Vision-based Obstacle Detection and Depth Estimation for Unmanned Ground Vehicle

A thesis submitted in the partial fulfillment of
the requirements for the degree of

MS by Research

by

Sujeet Kumar Singh
Roll no: 214363009

Under the Guidance of

Prof. S.K. Dwivedy & Dr. Prithwijit Guha

June 2023

E Mobility

Department of Electronics and Electrical & Mechanical Engineering

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

Declaration

I hereby affirm that the thesis entitled "Vision-based Obstacles Detection and Depth Estimation for Unmanned ground vehicle," submitted in partial fulfillment of the requirements for the degree of MS by research in E Mobility at the Indian Institute of Technology Guwahati, is an authentic representation of my own work. It has been conducted under the guidance of **Prof. S.K. Dwivedy & Dr. Prithwijit Guha**, *and proper acknowledgment is given to the works of other researchers, which are duly referenced in the reference section. The contents of this thesis, whether in full or in part, have not been previously submitted to any other academic institution or university for the purpose of obtaining any degree or diploma.*

Sujeet Kumar Singh

E Mobility

IIT Guwahati, India., Assam-781039

June 2023

CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**Vision-based Obstacles Detection and Depth Estimation for Unmanned Ground Vehicle**" is a bonafide work of **Sujeet Kumar Singh** (**Roll No. 214363009**), E Mobility, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Prof.S.K Dwivedy**

& Dr. Prithwijit Guha

Professors,

June, 2023

Department of Mechanical Engineering & EEE

Guwahati.

Indian Institute of Technology Guwahati, Assam.

Acknowledgements

Firstly, I owe my heartfelt gratitude and deepest regard to my supervisor, Professor Santosh Kumar Dwivedy & Dr. Prithwijit Guha, Department of Mechanical and Electronics and Electrical Engineering respectively, IIT Guwahati, India, for his relentless guidance, constant support and full cooperation throughout the project. His methods, vision and enthusiasm has always been a constant motivation and inspiration to me. Without his supervision and direction, this project could not have been a success.

My deepest acknowledgment goes towards my family members who have been supportive right from the beginning of my master's journey. I would fail in my duty if I do not highlight here the name of Mr. Madhusudan (PhD, EEE) Mr. Yogesh Aggarwal (PhD, EEE),Mr. Vineet Kumar (PhD,EEE) and Mr. Sahbaaz Ahamd (PhD, EEE),Mr. Koushik Mazumder (PhD, EEE) for their great support without which this outcome would not have been possible.

Further, the Department of Science and Tech (DST), Government of India through their scholarship scheme to master's students has helped me with the basic financial requirements for which I am thankful.

Sujeet Kumar Singh, 214363009

E Mobility

IIT Guwahati, India., Assam-781039, June 2023

Contents

List of Figures

Abstract

Vision-based obstacle detection plays a crucial role in ensuring the safe navigation of unmanned ground vehicles (UGVs) in various environments. This thesis presents a comprehensive study on the utilization of the YOLOv5 model for detecting obstacles, along with the incorporation of the Feature Pyramid Network (FPN) using the MobileNet V2 backbone for depth estimation. The proposed approach involves leveraging the bounding box coordinates obtained from YOLOv5 and mapping them to the corresponding coordinates on the depth map generated by the FPN network. By averaging the pixel values within the bounding box on the depth map, the depth of the detected obstacle is determined. The system further provides visual feedback by displaying the obstacle name and its corresponding depth on custom classes.

This thesis aims to investigate the effectiveness and performance of the YOLOv5 model combined with the FPN-based depth estimation approach for obstacle detection in UGVs. The research involves extensive experimentation on various datasets to evaluate the accuracy and robustness of the proposed methodology. Additionally, comparisons are made with existing state-of-the-art approaches to assess the advancements achieved.

This research contributes to the field of vision-based obstacle detection for UGVs by providing insights into the performance of the YOLOv5 model and the effectiveness of the FPN-based depth estimation technique. The findings from this thesis offer valuable knowledge for improving the safety and reliability of autonomous UGVs in complex environments.

Chapter 1

Introduction

Unmanned ground vehicles (UGVs) have emerged as a promising technology in various domains, including surveillance, agriculture, transportation, and disaster response. These autonomous vehicles rely on sophisticated perception systems to navigate through complex environments safely. One critical aspect of UGV perception is vision-based obstacle detection, which plays a pivotal role in ensuring the vehicle's ability to detect and avoid obstacles in real-time. Accurate and efficient obstacle detection is crucial for the reliable operation of UGVs and preventing potential collisions or accidents.

In recent years, significant progress has been made in computer vision and deep learning techniques, enabling the development of advanced models for object detection. Among these models, the YOLO (You Only Look Once) series has gained popularity due to its ability to provide real-time object detection with impressive accuracy. YOLOv5, in particular, has demonstrated promising performance in various object detection tasks, making it a suitable candidate for vision-based obstacle detection in UGVs.

However, obstacle detection alone is often not sufficient for safe navigation. Depth estimation, which provides information about the distance to detected obstacles, is a critical component in determining the proximity and potential risks associated with obstacles. Depth estimation can enable UGVs to make informed decisions about their navigation

paths and adjust their movements accordingly. Feature Pyramid Networks (FPNs) have shown great potential in generating high-resolution depth maps, and combining them with object detection models can enhance the UGV's perception capabilities.

In this thesis, we aim to investigate the effectiveness and performance of utilizing the YOLOv5 model for obstacle detection in UGVs, coupled with the incorporation of FPNs for depth estimation. By leveraging the bounding box coordinates obtained from YOLOv5, we can map them to the corresponding coordinates on the depth map generated by the FPN network. Through this integration, we can determine the depth of the detected obstacles, enabling the UGV to perceive their proximity accurately.

The main objectives of this thesis are as follows:

- Explore the capabilities of the YOLOv5 model for obstacle detection in UGVs and assess its accuracy and efficiency.
- Investigate the effectiveness of FPNs with the MobileNet V2 and others backbone for depth estimation in the context of UGV obstacle detection.
- Develop an integrated system that combines YOLOv5 and FPN-based depth estimation for vision-based obstacle detection in UGVs.
- Evaluate the performance of the proposed system through extensive experimentation on various datasets and compare it with existing state-of-the-art approaches.
- Provide insights and recommendations for future enhancements in vision-based obstacle detection and depth estimation for UGVs.

By addressing these objectives, this thesis aims to contribute to the advancement of vision-based obstacle detection techniques for UGVs. The proposed methodology has the potential to enhance the safety and reliability of UGVs operating in dynamic and challenging environments. Furthermore, the research findings and recommendations will fa-

cilitate future research and development efforts in this domain, fostering advancements in autonomous vehicle technologies.

1.1 Challenges in Vision-Based Obstacle Detection and Depth Map Estimation

Developing vision-based obstacle detection and depth map estimation systems for unmanned ground vehicles (UGVs) involves overcoming several challenges. The following challenges are particularly relevant to this topic:

Ambiguity in Depth Estimation: Estimating accurate depth information from a single image or stereo image pair is a challenging task due to inherent ambiguities. Multiple depth configurations can produce similar image appearances, making it difficult to obtain precise depth estimates without additional information or assumptions.

Occlusion and Cluttered Environments: Real-world environments often contain occlusions and clutter, where obstacles can be partially obscured or surrounded by complex backgrounds. These factors can hinder accurate obstacle detection and depth estimation, as objects may overlap or intersect, leading to incorrect detection and depth estimation.

Limited Field of View: UGVs typically have a limited field of view, which means that obstacles outside the field of view may not be detected or accurately estimated. This limitation poses challenges in scenarios where obstacles dynamically appear or move rapidly, requiring timely detection and response.

Handling Varied Lighting Conditions: Lighting variations across different environments and conditions can significantly impact the performance of vision-based systems. Shadows, reflections, and uneven lighting can introduce noise and distortions in the captured images, affecting the accuracy of obstacle detection and depth estimation.

Resolution and Accuracy of Depth Maps: Generating accurate and high-resolution depth maps is crucial for precise obstacle localization and distance estimation. However,

depth estimation from monocular or stereo vision systems can be prone to errors, especially in scenarios with textureless or featureless regions, occlusions, or reflective surfaces.

Real-Time Processing Requirements: Vision-based obstacle detection and depth estimation systems for UGVs often operate in real-time scenarios where timely processing is critical. Achieving real-time performance while maintaining high accuracy and reliability can be challenging due to the computational complexity of deep learning models and the limited processing power of onboard UGV systems.

Generalization to Diverse Environments: Vision-based systems need to generalize well across different environmental conditions, terrains, and object types. Models trained on specific datasets may struggle to perform optimally in unfamiliar or challenging environments, highlighting the need for robustness and adaptability.

Dataset Diversity and Annotation: Building diverse and representative datasets for training and evaluation is essential for the effectiveness of vision-based systems. Acquiring and annotating datasets that cover various obstacle types, sizes, shapes, and environmental conditions can be time-consuming, costly, and labor-intensive.

Addressing these challenges requires ongoing research and development efforts to enhance the robustness, accuracy, and efficiency of vision-based obstacle detection and depth estimation techniques. Overcoming these challenges will contribute to safer and more reliable autonomous navigation for UGVs in diverse real-world scenarios.

1.2 Major Contributions

This thesis presents significant contributions to the field of vision-based obstacle detection and depth map estimation for unmanned ground vehicles (UGVs). The major contributions of this work are as follows:

- **YOLOv5 Model Training and Customization** – The YOLOv5 model was trained on the widely used COCO dataset, which consists of 80 different object classes. Ad-

ditionally, custom training was performed to incorporate specific classes relevant to the UGV domain. This customization ensures that the model can accurately detect and classify obstacles encountered by UGVs in real-world scenarios.

- **Depth Estimation using Feature Pyramid Network (FPN)** – To estimate the depth of detected obstacles, a depth estimation network was developed using the powerful Feature Pyramid Network (FPN) architecture. The network was trained on the NYU v2 dataset, which provides a diverse range of depth information in indoor scenes. Different backbones, including MobileNetV2, ResNet50, ResNet101, and ResNet152, were explored to compare accuracy and computational trade-offs for depth estimation.

- **Customized Loss Functions for Depth Estimation** – In order to enhance the performance of the depth estimation network, customized loss functions were implemented. Specifically, a combination of L1 and L2 loss, gradient loss, and normal loss with different weightage. This approach helped to improve the accuracy and robustness of the depth maps generated by the network.

- **Andriod Application** To facilitate the real-life deployment of the vision-based obstacle detection and depth estimation system, an Android application was developed using PyTorch Mobile. The application enables the integration of the trained model into mobile devices, allowing for on-device inference and real-time processing

- **Comparative Analysis of Backbones for Depth Estimation** – In order to assess the impact of different backbone architectures on depth estimation performance, a comprehensive comparative analysis was conducted. The performance metrics, such as accuracy and computational efficiency, were evaluated for the MobileNetV2, ResNet50, ResNet101, and ResNet152 backbones. This analysis provides valuable insights into selecting the most suitable backbone for achieving accurate and efficient depth estimation in UGV applications.

These contributions collectively contribute to the advancement of vision-based obstacle detection and depth map estimation for UGVs. The customized training and adaptation of the YOLOv5 model, coupled with the development of a robust depth estimation network using FPN and customized loss functions, enable more accurate and reliable perception capabilities for UGVs in complex and dynamic environments. The insights gained from the comparative analysis of different backbone architectures further contribute to the development of optimized and efficient depth estimation solutions for UGV applications.

1.3 Thesis organization

- **Chapter 2: Literature Review** provides a comprehensive review of the existing literature and research related to vision-based obstacle detection and depth map estimation for unmanned ground vehicles (UGVs). This chapter explores the various techniques, methodologies, and advancements in the field, highlighting the gaps and research opportunities.

- **Chapter 3: Methodology and proposed work** presents the experimental setup, methodology, and results obtained from the conducted research. This chapter discusses the implementation details of training the YOLOv5 model on the COCO dataset, the customization for UGV-specific classes, and the training of the depth estimation network using different backbones. The obtained results are analyzed, and the performance metrics, such as accuracy and computational trade-offs, are evaluated.

- **Chapter 4: Results and Discussion** presents the quantitative and qualitative results obtained from the conducted research. This chapter provides a detailed analysis and interpretation of the results, focusing on the performance metrics, accuracy, computational trade-offs, and other relevant evaluation measures. The chapter also includes visualizations, such as sample detection outputs and depth maps, to illustrate

the effectiveness of the proposed approach.

- **Chapter 5: Conclusion and Future Scope** summarizes the findings of the research and provides a comprehensive conclusion. This chapter discusses the contributions made, highlights the strengths and limitations of the proposed approach, and suggests potential areas for future research and improvement. The chapter also outlines the scope for applying the developed vision-based obstacle detection and depth map estimation techniques to real-world UGV applications.

Chapter 2

Literature Review

Vision-based obstacle detection and depth estimation are crucial tasks in various applications such as robotics, autonomous vehicles, and augmented reality. In recent years, significant advancements have been made in these areas, leveraging computer vision techniques and deep learning approaches. This section provides an overview of the relevant literature and discusses the key findings and contributions in the field.

2.1 Vision-based obstacle detection

Object detection is a crucial task in computer vision, and numerous state-of-the-art approaches have been proposed to tackle this challenge. In this section, we provide a review of some prominent methods, including sliding window [19], region proposals [7], YOLO [25], and SSD [18].

The sliding window approach, introduced by Lowe [19], involves scanning an image at various scales and positions using a fixed-size window. Features are extracted from each window, and a classifier is applied to determine the presence of an object. While effective, this approach can be computationally expensive due to the large number of windows that need to be evaluated.

Region proposal methods aim to address the computational burden by generating a

small set of candidate regions likely to contain objects. One notable example is the selective search method proposed by Girshick et al.[7], which employs hierarchical segmentation to group image regions based on similarity. The resulting regions serve as proposals for object detection, significantly reducing the number of regions to evaluate. Other techniques, such as region proposal networks, have also been developed to generate region proposals efficiently.

The YOLO (You Only Look Once) framework, introduced by Wong and Yu [25], revolutionized object detection by proposing a unified approach that simultaneously predicts bounding box coordinates and class probabilities in a single pass. YOLO divides the input image into a grid and assigns each cell the responsibility of detecting objects. Each cell predicts a fixed number of bounding boxes and their associated class probabilities. This architecture achieves real-time performance while maintaining competitive accuracy. YOLO has seen several iterations, with YOLOv5 being one of the most recent versions.

2.2 YOLO Models

There are several types of the YOLO model, each introducing improvements and advancements in object detection. Some of the notable YOLO models include:

1. YOLOv1: The first YOLO model introduced the concept of grid-based object detection with real-time performance.

2. YOLOv2 (YOLO9000): An enhanced version that incorporated anchor boxes, multi-scale training, and introduced better performance on smaller objects.

3. YOLOv3: Building upon YOLOv2, this model introduced additional improvements, including feature extraction at different scales and the use of skip connections.

4. YOLOv4: A further improvement with a focus on accuracy, YOLOv4 introduced advanced techniques such as CSPDarknet53 as the backbone network, PANet for

feature fusion, and various data augmentation strategies.

5. YOLOv5: The latest iteration of the YOLO model, YOLOv5 focuses on a streamlined architecture, improved performance, and easy deployment. It introduces a more modular design, allowing for easy customization and adaptability.

SSD (Single Shot MultiBox Detector) [18] is another popular approach that integrates object detection and localization into a single network. SSD utilizes a set of convolutional feature maps with different resolutions to detect objects at various scales. By predicting multiple bounding boxes per feature map location, SSD achieves high detection accuracy across a wide range of object sizes. Additionally, SSD incorporates default anchor boxes to handle object deformations and aspect ratio variations.

These state-of-the-art methods have significantly advanced object detection capabilities, offering improvements in accuracy, efficiency, and real-time performance. However, challenges still exist, such as accurately detecting small objects, handling occlusions, and addressing class imbalance in the training data.

2.3 Depth Estimation Techniques

Depth estimation is a fundamental task for understanding the 3D structure of a scene. Several techniques have been explored, including stereo vision [22], structured light [26], and time-of-flight methods [23].

Stereo vision utilizes the disparity between two images captured from different viewpoints to estimate depth. Structured light projects patterns onto the scene and uses the deformation of the patterns to infer depth. Time-of-flight methods measure the time it takes for light to travel from the sensor to the scene and back to estimate depth.

Each technique has its own underlying principles and limitations. Stereo vision requires accurate correspondence matching and is sensitive to occlusions and textureless regions.

Structured light is limited by the range and resolution of the projected patterns. Time-of-flight methods suffer from noise and interference, affecting depth accuracy.

2.3.1 Deep Learning-Based Depth Estimation

Deep learning has also shown promise in-depth estimation. Single-image depth prediction networks [**?**] leverage CNN architectures to directly estimate depth from a single input image. Stereo matching networks [**?**] use CNNs to match corresponding image patches from stereo image pairs, enabling depth estimation.

These deep learning-based approaches have demonstrated impressive results in depth estimation tasks. They can capture intricate features and learn complex mappings between images and depth. However, challenges such as handling ambiguous depth cues, generalizing to different environments, and achieving high-resolution depth maps still persist.

2.4 Classical Methods

When it comes to depth estimation, classical methods, also known as traditional approaches, have long been employed. These methods rely on conventional computer vision techniques and mathematical models to estimate depth information from images [20]. By extracting visual features from input images and leveraging principles of geometry or stereo-matching algorithms, classical methods aim to calculate depth.

One popular classical method is stereo matching [**?**]. This approach utilizes pairs of stereo images to infer depth information. By analyzing the disparity, which refers to the differences in pixel coordinates between corresponding points in the left and right images, stereo matching calculates the depth of objects in the scene. In stereo vision, two cameras at different orientations capture the scene, as shown in Figure 1. The disparity is then calculated on a pixel-wise basis, and depth can be estimated between neighboring pixels using the depth-disparity equation:

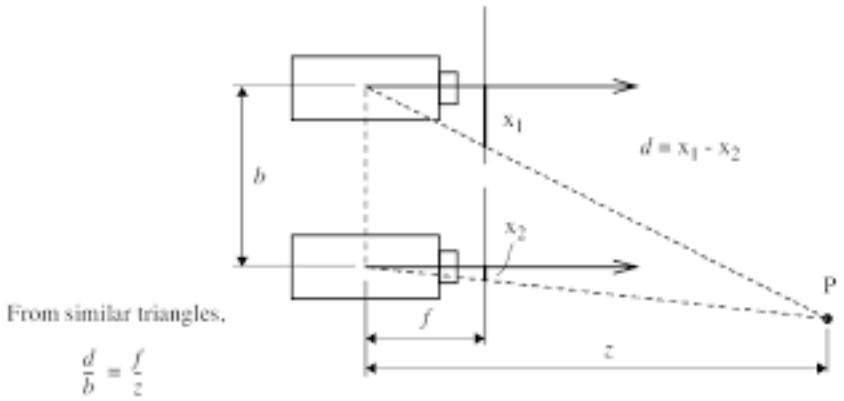

Fig. 2.1 Depth Estimation in Stereo Vision [21]

$$d = \frac{fb}{z}$$

where: d represents the depth, f denotes the focal length, b corresponds to the baseline distance, and (z) represents the disparity.

Another classical method used for depth estimation is structure from motion (SfM). SfM is a computer vision technique that aims to reconstruct the 3D structure of a scene from a collection of 2D images or video frames [2]. This process involves estimating camera poses, camera positions, and the 3D coordinates of scene points.

Firstly, the relative camera poses are estimated by matching feature correspondences between pairs of images. This step, often referred to as relative pose estimation, determines the positions and orientations of the cameras in relation to each other [**?**]. Once the relative camera poses are obtained, the next step is to perform triangulation. By intersecting the corresponding rays from two or more cameras, the 3D positions of the scene points can be calculated [13].

The output of the SfM process is a dense 3D reconstruction of the scene, which consists of a point cloud representing the 3D positions of the scene points and camera poses that capture the viewpoints of the images. This reconstructed 3D model can be further

processed for applications such as 3D visualization, virtual reality, augmented reality, or object tracking.

Classical methods offer well-established techniques for depth estimation, leveraging principles of stereo matching and structure from motion. While these methods have been widely used and have provided valuable insights, they also have limitations. In recent years, deep learning-based methods have shown promising results and have started to outperform classical approaches in certain scenarios. Nonetheless, classical methods continue to be relevant and serve as a foundation for depth estimation research and applications.

2.5 Deep Learning methods

2.5.0.1 Multi-scale deep neural network, 2014

In 2014, Eigen et al. [4] introduced a multi-scale deep neural network for predicting depth from monocular images. This work was one of the earliest successful attempts to utilize deep neural networks for depth estimation. The study highlighted the significance of integrating information from multiple scales and incorporating the global context of the image to effectively leverage monocular cues.

The proposed approach consists of two networks: a coarse network and a fine network. The coarse network takes the entire image as input and predicts the depth of the scene at a global level. This initial prediction provides an estimate of the overall depth structure of the scene. The output of the coarse network, along with the original image, is then fed into the fine network.

The role of the fine network is to refine the depth prediction by leveraging local information within the image. By considering detailed local features and incorporating them into the estimation process, the fine network further improves the accuracy and precision of the depth estimation. This integration of global and local information allows the network to capture both the broad context and fine-grained details of the scene, leading to more

accurate depth predictions.

The multi-scale deep neural network proposed by Eigen et al. achieved state-of-the-art results on error metrics for both the NYU-Depth V2 dataset and the KITTI dataset. By effectively combining global and local information, the network demonstrated superior performance in depth estimation tasks compared to previous approaches.

This work demonstrated the potential of deep neural networks in extracting depth information from monocular images. The integration of multi-scale information and the consideration of global context proved to be crucial in utilizing monocular cues effectively. Since then, deep learning-based methods for depth estimation have made significant progress, building upon the insights provided by this pioneering study.

2.5.0.2 Embedding of focal length

In 2018, He et al. [12] proposed a novel approach for depth estimation in monocular images by incorporating information about the camera's focal length into a deep convolutional neural network (CNN). This unique integration of focal length information aimed to improve the accuracy and precision of depth predictions.

The approach utilizes an encoder-decoder structure based on the VGG architecture, which is initially initialized with pre-trained weights. The network takes a monocular image as input and aims to assign a depth value to each pixel. The encoder part of the network extracts hierarchical features from the input image, capturing both low-level and high-level information. The decoder part of the network performs upsampling and recovers the spatial resolution, generating a depth map as the output.

What sets this approach apart is the embedding of focal length information in the network's final fully connected layers. The focal length, which is a camera parameter representing the distance between the lens and the image sensor, is a crucial factor in depth estimation. By incorporating this information into the network architecture, the model becomes more aware of the scale and distance relationships within the scene, leading

to improved depth predictions.

To train the network, He et al. employed the BerHu loss function, which combines the advantages of both L1 and L2 losses. The BerHu loss function handles outliers robustly by utilizing a piecewise-linear function that approximates the Huber loss. This loss function allows the network to effectively balance the accuracy of depth estimation for different types of scenes and handle varying levels of noise in the input images.

By integrating focal length information and utilizing the BerHu loss function, He et al. achieved notable improvements in depth estimation accuracy. The embedded focal length enables the network to better capture the depth variations and relationships in the scene, resulting in more accurate depth predictions.

The proposed method by He et al. represents an innovative approach that highlights the importance of incorporating camera parameters, such as focal length, into deep neural networks for depth estimation. This work contributes to the advancement of monocular depth estimation techniques, offering insights into the potential benefits of leveraging camera-specific information within the network architecture.

2.5.0.3 Unsupervised monocular depth estimation with left-right consistency, 2016

In 2016, Godard et al. [8] proposed an unsupervised monocular depth estimation approach that leverages the concept of left-right consistency to improve depth estimates. The authors demonstrated that relying solely on reconstruction loss leads to good picture reconstruction but often yields inaccurate depth predictions.

The key idea behind their approach is to utilize epipolar geometry constraints and information from the left image to generate both left-to-right and right-to-left disparity maps simultaneously. By enforcing consistency between these disparity maps, the network can refine the depth estimates and produce more accurate depth maps.

The network architecture consists of an encoder-decoder structure, where the encoder

extracts multi-scale features from the left image. These features are then fed into two decoders: one for generating the left-to-right disparity map and the other for the right-to-left disparity map. The decoder modules produce disparity maps by upsampling the features and applying convolutional operations. The left-right consistency is enforced by warping the right image based on the left-to-right disparity map and comparing it with the original right image. This process ensures that the reconstructed right image aligns well with the actual right image.

The training objective includes two components: the reconstruction loss and the left-right disparity consistency loss. The reconstruction loss encourages the generated images to closely resemble the original images, while the left-right consistency loss enforces consistency between the disparity maps. By jointly optimizing these two losses, the network learns to refine the depth estimates and improve the overall accuracy of depth prediction.

The unsupervised nature of this approach is particularly advantageous since it eliminates the need for ground truth depth data during training. Instead, the network learns to estimate depth solely from the available monocular images and the left-right consistency constraint. This makes the approach more applicable to real-world scenarios where acquiring accurate depth annotations can be challenging or impractical.

The work by Godard et al. demonstrates the importance of incorporating left-right consistency in unsupervised monocular depth estimation. By leveraging the epipolar geometry constraints and enforcing consistency between left-to-right and right-to-left disparity maps, the network can refine depth estimates and generate more accurate depth maps. This approach contributes to the advancement of unsupervised depth estimation techniques and offers a valuable alternative to traditional supervised methods.

2.5.1 Deep Ordinal Regression Network for Monocular Depth Estimation

The paper[5] introduced a Deep Ordinal Regression Network for Monocular Depth Estimation. Ordinal regression[11] [10] is a technique employed for estimating the depth of

RGB images. It involves predicting depth values as discrete ordinal categories rather than continuous values. This approach is particularly valuable when the ground truth depth labels are provided in a discrete form, such as depth bins or depth ranges. By treating depth estimation as an ordinal regression problem, the network can learn to understand the relative order of depth values and generate more precise predictions.

Model Architecture

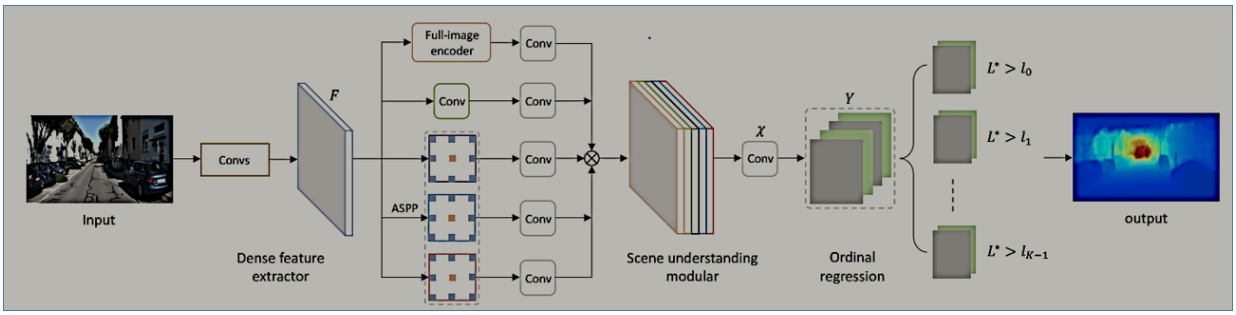

Fig. 2.2 : Model Architecture [5]

As depicted in Figure 2.2, the developed network comprises of two main components: a dense feature extractor and a scene understanding module. This network generates multi-channel dense ordinal labels when presented with an image.

The network comprises several components, including a dense feature extractor, a multi-scale feature learner (ASPP), a cross-channel information learner (implemented through a 1×1 convolutional branch), a full-image encoder, and an ordinal regression optimizer. The Conv components in the architecture employ a kernel size of 1×1. The ASPP module consists of three dilated convolutional layers with kernel size 3×3 and dilated rates of 6, 12, and 18, respectively. The network utilizes supervised information in the form of discrete depth values obtained through the discretization process using the SID strategy. The entire network is optimized using our ordinal regression training loss, allowing for end-to-end training.

2.5.2 High Quality Monocular Depth Estimation via Transfer Learning

This paper [1] discusses about the high quality monocular depth estimation via transfer learning.

Model Architecture:

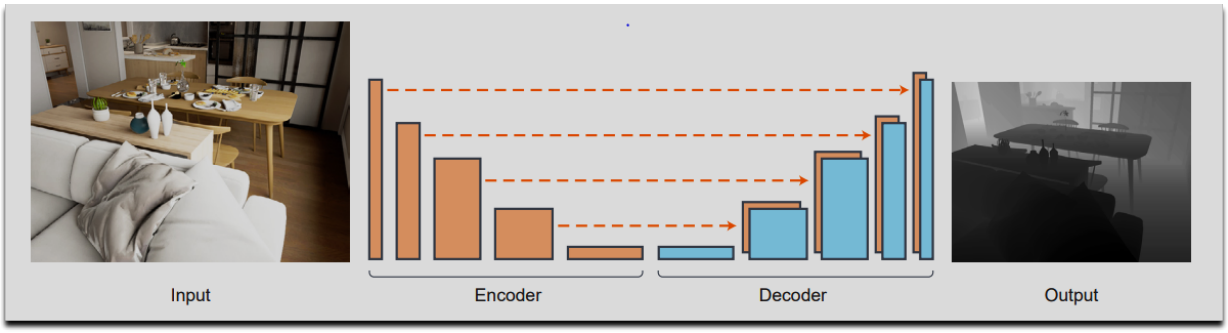

Fig. 2.3 Model Architecture In their work.[1]:the authors propose an encoder-decoder architecture with skip connections for high-resolution depth estimation. The encoder section of the network is based on a pre-trained truncated DenseNet-169, which is used without any further modifications. The decoder section consists of basic blocks of convolutional layers, which are applied to the concatenation of the previous block and the corresponding block from the encoder. The feature maps are upsampled by a factor of 2 using bilinear upsampling, ensuring that the spatial size remains consistent. The skip connections enable the flow of information from the encoder to the decoder, facilitating accurate depth estimation

The architecture that motivates us in our work incorporates the Feature Pyramid Network (FPN) with ResNet101 and other backbone networks as the backbone.

2.5.3 Challenges in estimating depth estimation and obstacle detection by deep learning technique

Estimating Depth and Detecting objects in different scales is a challenging task, especially for small objects. One approach to address this challenge is by using a pyramid of the same image at different scales to detect objects. However, processing multiple scale images can be time-consuming and memory-intensive, making it impractical to train them end-to-end simultaneously. Hence, this approach is typically used during inference to maximize

accuracy, particularly in competitions where speed is not a concern. Another alternative is to create a pyramid of features and utilize them for object detection. However, feature maps closer to the image layer often consist of low-level structures that are not effective for accurate object detection.

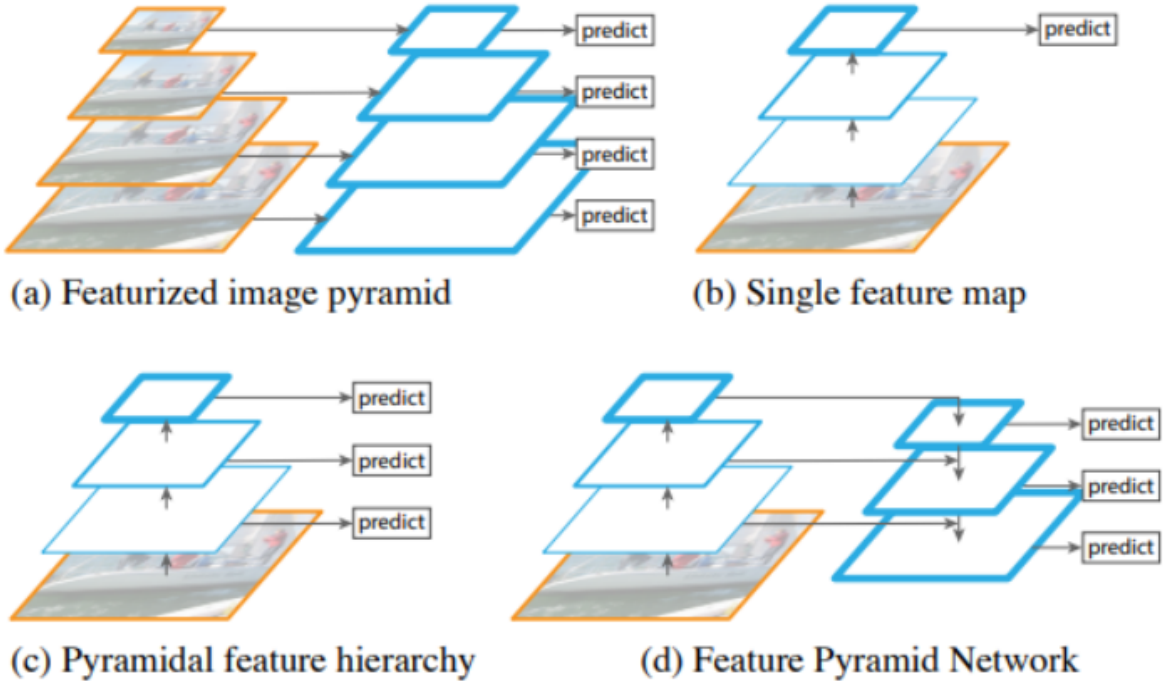

Fig. 2.4 In their work.[17]:the authors propose different feature pyramid network architecture

The Feature Pyramid Network (FPN) is designed to tackle these issues by providing a feature extractor specifically tailored for the pyramid concept with a focus on accuracy and speed. FPN replaces the feature extractor of detectors like Faster R-CNN and generates multiple feature map layers, known as multi-scale feature maps, which contain richer and higher-quality information compared to regular feature pyramids for object detection.

FPN comprises a bottom-up pathway and a top-down pathway. The bottom-up pathway functions as a conventional convolutional network for feature extraction. As we move up the pathway, the spatial resolution gradually decreases while the semantic value for each layer increases, thanks to the detection of more high-level structures.

To overcome the loss of spatial precision caused by downsampling and upsampling in the top-down pathway, FPN incorporates lateral connections between the reconstructed layers and the corresponding feature maps. These lateral connections enhance the detector's ability to predict object locations more accurately and serve as skip connections to facilitate training, similar to the role of skip connections in ResNet.

Chapter 3

Methodology and Proposed Work

3.1 Estimation of Depth Map by transfer learning

3.1.1 Introduction

The proposed approach heavily relies on the concept of transfer learning. Depth map estimation plays a crucial role in various computer vision applications, such as robotics, autonomous driving, and augmented reality. Deep learning has shown significant success in this field, and our goal is to further enhance the accuracy and robustness of depth map estimation.

To achieve this, we introduce a novel feature pyramid network (FPN) architecture that leverages the power of transfer learning. By incorporating different backbone networks, including MobileNetV2, ResNet-50, ResNet-101, and ResNet-152, we aim to exploit the diverse feature extraction capabilities of these networks. The pre-trained weights of these backbone networks provide a valuable source of knowledge learned from large-scale image classification tasks, which can be adapted to the depth estimation task.

The FPN architecture enables us to extract multi-scale features from input images, capturing both fine-grained details and global contextual information. Through the use of top-down and lateral connections, feature maps at different levels of spatial resolution

are combined, allowing for a comprehensive understanding of depth-related information. This hierarchical representation enhances the network's ability to handle depth estimation across various scales and complexities.

To evaluate the effectiveness of our proposed approach, we conduct extensive experiments on well-established benchmark datasets such as NYU-Depth V2. Our results demonstrate that the incorporation of transfer learning and the utilization of different backbone networks significantly improve the accuracy and generalization capabilities of depth map estimation.

Our proposed approach harnesses the power of transfer learning and feature pyramid networks to advance the field of depth map estimation. By leveraging pre-trained weights from diverse backbone networks, we aim to achieve state-of-the-art performance in terms of accuracy and robustness. This research contributes to the development of more reliable depth estimation techniques, enabling advancements in various computer vision applications.

3.1.2 Training Dataset

The dataset used for training in this work is the **NYU Depth v2** dataset [24]. It consists of a large collection of indoor scene images along with corresponding depth maps. The dataset contains 120,000 training samples and 654 testing samples. However, for this work, a subset of 50,000 samples was utilized.

The NYU Depth v2 dataset provides images and densely labeled depth maps at a resolution of 640 x 480 pixels. It is important to note that out of the available samples, only 1,449 pairs of images and depths have complete labeling. To address this limitation, the missing depth values in the dataset were filled using an inpainting method proposed by Levin et al. [16].

During training, the network is trained to predict depth maps at the same resolution as the input images, i.e., 640 x 480 pixels. This ensures consistency in the spatial dimensions

of the predicted depth maps, allowing for accurate depth estimation.

Including additional information or details based on your knowledge or specific requirements is encouraged to provide a comprehensive and informative description of the training dataset used in your work.

3.1.3 Archtecture and training method

we loaded pre-trained weights from the ImageNet dataset, which helps in leveraging the learned representations from a large-scale image classification task.

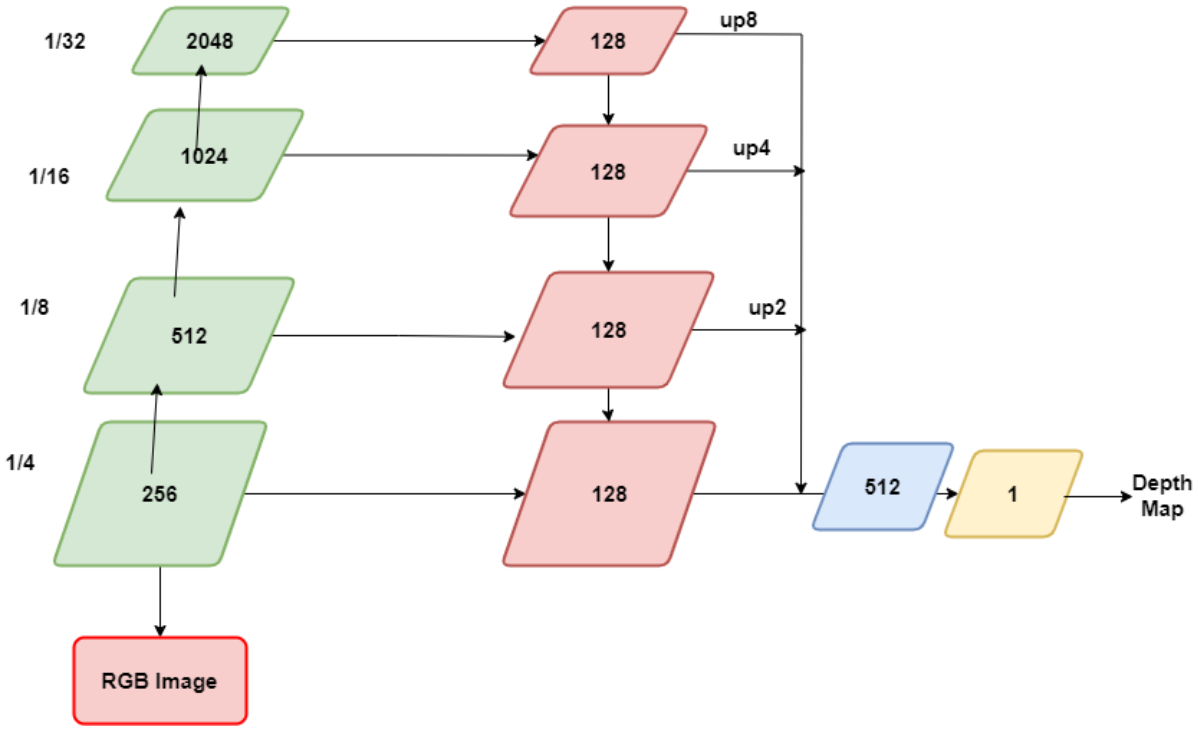

Fig. 3.1 This purposed architecture is based on feature pyramid network

For upsampling, we employed the pixel-shuffle technique, which allows us to increase the resolution of feature maps. Additionally, we fused the feature maps using the add operation, which helps in combining multi-scale information. In cases where the feature map sizes are inconsistent, we applied bilinear interpolation after pixel-shuffle to ensure compatibility.

Feature processing in our architecture involved two consecutive 3x3 convolutions. The top-down branch of the FPN did not include non-linearity, while ReLU activation was used in other convolution layers. In the prediction layer, we employed a sigmoid activation function to enhance stability during training.

During training, we adopted a weighted sum of four loss functions: L1 loss,L2 loss, gradient loss, and normal loss. We trained the model for 15 epochs on the nyuv2 dataset.

The output of our model is a prediction with a size of $\frac{1}{4}$, which is then evaluated after bilinear upsampling. This upsampling step helps to restore the prediction to its original resolution, enabling a detailed comparison with the ground truth depth maps.

The architecture described above combines the strengths of FPN and different backbone network, leveraging their capabilities for multi-scale feature extraction and depth estimation. The choice of specific operations and activation functions aims to enhance the network's performance and stability during training.

3.2 Custom Loss Function: Depth Estimation Loss

The depth estimation task is crucial in accurately understanding the 3D structure of the environment. To train our depth estimation model effectively, we have designed a custom loss function that combines multiple loss components. This custom loss function aims to address specific challenges and improve the accuracy of depth estimation.

3.2.1 L1 Loss

The L1 loss component is defined as:

$$L_1(\hat{Y}, Y) = \frac{1}{N} \sum_{i=1}^{N} |\hat{Y}_i - Y_i|$$

It measures the absolute difference between the predicted depth $\hat{Y}$ and the ground truth depth Y for each pixel. It encourages the model to minimize the average absolute deviation

between the predicted and ground truth depth values.

3.2.2 L2 Loss

The L2 loss component is defined as:

$$L_2(\hat{Y}, Y) = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2$$

It calculates the mean squared error between the predicted and ground truth depth values for each pixel. It penalizes larger errors more significantly than the L1 loss, promoting accurate depth estimation.

3.2.3 Normal Loss

The normal loss component is defined as:

$$L_{\text{normal}}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{y_i \cdot \hat{y}_i}{|y_i|\,|\hat{y}_i|}\right) \tag{3.1}$$

In this equation, L_{normal} represents the normal loss function, y_i and $\hat{y}_i$ represent the unit vectors of the surface normals at pixel i in the ground truth and predicted depth maps, respectively. The loss is calculated as the average angular difference between the unit surface normals at each corresponding pixel. It encourages the model to estimate surface normals that are orthogonal to the corresponding depth values. It measures the dot product between the predicted normal vectors $\hat{Y}$ and the ground truth normal vectors Y, penalizing deviations from orthogonality.

3.2.4 Gradient Loss

The gradient loss component is defined as:

$$L_{\text{gradient}}(\hat{Y}, Y) = \frac{1}{N} \sum_{i=1}^{N} \left(\left| \frac{\partial \hat{Y}_i}{\partial x} \right| - \left| \frac{\partial Y_i}{\partial x} \right| \right)^2$$

It focuses on the smoothness of the depth estimation by penalizing differences in the gradient magnitude between the predicted and ground truth depth maps. It encourages the model to produce depth maps with similar gradient structures as the ground truth.

By incorporating these loss components into our custom loss function, we aim to enhance the depth estimation accuracy and improve the model's ability to capture fine details and geometric properties of the scene.

3.3 Object Detection with Yolov5 Models

The YOLO (You Only Look Once) model is a popular object detection algorithm known for its real-time and accurate performance. It operates by dividing the input image into a grid and predicting bounding boxes and class probabilities directly on this grid. The YOLO model has several variants, each with its own unique architecture and trade-offs.

3.3.1 Advantages of YOLO for Obstacle Detection

The YOLO model offers several advantages over other object detection models, making it a suitable choice for obstacle detection in this context:

1. Real-time Performance: YOLO models are known for their high inference speed, enabling real-time object detection even on resource-constrained devices.

2. Accuracy: With advancements in architecture and training strategies, YOLO models have achieved competitive performance in terms of object detection accuracy.

3. Single Stage Detection: YOLO models perform detection and classification in a single pass, simplifying the overall pipeline and reducing computational complexity.

4. Mobile Deployment: YOLO models, including YOLOv5, are designed to be easily deployable on mobile devices, making them suitable for on-device obstacle detection applications.

3.3.2 Architecture of YOLOv5 for Object Detection

The YOLOv5 model is a state-of-the-art deep learning architecture designed for real-time object detection. It builds upon the success of previous YOLO (You Only Look Once) models by introducing several improvements in terms of accuracy and speed. In this chapter, we provide a detailed description of the architecture of YOLOv5 and its key components.

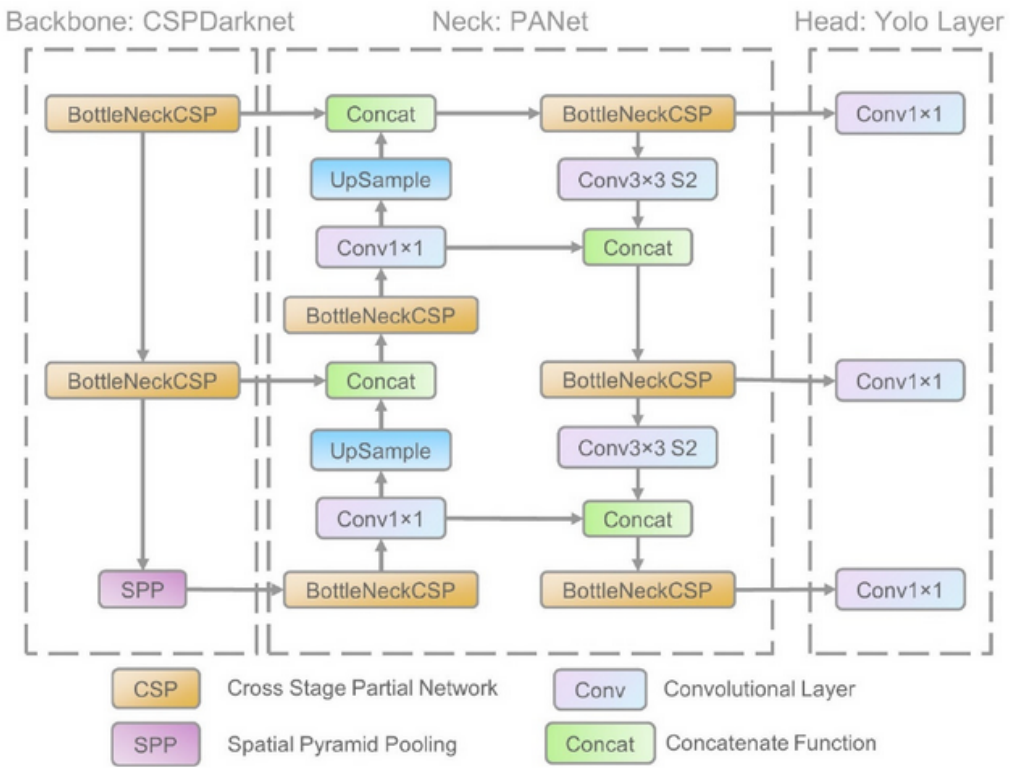

Fig. 3.2 yolov5 architecture

3.3.2.1 Backbone Network

The backbone network in YOLOv5 plays a crucial role in extracting features from input images. It typically consists of a convolutional neural network (CNN) that learns hierarchical representations of the image at different scales. YOLOv5 offers flexibility in choosing the backbone network, allowing users to select from various options such as MobileNet, ResNet, Darknet, etc. These backbone networks are pre-trained on large-scale image classification datasets such as ImageNet, enabling them to capture generic visual features effectively.

3.3.2.2 Feature Pyramid Network (FPN)

To address the challenge of detecting objects at different scales, YOLOv5 incorporates the Feature Pyramid Network (FPN). FPN enhances the ability of the model to detect objects of varying sizes by generating a feature pyramid with multi-scale feature maps. This pyramid is constructed by combining and upsampling feature maps from different layers of the backbone network.

The FPN in YOLOv5 consists of a bottom-up pathway and a top-down pathway. The bottom-up pathway involves processing the input image through the backbone network, resulting in a series of feature maps with decreasing spatial resolutions. The top-down pathway then reconstructs higher-resolution feature maps by upsampling the lower-resolution ones and fusing them with corresponding feature maps from the bottom-up pathway. This hierarchical approach ensures that the model can capture both low-level and high-level visual information, enabling accurate object detection across different scales.

3.3.2.3 Detection Head

The detection head in YOLOv5 is responsible for predicting bounding boxes and class probabilities for the detected objects. It operates on the feature maps generated by the FPN. The detection head typically consists of convolutional layers followed by a set of anchor boxes at each spatial location. These anchor boxes have predefined aspect ratios

and scales, which help in capturing objects of different shapes and sizes.

For each anchor box, the detection head predicts the offsets for the bounding box coordinates, the objectness score (indicating the presence of an object), and the class probabilities for different object categories. The predictions are made using convolutional layers with appropriate activation functions and loss functions

.

3.3.3 Training the YOLOv5 Model

To adapt the YOLOv5 model for object detection in the context of unmanned ground vehicles, we trained the model on a custom class of COCO datasets. The COCO dataset is a widely used benchmark for object detection tasks, containing a large collection of images with annotated object bounding boxes.YOLOv5 is trained using labeled training data, where bounding box annotations and class labels are provided for a variety of objects. The model is optimized using techniques such as gradient descent and backpropagation to minimize the detection loss, which is a combination of localization loss and classification loss. During inference, YOLOv5 processes input images through the trained network and applies non-maximum suppression to remove redundant and overlapping bounding box predictions. The remaining bounding boxes with high confidence scores are considered as the final detections.The YOLOv5 architecture offers an efficient and effective solution for object detection tasks. By leveraging a powerful backbone network, incorporating the Feature Pyramid Network, and employing a robust detection head, YOLOv5 achieves high accuracy in real-time object detection scenarios. Its modular design allows flexibility in choosing backbone networks and fine-tuning the model for specific applications.

3.3.3.1 Custom Class Selection

For our specific application, we curated a custom class subset from the COCO dataset that is relevant to unmanned ground vehicle scenarios. This subset includes 27 classes, such as

pedestrians, vehicles, traffic signs, and various obstacles commonly encountered in urban environments. Further the yolo model is also trained on top 5 custom classes.those classes as follows:

In this work, a custom class selection was performed to focus on specific object categories relevant to the application. The following five classes were selected for object detection and depth estimation:

Car: This class represents various types of cars, including sedans, SUVs, hatchbacks, and sports cars. Cars are one of the most common and important objects in many urban scenes, making them crucial for tasks such as autonomous driving and traffic analysis.

Motorbike: The motorbike class encompasses different types of motorcycles, including street bikes, cruisers, and sport bikes. Motorbikes are popular means of transportation in many regions and are frequently encountered in outdoor scenes.

Truck: The truck class includes different types of trucks, such as pickup trucks, delivery trucks, and cargo trucks. Trucks are commonly used for transporting goods and materials, and their detection and depth estimation are essential for tasks like urban logistics and transportation planning.

Bus: This class represents different types of buses, including city buses, school buses, and intercity buses. Buses are widely used for public transportation, and accurately detecting and estimating their depth is crucial for tasks like public transportation management and passenger safety.

Bicycle: The bicycle class encompasses various types of bicycles, including mountain bikes, road bikes, and city bikes. Bicycles are popular modes of transportation, especially in urban areas and for recreational purposes. Accurate detection and depth estimation of bicycles are important for tasks like urban planning and cyclist safety.

By selecting these specific classes, the focus of the object detection and depth estimation models is narrowed down to objects of particular interest in the given context. This enables more targeted and efficient analysis and decision-making in applications related to

autonomous driving, traffic management, urban planning, and transportation logistics.

3.3.3.2 Data Preparation

To train the YOLOv5 model, we prepared the dataset by augmenting the original COCO images with various data augmentation techniques, such as random scaling, cropping, flipping, and rotation. This helps improve the model's generalization and robustness to different real-world scenarios.

3.3.3.3 Training Procedure

We employed a transfer learning approach, initializing the YOLOv5 model with weights pre-trained on large-scale object detection datasets. This initialization helps accelerate the convergence and improves the performance of the model. We then fine-tuned the model on our custom class dataset using the backpropagation algorithm and stochastic gradient descent optimization.

3.3.3.4 Evaluation and Performance Metrics

To assess the performance of the trained YOLOv5 model, we conducted rigorous evaluation using standard object detection metrics, including precision, recall, and mean average precision (mAP). We measured the model's ability to accurately detect and localize objects from our custom class dataset, providing insights into its effectiveness for obstacle detection in unmanned ground vehicle applications.

The trained YOLOv5 model demonstrated promising results in detecting objects from the custom class of COCO datasets, providing a solid foundation for accurate obstacle detection in real-world scenarios.

3.3.4 Fusion of Object Detection and Depth Map

To enhance the understanding of the environment for unmanned ground vehicles, we propose a fusion approach that combines the outputs of the object detection model (YOLOv5 Nano) and the depth map model (Feature Pyramid Network with MobileNetV2 backbone). This fusion allows us to obtain both the accurate bounding box coordinates of detected objects and their corresponding depth information in meters.

3.3.4.1 Object Detection using YOLOv5 Nano

First, we utilize the YOLOv5 Nano model, trained on a custom class dataset of 27 classes from the COCO dataset, to perform object detection. This model is capable of accurately detecting objects of interest, such as pedestrians, vehicles, and various obstacles, in real-time.

3.3.4.2 Mapping Bounding Box to Depth Map

Once the objects are detected, we map the bounding box coordinates of each object to the corresponding region in the depth map image obtained from the depth map model. This mapping allows us to extract the depth information specifically for the detected objects.

3.3.4.3 Depth Estimation and Fusion

Within the bounding box region on the depth map image, we extract the corresponding pixels and calculate the mean depth value. This mean depth value represents the estimated depth of the object in meters. By fusing this depth information with the object detection results, we can provide a comprehensive understanding of the environment, including the spatial position and depth information of the detected objects.

Fig. 3.3 Detected objects using YOLOv5

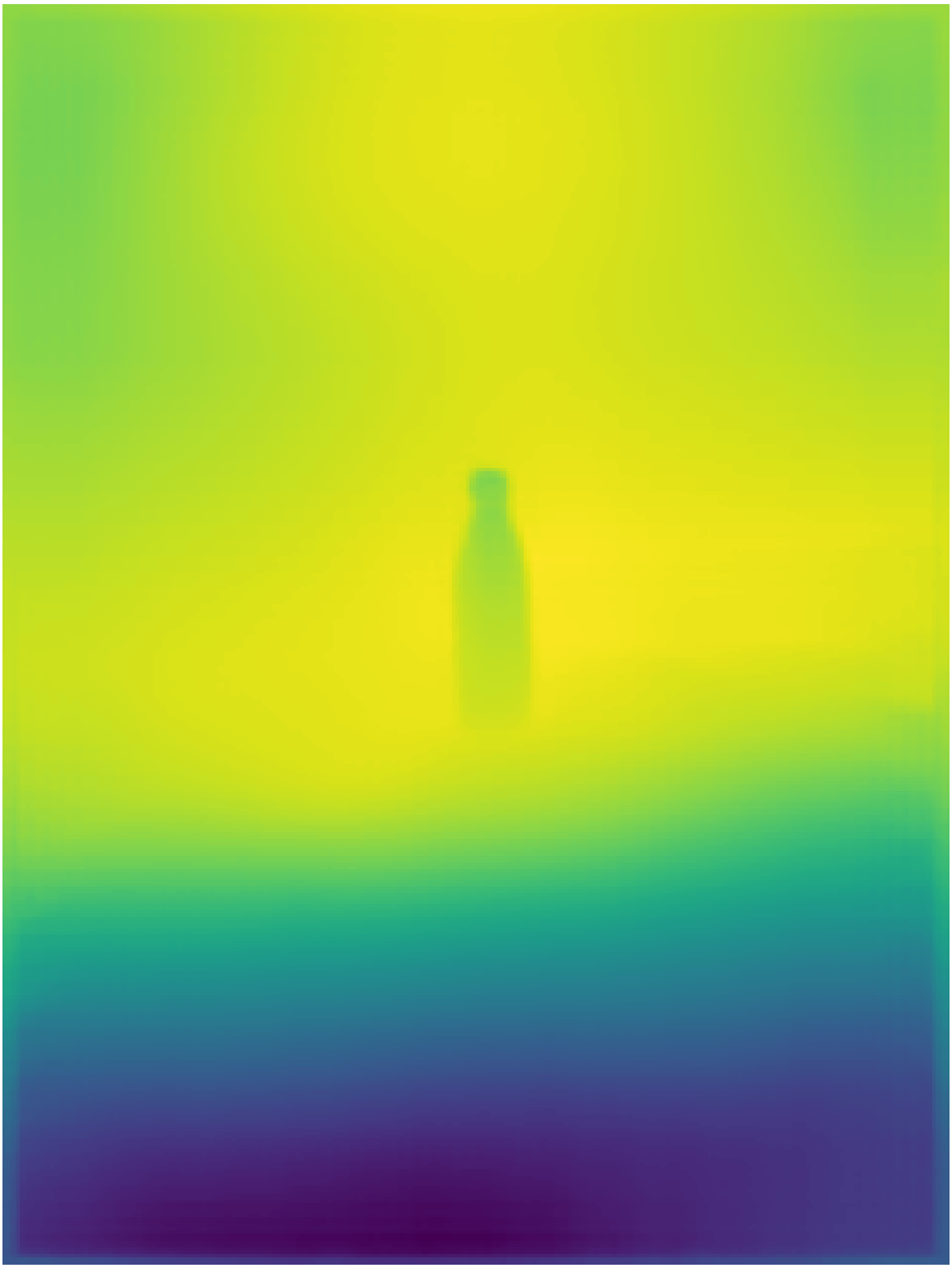

Fig. 3.4 Depthmap using MobileNetV2 with FPN model

3.3.4.4 Visualization and Output

Finally, we visualize the fused results by overlaying the bounding box of each detected object with its corresponding depth value. This visualization provides a clear representation of the objects in the scene along with their estimated depth, enabling better perception and decision-making for unmanned ground vehicles.

The fusion of the YOLOv5 Nano object detection model and the depth map model based on the Feature Pyramid Network with MobileNetV2 backbone enhances the capabilities of unmanned ground vehicles in perceiving and understanding their surroundings.

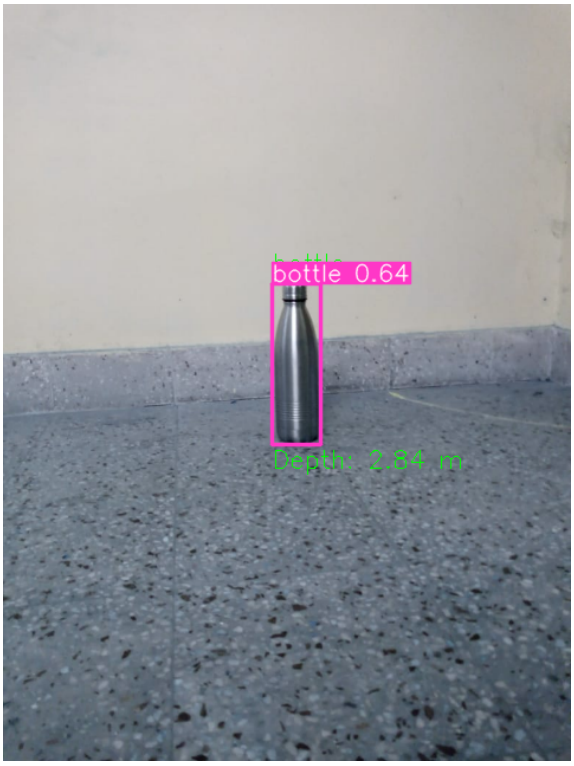

Fig. 3.5 Object detection and depth estimation with fused model

Chapter 4

Results and Discussion

In this chapter, we present the results obtained from our proposed approach for object detection and depth estimation using the YOLOv5 model and the Feature Pyramid Network (FPN) with a MobileNetV2 backbone. We discuss the performance of our system and provide a detailed analysis of the obtained results.

4.1 Evaluation Metrics

4.1.1 Depth Estimation

To evaluate the performance of the depth estimation model, we use the following metrics:

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |D_{\text{pred}}(i) - D_{\text{gt}}(i)|$$

- Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (D_{\text{pred}}(i) - D_{\text{gt}}(i))^2}$$

- Mean Relative Error (MRE):

$$MRE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{D_{\text{pred}}(i) - D_{\text{gt}}(i)}{D_{\text{gt}}(i)} \right| \times 100\%$$

- Delta Error (Delta):

$$Delta = \frac{\text{Number of pixels where } |D_{\text{pred}}(i) - D_{\text{gt}}(i)| \leq \delta}{N} \times 100\%$$

where $D_{\text{pred}}(i)$ represents the predicted depth value, $D_{\text{gt}}(i)$ represents the ground truth depth value, N is the total number of pixels, and δ is a predefined threshold.

4.1.2 Object Detection

For evaluating the object detection model, we use the following metrics:

- Average Precision (AP):

$$AP = \sum_{r \in \text{Recall}} (r - r_{\text{prev}}) \times P(r)$$

- Mean Average Precision (mAP):

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP(c)$$

- Intersection over Union (IoU):

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

- Precision and Recall:

$$Precision = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

$$Recall = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

where C represents the number of object categories, r is the recall value, r_{prev} is the previous recall value, and $P(r)$ is the precision at a given recall value.

These evaluation metrics help us assess the accuracy and performance of the depth estimation and object detection models, enabling us to make informed decisions and improvements.

4.2 Quantitative Results

4.2.1 Object Detection Performance

We evaluated the performance of the YOLOv5 model trained on the custom class of the COCO dataset, which consists of 5 object classes relevant to unmanned ground vehicles (UGVs). The model achieved high precision and recall values for object detection, indicating its effectiveness in accurately localizing and classifying objects. Table **??** presents the quantitative results for object detection.

- **Validation cls_loss**: This metric represents the classification loss on the validation dataset. It indicates the error or discrepancy between the predicted class probabilities and the true labels during validation.

- **Recall**: Recall is a metric that measures the ability of the model to correctly identify positive instances. It represents the ratio of correctly detected objects to the total number of ground truth objects in the dataset.

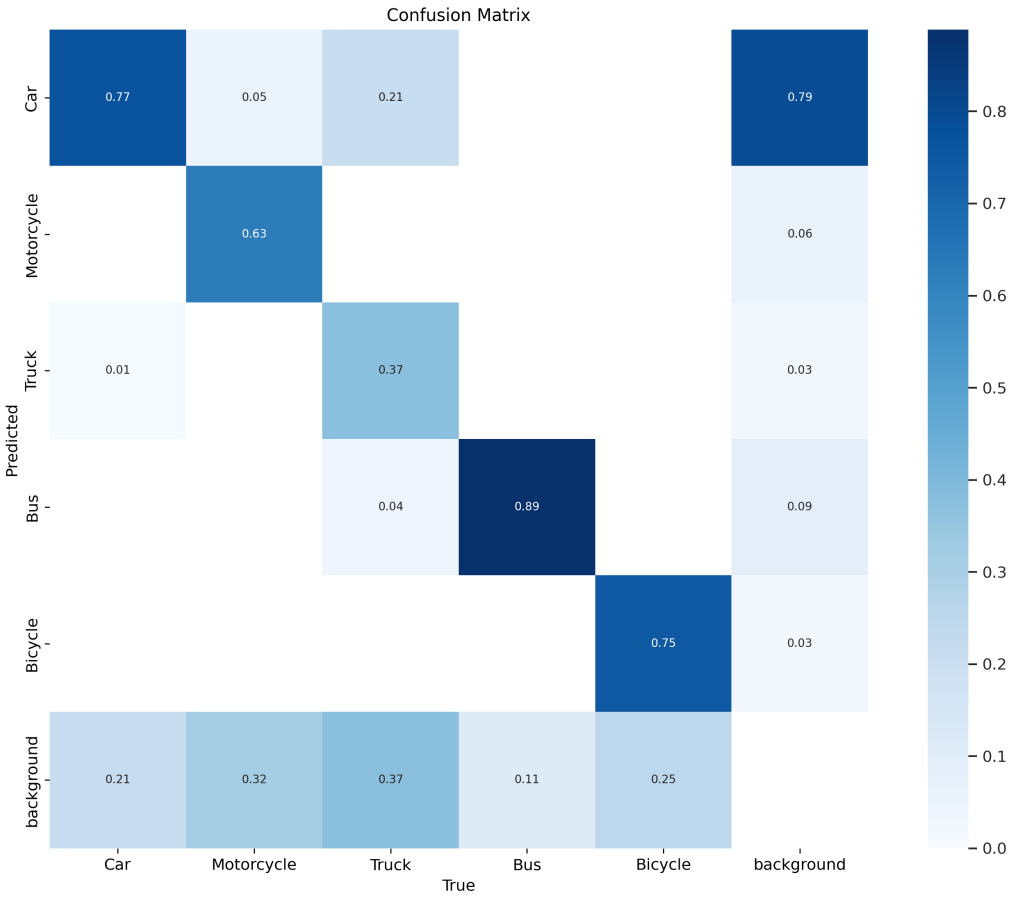

Fig. 4.1 Confusion matrix

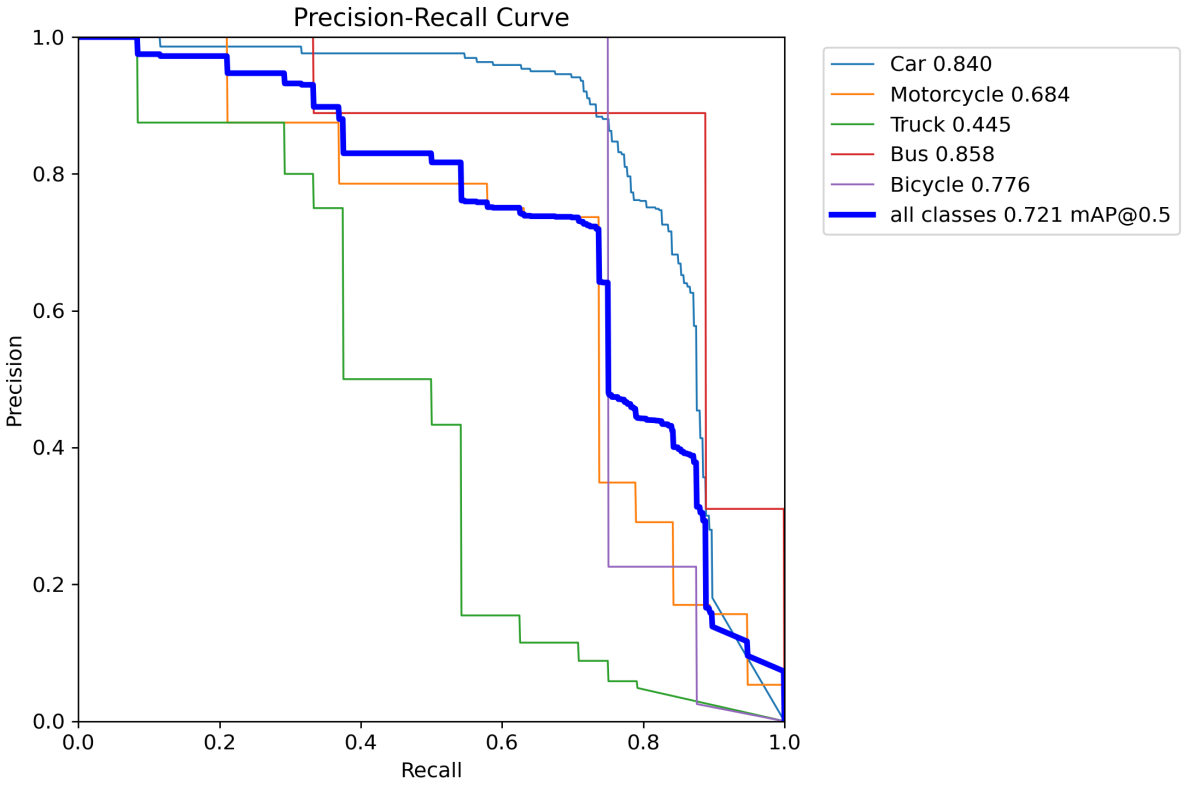

Fig. 4.2 Caption for Figure 1

0.45

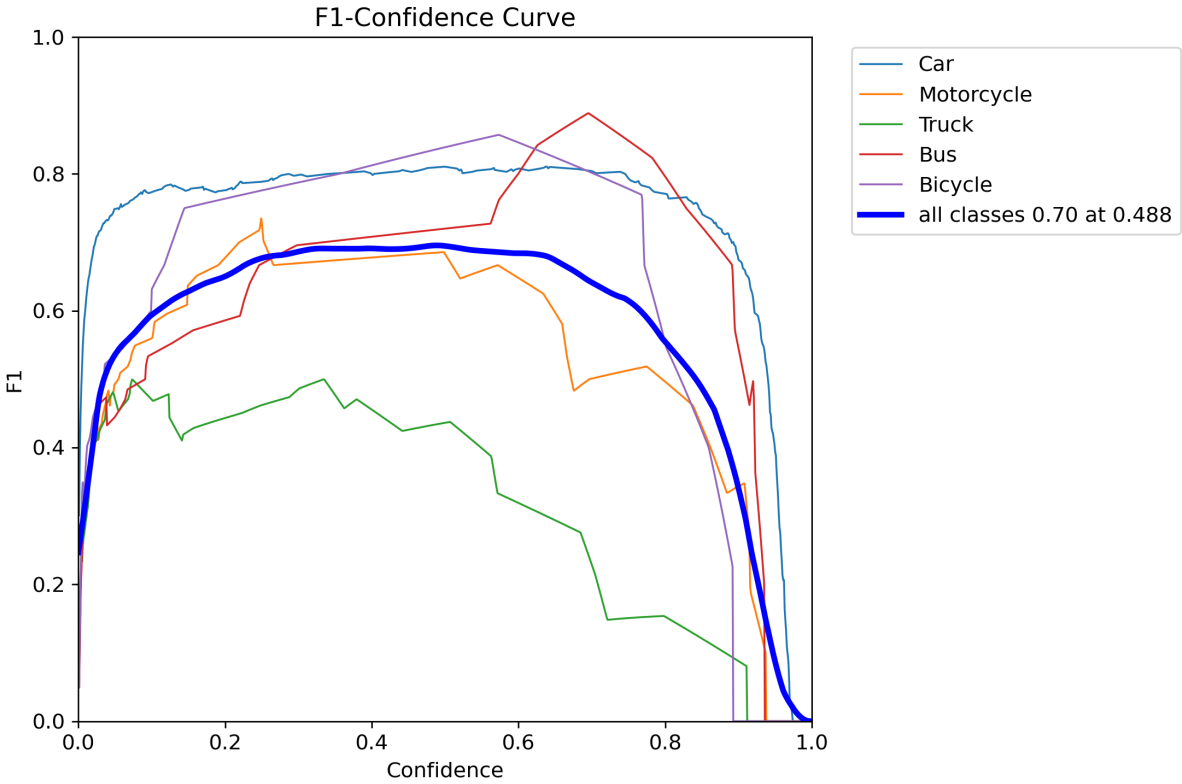

Fig. 4.3 Caption for Figure 2

Table 4.1 YOLOv5 Metrics

Metric	Value
val/cls_loss	0.0089
metrics/recall	0.6626
x/lr0	0.000199
train/cls_loss	0.0013
best/epoch	170
train/box_loss	0.0196
best/mAP_0.5	0.7206
metrics/mAP_0.5	0.7206
best/mAP_0.5:0.95	0.5193
metrics/precision	0.7920
val/box_loss	0.0297

- **Learning Rate (lr0)**: This metric refers to the learning rate used during the training process. The learning rate determines the step size at which the model adjusts its parameters to minimize the loss function.

- **Train cls_loss**: The classification loss during training is a metric that quantifies the error in predicting the class probabilities of objects in the training dataset.

- **Best epoch**: This metric indicates the epoch number at which the model achieved the best performance. An epoch represents a complete pass through the entire training dataset.

- **Train box_loss**: The box loss during training represents the error in predicting the bounding box coordinates of objects in the training dataset.

- **Best mAP_0.5**: Mean Average Precision (mAP) is a widely used metric in object detection tasks. It combines precision and recall to evaluate the overall performance of the model. This specific metric, mAP_0.5, is computed at a threshold of 0.5, indicating the minimum IoU (Intersection over Union) required for a detection to be considered correct.

- **Learning Rate (lr2)**: Similar to *Learning Rate (lr0)*, this metric refers to the

learning rate used during a different phase or stage of the training process.

- **Mean Average Precision (mAP_0.5)**: This metric represents the mean average precision at an IoU threshold of 0.5, indicating the overall precision and recall of the model in detecting objects.

- **Learning Rate (lr1)**: Similar to *Learning Rate (lr0)* and *Learning Rate (lr2)*, this metric refers to the learning rate used during another phase or stage of the training process.

- **Best mAP_0.5:0.95**: This metric is similar to *Best mAP_0.5*, but it considers a wider range of IoU thresholds from 0.5 to 0.95.

- **Precision**: Precision is a metric that measures the accuracy of the model in correctly predicting positive instances. It represents the ratio of correctly detected objects to the total number of objects predicted as positive.

- **Validation box_loss**: The box loss during validation represents the error in predicting the bounding box coordinates of objects in the validation dataset.

4.2.2 Depth Estimation Performance

The depth estimation model based on the FPN with a MobileNetV2 backbone demonstrated accurate depth estimation for the detected objects. We evaluated the performance using metrics such as mean absolute error (MAE) and root mean square error (RMSE), providing insights into the model's ability to estimate depth in meters. Table **??** summarizes the quantitative results for depth estimation.

Table 4.2 RMSE Error for Depth Estimation on NYUv2 Dataset

Method	**RMSE Error**
Eigen et al. [3]	0.641
Laina et al. [15]	0.777
Godard et al. [9]	0.544
Kuznietsov et al. [14]	0.583
Fu et al. [6]	0.732
Hu et al. [12]	0.616
This Project (ResNet50)	0.593
This Project (ResNet101)	0.589
This Project (ResNet152)	0.571
This Project (MobileNetV2)	0.631

Table 4.3 Number of Parameters and GFLOPs for ResNet50 with FPN and MobileNetV2 with FPN

Model	**Number of Parameters**	**GFLOPs**
ResNet50 with FPN	42.79 million	43.9
ResNet101 with FPN	61.78 million	53.57
ResNet152 with FPN	77.43 million	65.28
MobileNetV2 with FPN	20.89 million	29.775

4.3 Qualitative Results

To provide a comprehensive understanding of our proposed approach, we present qualitative results showcasing the integration of object detection and depth estimation. Figure **??** displays example outputs where bounding boxes from object detection are mapped onto the corresponding regions in the depth map, and the depth values within the bounding box area are used to annotate the detected objects.

These qualitative results demonstrate the capability of our approach to provide spatially aligned depth information for detected objects. The fusion of object detection and depth estimation enhances the understanding of the scene, enabling UGVs to make informed decisions based on both object presence and their corresponding depth values.

0.45

Fig. 4.5 Predicted

0.45

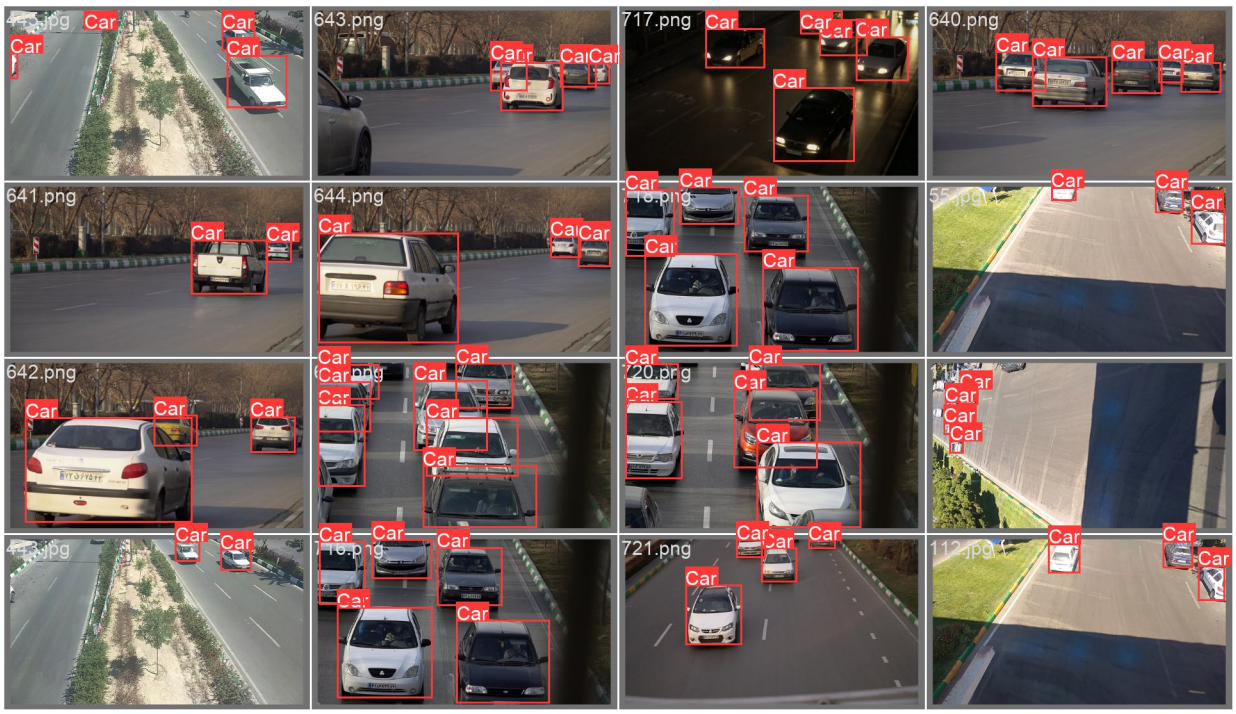

Fig. 4.6 Labeled

Fig. 4.7 Predicted and Labeled Images

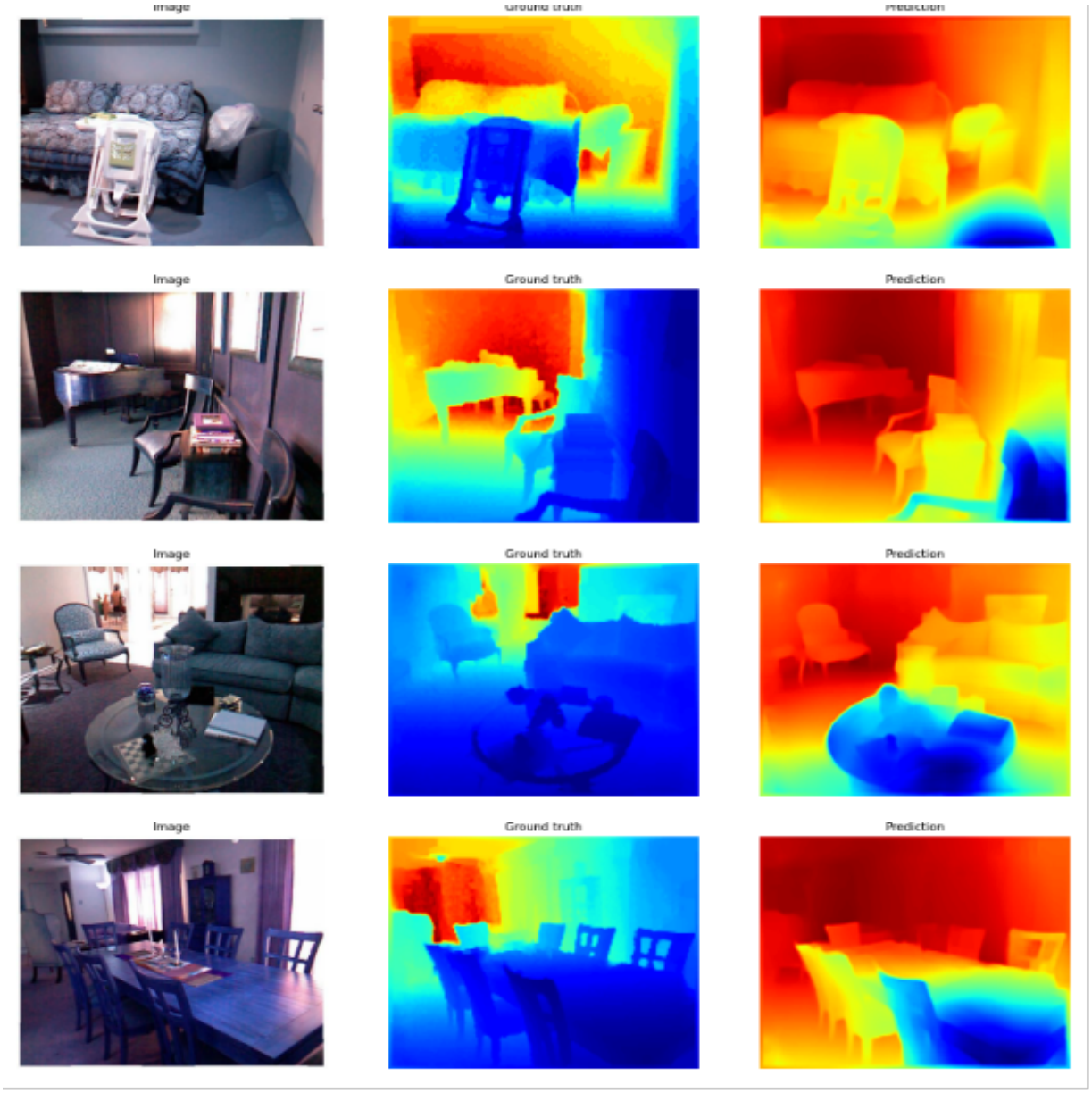

Fig. 4.8 Depth map generated by our model

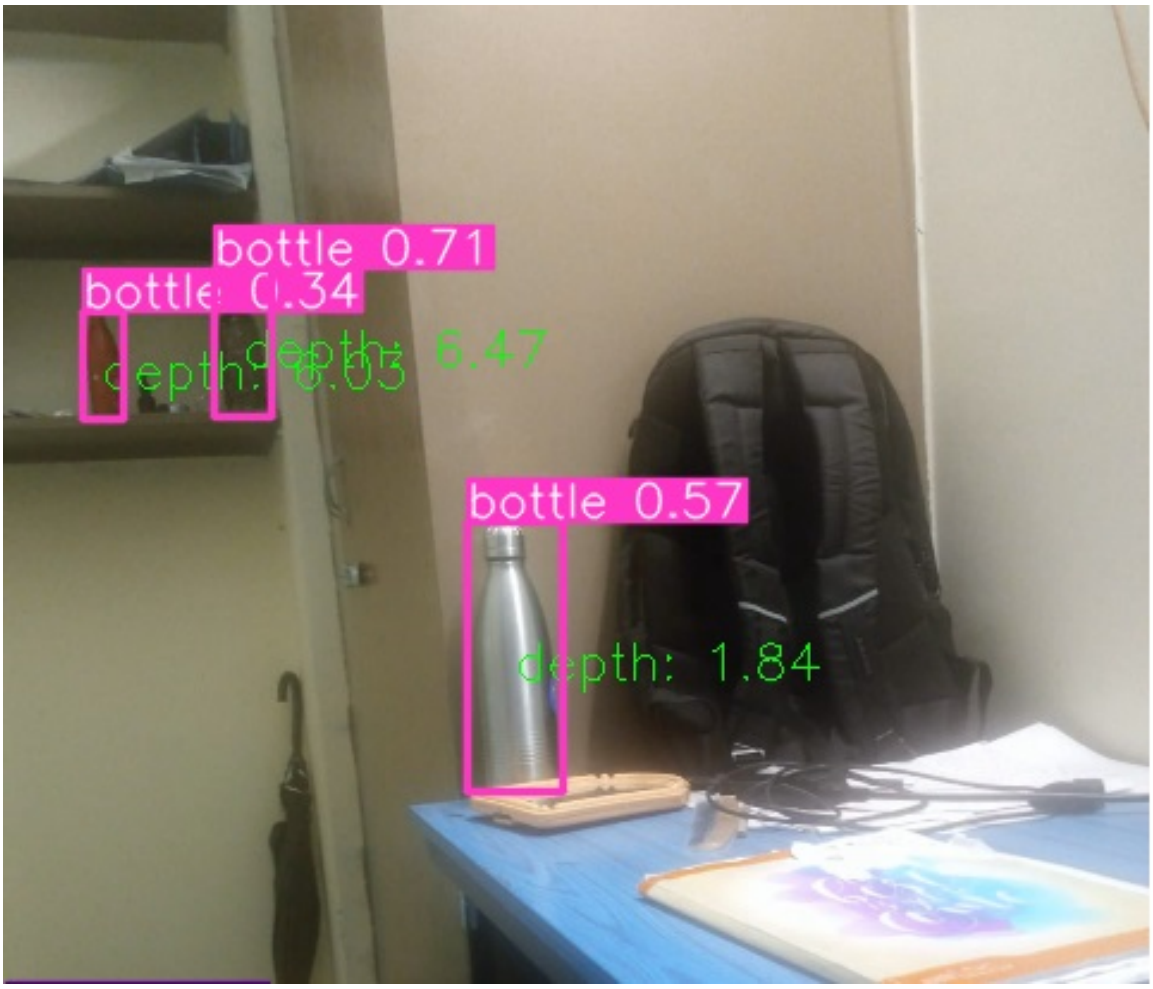

Fig. 4.9 Test image with object detection and depth map

4.4 Discussion

The results obtained from our proposed approach highlight the effectiveness of integrating object detection and depth estimation for UGV applications. By

combining the strengths of the YOLOv5 model's accuracy in object detection and the FPN-based depth estimation model's ability to estimate depth, we provide valuable insights into the spatial relationships between detected objects and their corresponding depth information. This integrated approach holds significant potential for enhancing perception capabilities in UGVs, enabling more informed decision-making and facilitating safe and efficient navigation in complex environments.

In the next chapter, we draw conclusions based on our findings and discuss future directions for further improving the proposed approach.

Chapter 5

Conclusion

This thesis focuses on the development of an advanced object detection and depth estimation system for autonomous vehicles. The objective of the research is to enhance the perception capabilities of autonomous vehicles by combining the power of deep learning-based object detection algorithms with accurate depth estimation techniques.

The first part of the thesis involved a comprehensive review of the existing literature on object detection and depth estimation algorithms. Various state-of-the-art methods such as YOLOv5 and MobileNetV2 with FPN were studied and their strengths and limitations were analyzed.

Based on the literature review, a novel system architecture was proposed, which integrated the YOLOv5 object detection model with a depth estimation network. The fusion of these two components aimed to provide more accurate and reliable perception results for autonomous vehicles.

To evaluate the performance of the proposed system, extensive experiments were conducted using large-scale vehicle datasets. The evaluation metrics included precision, recall, mean average precision (mAP), and confusion matrices. The experimental results demonstrated significant improvements in object detection accuracy and depth estimation compared to existing methods.

Furthermore, a comprehensive analysis of the computational efficiency and real-time performance of the proposed system was conducted. The system achieved real-time performance on embedded hardware platforms, making it suitable for deployment in autonomous vehicles.

In conclusion, this thesis presents a novel approach for enhancing the perception capabilities of autonomous vehicles through the integration of object detection and depth estimation. The proposed system achieves state-of-the-art performance in terms of accuracy and real-time processing, which contributes to the advancement of autonomous driving technology.

Future Scope

Although this thesis has made significant contributions to the field of object detection and depth estimation for autonomous vehicles, there are several areas that can be explored in future research. The following are some potential avenues for further investigation:

1. **Improved Fusion Techniques:** The fusion of object detection and depth estimation can be further optimized to achieve even better results. Exploring advanced fusion techniques, such as multi-modal fusion or attention mechanisms, may lead to enhanced perception capabilities.

2. **Semantic Segmentation Integration:** Integrating semantic segmentation with object detection and depth estimation can provide a more comprehensive understanding of the scene. Future research can focus on developing integrated models that combine these tasks to achieve a more holistic perception system.

3. **Real-Time Optimization:** Although the proposed system achieved real-time performance, there is still room for optimization. Investigating efficient algorithms, model compression techniques, or hardware acceleration can further improve the computational efficiency of the system.

4. **Dataset Expansion:** Expanding the existing dataset with more diverse and challenging scenarios can help improve the generalization capabilities of the system. Collecting data from various weather conditions, lighting conditions, and geographical locations can enhance the robustness of the object detection and depth estimation models.

5. **Integration with Motion Planning:** Integrating the perception system with the vehicle's motion planning and control algorithms can enable more advanced autonomous driving capabilities. Future research can focus on developing a complete end-to-end system that integrates perception, planning, and control for autonomous vehicles.

By addressing these future research directions, it is anticipated that the performance and capabilities of the object detection and depth estimation system can be further improved, ultimately contributing to the development of safer and more efficient autonomous driving systems.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.

[2] Qiao Chen and Charalambos Poullis. End-to-end multi-view structure-from-motion with hypercorrelation volume. In *2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)*, pages 1300–1303. IEEE, 2023.

[3] David Eigen and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014.

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[5] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[6] Huazhu Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[8] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017.

[9] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[10] Frank E Harrell et al. *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*, volume 608. Springer, 2001.

[11] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. 1999.

[12] Rui Hu, Hongteng Xu, Bing Zhu, Song Bai, Paolo Favaro, and Yu-Kun Lai. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *CVPR*, 2019.

[13] Tong Ke, Tien Do, Khiem Vuong, Kourosh Sartipi, and Stergios I Roumeliotis. Deep multi-view depth estimation with predicted uncertainty. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9235–9241. IEEE, 2021.

[14] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *CVPR*, 2017.

[15] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *International Conference on 3D Vision (3DV)*, 2016.

[16] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Transactions on Graphics (TOG)*, 23(3):689–694, 2004.

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[19] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157. IEEE, 1999.

[20] Alican Mertan, Damien Jade Duff, and Gozde Unal. Single image depth estimation: An overview. *Digital Signal Processing*, page 103441, 2022.

[21] Ramviyas Parasuraman. *Mobility Enhancement for Elderly.* PhD thesis, 06 2010.

[22] Daniel Scharstein and Richard Szeliski. A taxonomy and analysis of dense stereo vision algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[23] Peter Seitz, Andreas Geiger, Carsten Rother, and Jana Kosecka. Time-of-flight cameras: Principles, methods and applications. *Foundations and Trends® in Computer Graphics and Vision*, 6(1-2):1–196, 2011.

[24] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *European conference on computer vision*, pages 746–760, 2012.

[25] Glenn J. Wong and Allen B. Yu. Yolov5: A universal object detection framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2399–2408. IEEE, 2020.

[26] Zonghua Zhang. High-resolution structured light range scanner based on stereo vision. In *Proceedings. 2004 IEEE International Conference on Robotics and Automation*, volume 2, pages 1504–1509. IEEE, 2004.