**CERTIFICATE**

Certified that **ANINDYA GHOSAL (Enrollment No.: 2020PSP3007)** has carried out their search work presented in this thesis entitled **“Bounding Box Refinement Agent For Overlapping Objects"** for the award of **Master of Technology** from Netaji Subhas University of Technology, New Delhi, under my supervision. The thesis embodies the results of original work, and studies are carried out by the student himself/herself (print only that is applicable)and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Prof. Jyotsna Singh**

**Professor**

**Division of Electronics and Communication Engineering**

**Netaji Subhas University Of Technology, New Delhi**

Date:

**ABSTRACT**

With the advancement in the field of computer vision and neural networks, it feels like every day a new architecture with better performance than the previous architectures is being created. Massive networks with multiple layers and skip connections with a plethora of loss functions and approaches are being used. One of the most important areas of the use of convolutional neural networks is in the field of object detection. Now in a business use case when an object detection algorithm is used, all the compatible packages are installed and the functioning of the business is smooth. Now it is not possible for the organizations to scrap the framework they put in place every time a new detection algorithm is in the market.

So to save cost, effort, and time, the work in this literature is aimed at correcting incorrect bounding boxes. Incorrect bounding boxes in the context of this literature are the change in the target object of object detection. If the target object in a detection algorithm is changed, it is not possible to re-annotate the complete dataset and retrain the complete detection model. So the approach in this work aims at using the concept of reinforcement learning on a subset of data annotated separately. Using the agent trained on the subset of data, make it learn how to correct the bounding boxes and then used the agent to correct the changed target object detection over the complete dataset. An object detection algorithm is used to first detect target objects, then based on ground truth boxes the agent learns to correct the boxes on the training dataset, and the knowledge it learns is transferred for testing.

**ACKNOWLEDGEMENTS**

I would like to express our sincere gratitude to Prof. Jyotsna Singh, Department of Electronics and Communication Engineering, Netaji Subhas University of Technology, for granting me this chance to work on a trending research area as a part of my Master’s Thesis Project. Without her thoughtful guidance, meticulous supervision, and incessant encouragement, this thesis would not have emerged as it has. Her insights and experience have been the guiding source for our research and study. I would also like to thank the entire Department of ECE for providing an opportunity to study this project and enhance my knowledge of relevant industrial skills. I also want to thank the department for ensuring an environment conducive to learning. This project has not only provided me with in-depth knowledge of relevant and new-age technologies but also provided a platform for me to work together with her as a team. I would like to thank my parents and family members for encouraging and supporting me in the pursuit of my thesis work.

**ANINDYA GHOSAL**

**2020PSP3007**

**Master of Technology, Signal Processing**

**Department of Electronics and Communication Engineering**

**Netaji Subhas University of Technology**

**New Delhi – 110078**

**CONTENTS**

Page No.

**Title**

**Certificate ii**

**Abstract iii**

**Acknowledgment iv**

**List of Tables vii**

**List of Figures viii**

**CHAPTER 1: INTRODUCTION 1-4**

1.1. PRELIMENARY INSIGHT 1

1.2. BRIEF SURVEY OF LITERATURE 3

1.3. PROBLEM STATEMENT 3

1.4. ORGANIZATION OF THESIS 3

**CHAPTER 2: MACHINE LEARNING TECHNIQUES 5-13**

2.1. BACKGROUND 5

2.2. SUPERVISED LEARNING 7

2.3. UNSUPERVISED LEARNING 11

2.4. REINFORCEMENT LEARNING 12

**CHAPTER 3: CONVOLUTIONAL NEURAL NETWORKS 14-32**

3.1. NEURONS 14

3.2. PERCEPTRON 16

3.3. BACKPROPAGATION 17

3.4. NEOCOGNITRON 18

3.5. LeNet : THE FIRST MAINSTREAM CNN 20

3.6. AlexNet AND GoogleNet 21

3.7. VGG 23

3.8. ResNet 25

3.9. OBJECT DETECTION 27

3.10. FASTER R-CNN 30

**CHAPTER 4: REINFORCEMENT LEARNING 33-40**

4.1. INTRODUCTION 33

4.2. BASIC REINFORCEMENT LEARNING PROBLEM 34

4.3. MARKOV DECISION PROCESS 35

4.4. Q-LEARNING 38

4.5. DEEP Q-NETWORKS 39

**CHAPTER 5: PROBLEM STATEMENT 41-44**

5.1. MOTIVATION 41

5.2. INTERSECTION OVER UNION AND BOUNDING BOX 42

5.3. METHODOLOGY 42

**CHAPTER 6: RESULT AND DISCUSSION 45-52**

6.1. MODIFIED Faster R-CNN IMPLEMENTATION 45

6.2. DQN IMPLEMENTATION 49

**CHAPTER 7: FUTURE SCOPE 53**

**REFERENCES 54-59**

**List of Tables**

Page No.

**Table 1:** mAP values corresponding to modified implementation Faster

R-CNN 51

**Table 2:** AP values corresponding to DQN agent refinement 51

**List of Figures**

Page No.

Fig 1.1: Evolution of the field of machine learning and artificial intelligence as

we know it to date 2

Fig 2.1: Examples of handwritten digits 6

Fig 2.2: Different types of machine learning algorithms 7

Fig. 2.3: Organizing input dataset into known classes 7

Fig. 2.4: A pictorial representation of a supervised learning process 8

Fig. 2.5: A pictorial representation of a Supervised model 9

Fig. 2.6: An example of a logistic function 9

Fig. 2.7: Support Vector Machine 10

Fig. 2.8: A decision tree approach 11

Fig. 2.9: Example of KNN 12

Fig. 2.10: An example of how reinforcement learning works 13

Fig. 3.1: A schematic representation of a biological neuron 15

Fig. 3.2: A basic perceptron model 16

Fig. 3.3: A perceptron learning algorithm 17

Fig. 3.4: Backpropagation algorithm for a neural network 18

Fig. 3.5: Figure denoting the relationship between the hierarchical model and

neocognitron 19

Fig 3.6: Diagram outlining the interconnections layerwise in neocognitron 19

Fig. 3.7: LeNet Architecture 21

Fig. 3.8: Architecture of AlexNet 22

Fig. 3.9: Inception Block 23

Fig. 3.10: Inception Block with dimension reduction 23

Fig. 3.11: The Layer structure of various VGG types 24

Fig. 3.12: 34 Layered ResNet Architecture 25

Fig. 3.13: A residual block from the ResNet Architecture 26

Fig. 3.14: Different varieties of residual blocks 26

Fig. 3.15: R-CNN Object Detection Method 28

Fig. 3.16: Spatial Pyramid Pooling Layer 29

Fig. 3.17: Fast R-CNN Object Detection 30

Fig. 3.18: A Region Proposal Network Architecture in Faster R-CNN 31

Fig. 4.1: Agent-Environment Interaction in a Markov Decision Process 36

Fig. 4.2: A working flow of an MDP in action 37

Fig. 4.3 Different reinforcement learning approaches 38

Fig. 4.4: A Q-learning algorithm 39

Fig. 4.5: A Deep Q-Network 40

Fig. 5.1: Intersection-Over-Union and Bounding box coordinates 42

Fig. 6.1.1: RoI and RPN box losses 45

Fig. 6.1.2: RoI and RPN class loss 45

Fig. 6.1.3: Total Loss in the approach 46

Fig. 6.1.4: RoI and RPN box losses for Approach 1 46

Fig. 6.1.5: RoI and RPN class loss for Approach 1 46

Fig. 6.1.6: Total Loss in Approach 1 47

Fig. 6.1.7: RoI and RPN box losses for Approach 2 47

Fig. 6.1.8: RoI and RPN class loss for Approach 2 47

Fig. 6.1.9: Total Loss in Approach 2 48

Fig. 6.1.10: RoI and RPN box losses for Approach 3 48

Fig. 6.1.11: RoI and RPN class loss for Approach 3 48

Fig. 6.1.12: Total Loss in Approach 3 49

Fig. 6.2.1: IoU Loss Plot for Approach 1 49

Fig. 6.2.2: Average IoU after correction in Approach 1 49

Fig. 6.2.3: Average predicted Q-Value of Approach 1 50

Fig. 6.2.4: IoU Loss Plot for Approach 2 50

Fig. 6.2.5: Average IoU after correction in Approach 2 50

Fig. 6.2.6: Average predicted Q-Value of Approach 2 50

Fig. 6.2.7: IoU Loss Plot for Approach 3 51

Fig. 6.2.8: Average IoU after correction in Approach 3 51

Fig. 6.2.9: Average predicted Q-Value of Approach 3 51