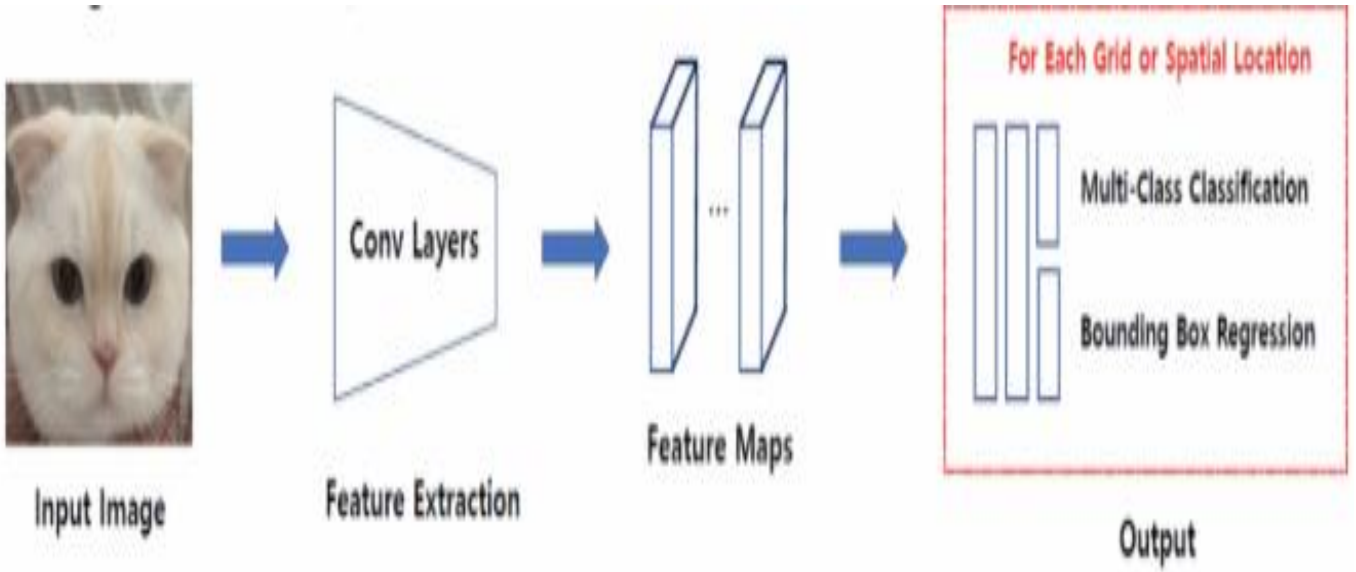
Side-Aware Boundary Localization for More Precise Object Detection

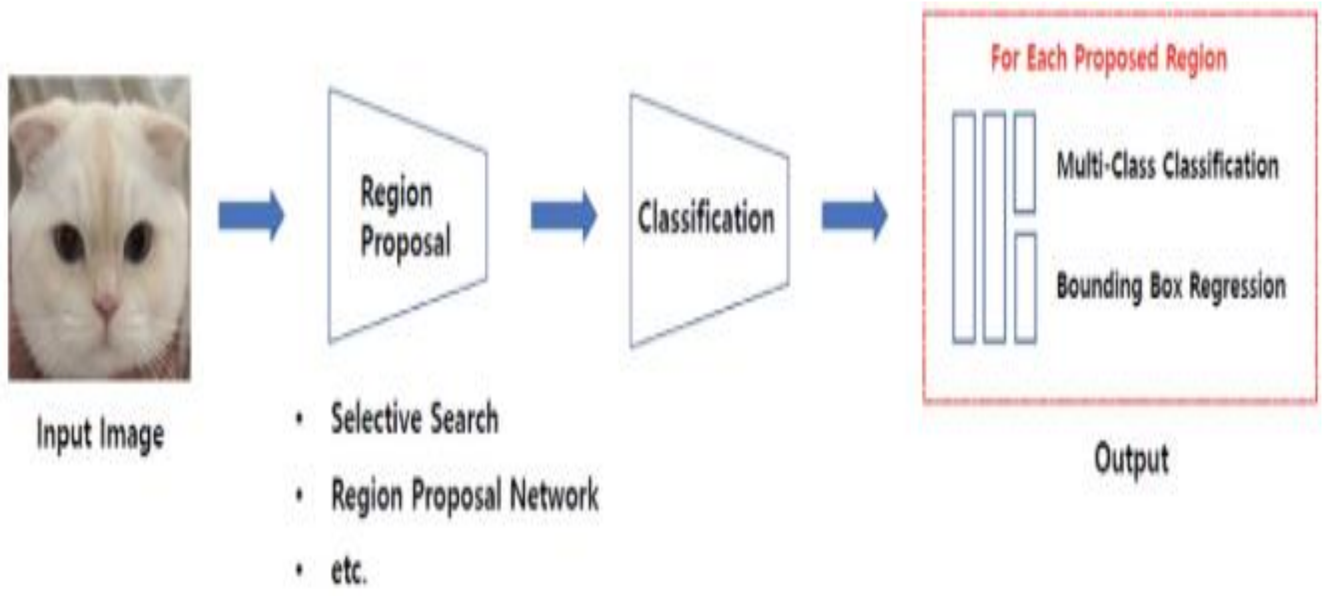
1-stage Detector

- Solve classification and localization problems at the same time
- Fast but low accuracy



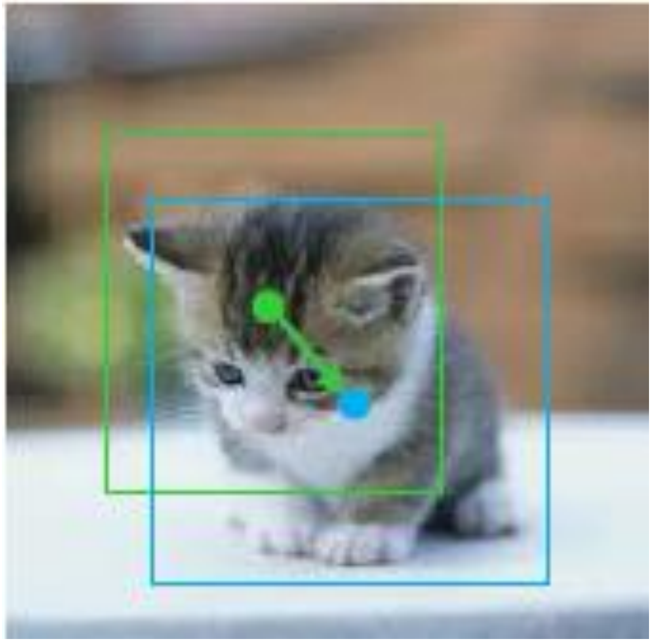
2-stage Detector

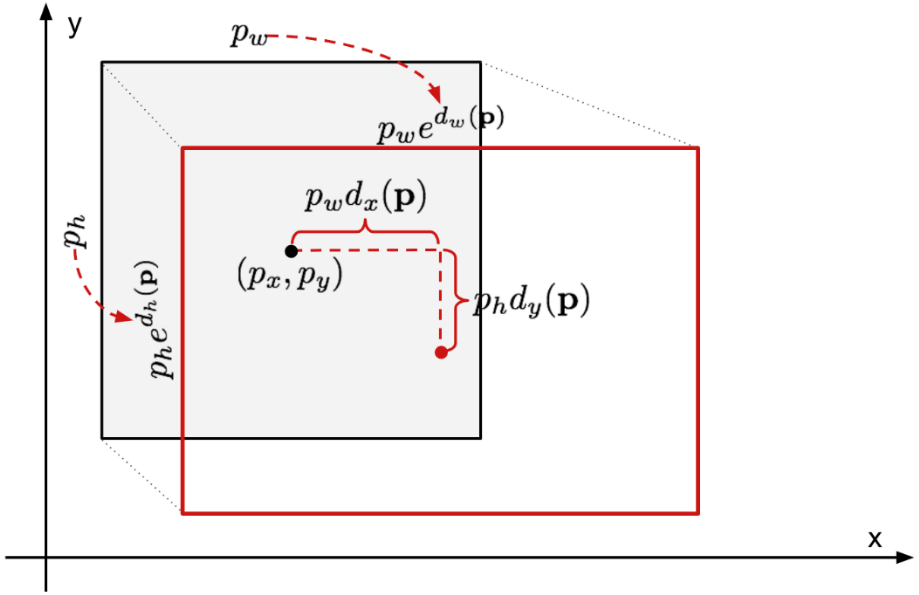
- Classification and localization problems are solved sequentially.
- Slow but high accuracy



Bounding Box Regression

- Predicts the offsets of the centers and size based on the features of RoI.
- Limit performance when there exists displacements with large variance between anchors and target.



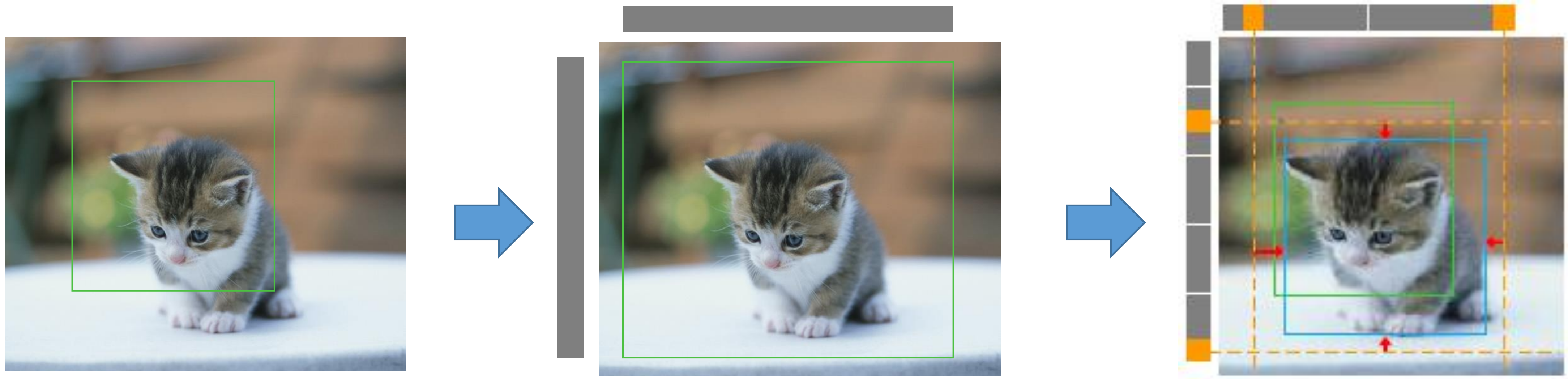


Proposal Box

Target Box

Side-Aware Boundary Localization

- Align each side of the box to the object boundary.
- Divide target space into multiple buckets.
- Select the correct bucket based and perform fine regression.

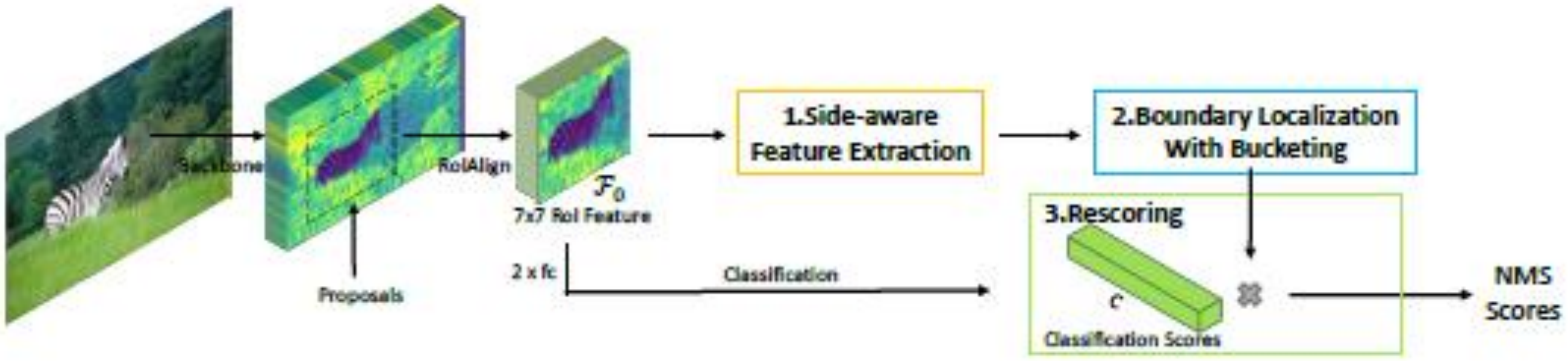


Proposal Box

Target Box

Pipeline of Side-Aware Boundary Localization

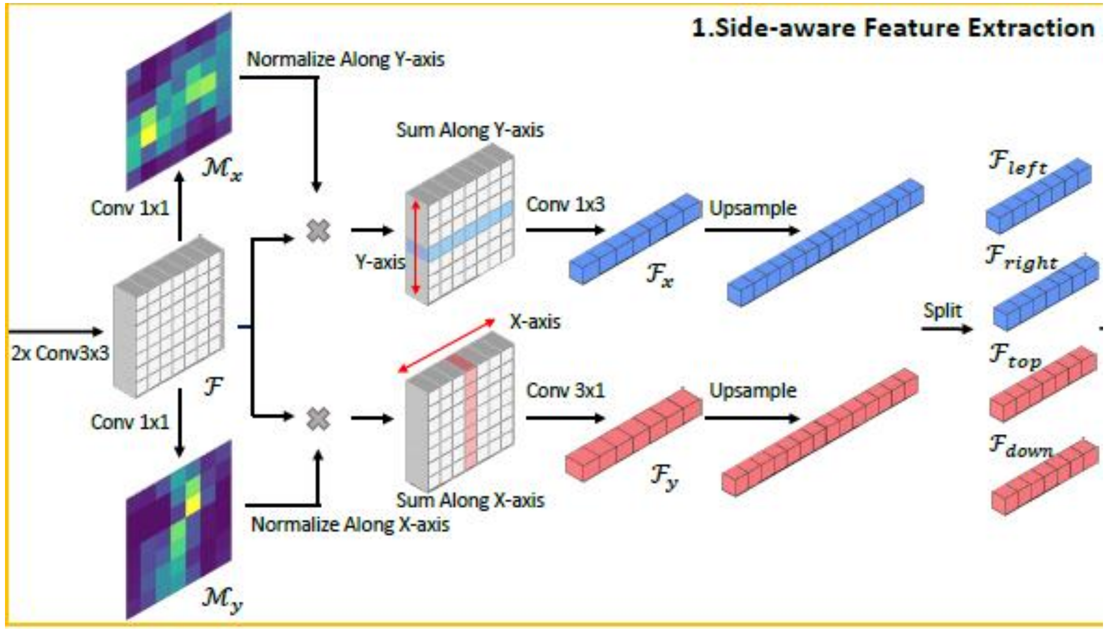
- RoI features are aggregated to produce side-aware features.
- Boundary Localization is performed to localize the boundaries by bucketing scheme.
- The confidences of buckets are adopted to assist the classification scores.



$$\mathcal{L} = \lambda_1 \mathcal{L}_{rpn} + \mathcal{L}_{cls} + \lambda_2 (\mathcal{L}_{bucketing} + \mathcal{L}_{reg})$$

Side-Aware Feature Extraction

- Extracts horizontal and vertical features F_x and F_y .
- Employ self-attention to enhance the RoI feature.
- Splits F_x and into F_y side-aware features F_{left}, F_{right}, F_{top}, and F_{down}.

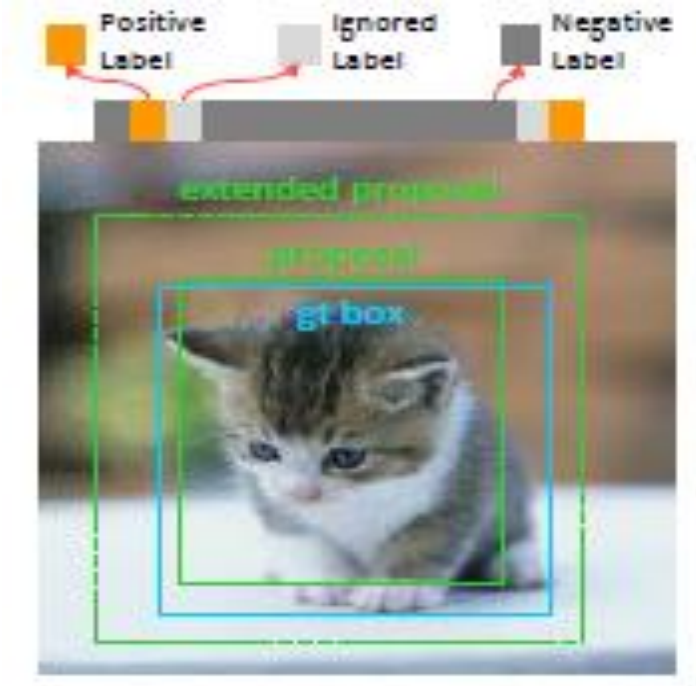


$$\mathcal{F}_x = \sum_y \mathcal{F}(y,:) * \mathcal{M}_x(y,:),$$

$$\mathcal{F}_y = \sum_x \mathcal{F}(:,x) * \mathcal{M}_y(:,x).$$

Boundary Localization with Bucketing

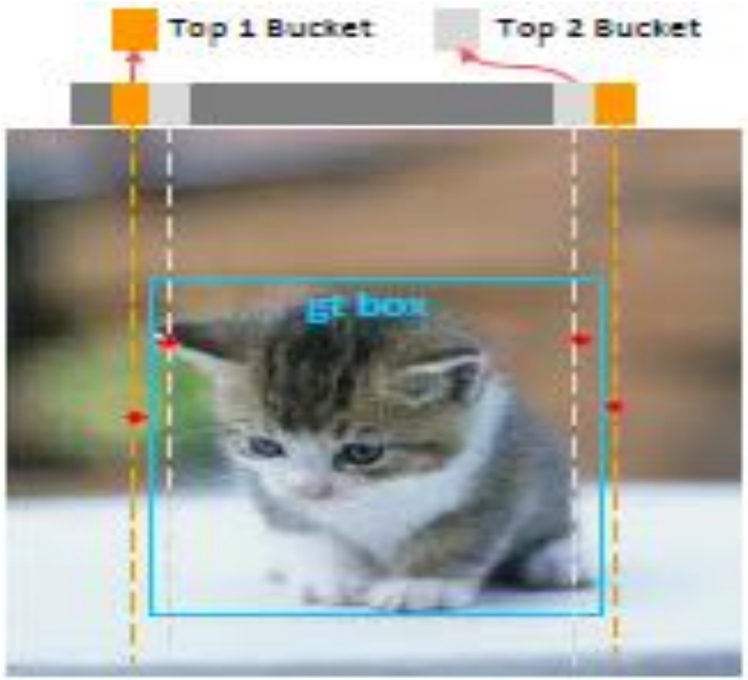
- Relax the candidate region of boundaries by a scale factor to cover the entire object.
- Candidate regions are divided into buckets.
- Adopt a binary classifier to predict correct buckets based on the side-aware features.



Boundary Localization with Bucketing

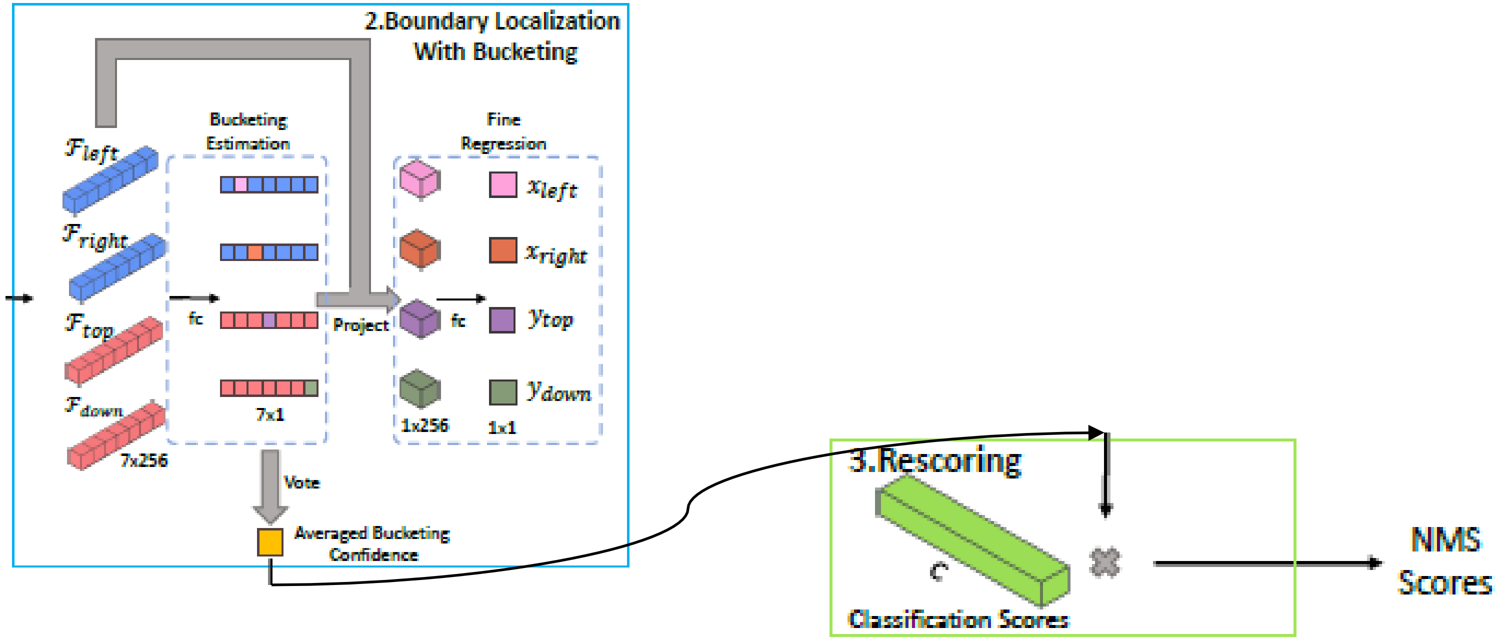
- Apply a regresser to predict the offset from the centerline of the selected bucket to the ground-truth boundary.

- To increase the robustness, include both the nearest and the second nearest bucket to train the regressor.



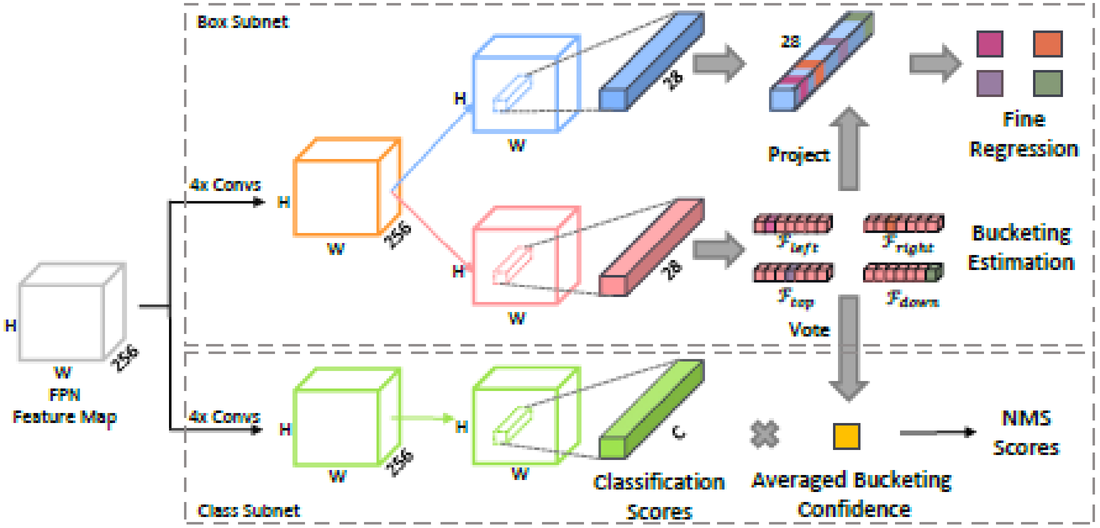
Bucketing-Guided Rescoring

- Bucketing estimation confidences can represent the reliability of predicted locations.
- SABL averages the bucketing estimation confidence scores of four boundaries.
- Rescoring helps maintain the best box with both high classification confidence and accurate localization.



Application to Single-Stage Detectors

- On top of the FPN features, four convolution layers are adopted to classification and localization.
- For training, use 9 anchors to cover more ground-truths for this location.
- Use only one anchor for detection.



Experiments

Comparison to mainstream methods with ResNet-101 FPN backbone

Method	Backbone	Sch.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FPS
RetinaNet [20]	ResNet-101	1x	38.8	60.0	41.7	21.9	42.1	48.6	13.0
FSAF [42] (m.s.)	ResNet-101	1.5x	40.9	61.5	44.0	24.0	44.2	51.3	12.4
FCOS [30] (m.s.)	ResNet-101	2x	41.5	60.7	45.0	24.4	44.8	51.6	13.5
GA-RetinaNet [32] (m.s.)	ResNet-101	2x	41.9	62.2	45.3	24.0	45.3	53.8	11.7
CenterNet [41] (m.s.)	Hourglass-104	50e	42.1	61.1	45.9	24.1	45.5	52.8	8.9
FoveaBox [17] (m.s.)	ResNet-101	2x	42.0	63.1	45.2	24.7	45.8	51.9	12.8
RepPoints [35] (m.s.)	ResNet-101	2x	42.6	63.5	46.2	25.4	46.2	53.3	12.2
RetinaNet w/ SABL	ResNet-101	1x	40.5	59.3	43.6	23.0	44.1	51.3	13.0
RetinaNet w/ SABL (m.s.)	ResNet-101	1.5x	42.7	61.4	46.0	25.3	46.8	53.5	13.0
RetinaNet w/ SABL (m.s.)	ResNet-101	2x	**43.2**	62.0	46.6	25.7	47.4	53.9	13.0

- Dataset : COCO
- m.s. : multi-scale training
- Sch : training schedule

Comparison to mainstream methods with ResNet-101 FPN backbone

Method	Backbone	Data	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FPS
Faster R-CNN [28]	ResNet-101	val	38.5	60.3	41.6	22.3	43.0	49.8	13.8
Faster R-CNN [28]	ResNet-101	test-dev	38.8	60.9	42.3	22.3	42.2	48.6	13.8
IoU-Net [16]	ResNet-101	val	40.6	59.0	-	-	-	-	-
GA-Faster R-CNN [32]	ResNet-101	test-dev	41.1	59.9	45.2	22.4	44.4	53.0	11.5
Grid R-CNN Plus [24]	ResNet-101	test-dev	41.4	60.1	44.9	23.4	44.8	52.3	11.1
Faster R-CNN w/ SABL	ResNet-101	val	41.6	59.5	45.0	23.5	46.5	54.6	12.4
Faster R-CNN w/ SABL	ResNet-101	test-dev	**41.8**	60.2	45.0	23.7	45.3	52.7	12.4
Cascade R-CNN [1]	ResNet-101	test-dev	42.4	61.1	46.1	23.6	45.4	54.1	11.2
Cascade R-CNN w/ SABL	ResNet-101	test-dev	**43.3**	60.9	46.2	23.8	46.5	55.7	8.8

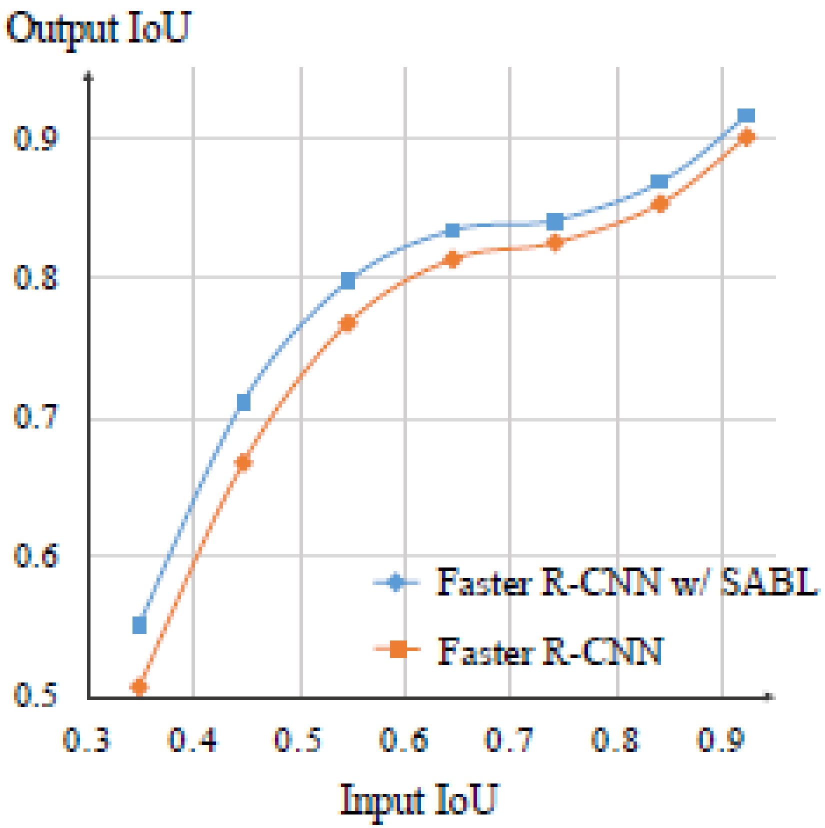
- Dataset : COCO
- m.s. : multi-scale training
- Sch : training schedule

The effects of each module

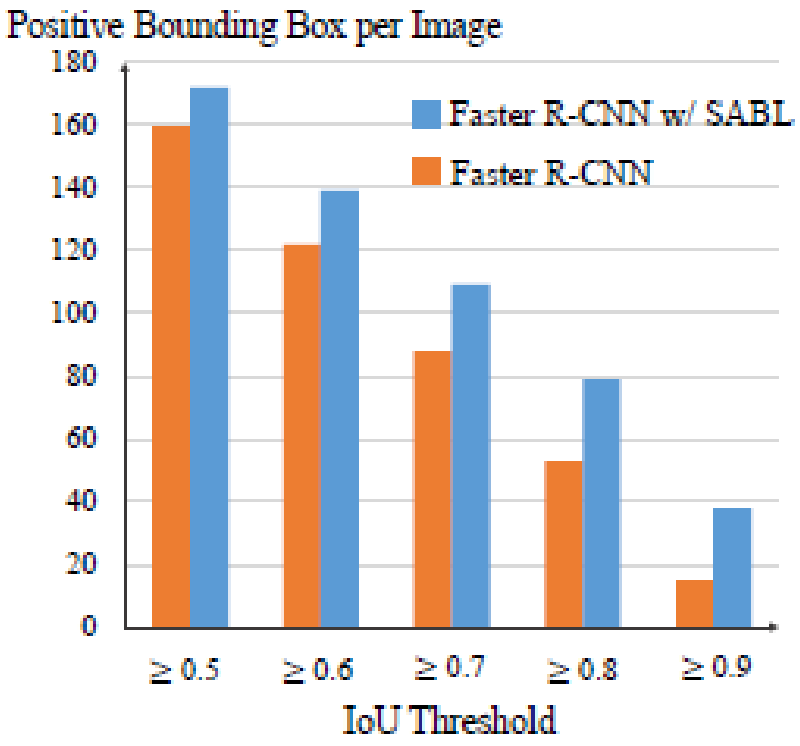
SAFE	BLB	BGR	AP	AP_{50}	AP_{75}	AP_{90}	AP_S	AP_M	AP_L
			36.4	58.4	39.3	8.3	21.6	40.0	47.1
✓			38.5	58.2	41.6	14.3	23.0	42.5	49.5
	✓		38.3	57.6	40.5	16.1	22.3	42.6	49.7
	✓	✓	39.0	57.5	41.9	17.1	22.6	43.2	50.9
✓	✓		39.0	57.9	41.4	17.8	22.7	43.4	49.9
✓	✓	✓	39.7	57.8	42.8	18.8	23.1	44.1	51.2

- SAFE : Side-Aware Feature Extraction
- BLB : Boundary Localization with Bucketing
- BGR : Bucketing-Guided Rescoring

Average IoU of proposals before and after localization branch



Number of positive boxes per image with different IoU threshold after localization



Conclusion

- Focus on the content of boundaries for localization
- A Lightweight two-step bucketing scheme.