CNN Applications

Deep Learning

Brad Quinton, Scott Chin

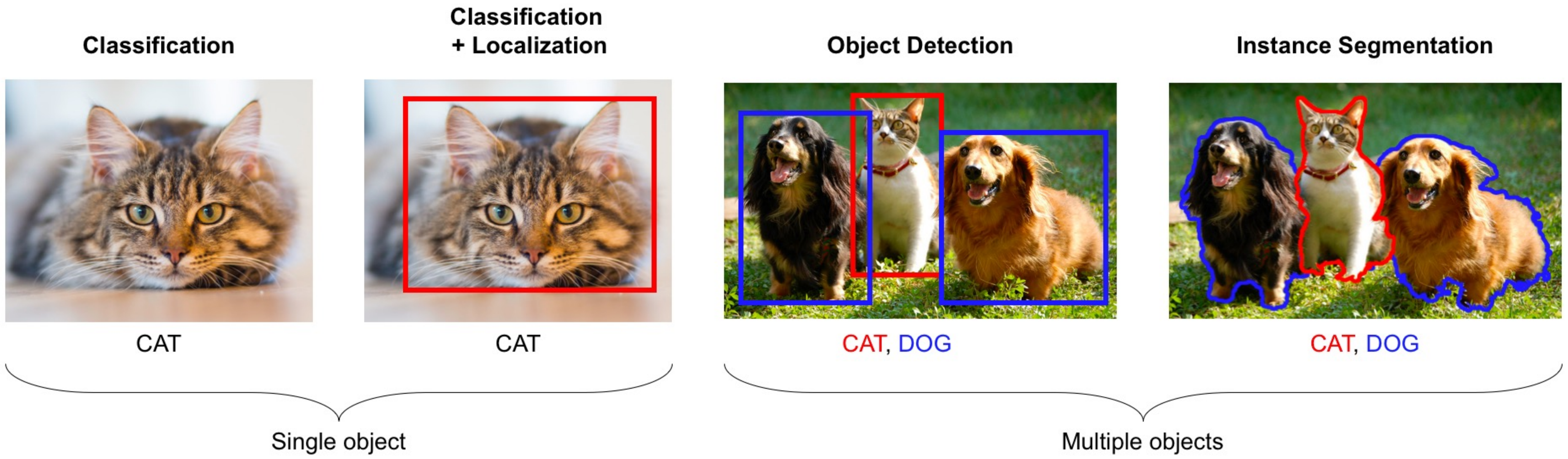
Learning Objectives

- Introduction to Object detection
- Sliding window via convolution
- Quick introduction to other vision applications beyond classification
  - Localization
  - Landmark detection
  - Face detection
  - Pose detection
  - Image retrieval
  - Visualization
  - Segmentation

Brad Quinton, Scott Chin

Object Localization and Detection

What is Localization and Detection



Brad Quinton, Scott Chin

Localization

Output:

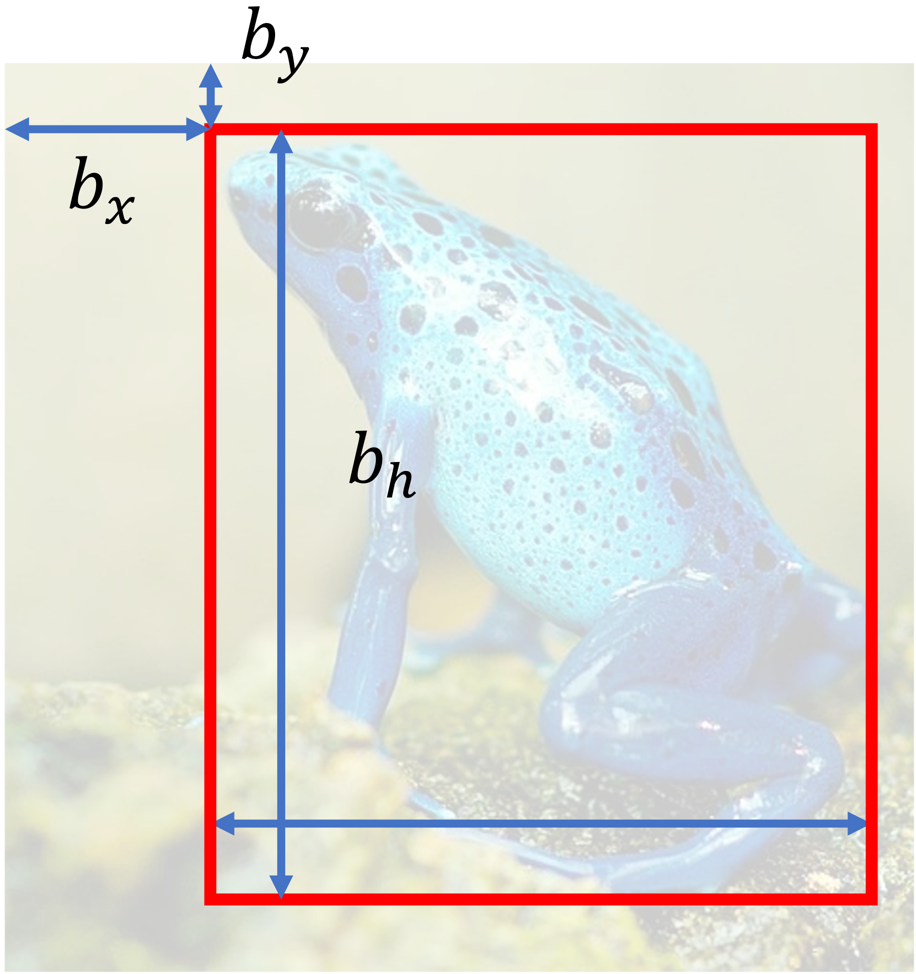
- Class prediction
- Bounding box b_x, b_y, b_w, b_h

Brad Quinton, Scott Chin

Localization

Output:

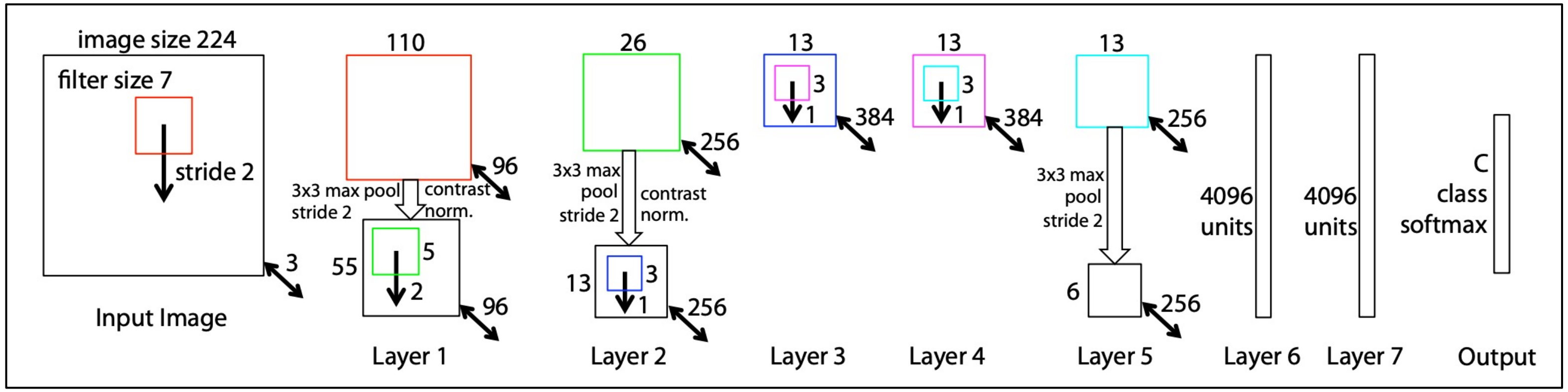
- Class prediction
- Bounding box b_x, b_y, b_w, b_h



Brad Quinton, Scott Chin

Start with a CNN Classifier Architecture

A CNN Classifier (e.g. ZFNet)



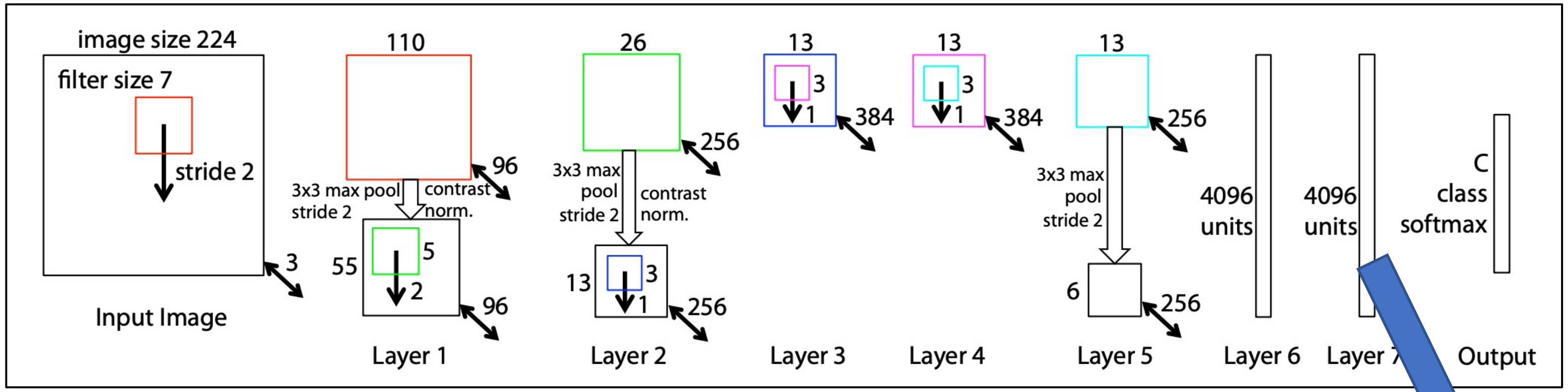
Class Prediction

$\hat{y}$

Image source: Davi Frossard https://www.cs.toronto.edu/~frossard/post/vgg16/

Start with a CNN Classifier Architecture

A CNN Classifier (e.g. ZFNet)



Class Prediction

$\hat{y}$

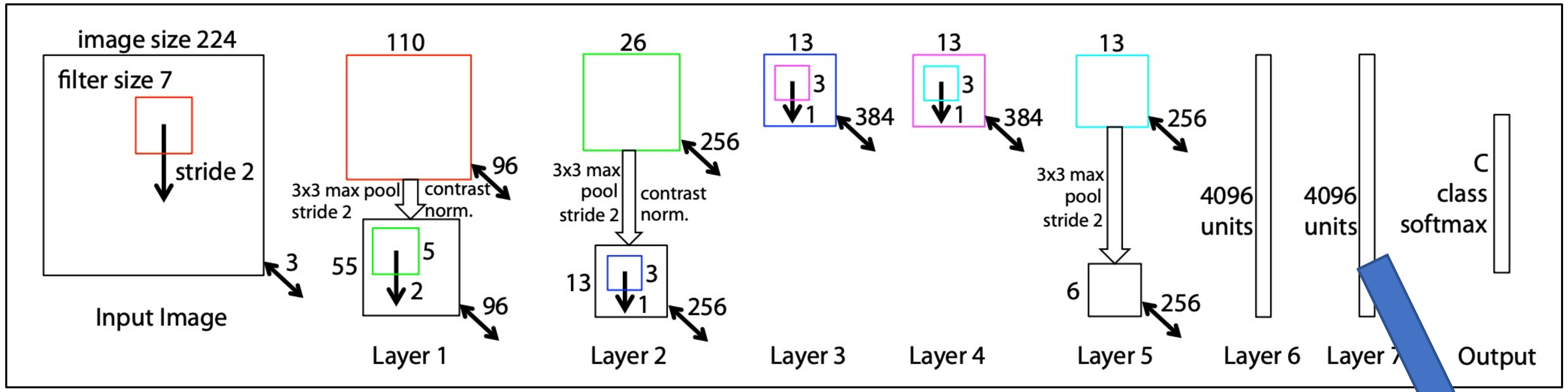
4096 Vector

Image source: Davi Frossard https://www.cs.toronto.edu/~frossard/post/vgg16/

Brad Quinton, Scott Chin

Add FC layer to predict bounding box

A CNN Classifier (e.g. ZFNet)

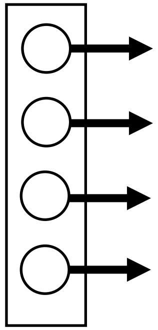


Class Prediction

$\hat{y}$

Bounding Box Prediction

4096 Vector



$\widehat{b_x}$

$\widehat{b_y}$

$\widehat{b_h}$

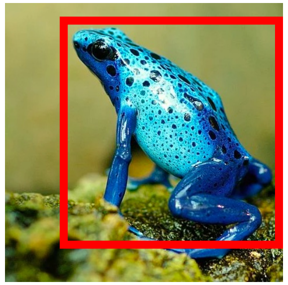
$\widehat{b_w}$

4 Unit FC Layer

How to train this?

Image source: Davi Frossard https://www.cs.toronto.edu/~frossard/post/vgg16/

Training for Localization



CNN
Classifier with
BB output

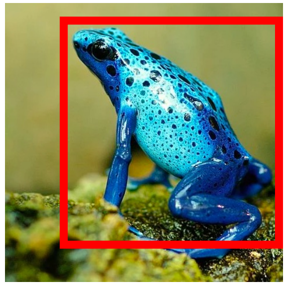
Class Prediction
(Softmax Output)

$\hat{y}$

$\widehat{b_x}$
$\widehat{b_y}$
$\widehat{b_h}$
$\widehat{b_w}$

Bounding Box Prediction
(Real Numbers)

Training for Localization



CNN
Classifier with
BB output

Class Prediction
(Softmax Output)

$\hat{y}$

Categorical Cross
Entropy Loss

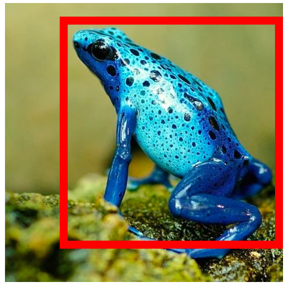
Cost =
Average Loss

$\widehat{b_x}$
$\widehat{b_y}$
$\widehat{b_h}$
$\widehat{b_w}$

Bounding Box Prediction
(Real Numbers)

Brad Quinton, Scott Chin

Training for Localization



CNN Classifier with BB output

Class Prediction
(Softmax Output)

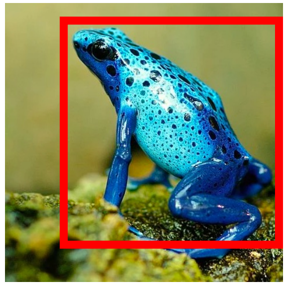
$\hat{y}$ → Categorical Cross Entropy Loss → Cost = Average Loss

$\widehat{b_x}$
$\widehat{b_y}$
$\widehat{b_h}$
$\widehat{b_w}$
→ ????

Bounding Box Prediction
(Real Numbers)

Brad Quinton, Scott Chin

Training for Localization



CNN Classifier with BB output

Class Prediction (Softmax Output)

$\hat{y}$

Categorical Cross Entropy Loss

Cost = Average Loss

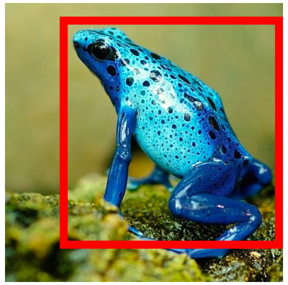
$$\widehat{b_x}$$
$$\widehat{b_y}$$
$$\widehat{b_h}$$
$$\widehat{b_w}$$

????

Bounding Box Prediction (Real Numbers)

Treat BB prediction regression problem

Training for Localization



CNN
Classifier with
BB output

Class Prediction
(Softmax Output)

$\hat{y}$

Categorical Cross
Entropy Loss

Cost =
Average Loss

$\widehat{b_x}$
$\widehat{b_y}$
$\widehat{b_h}$
$\widehat{b_w}$

Bounding Box Prediction
(Real Numbers)

Squared Loss
(i.e. L_2 loss)

Cost =
Average Loss
With L_2 loss, has special name
Mean Squared Error (MSE)

Treat BB prediction regression problem

Brad Quinton, Scott Chin

Aside - Squared Loss (aka L2 Loss)

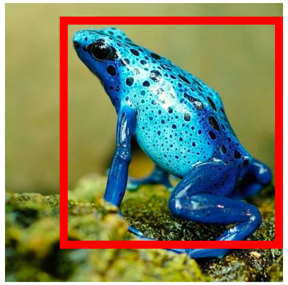
Square of the difference between prediction and true values

Example (BB prediction for One sample):

$$L\left(b_x, b_y, b_w, b_h, \widehat{b_x}, \widehat{b_y}, \widehat{b_w}, \widehat{b_h}\right) =$$
$$(b_x - \widehat{b_x})^2 +$$
$$(b_y - \widehat{b_y})^2 +$$
$$(b_h - \widehat{b_h})^2 +$$
$$(b_w - \widehat{b_w})^2$$

Brad Quinton, Scott Chin

Training for Localization



CNN
Classifier with
BB output

Class Prediction
(Softmax Output)

$\hat{y}$

Categorical Cross
Entropy Loss

Cost =
Average Loss

$\widehat{b_x}$
$\widehat{b_y}$
$\widehat{b_h}$
$\widehat{b_w}$

Bounding Box Prediction
(Real Numbers)

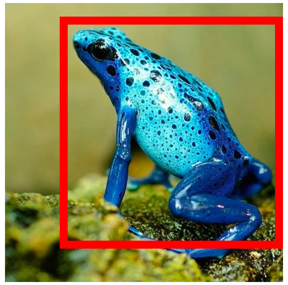
Squared Loss
(i.e. L_2 loss)

Cost =
Average Loss

With L_2 loss, has special name
Mean Squared Error (MSE)

Treat BB prediction regression problem

Training for Localization



CNN
Classifier with
BB output

Class Prediction
(Softmax Output)

$\hat{y}$

Categorical Cross
Entropy Loss

Cost =
Average Loss

Need to normalize or weight
the two cost components

+

Final
Cost

$\widehat{b_x}$
$\widehat{b_y}$
$\widehat{b_h}$
$\widehat{b_w}$

Bounding Box Prediction
(Real Numbers)

Squared Loss
(i.e. L_2 loss)

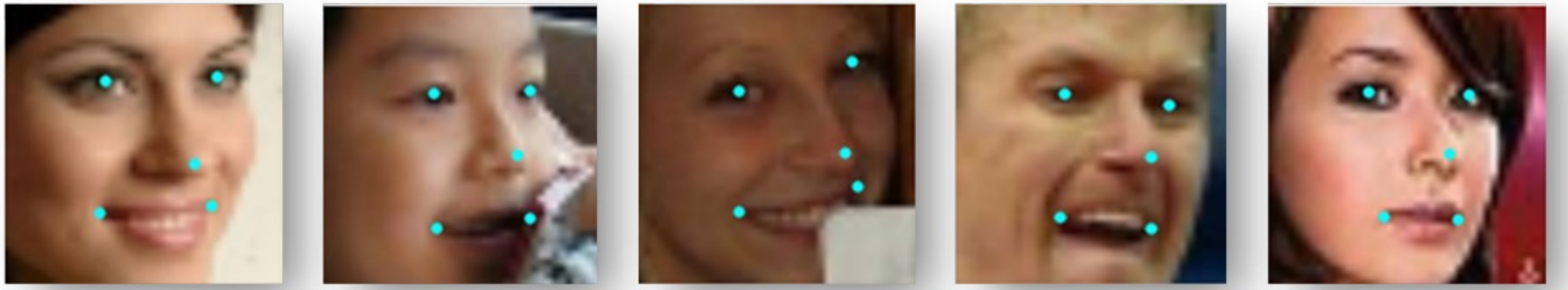
Cost =
Average Loss

With L_2 loss, has special name
Mean Squared Error (MSE)

Treat BB prediction
regression problem

Landmark Detection

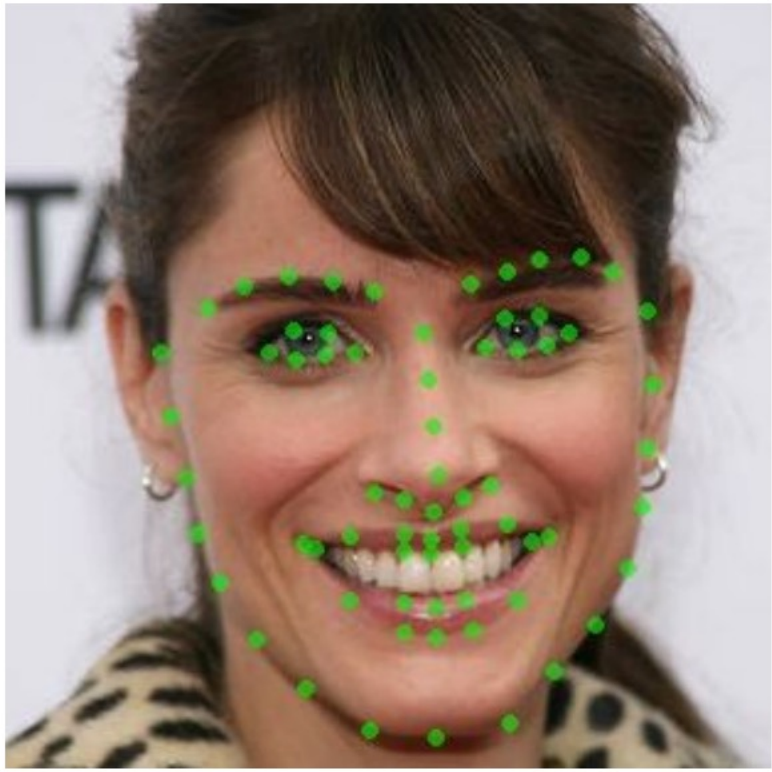
- In Localization, we want to output x, y coordinates (along with h, w) of a bounding box
- In other applications, you may want to output the (x,y) coordinate of several special locations (a.k.a. landmarks)



 "Facial Landmark Detection with Tweaked Convolutional Neural Networks", Wu et al, 2015, https://arxiv.org/abs/1511.04031 Brad Quinton, Scott Chin

Face Detection

- FC layer predicts two numbers (x,y) for each landmark



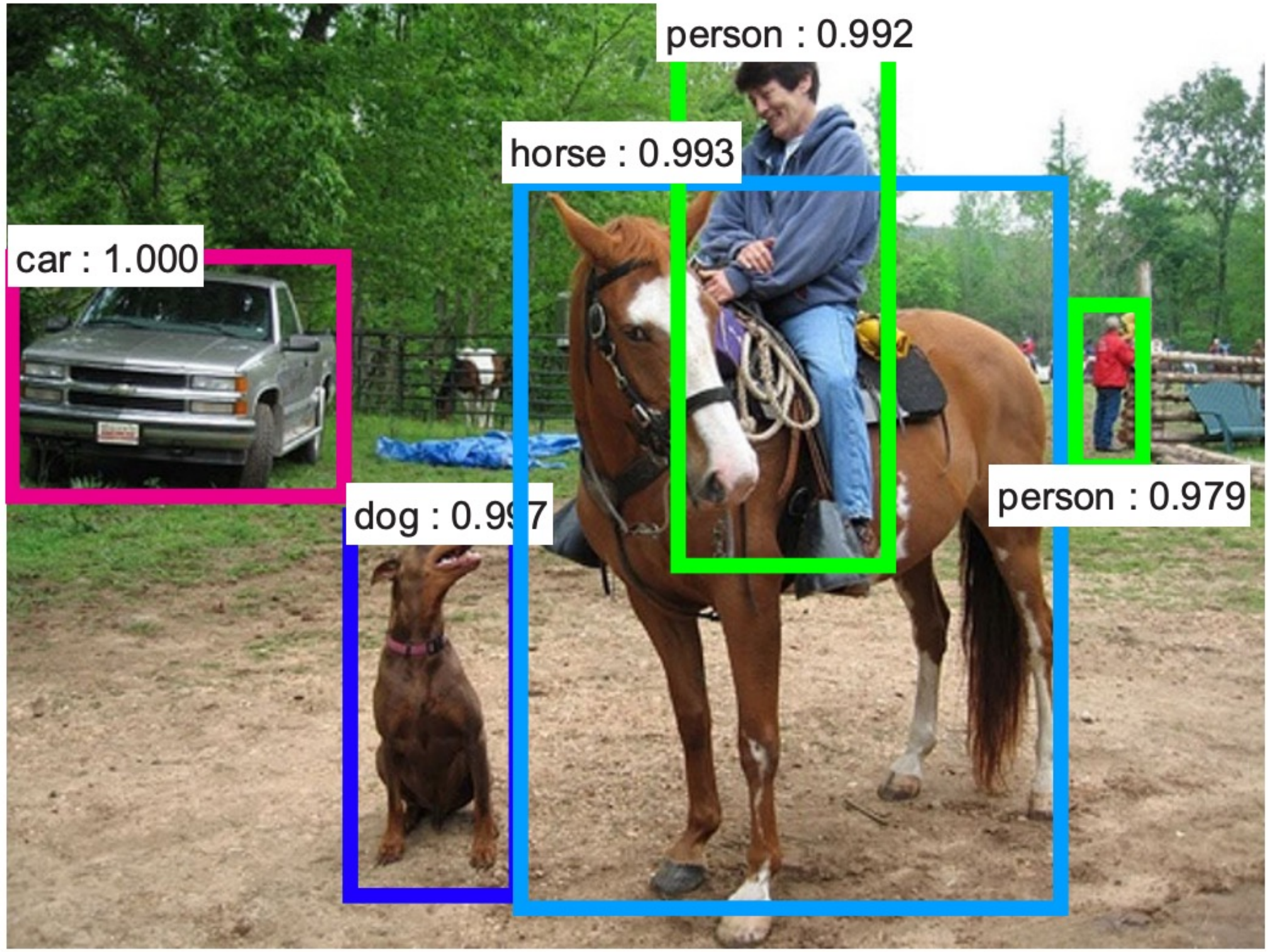
 “Facial Landmark Detection: a Literature Survey”, Wu, Ji, 2018, https://arxiv.org/abs/1805.05563

Pose Detection

- Define a landmark for each joint
  (e.g. shoulders, elbows, knees, ankles, neck, wrists)

“DeepPose: Human Pose Estimation via Deep Neural Networks”, Toshev, Szededy, 2013, https://arxiv.org/abs/1312.4659

Object Detection – Classification and Localization of zero or more objects

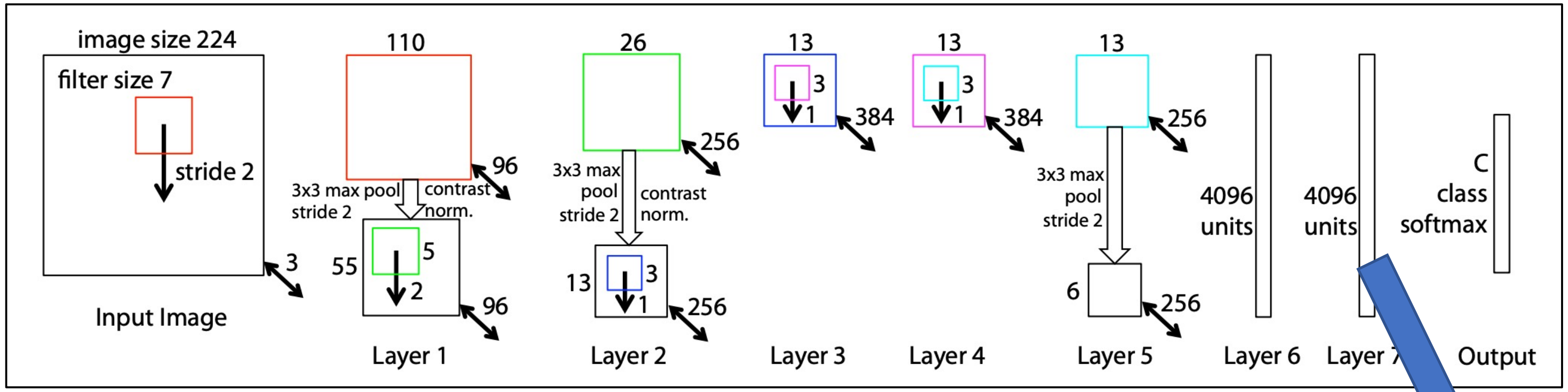


“Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, Ren, He, Girshick, Sun, 2015, https://arxiv.org/abs/1506.01497

Brad Quinton, Scott Chin

Can we use our localization approach?

A CNN Classifier (e.g. ZFNet)

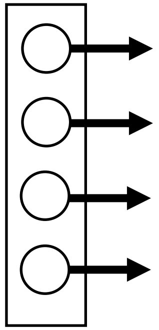


Class Prediction

$\hat{y}$

Bounding Box Prediction

4096 Vector



$\widehat{b_x}$

$\widehat{b_y}$

$\widehat{b_h}$

$\widehat{b_w}$

4 Unit FC Layer

Not directly. This method is for a **FIXED** number of objects.

Image source: Davi Frossard https://www.cs.toronto.edu/~frossard/post/vgg16/

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Brad Quinton, Scott Chin

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a "none of the above" class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Brad Quinton, Scott Chin

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Brad Quinton, Scott Chin

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Brad Quinton, Scott Chin

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Brad Quinton, Scott Chin

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Brad Quinton, Scott Chin

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a "none of the above" class.
- Supply various crops of the image to the CNN via sliding window

Window Size

Brad Quinton, Scott Chin

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

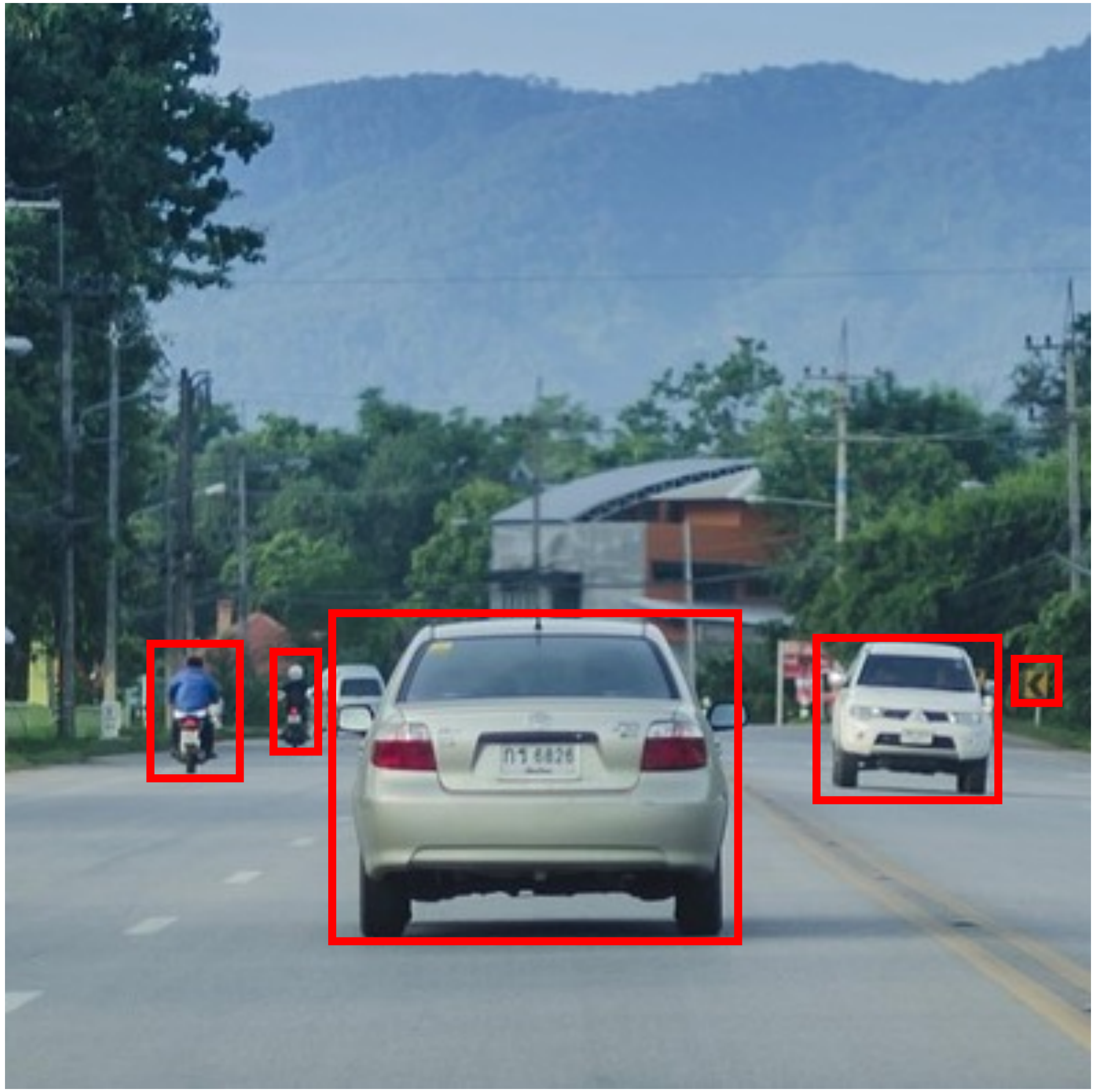
Window Size

Detecting Multiple Objects - Sliding Window

- Say you want to detect cars, motorcycles, signs
- Start with a **trained** CNN classifier that knows about these classes, and a “none of the above” class.
- Supply various crops of the image to the CNN via sliding window

Window Size

What Window Shapes to Use? What Stride?

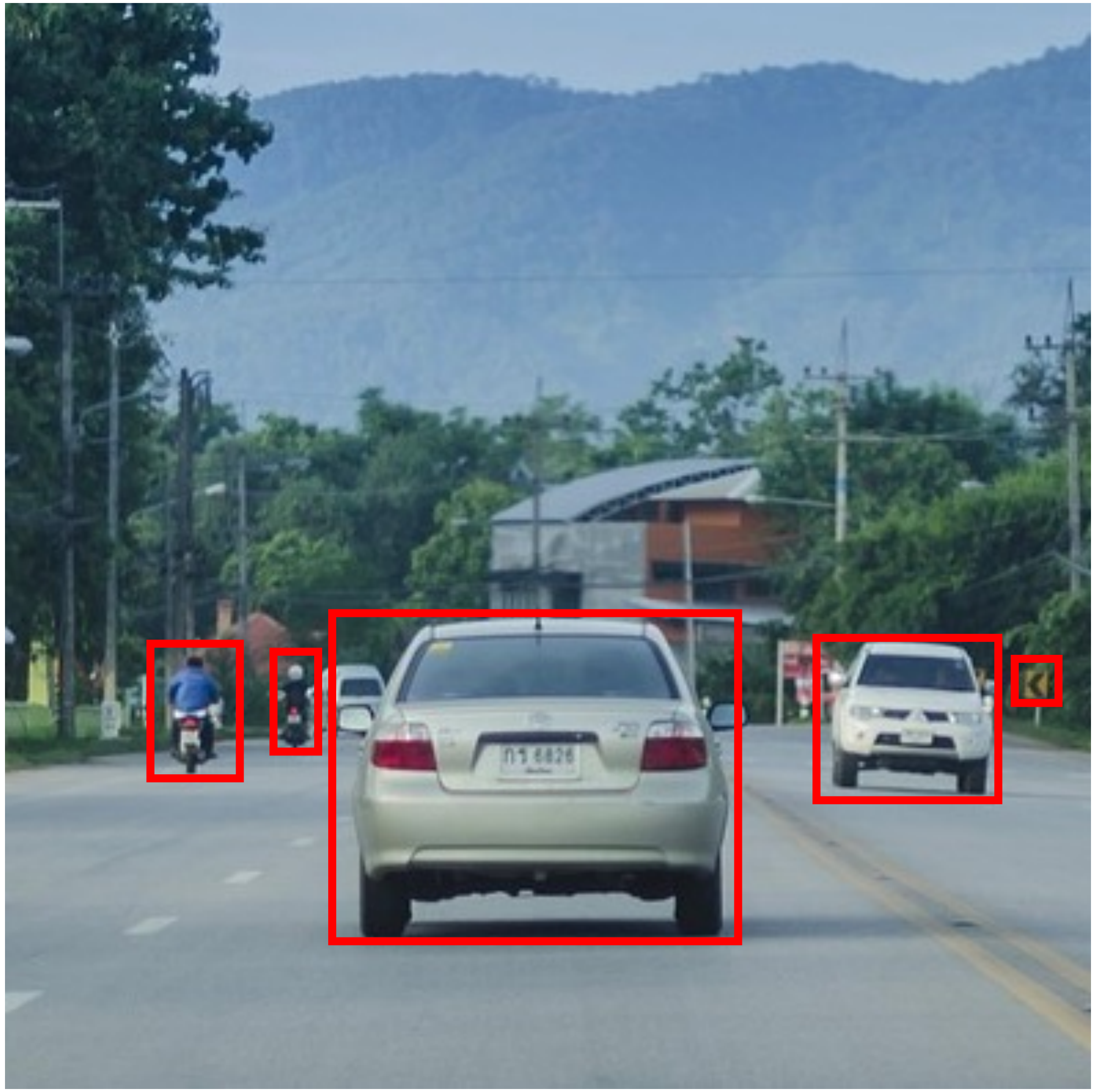


Brad Quinton, Scott Chin

What Window Shapes to Use? What Stride?

Sliding Window Locations for one window of shape (b_h, b_w) in an image of shape (H, W):

$$(H - b_h + 1) \cdot (W - b_w + 1)$$



Brad Quinton, Scott Chin

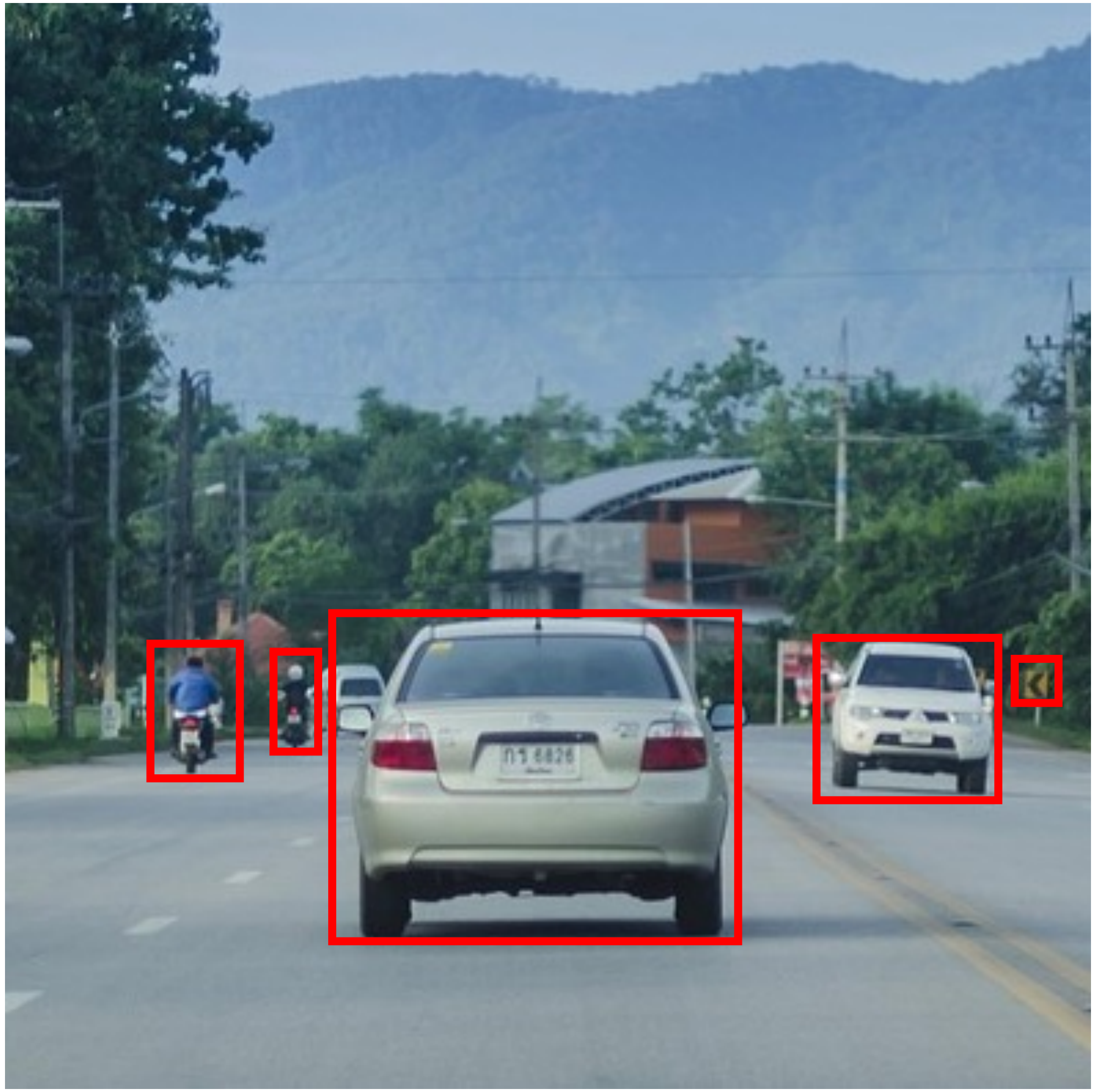
What Window Shapes to Use? What Stride?

Sliding Window Locations for one window of shape (b_h, b_w) in an image of shape (H, W):

$$(H - b_h + 1) \cdot (W - b_w + 1)$$

Repeat for all possible window shapes

$$\sum_{b_h=1}^{H} \sum_{b_w=1}^{W} (H - b_h + 1) \cdot (W - b_w + 1)$$



Brad Quinton, Scott Chin

What Window Shapes to Use? What Stride?

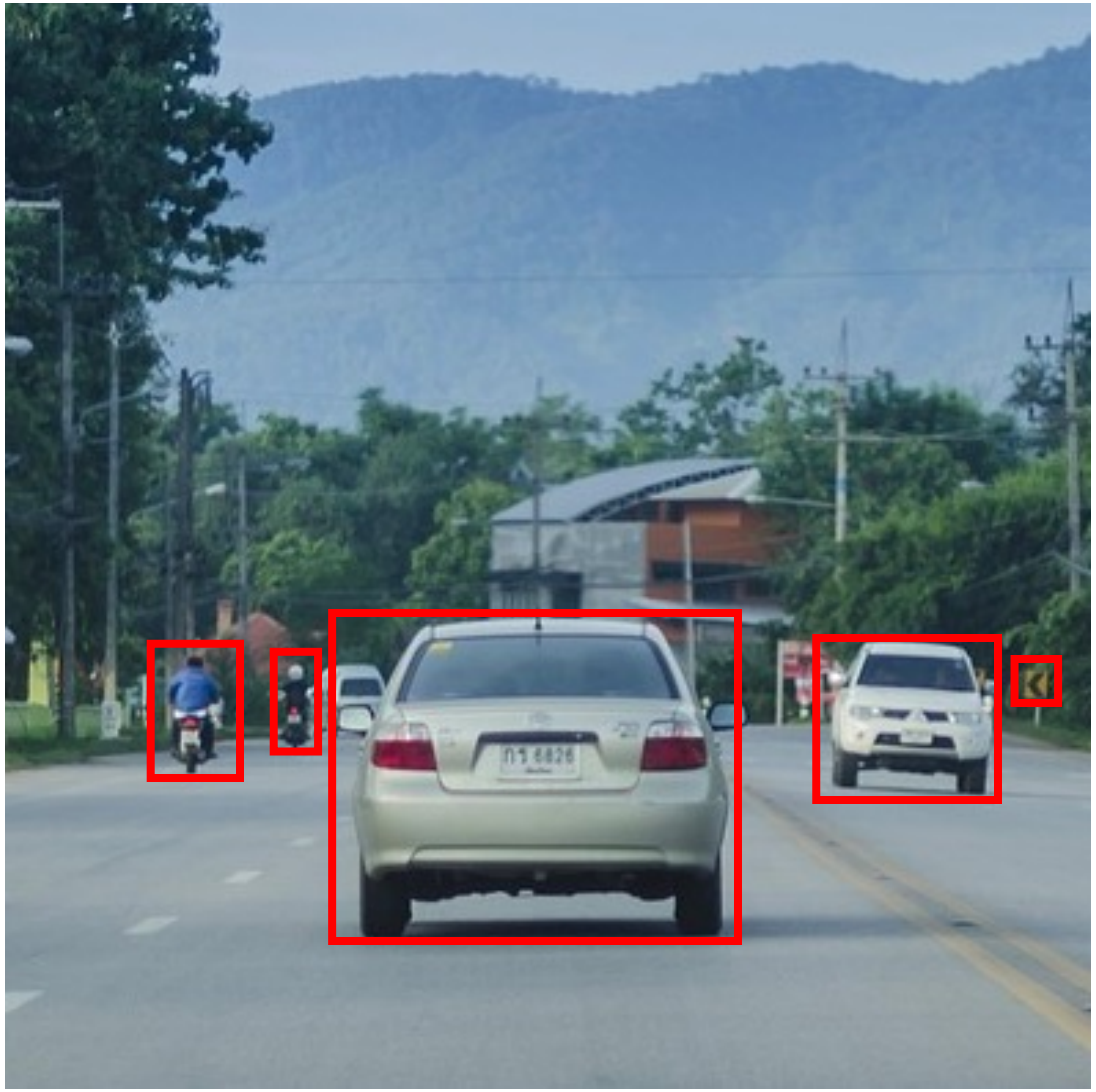
Sliding Window Locations for one window of shape (b_h, b_w) in an image of shape (H, W):

$$(H - b_h + 1) \cdot (W - b_w + 1)$$

Repeat for all possible window shapes

$$\sum_{b_h=1}^{H} \sum_{b_w=1}^{W} (H - b_h + 1) \cdot (W - b_w + 1)$$

Infeasible to look at all possible window sizes, at all locations iteratively

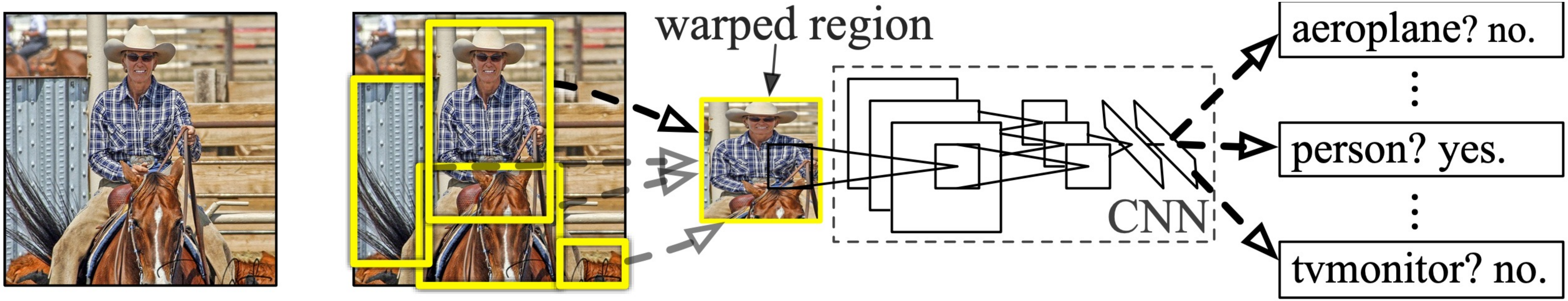


Regions with CNN Features (R-CNN)

“Rich feature hierarchies for accurate object detection and semantic segmentation”, Girshick et al, 2014, https://arxiv.org/abs/1311.2524

Brad Quinton, Scott Chin

Regions with CNN Features (R-CNN)



- First use a Region Proposal algorithm to find a manageable number of regions (crops) that potentially have an object
- Send region crops to classifier
- Region crop location and size is the bounding box prediction

"Rich feature hierarchies for accurate object detection and semantic segmentation", Girshick et al, 2014, https://arxiv.org/abs/1311.2524

Faster R-CNN

- R-CNN
  - Propose regions. Evaluate one region at a time
- Fast R-CNN
  - Classify all proposed regions at once
- Faster R-CNN
  - Uses a CNN to propose regions

"Rich feature hierarchies for accurate object detection and semantic segmentation", Girshick et al, 2014, https://arxiv.org/abs/1311.2524
"Fast R-CNN", Girshick, 2015, https://arxiv.org/abs/1504.08083
"Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", Ren, He, Girshick, Sun, 2015, https://arxiv.org/abs/1506.01497

Another Approach:
You Only Look Once (YOLO)

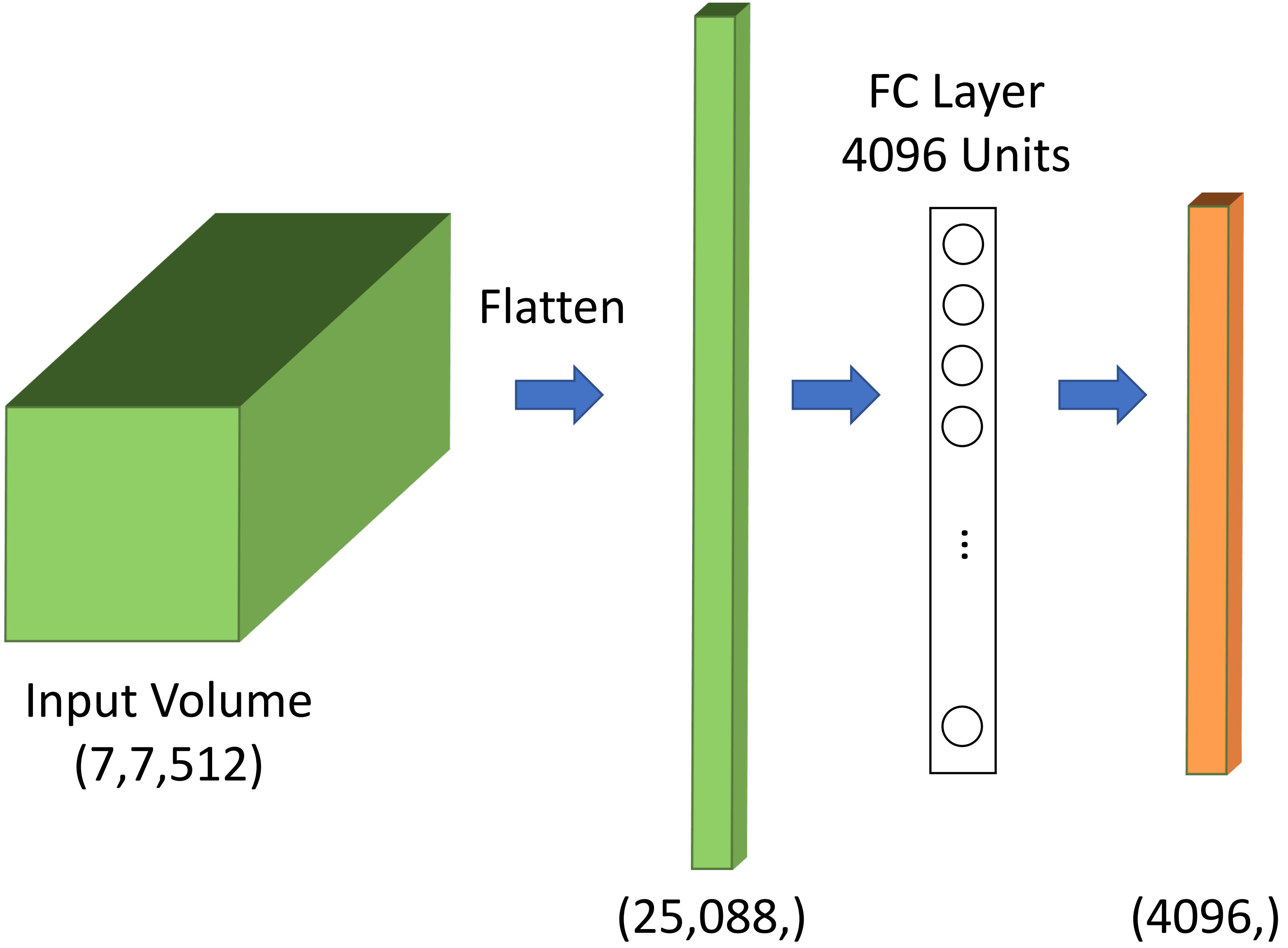
"You Only Look Once: Unified, Real-Time Object Detection", Redmond et al, 2015,
https://arxiv.org/abs/1506.02640

Selection of YOLO Innovations

- Implement Sliding window via convolution
- Can evaluate all sliding window locations in one pass
- This is fast!

- Some restrictions on stride and size of the sliding window

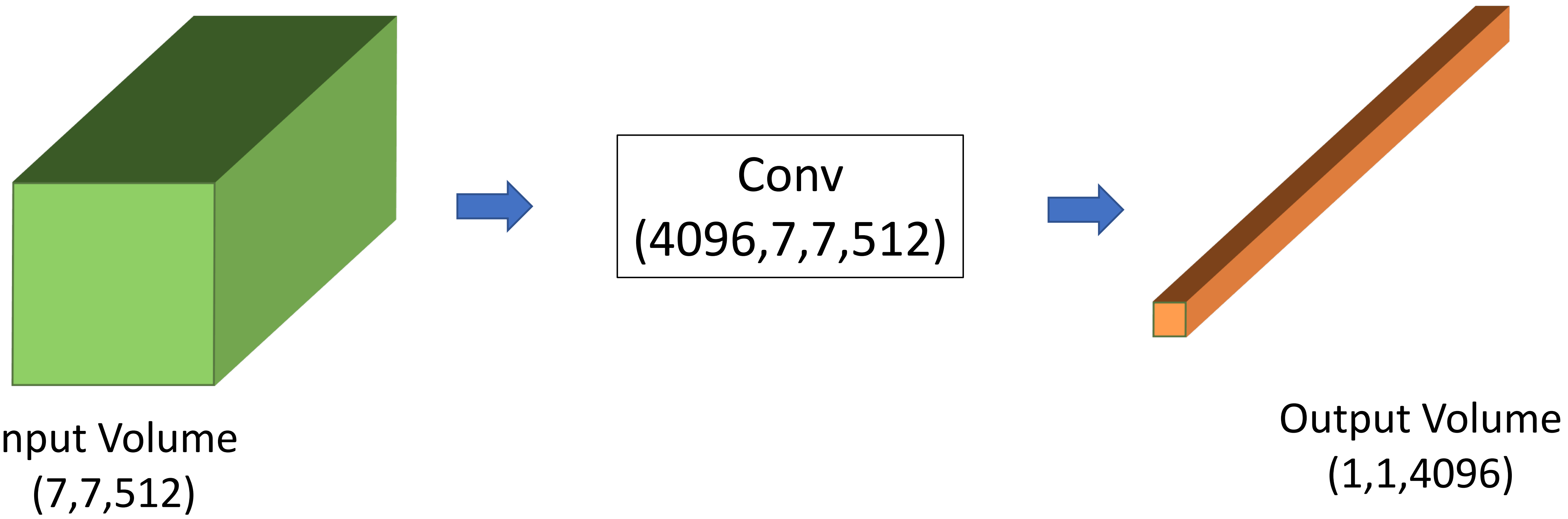
Building FC Layers with Conv Layers



Brad Quinton, Scott Chin

Building FC Layers with Conv Layers

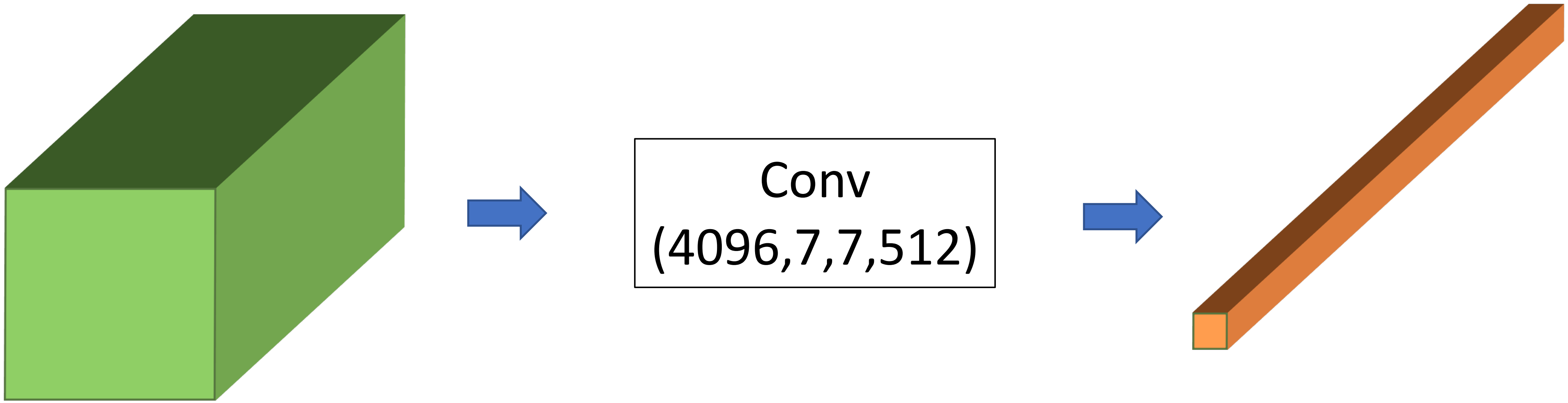
Can achieve equivalent result
with the following convolution



Brad Quinton, Scott Chin

Building FC Layers with Conv Layers

Can achieve equivalent result
with the following convolution



Input Volume
(7,7,512)

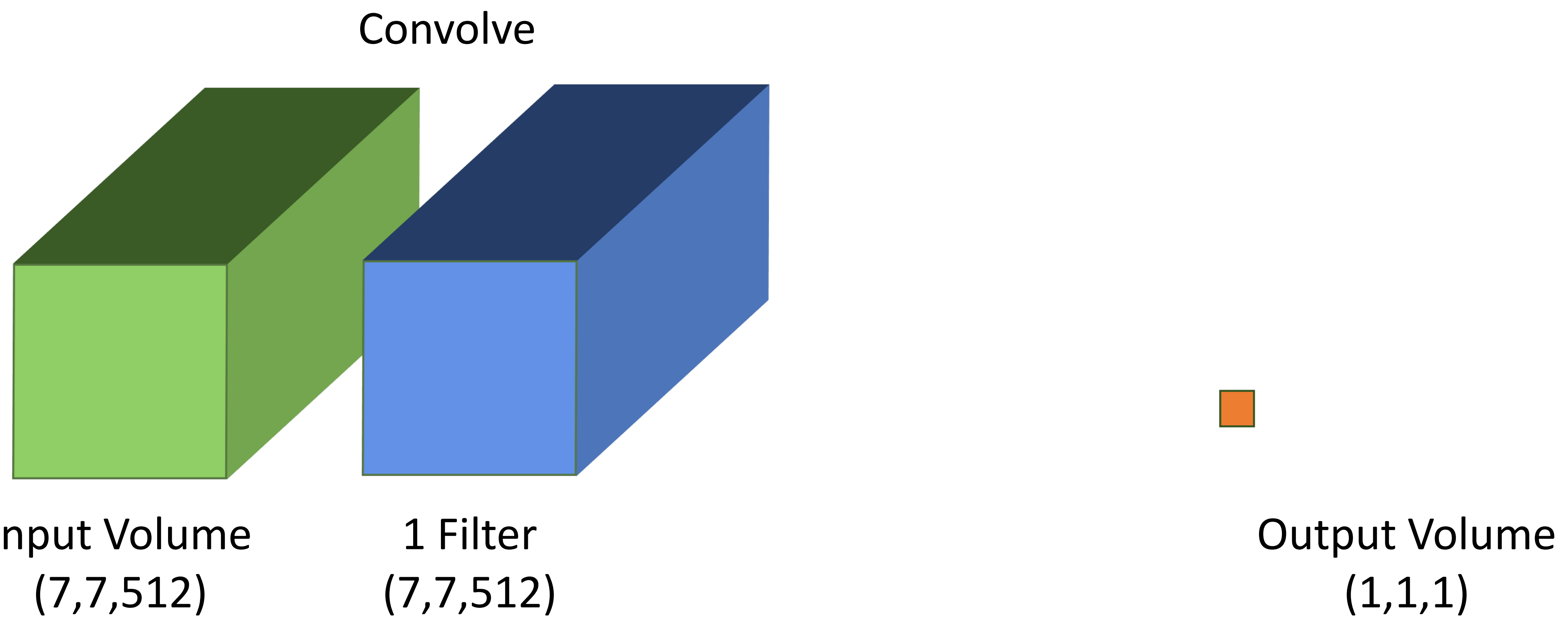
Output Volume
(1,1,4096)

Filters have same shape as input volume
One filter for each FC output unit

Brad Quinton, Scott Chin

Building FC Layers with Conv Layers

Example for one filter

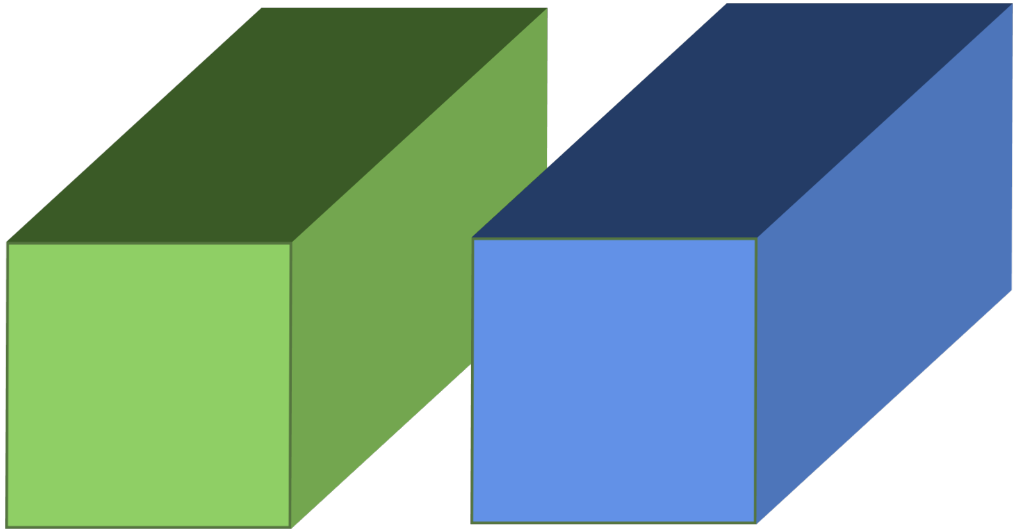


Brad Quinton, Scott Chin

Building FC Layers with Conv Layers

Example for one filter

Convolve



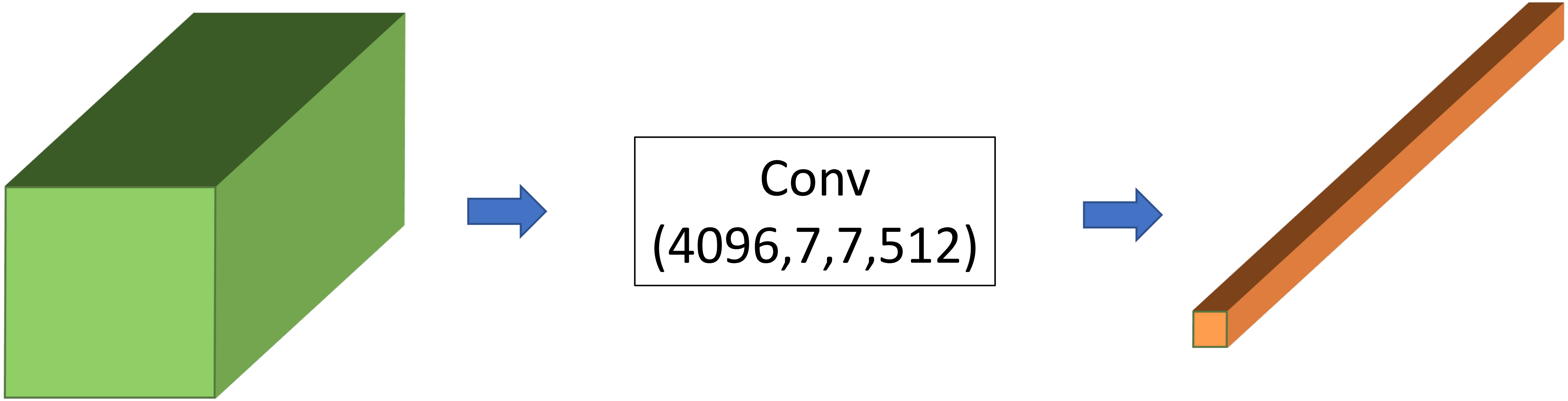
One weight for each input volume element. Mathematically equivalent to a single Fully Connected unit

Input Volume
(7,7,512)

1 Filter
(7,7,512)

Output Volume
(1,1,1)

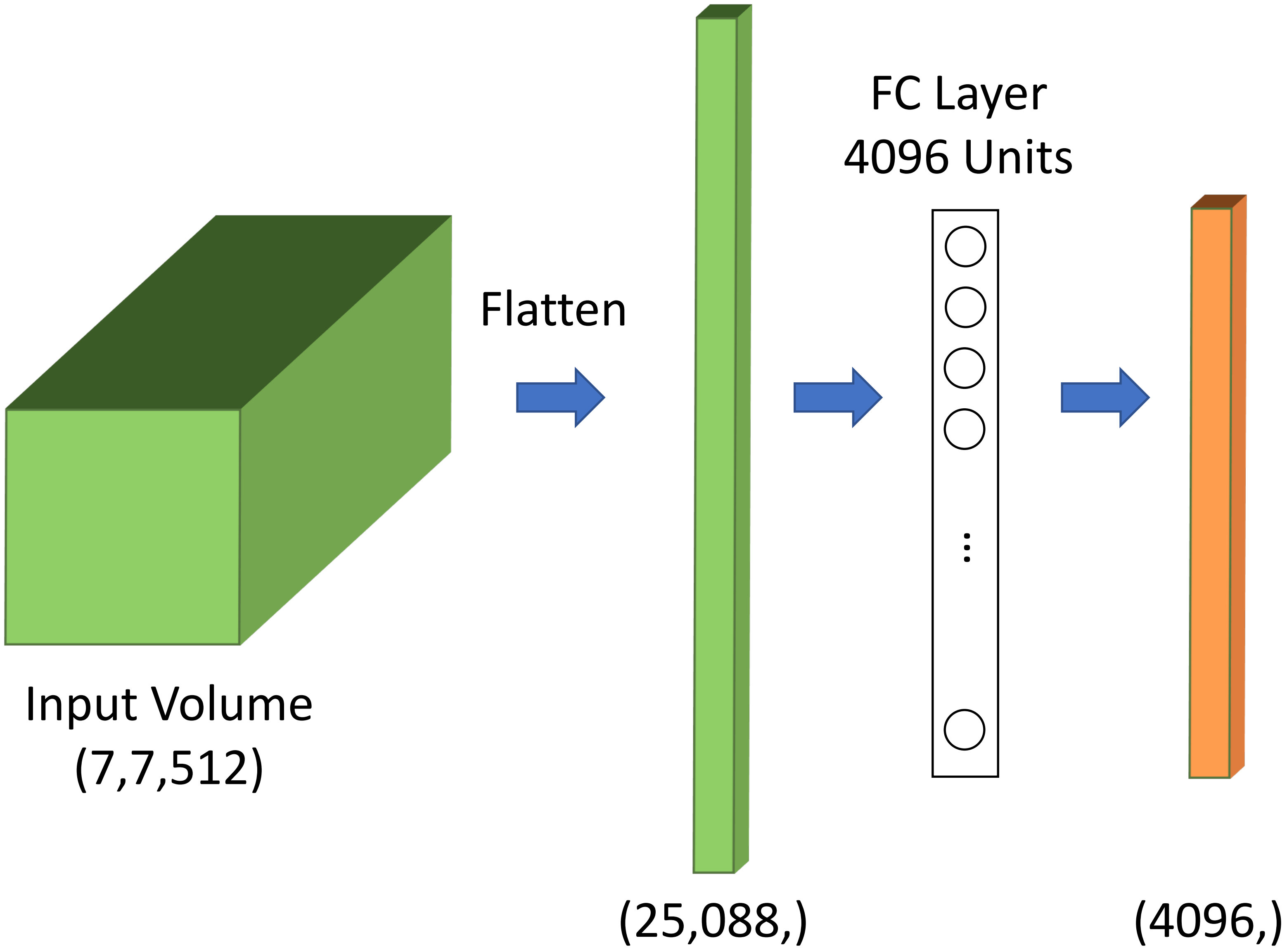
Building FC Layers with Conv Layers



Input Volume
(7,7,512)

Output Volume
(1,1,4096)

Building FC Layers with Conv Layers



Brad Quinton, Scott Chin

Sliding Window via Conv FC Layers

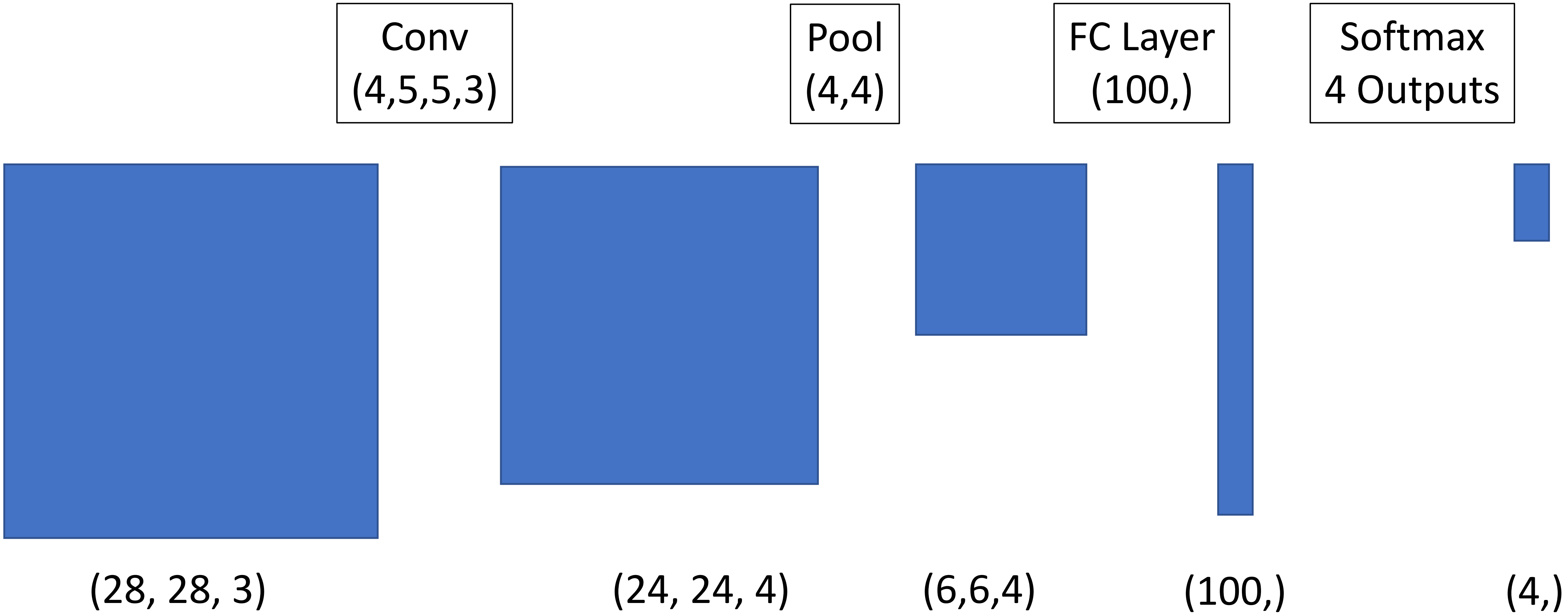
- Start with a trained CNN classifier
- Convert FC layers to use convolutional equivalent implementation
- Supply larger image for object detection.
- Each sliding window location is a potential bounding box for an object

Sliding Window via Conv FC Layers

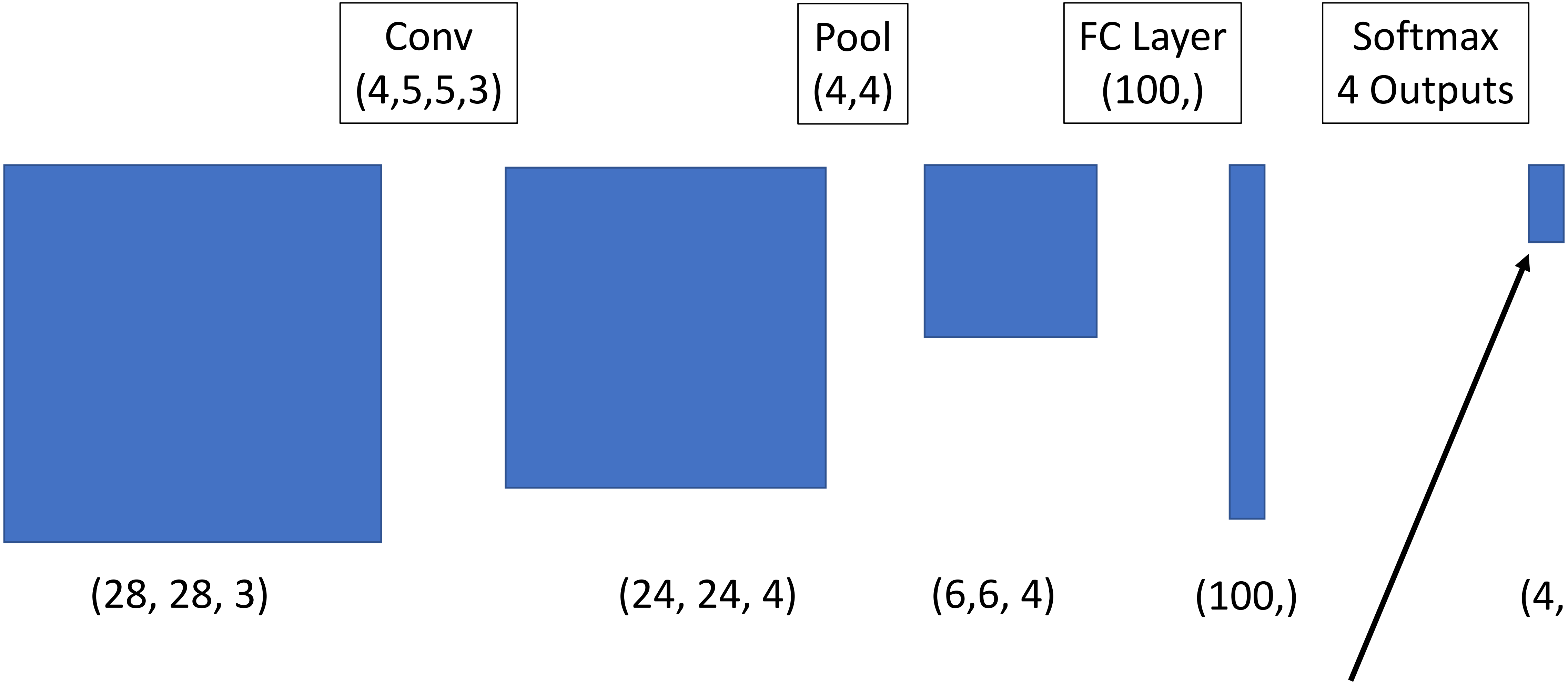
Example

- CNN Classifier with 4 softmax outputs to predict
  - Car
  - Motorcycle
  - Street sign
  - Other
- Input is 28x28 color image

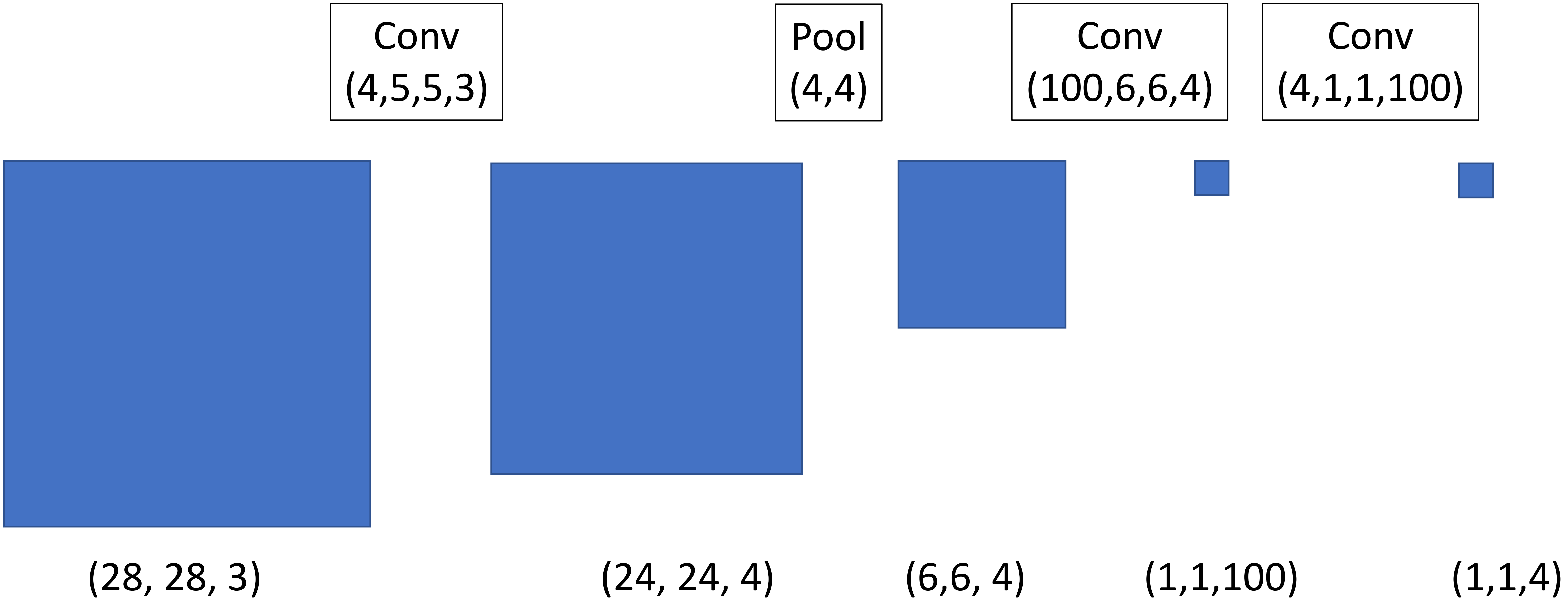
Brad Quinton, Scott Chin



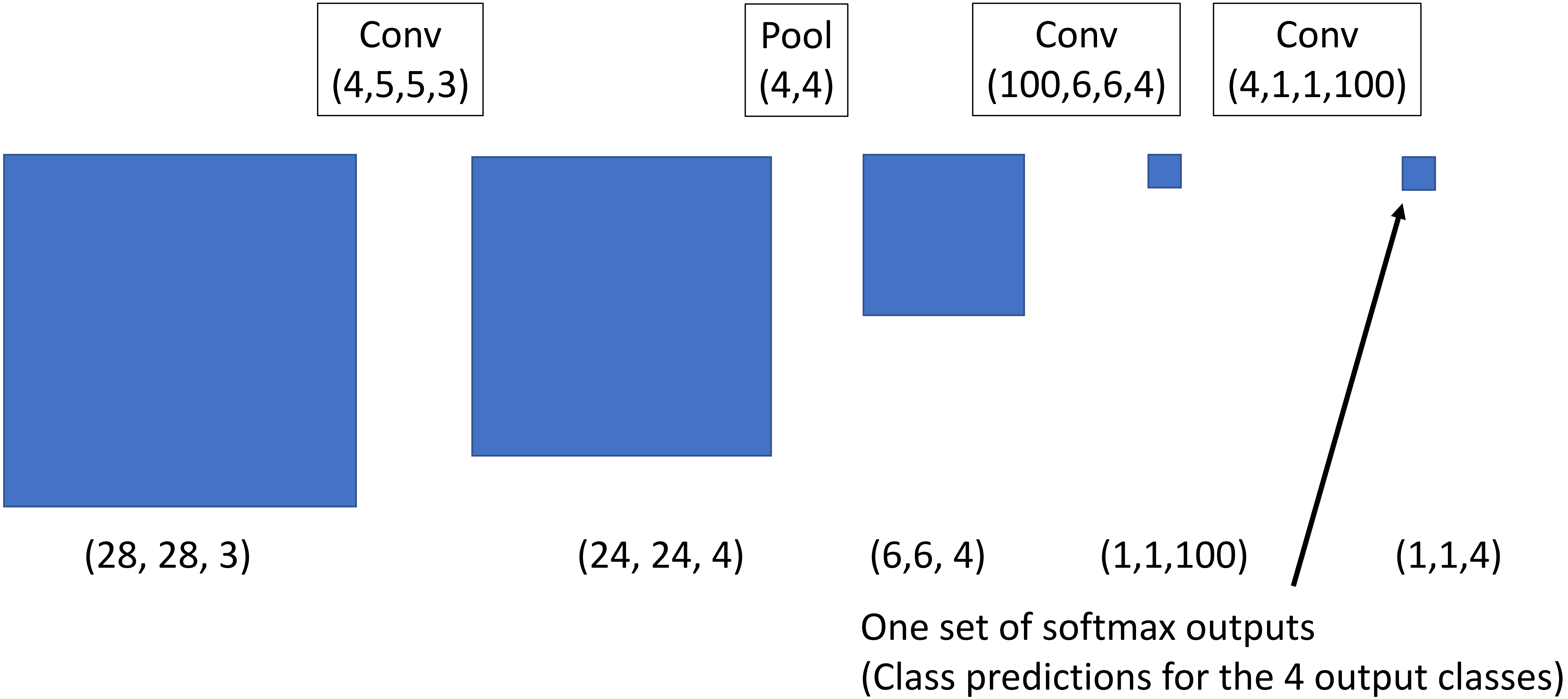
Original CNN classifier



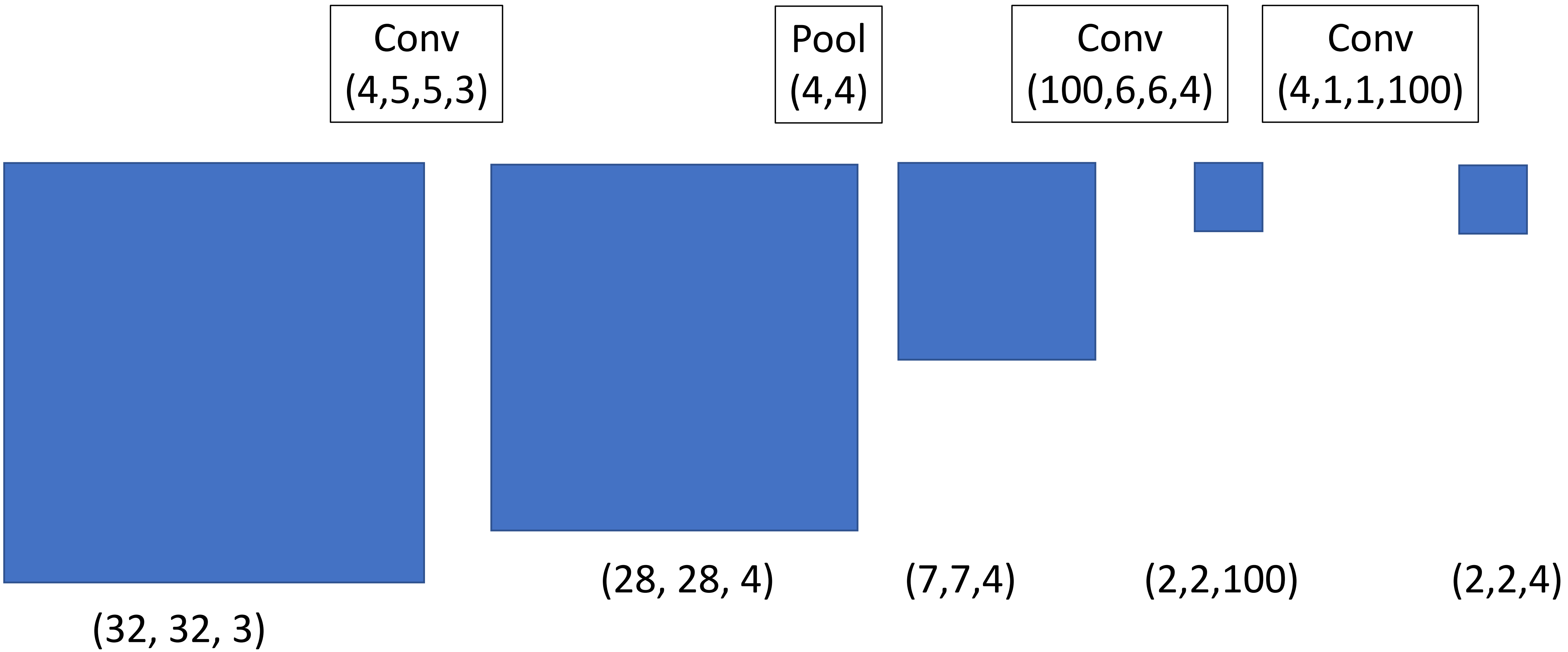
Original CNN classifier



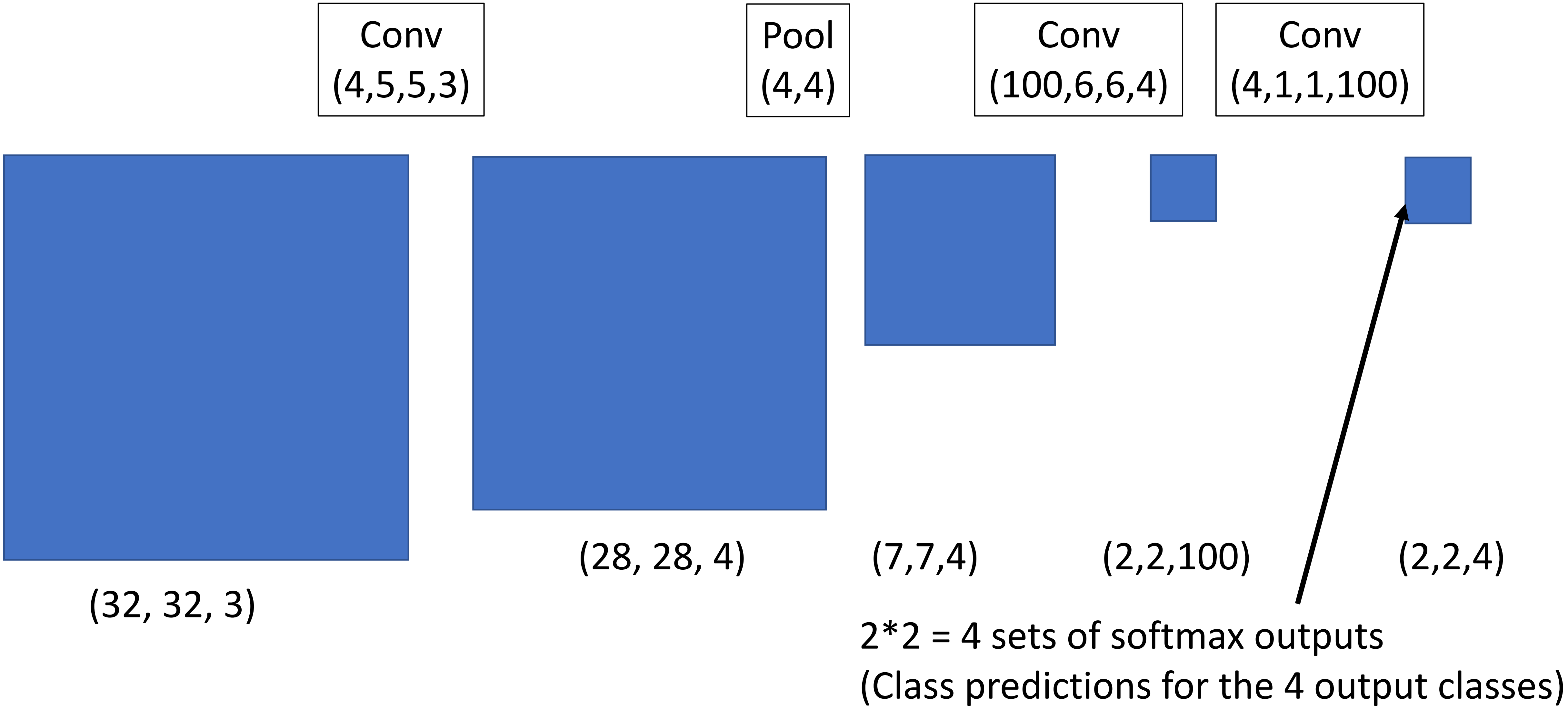
Convert FC layers to use convolutional implementation



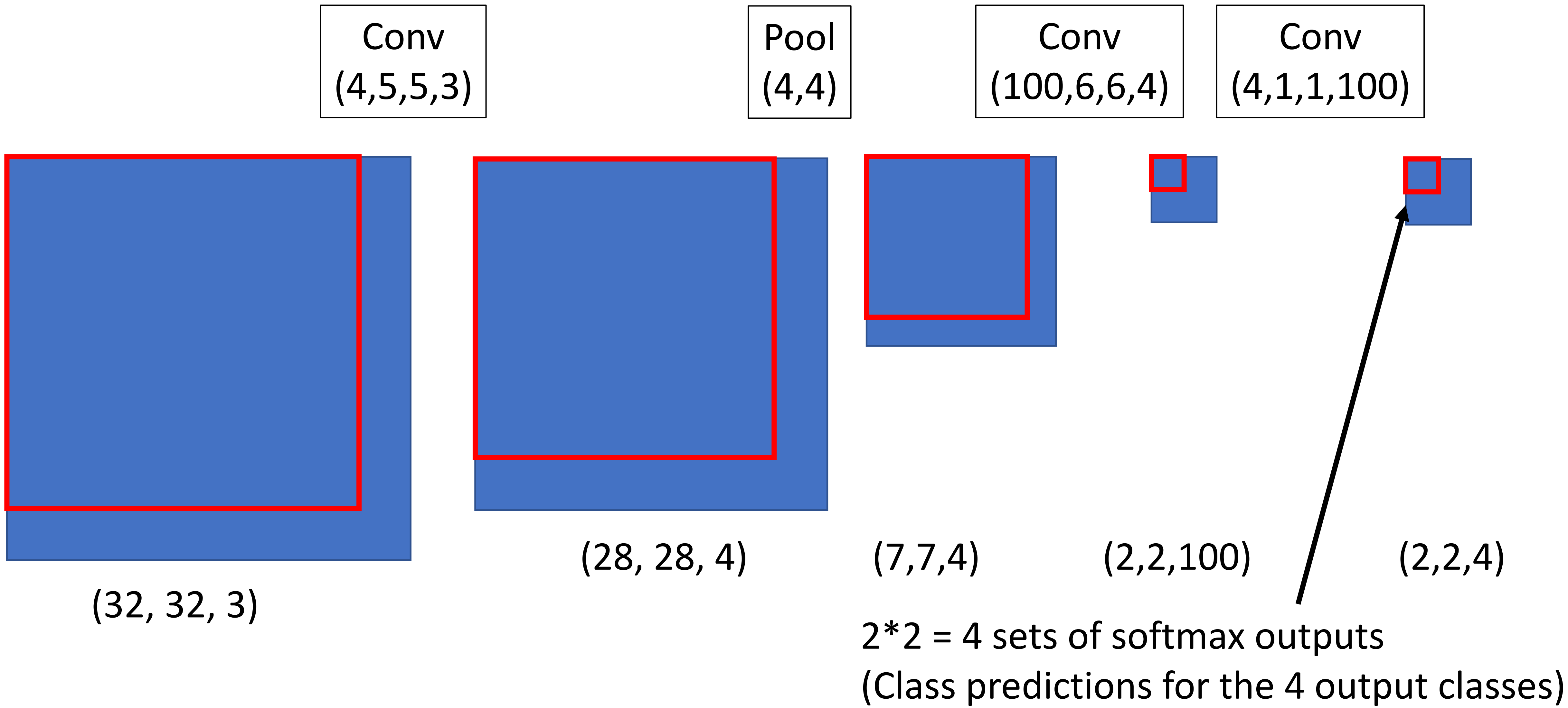
Convert FC layers to use convolutional implementation



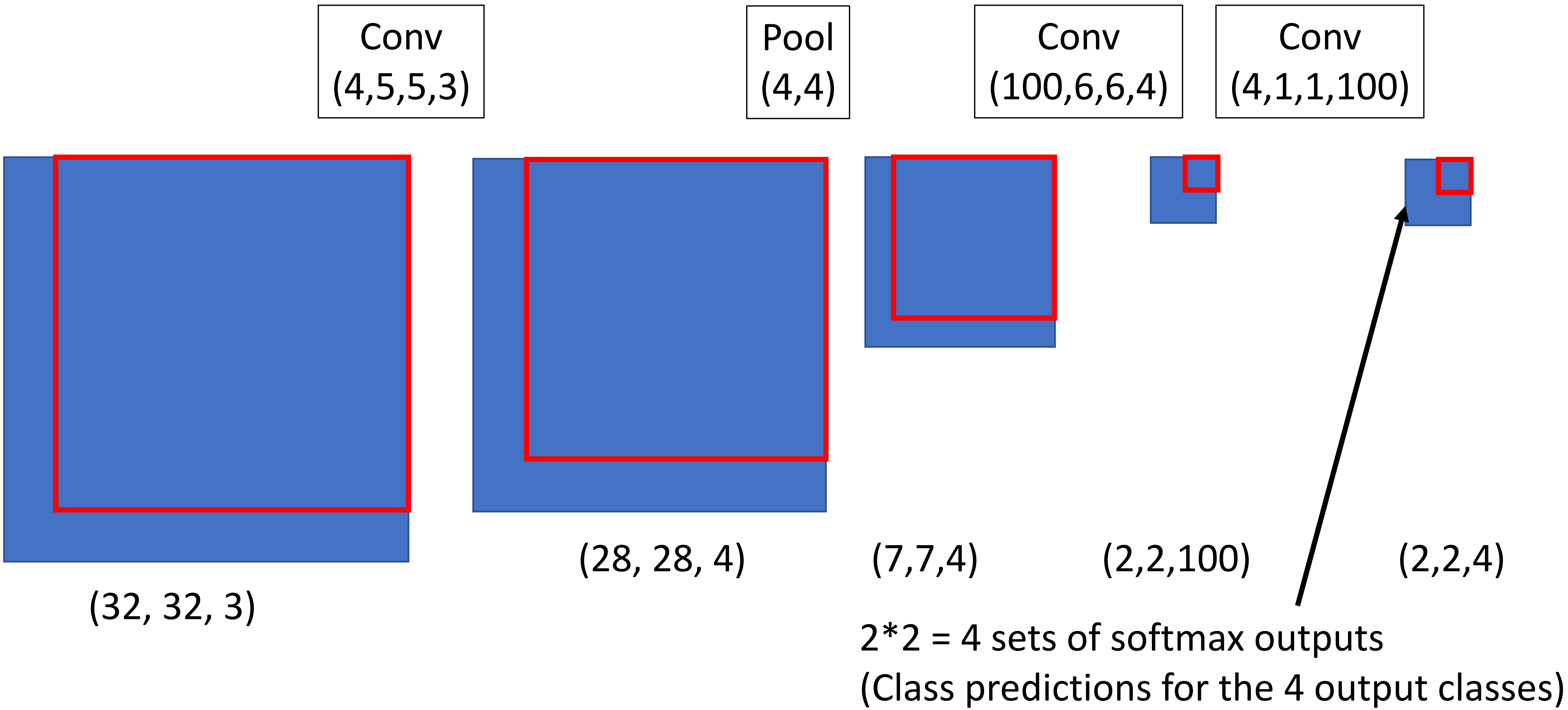
Now use for detection on a larger image



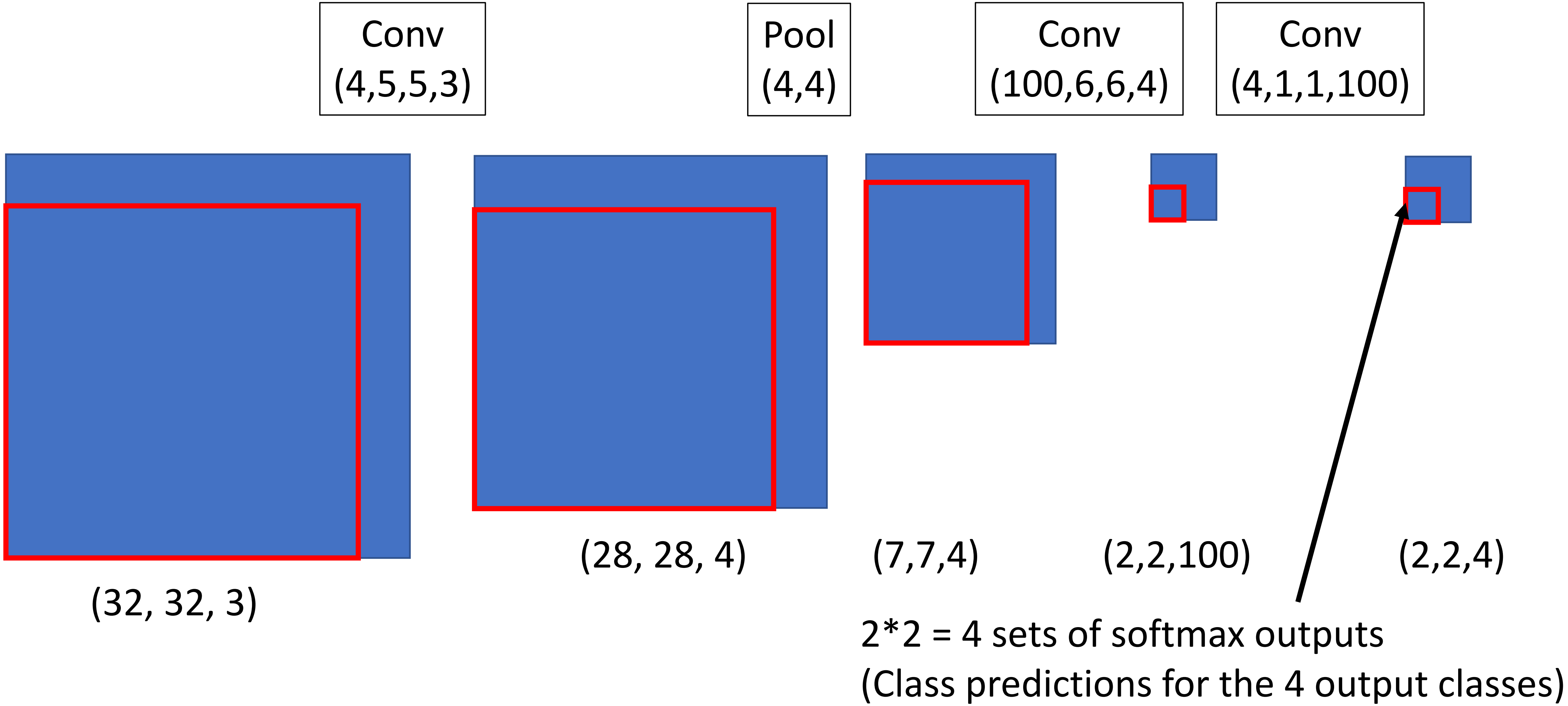
Now use for detection on a larger image



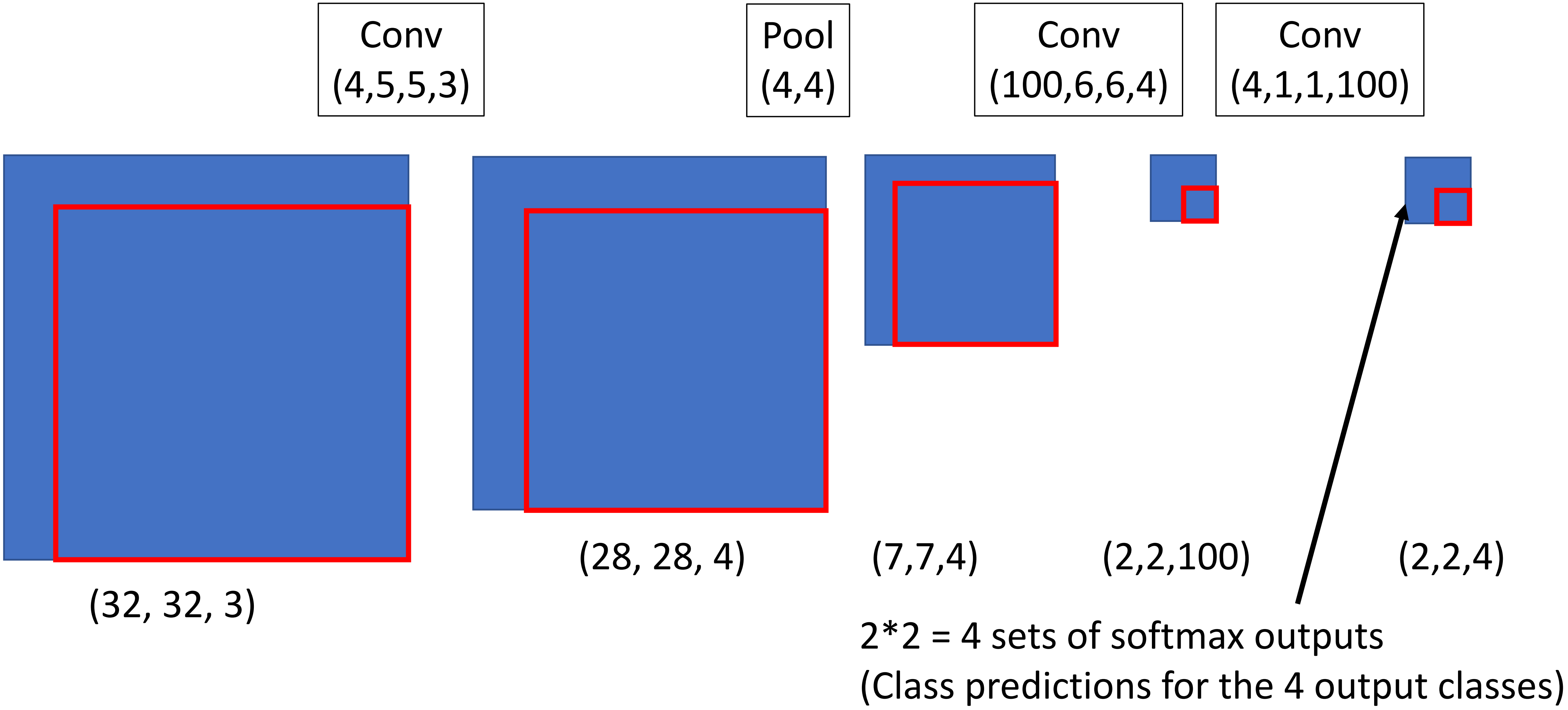
For each output set, we can map back to region of input



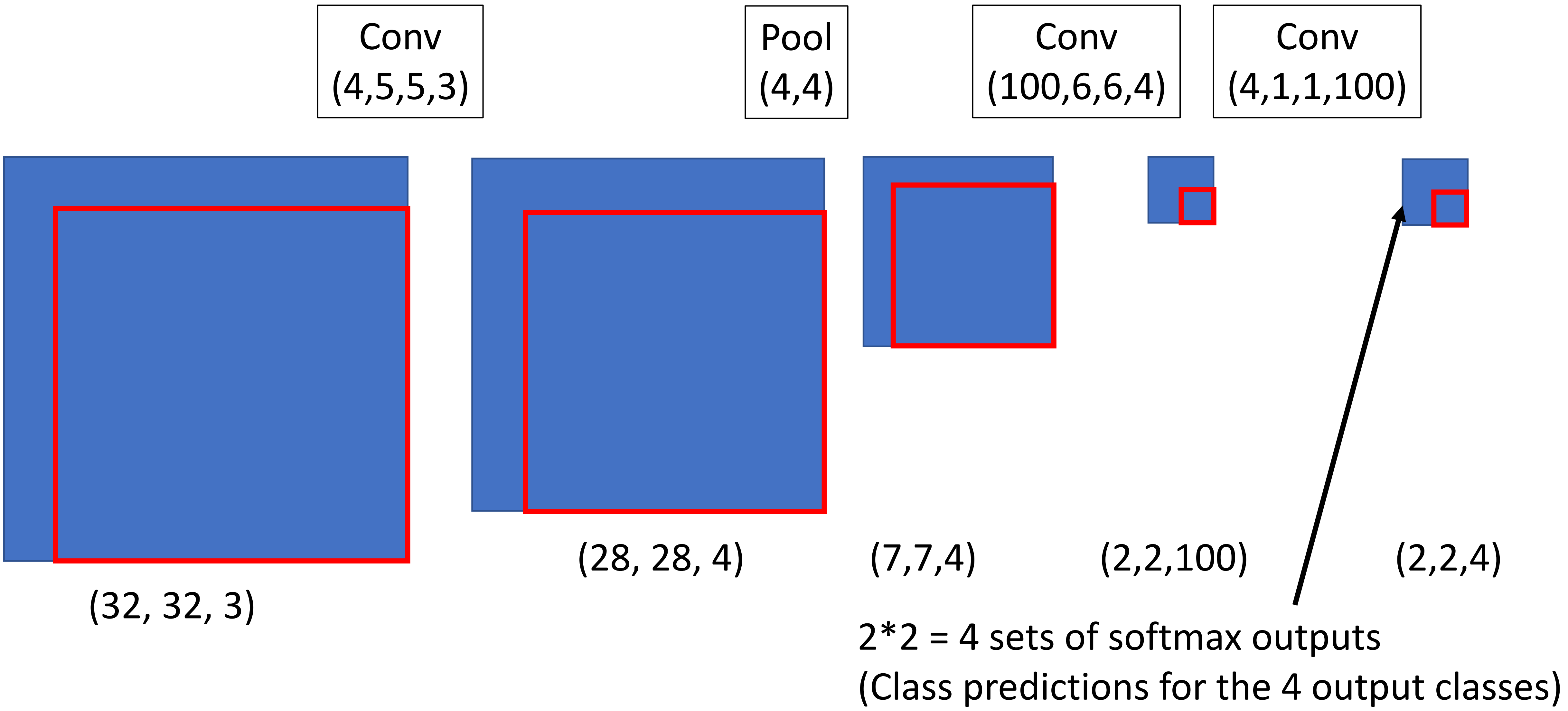
For each output set, we can map back to region of input



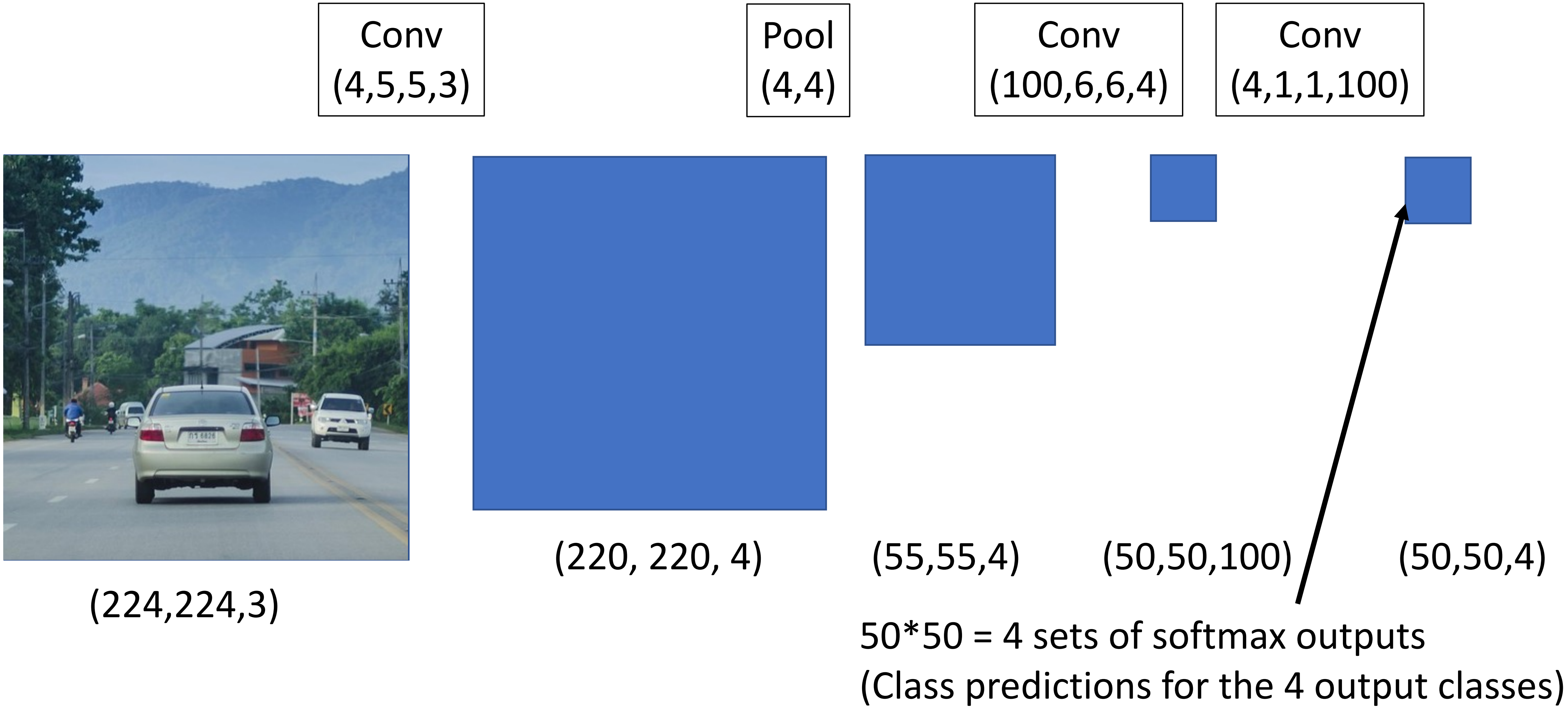
For each output set, we can map back to region of input



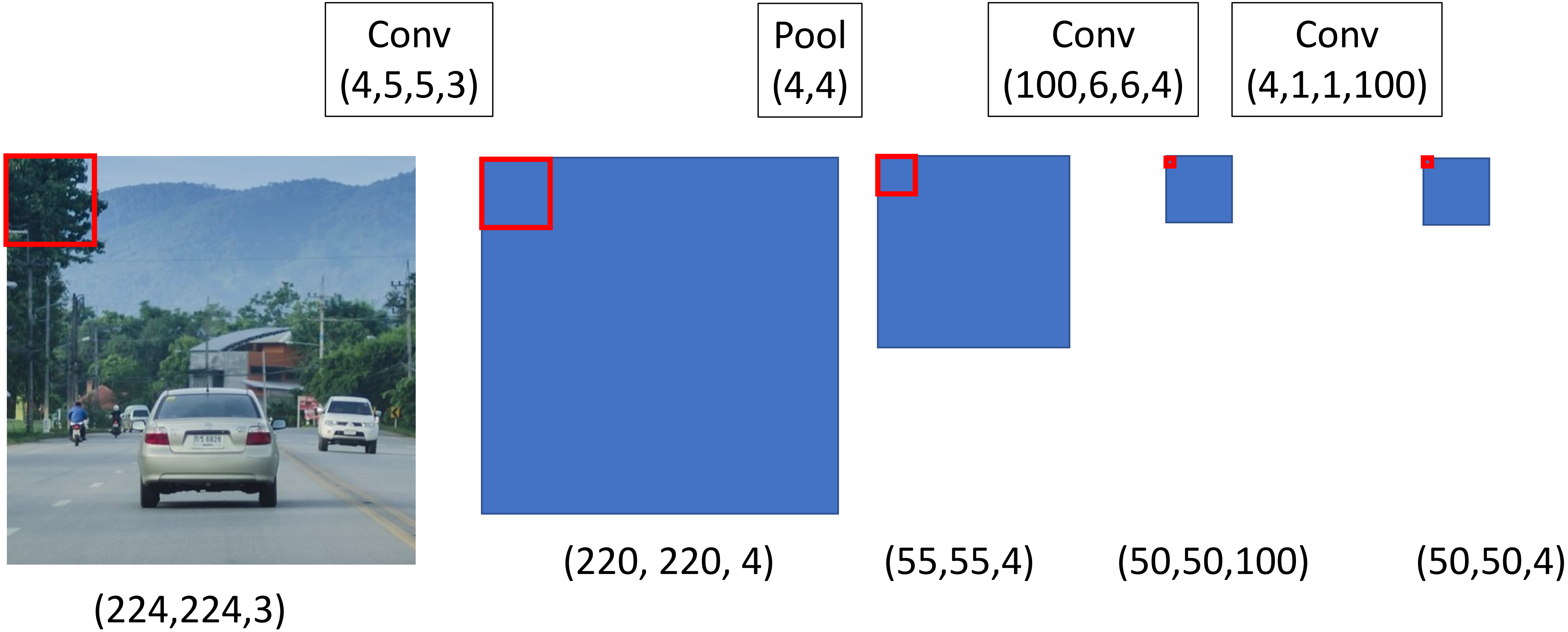
For each output set, we can map back to region of input



We have a sliding window for our classifier!



Let’s try an even larger image

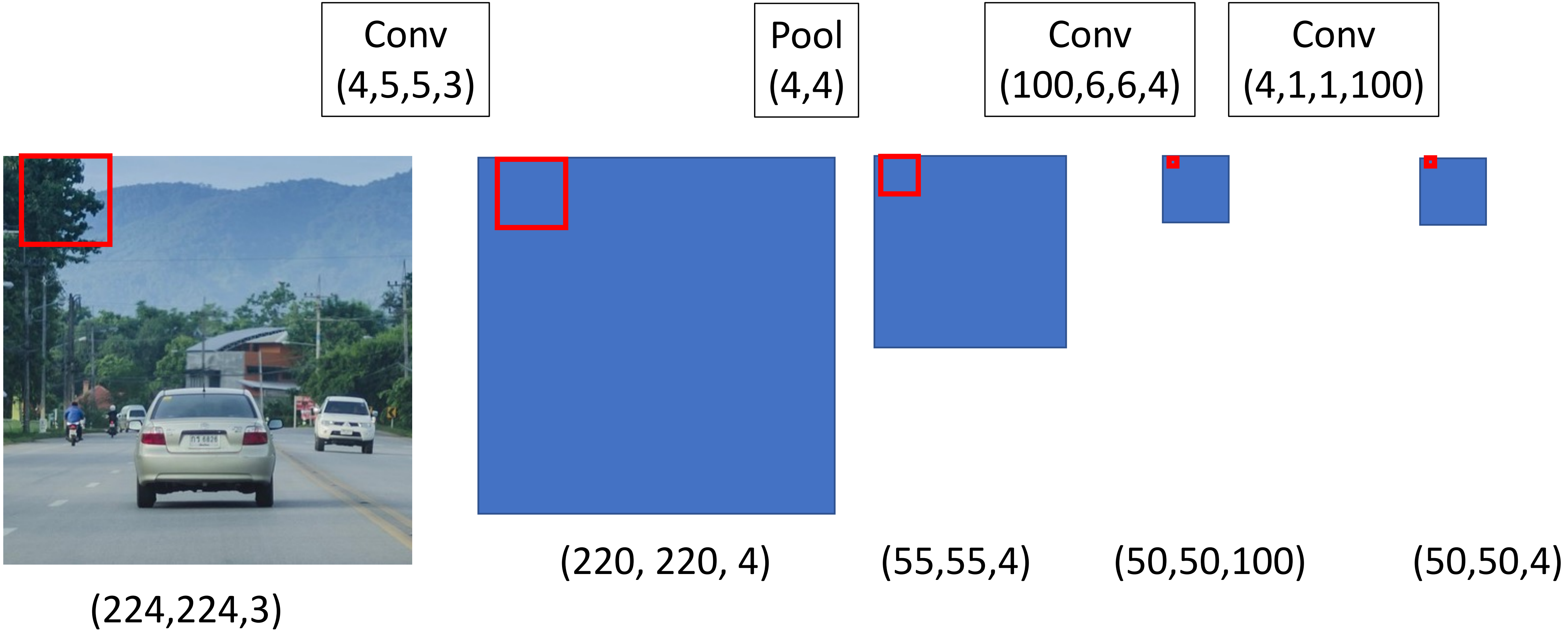


(28,28) Window
Stride = 4
(50,50) Locations

(24,24) Window
Stride = 4
(50,50) Locations

(6,6) Window
Stride = 1
(50,50) Locations

(1,1) Window
Stride = 1
(50,50) Locations

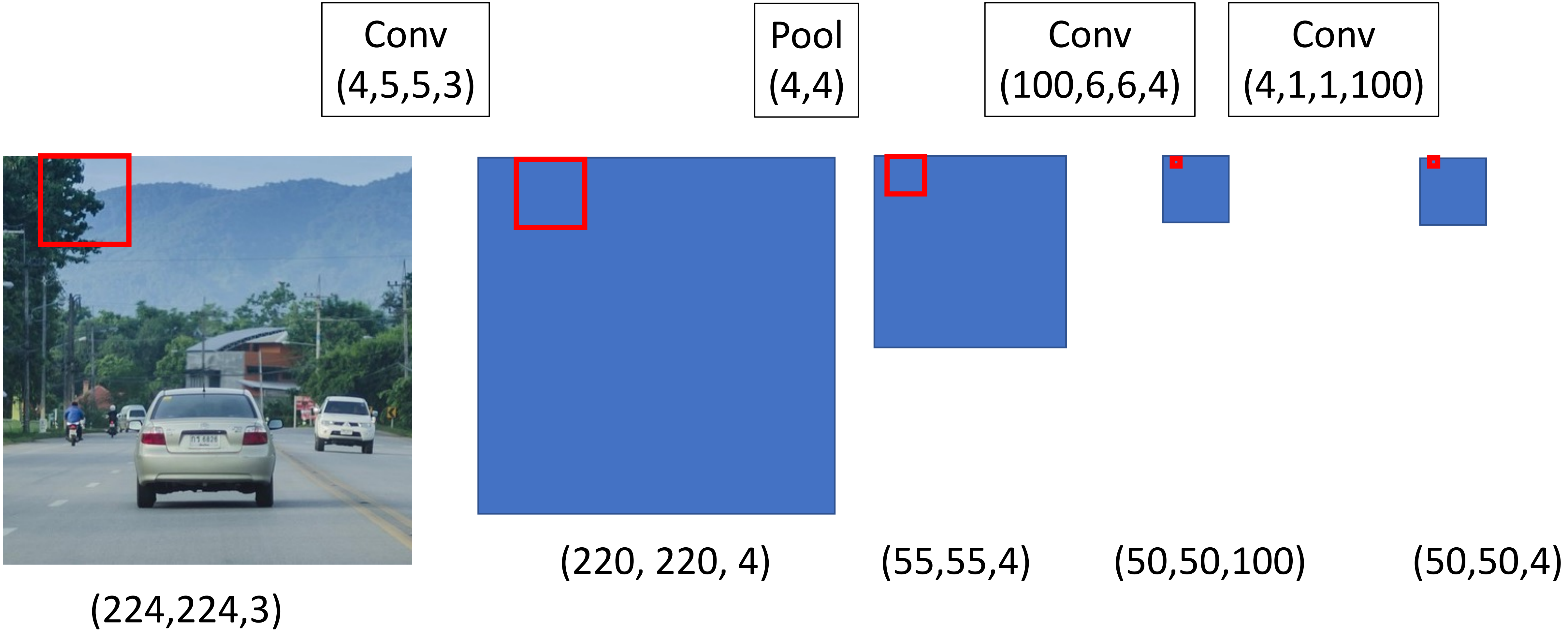


(28,28) Window
Stride = 4
(50,50) Locations

(24,24) Window
Stride = 4
(50,50) Locations

(6,6) Window
Stride = 1
(50,50) Locations

(1,1) Window
Stride = 1
(50,50) Locations

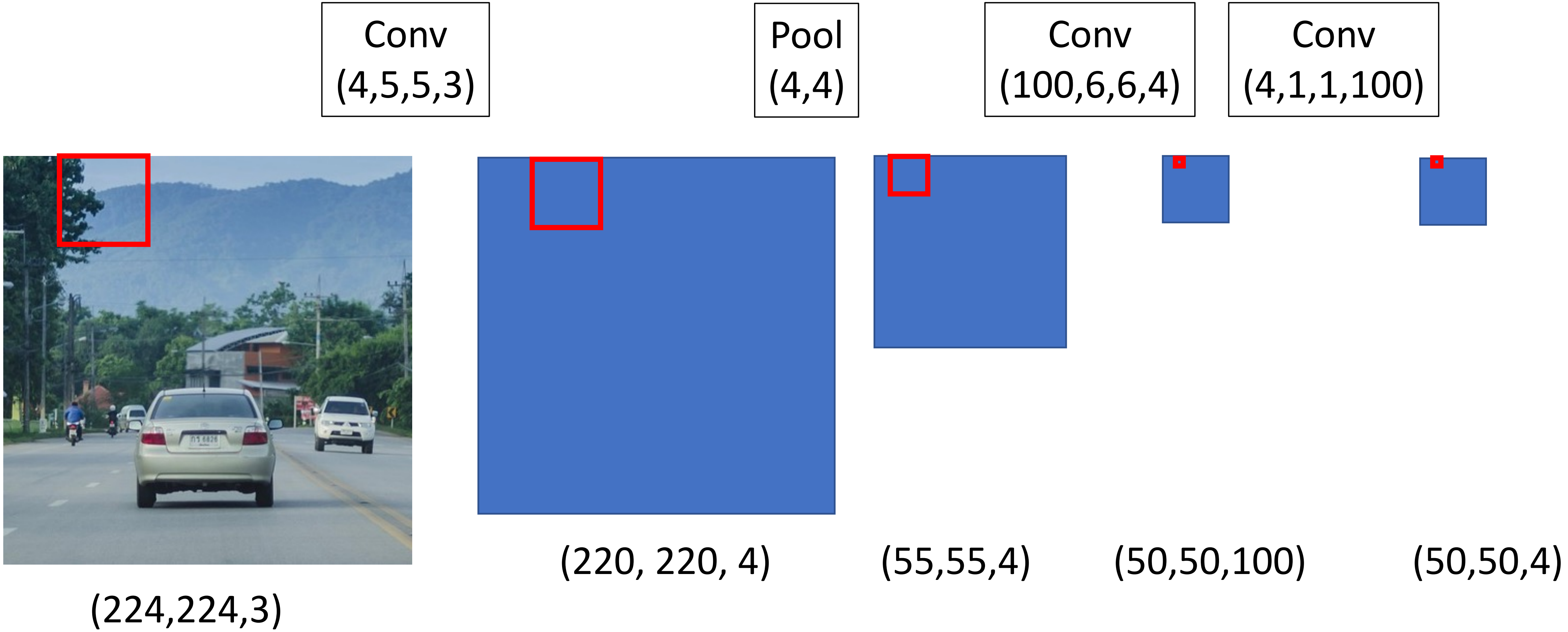


(28,28) Window
Stride = 4
(50,50) Locations

(24,24) Window
Stride = 4
(50,50) Locations

(6,6) Window
Stride = 1
(50,50) Locations

(1,1) Window
Stride = 1
(50,50) Locations



(28,28) Window
Stride = 4
(50,50) Locations

(24,24) Window
Stride = 4
(50,50) Locations

(6,6) Window
Stride = 1
(50,50) Locations

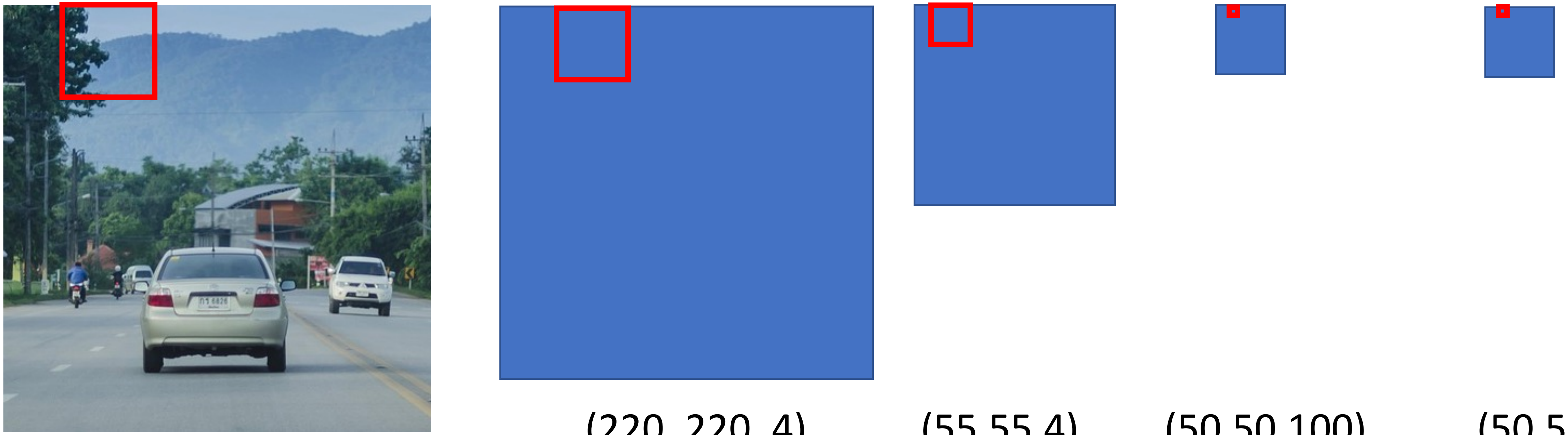
(1,1) Window
Stride = 1
(50,50) Locations

Conv (4,5,5,3)

Pool (4,4)

Conv (100,6,6,4)

Conv (4,1,1,100)



(224,224,3)

(220, 220, 4)

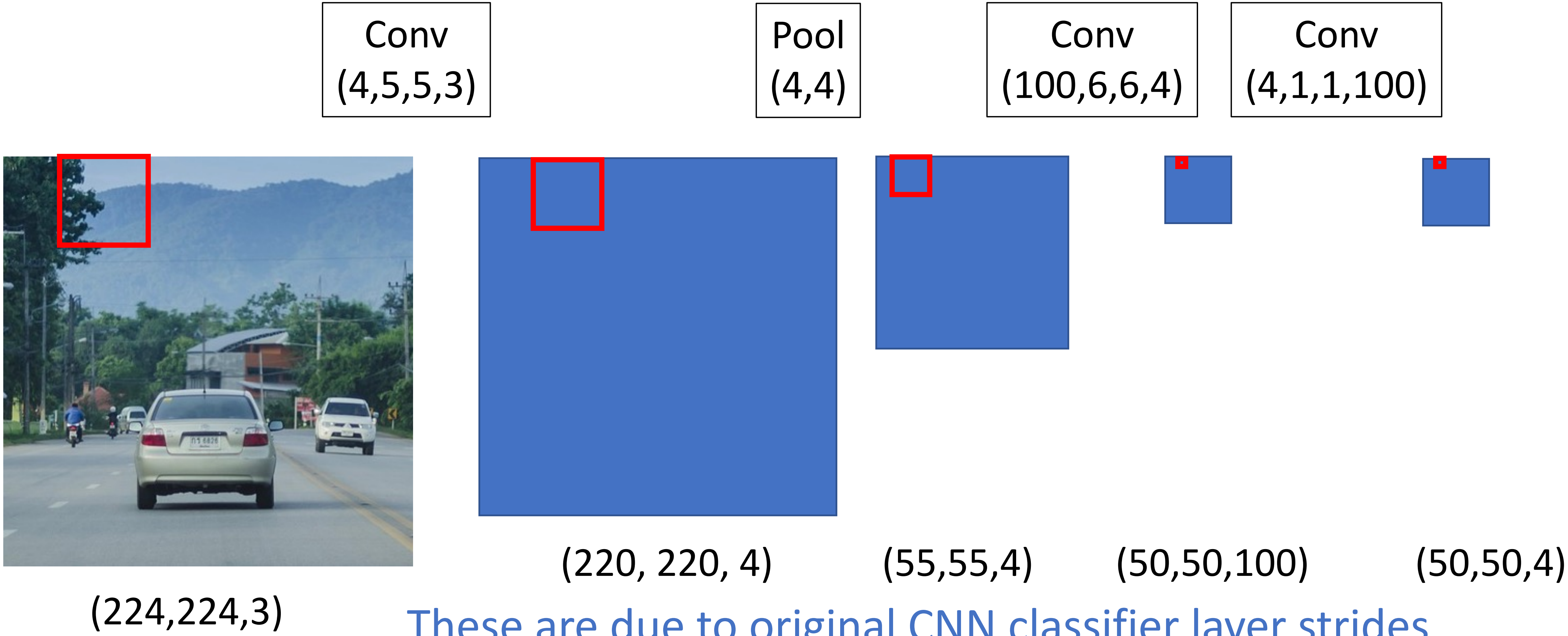
(55,55,4)

(50,50,100)

(50,50,4)

These are due to original CNN classifier dimensions

(28,28) Window	(24,24) Window	(6,6) Window	(1,1) Window
Stride = 4	Stride = 4	Stride = 1	Stride = 1
(50,50) Locations	(50,50) Locations	(50,50) Locations	(50,50) Locations

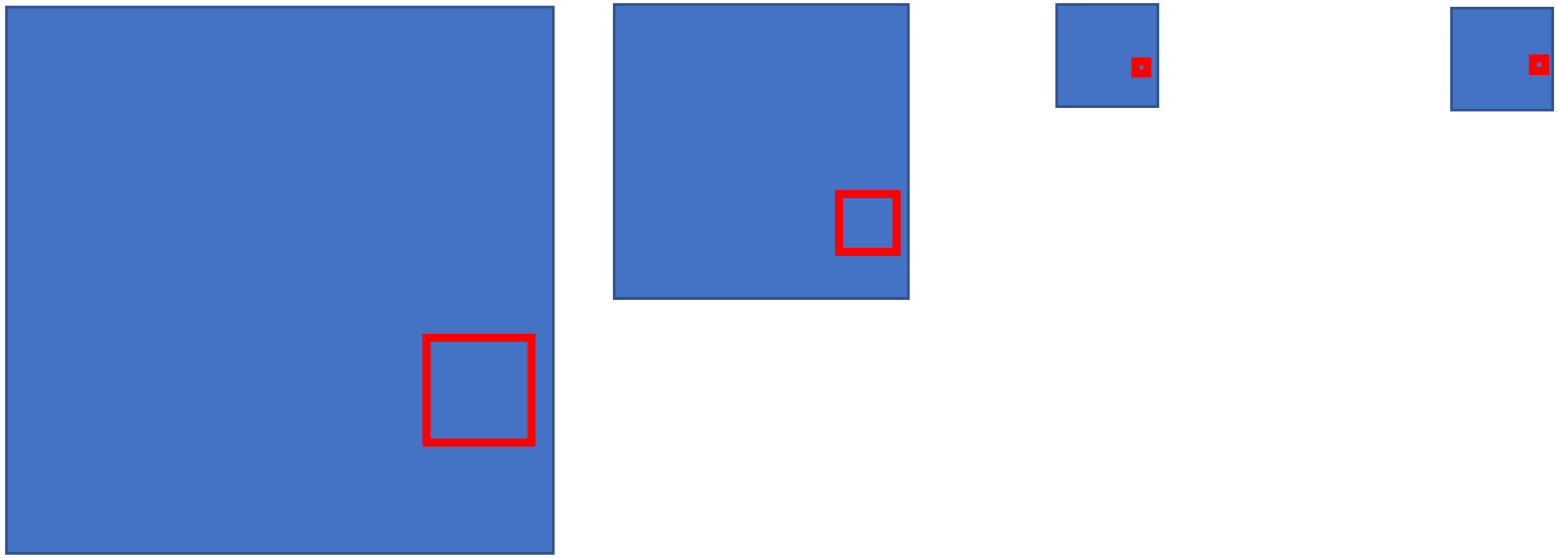


These are due to original CNN classifier layer strides

(28,28) Window	(24,24) Window	(6,6) Window	(1,1) Window
Stride = 4	Stride = 4	Stride = 1	Stride = 1
(50,50) Locations	(50,50) Locations	(50,50) Locations	(50,50) Locations

Conv (4,5,5,3) | Pool (4,4) | Conv (100,6,6,4) | Conv (4,1,1,100)

(224,224,3)



(220, 220, 4) (55,55,4) (50,50,100) (50,50,4)

(28,28) Window
Stride = 4
(50,50) Locations

(24,24) Window
Stride = 4
(50,50) Locations

(6,6) Window
Stride = 1
(50,50) Locations

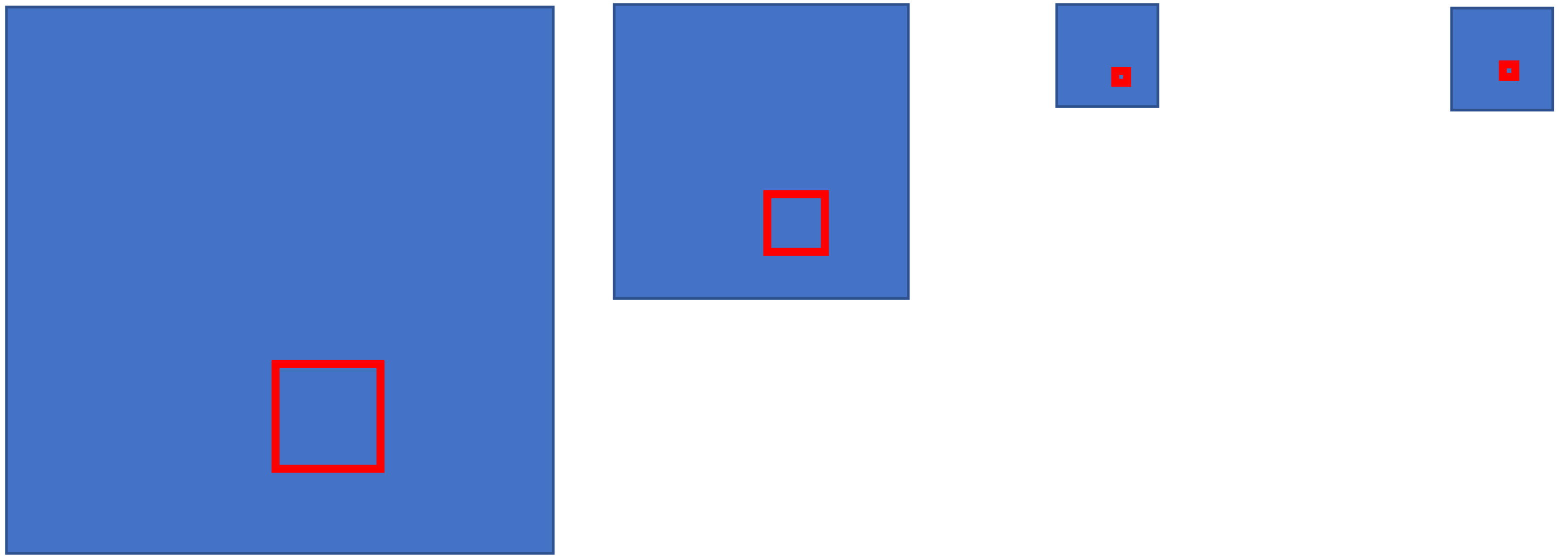
(1,1) Window
Stride = 1
(50,50) Locations

Conv (4,5,5,3)

Pool (4,4)

Conv (100,6,6,4)

Conv (4,1,1,100)



(224,224,3)

(220, 220, 4)

(55,55,4)

(50,50,100)

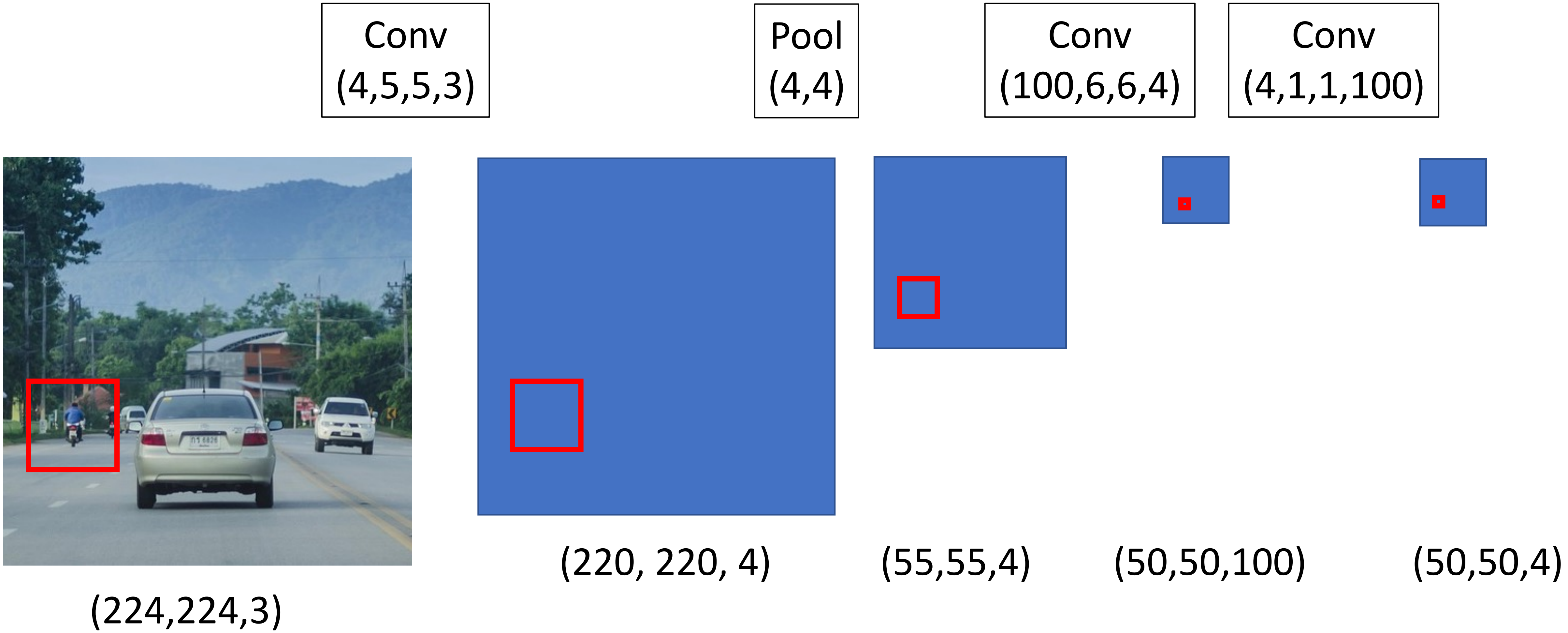
(50,50,4)

(28,28) Window
Stride = 4
(50,50) Locations

(24,24) Window
Stride = 4
(50,50) Locations

(6,6) Window
Stride = 1
(50,50) Locations

(1,1) Window
Stride = 1
(50,50) Locations



(28,28) Window
Stride = 4
(50,50) Locations

(24,24) Window
Stride = 4
(50,50) Locations

(6,6) Window
Stride = 1
(50,50) Locations

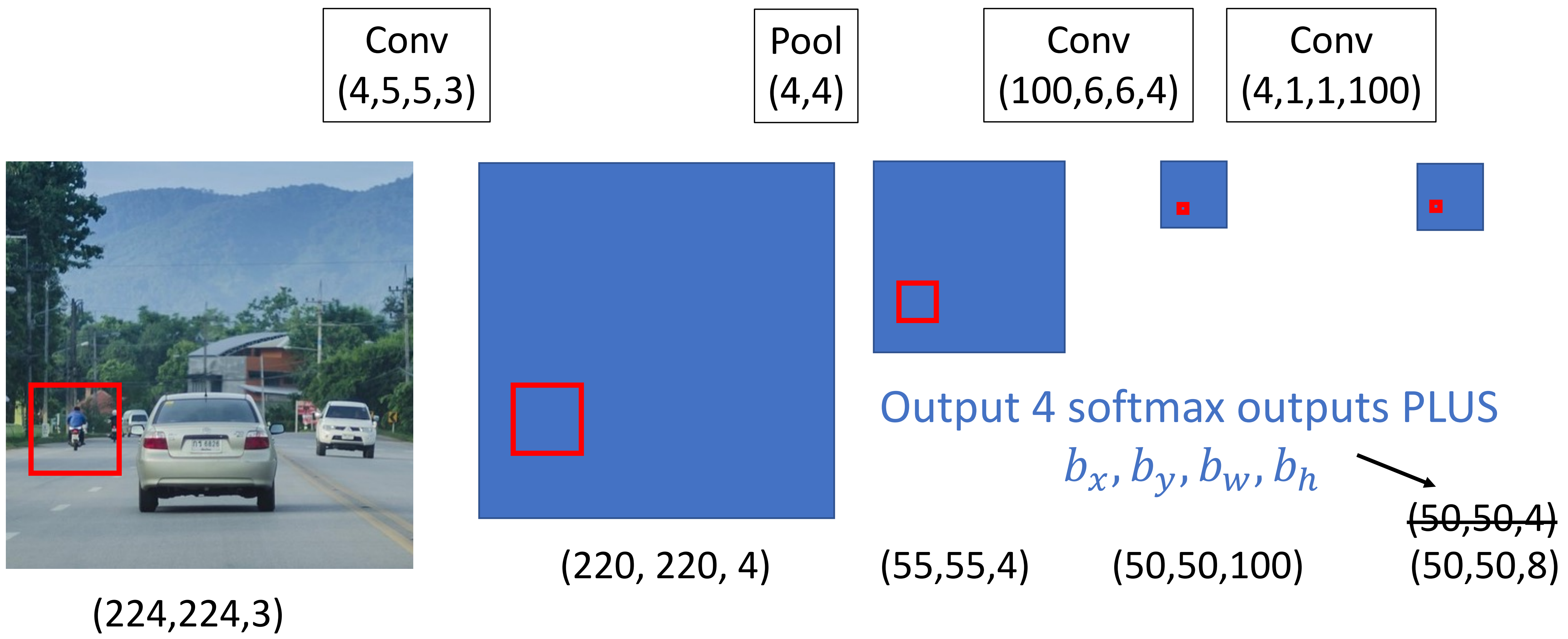
(1,1) Window
Stride = 1
(50,50) Locations

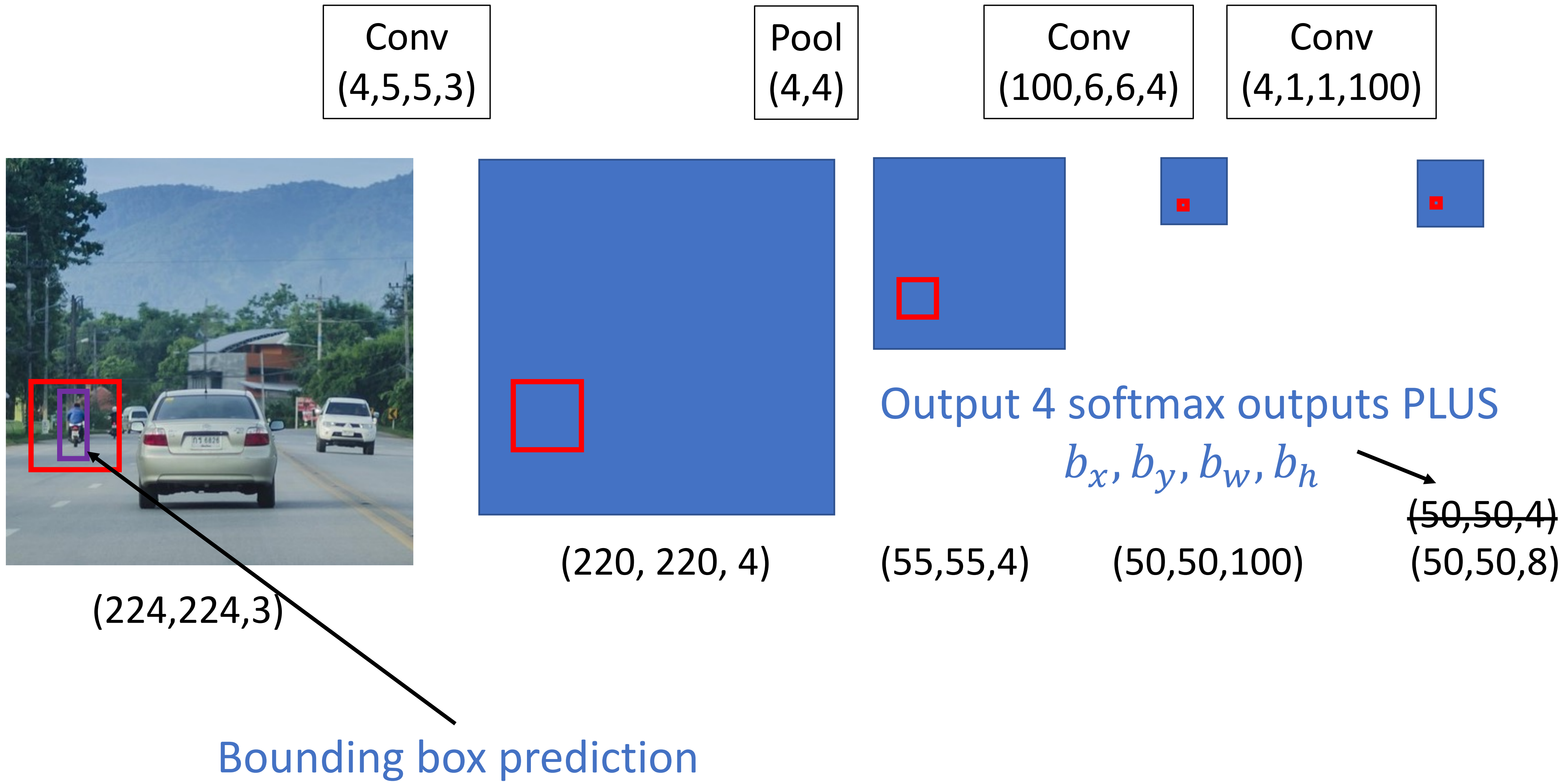
Problem of Using Sliding Window to define bounding boxes

- Objects may not fit perfectly inside of sliding window
  → Inaccurate bounding box predictions

Solution

- Instead of applying a CNN classifier at each sliding window location, apply a CNN classifier+localizer
  - i.e. outputs a bounding box prediction in addition to class predictions


Conv
(4,5,5,3)
Pool
(4,4)
Conv
(100,6,6,4)
Conv
(4,1,1,100)
Output 4 softmax outputs PLUS
b_x, b_y, b_w, b_h
~~(50,50,4)~~
(224,224,3)
(220, 220, 4)
(55,55,4)
(50,50,100)
(50,50,8)


Conv
(4,5,5,3)
Pool
(4,4)
Conv
(100,6,6,4)
Conv
(4,1,1,100)
Output 4 softmax outputs PLUS
b_x, b_y, b_w, b_h
~~(50,50,4)~~
(224,224,3)
(220, 220, 4)
(55,55,4)
(50,50,100)
(50,50,8)
Bounding box prediction

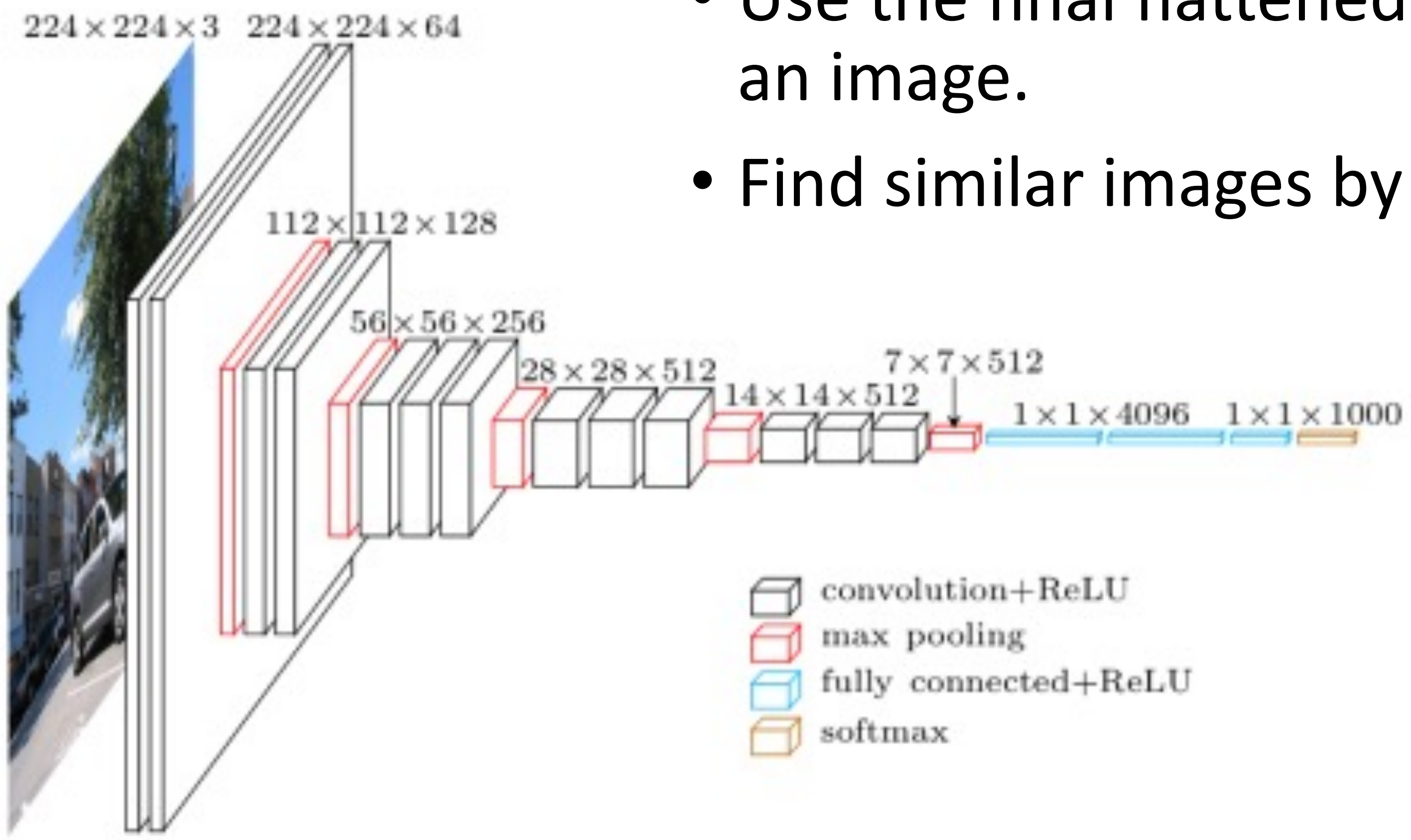
Detecting Multiple Objects in Same Sliding Window Location

- So far, can only detect one object at each sliding window location.
- Also doesn't seem like it could work too well for objects that are bigger than the sliding window
- YOLO uses something called Anchor Boxes
- Change localizer to predict up to X (e.g. 5 in YOLO) objects at each location with predefined bounding box shapes
- Refer to paper for more information

Brad Quinton, Scott Chin

Applications Beyond Classification and Detection

Image Retrieval



- Use the final flattened volume as a “signature” of an image.
- Find similar images by finding similar signatures

Image source: Davi Frossard https://www.cs.toronto.edu/~frossard/post/vgg16/

Image Retrieval

- With a trained network, compute and store signature vector of each image
- Given a new image, find images with the smallest Euclidian distance between signature vectors

"ImageNet Classification with Deep Convolutional Neural Networks", Krizhevsky, Sutskever, Hinton, 2012, https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Image Retrieval

- With a trained network, compute and store signature vector of each image
- Given a new image, find images with the smallest Euclidian distance between signature vectors

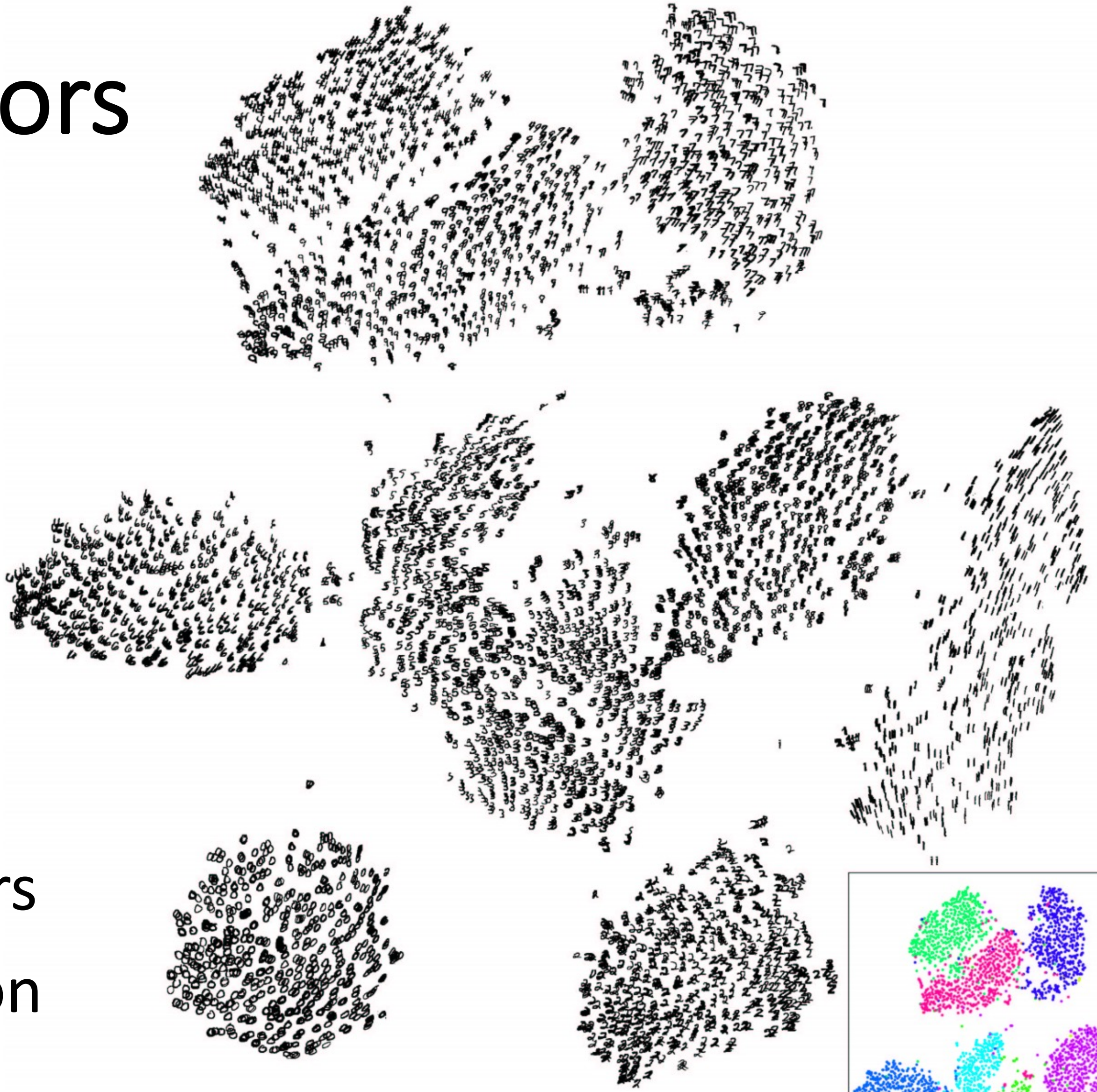
"ImageNet Classification with Deep Convolutional Neural Networks", Krizhevsky, Sutskever, Hinton, 2012, https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

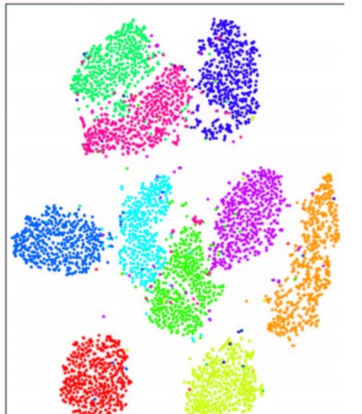
Visualization Feature Vectors

- Last volume flattened out
- Apply dimension reduction (e.g. Principal Component Analysis, t-SNE)
- Plot

Example

- 50,000 MNIST images
- Lenet5 produces 120 dimension vectors
- Using t-SNE to project onto 2 dimension



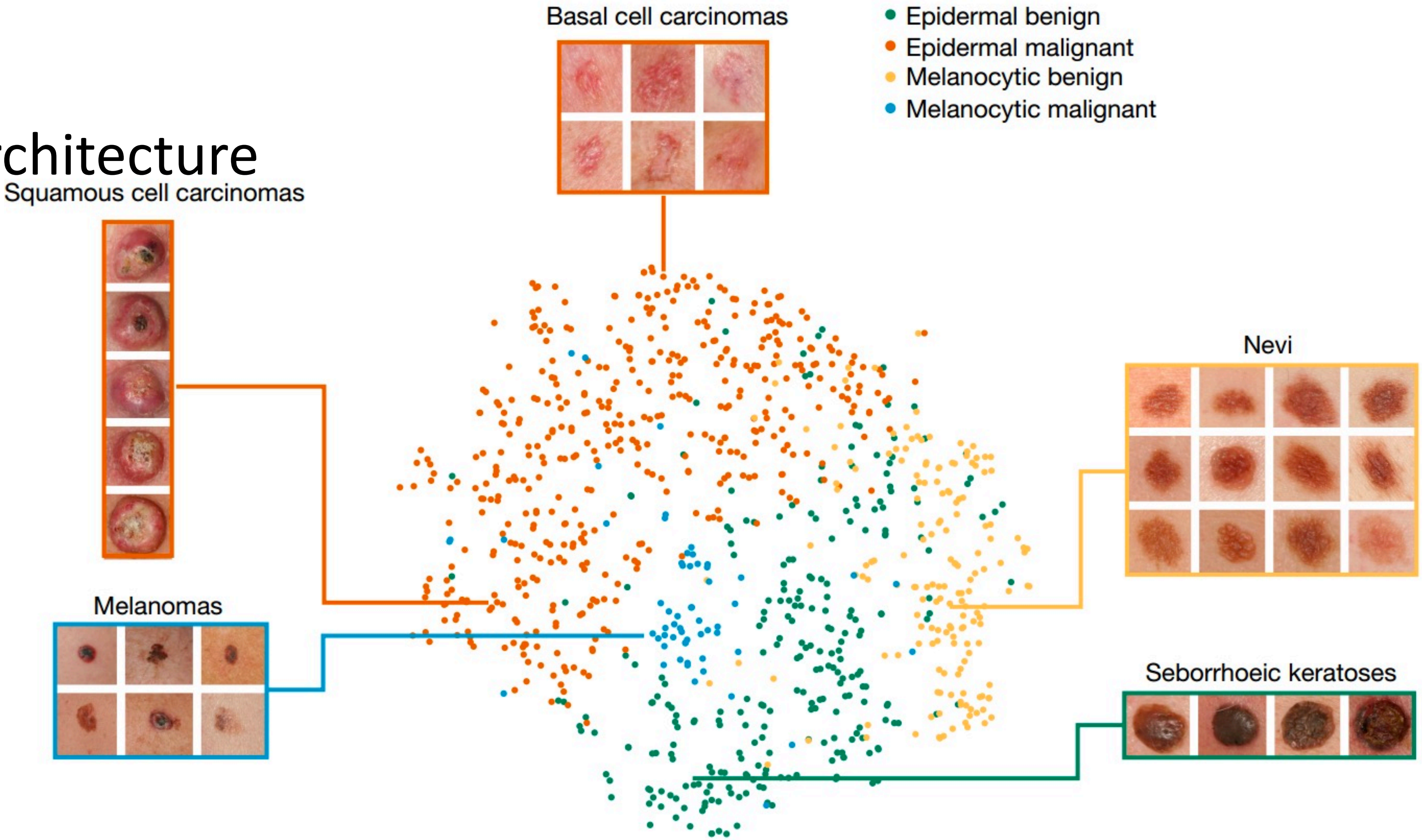


"Visualizing Data Using t-SNE", van der Maaten, Hinton, 2008, "http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf"

Visualization Feature Vectors

Example

- InceptionV3 CNN architecture
- (1024,) vectors
- Skin Cancer Data

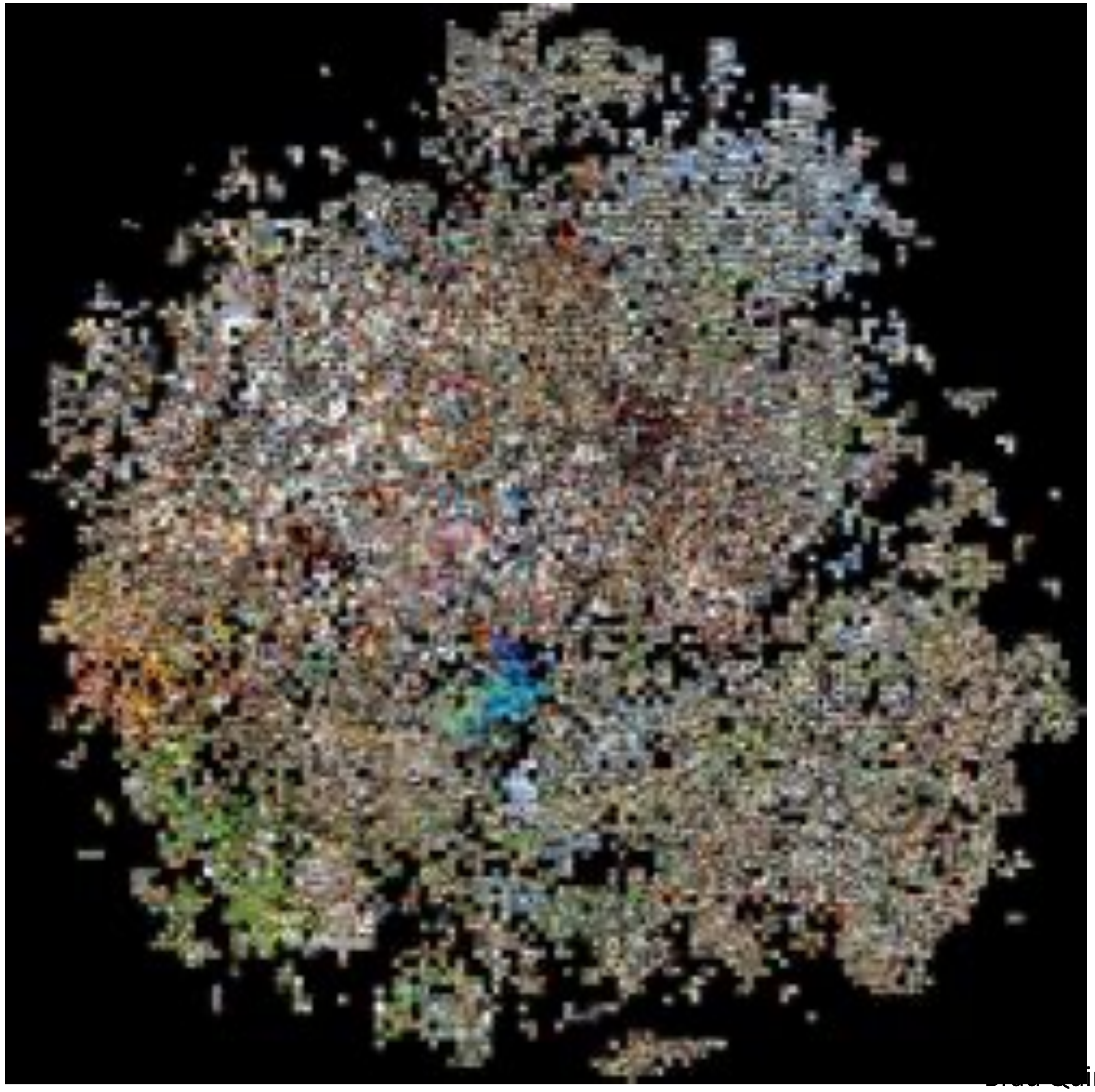


"Dermatologist-Level Classification of skin cancer with deep neural networks", Esteva et al., 2017, https://www.andreesteva.com/assets/nature_skincancer.pdf

Visualization Feature Vectors

Example

- AlexNet CNN architecture
- (1024,) vectors
- ImageNet Data



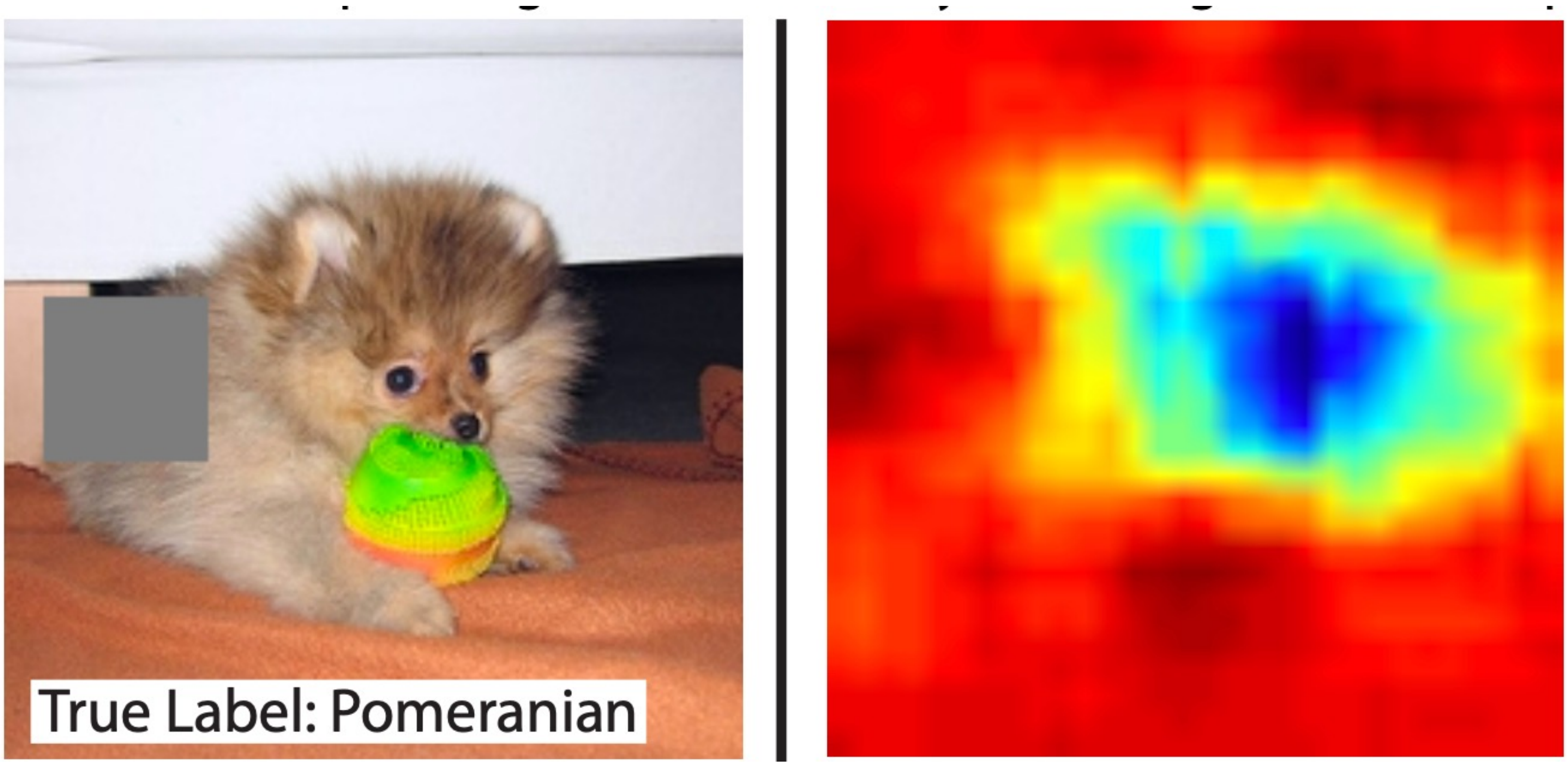
https://cs.stanford.edu/people/karpathy/cnnembed/

Saliency Maps

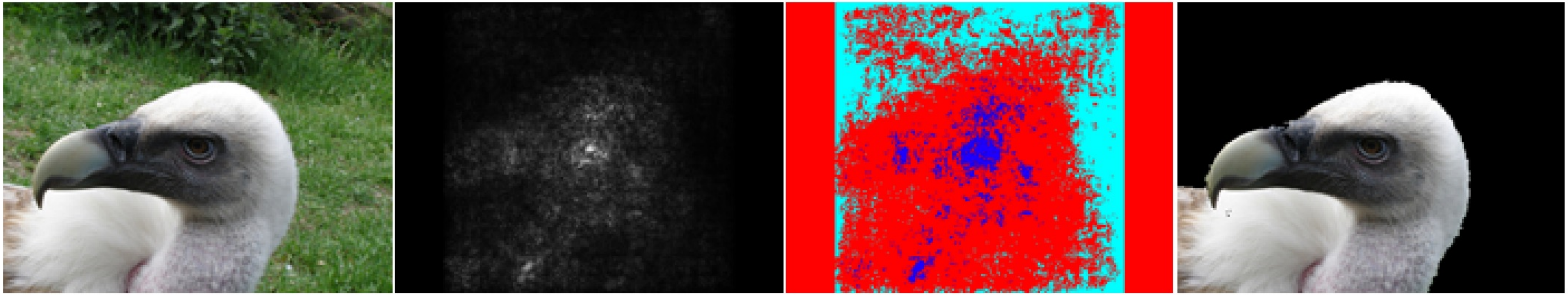
- What parts of the image were import for the prediction?

“Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, Simonyan, et al, 2014, https://arxiv.org/abs/1312.6034

Saliency via Occlusion



Segmentation with Saliency Maps



Learning Objectives

- Look at a few more successful CNN architectures
- Learn about Spatially-Separable and Depthwise-Separable Convolutions
- Introduction to Object detection
- Sliding window via convolution
- Quick introduction to other vision applications beyond classification
  - Localization
  - Landmark detection
  - Face detection
  - Pose detection
  - Image retrieval
  - Visualization
  - Segmentation