Policy-Based Object Detection with Deep Reinforcement Learning

Dr. Archana Nanade
Computer Department
Mukesh Patel School of Technology
Management and Engineering,
SVKM's NMIMS
Mumbai, India
archana.nanade@nmims.edu

Yuval Mehta
Computer Department
Mukesh Patel School of Technology
Management and Engineering,
SVKM's NMIMS
Mumbai, India
yuvalmehta.728@gmail.com

Vaishnavi Kamath
Computer Department
Mukesh Patel School of Technology
Management and Engineering,
SVKM's NMIMS
Mumbai, India
vaishnaveekamath@gmail.com

Nisha Kini
Computer Department
Mukesh Patel School of Technology Management and
Engineering, SVKM's NMIMS
Mumbai, India
kini.nisha24@gmail.com

Varun Nair
Computer Department
Mukesh Patel School of Technology Management and
Engineering, SVKM's NMIMS
Mumbai, India
nairvarun31@gmail.com

***Abstract*—In this project, we present a reinforcement learning-based approach for object detection, focused on generating bounding boxes using agent based framework that observes the current state, typically a region of the image and associated features—and selects an action to adjust the position and size of a bounding box. By leveraging powerful deep learning architectures—including ResNet, EfficientNet, MobileNet, ConvNeXt, Vision Transformer (ViT), and Swin Transformer—we built a robust detection pipeline. Our methodology involves class-wise agent training, enabling specialized detection for individual animal classes such as cats, dogs, and birds. We evaluated model performance using Average Precision (AP) at varying Intersection over Union (IoU) thresholds and compared our results with benchmark datasets.**

***Index Terms*—Object Detection, Reinforcement Learning, Deep Q-Network, Feature Extraction**

I. Introduction

Object detection is a fundamental task in computer vision, involving the identification and localization of objects within images. Traditional object detection techniques rely heavily on supervised learning and annotated datasets. However, reinforcement learning (RL) introduces a dynamic, decision-driven approach where an agent learns to optimize bounding box predictions through interactions with an environment, rather than static labeled inputs.

In this project, we explore a reinforcement learning-based pipeline to detect and localize animals in images. Our system is designed to draw accurate bounding boxes around objects by training agents to identify and adjust predictions in a class-wise manner. This targeted strategy enhances learning efficiency and detection accuracy, particularly when dealing with a diverse set of animal categories.

To support our pipeline, we integrated several state-of-the-art deep learning architectures as feature extractors: ResNet, EfficientNet, MobileNet, ConvNeXt, ViT, and Swin Transformer. These architectures were selected for their unique strengths in handling image representation and pattern recognition. The training process involved separate reinforcement learning agents for each class, allowing us to tailor detection strategies to the specific features of each object type.

Our evaluation focused on Average Precision (AP) metrics across various thresholds to rigorously assess the performance of each model. The comparative analysis and visualizations highlighted the strengths and trade-offs of each architecture, offering valuable insights into the suitability of different models for specific use cases, including real-time detection and complex pattern recognition.Through comparative analysis and visual inspection, we provide insights into their applicability for real-time detection scenarios and tasks requiring complex pattern recognition.

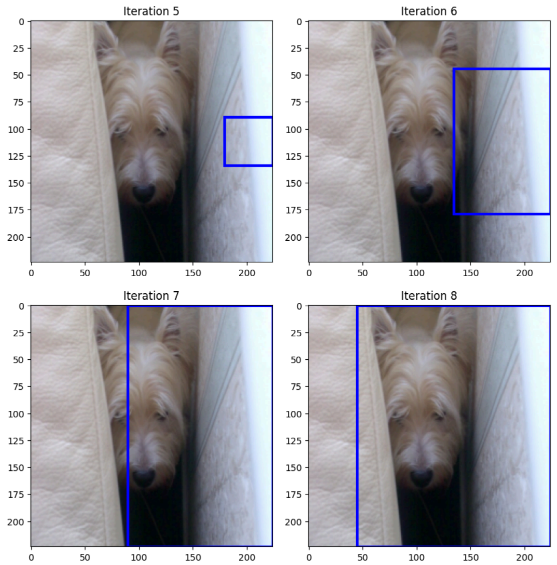

Fig. 1: Progressive refinement of object localization using a Deep RL-based detection algorithm.

Machine Learning Approach	**Key Technological Contribution**
Deep Reinforcement Learning (DRL)	Efficient object localization through sequential decision-making and region analysis
	Specialized applications like logo detection and organ localization in medical imaging
Convolutional Neural Networks (CNN)	Efficient localization without bounding boxes using convolutional architectures
	Model switching and superpixel training for improved accuracy in object detection
Weakly Supervised Learning	Localization and classification with minimal labeled data using weak supervision
	Suitable for scenarios with scarce annotations, focusing on structure preservation
Multitask and Hybrid Methods	Improved generalization through shared representations across multiple tasks
	Tree-structured and sequential approaches for dynamic object localization
Vision Transformers (ViT)	Simplified architecture for localization and segmentation with strong performance
	Knowledge distillation for dense object detection tasks

TABLE I: Overview of recent machine learning approaches for object localization.

II. Related Work

Object localization has been extensively studied in computer vision, with various approaches leveraging machine learning techniques to improve accuracy and efficiency. In this section, we review key works that have influenced our research, focusing on their methodologies and technological contributions in the context of object localization. The different architectures and their key technological contributions have been listed in Table I.

Deep Reinforcement Learning (DRL) has emerged as a powerful approach for active object localization. [1] proposed a DRL-based method for organ localization in CT images, introducing a novel action space for translation, scaling, and deformation, achieving efficient localization with limited training data (70 CT scans). Similarly, [2] applied DRL to logo recognition, using a confidence-guided reward function to handle varying logo sizes and positions without requiring position annotations. [3] introduced an active detection model that iteratively refines bounding boxes using DRL, achieving localization in 11–25 steps per object. [4] developed a tree-structured DRL approach for sequential object localization, enabling efficient detection of multiple objects with fewer proposals. [5] combined DRL with Bayesian filtering for active global localization, demonstrating faster localization in simulated environments. [6] explored DRL for object detection, comparing hierarchical and dynamic action settings. Additionally, [7] integrated DRL with multitask learning (MTL) to improve generalization across localization tasks.

Convolutional Neural Networks (CNNs) have also been widely used for object localization. [8] proposed an efficient CNN architecture for human pose estimation, using a multi-resolution approach to refine coarse predictions. [9] introduced a single CNN with model switching and superpixel training for object localization, improving accuracy by focusing on relevant image regions. [10]} enhanced CNN-based object detection with a novel bounding box regression loss (KL Loss) that models localization uncertainty, improving accuracy with minimal computational overhead.

Weakly Supervised Learning (WSL) has been explored to reduce the dependency on detailed annotations. [11] developed WILDCAT, a WSL approach using CNNs to jointly perform image classification, pointwise localization, and segmentation with only image-level labels. [12] proposed a WSL method that leverages structural information in convolutional features for more precise localization. [13] introduced a WSL approach using fully convolutional networks (FCNs) to locate objects without bounding boxes, validated on diverse datasets like surveillance and aerial images.

Vision Transformers (ViT) and Knowledge Distillation (KD) have also contributed to recent advancements. [14] presented a simple single-scale ViT for object detection and instance segmentation, achieving competitive performance with a simpler architecture. [15] applied KD to dense object detection, using a novel distillation strategy to transfer localization knowledge from a teacher to a student model, improving performance on the MS COCO dataset.

III. Methodology

In this work, we propose a reinforcement learning (RL)-based object detection framework tailored for animal localization in images. Our pipeline integrates multiple deep learning architectures as feature extractors within a Deep Q-Network (DQN) structure, enabling intelligent agent-based bounding box predictions. The overall methodology encompasses architecture integration, agent design, training strategy, and performance evaluation, as illustrated in

A. *Feature Extractor Integration*

To effectively capture diverse visual features, we employed six advanced convolutional and transformer-based architectures: ResNet, EfficientNet, MobileNet, ConvNeXt, Vision Transformer (ViT), and Swin Transformer. Each model was integrated into a unified FeatureExtractor module, which standardizes the output for downstream processing in the DQN. The final classification layers of these models were removed and replaced with identity mappings to extract high-dimensional features suitable for RL tasks.

TABLE II: Models Used for Feature Extraction

Model	**Variant**	**Layer Used for Feature Extraction**	**Feature Dim**	**Parameters (M)**
ResNet	ResNet18	Before fc layer	512	11.7
EfficientNet	EfficientNet-B0	Before second classifier layer	Variable	5.3
MobileNet	MobileNet V3 Large	First classifier layer	Variable	5.4
ConvNeXt	ConvNeXt Base	After second classifier layer (LayerNorm+Flatten)	Variable	89
ViT	ViT Base 32	heads.head	Variable	86
Swin Transformer	Swin V2 Base	head layer	Variable	88

B. *Deep Q-Network (DQN) Architecture*

The extracted features are passed into a DQN module defined by a three-layer fully connected neural network:

- Input: Concatenation of feature vector and an 81-dimensional one-hot encoded bounding box state.
- Hidden Layers: Two linear layers of size 1024, activated with ReLU and regularized with dropout (p = 0.2).
- Output: 9 discrete action values corresponding to bounding box adjustments (left, right, up, down, enlarge, shrink, fatten, stretch, trigger).

C. *RL Agent Design and Training*

Each object class (e.g., cat, dog, bird) is assigned a separate RL agent to enable class-specific learning and policy optimization. The agents follow an ε-greedy policy for exploration-exploitation balance and use experience replay memory to stabilize learning. Actions are designed to iteratively refine the bounding box placement on the object of interest. The agent interacts with the environment as follows:

1) **State**: Current image and bounding box.
2) **Action**: Predicted by DQN.
3) **Reward**: Based on IoU improvement with ground truth.
4) **Next State**: Updated bounding box.
5) **Experience Replay**: Transition stored and sampled for training.

The training and validation logic is implemented in the train_validate method, which handles class-wise looping, logging, and metrics calculation.

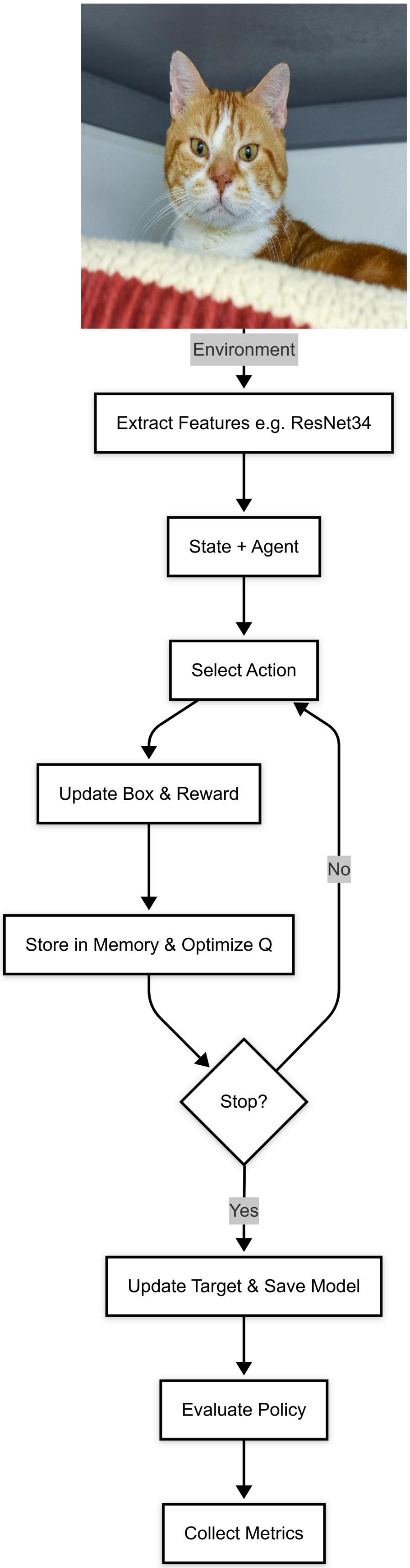

Fig. 2: Overview of Pipeline. The agent receives image input from the environment, extracts features, selects an action, and receives feedback in the form of reward. Transitions are stored and used to optimize the Q-network. The process repeats until termination, after which the model is updated and evaluated.

D. Feature Extractor-Specific Observations

- ResNet18: Served as a stable backbone with a 512-dimensional feature output. Known for its depth efficiency, it enabled consistent pattern learning. It demonstrated smooth convergence and robust results in the “cat” class experiments.
- EfficientNet: Provided a good trade-off between model size and accuracy. Feature extraction from its classifier yielded fast training with no major compromise in representational quality, making it ideal for constrained compute environments.
- MobileNet: Lightweight architecture favored for real-time detection. Feature extraction from its initial classifier layer allowed low-latency operation and decent bounding box alignment, especially on simpler objects.
- ConvNeXt: A modern convolutional model, ConvNeXt provided high-quality features and strong generalization. Used after its second classifier layer, it improved policy learning under complex visual contexts.
- ViT: The Vision Transformer modeled long-range dependencies and performed well in intricate scenes. Extracted features from the head layer led to superior object localization in cluttered or occluded regions.
- Swin Transformer: Showed robust performance in spatially hierarchical image regions. Extracted from its head layer, Swin excelled in scenarios requiring fine-grained detection.

E. Evaluation Metrics and Dataset

The models were trained and evaluated on the PASCAL VOC 2007 and 2012 datasets. For each class, performance was logged using Average Precision (AP) and Recall across multiple Intersection over Union (IoU) thresholds: 0.1, 0.2, 0.3, 0.4, and 0.5. These metrics were averaged over all Episodes to assess stabilized performance. Table III summarizes the results obtained using different feature extractors as shown in Table II.

IV. Results

A. Training Results

The figures given below depict the Average Precision (AP) of the different models across different IoU thresholds mentioned in Table II.

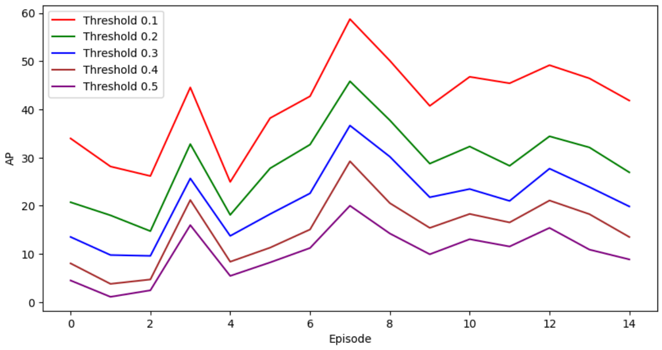

Fig. 3: AP for Different IoU Thresholds for ResNet

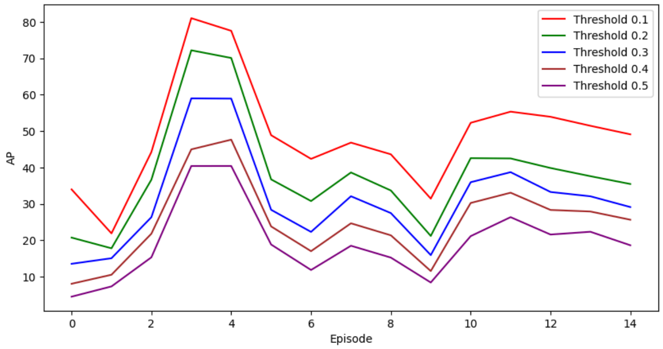

Fig. 4: AP for Different IoU Thresholds for Efficient Net

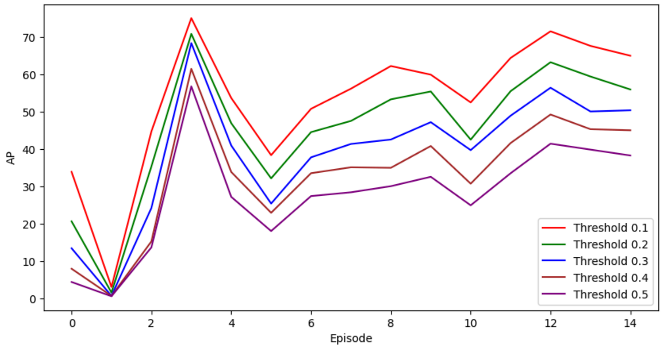

Fig. 5: AP for Different IoU Thresholds for MobileNet

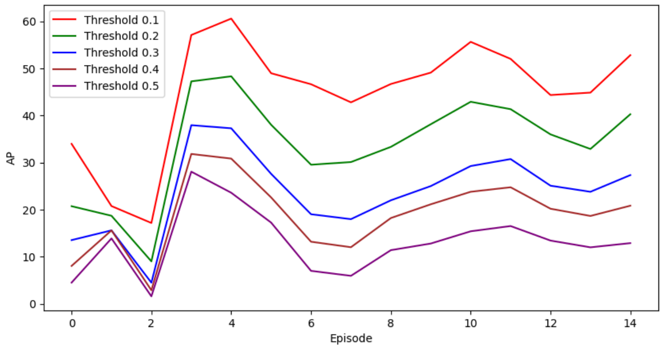

Fig. 6: AP for Different IoU Thresholds for ConvXNet

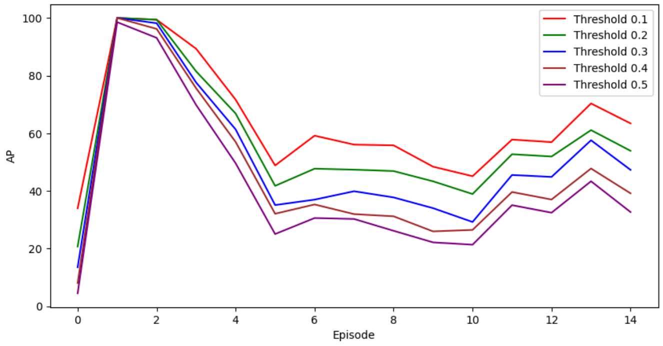

Fig. 7: AP for Different IoU Thresholds for ViT

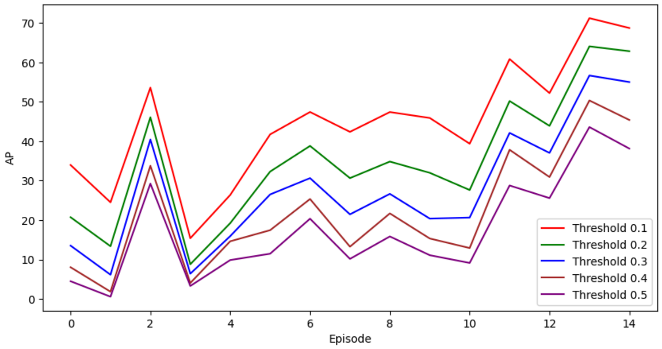

Fig. 8: AP for Different IoU Thresholds for Swin Transformer

B. *Evaluation Results*

TABLE III: Performance Metrics for Feature Extractors Averaged Over Each Class

Feature Extractor	Mean Average Precision (mAP)					Mean Recall				
	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
ResNet	88.17	83.03	77.80	71.90	66.80	93.06	89.82	86.12	82.50	78.54
EfficientNet	92.64	88.17	84.84	82.42	80.00	95.45	92.87	90.94	89.53	88.05
MobileNet	**94.48**	**93.41**	**90.46**	**87.32**	**83.79**	**96.72**	**95.99**	**94.38**	**92.57**	**90.57**
ConvNeXt	91.99	88.18	85.97	83.72	80.72	95.31	92.69	91.21	89.51	87.51
ViT	92.56	88.57	84.97	81.89	78.24	95.67	93.41	91.02	89.32	86.87
Swin Transformer	86.30	82.44	78.26	73.77	70.57	91.49	89.12	86.18	82.92	80.07

C. *Discussion*

From the figures and tables given above it can be clearly seen that **MobileNet** comes out as the leading model with the highest **mAP** and **Recall** across **all** the IoU Thresholds. While **ConvNeXt** and **EfficientNet** demonstrate strong overall performance with balanced results, **ViT** delivers moderate outcomes with **ResNet** and **Swin Transformer** lagging behind.

Most models show an overall decline in AP after the initial spike, suggesting at a potential over fitting or model degradation issue through the episodes.

It can be concluded that the choice of confidence threshold is critical -lower thresholds may benefit recall intensive applications whereas higher thresholds improve precision and reduce false postives

V. Future Scope

The current project effectively generates bounding boxes around animals in input images using reinforcement learning. While the current model demonstrates strong performance, several enhancements can be made to expand its capabilities and applicability:

- **Extension to Video Data:** Incorporating video data will allow the model to track animals across frames, providing better context for movement patterns and behavior analysis. This enhancement will improve the detection of animals in motion through motion tracking, which is particularly useful for applications like wildlife monitoring and surveillance in dynamic environments.
- **Multi-Class Detection:** The model can be extended to detect multiple animal species within the same image, enabling it to handle more complex scenes with diverse wildlife. This scalability will support real-world applications such as ecological studies where multiple species coexist, enhancing biodiversity monitoring efforts.
- **Multi-Object Detection:** Improving the model to identify and generate bounding boxes for multiple animals, even in crowded or overlapping situations, will enhance its robustness by addressing challenges with overlapping bounding boxes.
- **Instance Segmentation:** Moving towards instance segmentation will allow the model to distinguish between individual animals more accurately, providing detailed insights for research and conservation projects.
- **Real-Time Deployment:** Optimizing the model for real-time inference on edge devices like drones or mobile cameras through edge computing will enable on-the-spot animal detection in the field. Additionally, model compression techniques such as quantization and pruning will reduce model size, making it more efficient for resource-constrained environments without sacrificing performance.
- **Integration with Advanced RL Techniques:** Implementing curriculum learning strategies can improve model performance by gradually increasing the complexity of detection tasks. Furthermore, enabling the model to adapt quickly to new animal species with minimal data through meta-learning will enhance its generalization across different environments and datasets.

VI. Conclusion

In this project, we implemented a comprehensive pipeline for object detection using reinforcement learning, leveraging various deep learning architectures such as ResNet, EfficientNet, MobileNet, ConvNeXt, ViT, and Swin Transformer. The workflow encompassed dataset preparation, class-wise agent training, performance evaluation, and visualization of results.

The class-wise training approach handled each class (e.g., cat, dog, bird) individually, enabling targeted learning for diverse object categories. This method allowed the model to adapt to specific features of each class, thereby improving detection accuracy. For evaluation, we used Average Precision (AP) at different thresholds, comparing our results with benchmark paper values. The bar plots generated during this process clearly highlighted performance gaps and identified areas for potential improvement.

Our results provided several key insights. Models trained with architectures like ResNet and EfficientNet showcased strong generalization across classes. Transformer-based models, such as ViT and Swin Transformer, performed exceptionally well in capturing complex patterns, particularly in high-dimensional data scenarios. Notably, the ConvNeXt architecture demonstrated efficiency in both speed and accuracy, making it suitable for real-time applications.

While the current approach has shown promising results, several avenues for future work can be explored. These include hyperparameter optimization to achieve better performance, the use of ensemble methods to combine the strengths of different architectures, and real-time deployment using lightweight models like MobileNet for edge devices.

In conclusion, this project not only validated the effectiveness of reinforcement learning in object detection but also highlighted the importance of architecture selection and targeted training for optimal performance.

References

[1] F. Navarro, A. Sekuboyina, D. Waldmannstetter, J. Peeken, S. Combs, and B. Menze, “Deep Reinforcement Learning for Organ Localization in CT,” in *Proceedings of Machine Learning Research - MIDL*, 2020.

[2] M. Fujitake, “RL-LOGO: Deep Reinforcement Learning Localization for Logo Recognition,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024. doi: 10.1109/ICASSP48485.2024.10447388.

[3] J. C. Caicedo and S. Lazebnik, “Active Object Localization with Deep Reinforcement Learning,” *arXiv preprint arXiv:1511.06015*, 2015.

[4] Z. Jie, X. Liang, J. Feng, X. Jin, W. F. Lu, and S. Yan, “Tree-structured Reinforcement Learning for Sequential Object Localization,” in *Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

[5] D. S. Chaplot, E. Parisotto, and R. Salakhutdinov, “Active Neural Localization,” *arXiv preprint arXiv:1801.08214*, 2018.

[6] M. Samiei and R. Li, “Object Detection with Deep Reinforcement Learning,” *arXiv preprint*, 2022, [Online]. Available: https://arxiv.org/abs/2208.04511

[7] X. Liu, L. Gao, X. Zhen, J. Tang, and T. Huang, “Multitask Learning for Object Localization With Deep Reinforcement Learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 6, pp. 1705–1715, 2019, doi: 10.1109/TNNLS.2018.2877076.

[8] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient Object Localization Using Convolutional Networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] Q. Zhang, X. Wang, J. Yang, W. Li, and X. Wu, “Object Localization Through a Single Multiple-Model Switching CNN and a Superpixel Training Approach,” *Applied Soft Computing*, vol. 112, p. 107802, 2021, doi: 10.1016/j.asoc.2021.107802.

[10] Y. He, X. Zhang, M. Savvides, and K. Kitani, “Softer-NMS: Rethinking Bounding Box Regression for Accurate Object Detection,” *CoRR*, 2018, [Online]. Available: http://arxiv.org/abs/1809.08545

[11] T. Durand, T. Mordan, N. Thome, and M. Cord, “WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] X. Pan *et al.*, “Unveiling the Potential of Structure Preserving for Weakly Supervised Object Localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[13] J. Ribera, D. Guera, Y. Chen, and E. J. Delp, “Locating Objects Without Bounding Boxes,” *arXiv preprint arXiv:1806.07564*, 2018.

[14] W. Chen *et al.*, “A Simple Single-Scale Vision Transformer for Object Detection and Instance Segmentation,” *arXiv preprint arXiv:2112.09747v3*, 2022.

[15] Z. Zheng *et al.*, “Localization Distillation for Dense Object Detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.