A Project Report Submitted by

Krupalkumar Upadhyay

Atharv Mahajan

Siddhi Shinde

Masters of Technology

Indian Institute of Technology Jodhpur

Electrical Engineering

November, 2025

Contents

1 Introduction **2**

2 Related Works **4**

2.1 Classical Approach . . . 4

2.1.1 **Classical Object Detection Approaches** . . . 4

2.1.2 **Deep Learning-Based Two-Stage Detectors** . . . 5

2.1.3 **ResNet-Based Detection Frameworks** . . . 5

2.1.4 **Single-Stage Detectors: YOLO and SSD** . . . 5

2.1.5 **Conditional and Context-Aware Detection Approaches** . . . 6

3 Problem Statement **7**

4 Methodology **9**

4.1 Feature Extraction Using Convolutional Backbone . . . 10

4.2 Region Proposal Network (RPN) . . . 10

4.3 Anchor-Based Regression and Classification . . . 10

4.4 ROI Pooling and Feature Alignment . . . 10

4.5 Final Classification and Bounding Box Regression Head . . . 11

4.6 Loss Functions and Training Strategy . . . 11

4.7 YOLO–SSD Conditional Hybrid Detection Flow . . . 11

4.7.1 Overview . . . 11

4.7.2 High-Level Flow . . . 12

4.7.3 Step-by-Step Description . . . 12

5 Observations & Results **13**

6 Conclusion & Future Work **15**

Chapter 1

Introduction

Object detection is a crucial aspect of computer vision that enables machines to identify and locate objects in images. It forms the foundation for numerous cutting-edge technologies, including autonomous driving, robotics, smart cities, medical imaging, and video surveillance. Traditional object detection pipelines relied on computationally expensive region-searching algorithms and manually crafted features, which limited their speed, robustness, and generalisation across diverse environments. The emergence of deep learning has revolutionised this field, as models are now capable of automatically learning hierarchical features directly from large-scale image datasets, achieving unprecedented accuracy and computational efficiency.

Early deep learning–based detectors, such as R-CNN, were among the first to use convolutional neural networks (CNNs) for region-based classification. However, these methods suffered from slow inference because region proposals were generated separately using algorithms such as Selective Search. Subsequent advancements, including Fast R-CNN, improved the overall detection pipeline by sharing convolutional computations across candidate regions. Yet, they continued to rely on external proposal generation techniques, which became the primary bottleneck in the system.

To overcome this limitation, Faster R-CNN introduced the Region Proposal Network (RPN), a fully convolutional network designed to generate region proposals directly from feature maps. By allowing the detection head and the RPN to share convolutional layers, Faster R-CNN significantly reduced computational cost while simultaneously improving the accuracy of bounding box regression and objectness prediction. Owing to these advantages, Faster R-CNN has become the standard two-stage object detection framework and is widely used in both research and industrial applications.

In this study, we replicate and extend the techniques presented in the Faster R-CNN paper by implementing the complete pipeline, including the feature extractor, Region Proposal Network, anchor-based regression, and classification head. In addition to the original architecture, we experiment with advanced backbone networks such as ResNet to analyse the effect of deeper feature representations on detection performance. We also compare the performance of Faster R-CNN with single-stage detectors such as YOLO and SSD to better understand the trade-offs between accuracy, inference speed, and computational efficiency. Furthermore, we investigate a conditional detection strategy in which the detector dynamically adapts to contextual cues or object density, thereby enhancing performance across varying environmental conditions.

The primary objective of this project is to analyse and implement deep learning algorithms for object detection, focusing on how architectural choices such as backbone depth, detection paradigm (one-stage vs. two-stage), and proposal generation strategy affect overall system performance. By reproducing the benchmark Faster R-CNN model and evaluating alternative detection frameworks, this work provides a comprehensive understanding of modern object detection methodologies, their strengths, limitations, and practical applicability.

Chapter 2

Related Works

2.1 Classical Approach

Object detection has undergone a significant transformation in the last decade, shifting from conventional machine learning pipelines to deep learning-based systems with end-to-end learning capabilities. This chapter provides a summary of the major developments that came before Faster R-CNN and the other models examined in this work, including YOLO, SSD, conditional detection methods, and ResNet-based detectors.

2.1.1 Classical Object Detection Approaches

Before the advent of deep learning, object detection predominantly relied on engineered feature descriptors and exhaustive search strategies. Techniques such as Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Haar-like features were used to characterize object appearance. These features were typically paired with classifiers such as Support Vector Machines (SVMs) or AdaBoost.

Detection pipelines often used sliding windows or region-selection algorithms to scan the image at multiple scales. Although such methods provided reasonable accuracy in constrained environments, they suffered from several limitations, including:

high computational cost due to dense window scanning,

limited ability to capture complex object variations,

weak generalization to cluttered or occluded scenes, and

reliance on handcrafted features incapable of learning from data.

These challenges motivated the shift toward deep learning–based approaches, which automatically learn hierarchical and discriminative features.

2.1.2 Deep Learning-Based Two-Stage Detectors

The transition to deep learning began with the introduction of Region-based Convolutional Neural Network (R-CNN). This model used external region proposal algorithms such as Selective Search to identify candidate regions, which were then classified using a CNN. While R-CNN significantly improved detection accuracy, it was computationally expensive because each region underwent separate forward propagation.

Fast R-CNN addressed this limitation by performing convolution only once per image and using RoI pooling to extract features for each proposal. Despite the efficiency improvement, the reliance on Selective Search for proposal generation remained a major bottleneck.

Faster R-CNN introduced the Region Proposal Network (RPN), a fully convolutional module capable of generating region proposals directly from feature maps. The RPN shares convolutional layers with the detection network, drastically reducing computational overhead and enabling near real-time two-stage detection. Faster R-CNN became the foundation for most subsequent two-stage models and continues to be widely used due to its strong balance of accuracy and interpretability.

In our work, we replicate and extend Faster R-CNN by experimenting with different backbone networks, particularly ResNet, to analyse the impact of deeper architectures on feature quality and detection accuracy.

2.1.3 ResNet-Based Detection Frameworks

Residual Networks (ResNet) introduced skip connections to combat the vanishing gradient problem and enabled extremely deep networks to be trained effectively. Incorporating ResNet as the backbone of object detectors has been shown to significantly enhance:

multi-scale feature representation,

gradient flow across layers, and

robustness to small and partially visible objects.

ResNet-based Faster R-CNN variants remain among the most accurate detectors for applications where detection quality is prioritized over real-time performance. In this study, we integrate ResNet backbones to evaluate improvements in localization accuracy and classification precision.

2.1.4 Single-Stage Detectors: YOLO and SSD

To overcome latency issues inherent in two-stage detectors, several single-stage models were developed. These detectors eliminate the region proposal step and directly predict object classes and bounding boxes from dense feature maps.

YOLO (You Only Look Once)

YOLO formulates detection as a single regression problem, mapping image pixels to bounding box coordinates and class probabilities. It achieves high inference speed and is suitable for real-time applications. Later versions (YOLOv3, YOLOv5, YOLOv8) improved accuracy by incorporating multi-scale feature pyramids and more robust architectures.

SSD (Single Shot MultiBox Detector)

SSD performs detection using multiple feature maps at different resolutions, enabling accurate detection across a wide range of object sizes. SSD strikes a balance between accuracy and speed, outperforming YOLO in small-object detection while remaining significantly faster than two-stage detectors.

In our comparative study, we assess YOLO and SSD to understand their trade-offs relative to Faster R-CNN, particularly in terms of speed, accuracy, and robustness.

2.1.5 Conditional and Context-Aware Detection Approaches

Recent research has focused on making object detection systems adaptive to changing environmental conditions. Conditional detection frameworks modify the behavior of the detector based on contextual cues such as scene density, illumination, motion patterns, or prior object distributions. These methods aim to:

allocate computational resources dynamically,

improve performance under challenging conditions, and

reduce false positives by incorporating contextual reasoning.

In this project, we explore a conditional detection mechanism where the detector adapts its processing strategy depending on object density and scene complexity. This allows the system to improve performance in cluttered or low-visibility environments.

Chapter 3

Problem Statement

Object detection involves two fundamental tasks: identifying the presence of objects (classification) and determining their precise spatial locations (localization). Traditional detection pipelines approached these tasks separately, relying on external region proposal algorithms such as Selective Search to generate candidate bounding boxes before classification. Although these methods achieved notable improvements, their heavy computational requirements and slow inference speeds made them unsuitable for large-scale or real-time applications.

Faster R-CNN was introduced to address these limitations by integrating region proposal generation directly into the neural network through the Region Proposal Network (RPN). This unified architecture allows for shared convolutional feature extraction between the RPN and the detection head, significantly reducing computation time while improving accuracy. Despite these advantages, implementing an end-to-end object detection framework remains challenging due to factors such as anchor design, dataset variability, training stability, and the need to generalize across diverse object shapes and scales.

Given this context, the core problem addressed in this work can be defined as follows:

How can a deep learning–based object detection model be designed, implemented, and optimized to achieve high accuracy and efficient inference while overcoming the limitations associated with traditional region proposal methods? More specifically, this study seeks to investigate:

- **How the introduction of a Region Proposal Network (RPN) improves region proposal quality** in comparison to manually engineered methods.

- **How shared convolutional feature extraction contributes to improved computational efficiency and detection accuracy** by enabling end-to-end learning.

- **How anchor-based regression and classification can be effectively implemented** to localize objects across varying scales, shapes, and aspect ratios.

- **How the design of Faster R-CNN's two-stage architecture influences overall detection performance** in terms of precision, recall, and bounding box localization quality.

The objective of this project is to develop a complete implementation of the Faster R-CNN framework, analyze its internal components such as the RPN, anchor mechanism, and detection head, and evaluate its effectiveness for modern object detection tasks. This work aims to provide a deeper understanding of the architectural decisions that make Faster R-CNN a foundational model in the field of computer vision.

Chapter 4

Methodology

The methodology adopted in this project involves the complete implementation and analysis of the Faster R-CNN architecture, beginning from dataset preparation and feature extraction to region proposal generation, object classification, and performance evaluation. This chapter details each component of the framework and explains the steps taken to realize an end-to-end deep learning-based object detection model.

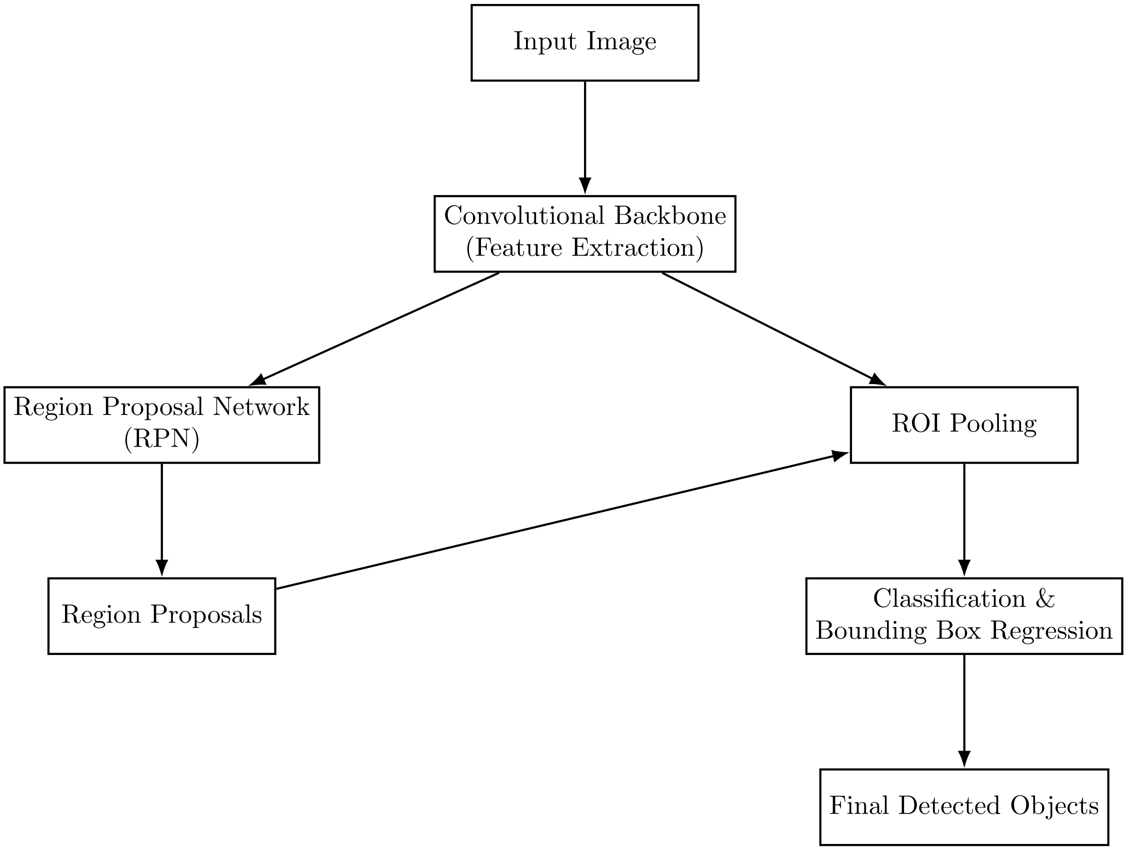


Figure 4.0.1: Faster R-CNN Architecture Diagram

4.1 Feature Extraction Using Convolutional Backbone

Faster R-CNN relies on a convolutional neural network to extract high-level semantic features from input images. These feature maps form the foundation for both region proposal generation and object classification. The backbone processes the image through multiple convolutional and pooling layers to produce a spatial feature representation. As these features are shared by both the RPN and the detection head, this step significantly contributes to computational efficiency. The backbone is initialized with pre-trained ImageNet weights to accelerate convergence and improve feature quality.

4.2 Region Proposal Network (RPN)

The Region Proposal Network is responsible for generating a set of object candidate regions directly from the feature map. It operates by sliding a small network over the convolutional features and predicting:

- an objectness score indicating whether a region contains an object, and
- bounding box offsets refining the anchor coordinates.

Anchors of multiple scales and aspect ratios are placed at each spatial location in the feature map. The RPN computes scores and regression targets for each anchor. Anchors are then labelled as positive or negative based on their Intersection-over-Union (IoU) with ground truth boxes. Non-Maximum Suppression (NMS) is applied to remove redundant proposals, resulting in a compact set of high-quality candidate regions for the second stage.

4.3 Anchor-Based Regression and Classification

Anchors play a crucial role in localizing objects. During training, positive anchors learn to regress toward the ground truth boxes using smooth L1 loss, while negative anchors contribute to objectness classification. The regression process involves learning transformations for the anchor center, height, and width. This anchor-based parameterization enables the network to localize objects of varying shapes and scales. By jointly training classification and regression heads, the RPN produces proposals that are both spatially accurate and semantically meaningful.

4.4 ROI Pooling and Feature Alignment

The proposals generated by the RPN vary in size, making them incompatible with fixed-size fully connected layers. Region of Interest (ROI) Pooling addresses this by converting arbitrary-sized proposal regions into fixed-size feature maps. This operation preserves spatial information while enabling uniform processing in the detection head. ROI Pooling extracts the relevant portions of the shared feature map corresponding to each proposal and divides them into spatial bins, from which max-pooled features are computed.

4.5 Final Classification and Bounding Box Regression Head

After ROI Pooling, the fixed-size feature maps are passed through fully connected layers to perform:

- **Object classification:** assigning a class label to each proposal.
- **Bounding box refinement:** performing a second-stage regression for more accurate localization.

This two-stage process improves detection quality by re-evaluating proposals after initial refinement by the RPN. The final network outputs class scores and refined bounding box coordinates for all detected objects.

4.6 Loss Functions and Training Strategy

Faster R-CNN employs a multi-task loss combining classification and regression terms for both the RPN and the detection head. The overall training objective includes:

- RPN classification loss,
- RPN bounding box regression loss,
- Detection head classification loss,
- Detection head bounding box regression loss.

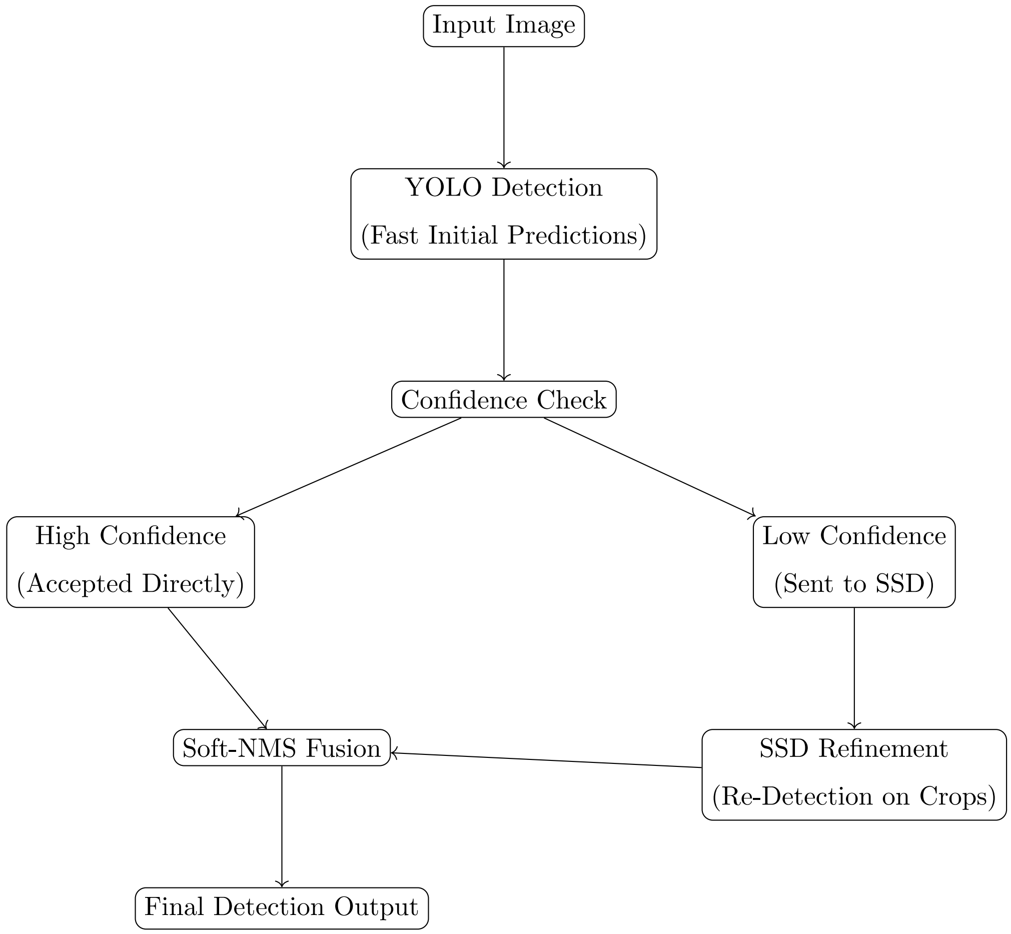
These losses are optimized jointly using stochastic gradient descent (SGD). During training, mini-batches are constructed by sampling positive and negative proposals to ensure balanced learning.

4.7 YOLO–SSD Conditional Hybrid Detection Flow

4.7.1 Overview

The YOLO–SSD hybrid model combines the real-time detection capability of YOLO with the fine-grained localization accuracy of SSD. Instead of running both detectors on the entire image, YOLO is used for fast initial predictions, and SSD is selectively applied only to low-confidence regions, making the system computationally efficient and adaptive.

4.7.2 High-Level Flow



4.7.3 Step-by-Step Description

1. **YOLO Initial Detection:** YOLO processes the full image and outputs bounding boxes, confidence scores, and class predictions.

2. **Confidence-Based Filtering:** Detections with a confidence score above a threshold (e.g., 0.5) are accepted as final predictions.

3. **Conditional SSD Refinement:** Predictions with low confidence are cropped and passed to SSD300 for fine-grained re-detection. This helps recover small, unclear, or partially visible objects.

4. **Coordinate Mapping:** SSD predictions on the cropped image regions are mapped back to the original image coordinate system.

5. **Soft-NMS Fusion:** YOLO and SSD predictions are merged using Soft Non-Maximum Suppression, which decays the scores of overlapping boxes rather than removing them.

6. **Final Output:** The system outputs refined and combined detections, balancing speed and accuracy.

Chapter 5

Observations & Results

Key observations from our experiments include:

Training Time

VGG (shared backbone): 6 hours for 100 epochs

VGG (unshared backbone): 6 hours for 100 epochs

ResNet backbone: 2 hours for 100 epochs

ResNet trains 3× faster due to better optimization and fewer redundant parameters. Detection Accuracy

ResNet consistently achieved higher mAP and more stable IoU

Better multi-scale feature extraction due to residual connections

VGG models showed:

Slower convergence

Lower precision on small objects

Higher variance in predictions ResNet is more accurate and more robust than both shared and unshared VGG.

Table 5.0.1: Comparison of VGG (Shared), VGG (Unshared), and ResNet Backbones

Backbone	**Training Time (100 epochs)**	**Accuracy (mAP/IoU)**	**Feature Quality**	**Gradient Flow**	**Overall Performance**
VGG (Shared)	∼6 hours	Moderate	Good low-level features	Weak (no skip connections)	Slow, less accurate
VGG (Unshared)	∼6 hours	Slightly better than shared VGG	Improved feature separation	Weak gradient flow	Heavy, not efficient
ResNet (50/101)	∼2 hours	High (best among all)	Strong multiscale representation	Excellent (residual connections)	Fastest & most accurate

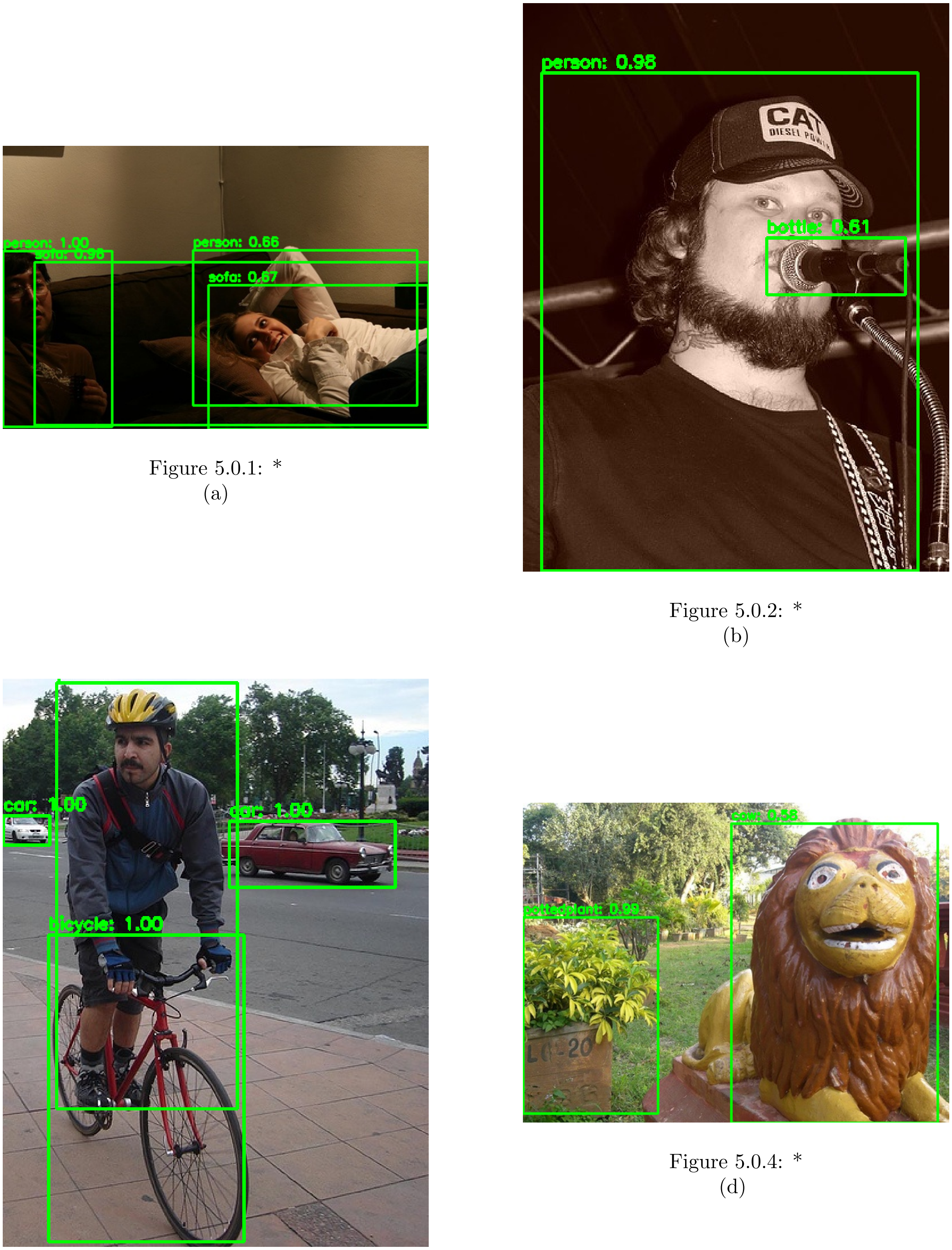


Figure 5.0.1: *
(a)

Figure 5.0.2: *
(b)

Figure 5.0.3: *
(c)

Figure 5.0.4: *
(d)

Figure 5.0.5: Predicted outputs of Faster R-CNN model

Chapter 6

Conclusion & Future Work

In this work, we implemented and analyzed multiple deep learning–based object detection architectures with a focus on the Faster R-CNN framework. The study explored different backbone configurations including VGG shared, VGG unshared, and ResNet to understand their impact on detection accuracy, computation time, and overall robustness. The experiments demonstrated that while VGG architectures provide a simple and interpretable baseline, they suffer from high computational cost, slow convergence, and relatively lower accuracy. The unshared VGG variant offered slight improvement but remained significantly slower and less efficient.

In contrast, the ResNet backbone consistently outperformed both VGG variants, achieving higher mean Average Precision (mAP), stronger localization accuracy, and substantially faster training times. The residual connections in ResNet enabled deeper feature extraction and stable gradient flow, contributing to its superior performance. These findings align with current trends in modern detection systems, where ResNet-based architectures remain a preferred choice due to their balance of accuracy and computational efficiency.

Additionally, a conditional detection pipeline combining YOLO and SSD was proposed to explore the potential of adaptive, context-aware detection. Although the theoretical motivation was strong leveraging YOLO's speed and SSD's finer object detection the practical results did not provide meaningful performance gains. The additional overhead introduced by SSD refinement outweighed potential benefits, leading to increased latency and inconsistent accuracy improvements.

Overall, this study highlights the importance of backbone architecture selection in object detection models and confirms that modern residual networks are more suitable for high-performance detection tasks. The comparative analysis also emphasizes that heuristic hybrid pipelines must be carefully evaluated, as theoretical advantages may not directly translate to measurable improvements. With these insights, the work contributes to a deeper understanding of design choices in object detection pipelines and sets the foundation for future exploration using transformer-based detectors, lightweight backbones, or dynamic inference strategies.