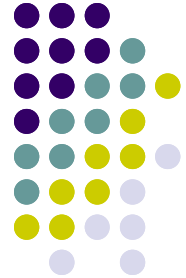


Faster Rotated Region-based Convolutional Neural Network



Viet Nguyen

viet.nguyen@tum.de

Supervisor: Dr. Csaba Domokos

Outline

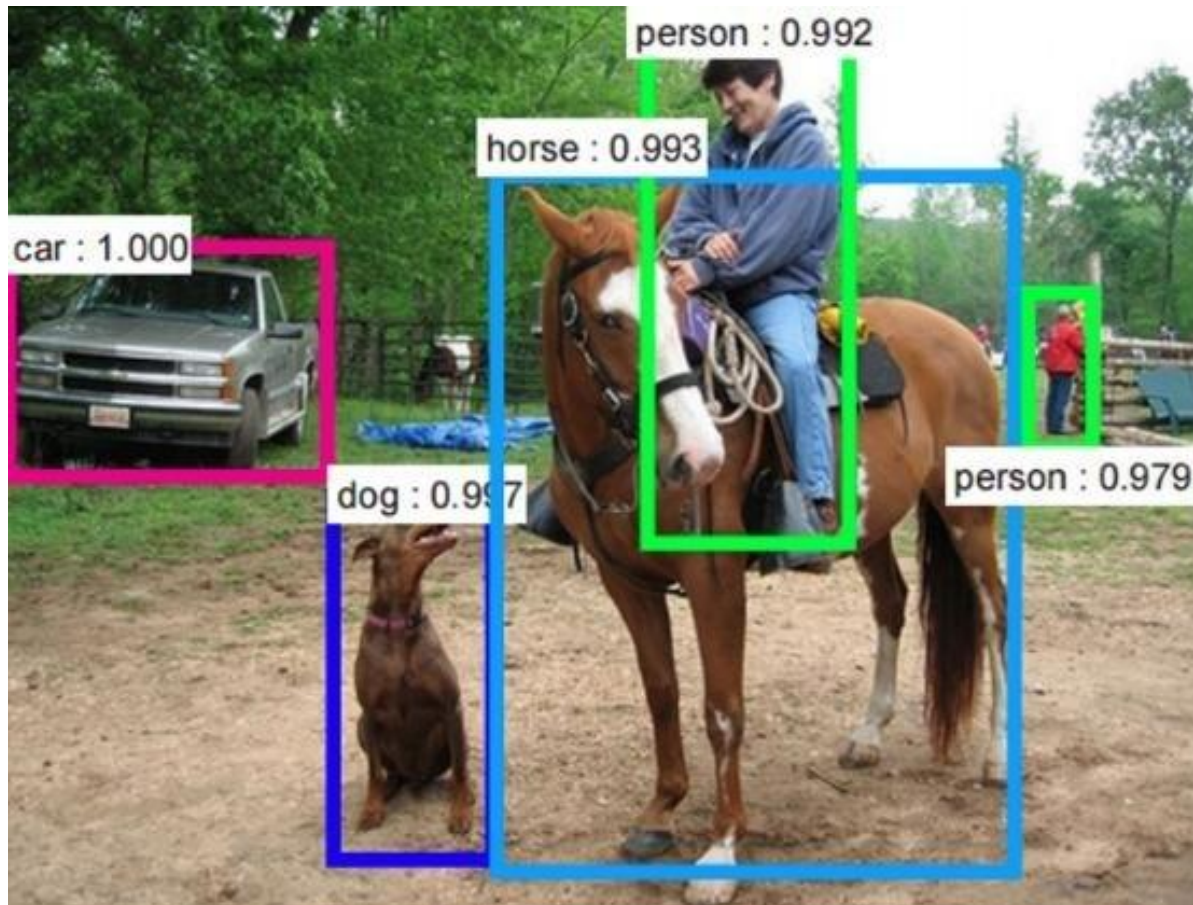


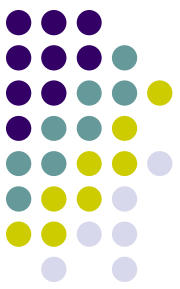
- Introduction
- Dataset
- Model
- Experiment
- Conclusion



Introduction

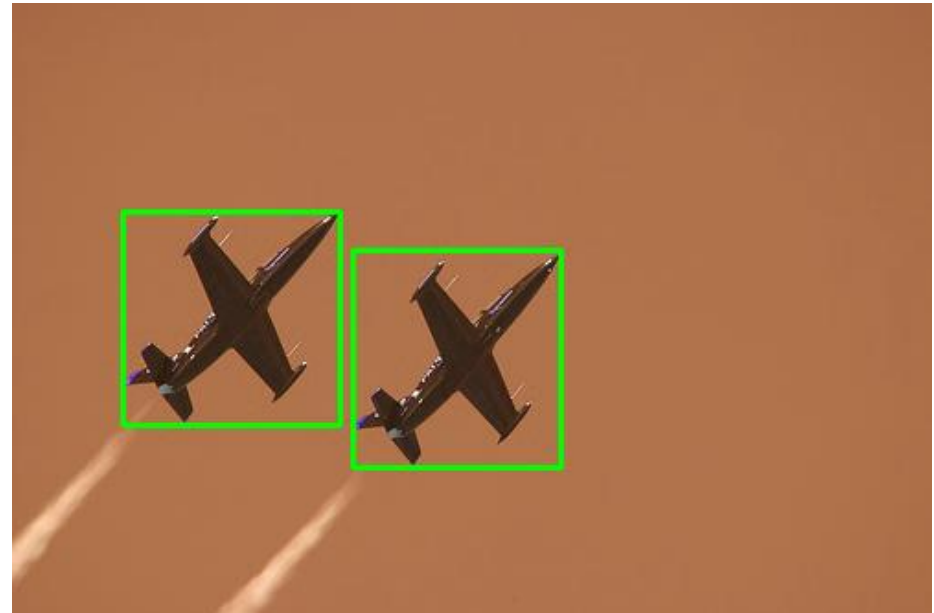
- Faster R-CNN for object detection

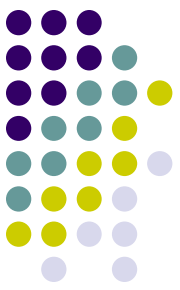




Introduction

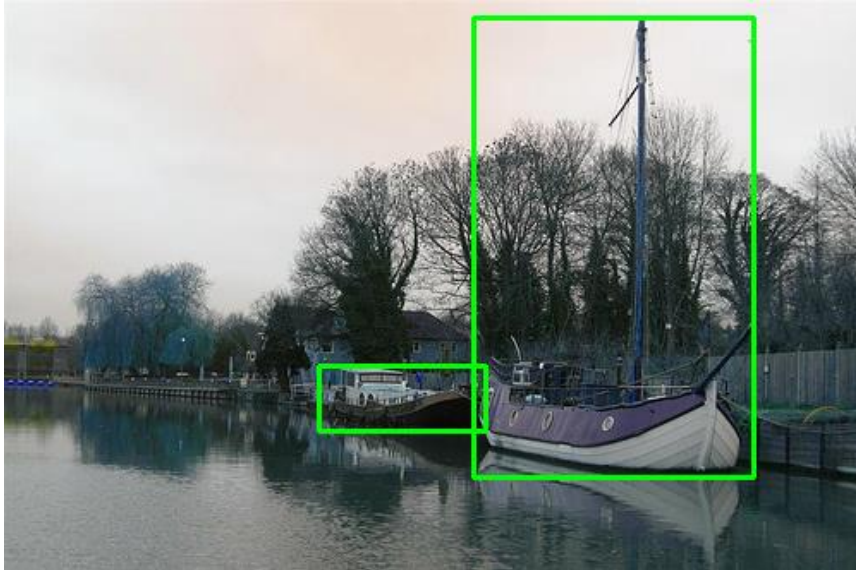
- Question: How many relevant pixels are there in a bounding box ?
- Challenge: What happen if object's orientation is not axis-aligned ?





Dataset

- VOC2012 dataset.
- Segmentation dataset → oriented detection dataset.



Dataset



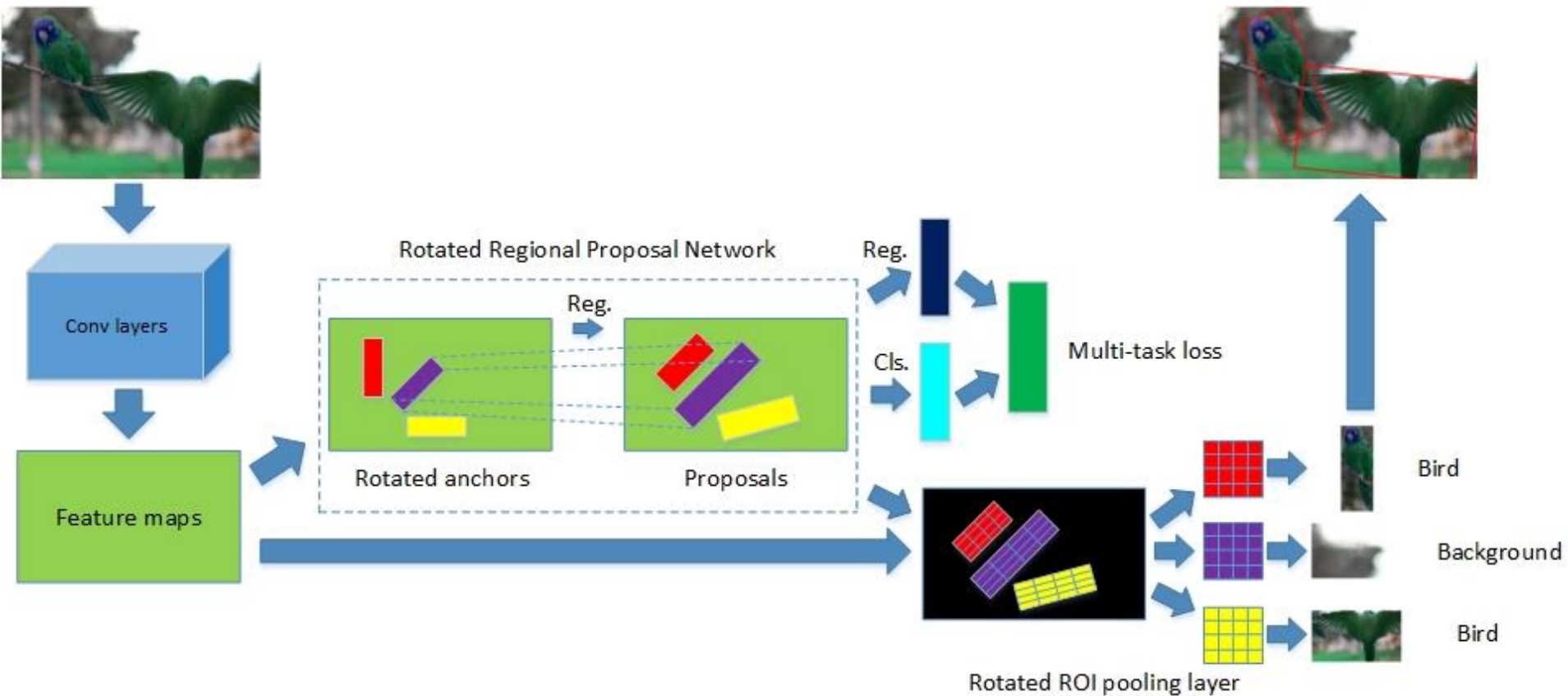
	Original dataset	New dataset
Number of images	2913	
Number of objects per image*	2.380	
Number of object-related pixel per object*	18957.812	
Percentage of object-related pixels per bounding box*	53.5%	56.7%
Bounding box's area*	35453.926	33462.425

*Average value



Model

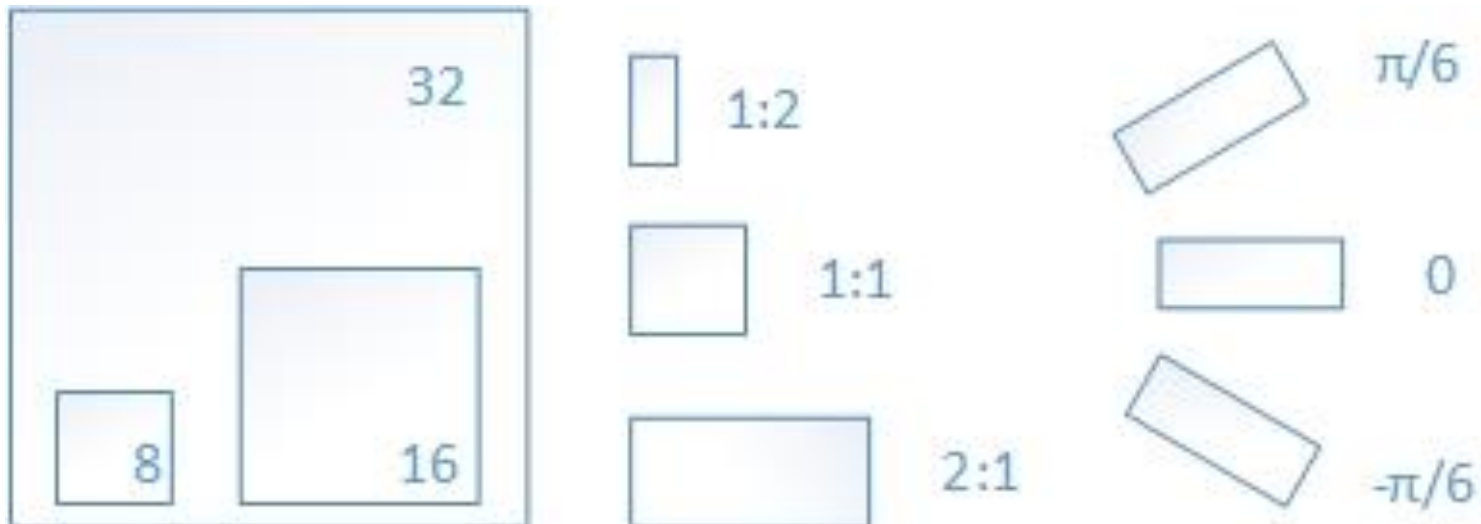
Architecture

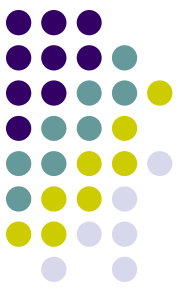




Rotation anchors

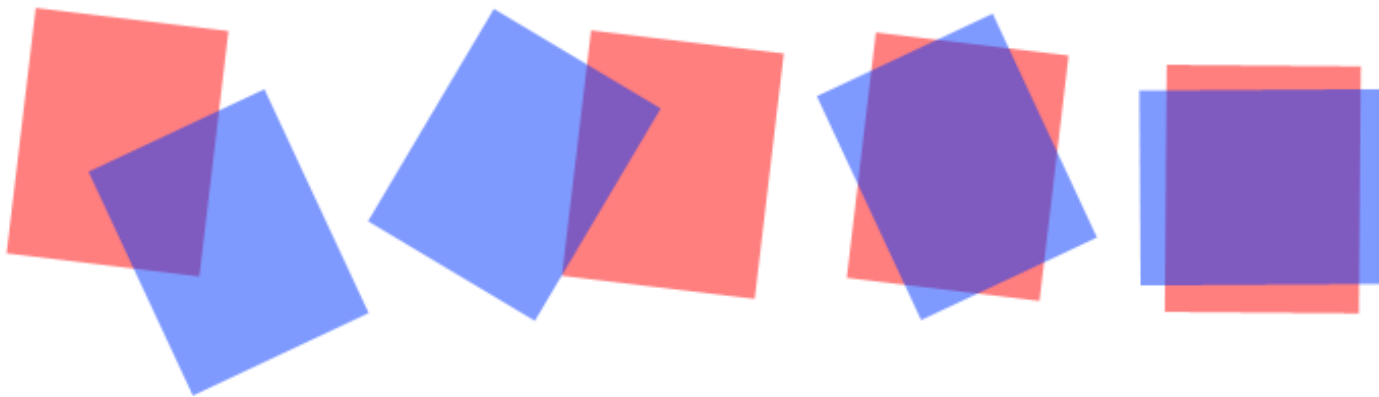
- Different scales.
- Different ratios.
- Different angles.





Intersection over Union

- Most important and difficult part: **Intersection**.
- Axis-aligned bounding boxes: **Simple**.
- Rotated bounding box: **Several challenges**.



- Algorithm still exists, but theoretically, not practically.



Intersection over Union

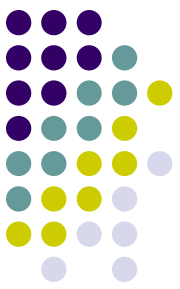
- Input: 2 rectangles (2 sets of MxN rectangles)
- Output: Intersection (MxN intersection matrix)
- Algorithm:
 - Find out vertices of the first rectangle inside the second rectangle and vice versa.
 - Find out intersection points between 2 rectangles.
 - Sort the above-found points.
 - Area of a polygon with n ordered vertices:

$$\text{Area} = \left| \frac{(x_1y_2 - x_2y_1) + (x_2y_3 - x_3y_2) + \dots + (x_ny_1 - x_1y_n)}{2} \right|$$

Skew Non-Maximum Suppression



- An important sub-module:
- 2 tasks:
 - Reduce set of proposals.
 - Assign score for proposals.
- Criteria:
 - IoU threshold (e.g. 0.7)
 - Angle difference threshold (e.g. $\pi/12$)



Rotated RoI Pooling layer

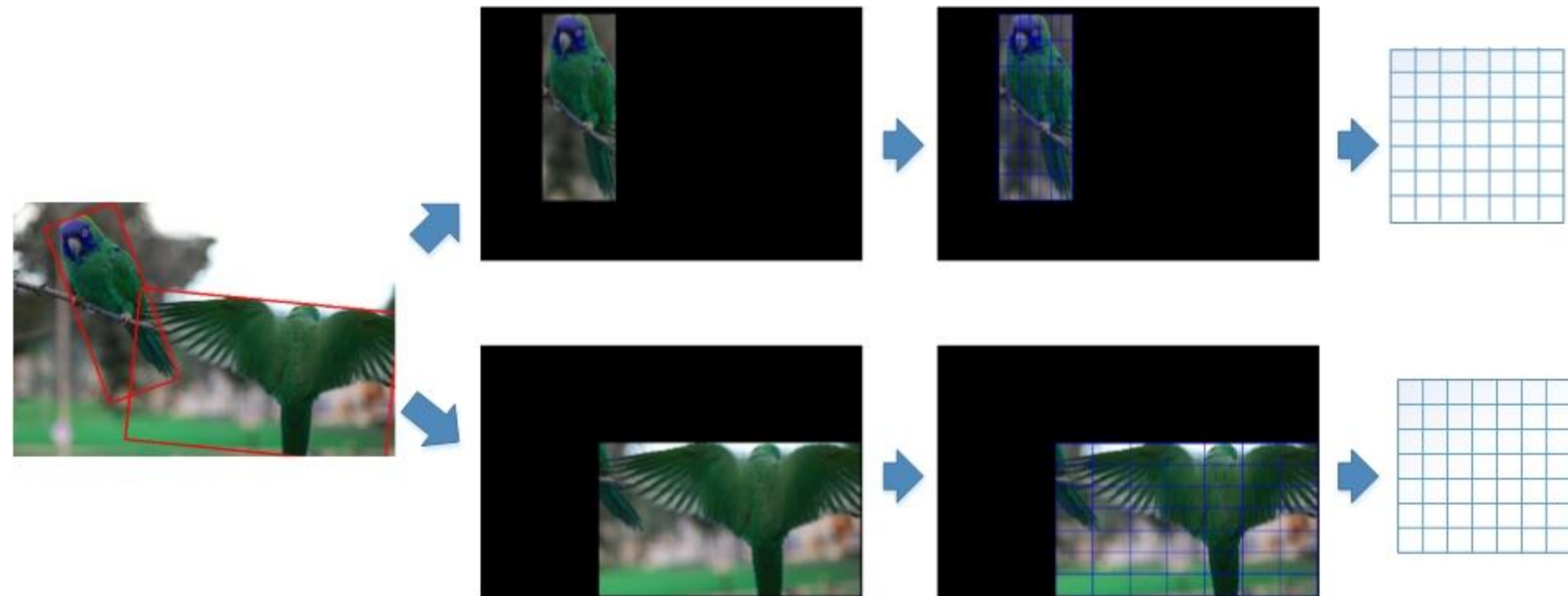
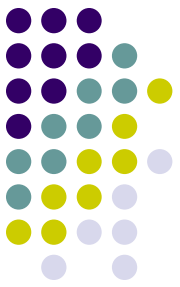
- How RoI pooling layer work ?

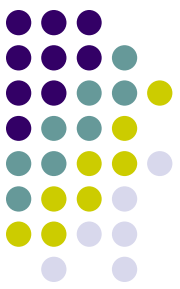
input

0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91

- How to apply the pooling operation on oriented RoIs ?

Rotated RoI Pooling layer





RPN Loss function

Softmax cross-entropy

Smooth L1 loss

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

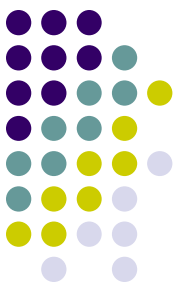
Trade-off between 2 components

Coordinates of the predicted bounding box for anchor i

Predicted probability of being an object for anchor i

True box coordinates

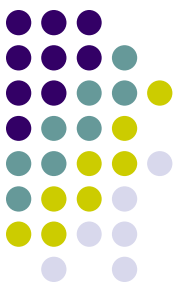
- i = anchor index in minibatch.
- N_{cls} = Number of anchor in minibatch.
- N_{reg} = Number of anchor location.



Experiment

- Base network: Pre-trained VGG16 (until conv5_3 layer).
- Dataset: VOC2012.
- Training: Learning rate starts with 0.001 and is reduced by a half after 50000 iterations. Training finishes at 70000 iterations.
- Evaluation: mAP (**M**ean **A**verage **P**recision).

Experiment

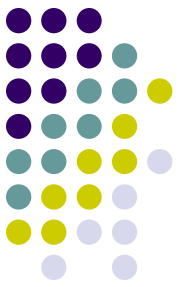


Groundtruth

Prediction



Experiment



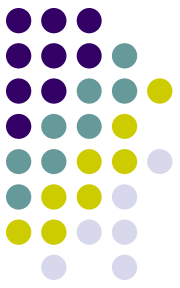
Groundtruth



Prediction



Experiment



Groundtruth



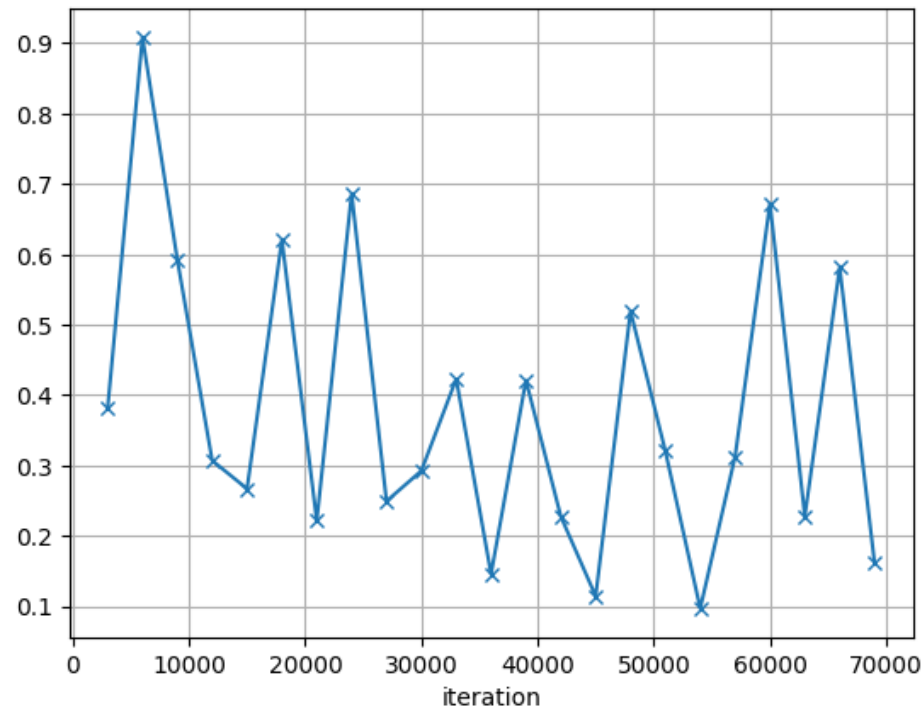
Prediction



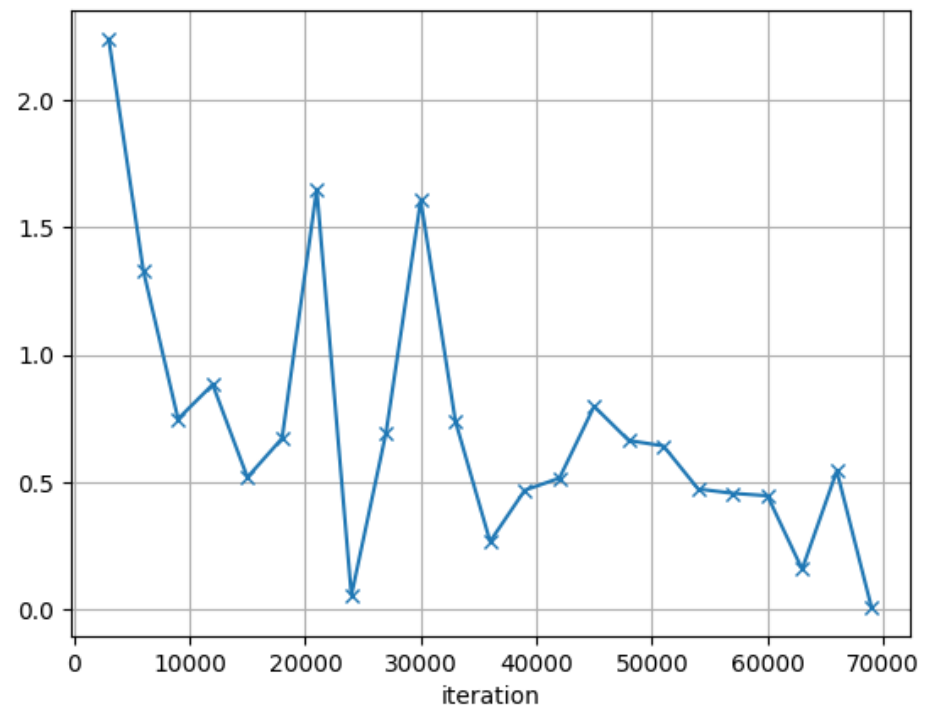
Experiment

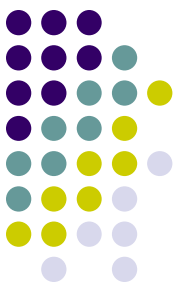


Faster R-CNN



Faster RR-CNN





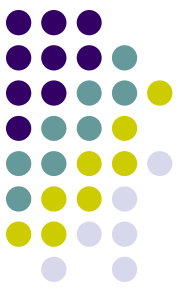
Experiment

Faster R-CNN vs Faster RR-CNN

mAP	Monitor	Car	Cat	Sheep	Train	Horse
70.0	67.6	80.4	82.0	68.3	81.1	79.8
70.6*	73.6*	80.1	79.1	72.2*	72.2	80.1*

Motorbike	Cow	Aeroplane	Person	Bus	Table	Boat
75.0	75.3	70.0	76.3	78.2	67.2	57.3
77.2*	73.3	69.2	77.7*	81.7*	67.0	59.1*

Plant	Bird	Dog	Sofa	Bottle	Chair	Bicycle
39.1	70.1	80.3	67.3	49.9	52.2	80.6
41.7*	68.4	82.3*	64.8	54.6*	53.6*	78.7



Conclusion

- Oriented object detection is a potential approach.
- Result in better performance in many cases in comparison with traditional axis-aligned object detection model
- Still have rooms for improvement:
 - Apply NMS for reducing set of proposals.
 - Fine-tune anchor generation's parameters.
 - Faster IoU and RRoi computation.



THANK YOU

Back Up

