R2SNet: Scalable Domain Adaptation for Object Detection in Cloud-Based Robotic Ecosystems via Proposal Refinement

MICHELE ANTONAZZI michele.antonazzi@unimi.it
MATTEO LUPERTO matteo.luperto@unimi.it
N. ALBERTO BORGHESE alberto.borghese@unimi.it
NICOLA BASILICO nicola.basilico@unimi.it

Department of Computer Science, University of Milan

IROS '24 ABU DHABI

Introduction

Context

- We consider a fleet of robots deployed in different indoor environments that need to perform object detection
- This ability is essential to carry out high-level tasks useful in several contexts[1]

Service Robots

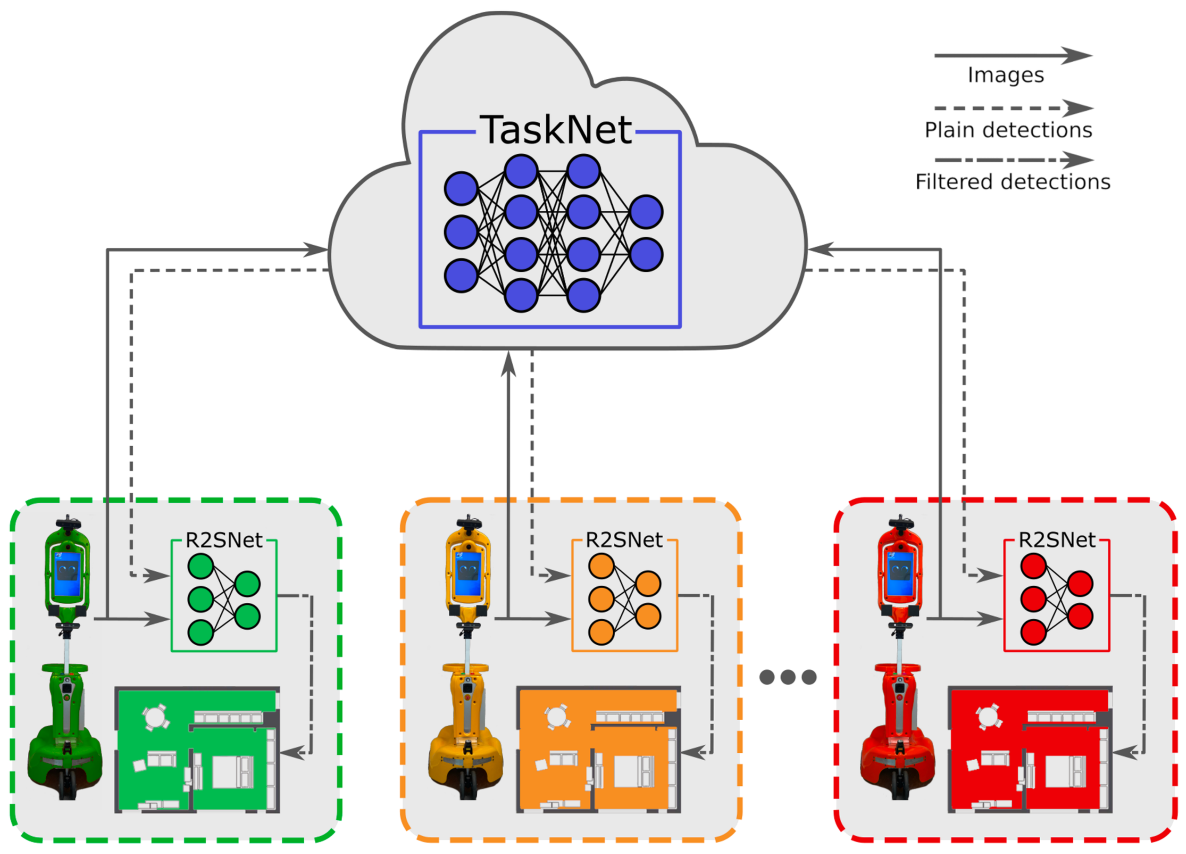
Assistive Robots

Robots as Computationally Limited Autonomous Agents

- A straightforward approach is to plug and play publicly-available Deep Neural Networks (DNNs) for object detection (OD)
- Running deep learning-based models on mobile robots is prohibitive
  - Low-powered and affordable hardware configuration
  - Limited computational capabilities affect real-time inference
  - Energy-preservation constraints for long-term autonomy

Cloud Robotics

- Offloading computationally intensive inference tasks to third-party cloud services running DNNs, here called TaskNets[2]
- Domain shift degrades the TaskNet's performance
- Classical domain adaptation[3] cannot be applied
  - The TaskNet is inaccessible
  - Train, deploy, and maintain a TaskNet for each robot is expensive

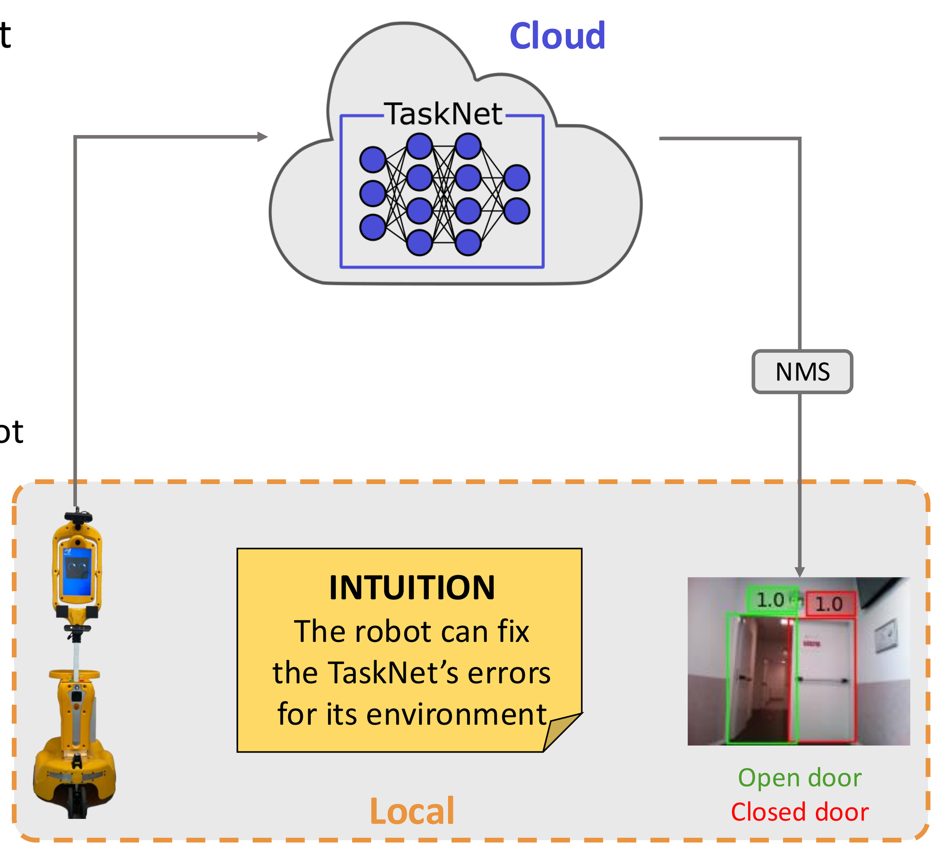


Preliminaries

Object Detection over the Cloud

- The robot sends remotely its perceptions (RBG images)
- The TaskNet predicts a dense set of object proposals $\hat{Y} = \{\hat{y}\}$
- Bounding boxes are expressed as $\hat{y} = [\hat{c}_x, \hat{c}_y, \hat{w}, \hat{h}, \hat{c}, \text{hot}(\hat{o})]$
  - $\hat{c}_x, \hat{c}_y$ are the center coordinates
  - $\hat{w}, \hat{h}$ are width and height
  - $\hat{c}, \text{hot}(\hat{o})$ are the confidence and the one-hot encoded label
- $\hat{Y}$ is filtered using Non-Maximum Suppression (NMS)
- The remaining bounding boxes are sent back to the robot

ISSUES
- NMS produces several errors
- Each robot incurs in a different domain shift

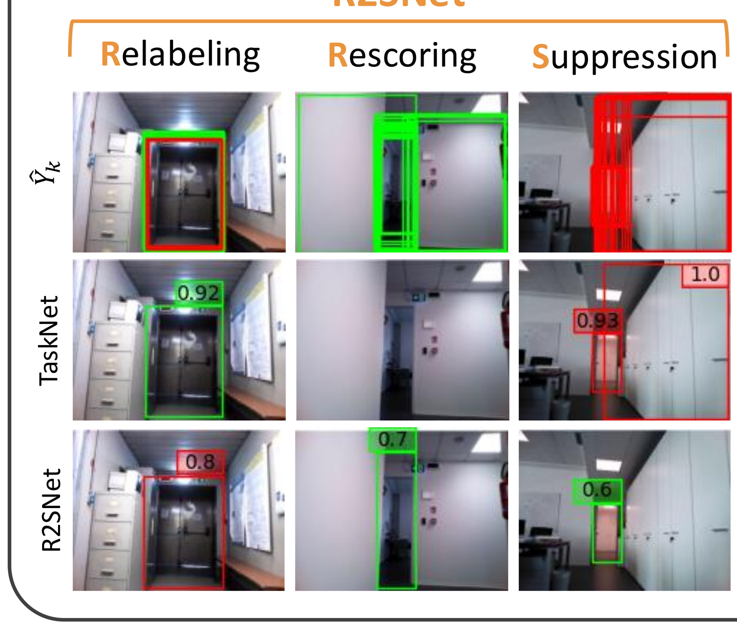


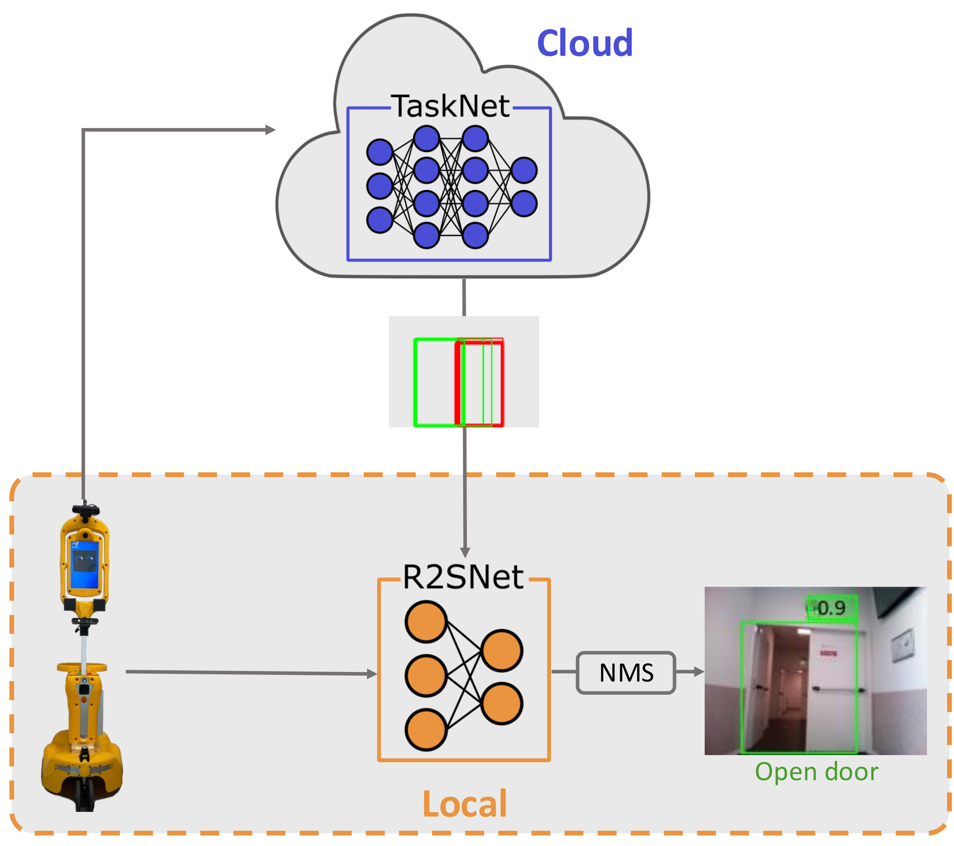
Approach

Downstream Proposal Refinement

- The robot receives $\hat{Y}$ and selects the first k most confident, obtaining $\hat{Y}_k$
- It refines their parameters with a lightweight DNN which performs 3 corrective actions
- $\hat{Y}_k$ is then filtered with NMS

BENEFITS
- Horizontally-scalable adaptation
- Computationally affordable by robots

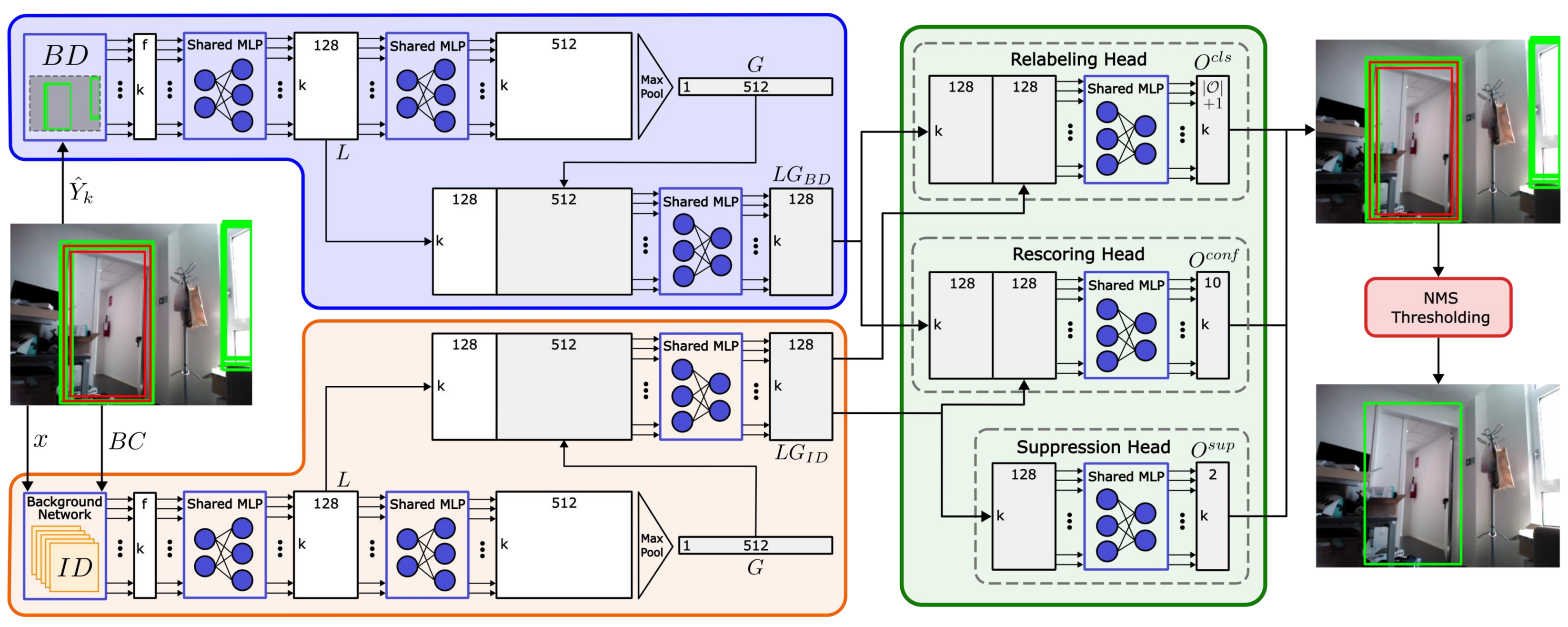




Architecture

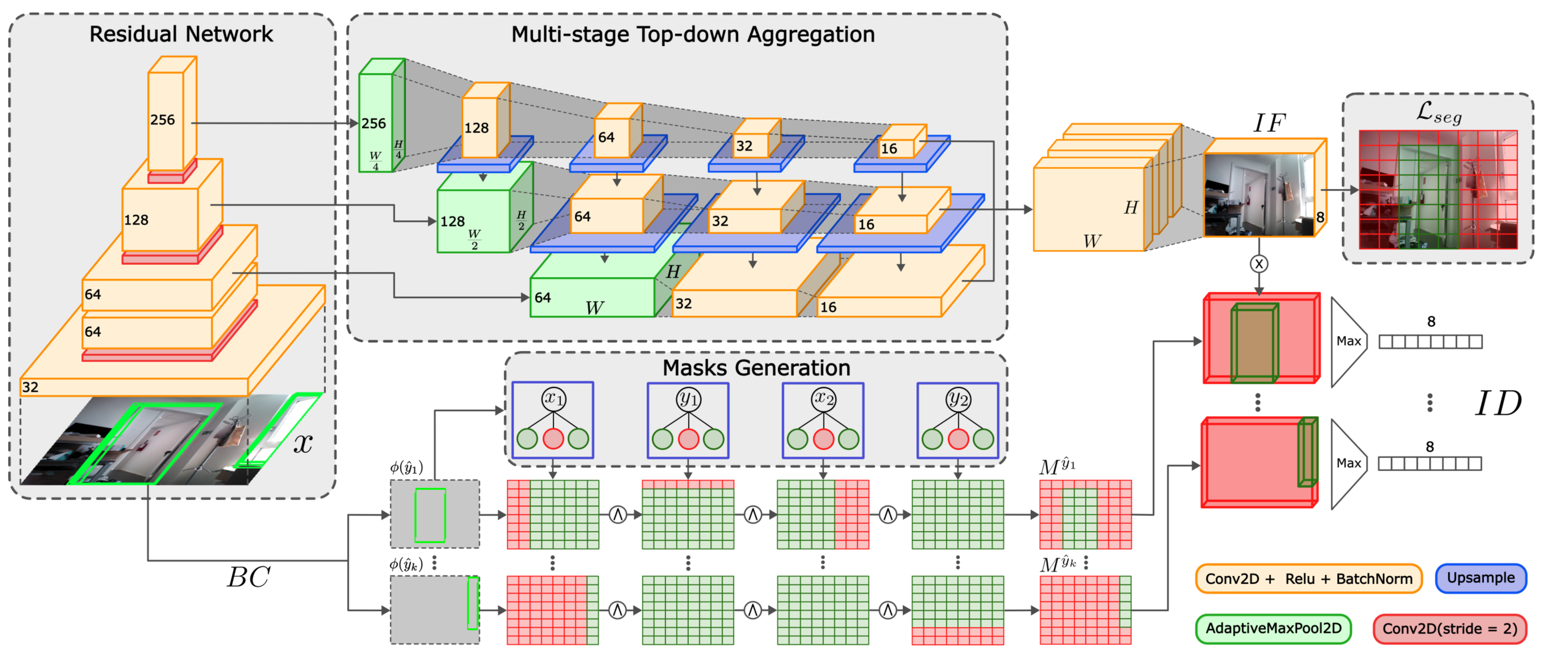
R2SNet Architecture

- Bounding boxes are expressed with two different descriptors:
  - Bounding-box Descriptors (BD): parameters of proposals received by the TaskNet
  - Image Descriptors (ID): visual features extracted by the Background Feature Network (BFNet)
- BD and ID are processed by two symmetric networks inspired by PointNet[4]
  - Local features (L) are extracted through shared MLPs and Global features (G) with a *max* operator
  - Local and global features are then concatenated and mixed with shared MLPs in an embedding *LG*
- The mixed features are fed into 3 heads to perform relabeling, rescoring, and suppression



BFNet Architecture

- Produces an image feature map *IF* with dimension $[W, H, 8]$
  - Extracts a multi-scale embeddings using a residual network
  - The last 3 levels are processed by 3 parallel convolutional networks and top-down aggregated
- Produces a binary masks M for each proposal
  - 4 MLPs with fixed weights and biases
  - Each MLP extracts a partial mask for each coordinate that are aggregated with an *and* operator
- Masks are multiplied with *IF* and then maxpooled obtaining visual features for each proposals



Evaluation

Datasets

- D_{DD2}: a real dataset (called DeepDoors2) with $\approx 3k$ examples
- D_G: photorealistic dataset obtained with Gibson simulator ($\approx 5k$ images)

→ Train the TaskNet (Faster R-CNN)

- D_{real}: a dataset collected with our robot in 4 environments ($\approx 2k$ images)[1]

→ Fine-tune R2SNet

Metrics

- Mean Average Prevision (mAP)
- The rates of true positive (TP), false positive (FP), and background false detections (BFD)[1]

Experiments

- We validate R2SNet in each environment of D_{real}:
  - Varying the number of training data (25%, 50%, 75%)
  - Varying the number of proposals (10, 30, 50, 100)
- Testing has been performed using the remaining 25%
- We perform an ablation study of the 3 heads

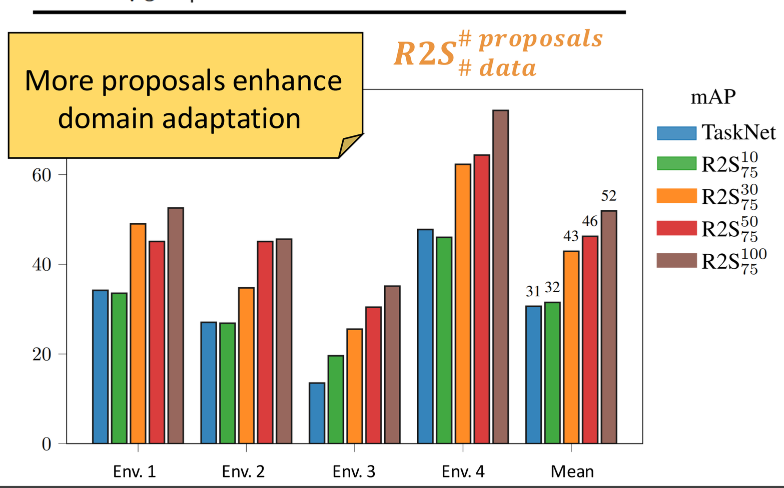
Performance increases even with a few data

Exp.	Mean mAP↑	TP↑	FP↓	BFD↓
TaskNet	30	36%	7%	20%
$R2S^{30}_{25}$	37	44%	6%	11%
$R2S^{30}_{50}$	39	45%	6%	9%
$R2S^{30}_{75}$	43	46%	5%	7%

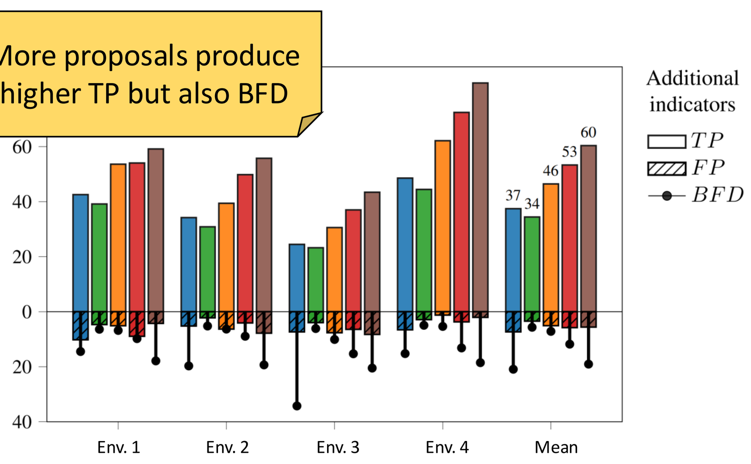
All heads contribute to domain adaptation

Ablation study Rel.	Res.	Sup.	mAP↑	TP↑	FP↓	BFD↓
			34	44%	10%	35%
✓			44	48%	4%	6%
	✓		41	54%	15%	34%
		✓	37	43%	9%	14%
✓	✓		52	61%	6%	20%
✓		✓	44	47%	4%	5%
	✓	✓	41	53%	15%	31%
✓	✓	✓	52	60%	6%	19%

More proposals enhance domain adaptation



More proposals produce higher TP but also BFD



References

[1] Antonazzi, Michele, et al. "Development and Adaptation of Robotic Vision in the Real-World: the Challenge of Door Detection," 2024.
[2] Hu, Guoqiang, et al., "Cloud robotics: architecture, challenges and applications." in IEEE Network 26.3. 2012
[3] Oza, Poojan, et al. "Unsupervised domain adaptation of object detectors: A survey," In IEEE Trans. Pattern Anal. 2023.
[4] Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation," In Proc. *IEEE CVPR*. 2017.