Results and Discussions

Below are the various observations calculated by the experiments done by me on PASCAL [13] dataset. mAP is calculated at 50

1) Aeroplane Class:

   i) Training Image : 25 Testing Image : 175

	50 IoU 50Epoch	50 IoU 80Epoch	70 IoU 50Epoch	70 IoU 80Epoch	80 IoU 50Epoch	80 IoU 80Epoch	85 IoU 50Epoch	85 IoU 80Epoch
IoU Average	0.4572	0.4685	0.4708	**0.4821**	0.4625	0.4780	0.4768	0.4448
IoU Inc.	188	155	128	**168**	139	155	145	120
IoU Dec	15	48	75	**35**	64	48	58	83
mAP bef	15.29	16.32	26.33	**29.20**	27.75	25.96	29.47	24.28
mAP aft	20.48	22.55	32.48	**37.98**	33.45	32.37	35.50	30.78

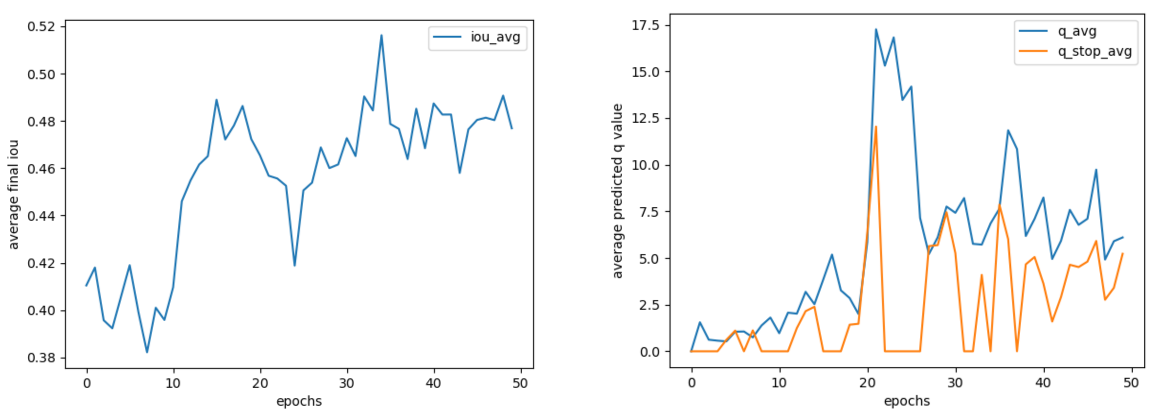


Fig. IoU: 50, Epoch: 50

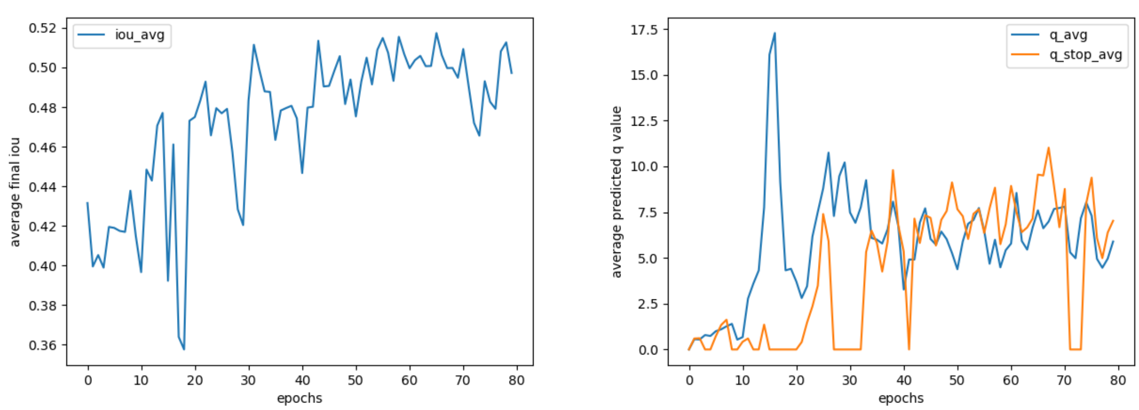


Fig. IoU: 50, Epoch: 80

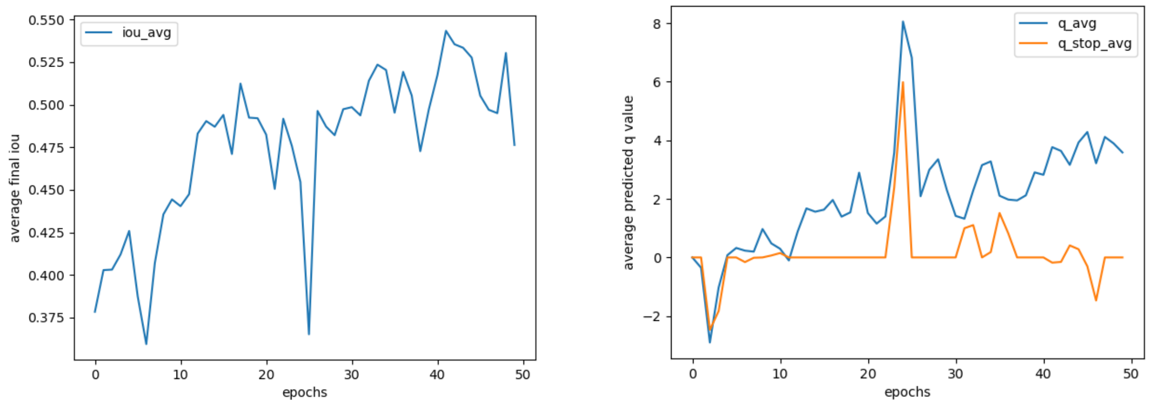


Fig. IoU: 70, Epoch: 50

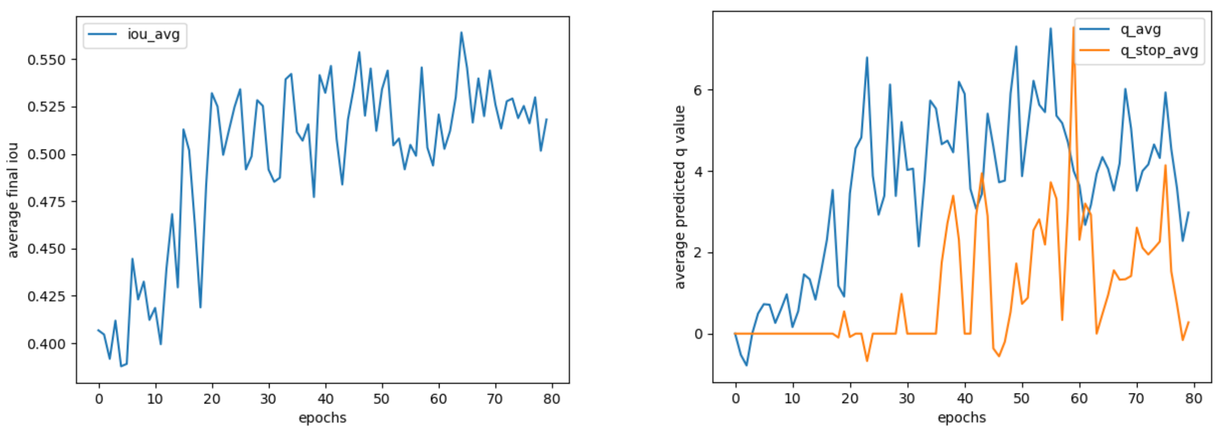


Fig. IoU: 70, Epoch: 80

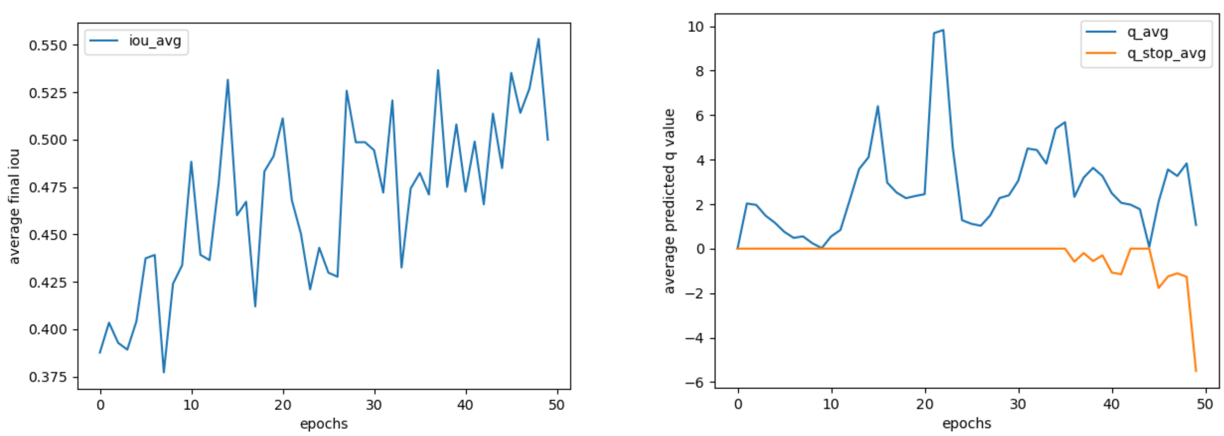


Fig. IoU: 80, Epoch: 50

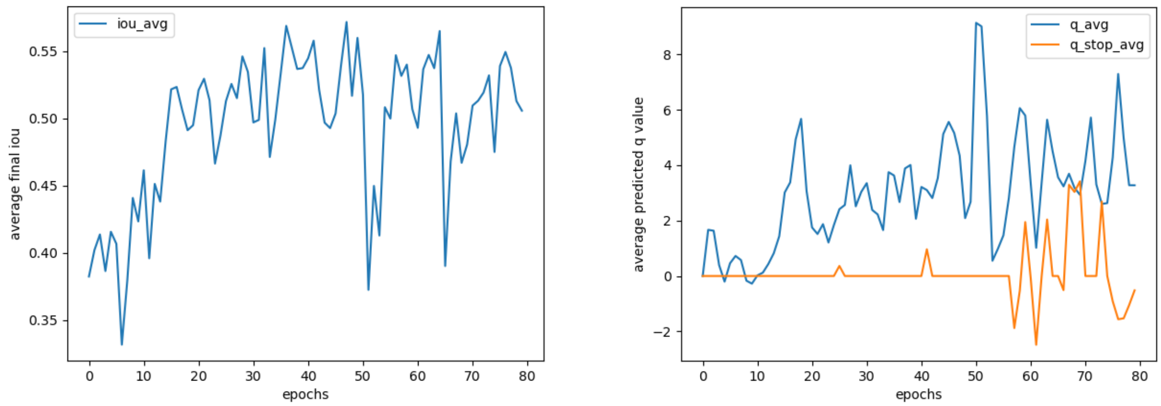


Fig: IoU: 80, Epoch: 80

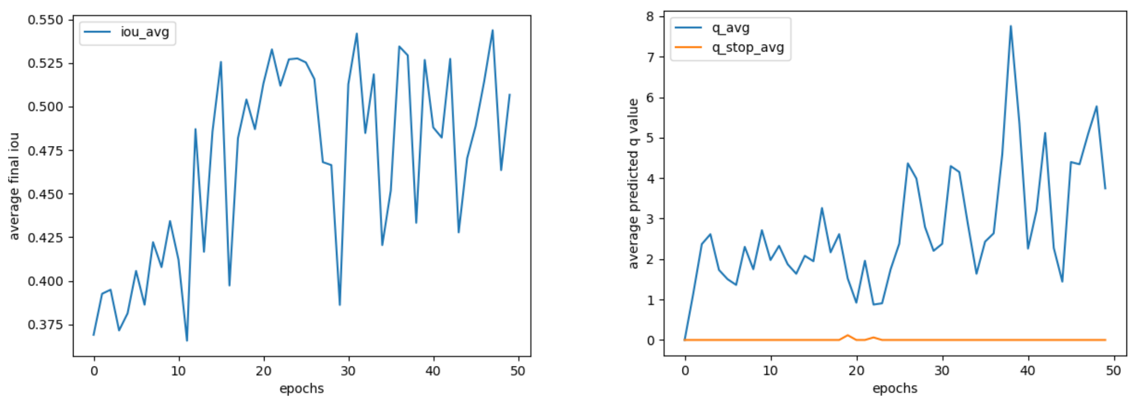


Fig. IoU: 85, Epoch: 50

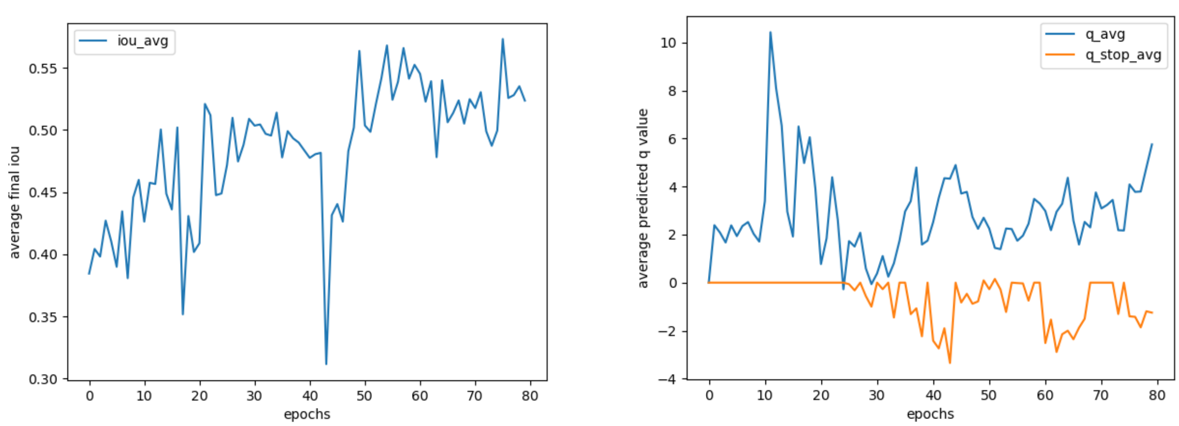
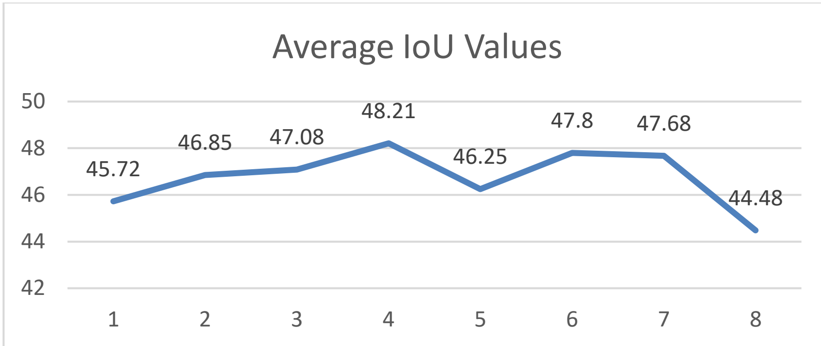


Fig. IoU: 85, Epoch: 80



ii) Training Image : 50 Testing Image : 150

	50 IoU 50Epoch	50 IoU 80Epoch	70 IoU 50Epoch	70 IoU 80Epoch	80 IoU 50Epoch	80 IoU 80Epoch	85 IoU 50Epoch	85 IoU 80Epoch
IoU Average	0.4433	0.4627	0.3957	0.4641	0.4771	**0.4869**	0.4642	0.4518
IoU Inc.	110	118	105	132	140	**132**	133	134
IoU Dec	65	57	70	43	35	**43**	42	41
mAP bef	13.52	16.45	12.83	22.58	25.82	**36.53**	23.47	25.66
mAP aft	18.24	21.89	18.01	25.12	35.98	**36.21**	25.50	35.50

iii) Training Image : 175 Testing Image : 25

	50 IoU 50Epoch	50 IoU 80Epoch	70 IoU 50Epoch	70 IoU 80Epoch	80 IoU 50Epoch	80 IoU 80Epoch	85 IoU 50Epoch	85 IoU 80Epoch
IoU Average before	0.4947	0.4617	0.5194	0.5274	0.5213	0.5282	0.5043	**0.5397**
IoU After update	0.5328	0.4612	0.5209	0.5587	0.5866	0.5752	0.5424	**0.5635**
IoU Inc.	25	23	22	23	24	23	23	**22**
IoU Dec	1	3	4	3	2	3	3	**4**
mAP bef	42.10	35.78	43.29	44.20	44.85	45.65	44.20	**48.87**
mAP aft	51.3	37.93	50.11	51.15	51.87	52.52	49.62	**53.69**

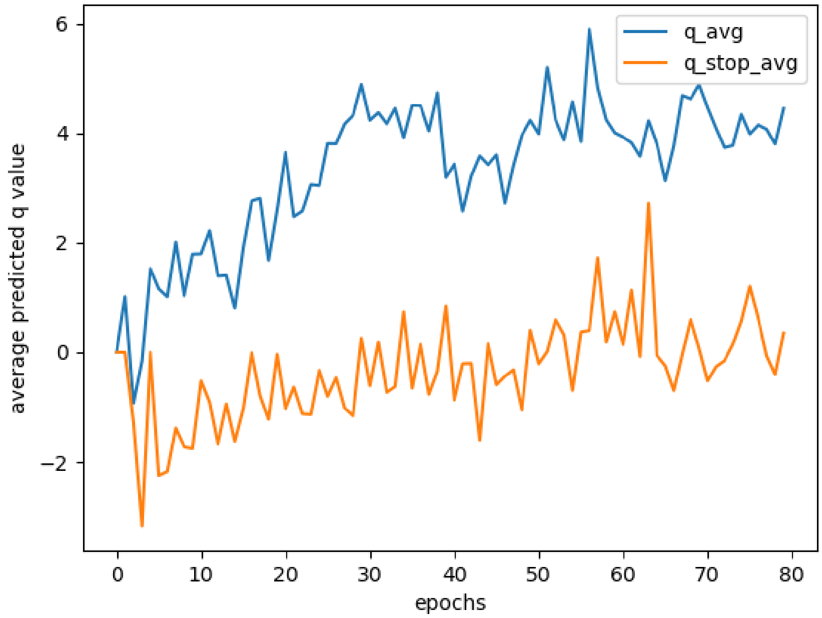


Fig. Q value plot of above experiment for IoU threshold set at 85 and epochs equal to 80

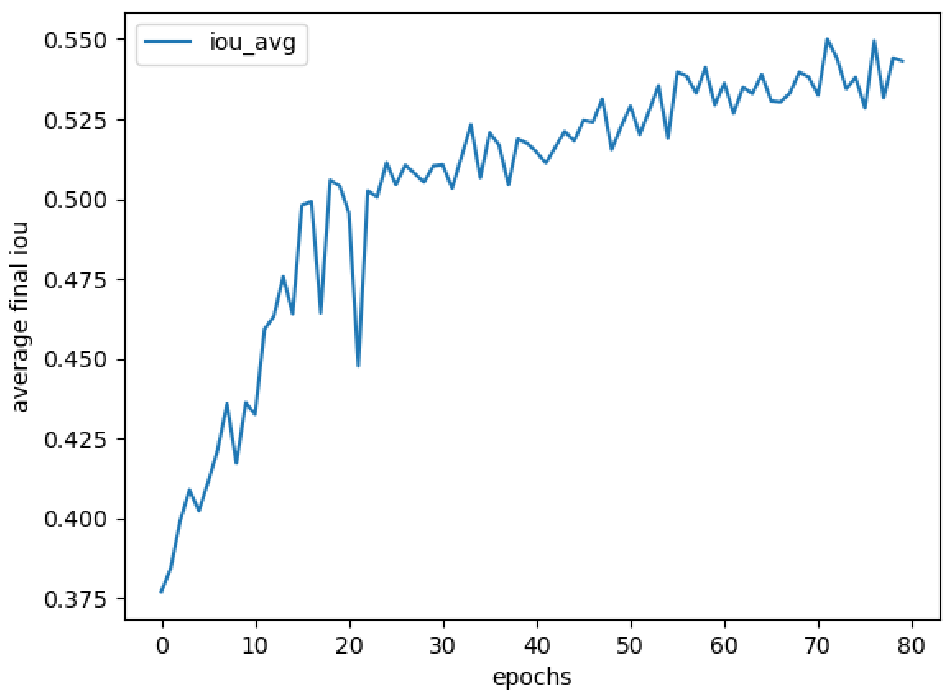


Fig. Average Intersection of Union Plot for the above experiment

Discussion: From the above 3 tables one can conclude that generally having the IoU threshold above 0.80 yielded better results with some better improvement of about 4-5% over all ranges. The mAP is calculated for a fixed threshold of 50. Since all the experiments were ran just once and updated just once and that too in the case when the number of images where more. So there is a lot of potential work that needs to be automated for the experiment that is to be performed. The average intersection of union was about 50 percent when number of images trained where more because of the ability of the agent to have a good amount of experience replay memory based on which it can take action when it lands on a specific state.

Conclusion

Reinforcement learning and it’s application brought together with the use of deep learning for the purpose of using it to create a new field of Deep reinforcement learning is very fascinating and exciting. Having just scratching the surface of this fascinating field is quite challenging and full of information on every way possible. Sometimes it becomes overwhelming to not find answers of peculiar issues. This specific report for the end sem evaluation consists of all the basics one needs to get started in the field of deep reinforcement learning. Though the experiments does not seem like much because it seems very similar and quite trivial to be honest, the main point of contention always boils down to processing power of the GPU available because we are dealing with raw pixels of a particular image. Further exploration is quite necessary and also my aim for the future work I am going to do in this particular topic.

Future Work

- The future work of mine which is the main focus of mine is actually suggesting bounding boxes of image based on state of the art approaches like YOLO and R-CNN.
- The main reason to explore the above two methods is their excellent approach to deal with overlapping object
- Since my main aim is to actually work on the approach to improve the bounding boxes for overlapping object the above state-of-art methods can help me explore various options regarding that.
- One possible approach that I am aiming to approach is modifying the present dqn with a custom dqn of mine based on most possibly one of the ResNet architechtures
- Every experiment done till now used a lot of manual labour on my individual part. For example calculation of mAP was the most cumbersome including manual updation of coordinates suggested by the DQN agent.
- DQN agent suggests just the class and corresponding coordinates of the probable coordinates to be updated. My future work also aims at automating this process so that whatever new annotation suggestions are given by the agent, it will overwrite the existing coordinates thus reducing manual labour.
- This report contains dataset of PASCAL. My other future work is exploring other standard datasets like COCO which is the bseline dataset used and also dataset images of warehouses because of the reason being that one of the area of potential application can be found in the field of logistics because it happens more often than not that we may find a huge number of images overlapped with each other at warehouses and store items.
- So from the above list is quite comprehensive and can be boiled down to 3 things, first being alternate detection algorithms for both images and dqn, changing the dataset, automating much of the manual labour especially regarding metric calculation

References

[1] Nasteski, Vladimir. "An overview of the supervised machine learning methods." *Horizons. b* 4 (2017): 51-62.

[2] Celebi, M. Emre, and Kemal Aydin, eds. *Unsupervised learning algorithms*. Berlin: Springer International Publishing, 2016.

[3] Sutton, Richard S., and Andrew G. Barto, "Reinforcement learning", *Journal of Cognitive Neuroscience* 11.1 (1999): 126-134.

[4] Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[5] Le, N., Rathour, V. S., Yamazaki, K., Luu, K., & Savvides, M. (2021). Deep reinforcement learning in computer vision: a comprehensive survey. *Artificial Intelligence Review*, 1-87.

[6] Watkins, C. J. C. H. (1989). Learning from delayed rewards, PhD Thesis

[7] Christopher JCH Watkins and Peter Dayan. Q-learning. Machine learning, 8(3-4):279–292, 1992.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

[9] Artificial Intelligence, Leonardo Araujo dos Santos, Independent Textbook

[10] Ayle, M., Tekli, J., El-Zini, J., El-Asmar, B., & Awad, M. (2020). BAR — A Reinforcement Learning Agent for Bounding-Box Automated Refinement. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(03), 2561-2568

[11] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[12] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255

[13] Everingham, M., Van Gool, L., Williams, C.K.I. *et al.* The PASCAL Visual Object Classes (VOC) Challenge. *Int J Comput Vis* **88,** 303–338 (2010).

[14] Howard, Ronald A. "Dynamic programming and markov processes.", Technology Press of Massachusetts Institute of Technology, (1960).

[15] Blog Post Series by Jake Bennett, (https://randomant.net/reinforcement-learning-concepts/)

[16] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[17] Lee, S., Kwak, S. and Cho, M., 2018, December. Universal bounding box regression and its applications. In *Asian Conference on Computer Vision* (pp. 373-387). Springer, Cham.

[18] Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

[19] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[20] Redmon, J. and Farhadi, A., 2017. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

[21] Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

[22] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, pp.1097-1105.