YOLO

Object Recognition and Ontology Generation for Qualitative Scene Description (with YOLO)

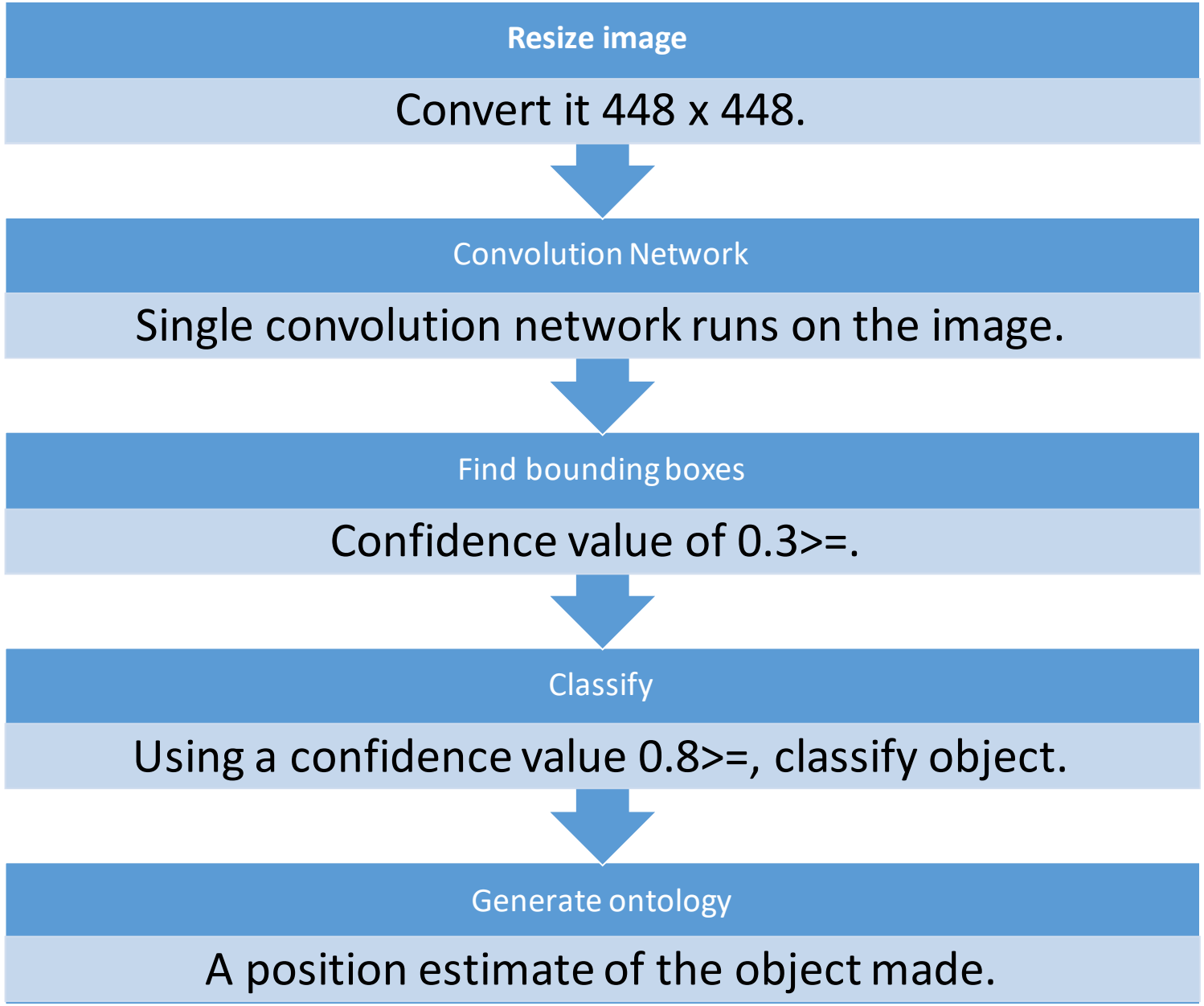
By- Dipika Boro, Shruti Mohanty, Zubin Bhuyan

5th May 2018

Problem Statement

- Correctly Identify objects in an image.
- Ontology generation for location of the image.

Steps


Resize image
Convert it 448 x 448.
Convolution Network
Single convolution network runs on the image.
Find bounding boxes
Confidence value of 0.3>=.
Classify
Using a confidence value 0.8>=, classify object.
Generate ontology
A position estimate of the object made.

YOLO

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

YOLO

- 19 convolutional layer
- 5 max-pooling layers.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

YOLO Algorithm

- System divides the input image into a 13 × 13 grid.
- Centre of the object is the point to be considered.
- Each grid cell predicts 5 bounding boxes + Confidence score.
- Each bounding box contains 5 predictions: x, y, h, w and the confidence and also the categorical confidences(20).
- The final output of our network is the 13 x 13 x 5 x (5 + 20) tensor of predictions.

Dataset

- PASCAL VOC
  - Train on VOC 2007 + 2012
  - Test on VOC 2007 (test)

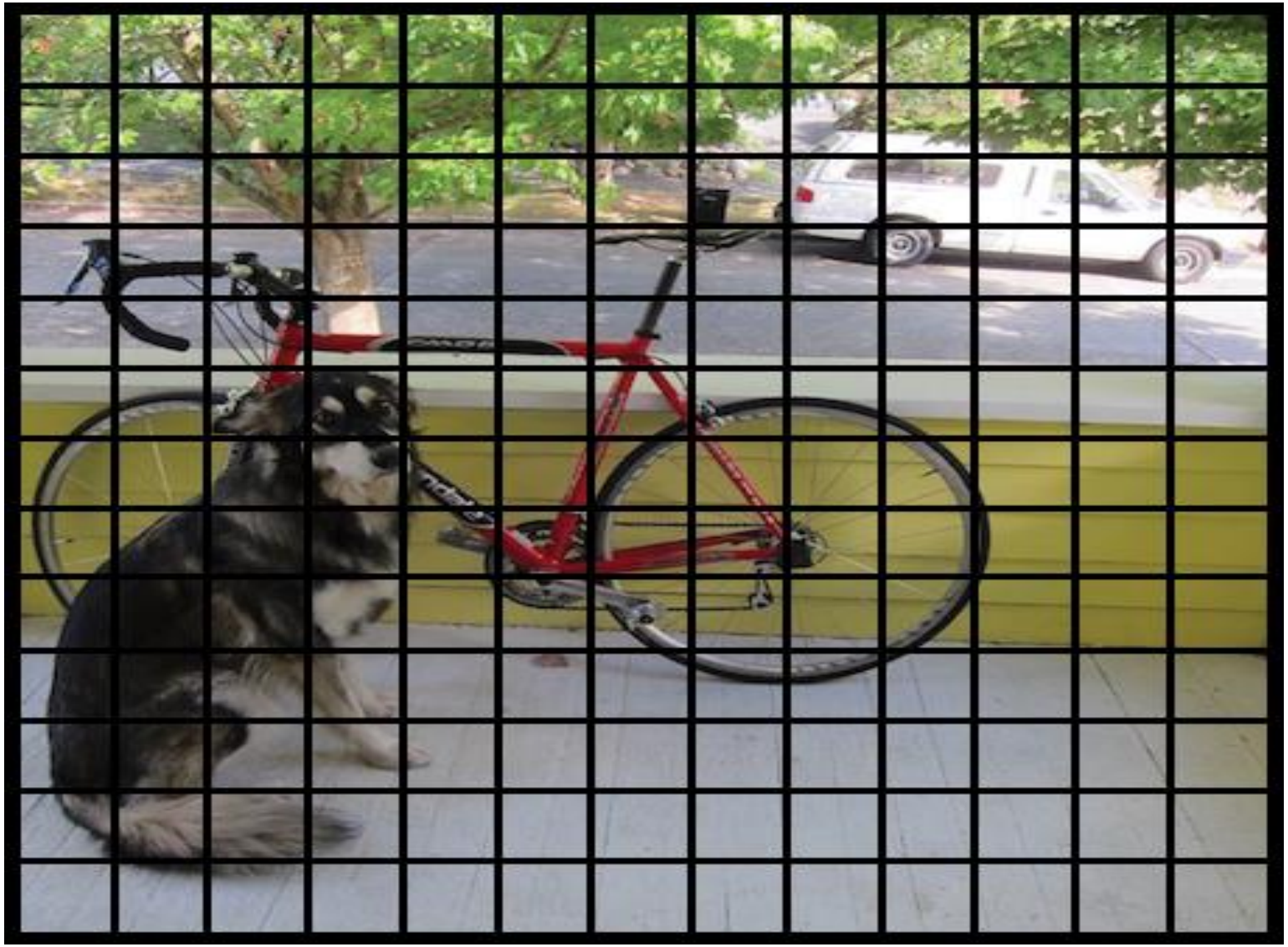
Training system specs

- (2.3 GHz Intel Xenon E5) x 2
- (NVIDIA Tesla K80 GPU) x 1
  - 4992 CUDA cores
  - (12 GB memory) x 2

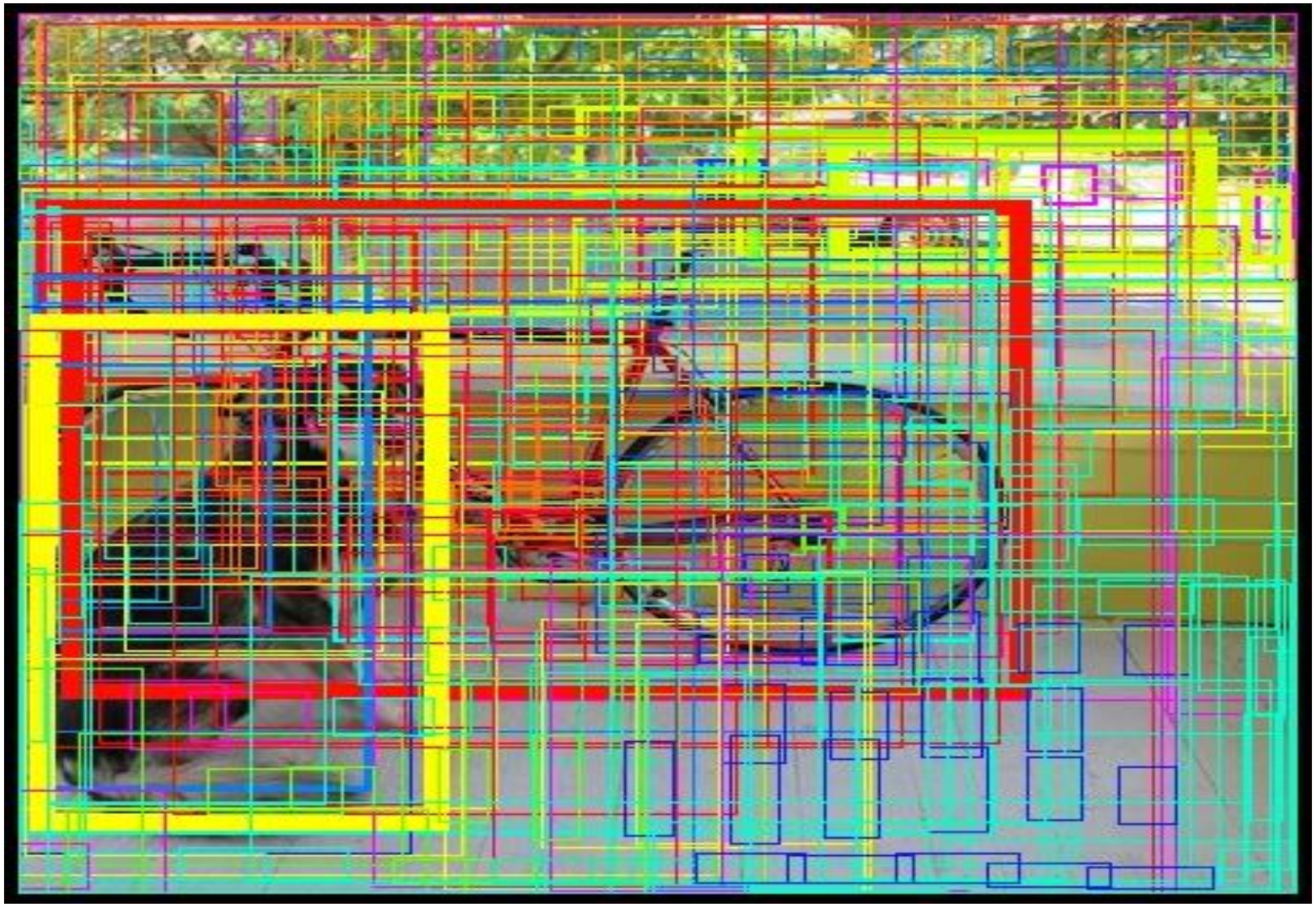
- Python 2.7
  - And PyTorch 0.4

Sample image

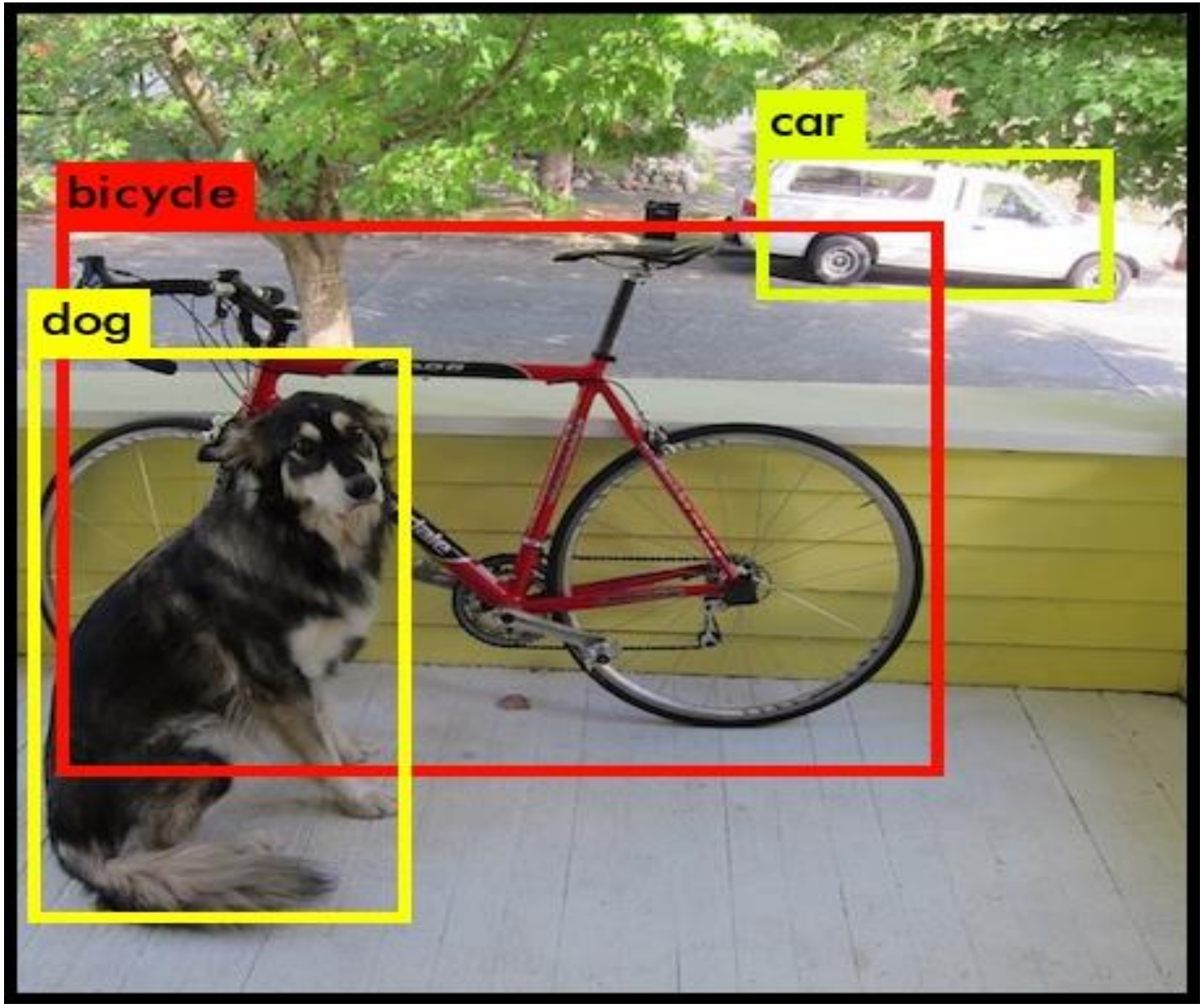
Grid 13x13



Max 845 boxes (BB > .3)



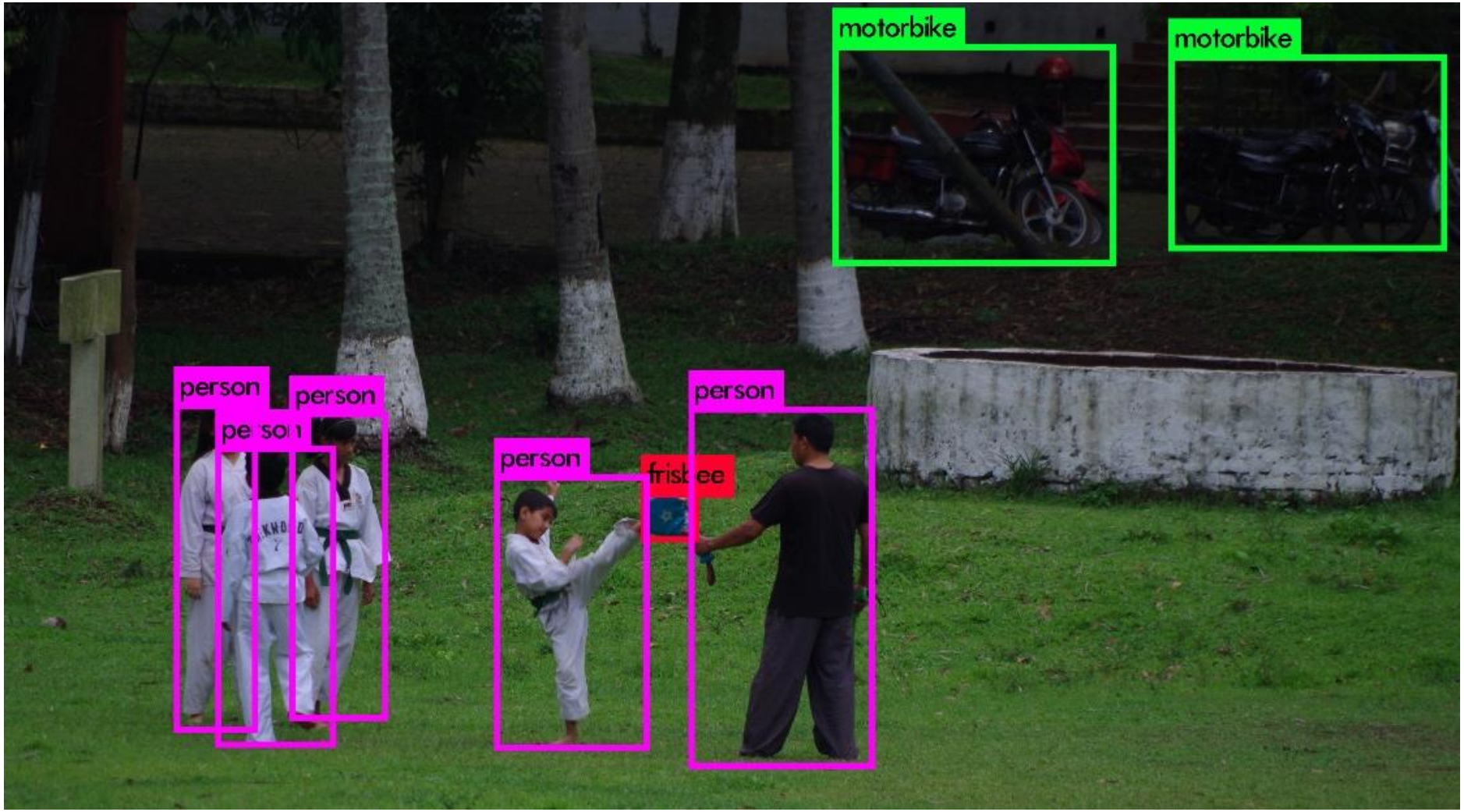
Result class > 0.8



Result

- Mean Average Precision : **0.6422**

Detecting images not in PASCAL VOC



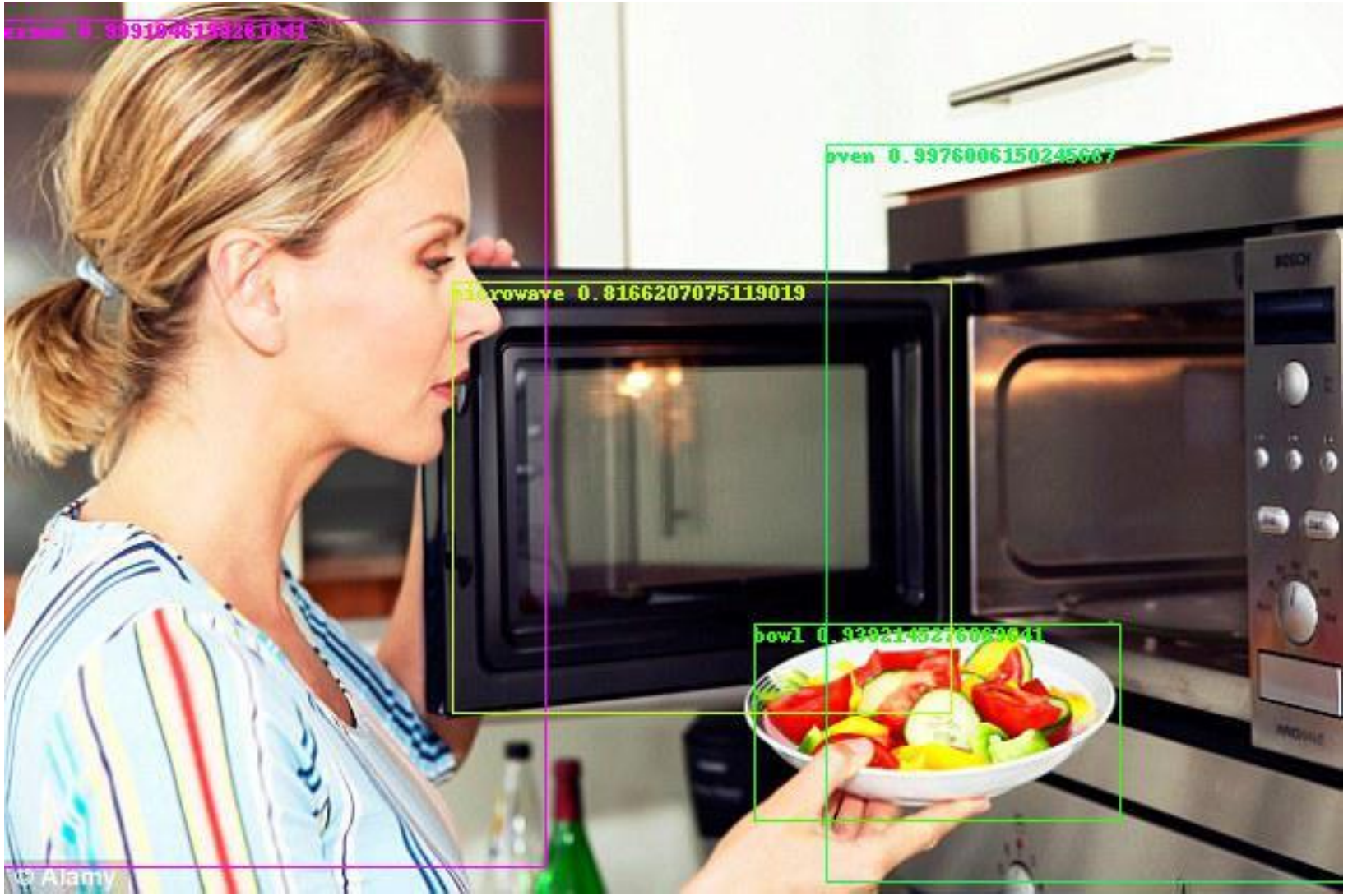
RDF

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:objProp="https://www.cs.uml.edu/~zbhuyan/objects">
    <rdf:Description rdf:about="https://www.cs.uml.edu/~zbhuyan/image/">
        <img:imgWidth>4928</objProp:id>
        <img:imgHeight>3264</objProp:id>
    </rdf:Description>
    <rdf:Description rdf:about="https://www.cs.uml.edu/~zbhuyan/objects/person">
        <objProp:id>0</objProp:id>
        <objProp:centroidX>2919.75634765625</objProp:idcentroidX>
        <objProp:centroidY>1724.7393798828125</objProp:idcentroidY>
        <objProp:x1>2606.12744140625</objProp:x1>
        <objProp:y1>1181.03662109375</objProp:y1>
        <objProp:x2>3233.385009765625</objProp:x2>
        <objProp:y2>2268.442138671875</objProp:y2>
    </rdf:Description>
    <rdf:Description rdf:about="https://www.cs.uml.edu/~zbhuyan/objects/person">
        <objProp:id>1</objProp:id>
        <objProp:centroidX>1306.197509765625</objProp:idcentroidX>
        <objProp:centroidY>1701.65478515625</objProp:idcentroidY>
        <objProp:x1>1125.68212890625</objProp:x1>
        <objProp:y1>1180.4398193359375</objProp:y1>
```

Detecting images not in PASCAL VOC

Detecting images not in PASCAL VOC

Detecting images not in PASCAL VOC



Thanks!

	train		val		trainval		test	
	img	**obj**	**img**	**obj**	**img**	**obj**	**img**	**obj**
Aeroplane	327	432	343	433	670	865	–	–
Bicycle	268	353	284	358	552	711	–	–
Bird	395	560	370	559	765	1119	–	–
Boat	260	426	248	424	508	850	–	–
Bottle	365	629	341	630	706	1259	–	–
Bus	213	292	208	301	421	593	–	–
Car	590	1013	571	1004	1161	2017	–	–
Cat	539	605	541	612	1080	1217	–	–
Chair	566	1178	553	1176	1119	2354	–	–
Cow	151	290	152	298	303	588	–	–
Diningtable	269	304	269	305	538	609	–	–
Dog	632	756	654	759	1286	1515	–	–
Horse	237	350	245	360	482	710	–	–
Motorbike	265	357	261	356	526	713	–	–
Person	1994	4194	2093	4372	4087	8566	–	–
Pottedplant	269	484	258	489	527	973	–	–
Sheep	171	400	154	413	325	813	–	–
Sofa	257	281	250	285	507	566	–	–
Train	273	313	271	315	544	628	–	–
Tvmonitor	290	392	285	392	575	784	–	–
Total	5717	13609	5823	13841	11540	27450	–	–