UNIVERSITAT ROVIRA I VIRGILI (URV) Y UNIVERSITAT OBERTA DE CATALUNYA (UOC)

Computational Engineering and Mathematics

Master's Thesis

Area: Artificial Intelligence

Underwater Species Detection in Images and Videos

Author: Ricard Fos Serdà

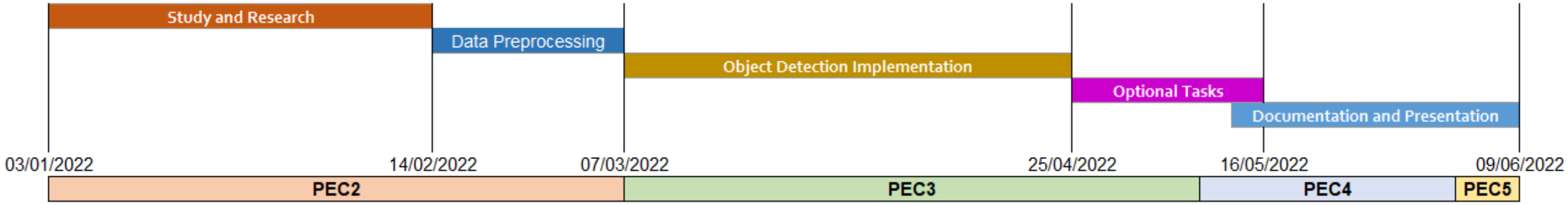
Consultant teacher: Antonio Burguera Burguera

Submission date: 01/06/2022

Introduction

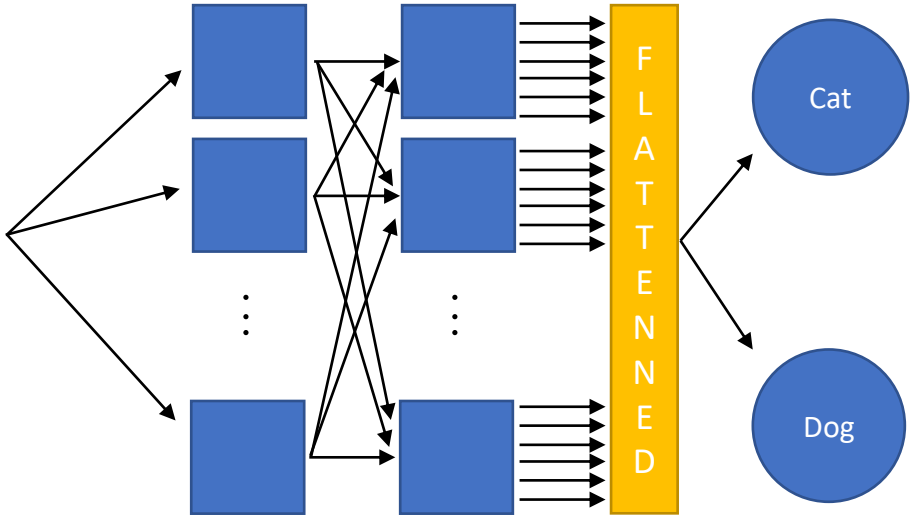
- **Deep Learning**:
  - Classify images
  - Detect Objects
  - Natural Language Processing
  - Motion Planning
  - More…
- **Underwater Species**:
  - Identify
  - Count
  - Study
  - Follow

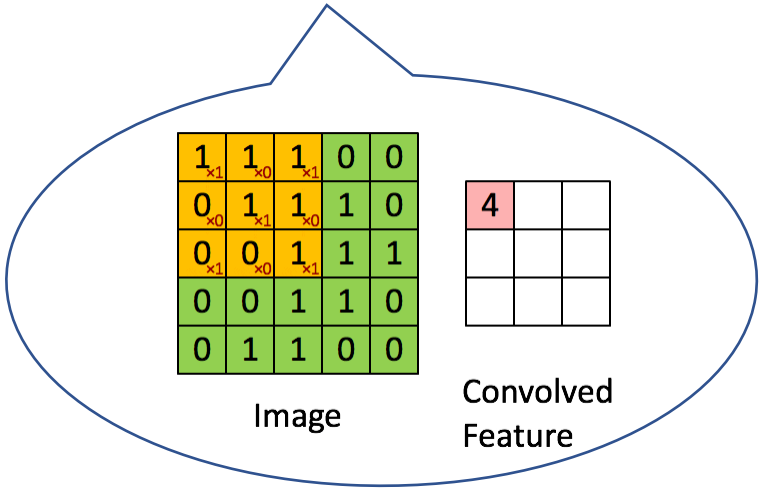
Work Planning



Convolutional Neural Networks (CNN)

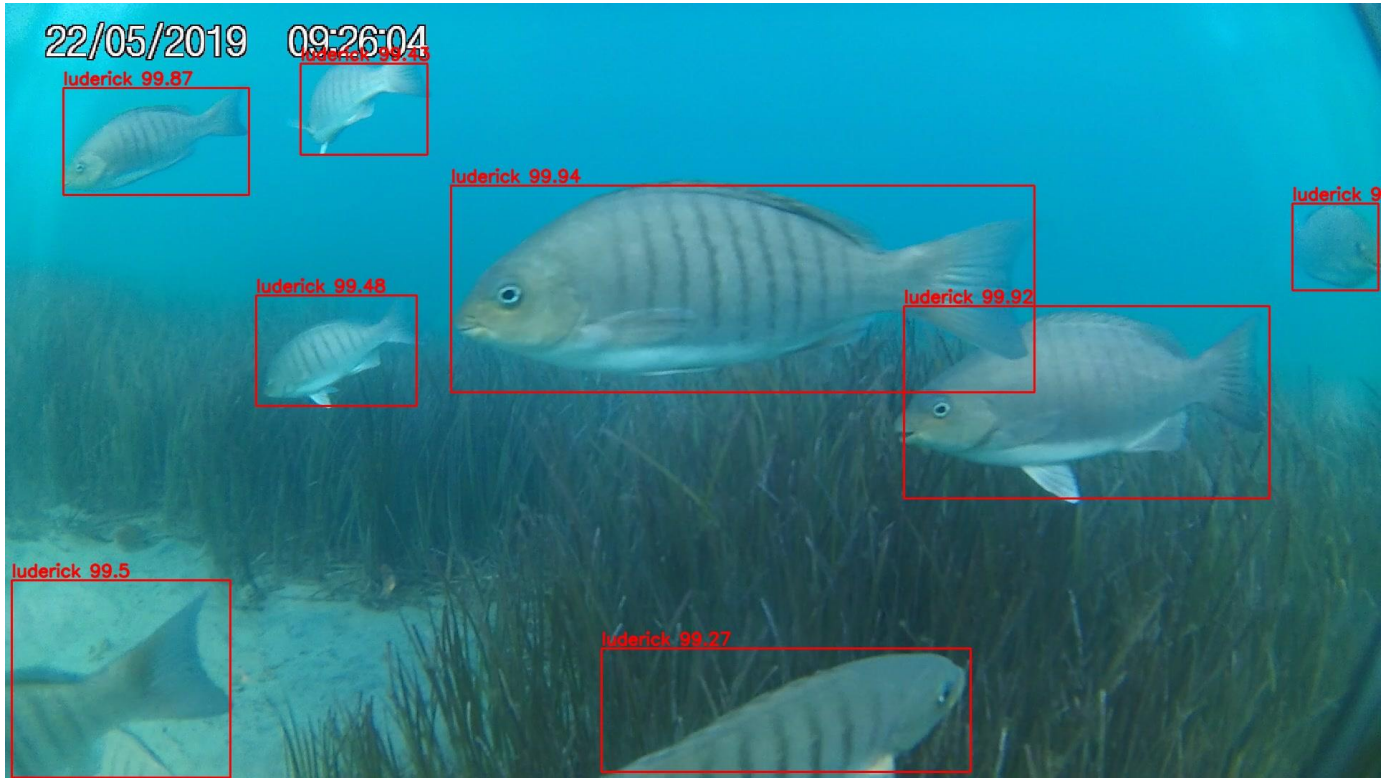
- **Image Classification**:
  1. Convolutional Layers
  2. Pooling Layers
  3. Flatten
  4. Classify





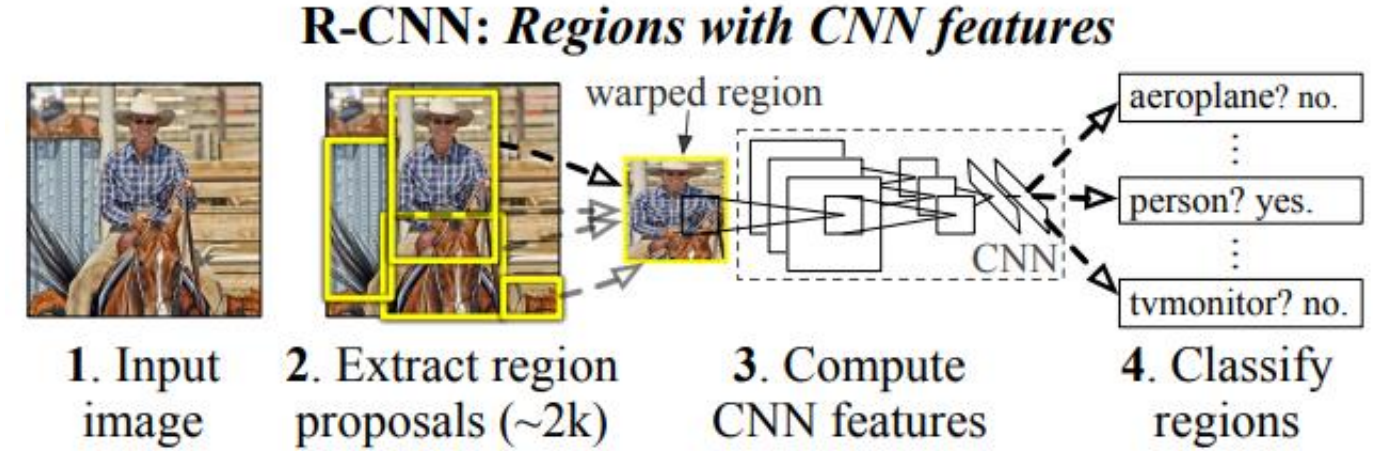
Two-Stage Object Detection

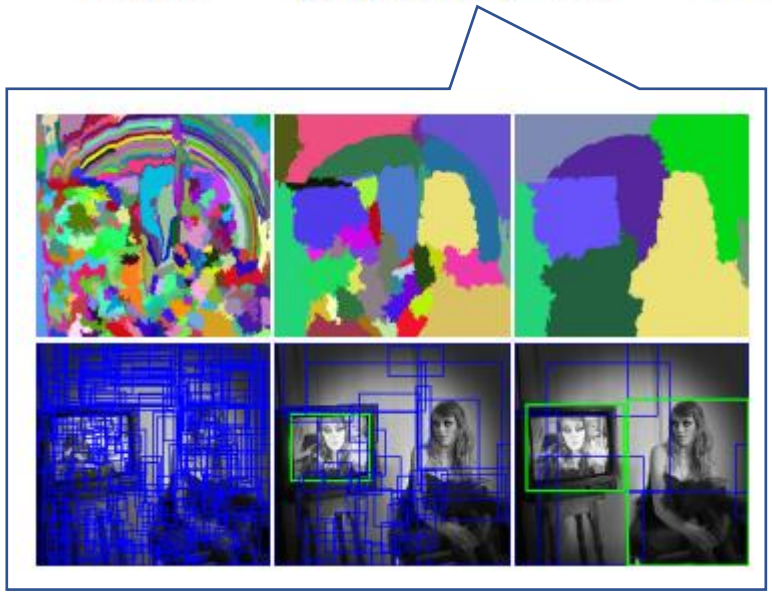
- Bounding Boxes
  - [xmin, ymin, xmax, ymax]
- Class label
- Confidence



Region-Based CNN (R-CNN)

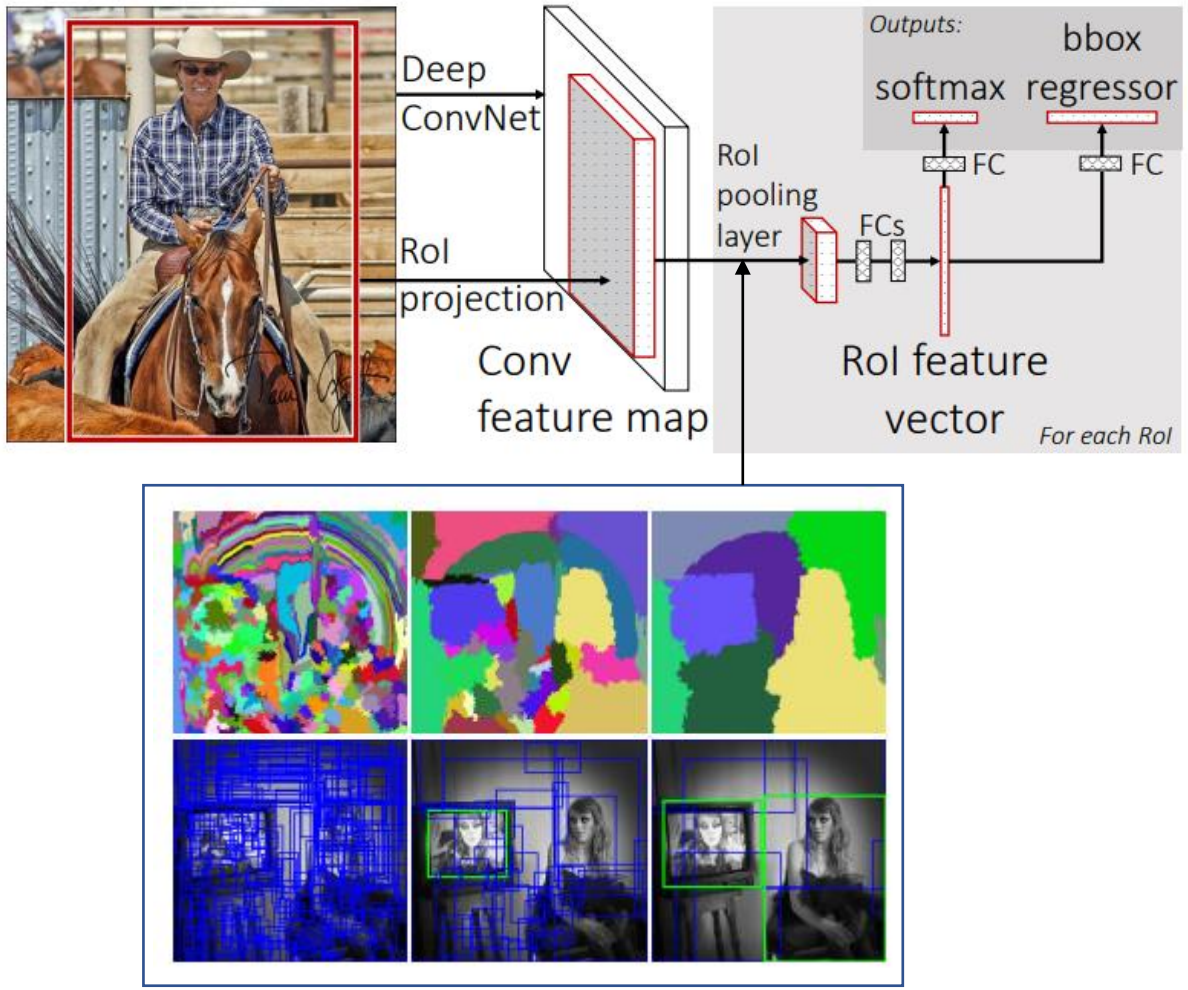
1. Region proposals
   - Selective Search
2. CNN Backbone
3. Classify
4. Box regression
5. Non-Max Suppresion





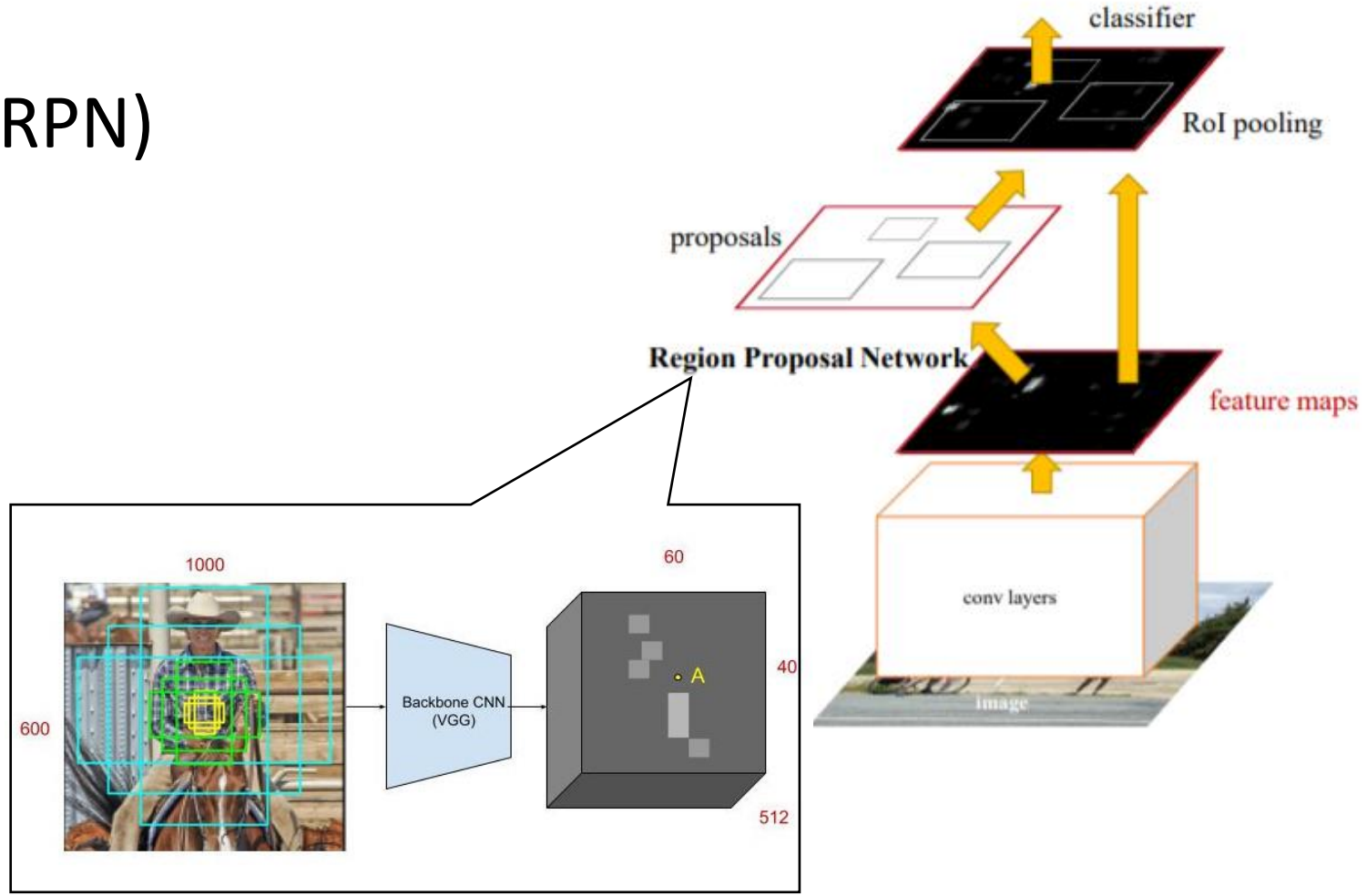
Fast R-CNN

- CNN backbone used once
- Region proposals
  - Selective Search
  - Feature Map
- Region of Interest Pooling
  - Fixed-Size ROIs



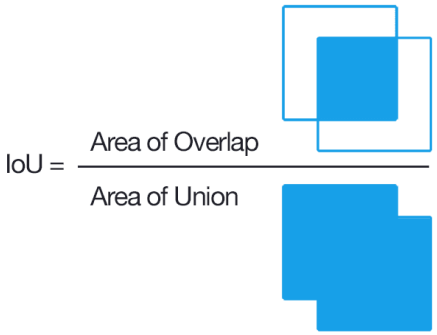
Faster R-CNN

- Region Proposal Network (RPN)
  - Anchor Boxes
  - Objectness score
  - Generate Region Proposals



Evaluation Metrics for Object Detection

- Intersection over Union (IoU)



- Precision $Precision(c) = \frac{TPc}{TPc + FPc}$
- Recall $Recall(c) = \frac{TPc}{TPc + FNc}$
- Mean Average Precision (mAP)
  - Area under the Precision-Recall Curve

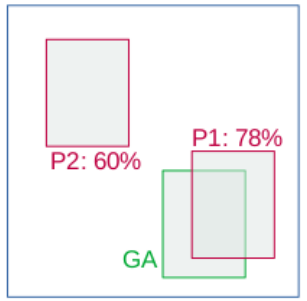


Image 1

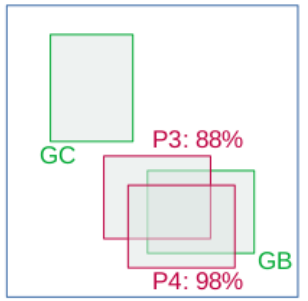


Image 2

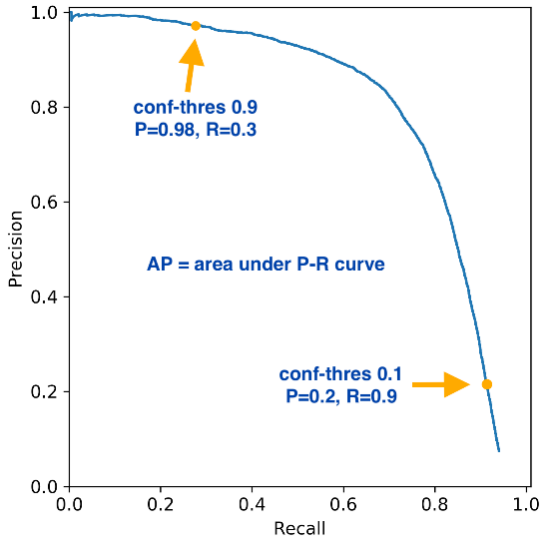


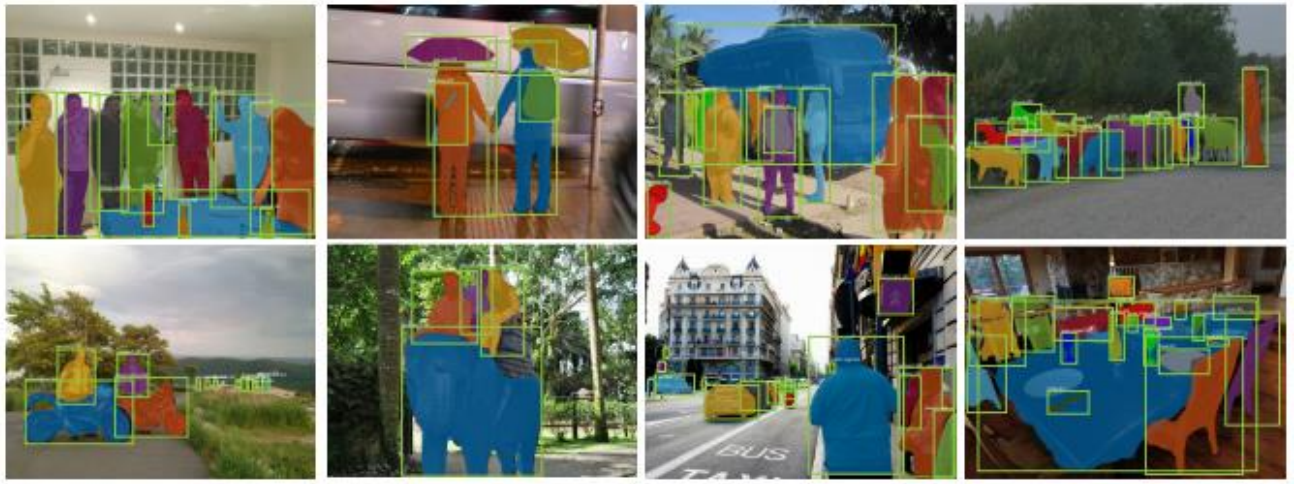
Image	Detection	Confidence	IoU	Ground Truth	TP/FP	Acc TP	Acc FP	Precision	Recall
Image 2	P4	98%	> 0.5	GB	TP	1	0	1	0.33
Image 2	P3	88%	> 0.5	GB	FP	1	1	0.5	0.33
Image 1	P1	78%	> 0.5	GA	TP	2	1	0.67	0.67
Image 1	P2	60%	< 0.5	-	FP	2	2	0.5	0.67

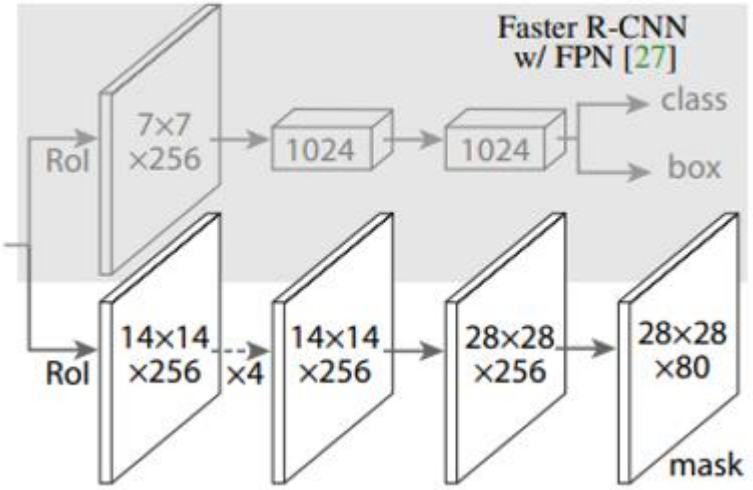
Faster R-CNN Results

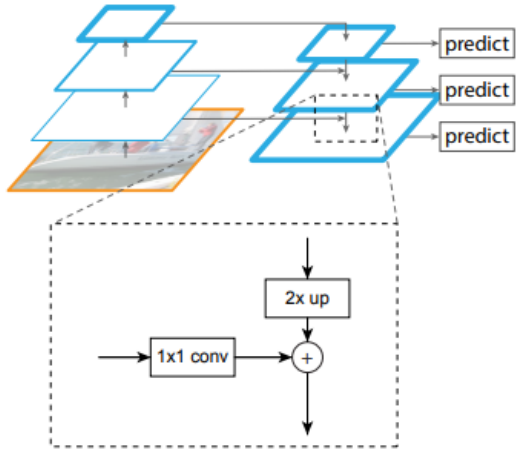
- Pytorch Library
- Transfer Learning
- **7.5 average FPS**
- **61.6 mAP**

Mask R-CNN

- Instance Segmentation
- Segmentation CNN
- Feature Pyramid Networks
- ROI Align





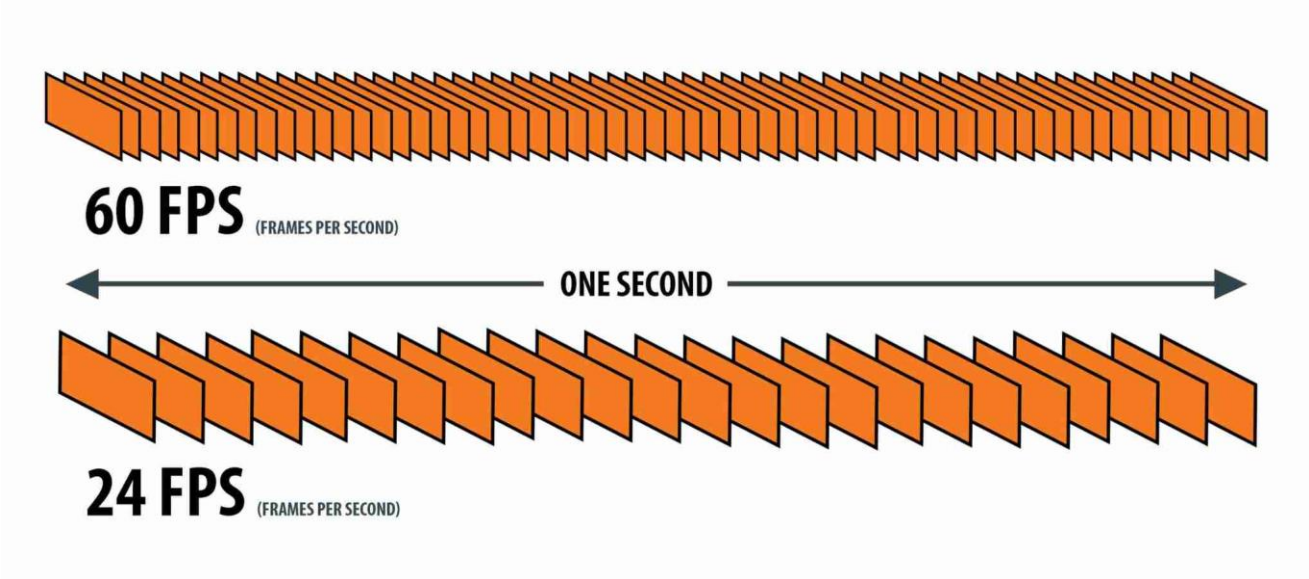


Mask R-CNN Results

- Mask data
- Pytorch Library
- Transfer Learning
- **7 average FPS**
- **61.3 mAP**

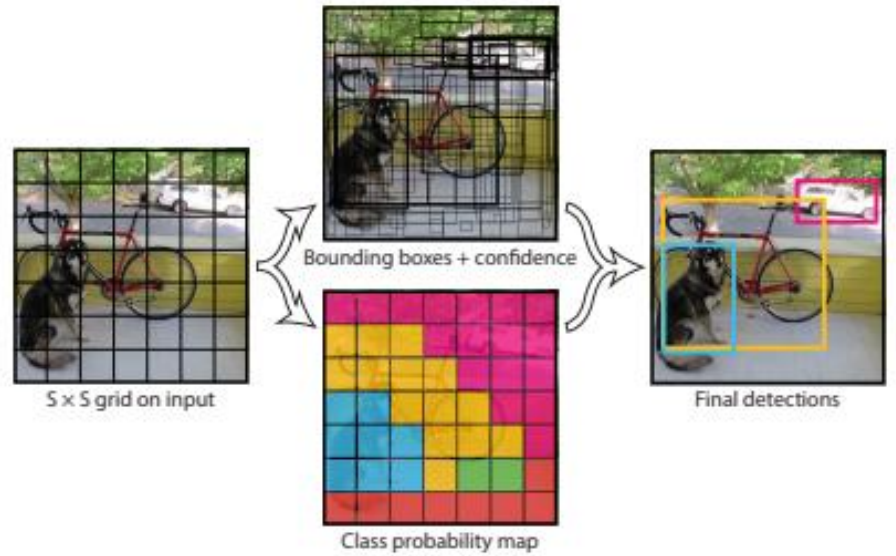
One stage detection

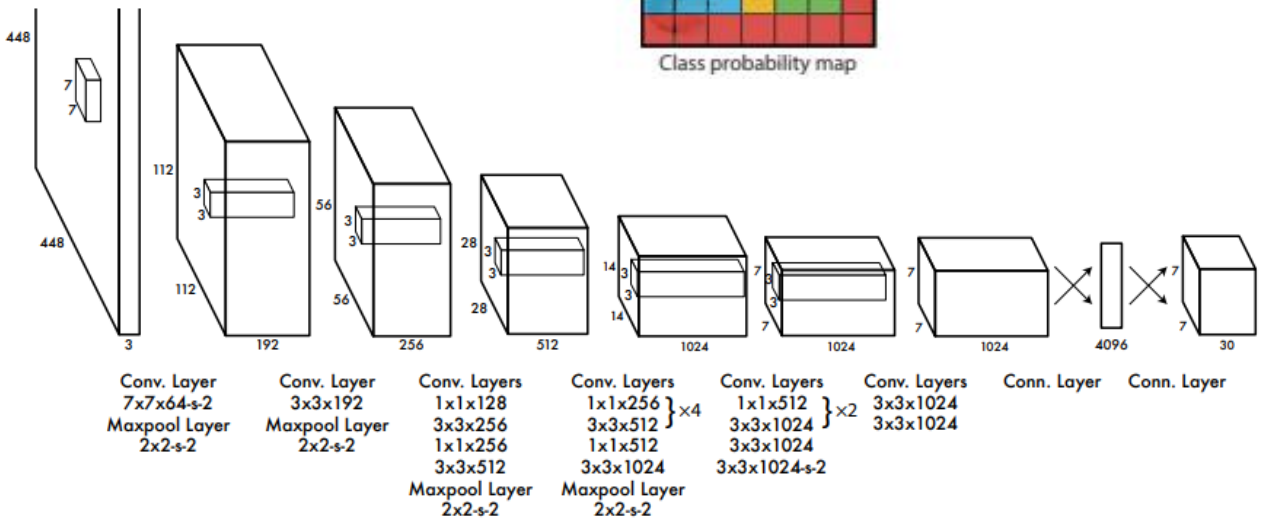
- Two-Stage detection is slow
  - 7 fps achieved
  - Want 30 fps minimum
- Avoid Region Proposal Stage



You Only Look Once (YOLO)

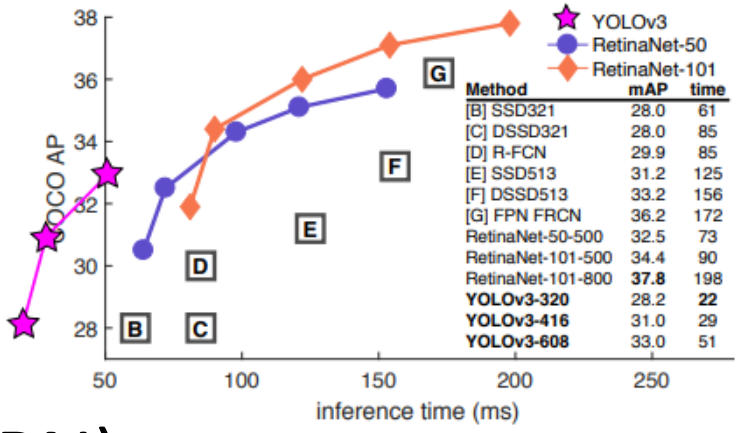
- Divide image in 7x7 grid cells
  - 1 Class predicted per cell
  - 2 boxes predicted per cell
- No anchor boxes
- No Region Proposals
- 45 FPS





YOLOv2 and YOLOv3

- Small improvements
- Anchor Boxes
- Darknet-53 backbone
- Feature Pyramid Networks (FPN)

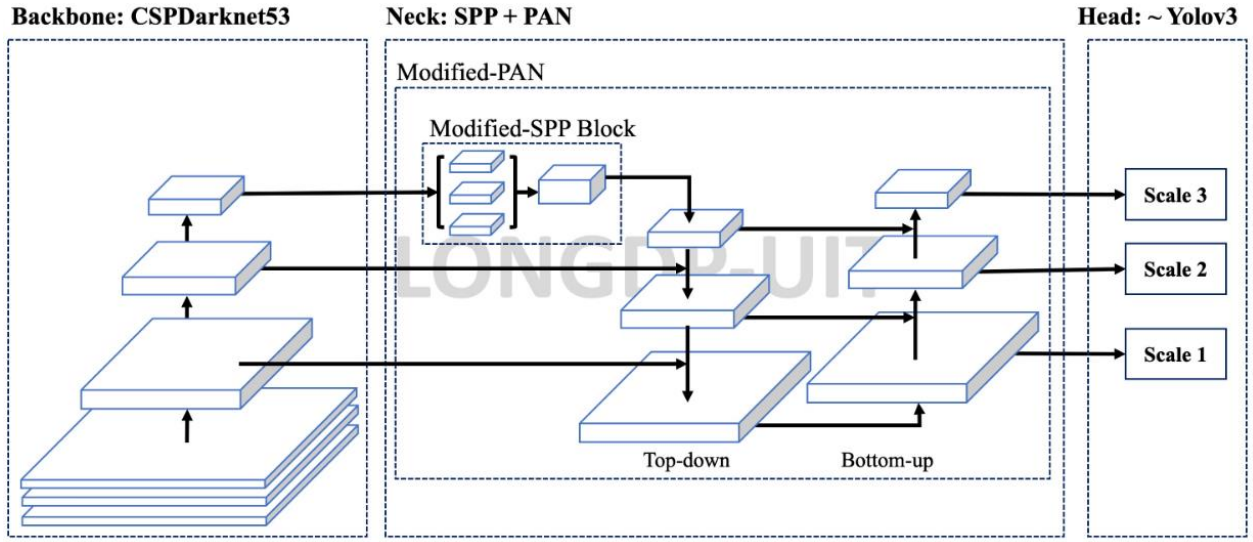


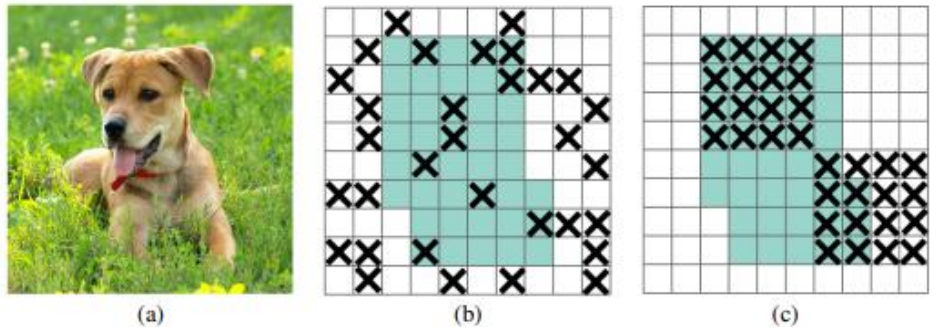
Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

YOLOv4

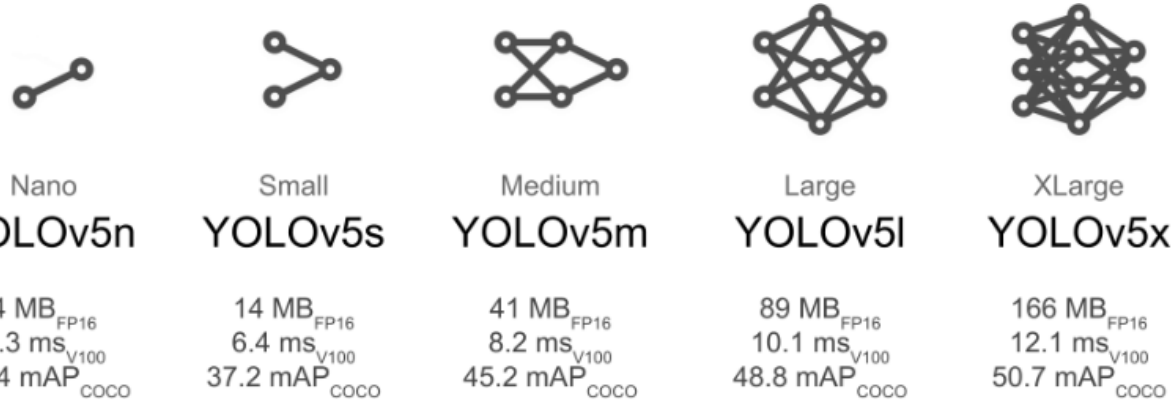
- Complex network
- Regularization
  - Data augmentation
  - DropBlock
- Modified FPN





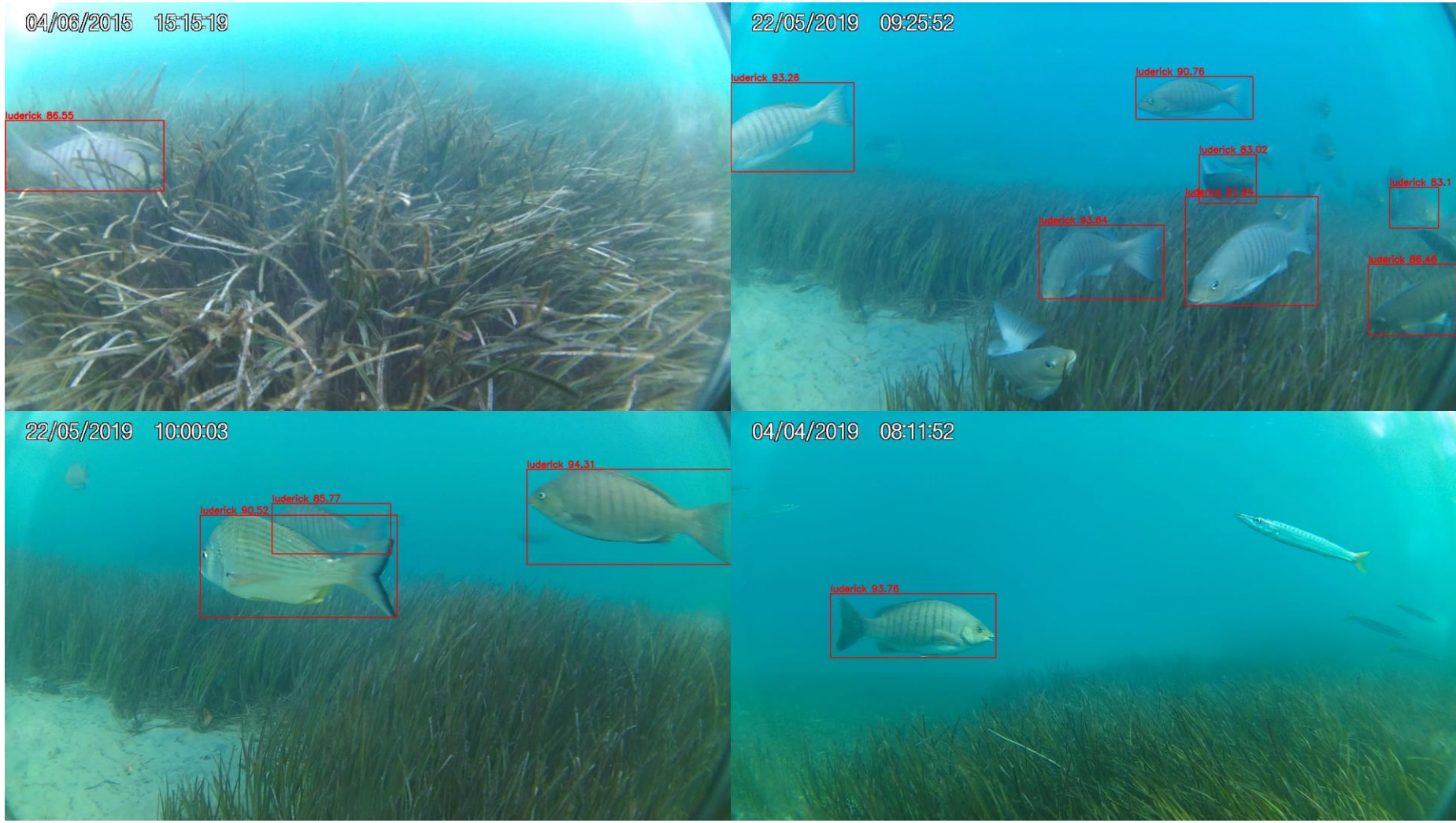
YOLOv5

- YOLOv4 with improvements
- Managed by a company
- Continuous development
- Multiple model size options
- User-Friendly



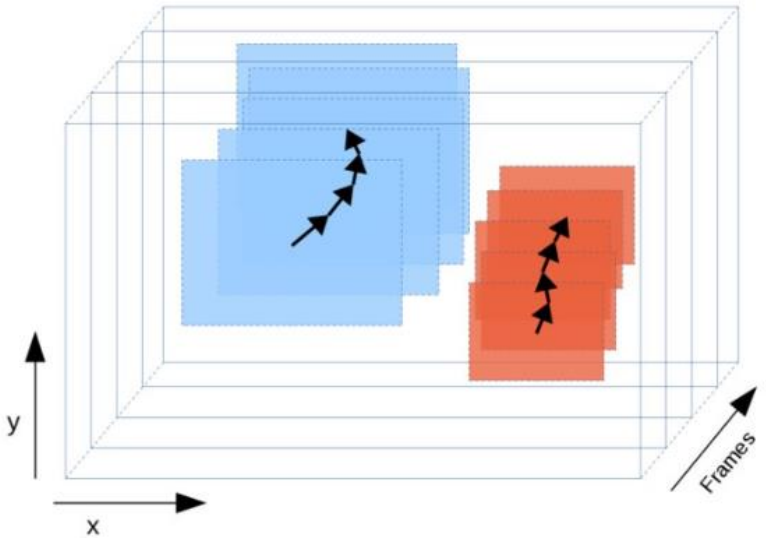
YOLOv5 Results

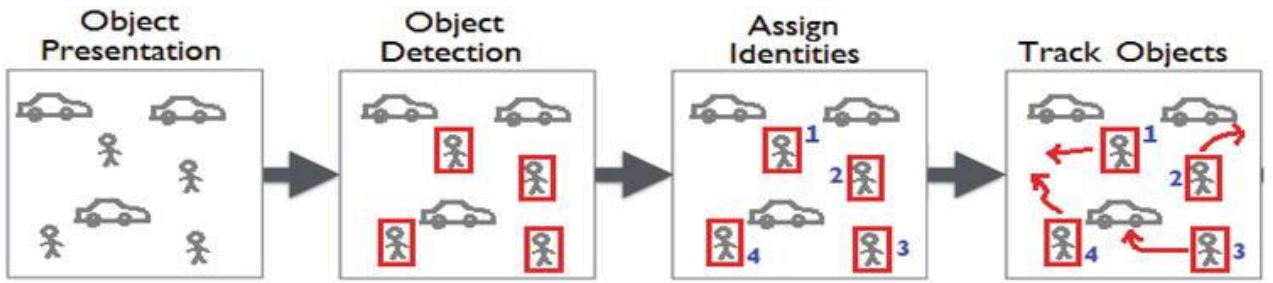
- Text input format
- YOLOv5 Github
- Small size option
- **47.3 average FPS**
- **63.4 mAP**



Object Tracking

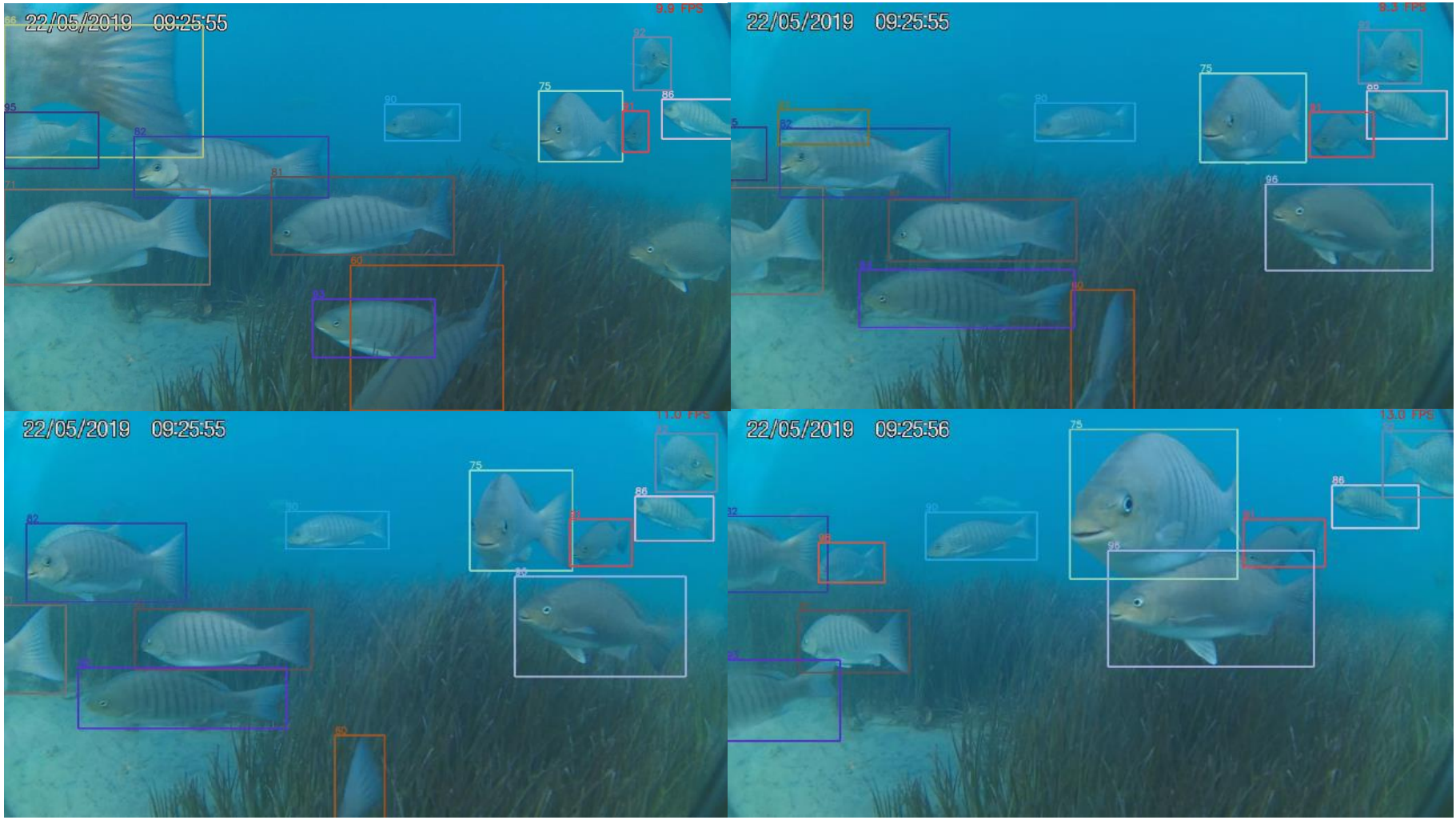
- Used on top of Detection models
- Track through frames
- IOU tracking
- Hungarian Algorithm
- Kalman Filters
- SORT





Tracking Results

- YOLOv5 detection
- Simple SORT
- No Kalman Filters
- **23.6 average FPS**



Conclusions

- YOLOv5 Achieves precise real-time detection
- Detection can be used for tracking and other purposes
- Deep Learning is replacing traditional computer vision
- New solutions will probably appear each year