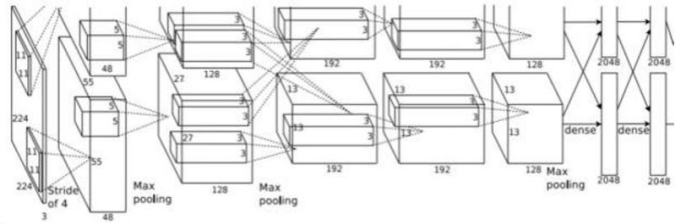


DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

12 numbers

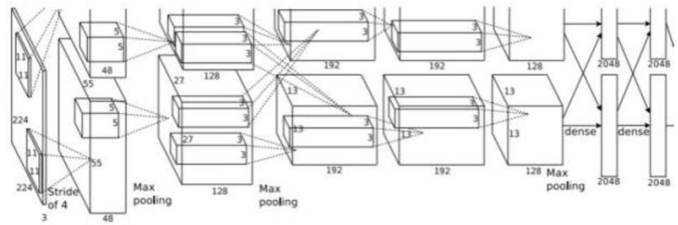
Detecting Multiple Objects

Need different numbers of outputs per image



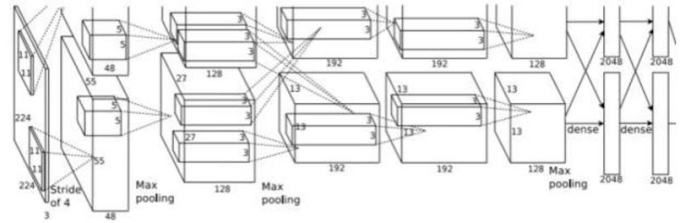
CAT: (x, y, w, h)

4 numbers



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

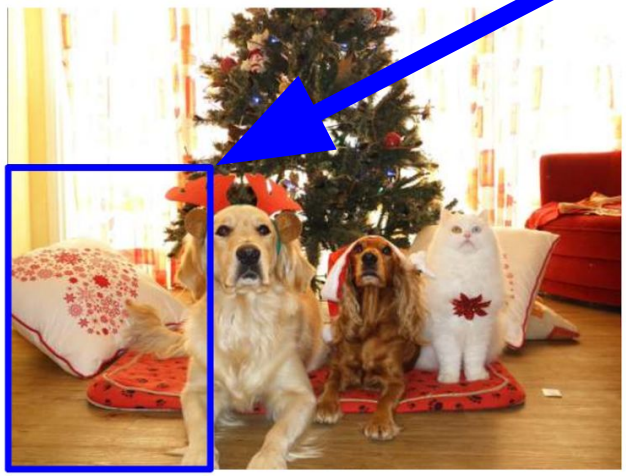
12 numbers



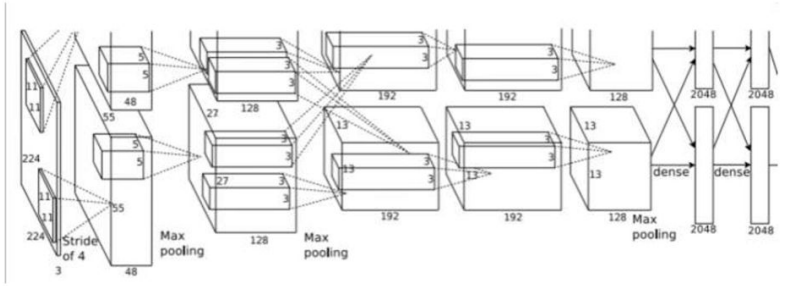
DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Many numbers!

Detecting Multiple Objects: Sliding Window



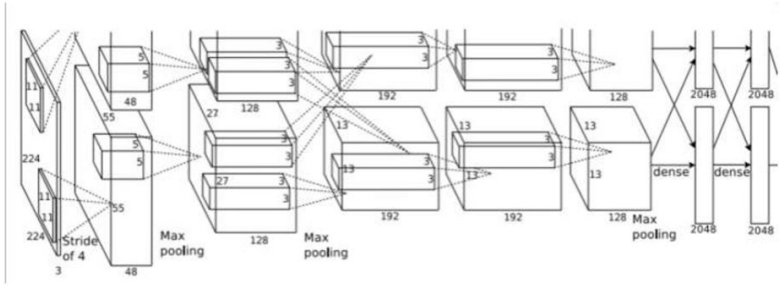
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

Detecting Multiple Objects: Sliding Window

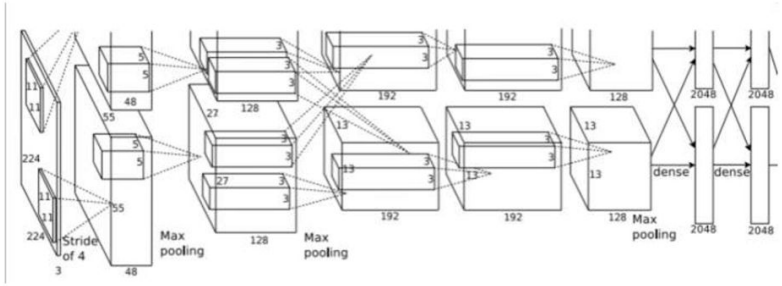
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

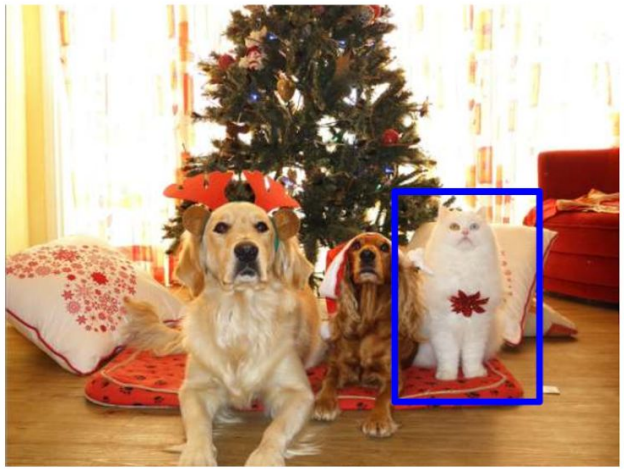
Detecting Multiple Objects: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

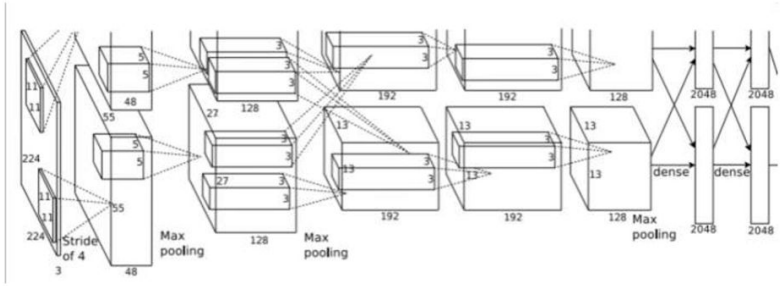


Dog? YES
Cat? NO
Background? NO

Detecting Multiple Objects: Sliding Window

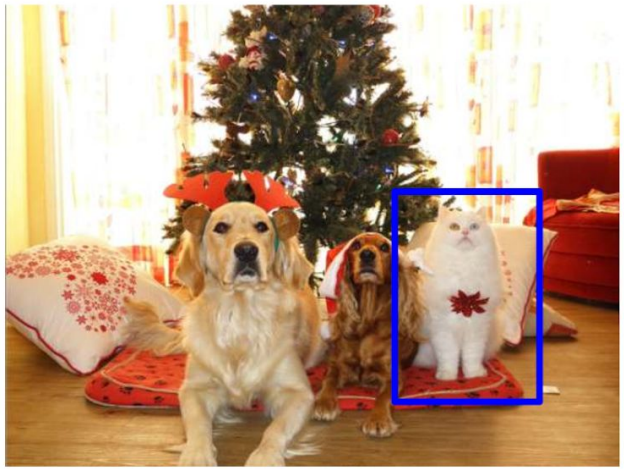


Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? YES
Background? NO

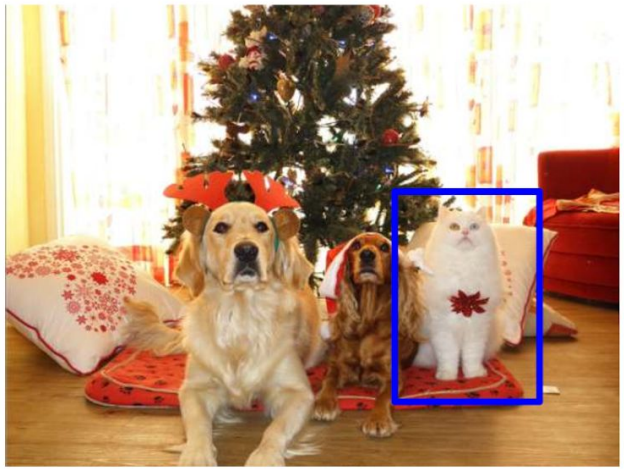
Detecting Multiple Objects: Sliding Window



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size H x W?

Detecting Multiple Objects: Sliding Window

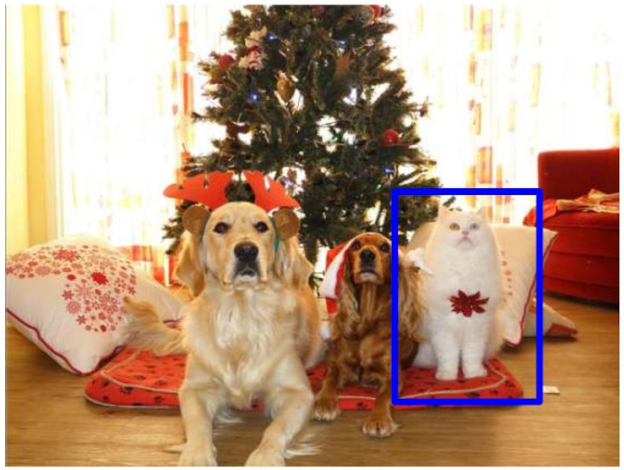


Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size H x W?

Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions:
(W – w + 1) * (H – h + 1)

Detecting Multiple Objects: Sliding Window



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

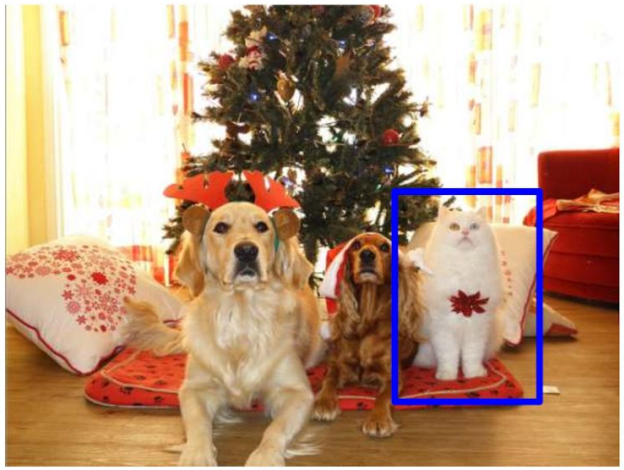
Question: How many possible boxes are there in an image of size H x W?

Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions:
(W – w + 1) * (H – h + 1)

Total possible boxes:

$$\sum_{h=2}^{H}\sum_{w=2}^{W}(W-w+1)(H-h+1)$$

Detecting Multiple Objects: Sliding Window



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

No way we can evaluate them all

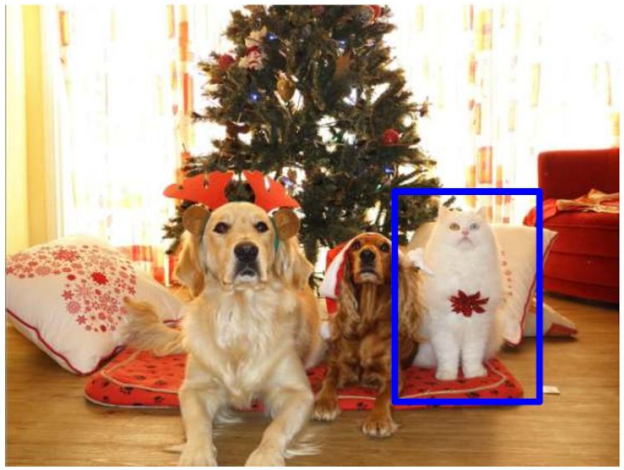
Question: How many possible boxes are there in an image of size H x W?

Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions:
(W – w + 1) * (H – h + 1)

Total possible boxes:

$$\sum_{h=2}^{H} \sum_{w=2}^{W} (W - w + 1)(H - h + 1)$$

Detecting Multiple Objects: Sliding Window



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

No way we can evaluate them all

Question: How many possible boxes are there in an image of size H x W?

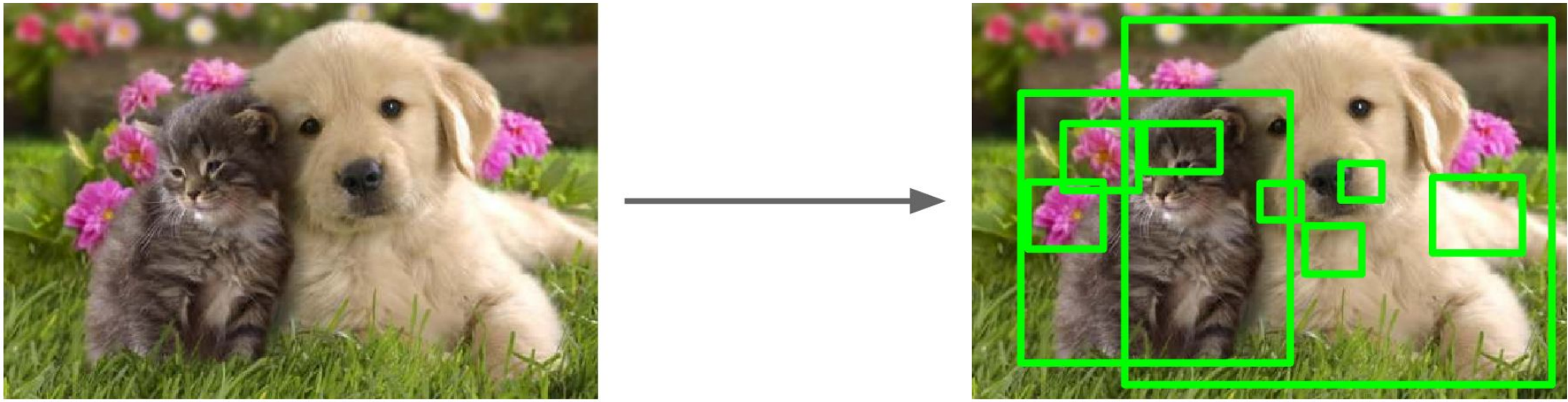
Need to apply CNN to huge number of locations and scales, computationally expensive!

Total possible boxes:

$$\sum_{h=2}^{H} \sum_{w=2}^{W} (W - w + 1)(H - h + 1)$$

Region Proposals

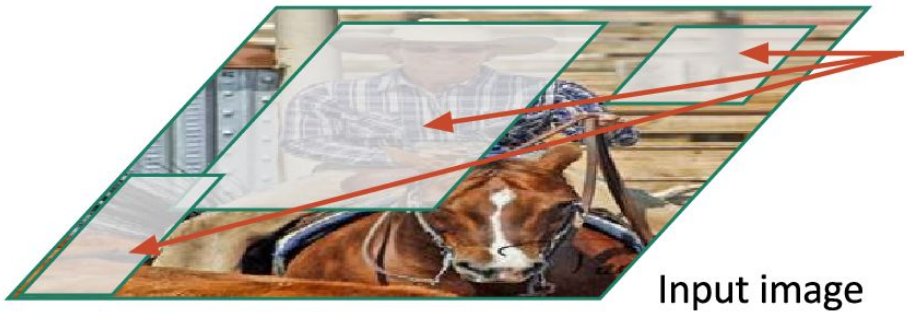
- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 1000 region proposals in a few seconds on CPU



R-CNN

Input image

R-CNN

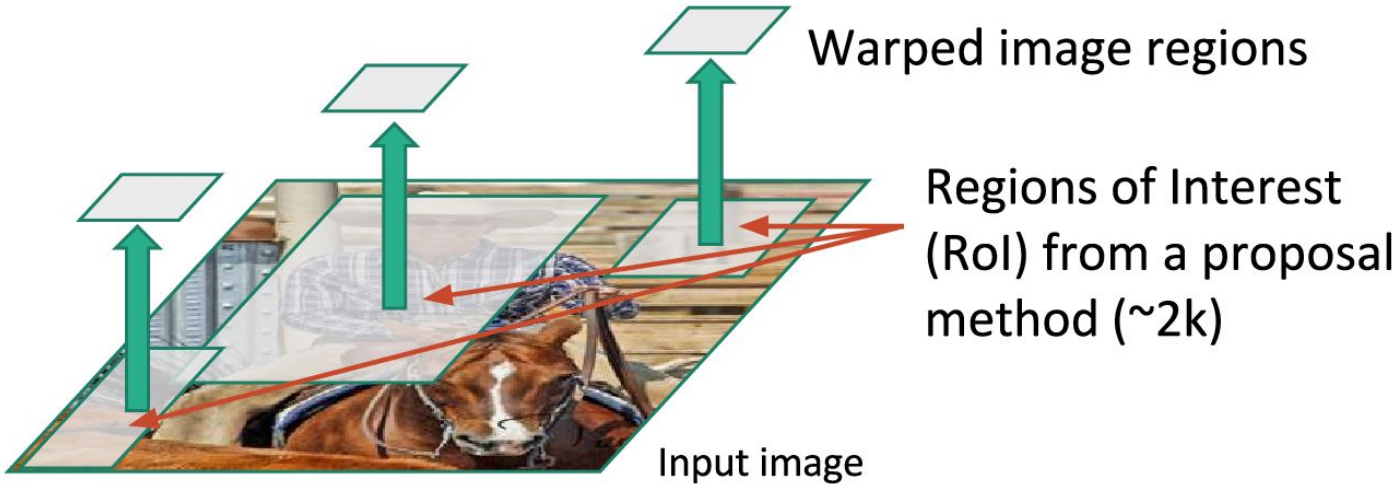


Input image

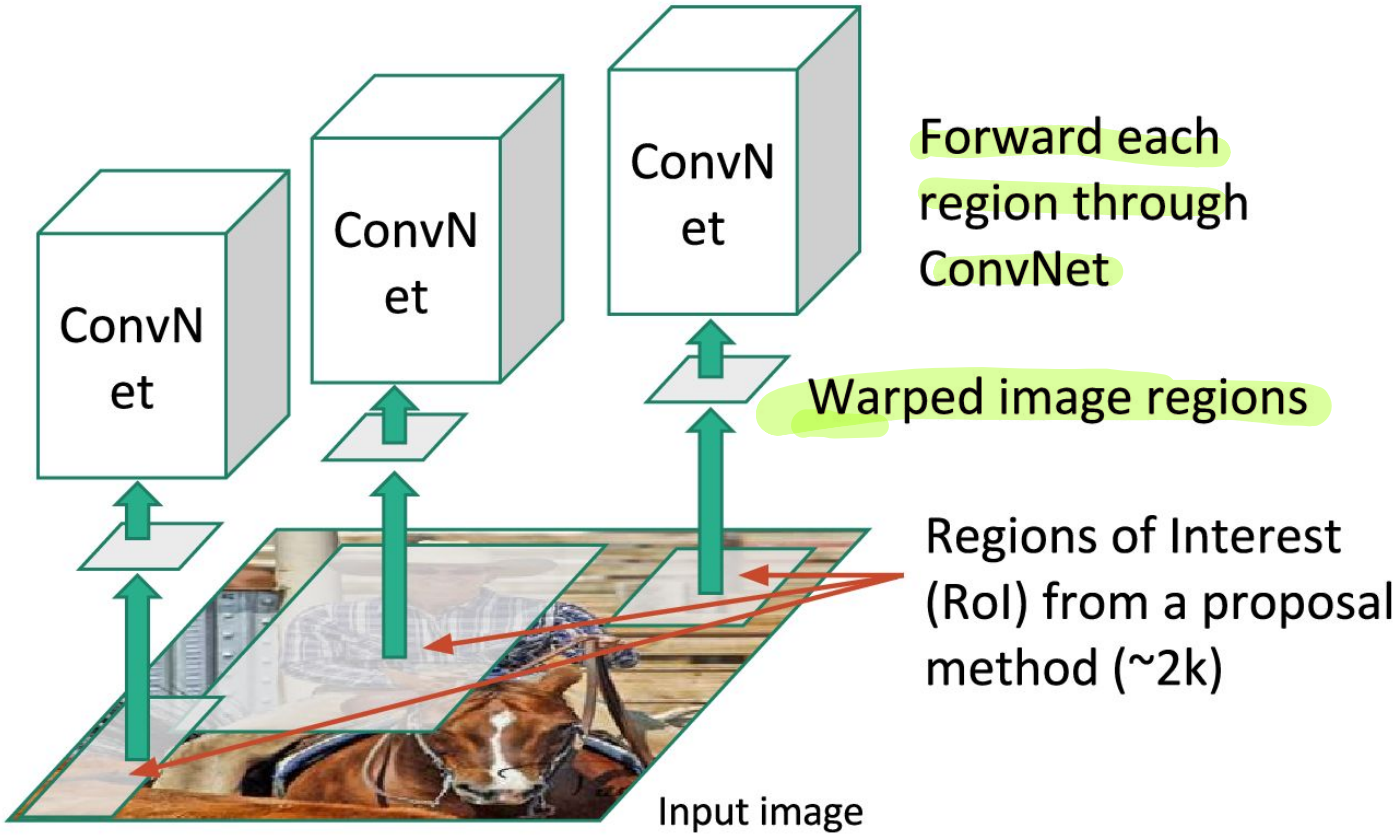
Regions of Interest (RoI) from a proposal method (~2k)

Selective Search.

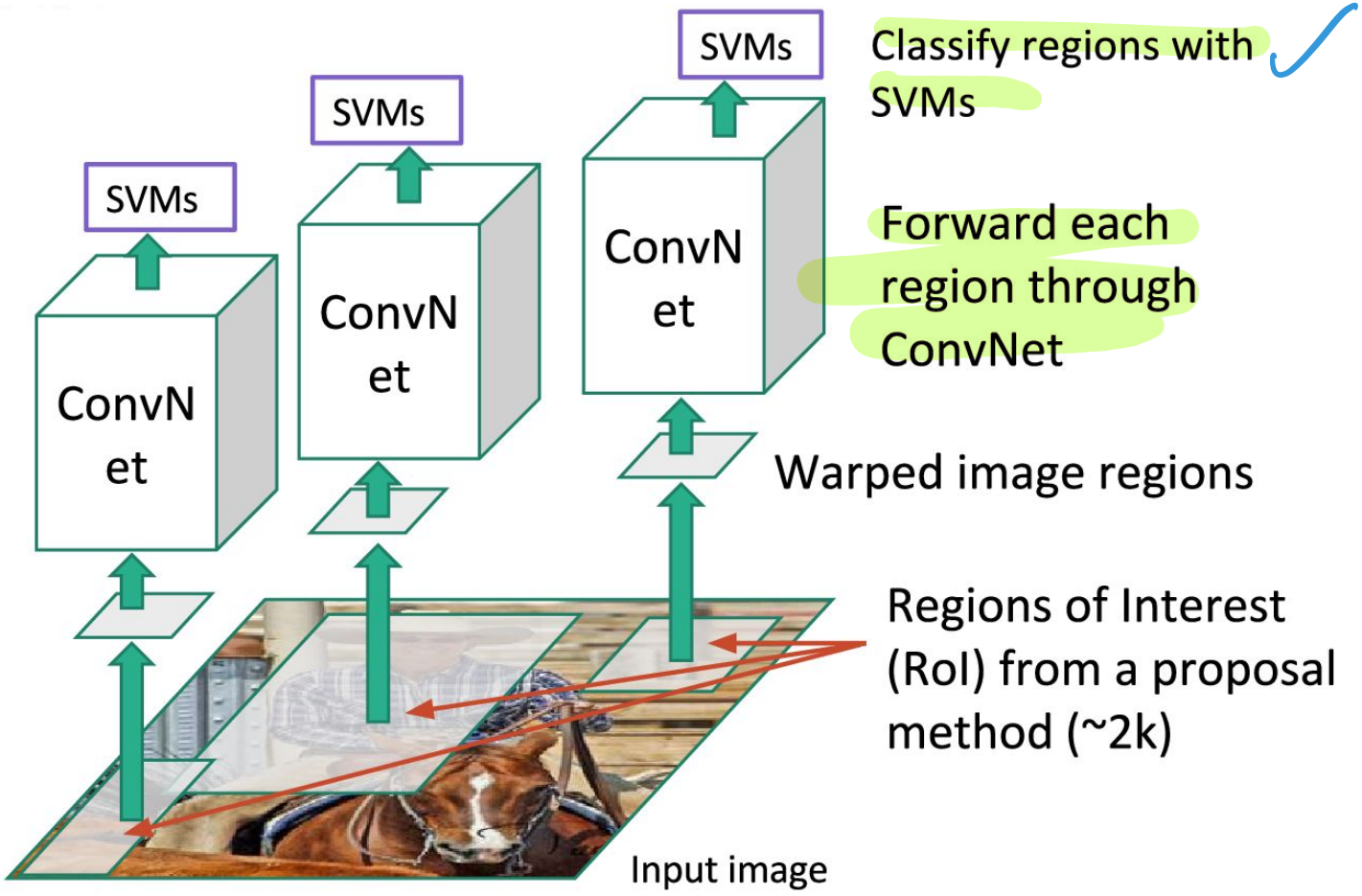
R-CNN


Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

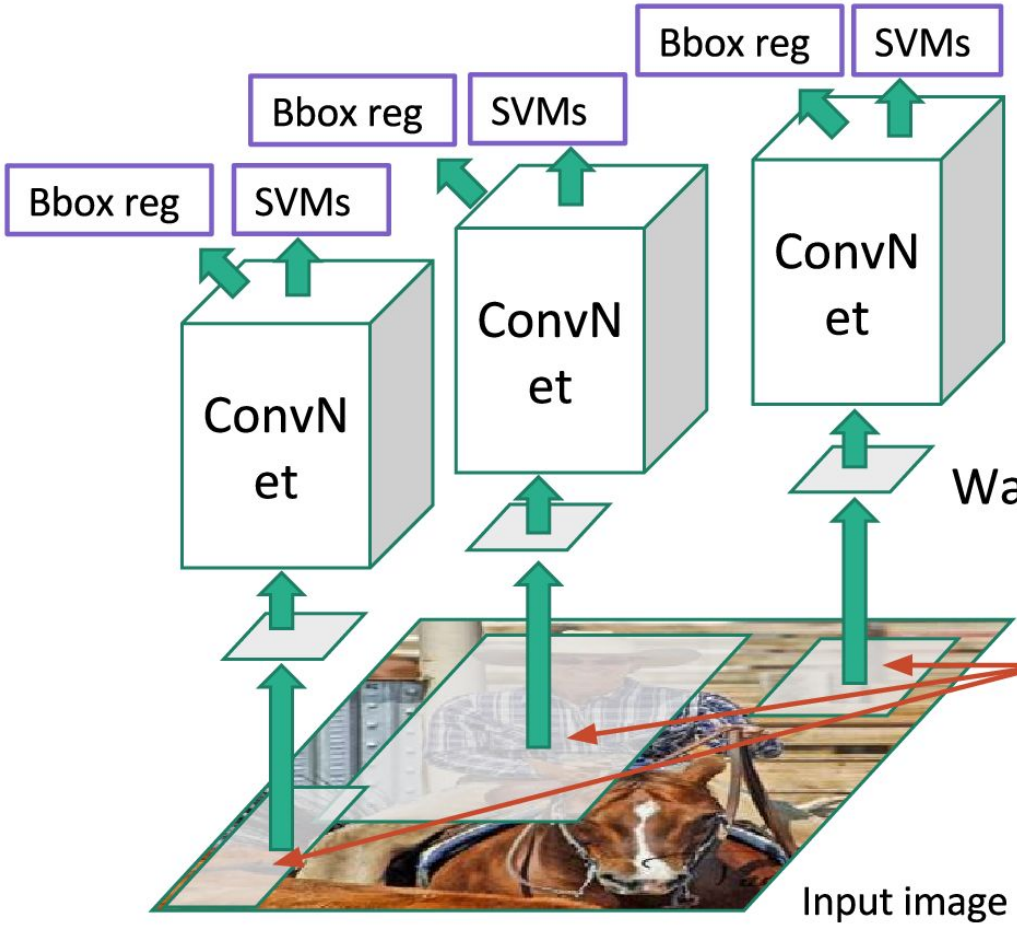
R-CNN


ConvN
et
ConvN
et
ConvN
et
Forward each
region through
ConvNet
Warped image regions
Regions of Interest
(RoI) from a proposal
method (~2k)
Input image

R-CNN


SVMs
SVMs
SVMs
ConvN et
ConvN et
ConvN et
Classify regions with SVMs
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

R-CNN



Linear Regression for bounding box offsets

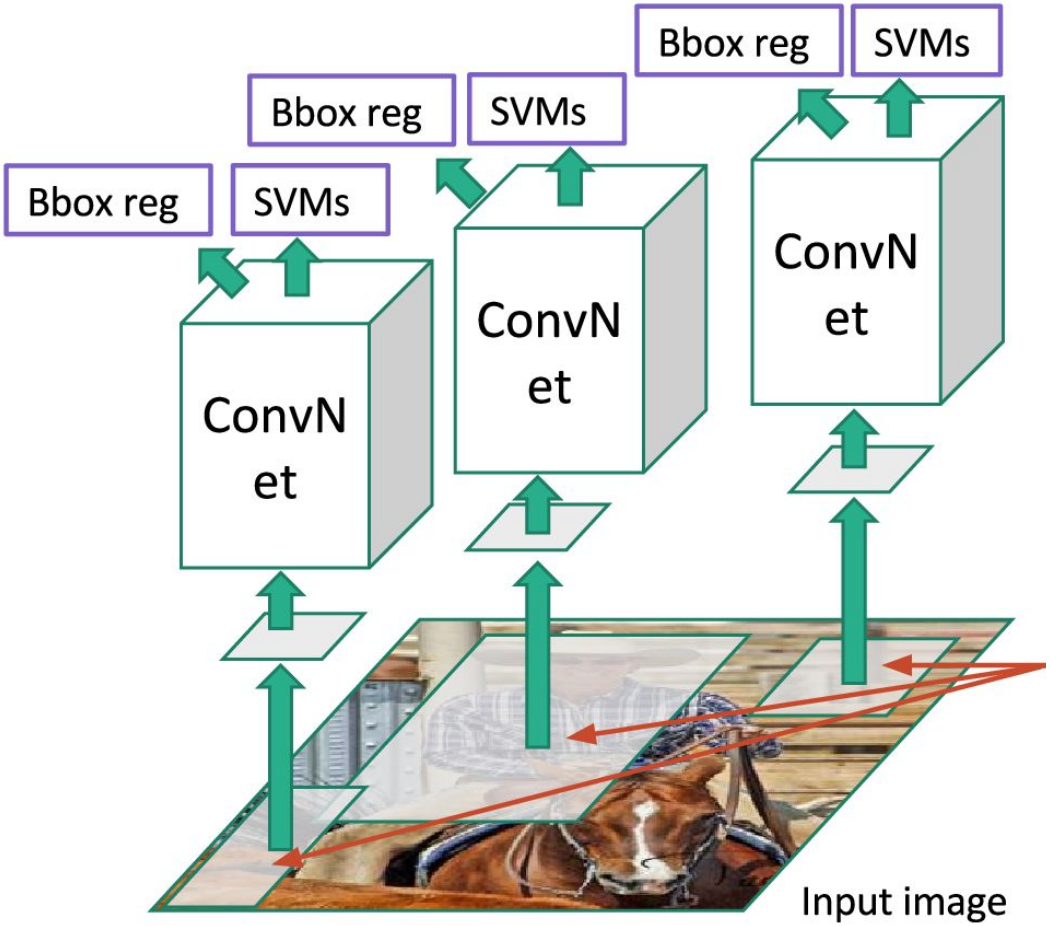
Classify regions with SVMs

Forward each region through ConvNet

Warped image regions

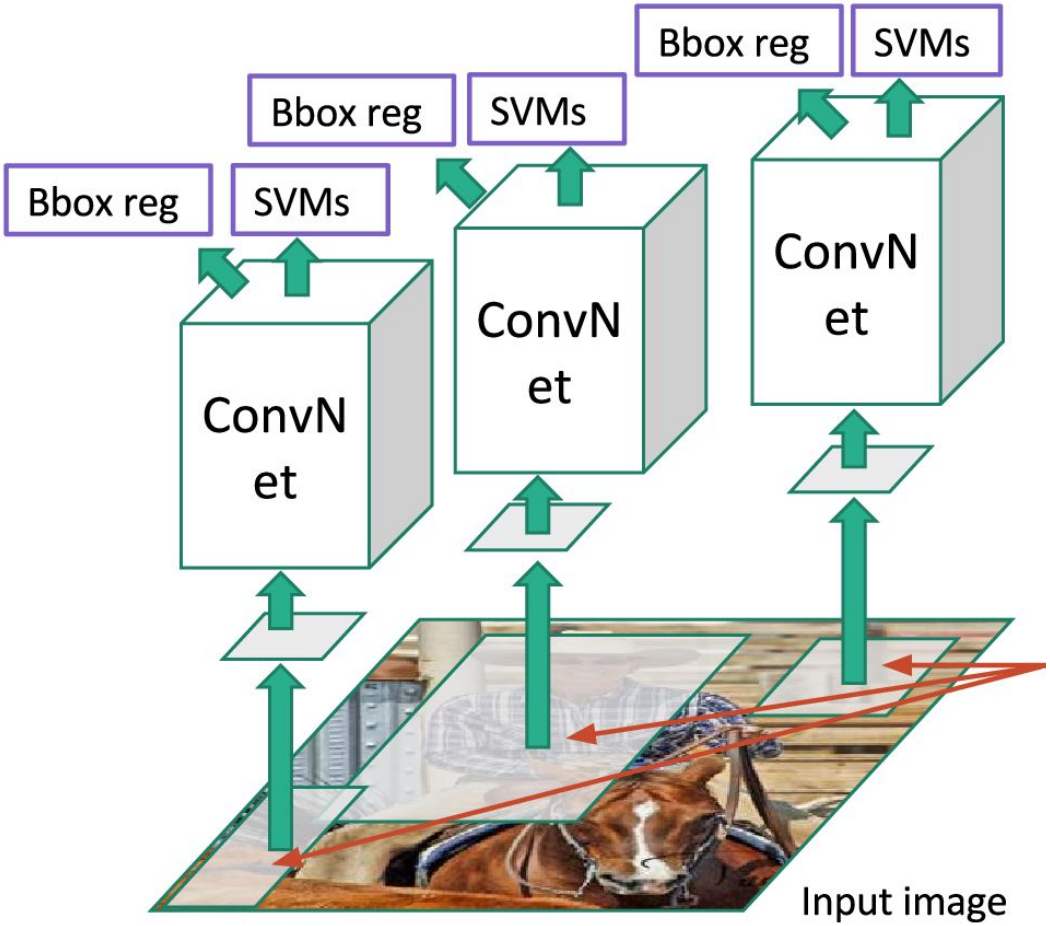
Regions of Interest (RoI) from a proposal method (~2k)

R-CNN



- Training is **slow** (84h), takes a lot of disk space
- Inference (detection) is slow

R-CNN



- Training is **slow** (84h), takes a lot of disk space
- Inference (detection) is slow

***Idea**: Pass the image through convnet before cropping! Crop the conv feature instead!*

References

- Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
- Girshick, Ross. "Fast R-CNN." Proceedings of the IEEE international conference on computer vision. 2015.
- Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).