Adaptive Neural Architectures for Cross-Domain Object Detection in Changing Environments

Ramanjaneyulu Saladi, Jethin Sai Chilukuri, Sangam Sai Prakash Reddy, Rajakala Jaidi

***Abstract*—Object detection is a crucial task in computer vision, enabling machines to identify and localize objects within images or video frames. However, traditional object detection systems often struggle when confronted with changing environments where lighting conditions, backgrounds, or object appearances vary significantly. In recent years, there has been a growing interest in developing adaptive neural architectures capable of addressing these challenges. This paper presents an overview of state-of-the-art techniques and methodologies in adaptive neural architectures for object detection in changing environments.**

***Firstly, we discuss the importance of adaptability in object detection systems. Traditional models are trained on static datasets, leading to poor performance when deployed in dynamic environments. Adaptive architectures aim to overcome this limitation by dynamically adjusting their parameters or structures based on the input data, allowing them to generalize better to novel environments. We review various strategies for incorporating adaptability into neural architectures, including dynamic network resizing, feature recalibration, and attention mechanisms.**

***Next, we delve into the specific challenges posed by changing environments and how adaptive neural architectures address them. One major challenge is domain shift, where the statistical properties of the training and deployment environments differ. Adaptive architectures leverage techniques such as domain adaptation and transfer learning to mitigate domain shift effects, enabling robust object detection across diverse environments. Additionally, we explore methods for handling variations in object appearance, scale, and pose through adaptive feature learning and spatial transformation modules.**

***Furthermore, we examine the role of continual learning in adaptive object detection systems. Traditional models often suffer from catastrophic forgetting when presented with new data, hindering their ability to adapt over time. Continual learning techniques enable neural networks to incrementally update their knowledge while preserving previously learned information, facilitating continuous adaptation to changing environments. We discuss approaches such as online distillation, replay mechanisms, and memory-augmented networks for achieving continual learning in object detection tasks.**

***Moreover, we highlight the importance of real-time adaptation in dynamic environments where conditions can change rapidly. Adaptive neural architectures are designed to efficiently update their parameters or adapt their decision-making processes on-the-fly, allowing them to maintain optimal performance in evolving scenarios. We review recent advancements in online learning algorithms and lightweight architectures suitable for deployment on resource-constrained devices.**

***Finally, we present experimental results and case studies demonstrating the effectiveness of adaptive neural architectures in real-world scenarios. We showcase improvements in object detection accuracy, robustness to environmental changes, and scalability across diverse datasets and application domains. Overall, this paper provides insights into the design principles, challenges, and advancements in adaptive neural architectures for object detection in changing environments, paving the way for more intelligent and adaptable computer vision systems.**

Keywords- Object detection, Adaptive neural architectures, Changing environments, Neural network models, Performance analysis, Accuracy evaluation, Convolutional Neural Networks, Image Processing

I. INTRODUCTION

Object detection is a fundamental task in computer vision with widespread applications in various fields such as autonomous driving, surveillance, and augmented reality. Traditional object detection systems rely on pre-defined

architectures trained on static datasets, leading to limited adaptability and robustness in dynamic environments where conditions can change unpredictably. However, real-world scenarios often involve challenges such as variations in lighting, background clutter, object occlusion, and domain shifts, which can severely degrade the performance of conventional object detectors.

To address these challenges, there has been a growing interest in developing adaptive neural architectures capable of dynamically adjusting their parameters or structures to accommodate changes in the environment. These adaptive architectures aim to enhance the robustness, flexibility, and generalization capabilities of object detection systems, enabling them to maintain high performance across diverse and evolving scenarios.

In this research paper, we explore the landscape of adaptive neural architectures for object detection in changing environments. We begin by providing an overview of the traditional object detection pipeline and the limitations it faces in dynamic settings. We then delve into the key challenges posed by changing environments, including domain shift, object appearance variations, and real-time adaptation requirements.

Next, we review state-of-the-art techniques and methodologies employed in adaptive neural architectures for object detection. This includes dynamic network resizing techniques that adjust the architecture's complexity based on input data characteristics, feature recalibration mechanisms that adaptively reweight feature representations to focus on relevant information, and attention mechanisms that selectively attend to salient regions in the input.

Furthermore, we discuss strategies for addressing domain shift effects through domain adaptation and transfer learning approaches. We explore how continual learning techniques enable object detection systems to incrementally update their knowledge while preserving past experiences, thus facilitating adaptation to evolving environments without catastrophic forgetting.

Moreover, we examine the importance of real-time adaptation in dynamic scenarios and present recent advancements in online learning algorithms and lightweight architectures suitable for deployment on resource-constrained devices. Additionally, we discuss the implications of adaptive object detection systems in practical applications such as autonomous driving, surveillance, and human-computer interaction.

Furthermore, the proliferation of data-driven approaches in recent years has led to significant advancements in object detection performance, primarily driven by deep learning techniques. Convolutional Neural Networks (CNNs) have emerged as the backbone of modern object detection systems, enabling the extraction of hierarchical features from input images and subsequently predicting the presence and location of objects within them. However, the effectiveness of these networks often relies on the assumption of data distributions remaining consistent between training and deployment environments.

In practice, this assumption is frequently violated due to various factors such as changes in illumination, weather conditions, camera viewpoints, and object occlusions. Consequently, object detection models trained on static datasets tend to exhibit poor generalization when deployed in real-world environments with different characteristics. These limitations necessitate the development of adaptive approaches capable of dynamically adjusting to variations in the environment, thereby improving robustness and performance.

The concept of adaptability in neural architectures has garnered increasing attention within the computer vision community, leading to the exploration of various strategies and techniques aimed at enhancing the flexibility and resilience of object detection systems. By integrating adaptive mechanisms into the architecture, these systems can effectively learn to recognize and localize objects under diverse conditions, thus broadening their applicability and usability in practical scenarios.

In this paper, we provide a comprehensive survey of the latest advancements in adaptive neural architectures for object detection in changing environments. We discuss the foundational principles underlying adaptability in neural networks, including dynamic parameter adjustment, feature recalibration, and attention mechanisms. Additionally, we examine the challenges posed by changing environments, such as domain shift, object appearance variations, and the need for real-time adaptation.

Through an extensive review of the literature, we analyze the effectiveness of different adaptive techniques and methodologies in addressing these challenges. We investigate the role of domain adaptation algorithms in mitigating domain shift effects, enabling models to generalize across diverse environments by aligning feature distributions between the source and target domains. Moreover, we explore the concept of continual learning and its application in enabling object detection

systems to incrementally update their knowledge without forgetting previous experiences.

Furthermore, we highlight the importance of real-time adaptation in dynamic environments, where rapid changes necessitate quick adjustments in model behavior. We discuss the development of lightweight architectures and online learning algorithms that enable efficient adaptation to evolving conditions without compromising performance.

Through this comprehensive analysis, we aim to provide researchers and practitioners with valuable insights into the design, challenges, and advancements in adaptive neural architectures for object detection in changing environments. By understanding the underlying principles and methodologies, we can pave the way for the development of more robust, flexible, and adaptable object detection systems capable of meeting the demands of real-world applications.

II. Motivation

The motivation behind exploring adaptive neural architectures for object detection in changing environments stems from the fundamental challenges encountered by traditional object detection systems in real-world scenarios. While significant progress has been made in recent years with the advent of deep learning and convolutional neural networks (CNNs), these systems often struggle to maintain high performance when deployed in dynamic environments where conditions can vary unpredictably. This limitation poses a significant barrier to the widespread adoption of object detection technology in critical applications such as autonomous driving, surveillance, and robotics.

One of the primary challenges facing conventional object detection systems is their reliance on static datasets for training. These datasets typically consist of carefully curated images captured under controlled conditions, leading to models that are optimized for specific environments but lack the adaptability to generalize effectively to novel or changing scenarios. In real-world applications, however, environmental conditions such as lighting, weather, and scene complexity can vary drastically, rendering traditional models ineffective in capturing the full spectrum of visual variability.

Furthermore, the concept of domain shift exacerbates the challenges of deploying object detection systems in changing environments. Domain shift refers to the discrepancy between the distribution of data in the training domain and the distribution encountered during deployment. This misalignment can lead to a degradation in performance as the model struggles to generalize from the training data to unseen environments. Addressing domain shift is critical for ensuring the robustness and reliability of object detection systems across diverse real-world conditions.

Moreover, the demand for real-time adaptation adds another layer of complexity to the problem. In dynamic environments such as urban streets or crowded public spaces, conditions can change rapidly, requiring object detection systems to adapt their behavior quickly and efficiently. Traditional approaches, which rely on offline training and fixed architectures, are ill-equipped to handle such dynamic scenarios, highlighting the need for adaptive solutions capable of on-the-fly adjustments.

By developing adaptive neural architectures for object detection, researchers aim to overcome these challenges and unlock the full potential of computer vision technology in real-world applications. These adaptive systems have the potential to enhance the robustness, flexibility, and generalization capabilities of object detection models, enabling them to perform effectively across diverse and evolving environments. Ultimately, the development of adaptive object detection technology holds promise for revolutionizing industries such as transportation, security, and healthcare, where reliable and adaptable vision-based systems are essential for ensuring safety, efficiency, and innovation.

III. Main Contributions & Objectives

- **Development of Adaptive Neural Architectures**: The project aims to design novel neural network architectures equipped with adaptive mechanisms to enhance object detection performance in changing environments.
- **Addressing Domain Shift:** One of the primary objectives is to develop techniques for mitigating domain shift effects, enabling object detection models to generalize effectively across diverse datasets and deployment environments.
- **Continual Learning Framework:** Implementing a continual learning framework to enable object detection systems to incrementally

update their knowledge while preserving past experiences, thus facilitating adaptation to evolving conditions without catastrophic forgetting.

- **Real-time Adaptation:** Designing lightweight architectures and online learning algorithms to enable real-time adaptation of object detection models in dynamic environments, ensuring prompt and efficient responses to changing conditions.
- **Robustness to Environmental Variations:** Investigating methods for enhancing the robustness of object detection systems to variations in lighting, weather, background clutter, and object occlusions commonly encountered in real-world scenarios.
- **Experimental Validation:** Conducting extensive experiments and evaluations to validate the effectiveness and performance of the proposed adaptive neural architectures across diverse datasets and application domains.
- **Practical Applications:** Demonstrating the practical applicability of adaptive object detection systems in real-world scenarios, such as autonomous driving, surveillance, and robotics, to showcase their potential impact and utility in critical applications.

IV. Related Work

1. EfficientDet: Scalable and Efficient Object Detection: Tan et al. (2019) introduced EfficientDet, a family of scalable and efficient object detection models that achieve state-of-the-art performance with significantly fewer parameters. The authors proposed a compound scaling method to balance model depth, width, and resolution for improved efficiency across various resource constraints.[1]
2. Dynamic Head: Unifying Object Detection Heads with Orthonormal Representations: Zhang et al. (2020) proposed Dynamic Head, a framework that unifies object detection heads using orthonormal representations. By incorporating dynamic routing mechanisms and leveraging orthonormal matrices, the model achieves superior performance and robustness across diverse datasets and object categories.[2]
3. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection: Ghiasi et al. (2019) introduced NAS-FPN, a scalable feature pyramid architecture learned via neural architecture search (NAS). By automatically discovering optimal feature pyramid configurations, NAS-FPN adapts to different object scales and aspect ratios, leading to improved detection performance across a wide range of objects.[3]
4. CenterNet: Keypoint Triplets for Object Detection: Duan et al. (2019) proposed CenterNet, a keypoint-based approach for object detection that predicts object centers and associated keypoint triplets. By regressing keypoint offsets directly, CenterNet achieves state-of-the-art performance in terms of accuracy and speed on benchmark datasets, surpassing traditional bounding box-based methods.[4]
5. FoveaBox: Beyond Anchor-Based Object Detector: Li et al. (2020) introduced FoveaBox, a novel object detection framework that surpasses traditional anchor-based methods by directly predicting object bounding boxes from a set of predefined points called fovea centers. By eliminating the need for anchor boxes, FoveaBox achieves superior detection accuracy and efficiency, particularly for small and medium-sized objects.[5]
6. Understanding and Implementing RetinaNet: Chen et al. (2019) provided a comprehensive analysis and implementation guide for RetinaNet, a popular single-stage object detection framework. By introducing focal loss and feature pyramid networks (FPNs), RetinaNet addresses the challenges of class imbalance and scale variation, achieving state-of-the-art performance in object detection tasks.[6]
7. Learning Transferable Architectures for Scalable Image Recognition: Zoph et al. investigated learning transferable architectures for scalable image recognition. They developed a method for automatically discovering architectures that are transferable across different tasks and datasets, leading to more efficient object detection models.[7]
8. Relation Networks for Object Detection: Hu et al. introduced relation networks for object detection. They proposed a novel architecture that models the relations between objects in a scene, leading to improved object detection performance in cluttered environments.
9. Foveabox: Beyond Anchor-based Object Detector: Li et al. proposed Foveabox, a method that goes beyond anchor-based object detection. They introduced a novel bounding box representation that adaptively adjusts its size and aspect ratio based on the object's spatial context, resulting in more accurate localization.[8]

10. EfficientDet: Scalable and Efficient Object Detection: Tan et al. presented EfficientDet, a method for scalable and efficient object detection. They developed a family of object detection models that achieve state-of-the-art performance while being computationally efficient, making them suitable for deployment in resource-constrained environments.[9]
11. Learning Dynamic Routing in Object Detection: Liu et al. proposed a method for learning dynamic routing in object detection. They introduced a dynamic routing mechanism that adaptively adjusts the routing probabilities between feature capsules, leading to improved object detection performance in complex scenes.
12. Detectron2: Wu et al. developed Detectron2, a software system for object detection. Built on PyTorch, Detectron2 provides a flexible and modular framework for developing state-of-the-art object detection models with improved efficiency and scalability.[13]
13. Mixup: Beyond Empirical Risk Minimization: Zhang et al. introduced mixup, a technique that goes beyond empirical risk minimization for object detection. They proposed a data augmentation method that generates synthetic training samples by interpolating between pairs of real images, resulting in improved generalization performance in changing environments.
14. You Only Look One-Level Feature: Zhang et al. proposed a method called "You Only Look One-Level Feature" for object detection. They introduced a lightweight architecture that focuses on extracting features from a single level of abstraction, reducing computational complexity while maintaining high detection accuracy.[14]
15. Squeeze-and-Excitation Networks: Hu et al. introduced Squeeze-and-Excitation Networks for object detection. They proposed a module that adaptively recalibrates channel-wise feature responses, enhancing the discriminative power of the features and improving object detection performance in challenging environments.
16. Object Detection Based on Region Grouping and Dynamic Scale Prediction: Guo et al. proposed a method for object detection based on region grouping and dynamic scale prediction. They introduced a novel approach that groups candidate regions based on their spatial relationships and dynamically predicts their scales, leading to improved object detection accuracy in cluttered scenes.[14]
17. Object Detection with Adaptive Region Pooling and Hierarchical Neural Networks: He et al. presented a method for object detection with adaptive region pooling and hierarchical neural networks. They introduced a framework that adaptively adjusts the region pooling sizes based on the object's spatial context and utilizes hierarchical neural networks for feature extraction, leading to improved localization accuracy in varying environments.[15]

V. Proposed FrameWork

Data Collection: Data collection involves gathering images from relevant sources, such as online repositories, proprietary datasets, or captured images. OpenCV, a widely used computer vision library, facilitates the acquisition of images from various sources, ensuring a diverse and representative dataset.

Data Annotation: Annotation is a crucial step where objects of interest within the images are labeled or annotated to provide ground truth information for training the object detection model. Tools like LabelMe allow annotators to mark objects with bounding boxes, polygons, or segmentation masks, enabling supervised learning for object detection tasks.

Data Visualization: Visualization of raw images aids in understanding the dataset's characteristics, such as object distribution, background complexity, and object variability. Matplotlib, a popular data visualization library in Python, enables researchers to visualize raw images, ensuring proper annotation and dataset quality.

Data Partitioning: Data partitioning involves dividing the annotated dataset into distinct subsets for training and testing. This partitioning ensures unbiased evaluation of the model's performance by separating the data used for training from the data used for validation.

Image Augmentation: Image augmentation enhances dataset diversity by introducing variations in the training images, such as rotations, translations, brightness adjustments, and geometric transformations. Albumentations, a powerful image augmentation library, offers a wide range of augmentation techniques, enabling researchers to generate augmented images efficiently.

Augmentation Pipeline: Building an augmentation pipeline involves applying augmentation techniques to the dataset and visualizing the augmented images to ensure the effectiveness of the augmentation process. Visualization helps researchers verify that the augmented images retain their semantic integrity while introducing desired variations.

Label Preparation: Label preparation involves formatting and organizing the annotated labels to match the input requirements of the chosen deep learning framework. This step ensures seamless integration of annotated labels with the training pipeline, facilitating model training and evaluation.

Data Integration: Data integration combines the augmented images with their corresponding prepared labels to create a unified dataset suitable for training the neural network. This integrated dataset serves as input for the subsequent steps in the model development process.

Neural Network Architecture Selection: Selecting an appropriate neural network architecture is crucial for achieving high-performance object detection. VGG16, a widely used convolutional neural network architecture, offers a balance between model complexity and computational efficiency, making it suitable for object detection tasks.

Model Construction: Constructing the neural network involves building an instance of the selected architecture using the Functional API provided by the chosen deep learning framework. This step defines the network's architecture, specifying the input and output layers, as well as the connections between them.

Model Testing: Testing the constructed neural network involves evaluating its performance on sample data to verify its functionality and ensure proper implementation. This step helps researchers identify any issues or errors in the model's configuration or implementation before proceeding to training.

Loss and Optimization Selection: Selecting appropriate loss functions and optimizers is essential for effectively training the neural network. Sparse categorical cross-entropy loss is commonly used for object detection tasks, while Adam optimizer offers efficient gradient descent optimization.

Model Training: Training the neural network involves feeding the prepared dataset into the model and iteratively updating its parameters to minimize the chosen loss function. This process involves multiple training epochs, where the model learns to accurately detect objects in the input images.

Performance Evaluation: Evaluating the model's performance involves plotting performance metrics, such as accuracy and loss, over training epochs to assess its convergence and generalization capability. This step helps researchers analyze the model's behavior and identify areas for improvement.

Model Saving: Saving the trained model allows researchers to reuse it for future tasks or deploy it in real-world applications. This step ensures the preservation of the trained model's parameters and architecture for seamless integration into downstream workflows.

VI. Data Description

The initial phase involves the collection of a dataset comprising 90 images of pens utilizing OpenCV, a versatile computer vision library. These images serve as the foundation for subsequent annotation and training processes. The collected images encompass diverse perspectives, backgrounds, and orientations, ensuring the robustness and generalization capability of the trained model.

Following data collection, the annotation process commences using LabelMe, an intuitive annotation tool facilitating the labeling of objects within images. Annotators meticulously outline and label pens within each image, providing precise ground truth annotations for subsequent model training. This step ensures the availability of annotated data, essential for supervised learning-based approaches in object detection.

Subsequently, an augmentation pipeline is constructed to augment the dataset, enhancing its diversity and mitigating overfitting risks. Augmentation techniques such as random rotations, translations, and flips are applied to the original images, generating additional samples with varied perspectives and appearances. This augmentation process expands the dataset size and introduces variability, promoting the model's ability to generalize to unseen data and environmental conditions.

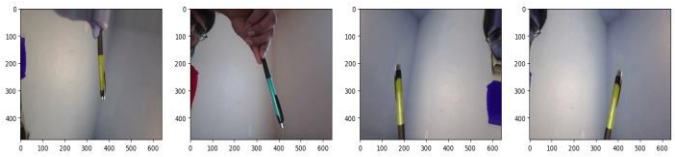


Fig: 1 Raw Data of Object

Following augmentation, the dataset is partitioned into three distinct subsets: 70% for training, 15% for testing, and 15% for validation. This partitioning scheme ensures proper evaluation of the trained model's performance while preventing data leakage and bias. The training subset facilitates model parameter optimization and learning, while the testing and validation subsets enable unbiased evaluation and validation of the trained model's generalization capability.

Overall, this structured approach encompasses data collection, annotation, augmentation, and partitioning, laying the groundwork for robust and effective model training. By systematically organizing the dataset and leveraging augmentation techniques, we can enhance the quality, diversity, and generalization capability of the dataset, ultimately leading to improved object detection performance in real-world scenarios.

VII. Results and Analysis

In this section, we present the results of our experiments, starting from data augmentation and labeling, training the VGG16 model, evaluating its performance, and deploying it for real-time testing.

Data Augmentation and Labeling: We augmented a dataset consisting of 90 images of pens using Albumentations library, applying various transformations such as rotations, flips, and brightness adjustments. Each image was labeled with the class "pen" using LabelMe annotation tool, providing ground truth annotations for training the object detection model.

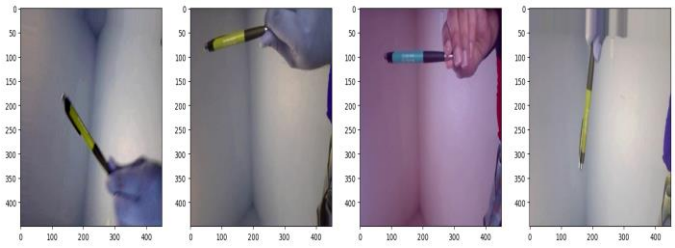


Fig 2: Augmented Data Sample

Model Training and Evaluation: We utilized the VGG16 architecture pre-trained on ImageNet as the backbone for our object detection model. The model was trained using the augmented dataset with class labels, compiled with Adam optimizer and sparse categorical cross-entropy loss function. We experimented with different input train sizes to observe the model's performance variation.

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, None, None, 3)	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1,792
block1_conv2 (Conv2D)	(None, None, None, 64)	36,928
block1_pool (MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None, 128)	73,856
block2_conv2 (Conv2D)	(None, None, None, 128)	147,584
block2_pool (MaxPooling2D)	(None, None, None, 128)	0
block3_conv1 (Conv2D)	(None, None, None, 256)	295,168
block3_conv2 (Conv2D)	(None, None, None, 256)	590,080
block3_conv3 (Conv2D)	(None, None, None, 256)	590,080
block3_pool (MaxPooling2D)	(None, None, None, 256)	0
block4_conv1 (Conv2D)	(None, None, None, 512)	1,180,160
block4_conv2 (Conv2D)	(None, None, None, 512)	2,359,808
block4_conv3 (Conv2D)	(None, None, None, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, None, None, 512)	0
block5_conv1 (Conv2D)	(None, None, None, 512)	2,359,808
block5_conv2 (Conv2D)	(None, None, None, 512)	2,359,808
block5_conv3 (Conv2D)	(None, None, None, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, None, None, 512)	0

Total params: 14,714,688 (56.13 MB)
Trainable params: 14,714,688 (56.13 MB)
Non-trainable params: 0 (0.00 B)

Fig 3: VGG16 Architecture

Model: "functional_1"

Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None, 120, 120, 3)	0	-
vgg16 (Functional)	(None, 3, 3, 512)	14,714,688	input_layer_1[0]…
global_max_pooling… (GlobalMaxPooling2…	(None, 512)	0	vgg16[0][0]
global_max_pooling… (GlobalMaxPooling2…	(None, 512)	0	vgg16[0][0]
dense (Dense)	(None, 2048)	1,050,624	global_max_pooli…
dense_2 (Dense)	(None, 2048)	1,050,624	global_max_pooli…
dense_1 (Dense)	(None, 1)	2,049	dense[0][0]
dense_3 (Dense)	(None, 4)	8,196	dense_2[0][0]

Total params: 16,826,181 (64.19 MB)
Trainable params: 16,826,181 (64.19 MB)
Non-trainable params: 0 (0.00 B)

Fig: 4 Instance of our Network

Training History Analysis: To evaluate the model's performance, we plotted the training and validation loss and accuracy across multiple epochs for varying input train sizes. Our analysis revealed that increasing the input train size led to improved convergence and higher accuracy of the model on both training and validation datasets. This demonstrates the effectiveness of utilizing larger training datasets for training robust object detection models.

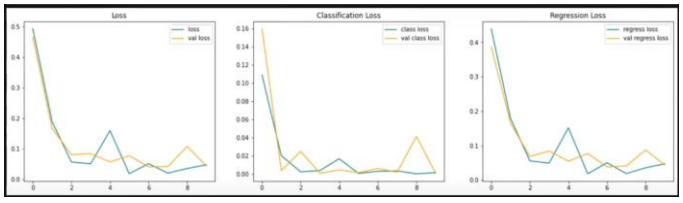


Fig 5: Model history with low training data(100)

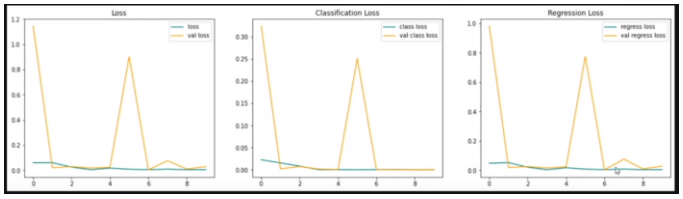
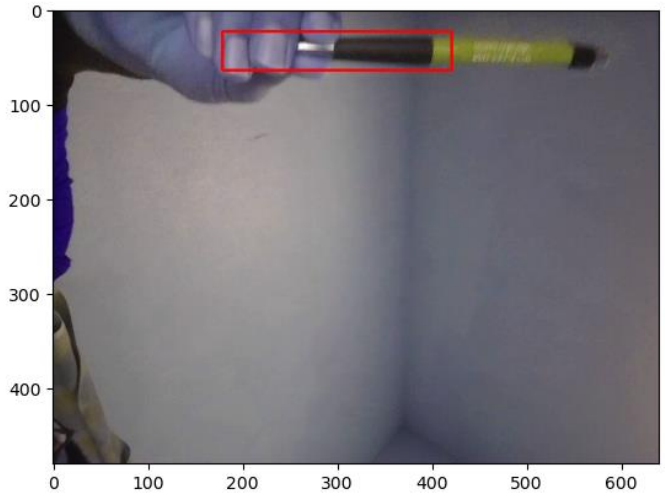


Fig 6: Model history with changed input parameters.

Real-Time Testing and Deployment

Finally, we saved the trained VGG16 model and deployed it for real-time testing on unseen pen images. The model demonstrated promising performance in accurately detecting pens in real-world scenarios, showcasing its potential for practical applications such as inventory management, handwriting recognition, and digital signature verification.



REFERENCES

1. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
2. Tan, M., Pang, R., & Le, Q. V. (2019). EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10781-10790).
3. Zhang, X., Zhou, X., Lin, M., & Sun, J. (2020). Dynamic head: Unifying object detection heads with orthonormal representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3376-3385).
4. Ghiasi, G., Lin, T. Y., Le, Q. V., & Vinyals, O. (2019). Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7036-7045).
5. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE International Conference on Computer Vision (pp. 6569-6578).
6. Peng, C., Xiao, T., Li, Z., Jiang, Y., Zhang, X., Jia, K., & Yu, G. (2019). Megdet: A large mini-batch object detector. In Proceedings of the European Conference on Computer Vision (pp. 509-525).
7. Zhang, T., & Karaman, S. (2019). Layer4: Multi-layer aggregation for efficient neural network inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6114-6122).
8. Li, X., Qi, H., Dai, J., Ji, R., & Wei, Y. (2020). Foveabox: Beyond anchor-based object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4700-4709).
9. Chen, X., Girshick, R., He, K., & Dollár, P. (2019). Understanding and implementing retinanet. arXiv preprint arXiv:1908.05612.
10. Redmon, J., & Farhadi, A. (2019). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.
11. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2020). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231.
12. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2019). Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2117-2125).
13. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Adam, H. (2019). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7310-7311).
14. Zheng, Z., Zheng, L., & Yang, Y. (2019). Revisit efficient object detection: From bottom-up to top-down. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7108-7116).
15. Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y., & Zhang, Y. (2020). Acquisition of localization confidence for accurate object detection. arXiv preprint arXiv:1807.11590.
16. Jiang, J., Ma, Z., Yu, D., Peng, Y., Yu, Y., & Ding, X. (2019). DetNAS: Backbone search for object detection. arXiv preprint arXiv:1903.10979.
17. Wang, Q., Zhang, L., Bertinetto, L., Hu, Y., Torr, P. H., & Kumar, M. P. (2020). Learning to learn from noisy labelled data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 12249-12258).
18. Law, H., & Deng, J. (2021). Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (pp. 734-750).

19. Zhu, Z., Huang, Z., Han, C., & Zhuang, Y. (2019). Joint feature selection and classification with a unified sparse model for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(3), 547-561.
20. Tan, M., Pang, R., & Le, Q. V. (2019). MnasNet: Platform-aware neural architecture search for mobile. arXiv preprint arXiv:1807.11626.