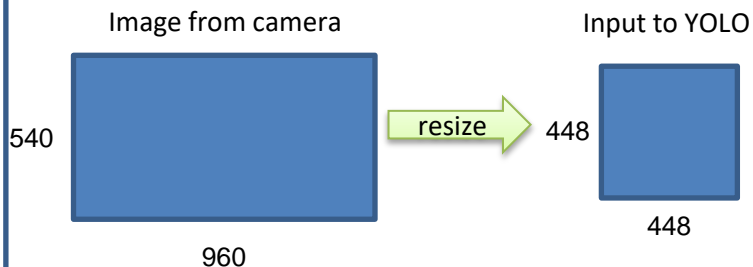
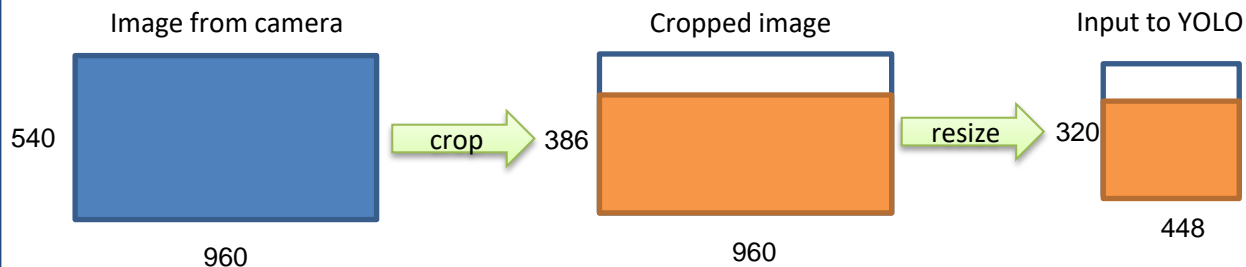


1 Default YOLO V1 Input image



2 Experiment detail – Reduced input image size



Model details

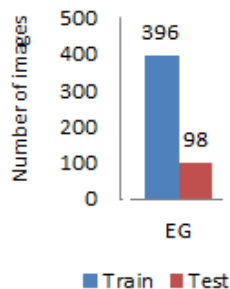
Architecture: Tiny YOLO V1
Number of layers: 16 layers
Number of classes: 1 class (electronic goods)

Layer ID	Type	Filters	Size
Layer 0	Conv	16	3 x 3 / 1
Layer 1	Max		2 x 2 / 2
Layer 2	Conv	32	3 x 3 / 1
Layer 3	Max		2 x 2 / 2
Layer 4	Conv	64	3 x 3 / 1
Layer 5	Max		2 x 2 / 2
Layer 6	Conv	128	3 x 3 / 1
Layer 7	Max		2 x 2 / 2
Layer 8	Conv	256	3 x 3 / 1
Layer 9	Max		2 x 2 / 2
Layer 10	Conv	512	3 x 3 / 1
Layer 11	Max		2 x 2 / 2
Layer 12	Conv	1024	3 x 3 / 1
Layer 13	Conv	256	3 x 3 / 1
Layer 14	Connected		
Layer 15	Detection		

Training parameters

Batch size	64
Steps	200, 400, 600, 800, 20000, 30000, 40000
Learning rate	0.0005
Scales	2.5, 2, 2, 2, 0.1, 0.1, 0.1

Data summary



Results

	Default YOLO V1 input image 448 x 448 (shown in 1)	Resized input image 320 x 448 (shown in 2)
mAP	99.29%	98.97%
Recall (Detection rate)	100%	98.97%
Thresh	0.2	0.2
Loss	0.22297	0.26684
Model size	61.7 MB	54 MB

Conclusion:

- Reducing the input size did not impact the accuracy for 1 class model much.
- Execution speed was increased by ~20-30ms