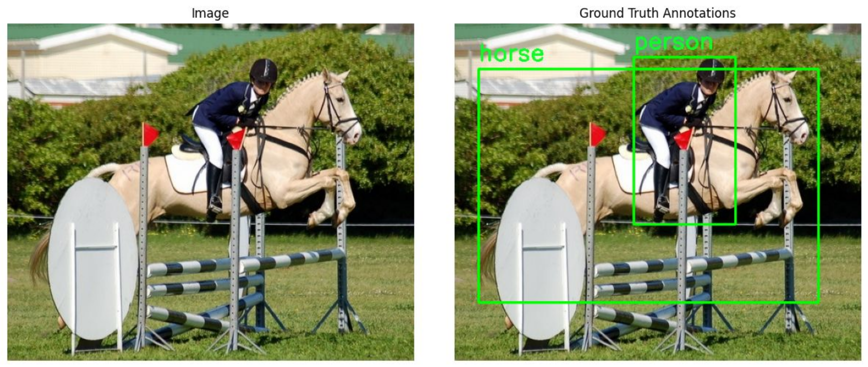
Integrating Saliency Ranking and Reinforcement Learning for Enhanced Object Detection

Matthias Bartolo
Dylan Seychell
Josef Bajada

L-Università ta' Malta
Faculty of Information & Communication Technology
Department of Artificial Intelligence

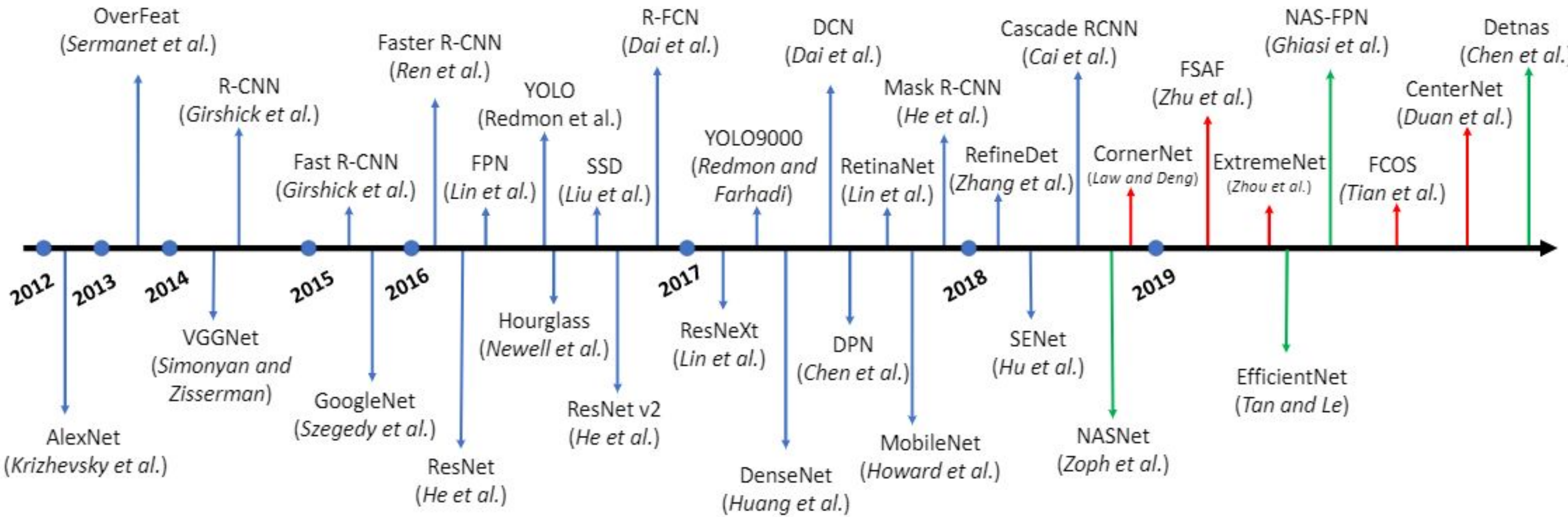
Object Detection

Find objects in an image that belong to a predefined set of classes (e.g. person, motorbike, cat, etc.)



Images from Pascal VOC

An Overview of Deep Learning Techniques for OD



(Source: [1])

Reinforcement Learning Object Detectors

1. Active object localization with deep reinforcement learning **(2015)**
2. Hierarchical object detection with deep reinforcement learning **(2016)**
3. Reinforcement learning for visual object detection **(2016)**
4. Tree-structured reinforcement learning for sequential object localization **(2017)**
5. Deep reinforcement learning of region proposal networks for object detection **(2018)**
6. Object localization using deep reinforcement learning mohammad otoofi **(2018)**
7. Multitask learning for object localization with deep reinforcement learning **(2019)**
8. BAR - A reinforcement learning agent for bounding-box automated refinement **(2020)**
9. Efficient object detection in large images using deep reinforcement learning **(2020)**
10. Étude de la localisation active d’objets par apprentissage par renforcement profond **(2020)**
11. A reinforcement learning embedded object detection framework with region selection network **(2021)**
12. Object detection with deep reinforcement learning **(2022)**

Problems that need to be solved

1. Very large number of bounding boxes
2. Definition of a classification rule
3. Redundant bounding boxes at output
4. Foreground-Background imbalance

Motivation

Object Detection Applications:

1. Visual Scene Content Interpretation
2. Environmental Monitoring and Wildlife Research
3. Obstacle Avoidance and Navigation
4. Medical Imaging and Diagnosis
5. Interactive Experiences and Entertainment
6. Automation

Need for Transparency (xAI)

Human Perception

How can object detection systems emulate such proficiency?

Not All Detectors Are Transparent

A tool to address this.

SaRLVision

"A reinforcement learning object detector leveraging saliency ranking, offering a self-explainable system with a fully observable action log."

Contributions

SaRLVision

✅ Ability to visualise the object detection training process (xAI)

✅ Smaller optimised deployable models (various DQN Architectures)

✅ System built via the renowned and used gymnasium API

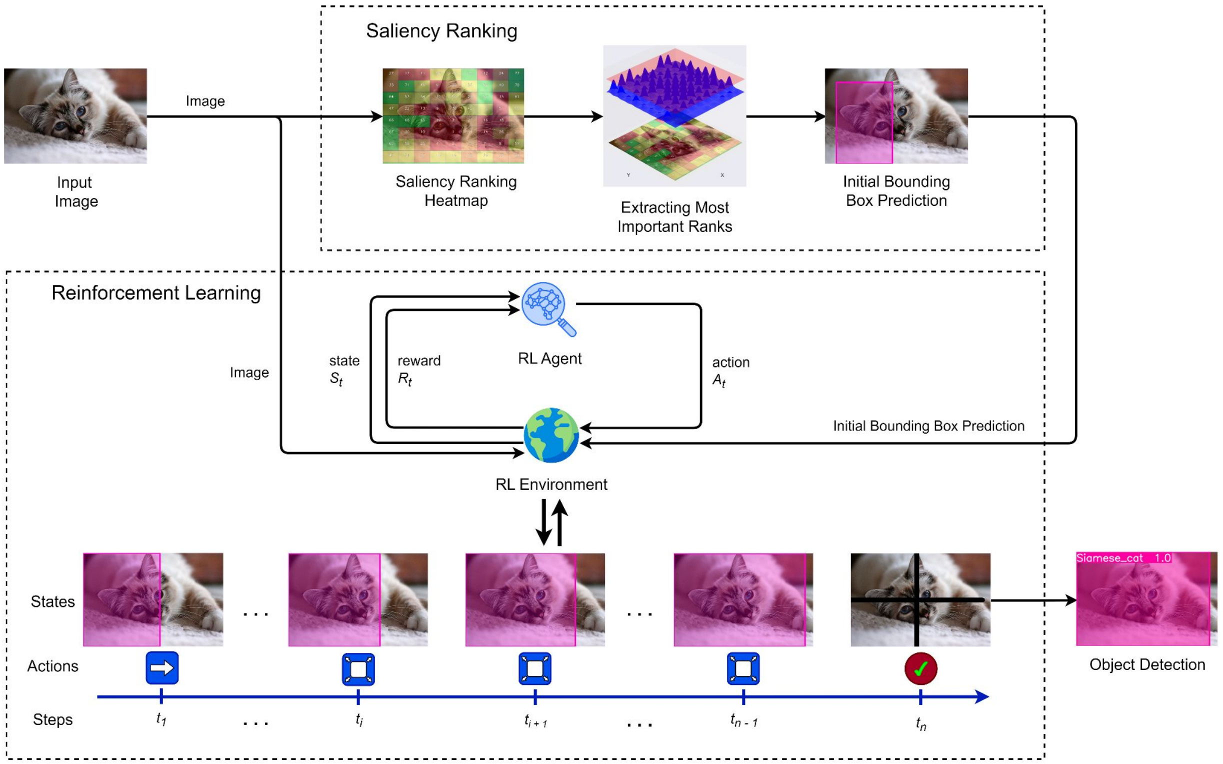
✅ Ability to classify the detected objects

✅ A more optimised pipeline

✅ Faster agent training

✅ Better accuracy and mAP

Proposed Architecture: Let Us Break Things Down . . .



1) Saliency Map

Laurent Itti's Saliency Model

User Exercise 1: Can you identify the most salient parts of the image?

2) Saliency Ranking

$$S_n = H_n + CB_n + DS_n$$

User Exercise 2: Can you identify the most salient regions in the image, from the saliency map?

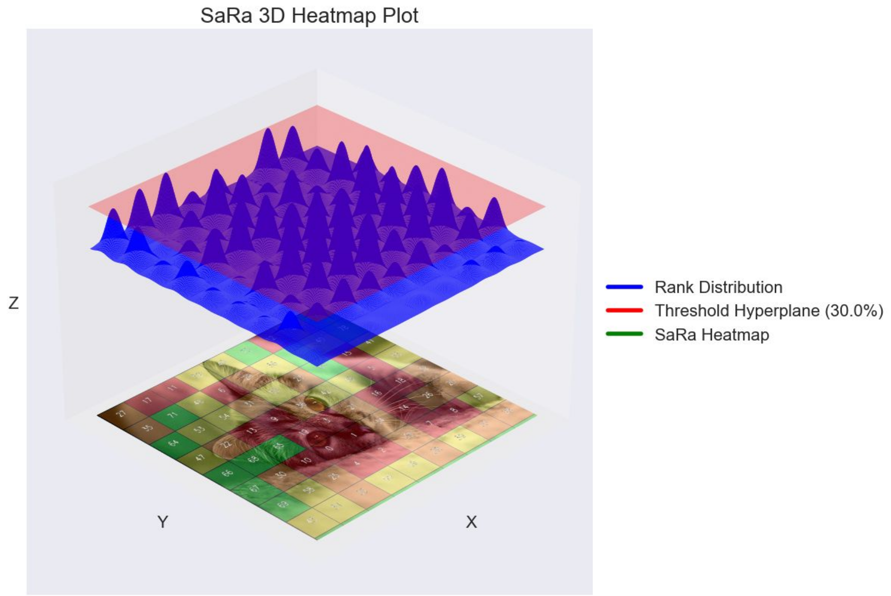
3) Extracting Most Important Ranks

Let $\boldsymbol{R} = [R_1, R_2, R_3, \ldots, R_n]$ be the saliency rank vectors sorted in order.

Let $\boldsymbol{P}$ be the percentage hyperplane representing the ranks to retain, based on the specified percentage.

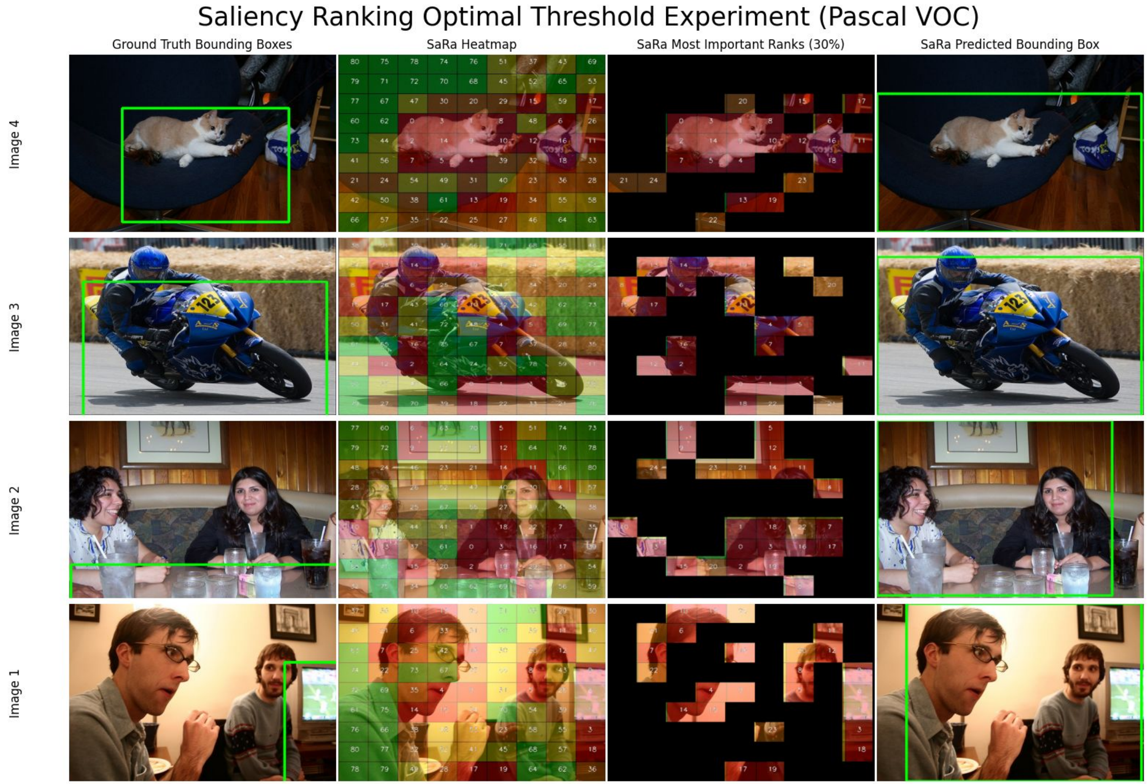
Let $\boldsymbol{R}_{\boldsymbol{bbox}} = [R_1, R_2, R_3, \ldots, R_{\boldsymbol{P}}]$ be the ranks to retain.

A Mathematical Representation

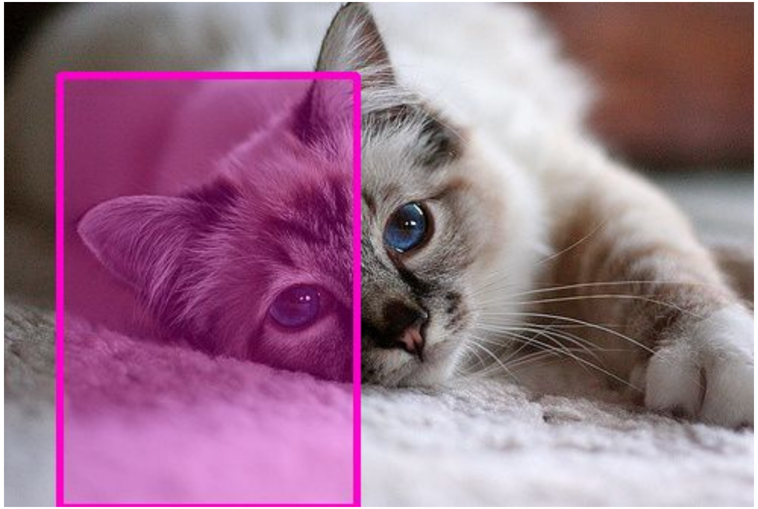


A Geometric Representation

4) Visualising the Process



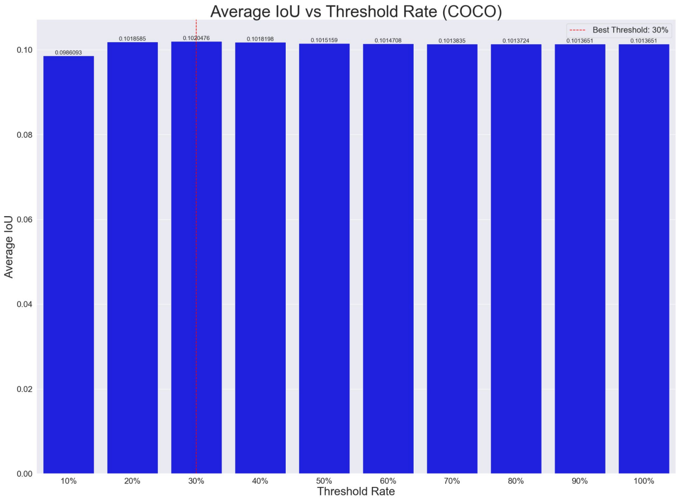
5) Initial Bounding Box Prediction

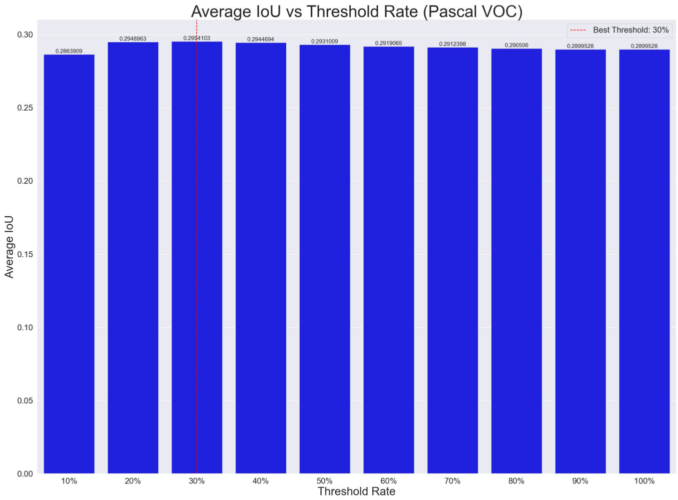


Bounding box formed from the percentage of most important ranks (30%)

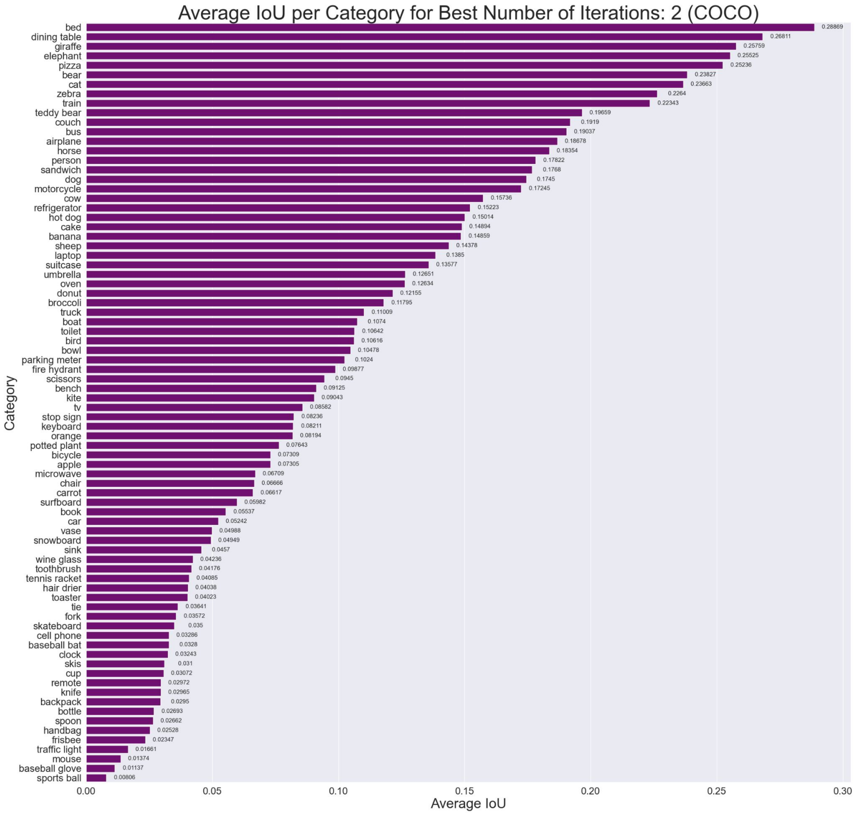
6) Why 30% as the Optimal Threshold (Experiment 1)

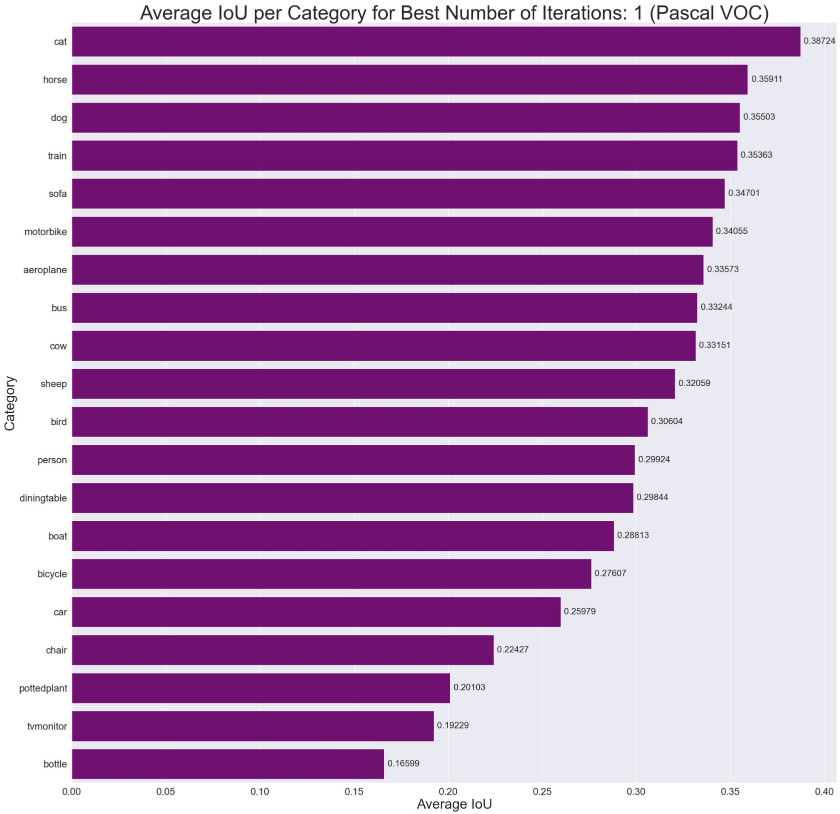
Dataset	COCO 2017 Train	Pascal VOC 2007+2012 Train
Number of Images	78018	12880
Optimal Threshold	0.3 (30%)	0.3 (30%)
Optimal Number of Iterations	2	1
Average IoU	0.1	0.3
Number of Categories	80	20





7) Experiment 1 Continued

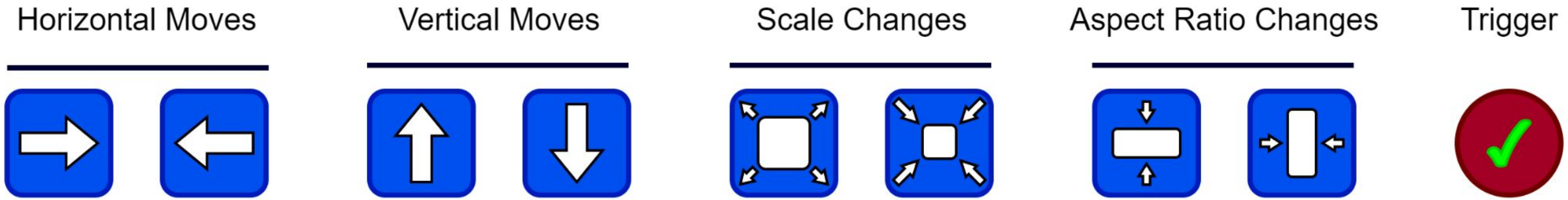




8) Reinforcement Learning

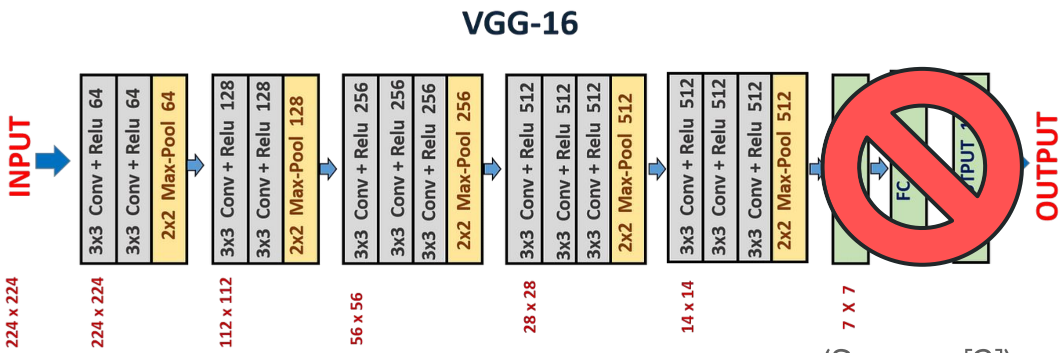
Architecture	VGG16 (Ours)	MobileNet (Ours)	ResNet50 (Ours)	CNN ([63])	VGG16 ([78])
State Size	602	1370	2138	4186	25178

State Space: *(o, h)* where *o* is denotes a feature vector corresponding to the observed region (current bounding box region), and *h* encapsulates the history of actions undertaken.



Action Space: *A* comprises of eight transformations applicable to the bounding box, alongside one action designed to terminate the search process. These transformations, i are categorised into four subsets: horizontal and vertical box movement, scale adjustment, and aspect ratio modification.

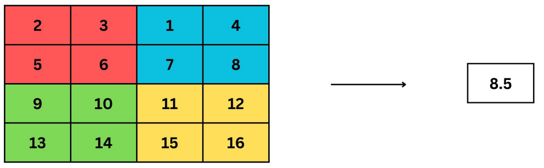
9) Reinforcement Learning - Feature Learning



(Source: [2])

+

Global Average Pooling



No Conecpt of Windows. Take Average of all Values in the Matrix.

(Source: [3])

Illustrating the procedure for extracting image features utilised in **state creation**. The proposed method involves employing a pre-trained feature learning network and subsequently eliminating the Dense layers at the end of the network. A **Global Average Pooling Layer**, which is non-trainable and fast, is incorporated, instead of the Dense Layers, in order to extract a **feature vector**.

10) Reinforcement Learning

$$\alpha_w = \alpha * (x_2 - x_1) \quad \alpha_h = \alpha * (y_2 - y_1)$$

Aspect ratio: Each transformation action induces a discrete change to the box's size relative to its current dimensions, influenced by the aspect ratio. The parameter α is established at **0.2**, as a smaller value resulted in sluggish transformations.

$$R_a(s_t, s_{t+1}) = \text{sign}\left(\text{IoU}(b_{t+1}, g) - \text{IoU}(b_t, g)\right)$$

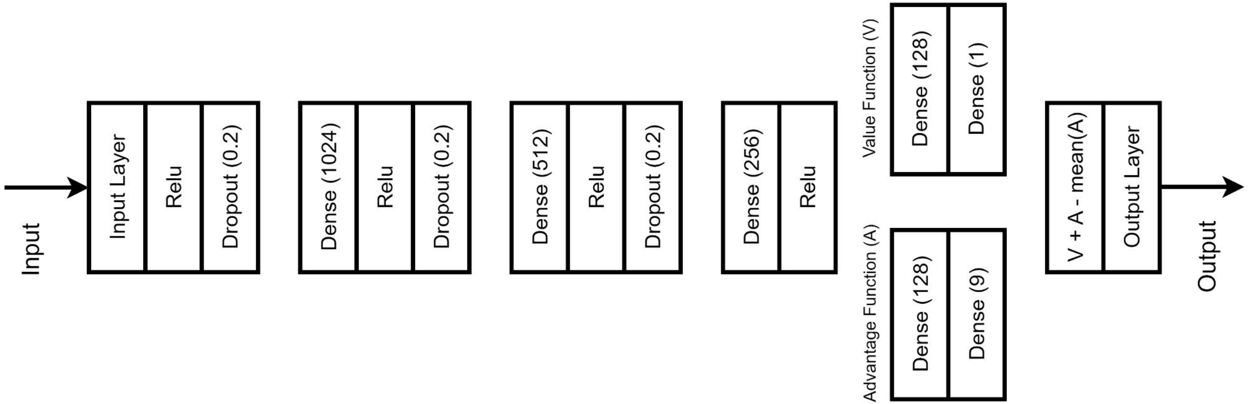
$$R_\omega(s_t, s_{t+1}) = \begin{cases} +\eta * 2 * \text{IoU}(b, g) & \text{if IoU}(b, g) \geq \tau \\ -\eta & \text{otherwise} \end{cases}$$

Reward Structure: The reward (R_a) is determined by the sign of the change in IoU, encouraging positive rewards for improvements and negative rewards otherwise *{ - 1, + 1}*. The trigger action employs a distinct reward scheme (R_ω) due to its role in transitioning to a terminal state without altering the box, resulting in a zero differential IoU (η is set to 3.0).

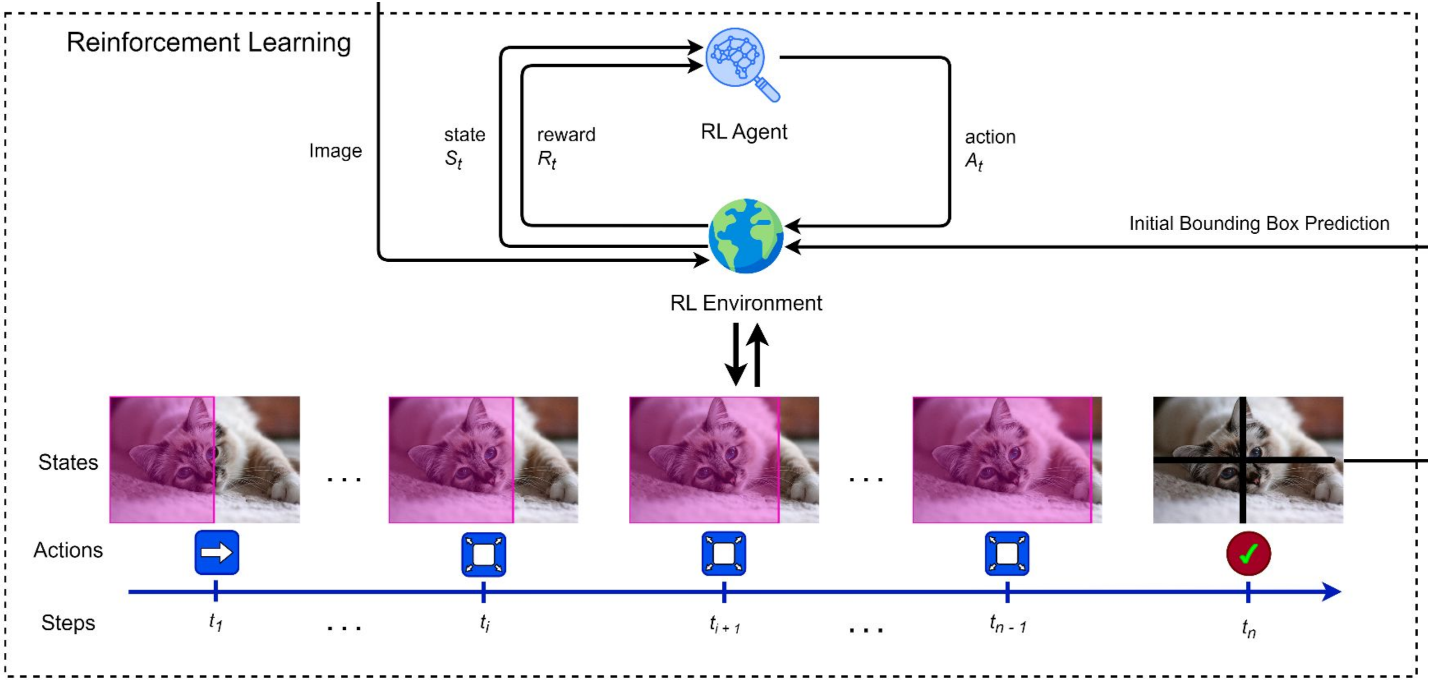
11) Reinforcement Learning - Network Architectures

Deep Q-Network Architecture

Dueling Deep Q-Network Architecture



12) Reinforcement Learning - Interaction Loop

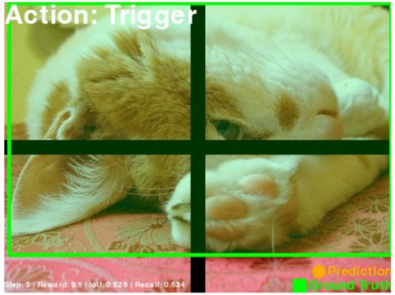


13) Object Classification

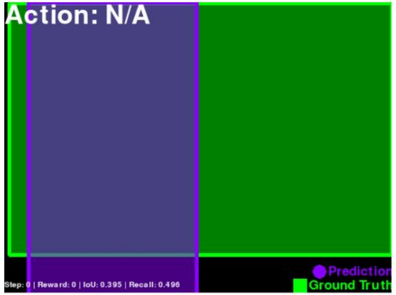
14) Self-Explainability - Render Modes

Training Mode (human)

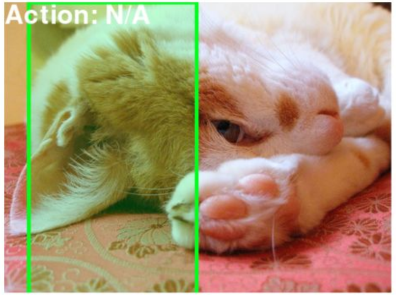
Training Mode (trigger_image)

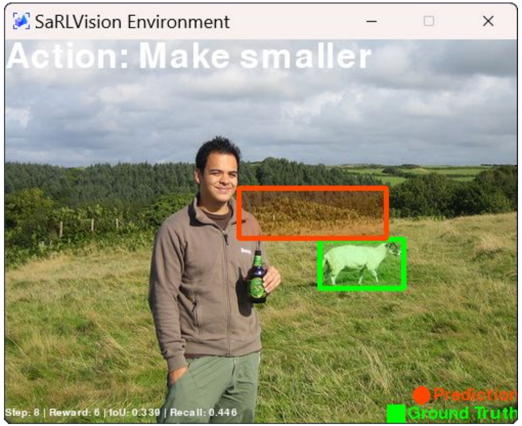


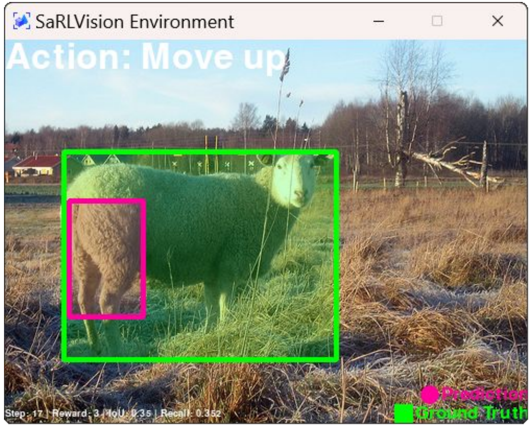
Training Mode (bbox)

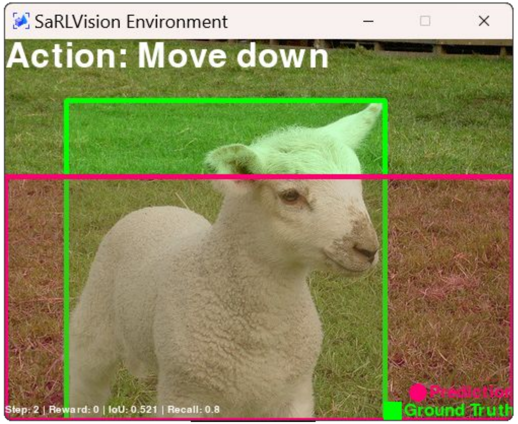


Testing Mode (human)









15) Self-Explainability - Fully Observable Action Log

16) Results and Evaluation (Experiment 2)

Index	Configuration					$mAP^{IoU=.50}$
	Agent	Exploration	Feature Network	SaRa Trained	SaRa Inference	
Config 1)	DQN	Random	VGG16	No	No	**46.86**
Config 2)	DQN	Random	VGG16	No	Yes	39.23
Config 3)	DQN	Random	VGG16	Yes	Yes	**42.12**
Config 4)	DQN	Random	VGG16	Yes	No	35.73
Config 5)	DQN	Guided	VGG16	No	No	40.10
Config 6)	DQN	Guided	VGG16	No	Yes	41.54
Config 7)	DQN	Guided	VGG16	Yes	Yes	32.67
Config 8)	DQN	Guided	VGG16	Yes	No	33.70

Exploring the effects of changing exploration modes and utilisation of saliency ranking.

17) Results and Evaluation (Experiment 3)

Index	Configuration					$mAP^{IoU=.50}$
	Agent	Exploration	Feature Network	SaRa Trained	SaRa Inference	
Config 1)	DQN	Random	VGG16	No	No	**46.86**
Config 3)	DQN	Random	VGG16	Yes	Yes	42.12
Config 9)	DQN	Random	MobileNet	No	No	41.42
Config 10)	DQN	Random	MobileNet	Yes	Yes	44.71
Config 11)	DQN	Random	ResNet50	No	No	43.16
Config 12)	DQN	Random	ResNet50	Yes	Yes	35.22

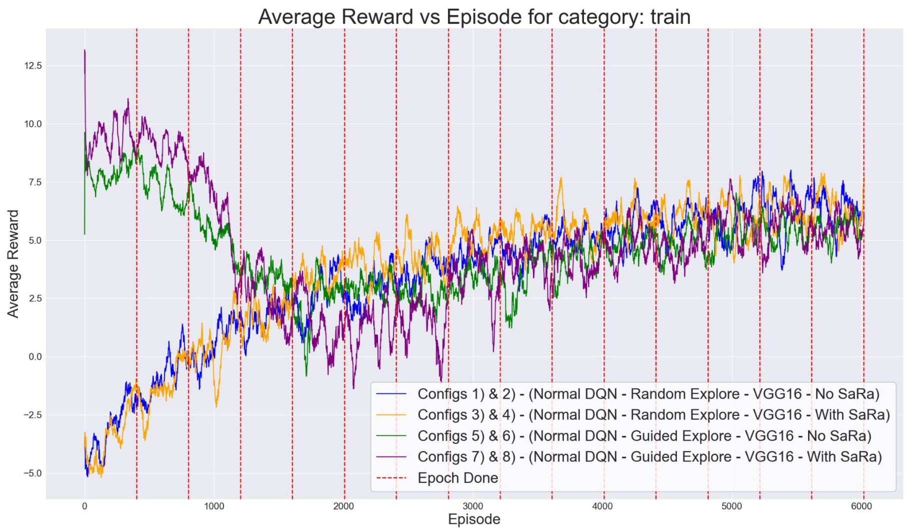
Exploring the effects of varying different feature learning architectures.

18) Results and Evaluation (Experiment 4)

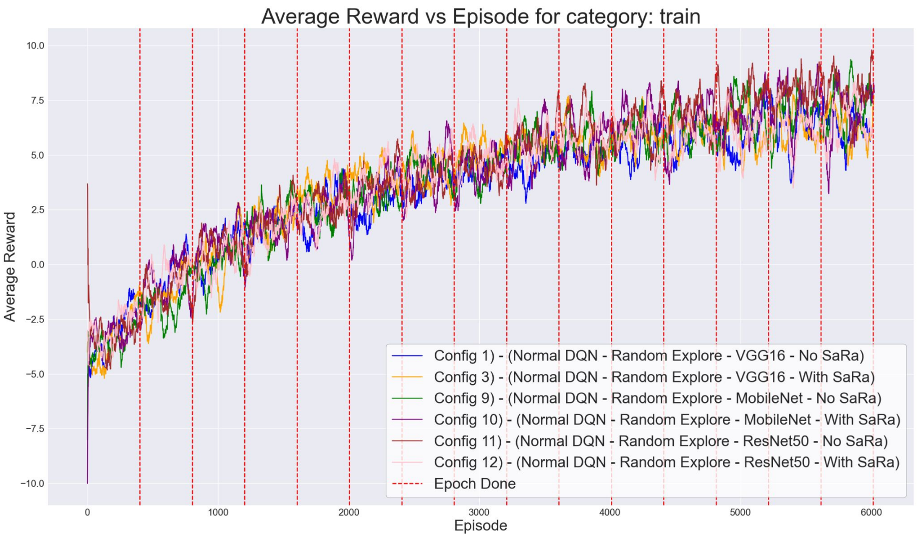
Index	Configuration					$mAP^{IoU=.50}$
	Agent	Exploration	Feature Network	SaRa Trained	SaRa Inference	
Config 1)	DQN	Random	VGG16	No	No	46.86
Config 13)	DDQN	Random	VGG16	No	No	39.94
Config 14)	Dueling DQN	Random	VGG16	No	No	48.96
Config 15)	D3QN	Random	VGG16	No	No	**51.37**

Exploring the effects of varying different DQN architectures.

19) Training Curves



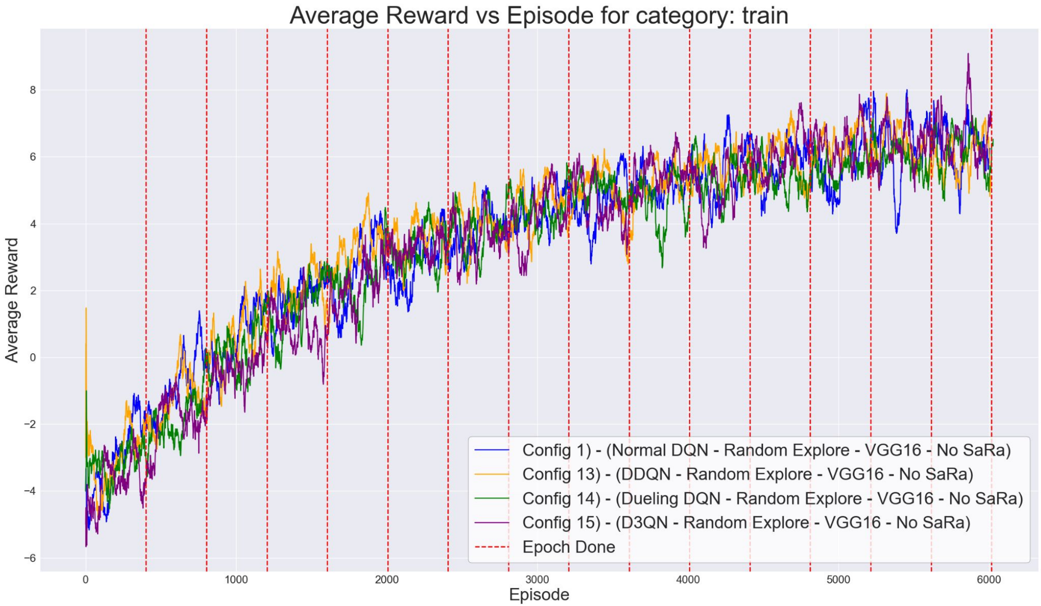
Experiment 2



Experiment 3

Average Reward vs Episode for category: train

20) Training Curves Continued

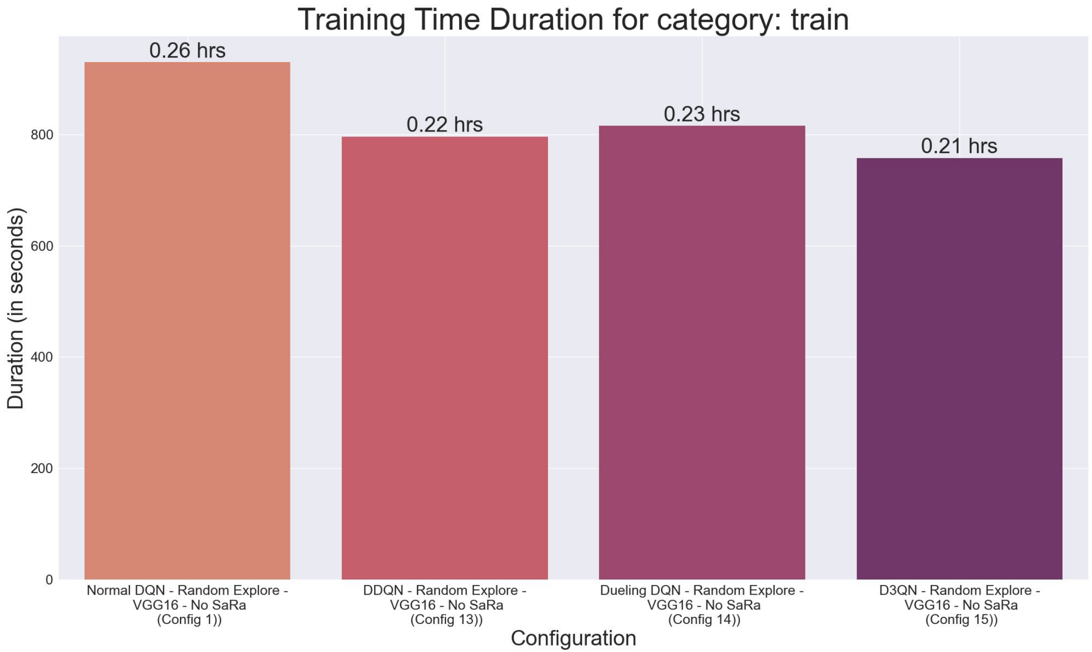


Experiment 4

Average Reward vs Episode for category: train

21) Training Time

Comparison wise, in the literature for single object detection, it took approximately **3 hours** to train one category [4]. This was confirmed when testing locally.



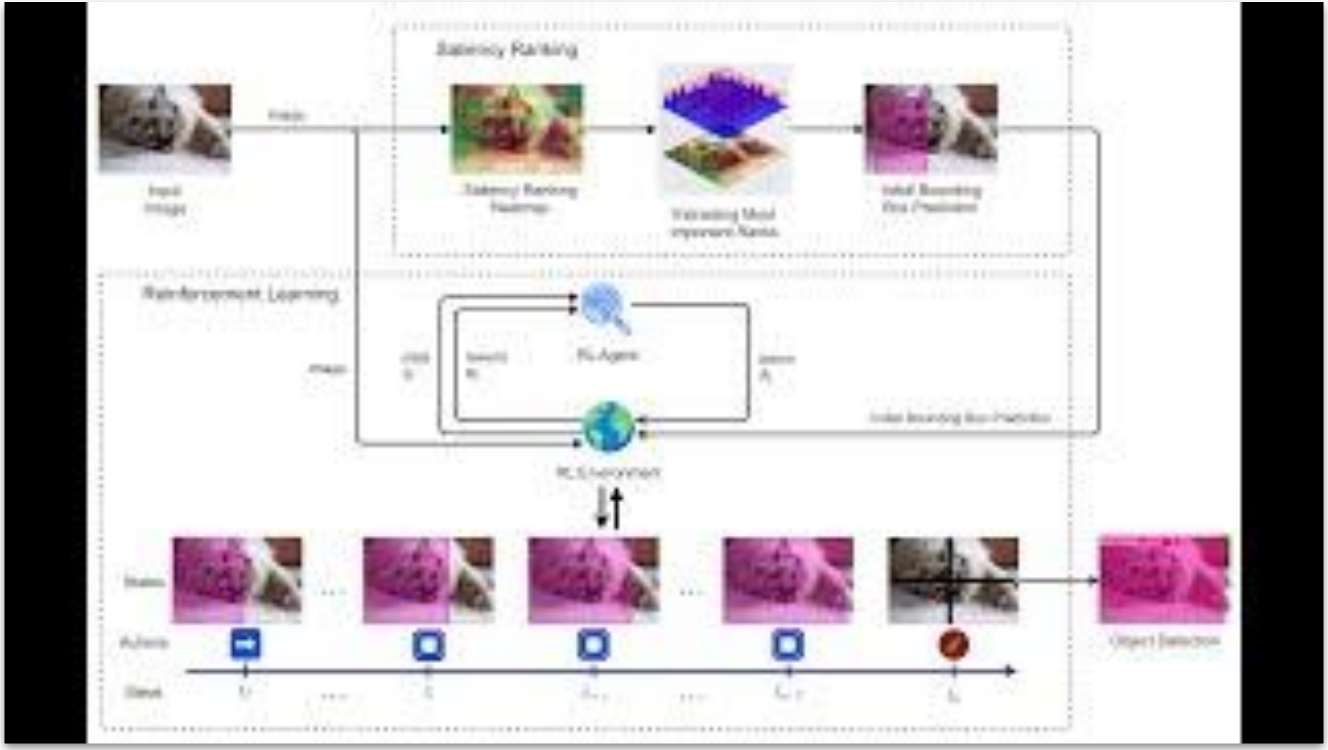
Training Time for category: train (Experiment 4)

22) A Comparative Analysis

Index	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
Caicedo et al (TR) [63]	57.9	56.7	38.4	33.0	17.5	51.1	52.7	53.0	17.8	39.1	47.1	52.2	58.0	57.0	45.2	19.3	42.2	35.5	54.8	49.0	43.9
Caicedo et al (AAR) [63]	55.5	61.9	38.4	36.5	21.4	56.5	58.8	55.9	21.4	40.4	46.3	54.2	56.9	55.9	45.7	21.1	47.1	41.5	54.7	51.4	46.1
S. R. Ramoul [78]	51.4	30.0	20.4	7.0	1.8	43.0	15.0	52.7	3.2	24.1	37.0	43.7	50.0	45.5	15.6	4.4	12.5	39.9	39.0	9.4	27.3
Config 1) (Ours)	76.4	25.7	64.3	18.3	4.1	**74.6**	**67.9**	73.1	4.1	64.3	34.1	29.8	**82.7**	73.9	38.8	6.6	68.9	21.3	52.9	55.7	46.9
Config 2) (Ours)	60.2	14.0	13.5	60.0	4.3	70.1	60.1	68.7	15.3	12.7	65.1	18.4	64.3	21.0	67.8	**65.2**	69.2	20.9	12.5	1.2	39.2
Config 3) (Ours)	70.8	30.3	**72.1**	17.7	19.9	49.3	63.9	66.9	6.2	21.6	33.4	**66.0**	70.1	72.6	22.4	7.2	44.4	21.5	72.1	14.1	42.1
Config 4) (Ours)	74.0	65.2	22.6	9.2	6.7	35.3	30.0	34.4	1.2	22.6	67.9	25.2	69.2	32.5	20.8	58.9	**70.0**	23.1	42.9	2.7	35.7
Config 5) (Ours)	65.5	17.8	67.8	11.3	2.2	39.1	64.5	64.8	**65.6**	26.5	66.9	62.0	74.8	32.0	22.5	5.0	16.4	21.6	71.9	3.8	40.1
Config 6) (Ours)	27.6	26.2	20.0	12.8	58.6	72.4	62.1	59.1	59.1	14.8	62.5	60.7	60.7	18.6	**68.8**	1.7	13.8	59.1	69.7	2.6	41.5
Config 7) (Ours)	74.3	18.8	21.3	17.7	1.9	25.7	66.4	24.3	4.3	64.8	71.9	30.1	30.6	62.3	22.1	1.8	12.5	25.3	73.7	3.9	32.7
Config 8) (Ours)	**76.8**	16.2	33.2	18.0	25.2	31.8	23.2	64.7	4.9	22.2	34.5	39.5	74.1	23.3	61.5	6.8	20.2	25.1	69.8	2.9	33.7
Config 9) (Ours)	56.2	23.6	45.6	26.5	3.5	68.6	67.0	33.7	4.3	17.5	68.2	45.1	76.2	45.2	20.4	4.8	66.3	63.9	**76.5**	15.4	41.4
Config 10) (Ours)	72.6	62.6	22.0	18.4	**65.1**	33.6	65.2	44.9	15.4	**65.7**	28.2	63.4	70.1	30.5	11.2	10.0	64.9	18.2	68.0	**64.3**	44.7
Config 11) (Ours)	74.4	27.6	35.6	61.5	14.7	34.2	64.2	43.9	4.7	62.8	**73.4**	24.8	81.1	71.8	16.2	5.3	22.5	65.4	76.3	2.9	43.2
Config 12) (Ours)	70.6	19.2	60.2	30.1	2.4	37.8	23.6	**75.2**	3.2	61.2	63.3	37.5	21.6	22.2	9.1	4.0	29.6	63.8	67.8	1.9	35.2
Config 13) (Ours)	64.0	26.5	20.9	20.7	7.4	40.8	48.7	71.5	4.2	36.3	25.7	**66.0**	80.5	38.1	25.1	15.0	63.8	65.2	73.8	4.4	39.9
Config 14) (Ours)	76.4	62.0	46.9	62.2	3.1	69.6	36.6	66.8	4.0	29.6	64.1	23.4	78.1	**75.3**	31.2	62.6	69.6	33.9	68.7	14.9	49.0
Config 15) (Ours)	76.0	**74.2**	67.1	**64.7**	4.7	72.7	64.5	68.7	3.6	33.7	23.4	34.0	77.2	71.5	64.9	3.2	23.1	**67.5**	73.6	59.3	**51.4**
Best Category APs (Ours)	**76.8**	**74.2**	**72.1**	**64.7**	**65.1**	**74.6**	**67.9**	**75.2**	**65.6**	**65.7**	**73.4**	**66.0**	**82.7**	**75.3**	**68.8**	**65.2**	**70.0**	**67.5**	**76.5**	**64.3**	**70.6**

Average Precision (AP) per Category and Mean Average Precision (mAP) in the Pascal VOC 2007 Test Set.

Need a Demo?



All you need is already made available online

Code is **open-source** and available on **GitHub**:

https://github.com/mbar0075/SaRLVision

✅ SaRLVision python library package

✅ Demo notebooks

✅ Experiment notebooks

✅ Requirements and environment files

✅ Necessary documentation and bibliography

Conclusion

Conclusions:

1. Employing Saliency Ranking results in poor results
2. Random Exploration generated better results
3. Smaller state sizes achieved better results
4. Dueling DQN and D3QN generated the best results
5. Better Accuracy, Speed, and Smaller models
6. A Self-Explainable Approach

Future Work

Future Works:

1. Adoption of a Continuous Action Space
2. Training state-of-the-art Rainbow DQN
3. Re-imagining the MDP problem
4. A tool to interpret between input features and action taken

References

[1] Ram Sagar, "How The Deep Learning Approach For Object Detection Evolved Over The Years," Analytics India Magazine, August 26, 2019. [Online]. Available: https://analyticsindiamag.com/how-the-deep-learning-approach-for-object-detection-evolved-over-the-years/. [Accessed: March 23, 2024].

[2] V. Khandelwal, "The Architecture and Implementation of VGG-16," Towards AI, Aug. 17, 2020. [Online]. Available: https://pub.towardsai.net/the-architecture-and-implementation-of-vgg-16-b050e5a5920b. [Accessed: March 23, 2024].

[3] M. Arham, "Diving into the Pool: Unraveling the Magic of CNN Pooling Layers," KDNuggets, Sep. 28, 2023. [Online]. Available: https://www.kdnuggets.com/diving-into-the-pool-unraveling-the-magic-of-cnn-pooling-layers. [Accessed: March 23, 2024].

[4] S. R. Ramoul, Rapport bibliographique: Étude de la localisation active d’objets par apprentissage par renforcement profond, Sorbonne Université, Encadré par Pr. Isabelle Bloch, Nov. 2020. [Online]. Available: https://github.com/rayanramoul/Active-Object-Localization-Deep-Reinforcement-Learning. [Accessed: March 23, 2024].

Thank you!

matthias.bartolo.21@um.edu.mt
@MatthiasBartolo