Extra Class

Naïve Object Detection

Nguyen Quoc Thai

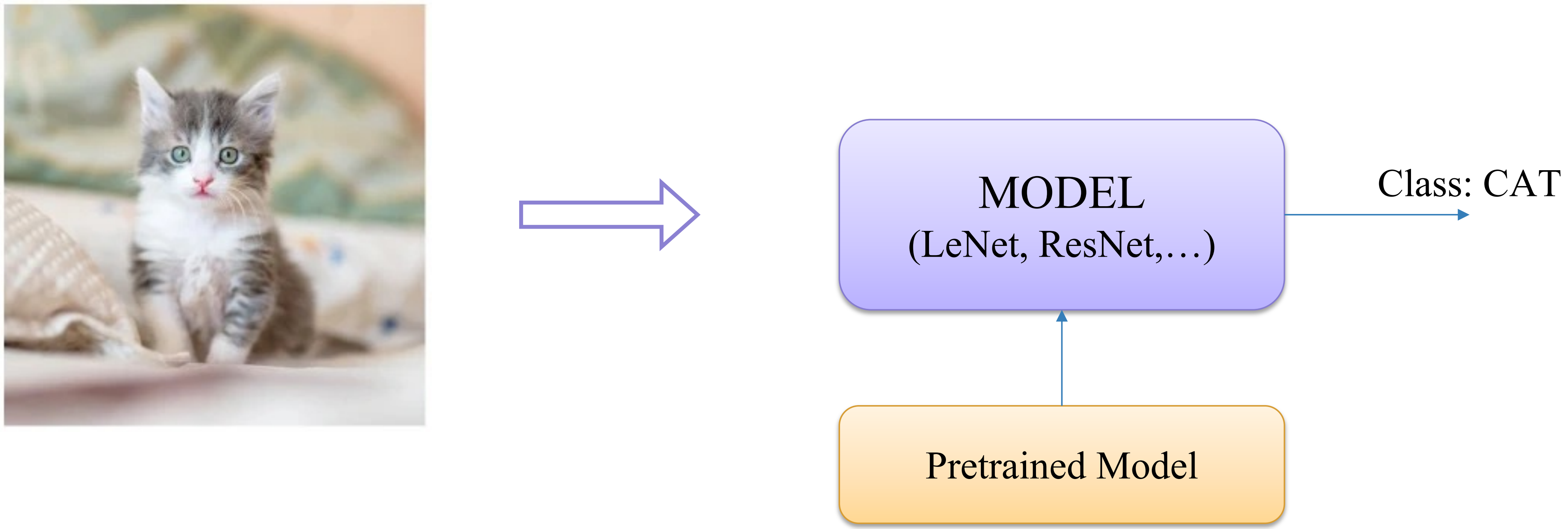
CONTENT

(1) – Object Detection

(2) – Naïve Object Detection (as Classification)

1 – Object Detection

Image Classification



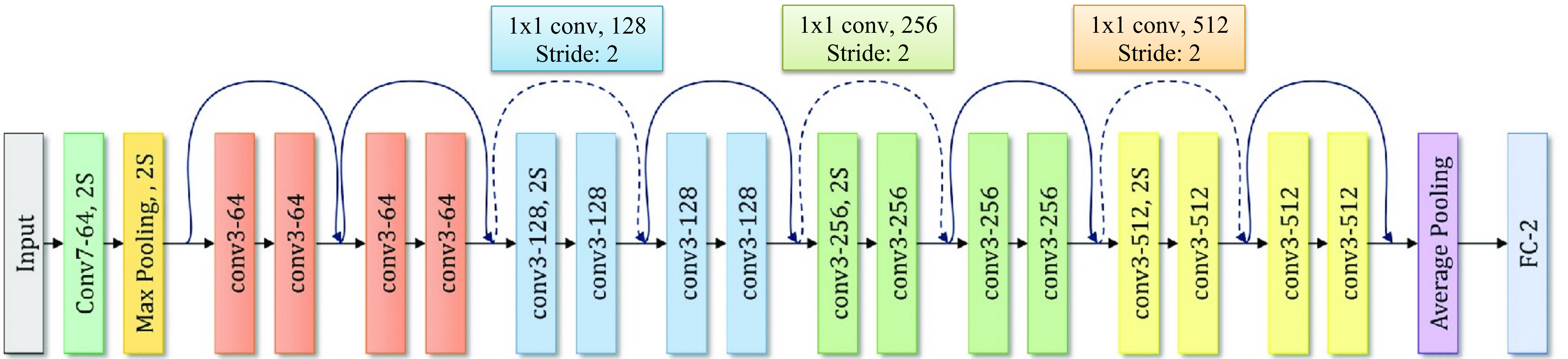
MODEL
(LeNet, ResNet,…)

Class: CAT

Pretrained Model

1 – Object Detection

! Image Classification using ResNet18



1x1 conv, 128
Stride: 2

1x1 conv, 256
Stride: 2

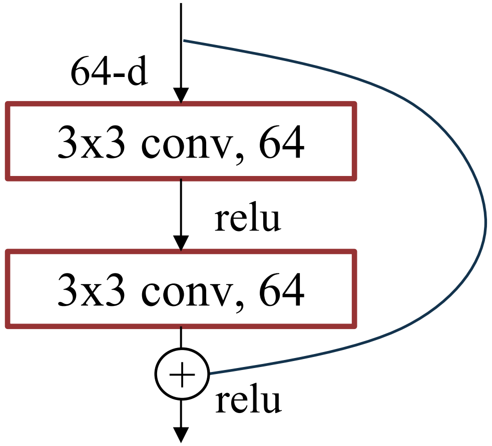
1x1 conv, 512
Stride: 2

2 strides to
down-sample

2 strides to
down-sample

2 strides to
down-sample

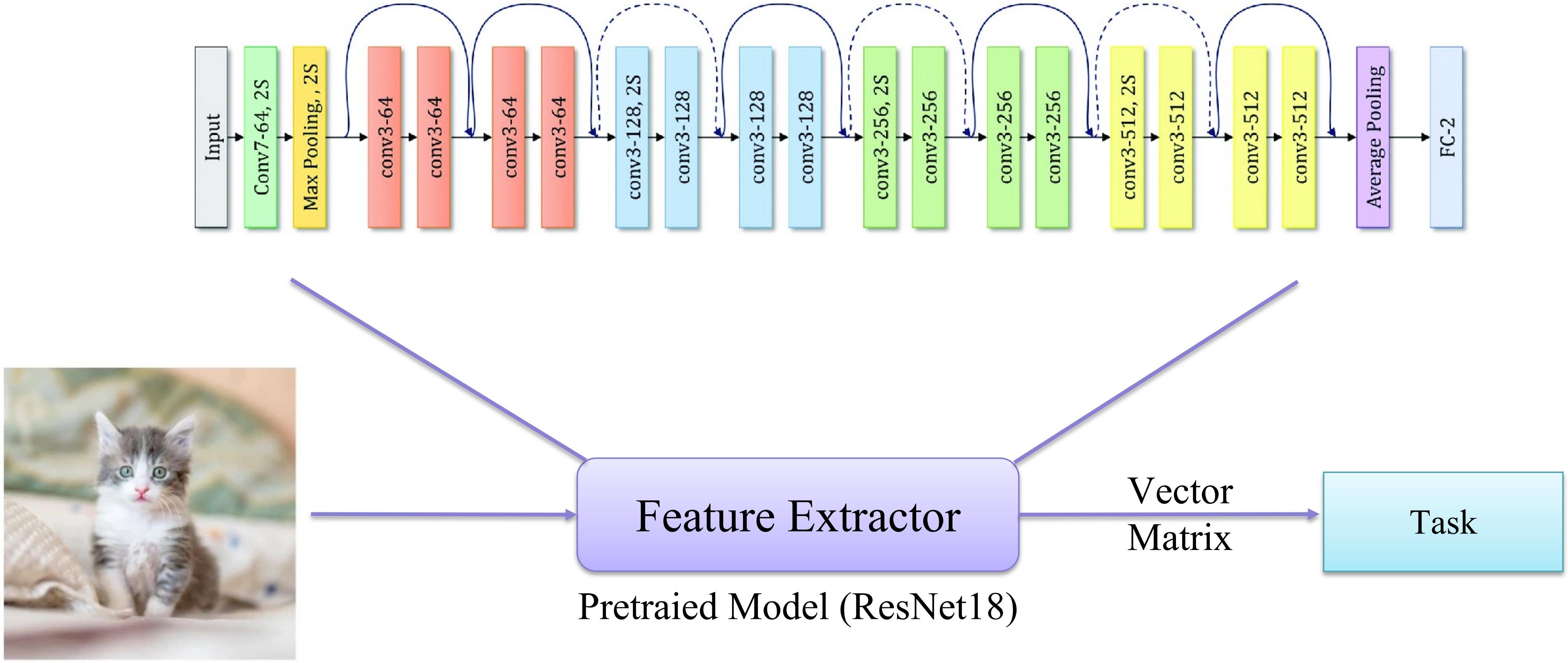
Classifier



Source

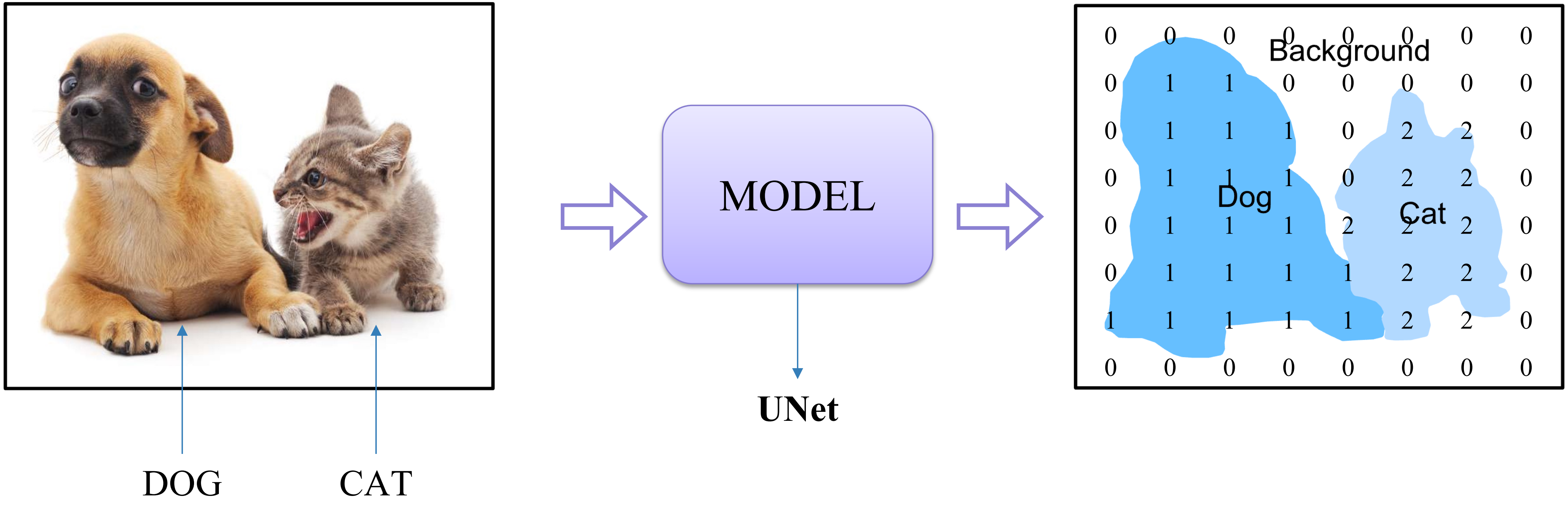
1 – Object Detection

Image Classification using ResNet18



1 – Object Detection

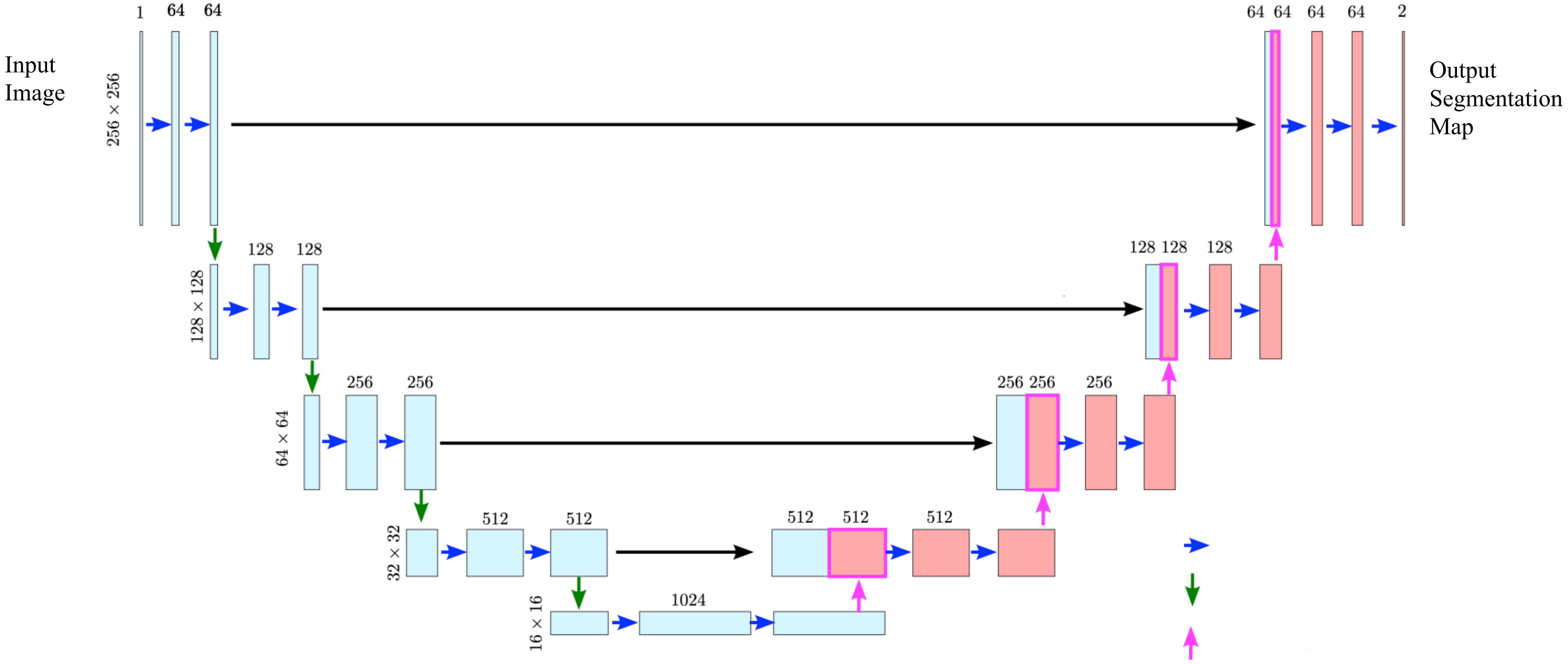
Image Segmentation



0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	2	2	0
0	1	1	1	0	2	2	0
0	1	1	1	2	2	2	0
0	1	1	1	1	2	2	0
1	1	1	1	1	2	2	0
0	0	0	0	0	0	0	0

1 – Object Detection

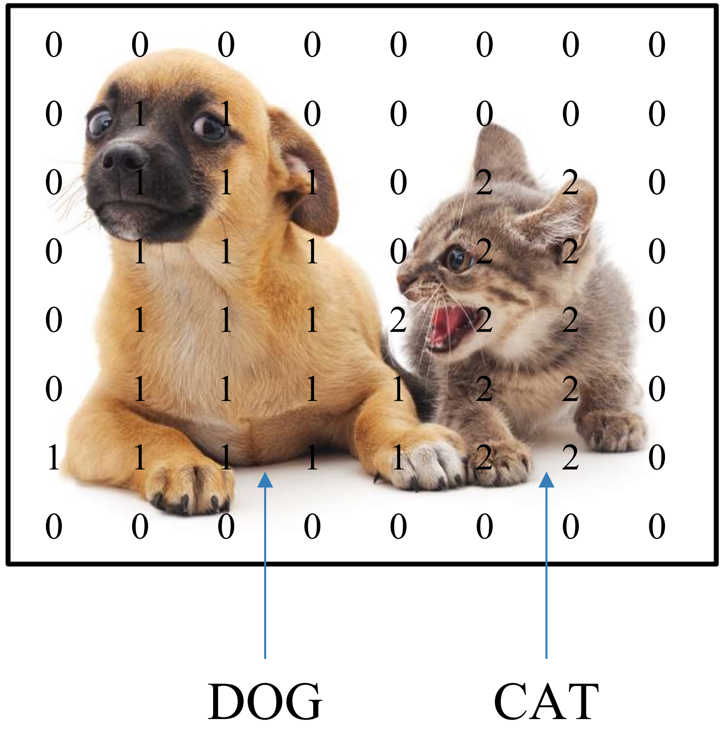
Image Segmentation using UNet



1 – Object Detection

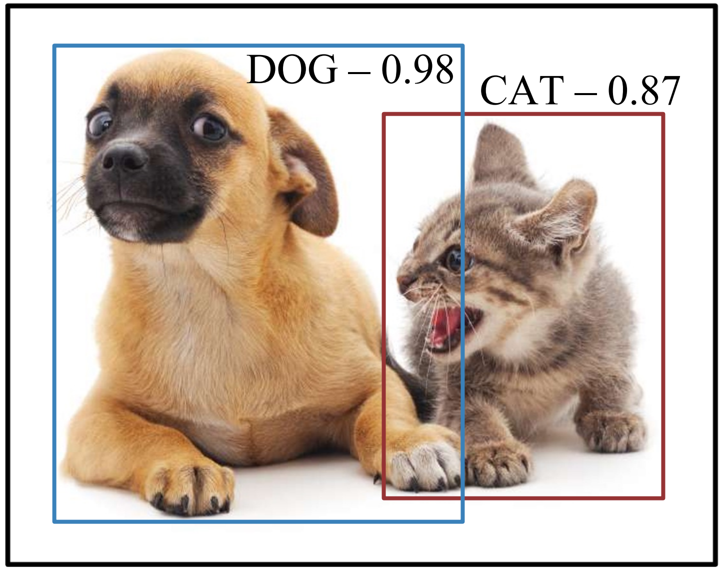
Object Detection

Image Segmentation



0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	2	2	0
0	1	1	1	0	2	2	0
0	1	1	1	2	2	2	0
0	1	1	1	1	2	2	0
1	1	1	1	1	2	2	0
0	0	0	0	0	0	0	0

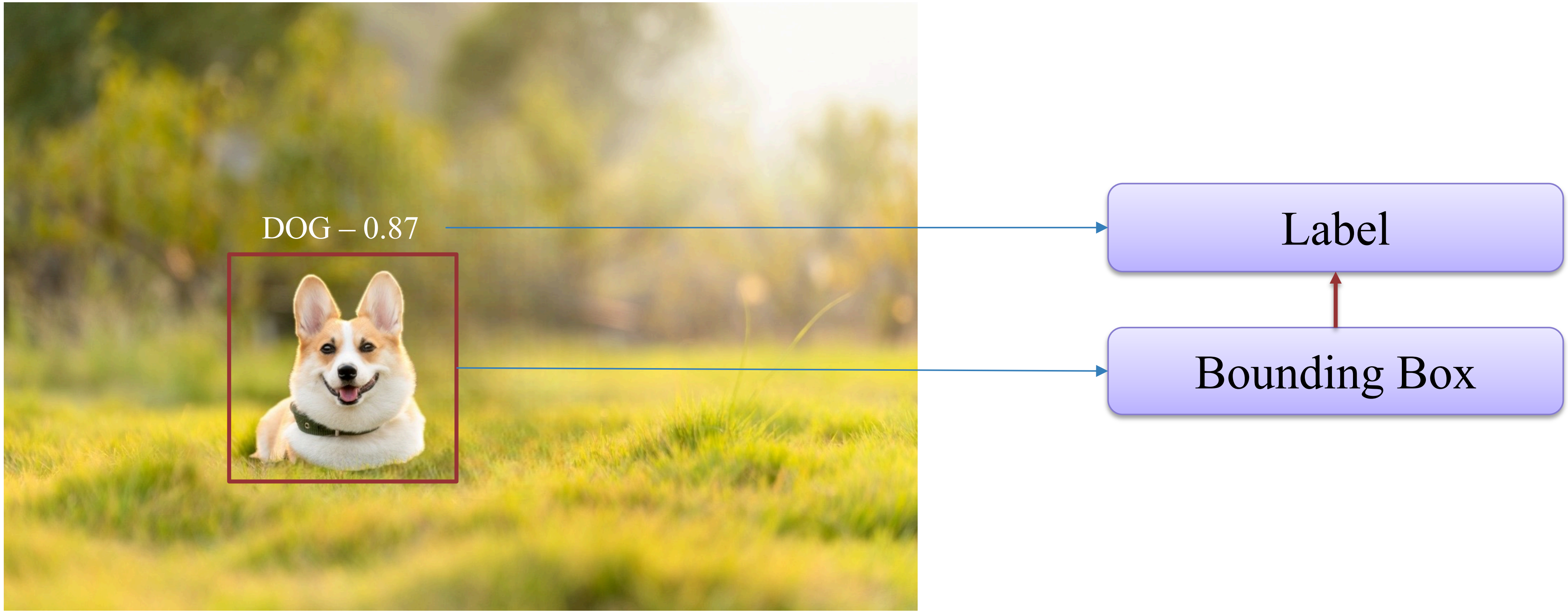
Object Detection



Assign labels, bounding boxes
to objects in the image

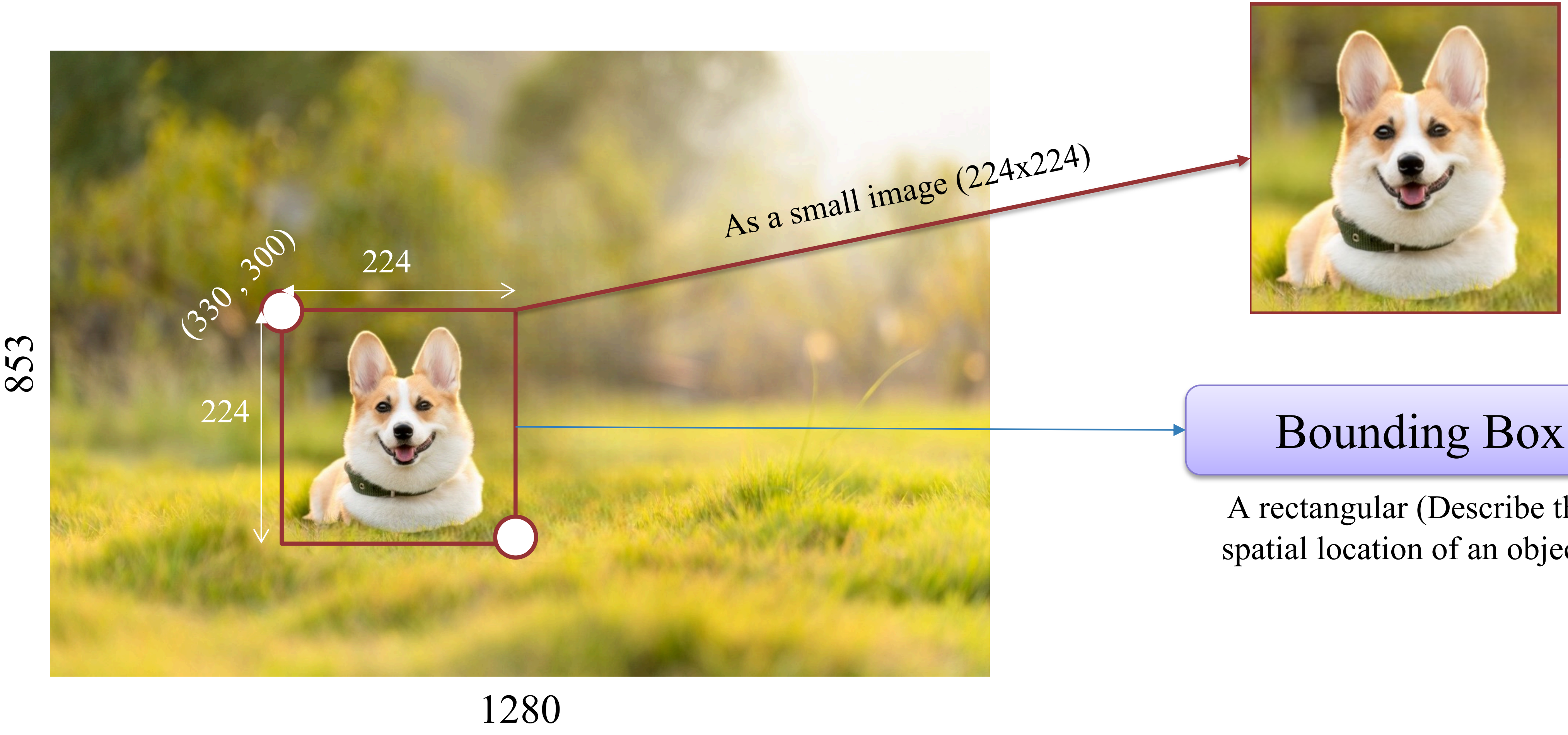
1 – Object Detection

Object Detection



1 – Object Detection

Object Detection

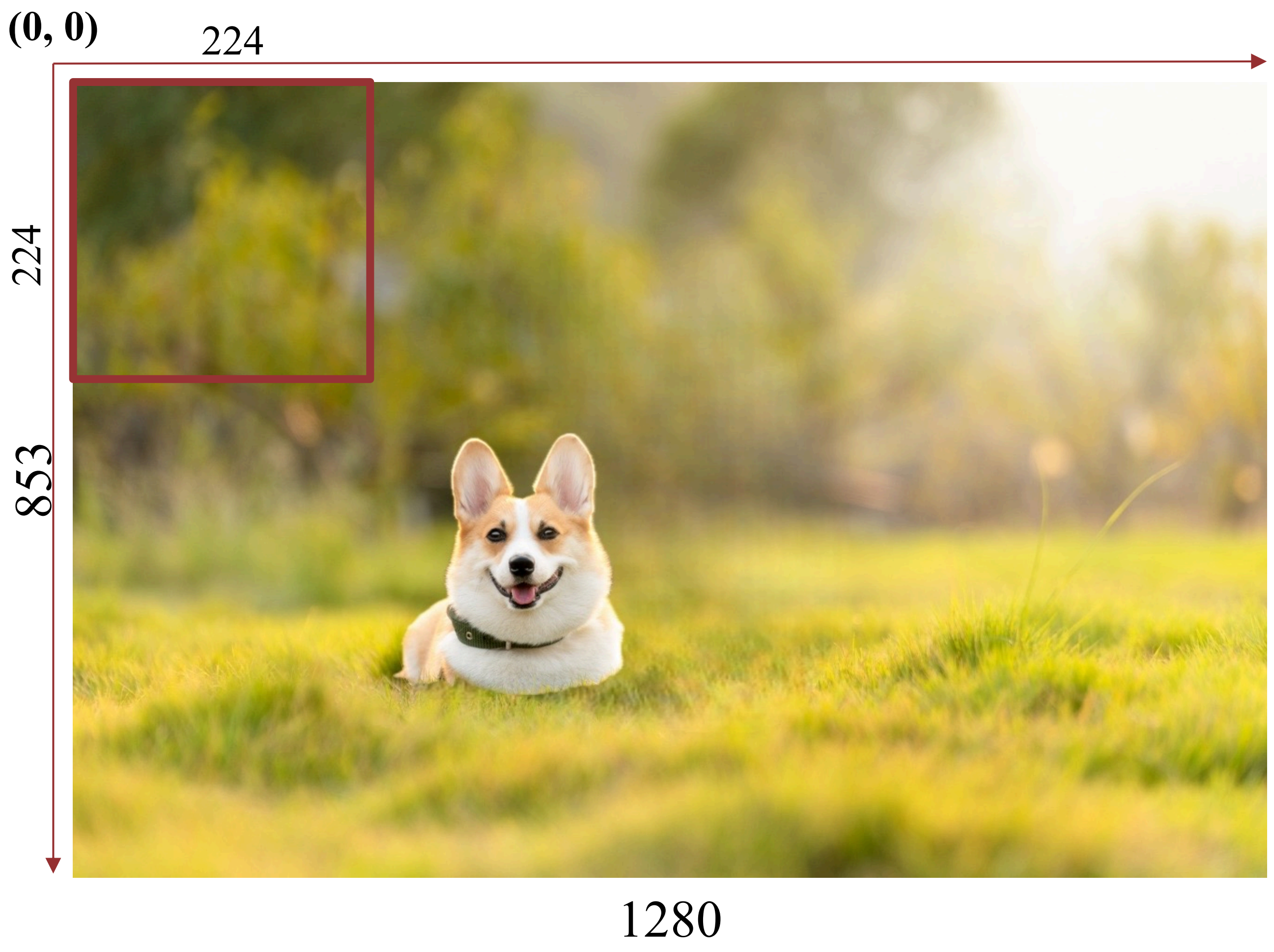


Bounding Box

A rectangular (Describe the spatial location of an object)

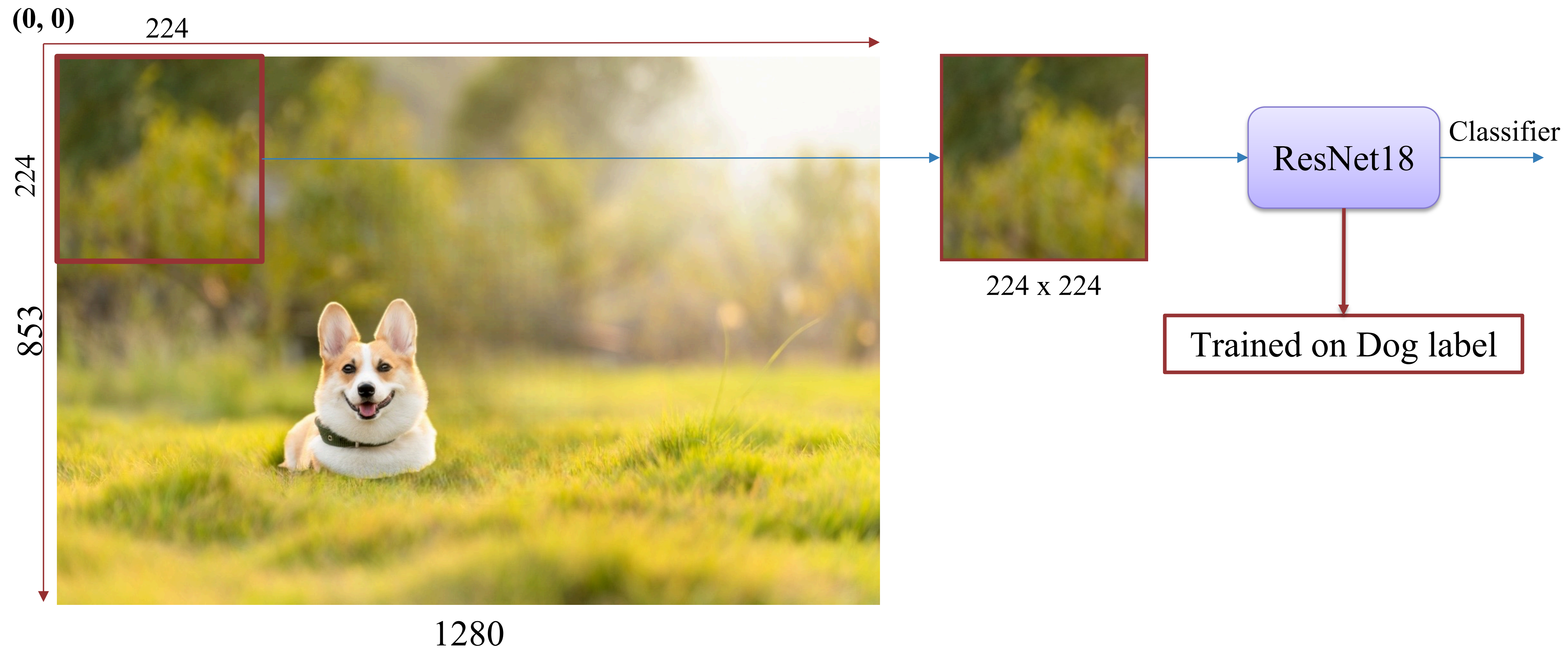
2 – Naïve Object Detection

Naïve Object Detection (as Classification)



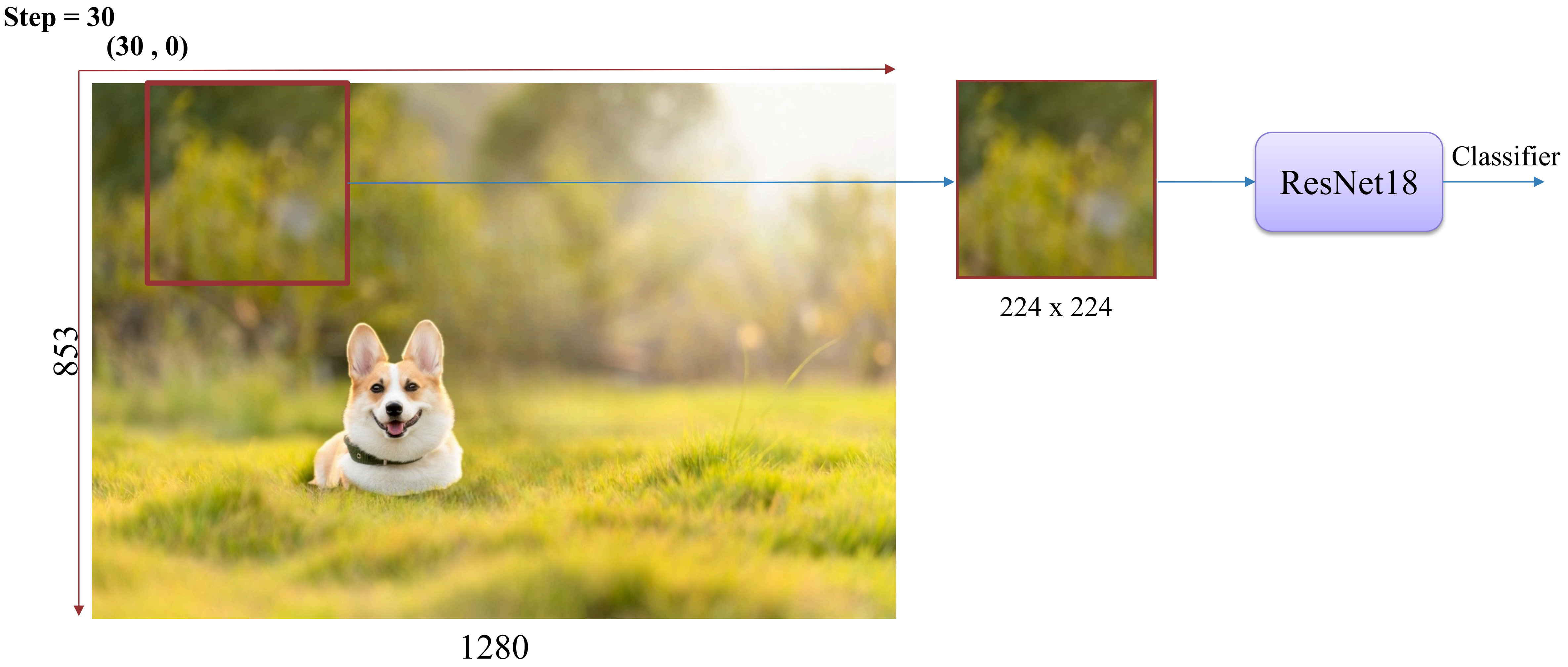
2 – Naïve Object Detection

Naïve Object Detection (as Classification)



2 – Naïve Object Detection

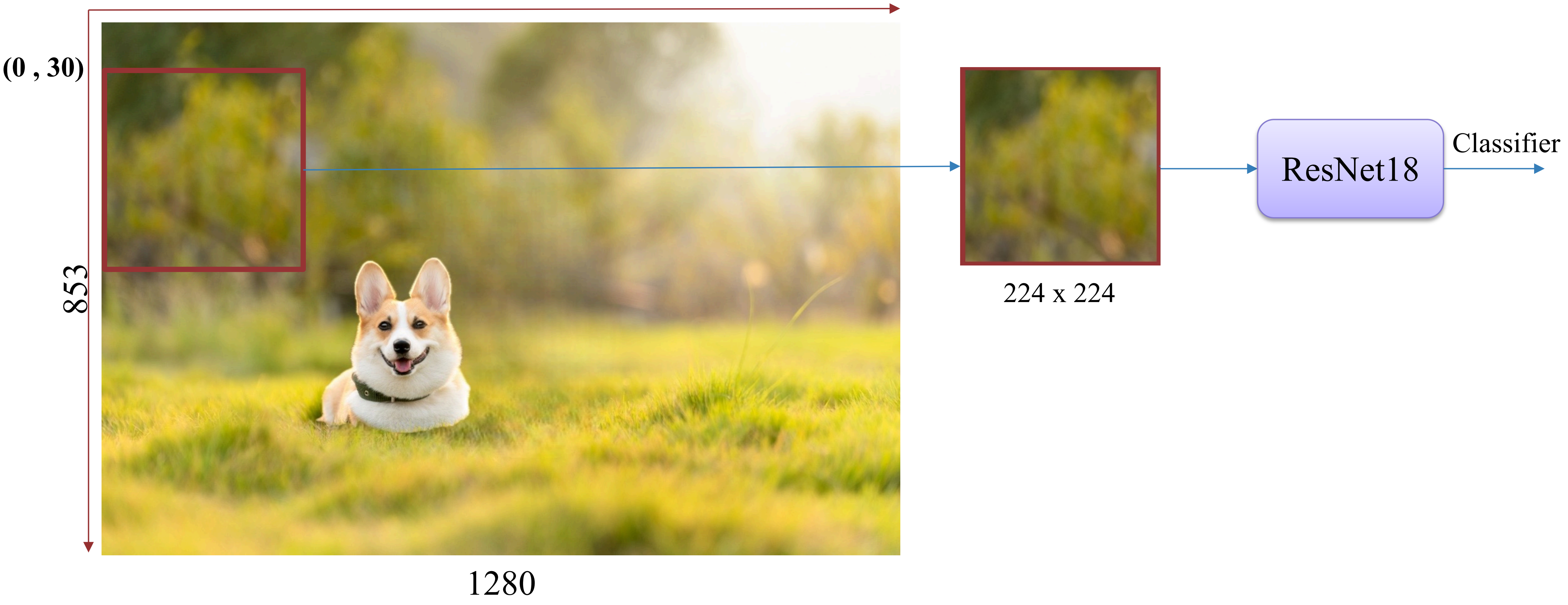
Naïve Object Detection (as Classification)



2 – Naïve Object Detection

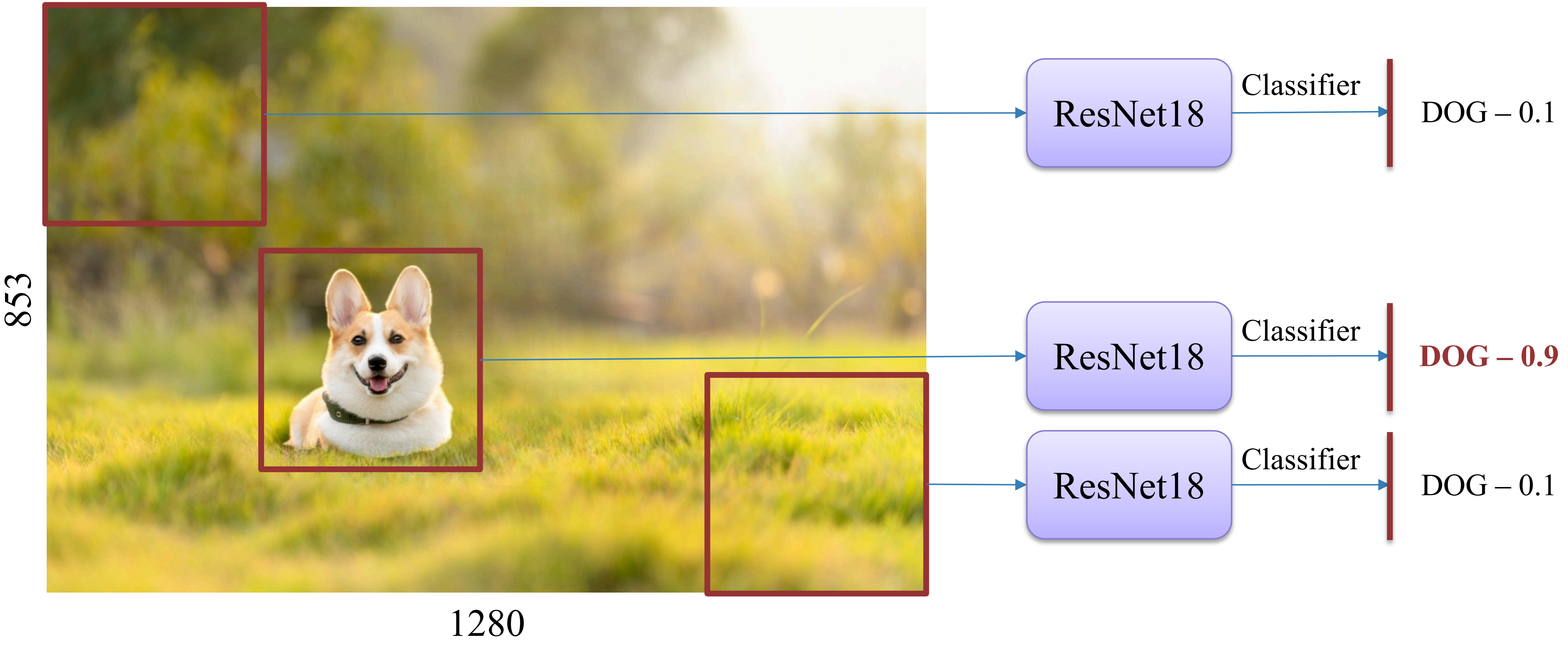
Naïve Object Detection (as Classification)

Step = 30



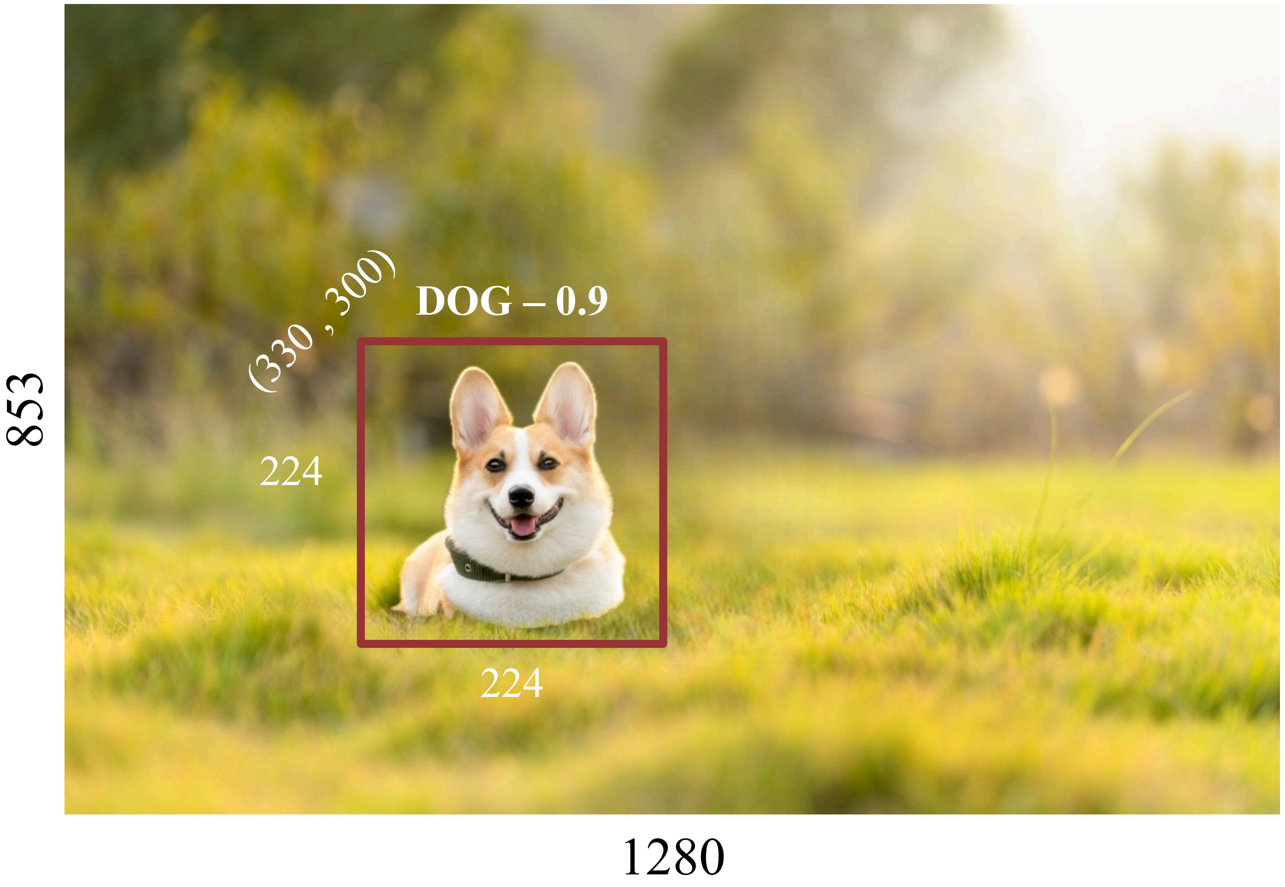
2 – Naïve Object Detection

Naïve Object Detection (as Classification)



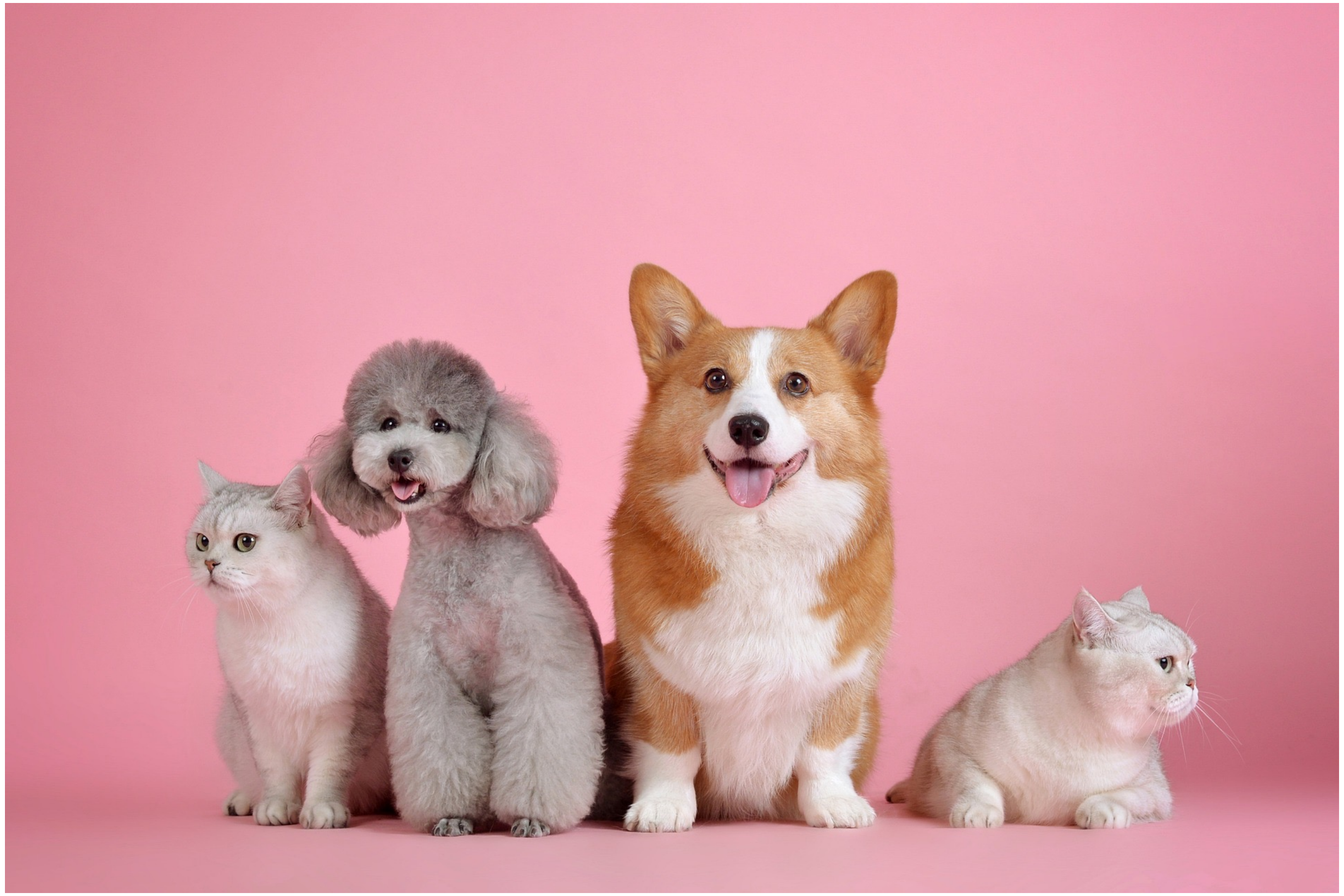
2 – Naïve Object Detection

! Naïve Object Detection (as Classification)



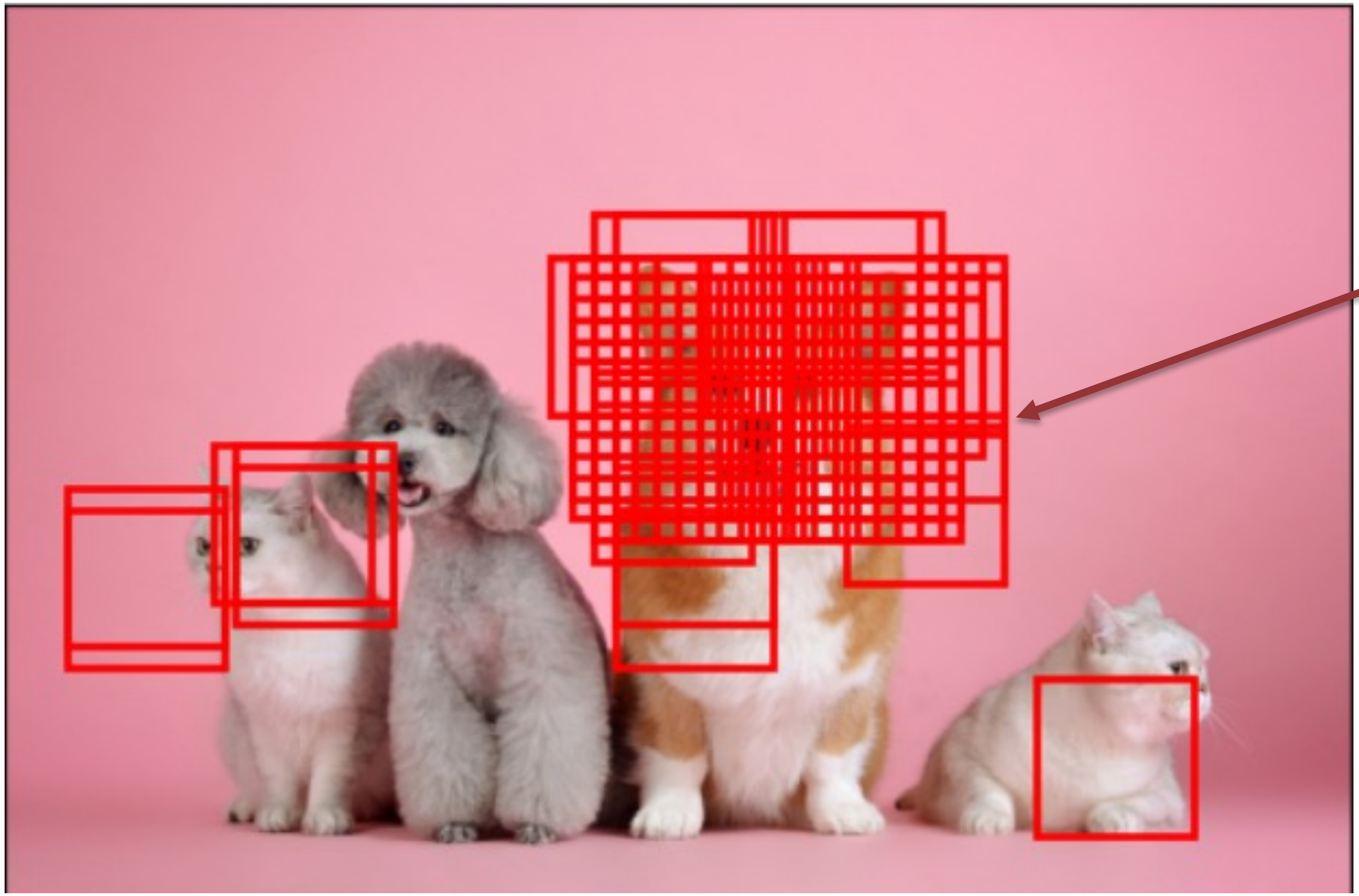
2 – Naïve Object Detection

! Multi Objects



2 – Naïve Object Detection

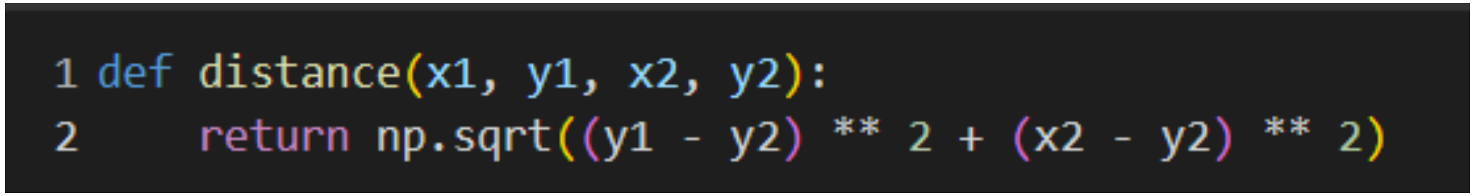
Multi Objects



Any way to reduce the overlapping bounding boxes

2 – Naïve Object Detection

Multi Objects



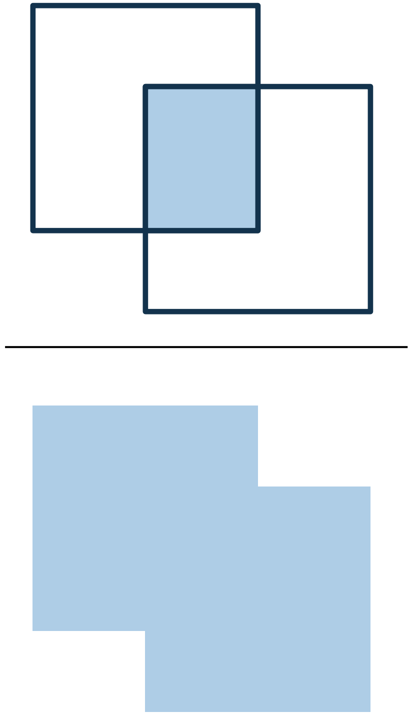
```python
def distance(x1, y1, x2, y2):
    return np.sqrt((y1 - y2) ** 2 + (x2 - y2) ** 2)
```

Pros	Cons
Simple	Hard to choose threshold
Easy to implement	Does not work well when bounding box shape vary

2 – Naïve Object Detection

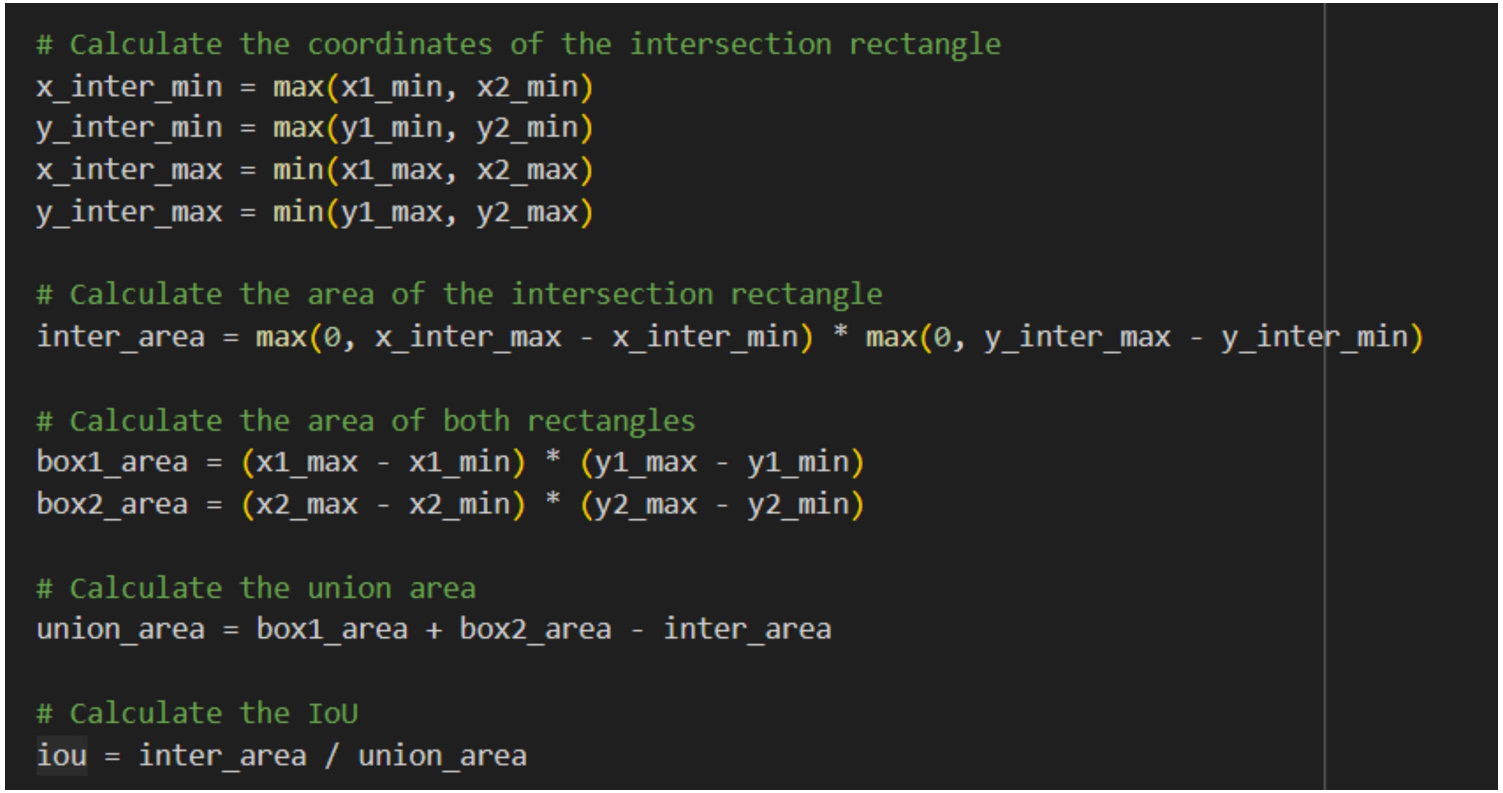
Multi Objects - IoU

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} =$$

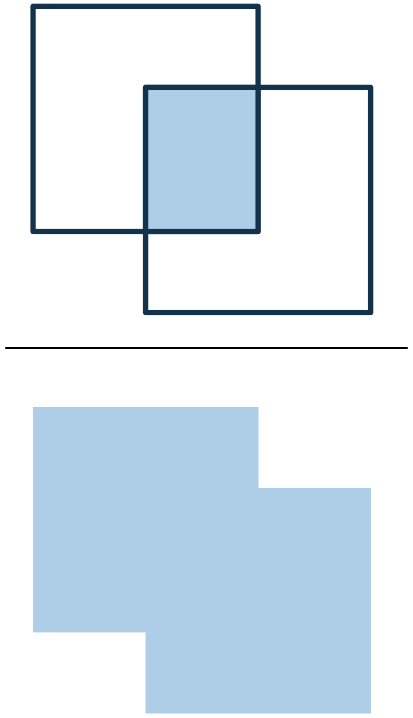


2 – Naïve Object Detection

Multi Objects - IoU

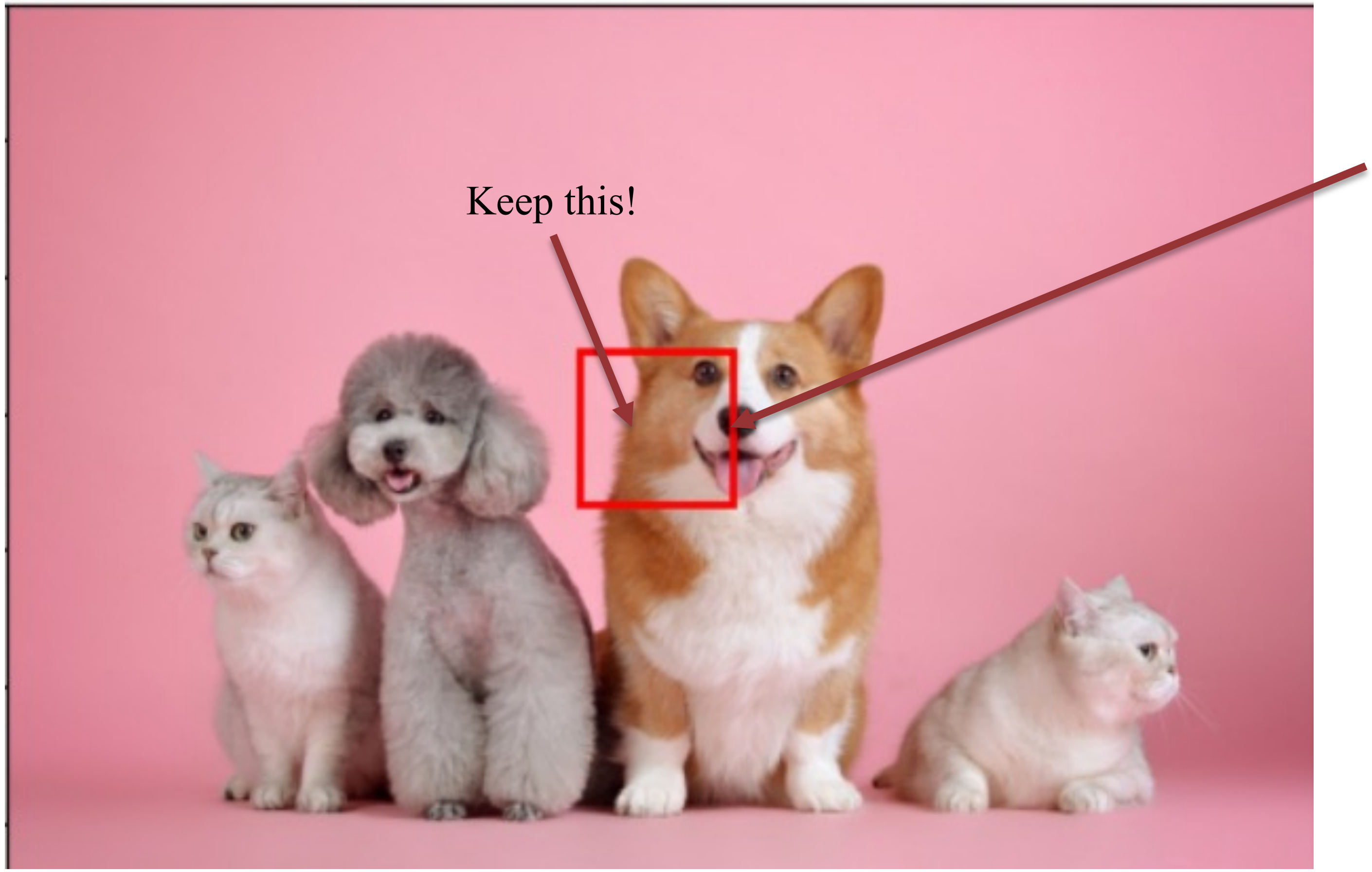


```
# Calculate the coordinates of the intersection rectangle
x_inter_min = max(x1_min, x2_min)
y_inter_min = max(y1_min, y2_min)
x_inter_max = min(x1_max, x2_max)
y_inter_max = min(y1_max, y2_max)

# Calculate the area of the intersection rectangle
inter_area = max(0, x_inter_max - x_inter_min) * max(0, y_inter_max - y_inter_min)

# Calculate the area of both rectangles
box1_area = (x1_max - x1_min) * (y1_max - y1_min)
box2_area = (x2_max - x2_min) * (y2_max - y2_min)

# Calculate the union area
union_area = box1_area + box2_area - inter_area

# Calculate the IoU
iou = inter_area / union_area
```



2 – Naïve Object Detection

Multi Objects – Filter Function



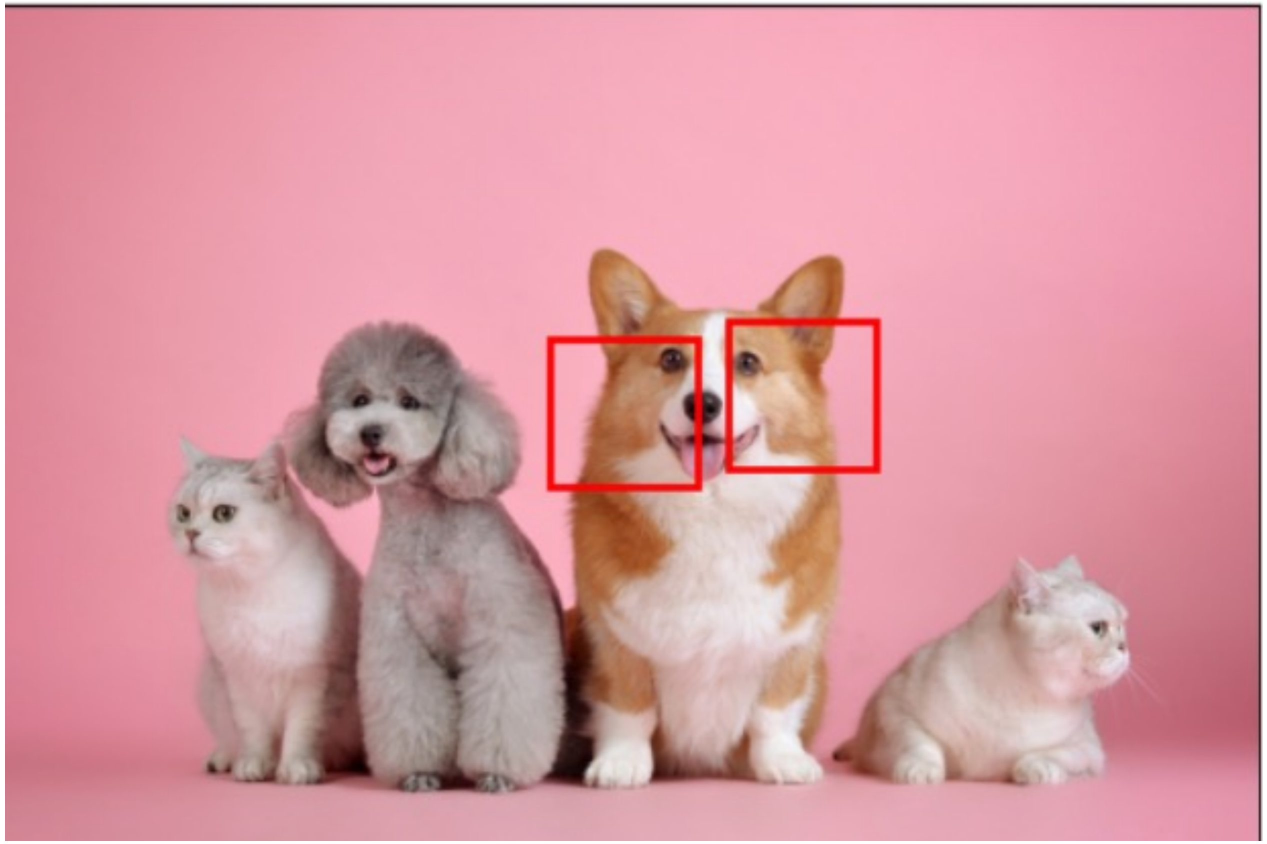
Get the bounding box with the highest score.

Filter the boxes with IoU > threshold

2 – Naïve Object Detection

!

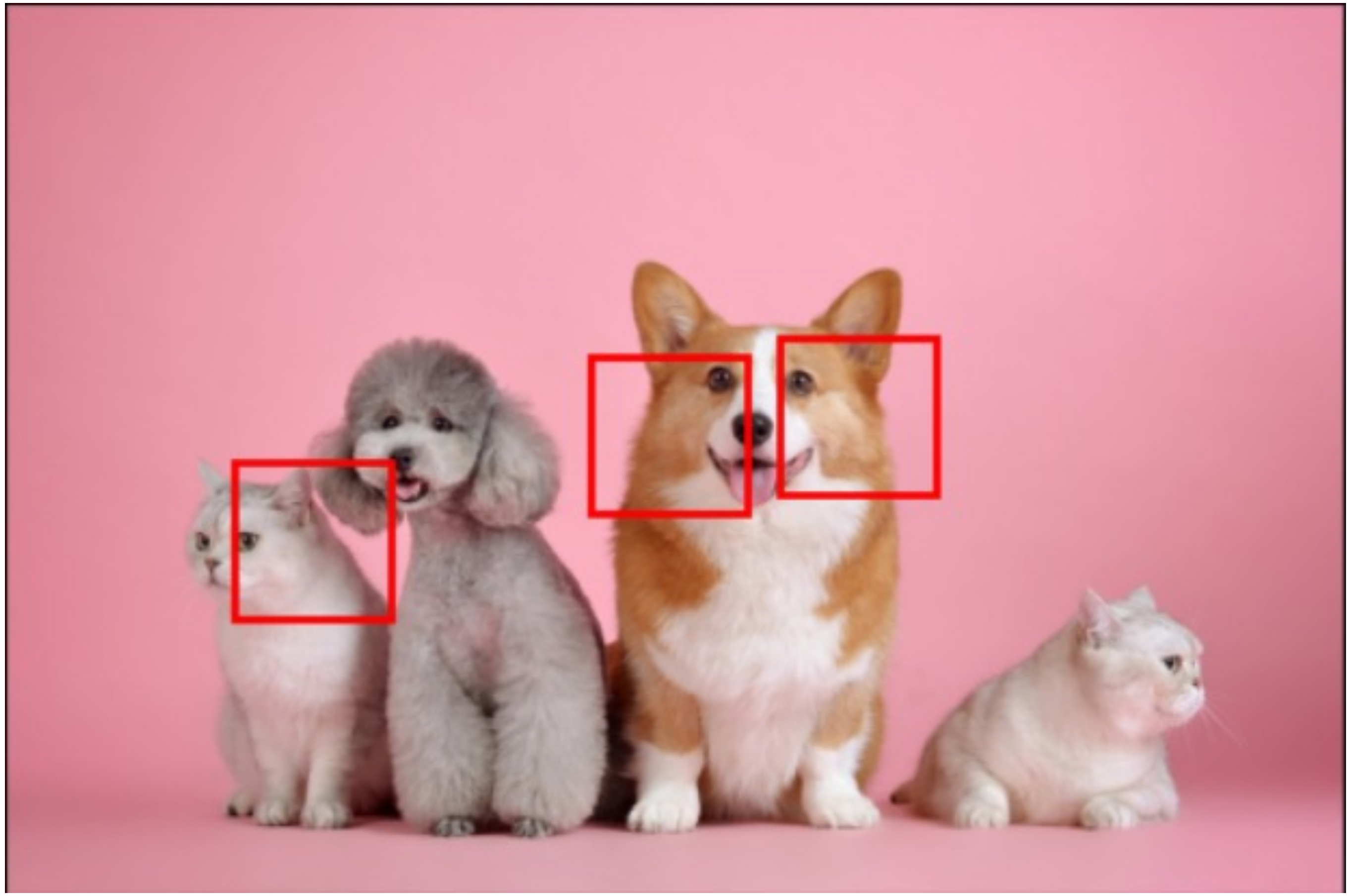
Multi Objects – Filter Function



2 – Naïve Object Detection

!

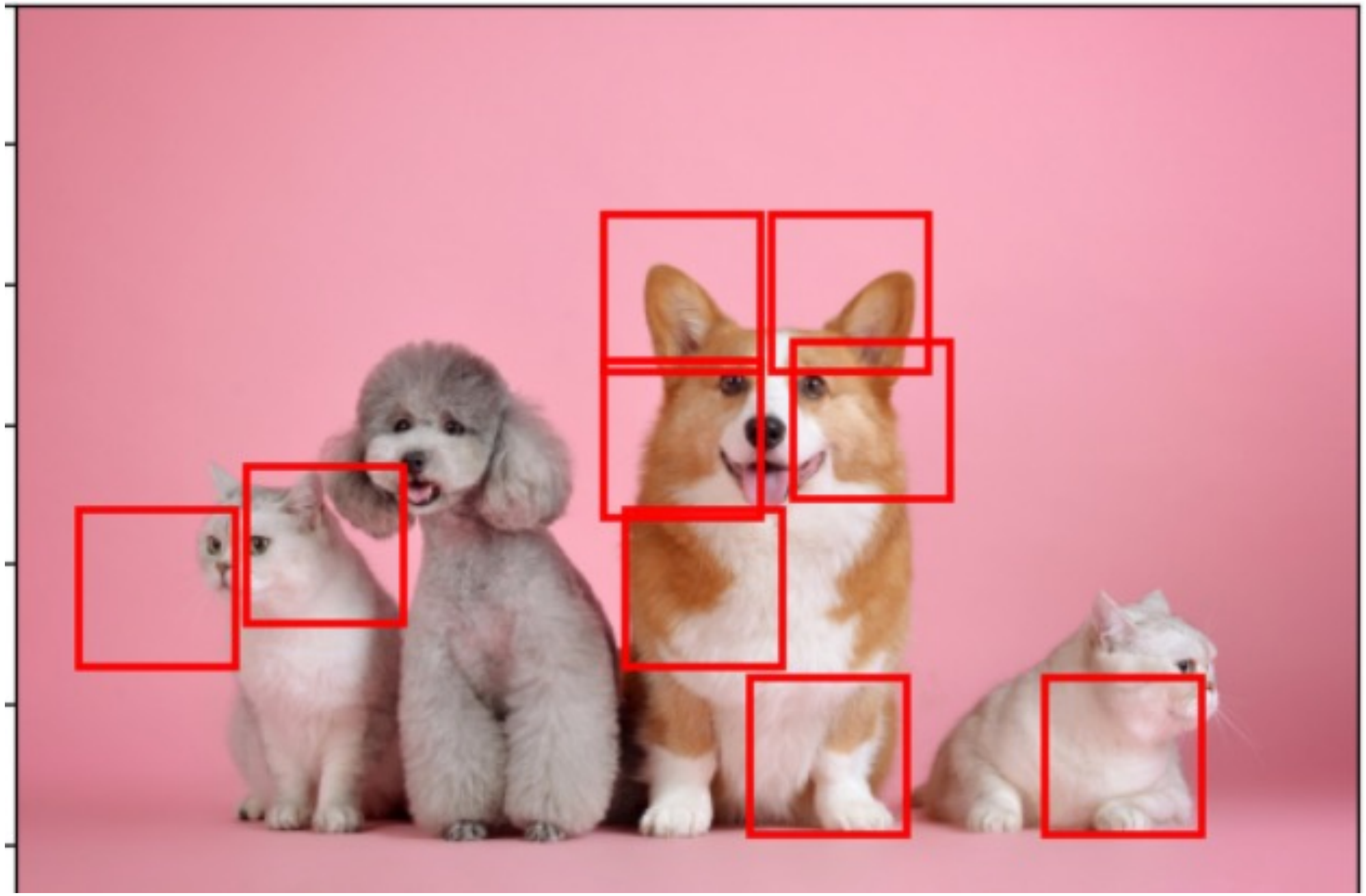
Multi Objects – Filter Function



2 – Naïve Object Detection

!

Multi Objects – Filter Function



Thanks!

Any questions?