

Topic 5

كلية الهندسة

الذكاء الصناعي العملي

Modern Computer Vision Architectures: From CNNs to Transformers

د. رياض سنبل

Computer Vision Tasks

Classification



CAT

No spatial extent

Semantic Segmentation



GRASS, CAT, TREE,
SKY

No objects, just pixels

Object Detection



DOG, DOG, CAT

Multiple Object

Instance Segmentation



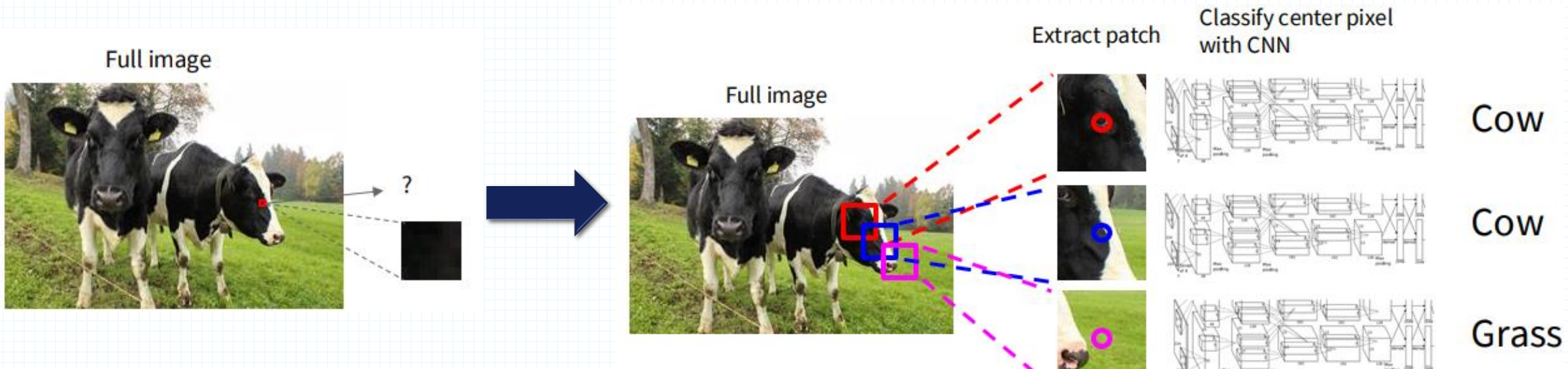
DOG, DOG, CAT

[This image is CC0 public domain](#)

Semantic Segmentation

Semantic Segmentation

Idea: Sliding Window

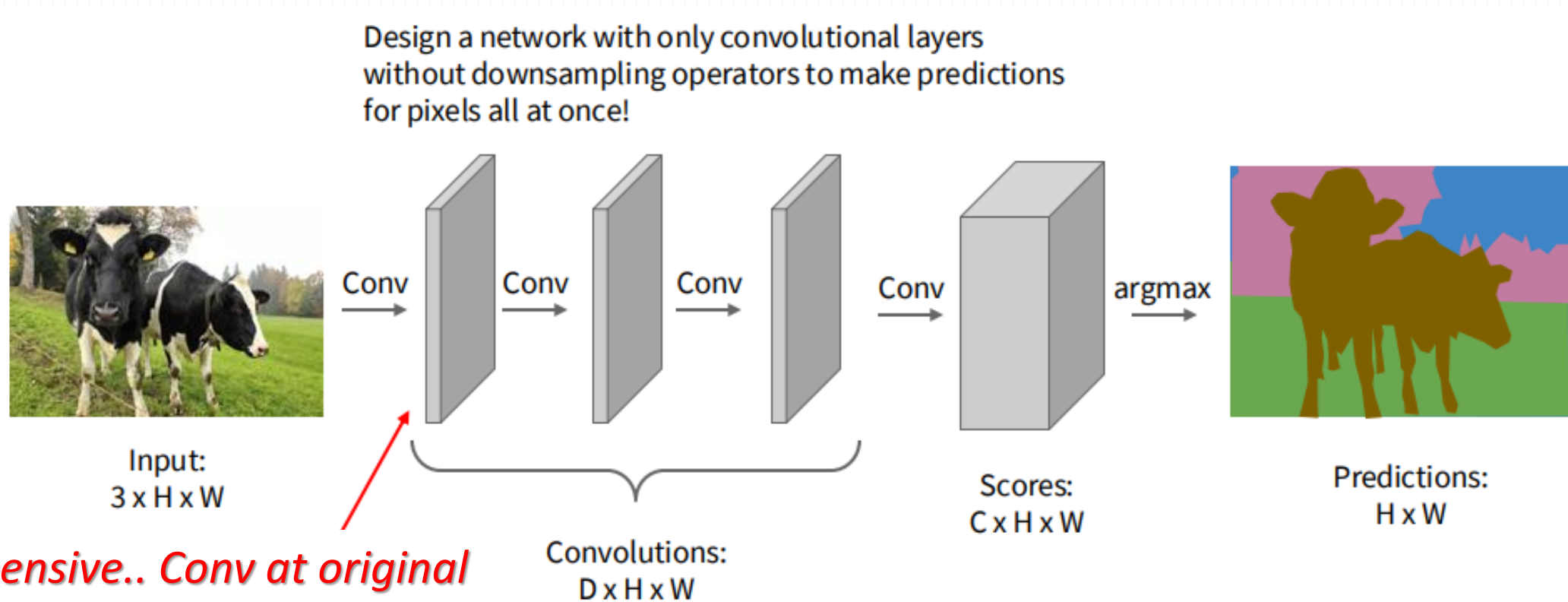


*Impossible to classify
without context!*

*But.. Inefficient!
(many overlapping patches)*

Semantic Segmentation

Idea: Convolution!



Still expensive.. Conv at original image resolution will be very expensive

Semantic Segmentation

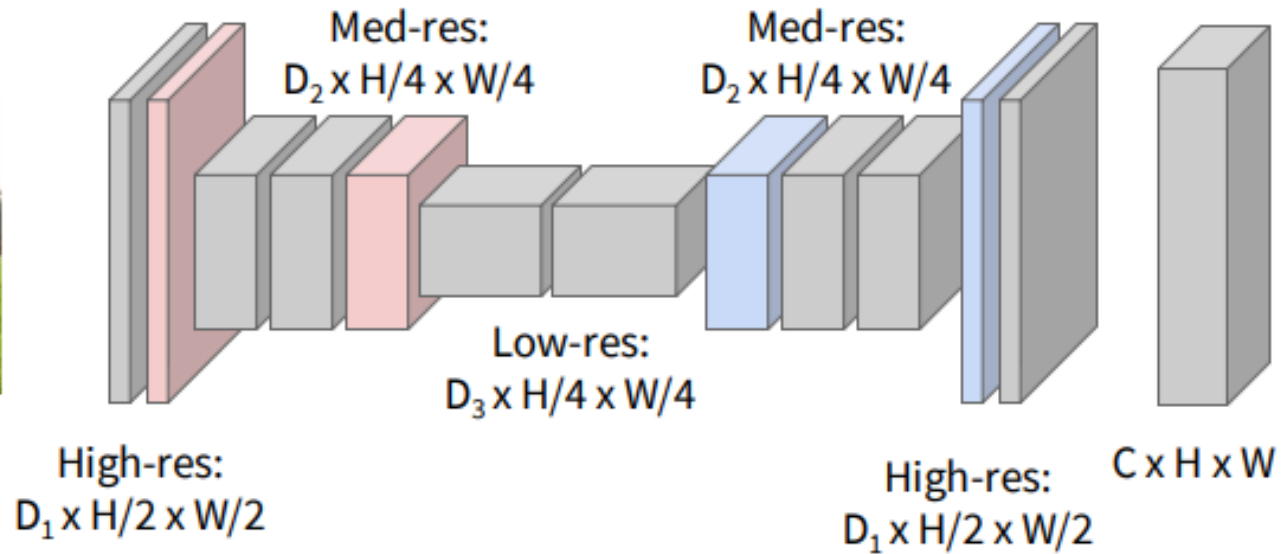
Solution: U-Net?

Downsampling:
Pooling, strided
convolution

Design network as a bunch of convolutional layers, with
downsampling and upsampling inside the network!



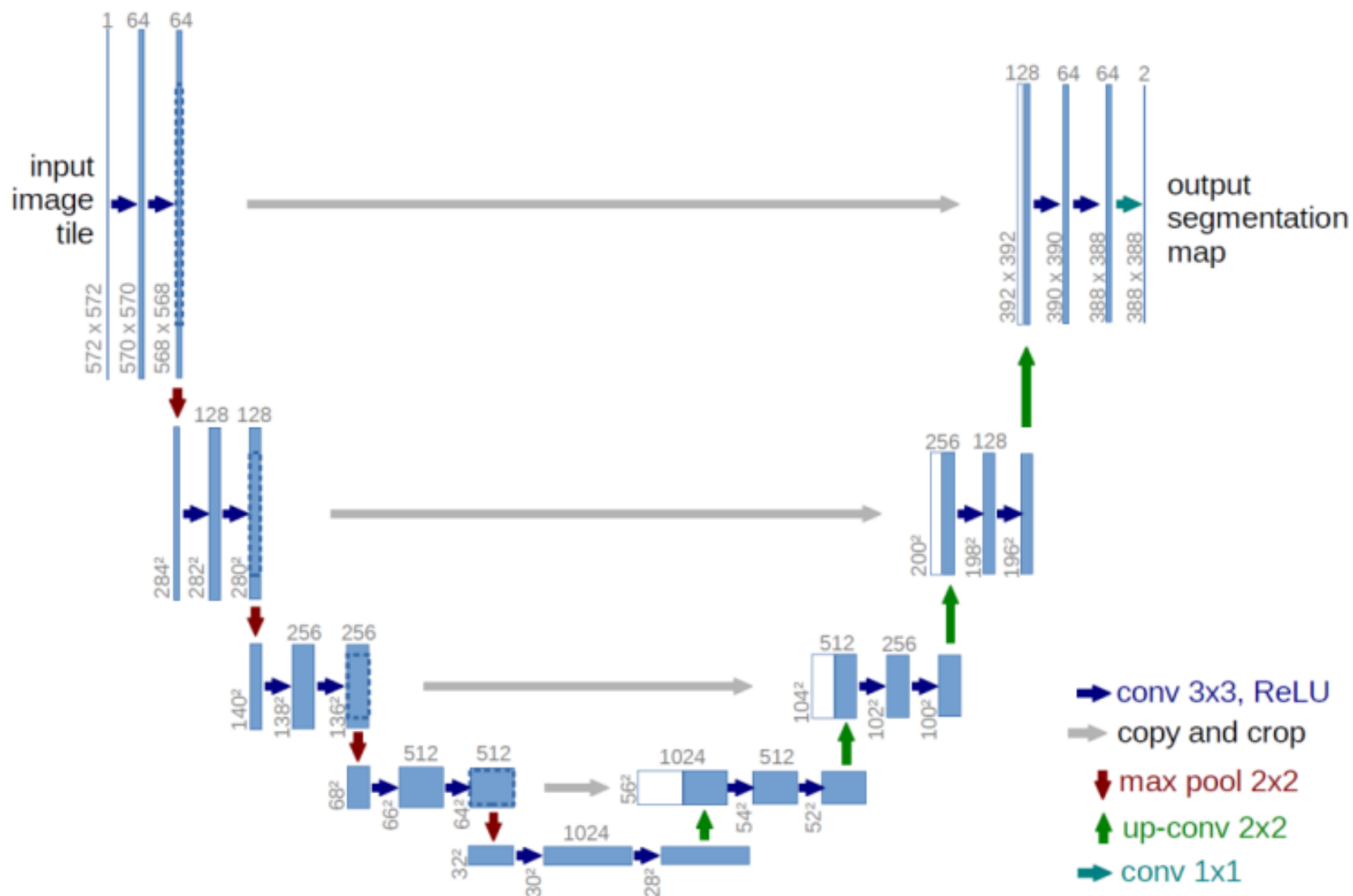
Input:
 $3 \times H \times W$



Predictions:
 $H \times W$

Semantic Segmentation

Solution: U-Net?

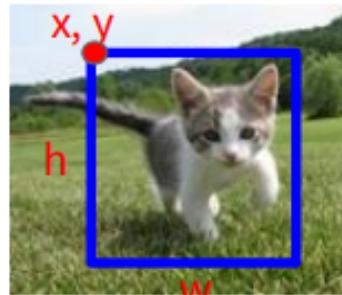


Object Detection

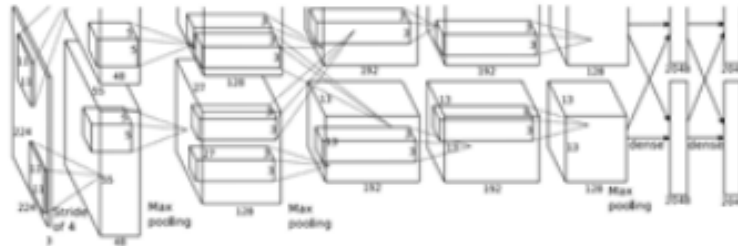


Single Object Detection

(Classification + Localization)



[This image is CC0 public domain](#)



Vector:
4096

Fully
Connected:
4096 to 1000

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

Multitask Loss

Fully
Connected:
4096 to 4

Box
Coordinates
(x, y, w, h)

Correct label:
Cat

Softmax
Loss

+ → Loss

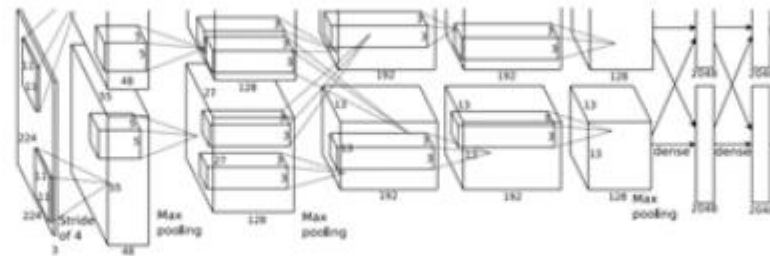
L2 Loss

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

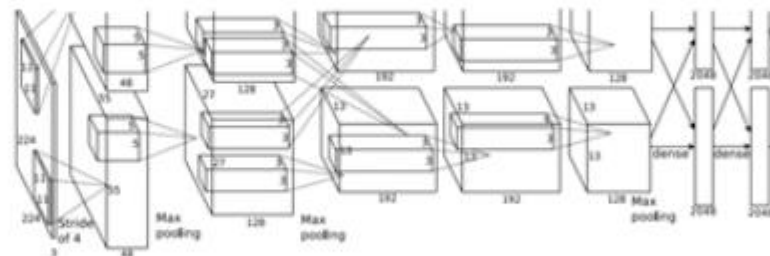
Treat localization as a
regression problem!

Multiple Object Detection!

- Sliding window Solution!



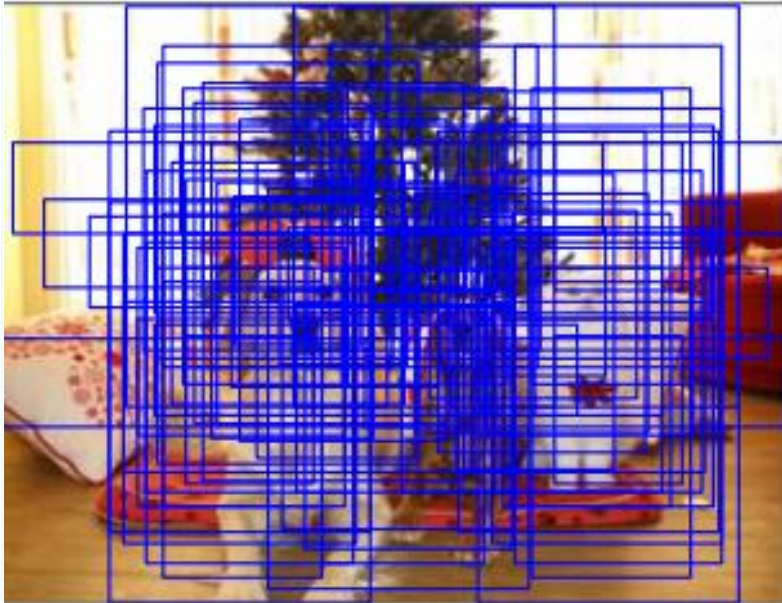
Dog? NO
Cat? NO
Background? YES



Dog? YES
Cat? NO
Background? NO

Multiple Object Detection!

- Sliding window Solution!

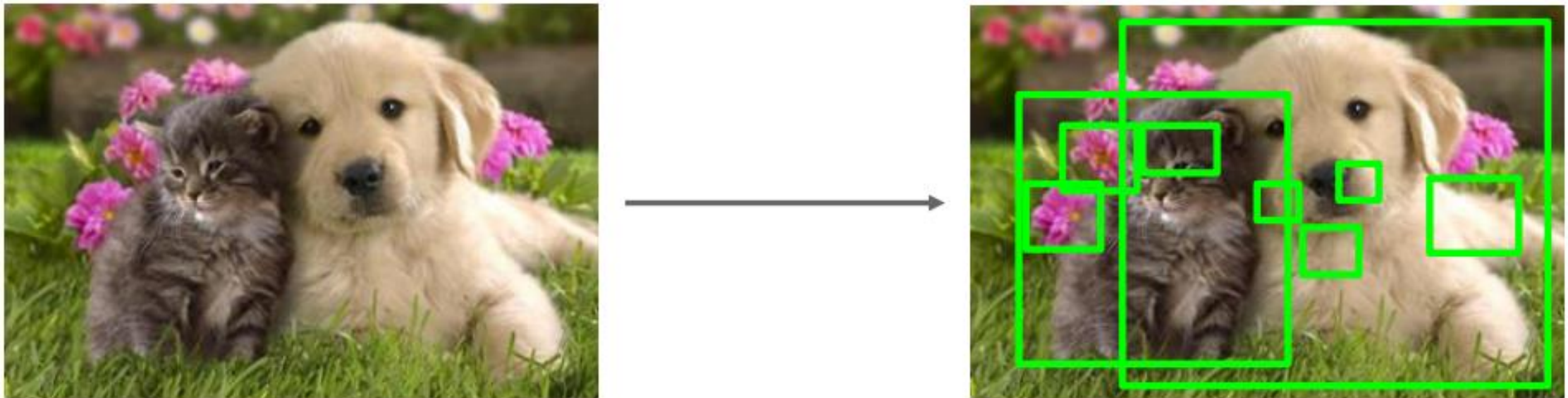


Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

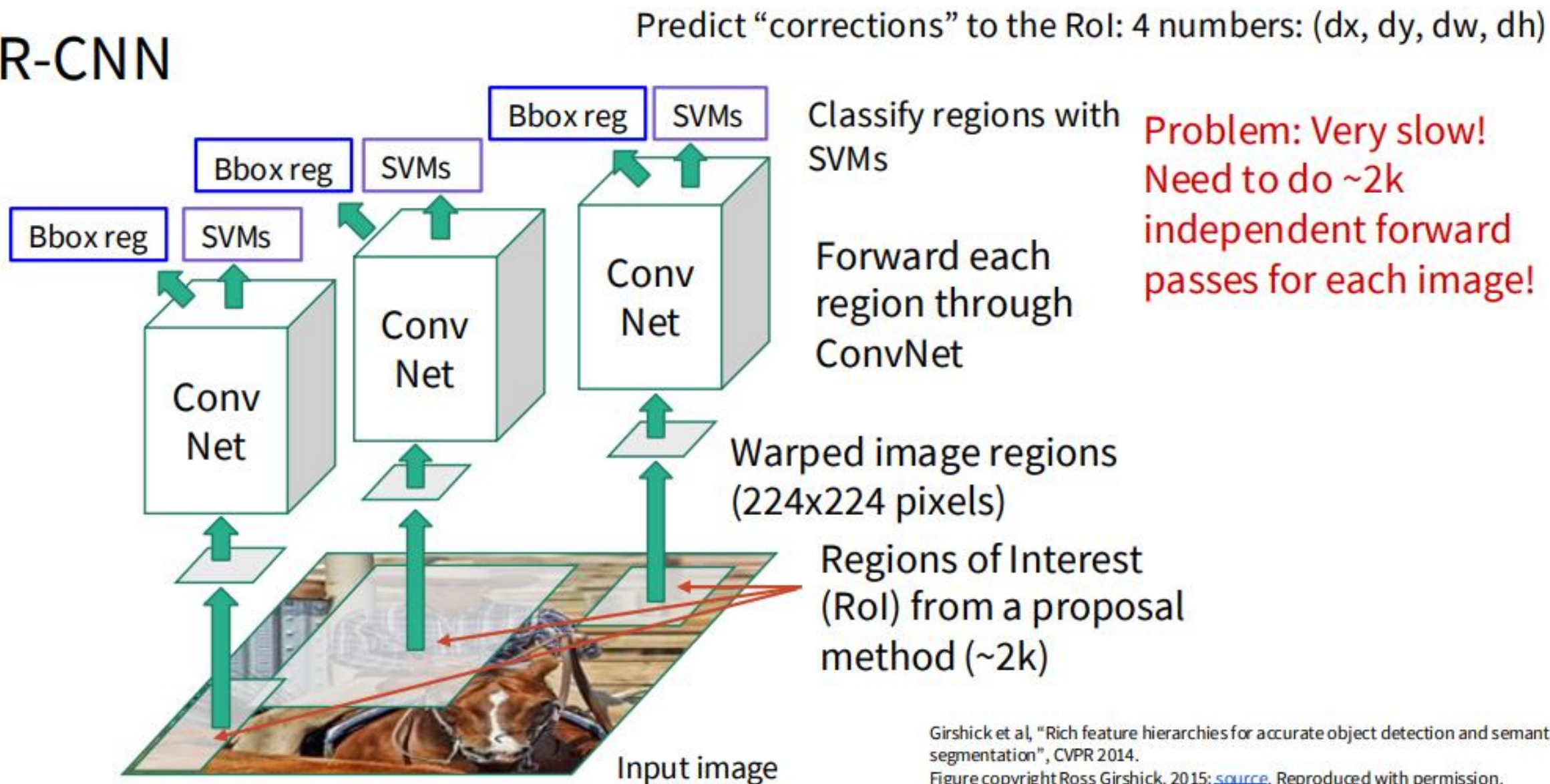
Multiple Object Detection

Better Solution – R CNN (Region Proposals)

- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



R-CNN



Multiple Object Detection – R-CNN

(1) Generate Region Proposal

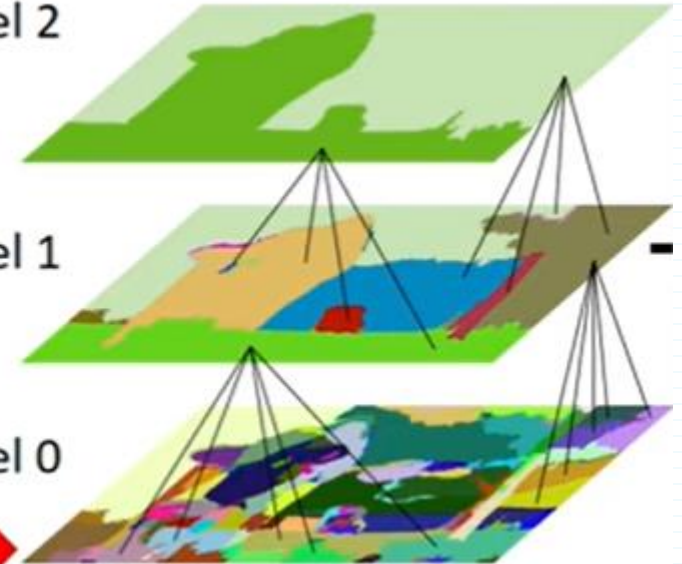
Original Image



Level 2

Level 1

Level 0



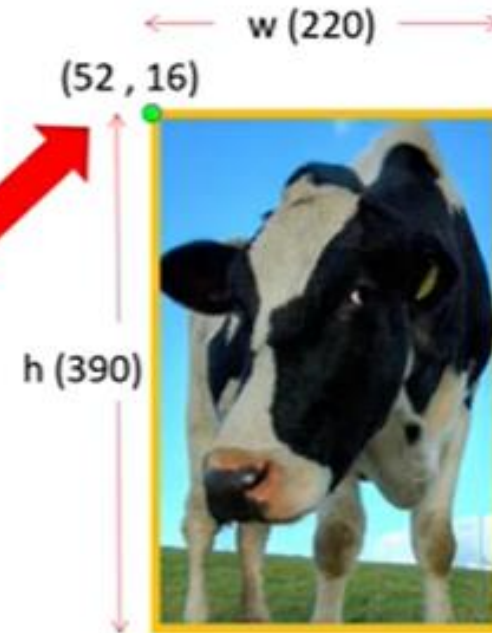
Output of the selective search is region proposals.

Region	x	y	w	h	level
r_1	21	35	10	12	0
r_2	...	...	...		0
r_3	...				0
					0

Multiple Object Detection – R-CNN

(2) Generate Region Proposal

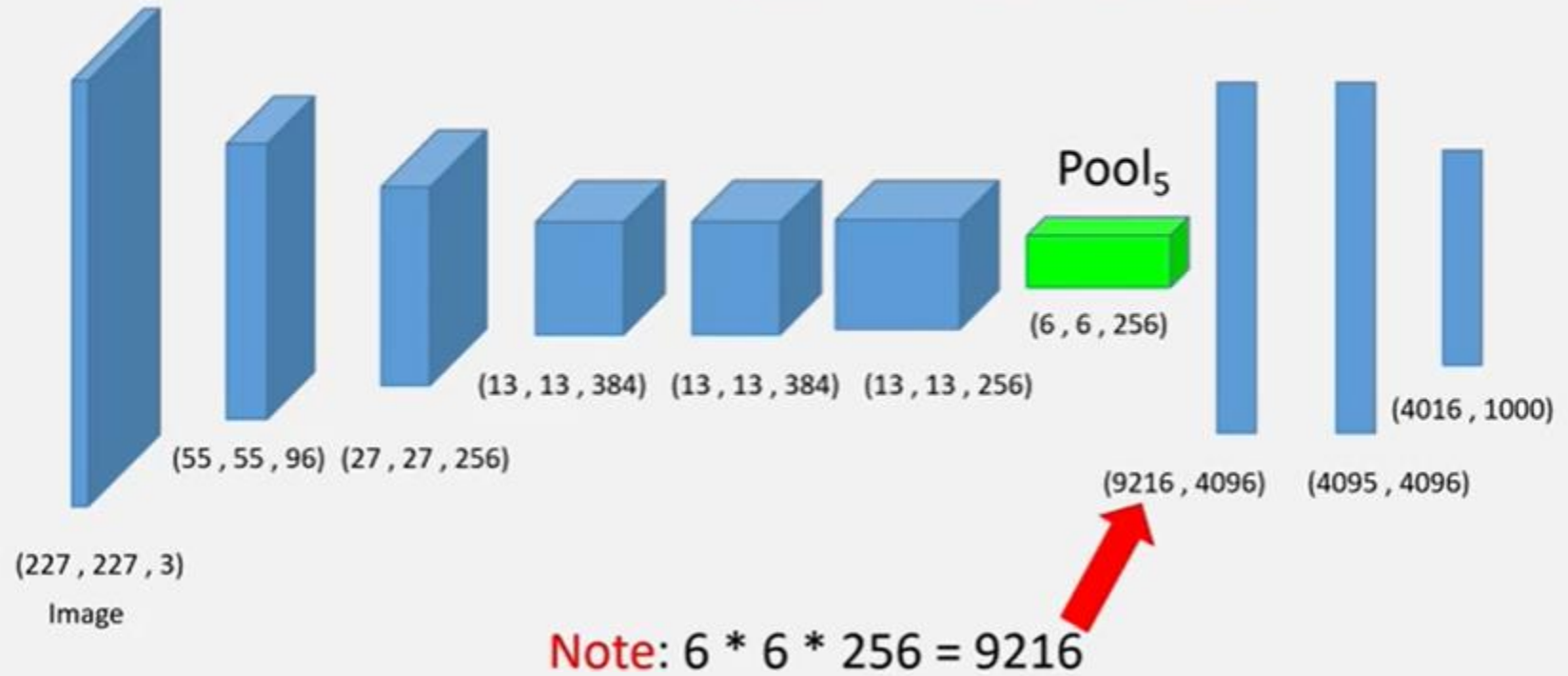
Region	x	y	w	h	level
r_1	21	35	10	12	0
r_2	...	...	...		0
r_3	...				0
...		...			0
...			...		0
r_n				...	0
r_{n+1}	52	16	220	390	1
...	...				1
r_m		...			2



Multiple Object Detection – R-CNN

(2) Classify Region Proposal

Extracting CNN features.



Fine Tune AlexNet.

last layer.

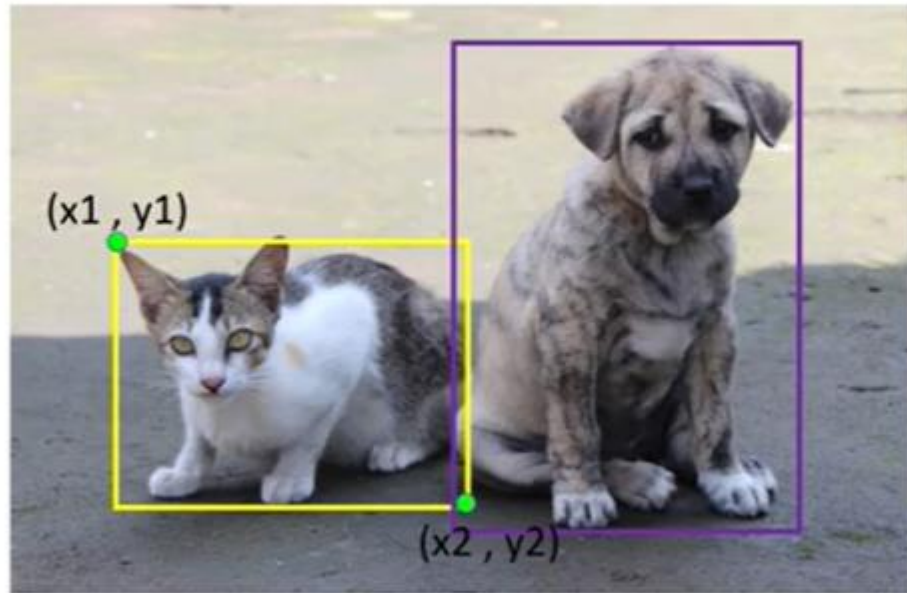


1. Remove last layer and replace by a new layer.
2. Redefine dataset and classify region proposals (RP).

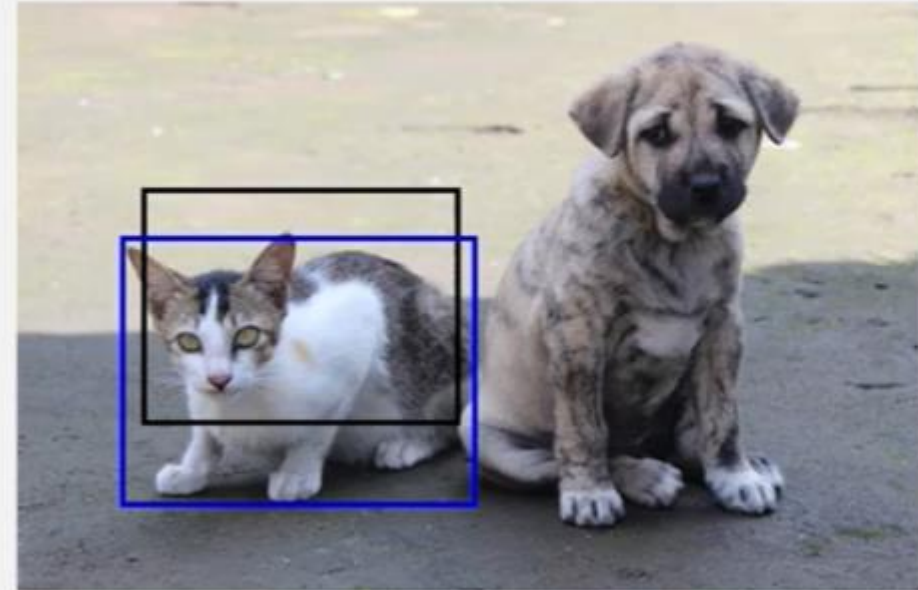
Multiple Object Detection – R-CNN

(2) Classify Region Proposal (Dataset?)

Ground Truth bounding boxes.



Region Proposal bounding boxes.



IOU = 0.7

Positive for cat

Ground truth bound

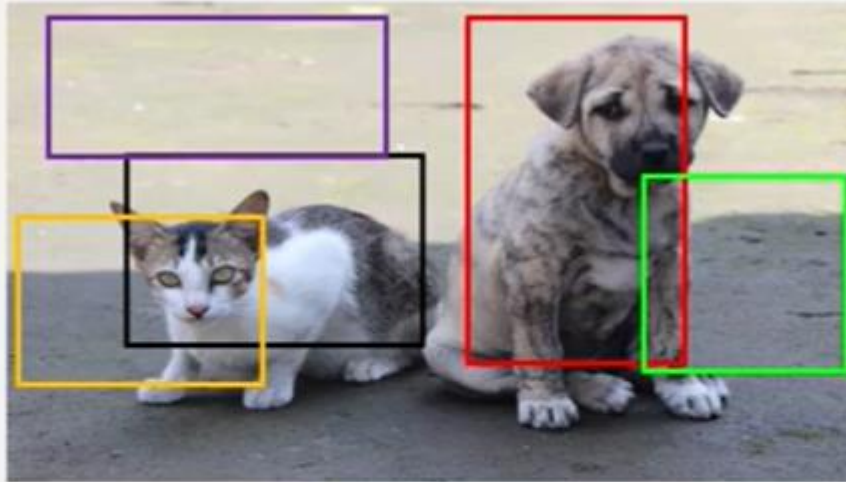
Classify Region Proposals.

Find Intersection Over Union (IOU) with Ground Truth Bounding Box

If $\text{IOU} > 0.5$ --> Positive for that class , else negative

Multiple Object Detection – R-CNN

(2) Classify Region Proposal (Dataset?)

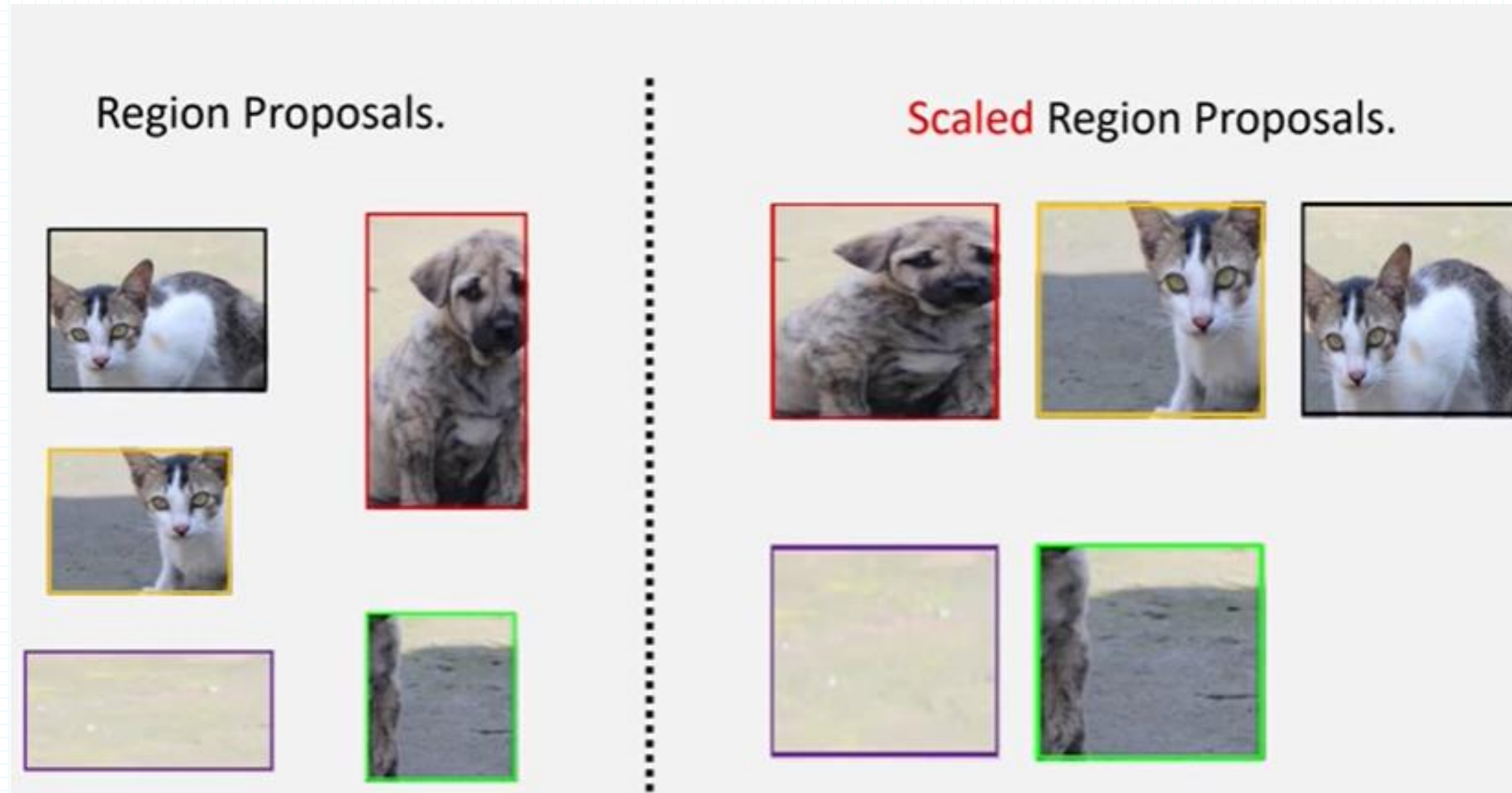


Dataset

Region	x1	y1	x2	y2	IOU	class (neuron)
Red	300	20	70	290	0.8	dog (1)
Green	375	180	60	70	0.3	negative (3)
Black	...	...	...	...	0.7	cat (2)
orange	...	...	...	...	0.4	negative (3)
purple	...	...	...	...	0.0	negative (3)

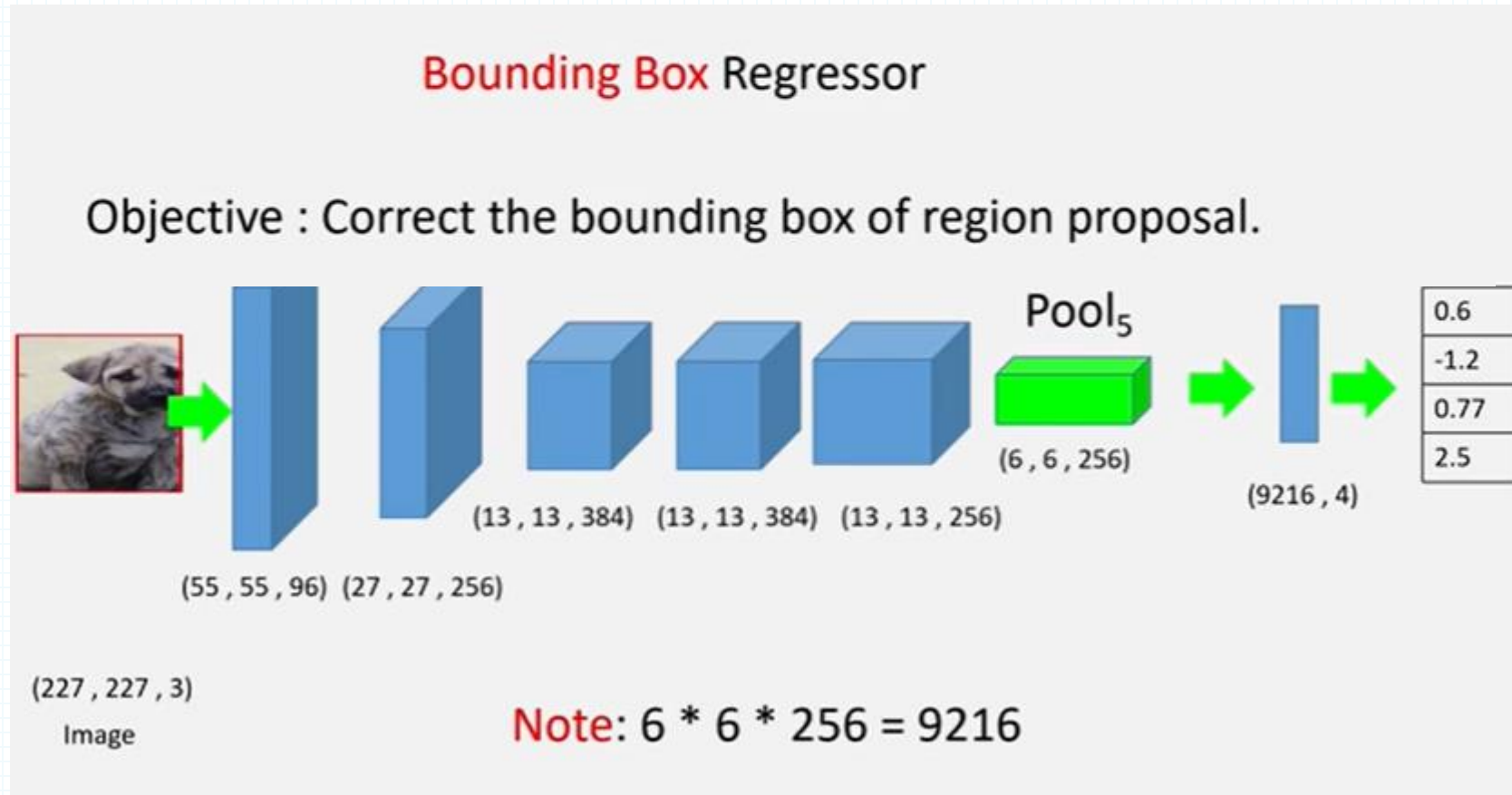
Multiple Object Detection – R-CNN

(2) Classify Region Proposal (Dataset?)



Multiple Object Detection – R-CNN

Extra Step.. Bounding Box Correction

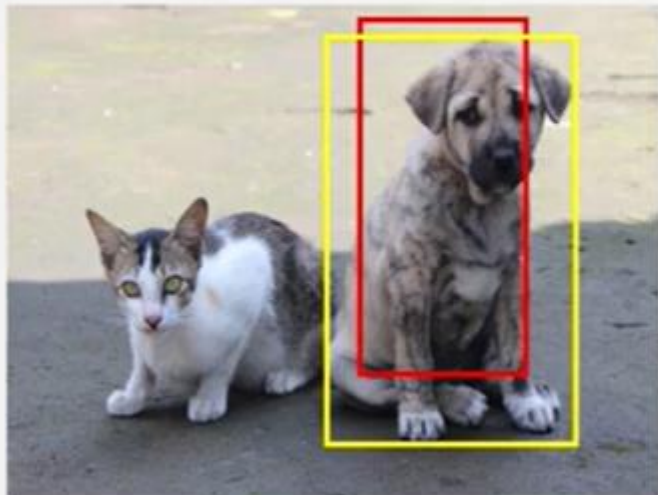


$$t_x = (G_x - P_x) / P_w$$
$$t_y = (G_y - P_y) / P_h$$
$$t_w = \log(G_w / P_w)$$
$$t_h = \log(G_h / P_h).$$

Multiple Object Detection – R-CNN

Extra Step.. Bounding Box Correction (Dataset?)

	Region	x1	y1	x2	y2	IOU	class (neuron)
✓	Red	300	20	70	290	0.8	dog
✗	Green	375	180	60	70	0.3	negative
✓	Black	...	...	...	...	0.7	cat
✗	orange	...	...	...	...	0.4	negative
✗	purple	...	...	...	...	0.0	negative



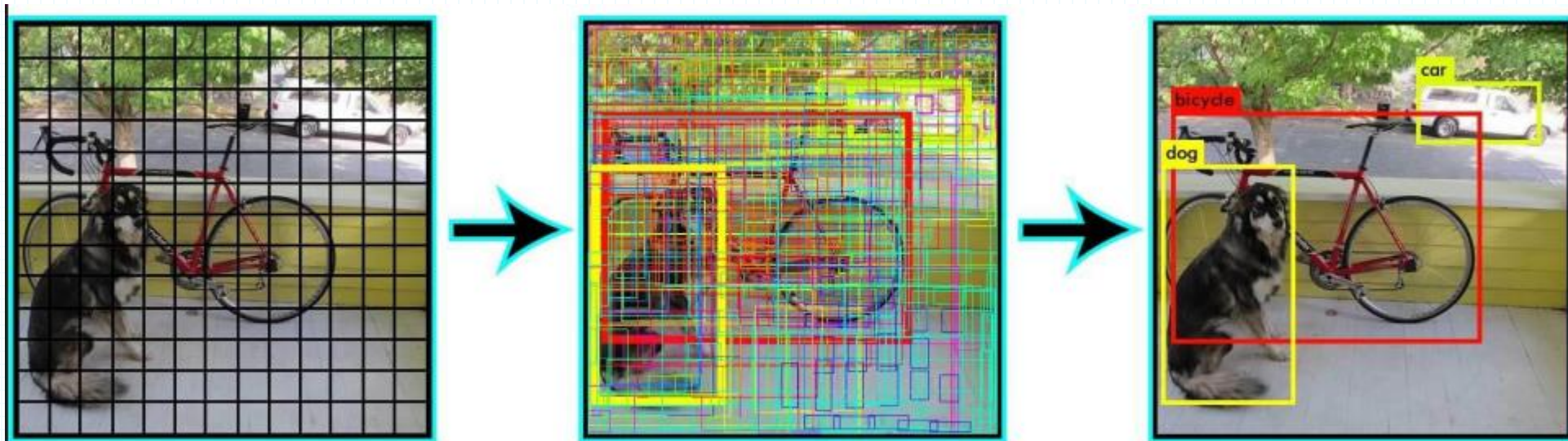
Ground truth



Region Proposal



YOLO



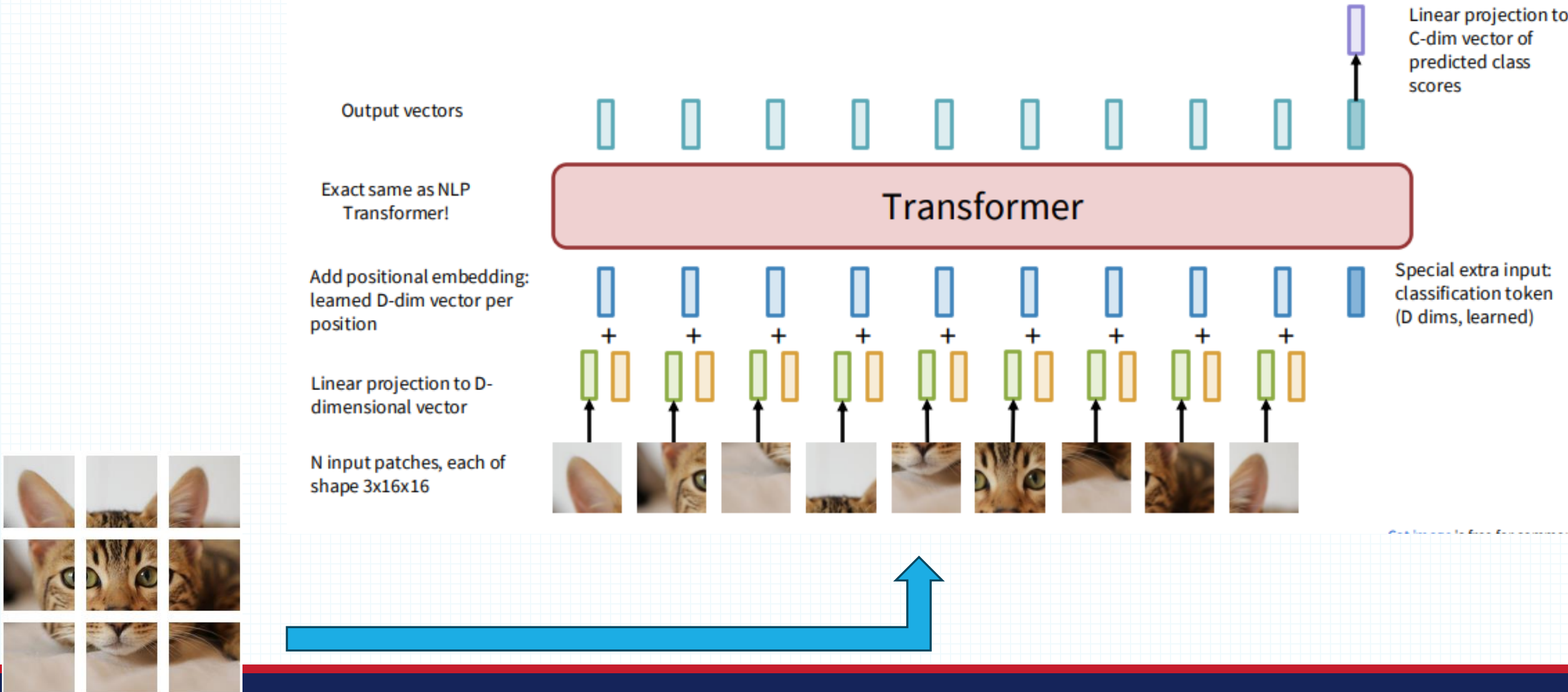
SxS Grid

For each box output:

- $P(\text{object})$: probability that the box contains an object
- B bounding boxes (x, y, h, w)
- $P(\text{class})$: probability of belonging to a class

Transformers in CV

ViT as backbone



Object Detection with Transformers (DETR)

