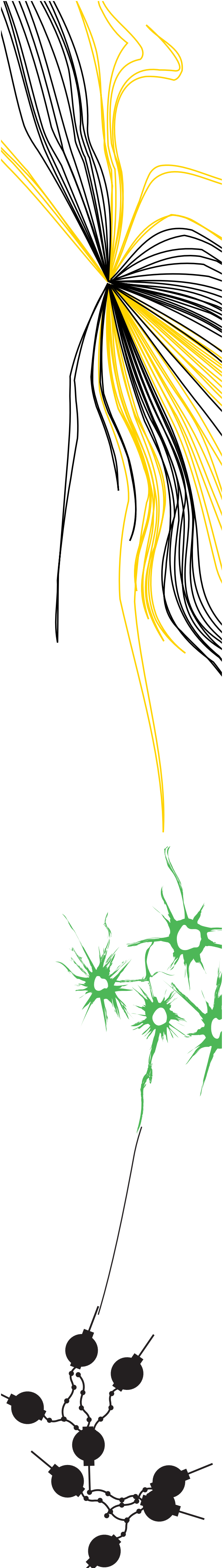
Comparative Analysis of Dynamic Object Segmentation Networks for Tele-robotic Applications

Cedric Damien DSouza

April 26, 2024



RAM. ROBOTICS AND MECHATRONICS

COMPARATIVE ANALYSIS OF DYNAMIC OBJECT SEGMENTATION NETWORKS FOR TELE-ROBOTIC APPLICATIONS

C.D. (Cedric Damien) Dsouza

MSC ASSIGNMENT

Committee:
dr. ir. D. Dresscher
dr. F.C. Nex

January, 2024

001RaM2024
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Summary

The existing telerobotic system captures images of a remote environment, generating a VR map accessible from a different location. However, an inherent challenge arises in the efficient tracking of dynamic objects within the SLAM framework. In particular, inadequate tracking of images hinders the accurate updating of the map, affecting its fidelity to the original environment.

The primary objective of this assignment is to develop a module dedicated to enhancing dynamic object tracking through segmentation in SLAM. The goal is to ensure the seamless and error-free map updates in VR, replicating the nuances of the remote environment faithfully.

Over the past decade, various techniques, including Background Subtraction and Optical flow algorithms, have been employed for dynamic object segmentation and tracking. In this assignment, a thorough exploration of diverse algorithms has been conducted and implemented. The evaluation of multiple methods for real-time object detection reveals varying benefits and limitations. Based on thorough research, a plausible recommendation is to combine a deep learning approach with traditional computer vision methods. This integration aims to overcome the limitations of each approach, proposing a hybrid method that leverages the strengths of both. Existing research substantiates the effectiveness of such hybrid approaches in enhancing the overall performance of real-time object detection systems.

Choosing an optimal model depends on a meticulous examination of application-specific needs, striking a balance between factors like computational resources, accuracy, and real-time performance. This nuanced evaluation underscores the varied strengths and applicability of each model, enriching the evolving toolkit for addressing semantic segmentation challenges in the field of computer vision.

A comprehensive comparative analysis has been conducted among various models, considering several crucial factors. However, the recommendation does not favor a specific model, as each model exhibits greater efficiency in different scenarios. The advantages of the compared models are thoroughly discussed to provide a nuanced understanding of their respective strengths in diverse contexts.

Acknowledgment

I express my sincere gratitude to my supervisors, Douwe Dresscher and Francesco Nex, for their steadfast support and guidance throughout my academic journey. Their understanding of my personal situation and continuous encouragement played a pivotal role in completing this thesis.

I would also like to convey my appreciation to Robin Lieftink for his assistance in attempting to execute the existing setup provided. His contributions were valuable to the project.

Lastly, heartfelt thanks go to my family for their unwavering support and encouragement. Their constant backing has been a source of strength.

This paper utilizes AI technology, specifically the GPT-3 model developed by OpenAI, to enhance the quality of writing in particular sections. The authors are accountable for critical analysis, interpretation, and drawing conclusions. The role of AI is to refine articulation, improve organizational structure, and enhance overall clarity, all while maintaining the authentic expertise of the authors.

In conclusion, this thesis is a testament to collective efforts, and I extend my heartfelt gratitude to all who have contributed to its realization.

List of Abbreviations

SGNet Additive Manufacturing

PSPNet Fused Deposition Modelling

ERFNet Palmiga Innovations Engineered Thermoplastic Polyurethane

ICNet Engineered Thermoplastic Polyurethane

BiSeNet Stress Concentration Factor

dd Computer Aided Design

IMDJ Injection Molded Direct Joining

PLA Polylactic Acid

EPLA Electrically Conductive Composite Polylactic Acid

FLEPS Flexible and Printable Sensors an Systems

Contents

1 Introduction

1.1 Context

Tele-robotics represents a specialized branch of robotics enabling users to command an entire robotic system remotely. This approach is particularly advantageous for tasks deemed too difficult for direct human intervention, such as altering fuel rods in a nuclear reactor or inspecting explosive ordnance. Tele-operated systems efficiently execute these challenging tasks, ensuring the safety of the controlling engineer. Moreover, tele-operation proves advantageous in reaching locations challenging for humans, such as deep-sea observation. This method enhances safety, accessibility, and efficiency in performing complex and hazardous tasks.

In recent years, engineers have developed tele-operated systems featuring comprehensive haptic and visual interfaces seamlessly integrated for users. This design enables users to visually perceive the tele-robot's surroundings, while the incorporated haptic feedback system allows them to feel any physical interactions encountered by the robot. By utilizing tele-operated systems of this nature, individuals can experience remote locations without the need to physically travel to a specified secondary site. For the purpose of this assignment, emphasis will be placed on the visual aspect of these systems, with no inclusion of haptic feedback.

In prior research, a tele-robotic system has been developed to capture images of a distant environment. This system employs a SLAM algorithm to construct a virtual replica of the remote surroundings, offering users a view from a secondary location. The real-time video feed necessary for this task is acquired through a Kinect 2 camera mounted on the James platform. By utilizing an HTC VIVE, individuals can immerse themselves in the virtual map generated from the live feed captured by the Kinect camera.

1.2 Problem Statement

A significant issue within the current setup is what is commonly referred to as the afterimage effect. Inaccurate detection of dynamic objects from the video feed can lead to errors in updating the virtual reality (VR) map. Another identified issue is the lack of proper image tracking due to the camera's motion.

The experiments conducted in the current setup are exclusively indoors at this stage. Consequently, additional challenges like sudden changes in illumination and dynamic backgrounds have not been taken into account. Addressing these factors would significantly complicate the task of dynamic object tracking.

Upon the realization of the objectives of this tele-robotic system, its successful implementation can be extended to serve various purposes. The ability to perceive an environment remotely provides numerous benefits. It can find applications in the entertainment industry, enabling individuals to experience the excitement of observing wildlife in its natural habitat. Moreover, it holds potential in aiding people to overcome specific fears, such as acrophobia(fear of heights), by providing a virtual exposure to challenging scenarios.

1.3 Problem Description

The primary objective of this assignment is to achieve precise image segmentation, requiring accurate recognition of all dynamic objects. Real-time execution becomes another crucial objective, especially because of the need to finish the segmentation task promptly. Utilizing the Kinect camera for image data input, which captures both RGB and depth information, introduces computational challenges due to the real-time constraint.

While image segmentation typically involves considering factors like rapid illumination changes and dynamic moving environments, the indoor setting of this setup allows for controlled illumination changes, and the likelihood of dynamic backgrounds is minimal. However, looking ahead to potential future uses of this device, it becomes evident that these factors will play a pivotal role in informing design choices.

1.4 Research Questions

1. How does the integration of deep learning approaches with traditional computer vision methods impact the accuracy and real-time performance of dynamic object segmentation?

2. What are the computational challenges associated with achieving real-time image segmentation, and how do different algorithms handle these challenges?

3. How do different evaluation metrics, such as accuracy, precision, recall, and processing time, influence the comparison of image segmentation algorithms, and what are the considerations when selecting appropriate metrics?

4. What role do machine learning techniques, such as deep learning models, play in improving the precision and real-time execution of image segmentation algorithms, and how do they compare to traditional methods?

5. What are the trade-offs between precision and real-time execution in image segmentation algorithms, and how can these trade-offs be managed effectively for applications requiring both high accuracy and prompt segmentation?

6. How do different image segmentation algorithms perform in terms of precision and real-time execution and what factors contribute to variations in performance?

2 Literature Review

The field of computer vision plays a crucial role in addressing the vision-based components of tele-operated systems. Computer vision, a field within computer science, focuses on processing, analyzing, and comprehending digital images and videos. The image data for processing can take various forms, including single camera views, multi-camera perspectives, or multi-dimensional data from a 3D scanner. Common tasks in computer vision encompass recognition, motion analysis, scene reconstruction, and image restoration.

In this assignment importance is given to two main tasks: image segmentation and motion analysis. While image segmentation has become more efficient with modern algorithms, challenges arise when dealing with video content, especially in real-time scenarios. Segmentation involves dividing multiple video frames to detect objects, offering a viable approach for analysis.

2.1 Image Segmentation and its Evolution over time

Image segmentation has been a cornerstone in the field of digital image processing since its inception. At its core, image segmentation entails assigning a unique label to each pixel within an image, thereby aiding in the identification of similarities among pixels. Although initially designed to identify boundaries like curves or lines, contemporary segmentation techniques have evolved to yield more detailed results, facilitating the recognition of complete objects with complex features. The effectiveness and versatility of image segmentation stems from the consideration of various factors, including color, intensity, and texture, making it an invaluable tool for analyzing digital images.

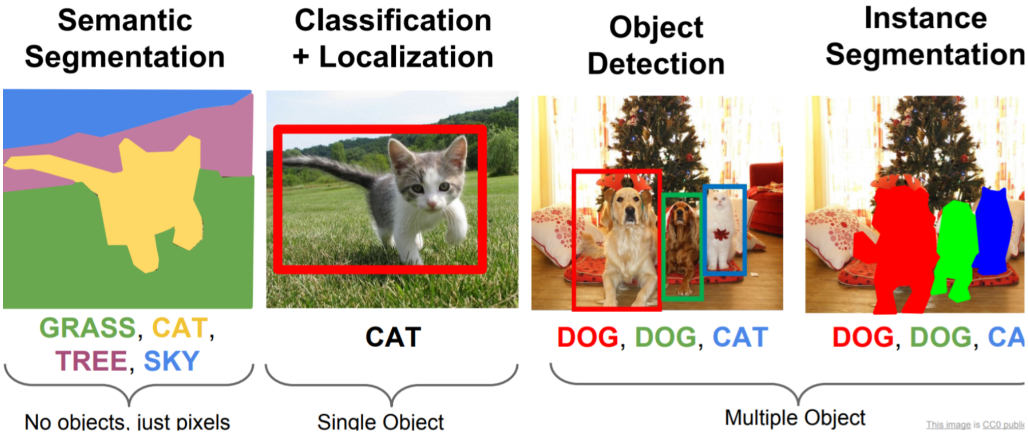


Figure 2.1: Image showing standard results of Segmentation and Object Detection

Accurate segmentation is the primary objective in this assignment. Various techniques can be employed to achieve image segmentation, a process deemed crucial among the foundational steps required for more complex processes like object detection and tracking. Numerous algorithms are available to aid in segmentation. Li and Ngan [1] have systematically classified several of these methods into seven common categories some of which will be elaborated upon below.

2.2 Research Framework

In their work in [2], the researchers grounded their investigation on two classifications outlined by Li and Ngan. [2] presents a comprehensive survey of the prevalent techniques utilized for image segmentation. Some frequently employed methods for image segmentation include the background subtraction method, energy minimization algorithms, clustering-based segmentation, and potentially a fusion of these approaches.

In [3], a clustering-based method is utilized for detecting multiple objects in a video sequence without needing prior training data. Furthermore, real-time object detection has been achieved through techniques like DynaSLAM [4] and DetectFusion [5].

Another approach for real-time object detection is discussed in [6], particularly tailored for soccer bots. These bots use computer vision to identify the ball and players, helping in navigation and strategy decisions. In [6], Kalman filters are implemented for both ball and obstacle detection. Notably, the considerable motion of the camera in soccer bots, which move freely with cameras mounted on them, introduces challenges. This motion results in heightened noise levels and an increased likelihood of false detections. While color segmentation could be used for ball detection, limitations such as susceptibility to lighting changes and potential failure when the ball is partially hidden require additional measures.

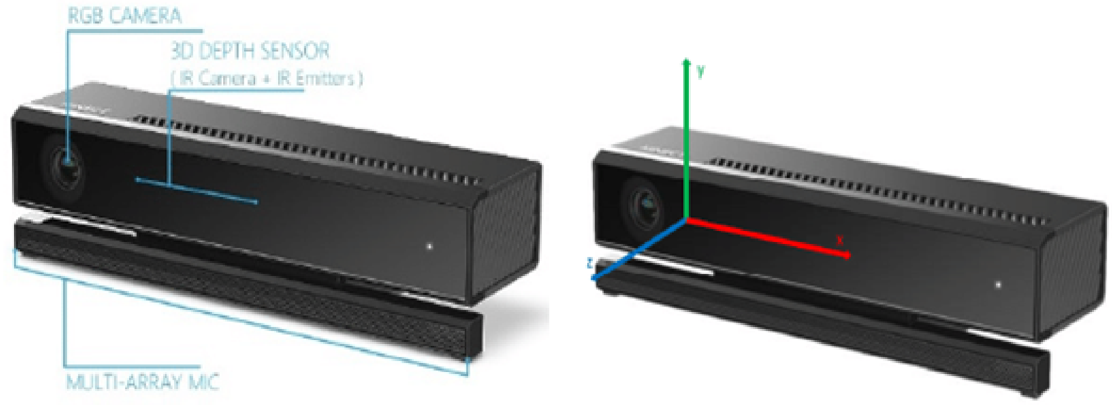


Figure 2.2: : Image of a Kinect 2 which is the Input device for the system

In [6], two additional steps are implemented to address the previously mentioned limitations. Edge detection is employed to identify the edges of the ball, and a RANSAC algorithm is utilized to fit a circle to the acquired points from the edge detection step. These supplementary measures significantly enhance the accuracy of ball position determination. For tracking multiple objects, a Kalman filter is applied. However, it is worth noting that this method has a drawback, as it tends to yield several false positives near the robots due to shadows. Despite this drawback, the combination of the RANSAC algorithm and the Kalman filter represents one of the simpler approaches for real-time object detection.

While the scenario may differ, the use of both the RANSAC algorithm and the Kalman filter suggests the need for further research, given the notably efficient computational time of the proposed approach. This efficiency is crucial for the soccer bots, as rapid information processing is essential for determining their next actions. In a broader context,it is generally recognized that a system exceeding a processing time of 20 milliseconds is not deemed real-time.

In [7], object detection utilizes depth data from the Kinect camera. Syarafuddin's proposed approach involves capturing an image of the object within a range of 1 foot to 5 feet, aimed at determining the optimal performance range of the Kinect camera. The findings indicate that the Kinect camera exhibits optimal performance between 4-5 feet. An important observation is that misalignment between the original image and the depth image occurs when the object is not at the optimal distance from the Kinect camera. Furthermore, Kinect cameras exhibit remarkable robustness in situations with minimal to no visible light.

Overcoming depth limitations of Kinect Device

In [8] an enhanced algorithm has been presented to overcome the depth limitations of the Kinect device. In this approach object detection and recognition have been achieved using the RGB image. In the method proposed in [8], the depth image's bounding box of an object is divided into grid cells. By calculating the mean value of pixels within these grid cells, issues related to faulty pixel values and pixel variation over the same object are effectively addressed. This approach results in an overall improvement in the depth accuracy estimation of the Kinect without compromising system performance.

Advanced Techniques for Moving Object Segmentation

In [9], a distinct approach is presented wherein moving objects are independently segmented from a 3D video. The input data for this approach can be obtained from either the Kinect camera or a stereo camera, with the method building 3D point clouds for object tracking. Meanwhile, Xie [10] proposes an algorithm that utilizes the ego-motion of a moving stereo camera to estimate and identify multiple moving objects through the Modified RANSAC algorithm. Although [9] uses a temporal modified RANSAC method for longer videos, this approach has drawbacks when assigning feature points to individual moving objects. To overcome these limitations, Tatematsu introduces the graph cut-based method, drawing inspiration from Ajay's work in [11], where fixation-based segmentation is used to segment regions lacking feature points. Unlike Ajay's manual selection of feature points, Tatematsu suggests using detected feature points as seeds for graph cut segmentation.

The fixation-based segmentation involves creating a probabilistic object boundary map using an edge map, modeling the probability that an edge is a physical boundary. The user then labels a fixation point within an object, and using this selected fixation point as a pole in log-polar space, the graph cut segmentation is applied.

The input data from the Kinect, once processed by the algorithm, can be leveraged to reconstruct each moving object or the background. The process involves computing 3D optical flow into 3D flow sets, simultaneously obtaining rotation and translation vectors for these sets. Subsequently, fixation-based segmentation is applied to multiple feature points to segment the initial object area. The 3D point clouds are then acquired and merged using the rotation and translation vectors. A notable advantage of this approach is its effectiveness in detecting objects against stationary backgrounds.

Object Detection Using Bag of Words Model

In [12], Jason Owens presents an approach tailored to identify various types of objects, including different planes or cars. The method utilizes a Bag of Words model, a straightforward yet powerful technique for object detection. This model extracts distinctive visual features across a dataset, representing an object's appearance within an image through a histogram of these features. The Histogram of Oriented Gradients (HOG) descriptor is employed, addressing the limitations of the Bag of Words model, particularly in discriminating individual features from cluttered scenes.

Point Cloud Processing for Object Segmentation

The point clouds, often noisy, undergo simultaneous filtering due to the presence of disparity discontinuities. A voxel grid filter is applied to significantly reduce the point cloud data's size, contributing to a noteworthy reduction in computational time. Smoothing of the point cloud is achieved using the moving least squares algorithm. Finally, the point cloud data is segmented using the connected components approach. Experimental results in [12] demonstrate that the system generates more detection's when utilizing the point cloud-derived object segmentation of image regions compared to using only an image-based approach.

2.3 Background subtraction Based techniques

In this section an examination of several background subtraction techniques for image segmentation has been performed. The approach taken by each of these methods differs from each other but each of them contribute to key features which will help improve image segmentation in key scenarios. The survey which was performed in [2] emphasizes that background subtraction is one of the commonly used techniques for image segmentation. In addition to background subtraction, [2] introduces other techniques like the energy minimization method, which is implemented to enhance the outcomes of existing segmentation techniques.

Advancements in Background Subtraction

In [13], a blend of techniques is employed to achieve more efficient segmentation results. This approach combines a background subtraction model with an optical flow and matting algorithm. Background subtraction, a fundamental step in video processing for advanced computer vision applications, proves valuable for addressing gradual illumination changes. However, its limitations in handling rapid illumination changes are addressed in [14], where a Gaussian mixture model is integrated with a Phong shading model. This combination adjusts to rapid lighting changes by utilizing frame information to reset the Gaussian mixture model. Notably, this method can detect not only fast-moving objects but also slow-moving objects and objects that have come to a complete stop.

Zhu and Song [15] present a model incorporating a recursive Bayesian estimator to update background parameters, enhancing the robustness of the background model. Background subtraction is executed only after the algorithm confirms the model's robustness. When compared with the Gaussian mixture model, this approach demonstrates superior results.

In [16], a pioneering approach for background subtraction with real-time semantic segmentation has been introduced. The system comprises two main components: a conventional Background Subtraction (BGS) segmenter and a real-time semantic segmenter. Notably, the progress in deep convolutional neural networks has significantly advanced semantic segmentation, demonstrating robustness to challenges such as illumination changes, dynamic backgrounds, shadows, and ghosts—issues often encountered by traditional BGS algorithms.

The success of the approach in [16] lies in its real-time segmentation capabilities, achieved through the parallel operation of two components. The standard BGS segmenter is responsible for constructing background models and segmenting foreground objects. Meanwhile, the real-time semantic segmenter refines the foreground segmentation and provides feedback to enhance the accuracy of model updates. This dual-component setup proves effective in addressing the limitations of traditional BGS algorithms while harnessing the strengths of real-time semantic segmentation.

In [17], SuBSENSE serves as a benchmark for the background subtraction segmenter, being recognized for outperforming most real-time unsupervised background subtraction algorithms. On the other hand, ICNet [18]is initially adopted as the benchmark semantic segmenter. ICNet is known for its superior efficiency and accuracy in real-time semantic segmentation, leveraging a multi-resolution cascade network architecture to reduce computational complexity. However, experimental results revealed that the proposed system using ICNet performed poorly. Consequently, the state-of-the-art PSPNet was introduced as a replacement for ICNet, leading to significantly improved results.

2.4 Depth and RGBD Based techniques

In [19], Runzhi Wang introduced a SLAM-based algorithm, which can be divided into three main components: input data processing, moving object detection and elimination, and camera pose estimation. Previously, Alcantarilla proposed a method for dynamic object detection

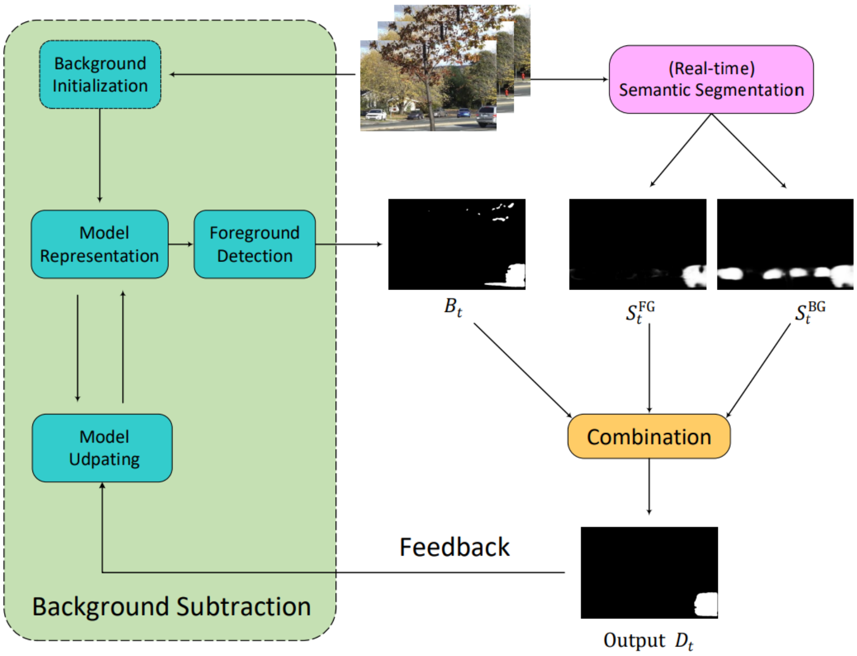


Figure 2.3: Flowchart of the model proposed in [16]

in SLAM in [20], where the identification and elimination of moving objects were achieved through a dense scene flow representation.

However, Alcantarilla's method had limitations, particularly in terms of distance, and struggled with determining static points accurately when the distance increased. Addressing similar challenges, Kim proposed a method in [21] that could estimate the background model from depth scenes and the ego-motion of the camera by mitigating the influence of moving objects.

In [19], the detection of moving objects required both the current and the previous RGB images for the extraction and matching of feature points, along with the current depth image. For the feature extraction and matching approach to be effective, a minimum of 20 pairs or more of matching feature points needed to exist in the static environment. The process involved detecting feature points from two consecutive frames, simultaneously clustering the current depth image, and then mapping the feature points onto the clustered depth image. This comprehensive approach aimed to enhance the accuracy and robustness of moving object detection within the SLAM framework.

In [19], an algorithm based on morphological reconstruction was employed to fill in the holes in the depth image. This step addressed limitations of the depth image on its own. The algorithm detected inliers and outliers using the fundamental matrix constraint, and further removal of outliers was achieved through a second fundamental matrix constraint. The final step involved entering the data into the model designed for detecting moving objects. Comparative analysis with results from DVO [22] and BaMVo [21] demonstrated that the method proposed in [19] outperformed them in various categories, particularly showing preference for indoor dynamic scenes.

In [23], an approach utilizing a Kinect sensor for real-time object tracking was presented. The integration of 3D range and color information was achieved using the depth and color camera intrinsic parameters. However, it's important to note that this method is more focused on tracking a specific region rather than detecting an object. Tracking is accomplished by processing depth pixels with color information.

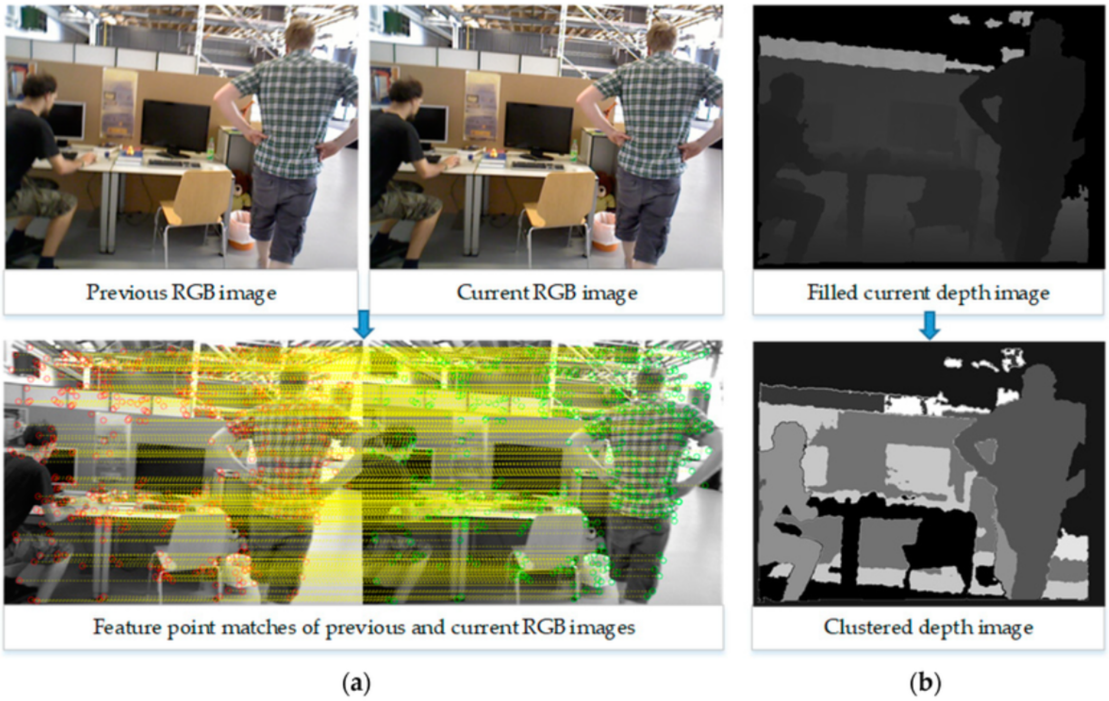

Previous RGB image
Current RGB image
Filled current depth image
Feature point matches of previous and current RGB images
Clustered depth image
(a)
(b)

Figure 2.4: (a) Extraction and matching of Feature points (b) Clustered Depth image

2.4.1 CNN

For the task of image classifi0cation in deep learning, Convolutional Neural Networks (CNNs) stand out as the most popular and widely used architecture. Also known as Space Invariant Artificial Neural Networks, CNNs are specifically designed for processing grid-like data, such as images. In this architecture, the input (image) is taken as a layer that represents, for example, a 10x10 pixel section of the image.

CNNs process input data through convolutional layers, which differ from the typical fully connected layers found in traditional neural networks. Convolutional layers are effective in capturing spatial hierarchies and local patterns within the input data. Additionally, CNNs often incorporate pooling layers to reduce the dimensional complexity of the data.

Fully connected layers, another key component of CNNs, establish connections between each neuron in one layer to every neuron in the subsequent layer. This interconnected structure enables the network to learn intricate features and relationships within the data.

The use of CNNs has greatly simplified the task of image classification, making it more efficient and accurate. The architecture's ability to automatically learn hierarchical representations from input images has led to significant advancements in various computer vision tasks.

2.4.2 Fast R-CNN

In the realm of object detection, relying solely on Convolutional Neural Networks (CNNs) poses computational challenges, especially with the use of the sliding window approach, which can be computationally expensive. To address this, researchers recommended a more efficient approach involving blobby image regions more likely to contain objects. This led to the development of the Region-based Convolutional Neural Network (R-CNN). In R-CNN, CNN is applied selectively to specific regions determined using a Selective Search algorithm, resulting in faster processing.

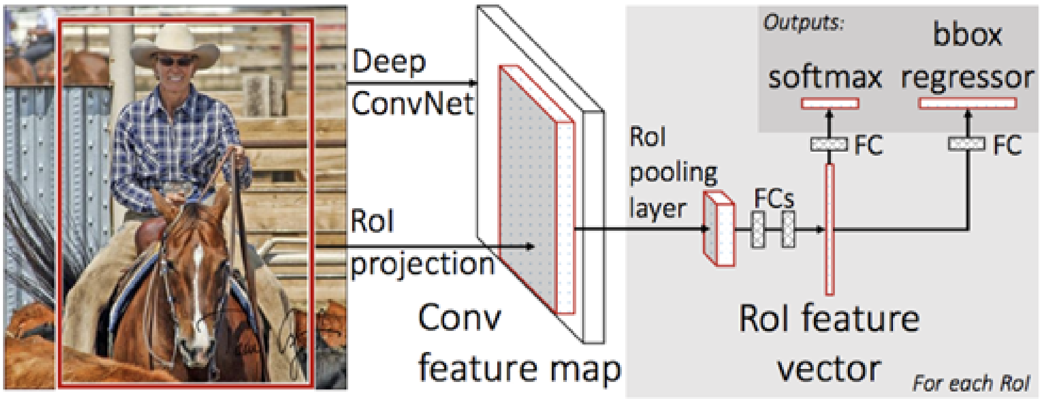


Figure 2.5: Representation of how a Fast R-CNN works

However, the CNN process in R-CNN remains relatively slow. To enhance feature extraction performance, an improvement was introduced in the form of Fast R-CNN. Fast R-CNN improves speed by running only one CNN over an entire image and replacing Support Vector Machines (SVM) with a SoftMax layer.

Further evolution in this line of research resulted in the Faster R-CNN, a more advanced model that replaces the Selective Search algorithm with a fast neural network aided by a Region Proposal Network (RPN). The RPN efficiently proposes regions of interest, significantly reducing the computational demands of the entire process. As a result, Faster R-CNN exhibits superior speed and higher accuracy compared to its predecessor, Fast R-CNN. This progression reflects a continual effort to optimize object detection models for both efficiency and performance. Changbo introduced an approach in [24] where the depth estimation module of SLAM was replaced with Fully Convolutional Networks (FCN). Using a monocular camera with ORBSLAM2, FCN was employed to automatically learn features layer by layer. The adoption of FCN in [24] eliminated Fully Connected (FC) layers, allowing the use of images of any size as input. FCN replaced the depth estimation module of ORBSLAM2 to address the contradiction of triangulation. Additionally, adjusting parameters in Faster R-CNN during training improved the learning rate for indoor object detection. Transfer learning was utilized to fine-tune the Region Proposal Network (RPN) in Faster R-CNN, significantly enhancing the accuracy of object detection.

2.5 Deep Learning Based Techniques

In this section we we discuss the Deep Learning based techniques which are used for image segmentation, object detection and instance segmentation. In the few papers cited in this section it can be seen that Object detection has been researched as most deep learning based techniques are used fro the purpose of object detection and not just image segmentation.

In [25], Lesole presents an approach that addresses various challenges encountered in real-time multi-object tracking systems, leading to enhanced system performance. Lesole provides a comprehensive summary of 95 variations made in deep learning over a span of 5 years. However, a drawback is observed when the system attempts to track and detect objects in overcrowded scenes.

The evolution from convolutional neural networks (CNNs) to deep convolutional neural networks (DCNNs) has significantly bolstered deep learning methods and tracking-by-detection, as documented in the literature. It is suggested that approaches based on a single camera view are more suitable for offline multi-object tracking due to their specific view angle limitations.

Improvements in Single-Camera Multi-Object Tracking

While single-camera multi-object tracking is adept at monitoring multiple objects simultaneously, its one-sided view poses challenges in handling rotations, scaling, affinity distortions, and occlusions. Lee and Hong [26] address this by incorporating separate detectors and classifiers to improve detection performance. Fajrado [20] further contributes to detector performance by labeling objects on the output of the maximal classifiers. However, challenges such as high fragmentation, velocity changes, and appearance alterations persist.

Multi-Camera Tracking and Inter-Camera Object Tracking

In the realm of multi-camera tracking for multiple objects, inter-camera object tracking is employed. DCNNs, along with tracking by detection, are introduced in [20] to address the challenge of associating a target with multiple potential views. A crucial consideration in employing deep learning methods is the quality of detection.

Trajectory Generation in World Coordinate Frame

Scheidegger [27] proposes an approach utilizing a DCNN and a Poisson multi-Bernoulli mixture filter to generate trajectories of detected objects in a world coordinate frame. Deep neural networks are employed to detect the distance of objects from a single image, and these detections are integrated into a Poisson multi-Bernoulli mixture filter. Additionally, a single short multi-box detector is used to reinforce the detection of small objects on deeper layers.

Enhanced Tracking Accuracy and Speed

In [28], Shin introduces an approach incorporating three functional modules, including tracking failure detection, retracking using multiple search windows, and motion vector analysis, onto a kernelized filter-based tracking. This approach enhances both tracking accuracy and speed but requires additional computational time due to the load of multiple search windows.

Kampker [29] proposes a real-time framework for multi-object detection and maneuver-aware tracking, specifically designed for 3D LIDAR applications to address object uncertainty in cluttered urban environments. The technique involves implementing an algorithm that takes a 3D point cloud as input, dividing it into non-ground and elevated measurements.

Wen [30] introduces a method implementing the CLEAR MOT evaluation metric in neurotic work on deep learning-based real-time multi-object tracking methods with multi-camera tracking techniques and DCNNs, employing the tracking by detection approach.

In [31], a new approach called SwiftNet is proposed for real-time video segmentation, aiming to enhance segmentation accuracy. SwiftNet utilizes spatiotemporal redundancy to address challenges in real-time video object segmentation. A light aggregation encoder is employed to improve reference encoding, resulting in highly accurate results through the compression of spatiotemporal redundancy via pixel adaptive memory.

2.5.1 YOLO

Unlike traditional region-based object detection algorithms, YOLO (You Only Look Once) takes a unique approach to localization. Instead of relying on regions of interest, YOLO divides the image into a grid and assigns bounding boxes within this grid. The method involves setting a threshold value, and bounding boxes with a class probability surpassing this threshold are chosen to identify the object in the image.

In essence, YOLO streamlines the object detection process by handling it in a single pass, making it more efficient compared to region-based approaches. The grid-based approach, combined with the use of bounding boxes and a class probability threshold, allows YOLO to detect and locate objects with a simplified yet effective methodology. This innovation has contributed to the popularity of YOLO in the field of object detection.

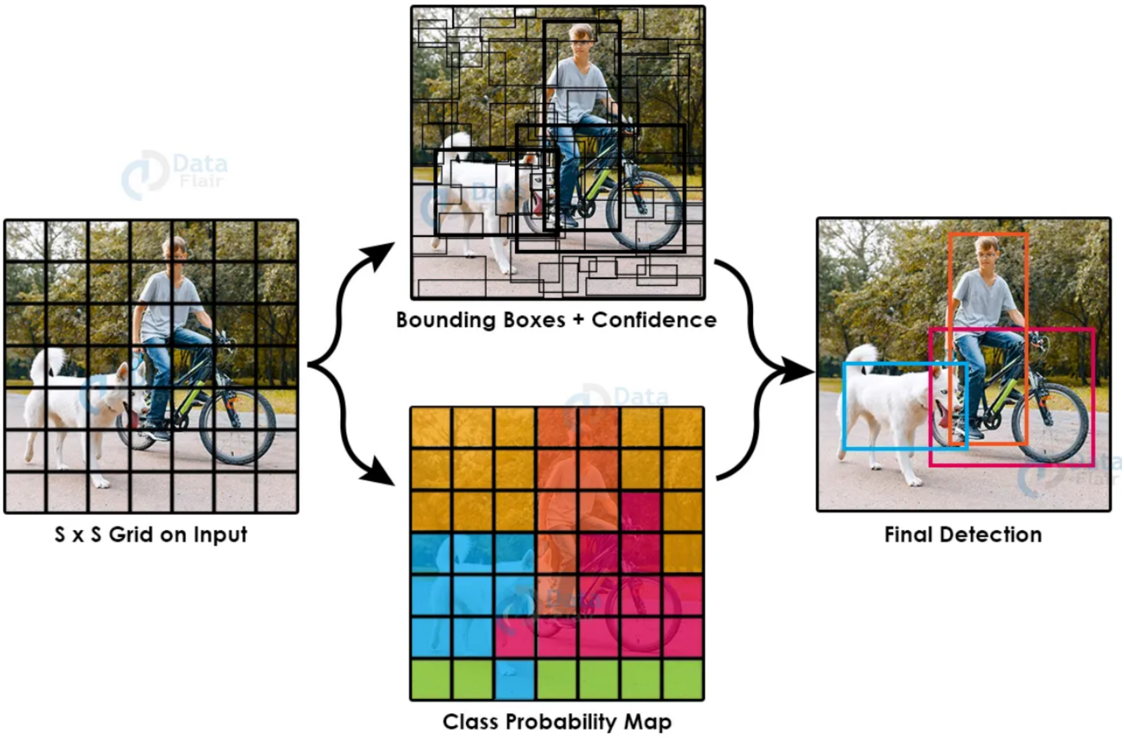


Figure 2.6: : Representation of how a YOLO system works

2.6 Comparison between Deep Learning and Traditional Computer Vision Methods

In recent years, deep learning has exceeded the expectations of computer vision engineers, introducing groundbreaking advancements in digital image processing. This branch of artificial intelligence has enabled the integration of computer vision applications, such as face recognition and photo stylization, into our daily lives. Deep learning's efficiency in processing unstructured data, particularly images and videos, has paved the way for its widespread adoption in various domains. Problems in computer vision once considered unsolvable have found solutions through deep learning methodologies. One notable example is image classification, a challenge deemed too difficult for traditional computer vision techniques.

While many deep learning methods, such as Convolutional Neural Networks (CNNs), leverage big data and substantial computing resources to enhance performance, it's essential to recognize that deep learning doesn't universally address all computer vision problems. Traditional computer vision techniques have demonstrated success in overcoming certain challenges that prove difficult for deep learning-based approaches. In [32], a comprehensive comparison between deep learning-based methods and traditional computer vision techniques provides valuable insights into the strengths and limitations of each approach.

Advantages of Deep Learning over Traditional Computer Vision

Deep learning, a subset of machine learning, is rooted in the concept of Artificial Neural Networks (ANN), which mirrors the functioning of the human brain. Some notable advantages of deep learning over traditional computer vision approaches include:

- Increased Memory Capacity and Computing Power: Advances in memory capacity and computing power contribute to improved performance.

- Less Fine-Tuning and Expert Analysis: Deep learning relies on training rather than traditional programming, reducing the need for extensive fine-tuning and expert analysis.
- Greater Accuracy in Tasks: Deep learning exhibits higher accuracy in tasks like semantic segmentation, object detection, and Simultaneous Localization and Mapping (SLAM).
- Flexibility Through CNN Models: CNN models in deep learning can be re-trained, providing more flexibility.
- Elimination of Feature Descriptors: The need for traditional feature descriptors (e.g., SIFT, SURF) is reduced as deep learning employs end-to-end learning concepts.

Deep learning excels in solving closed-end classification problems, particularly when ample data is available. In [33], Marcus addresses ten major concerns about deep learning and suggests that combining deep learning with traditional techniques can overcome drawbacks and enhance standard deep learning methods' performance. Traditional computer vision techniques contribute to faster training times and reduced data processing requirements in applications like SLAM, panoramic stitching, geometric deep learning, and 3D vision.

Advantages of Traditional Computer Vision Techniques

While deep learning has clear advantages, traditional feature-based approaches, such as the Hough Transform, Geometric Hashing, SIFT, and FAST, also offer strengths:

- Hough Transform: Used for detecting shapes, lines, and curves.
- Geometric Hashing: Applied in matching and recognizing objects in images.
- SIFT (Scale Invariant Feature Transform): Provides scale and rotation-invariant features, suitable for 3D mesh reconstruction.
- FAST (Features from Accelerated Segment Test): Focuses on corner detection in images.

Traditional approaches like SIFT are not class-specific and offer general applicability, making them useful for tasks like 3D mesh reconstruction. In scenarios with limited training datasets for Deep Neural Networks (DNN), traditional techniques may outperform by providing generalization without extensive parameter adjustments. While traditional computer vision methods are established, transparent, and optimized, deep learning emphasizes accuracy and versatility, often demanding significant computing resources. The choice between the two depends on the specific requirements of a given task.

2.7 ERFNet

The conventional paradigms of computer vision often prioritize patch-based object detection, employing diverse algorithms to independently identify objects within an image. ERFNet, however, was purposefully crafted to achieve precise segmentation outcomes while concurrently minimizing computational complexity. This distinctive design renders ERFNet particularly well-suited for real-time applications, where swift processing is paramount.

Factorized Convolutions

In the architecture of ERFNet in [34], factorized convolutions play a central role, executed in two sequential steps. Initially, a 1xk convolution is applied, followed by a kx1 convolution. This factorized convolution methodology effectively diminishes computational costs while preserving the requisite receptive field crucial for capturing spatial information. The result is an optimized balance between accuracy and computational efficiency, making ERFNet an ideal choice for scenarios where real-time processing speed is of utmost importance.

ERFNet leverages residual learning, incorporating residual mappings that facilitate gradient flow during training. This design choice eases network optimization and convergence, addressing challenges such as the vanishing gradient problem. Additionally, ERFNet employs densely connected blocks, promoting efficient feature propagation and enhancing information flow within the network.

Residual learning and Context Aggregation

In the realm of full-image semantic segmentation, the goal is to classify various object classes at the pixel level of an image. Leveraging convolutional neural networks originally designed for image classification, the network proposed in [34] harnesses factorized convolutions to optimize performance while maintaining efficiency. ERFNet employs skip connections to combine features from different scales, allowing for the fusion of information at varying levels of abstraction. This strategy enhances the network's ability to achieve a more holistic understanding of the image.

Furthermore, ERFNet incorporates a spatial pyramid pooling module for context aggregation, contributing to its capacity for comprehensive semantic segmentation. This module enables the network to capture contextual information at multiple scales, further enriching its ability to discern intricate details and relationships within the image.

2.8 ICNet

A significant challenge arising from real-time semantic segmentation lies in the complexity of reducing computation for pixel-wise label inference. To tackle this issue and strike a balance between accuracy and processing speed, ICNet was introduced in [18]. Notably, ICNet demonstrates faster inference times in comparison to many other segmentation networks.

ICNet adopts a cascade architecture comprising three branches that operate on different resolutions of an image—coarse, medium, and fine. These branches operate concurrently, facilitating simultaneous multiscale feature extraction. Within this network, pyramid pooling is implemented in a cascading manner, involving a series of pooling operations. These operations employ various kernel sizes to capture multiple scales within each branch, resulting in the identification of spatial hierarchies within ICNet.c

Adjusting kernel sizes based on different scales is imperative in the pooling process. In this regard, the coarse branch strategically employs a downsampled version of the input image to enhance the efficiency of capturing global context. This entails utilizing pooling operations with larger kernel sizes, ensuring coverage of more extensive spatial regions. Conversely, the medium and fine branches adopt pooling operations with smaller kernel sizes, given their utilization of intermediate and full-resolution images as inputs, respectively. This choice preserves finer details within the segmentation.

To address these considerations, an image cascade network (ICNet) has been proposed, introducing multi-resolution branches guided by appropriate labels. This architectural approach effectively navigates the challenge of balancing global context capture and preservation of finer details in semantic segmentation.

2.9 BiSeNet

The approach introduced in [35] tackles the common issue of sacrificing spatial resolution for the sake of real-time inference speed, often resulting in diminished performance. To overcome this challenge, the Bilateral Segmentation Network (BiSeNet) has been proposed. The model presented in [] establishes a more optimal equilibrium between speed and segmentation performance, addressing the trade-off and enhancing overall effectiveness.

2.9.1 Bilateral Segmentation

The work presented in [BiSeNet] introduces the Bilateral Segmentation Network, featuring both a Spatial Path and a Context Path. The paper not only introduces these paths but also delves into an exploration of their individual effectiveness. Furthermore, the synergistic combination of these paths, along with the Future Fusion module, is demonstrated in [35].

BiSeNet stands out for its distinctive ability to capture both local and global context information, achieved through the utilization of the Spatial Path and the Context Path. This dual-path structure is pivotal in establishing BiSeNet's robust standing, effectively balancing segmentation accuracy and computational efficiency.

Spatial and Context Paths: Capturing Local and Global Context

A Spatial Path is introduced to preserve the spatial dimensions of the input image and encode rich spatial information, primarily focusing on capturing local context details. Operating at a lower spatial resolution, this path ensures the precise delineation of object boundaries. Comprising three layers, each consisting of a convolution followed by batch normalization, the output feature map obtained from the spatial path is 1/8th the size of the original input image.

In contrast, the Context Path operates at a higher spatial resolution and is designed to capture global context information, fostering an understanding of relationships between various objects and structures in the image. Unlike the Spatial Path, which encodes abundant spatial information, the Context Path emphasizes the acquisition of a sufficient receptive field—a crucial factor for semantic segmentation performance. Leveraging a lightweight model and global average pooling, the Context Path achieves a large receptive field. Certain lightweight models efficiently down-sample feature maps to obtain a significant receptive field, encoding high-level semantic context information. The global average pooling at the end of the lightweight model provides the maximum receptive field with global context information.

Fusion of Local and Global context Enhanced Segmentation Performance

The fusion of local and global context, facilitated by combining the up-sampled output feature of global pooling with the features of the lightweight model, enhances overall segmentation performance. To refine features at each stage, a specific Attention Refinement Module is proposed within the Context Path. This module enables the network to focus on more relevant parts of the image, thereby improving segmentation quality. Additionally, an Attention Guidance Module is incorporated to direct the network's attention to regions where detailed information is critical for accurate segmentation. Together, these components contribute to a comprehensive and refined approach to semantic segmentation.

2.10 PSPNet

One of the more fundamental topics in the field of computer vision is semantic segmentation based on scene parsing. Through the implementation of scene parsing, it becomes possible to predict the location, label, and shape of each element within a given scene. Optimally, scene parsing networks are structured on fully convolutional networks (FCN). The [36] delves into harnessing the potential of global context information, achieved through region-based context aggregation facilitated by the Pyramid Pooling Module, in tandem with the Pyramid Scene Parsing Network (PSPNet).

2.10.1 Pyramid Pooling

The necessity for pyramid pooling arises due to the inherent challenge in networks like ResNet, where receptive fields extend beyond the dimensions of the input image. Consequently, these networks struggle to effectively integrate global contextual information. Pyramid pooling, conceived for the explicit purpose of enhancing a network's proficiency in comprehending both local intricacies and the broader global context of an image, addresses this limitation. It can be argued that factors such as multiscale information and context awareness play pivotal roles in motivating the adoption of pyramid pooling.

Bridging Local and Global Context

Pyramid pooling entails the segmentation of the input feature map into distinct regions of varying sizes, capturing features from each region independently. This is accomplished through the application of average pooling or max pooling to each region separately. By employing pooling operations with diverse kernel sizes and accounting for different scales, pyramid pooling empowers the network to aggregate information from both local and global perspectives. The features extracted at these varied scales are then harmoniously concatenated to form a holistic and comprehensive representation.

A more in-depth exploration of the implemented PSPNet unfolds as follows. The pyramid pooling module intricately integrates features across four distinct pyramid scales. The foremost level, highlighted in red in the accompanying figure, symbolizes global pooling, generating a singular bin output. Directly beneath it, the subsequent pyramid level segregates the feature map into diverse sub-regions, crafting pooled representations for various locations. The outputs from these diverse pyramid levels contain feature maps of varied sizes.

Notably, the weight of the global feature is meticulously preserved through the utilization of a 1x1 convolutional layer after each pyramid level. This strategic step serves to diminish the dimension of the contextual representation to 1/N of the original size, where N denotes the size level of the pyramid. Subsequently, the low-dimensional feature map undergoes direct upsampling to match the size of the original feature map. In culmination, all distinct levels are harmoniously concatenated, forming the ultimate pyramid pooling global feature.

Adaptive Architecture

It's crucial to highlight that the number of pyramid levels and their respective sizes are adaptable, allowing for modification based on the size of the feature map fed into the pyramid pooling layer. This dynamic feature empowers tailored adjustments for optimal performance in diverse scenarios. The architecture of PSPNet and a brief discussion about the advantages of implementing it have been discussed in the next chapter.

2.11 Spatial Information Based Method

The algorithm presented here leverages spatial information to accomplish real-time object detection goals. Some segmentation approaches rely on RGB and 3D spatial information independently, resulting in a two-stream network. However, this two-stream segmentation approach poses a significant limitation on real-time applications.

In [37], a novel approach is introduced that enables real-time utilization of both RGB and spatial information. This approach, known as Spatial Information Guided Convolution (S-Conv), facilitates the seamless integration of RGB and 3D spatial information. Recent advancements in Convolutional Neural Networks (CNNs) have demonstrated improved performance in indoor segmentation tasks. The geometric structure of objects plays a crucial role in segmentation tasks.

The S-Conv method proposed in [37] dynamically adjusts to changes in spatial information. It involves generating convolution kernels with different sampling distributions tailored to spatial information, enhancing the spatial adaptability and receptive field regulation of the network. The S-Conv method has shown superiority over other two-stream methods, reducing the number of parameters and computational time required. The adaptability of S-Conv is demonstrated by the success of the Spatial Information Guided Convolutional Network (SGNet) on the tested datasets.

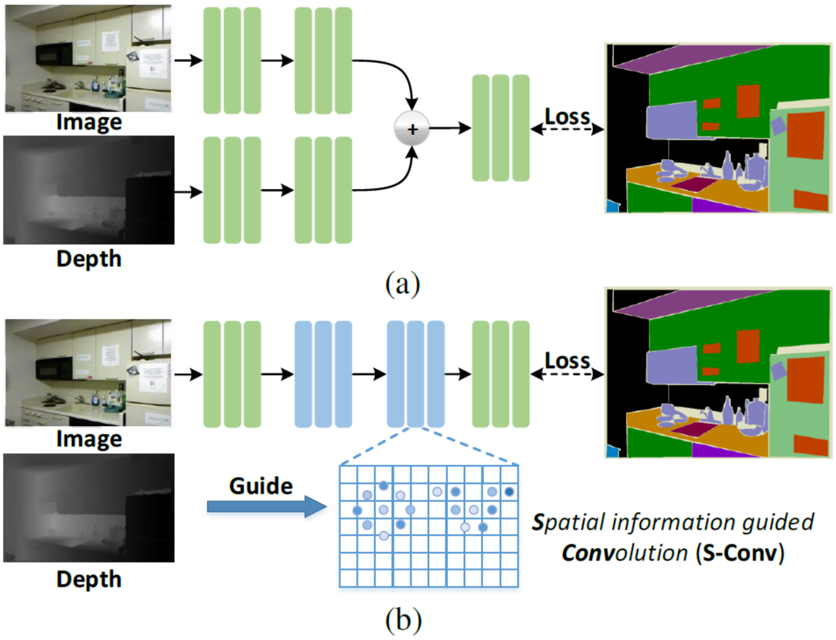


Figure 2.7: (a) Conventional structure of the two-stream network (b) S-Conv is used to replace normal convolution in the two-stream network (SGNet)

The core architecture for the semantic segmentation network consists of S-Conv, a backbone network, and a decoder. Atrous convolution serves as the backbone in the proposed SGNet. The experimental results in [37] demonstrate real-time inference capabilities on the tested datasets. Additionally, the depth-adaptive receptive field is visualized in each layer to illustrate its effectiveness. SGNet surpasses the performance of many other convolutional networks and proves to be a viable solution for real-time applications.

Enhancing Semantic Segmentation with S-Conv and ResNet

Spatial Information Guided Convolution (S-Conv) can be characterized as an evolution of the traditional RGB-based convolution. What sets S-Conv apart is its incorporation of spatial information in the RGBD scenario.

The process of S-Conv involves generating an offset based on the spatial information, specifically the RBGD data. Subsequently, this spatial information, aligned with the previously generated offset, is utilized to formulate new spatially adaptive weights. Leveraging RGBD data enhances the efficacy of semantic segmentation tasks. The attention mechanisms integrated into spatial guided convolution play a pivotal role in adjusting the weights of convolutional filters, dynamically emphasizing the significance of spatial regions in the input.

Convolutional Neural Networks (CNNs) hold a paramount position in the realm of image processing, often considered the backbone for tasks like object detection. However, as more layers are added to a CNN, a saturation point is inevitably reached where accuracy plateaus. This saturation poses a challenge in tasks such as object detection, where accuracy is pivotal for successful outcomes.

Evolution of Convolution: Integrating Residual Networks

To address the challenge of accuracy degradation in deep convolutional networks, ResNet was developed, introducing skip connections and batch normalization. An additional advantage of employing the ResNet model is its departure from false color image processing, opting for average pooling instead. This not only prevents overfitting but also significantly enhances precision. Average pooling computes the average for each patch of the feature map.

ResNet, short for residual network, is an artificial neural network that strategically stacks residual blocks to form a network. In conventional deep convolutional networks, increasing the number of stacked layers enriches the features of the model. However, as mentioned earlier, this can lead to accuracy degradation. ResNet aims to solve this problem by utilizing skip connections in two ways.

Skip connections serve to mitigate the vanishing gradient problem and enable the model to learn an identity function. Consequently, higher layers of the model exhibit better performance than the lower layers. ResNet101 and ResNet50 are among the more widely used variants. The ResNet101 architecture, for instance, is composed of 3-layer blocks. The SGNet algorithm leverages the ResNet101 architecture, benefiting from its skip connections and effective layer organization to enhance the network's overall performance.

2.12 Summary Of The Literature Review

This literature survey delves into various approaches for object detection, emphasizing the need for real-time performance. The spectrum of techniques explored ranges from traditional computer vision, like background subtraction, to more contemporary methods, such as deep learning. For real-time applications, the preference is toward leveraging deep learning, particularly due to its efficiency in processing pretraining data, which is particularly beneficial for indoor scenes, typical in this assignment's context.

While deep learning is favored for its efficiency, it alone may not suffice for real-time applications, especially when prioritizing accuracy, a crucial aspect in overcoming challenges discussed in the drawbacks section. The traditional computer vision approaches, though detailed, prioritize accuracy over processing speed.

Several methods are evaluated for their benefits and limitations. A reasonable suggestion based on research is to integrate a deep learning approach with traditional computer vision methods, addressing limitations and proposing a hybrid method. Existing research supports the effectiveness of such hybrid approaches.

PSPNet, ICNet, ERFNet, BiSeNet, and SGNet are all deep learning models commonly used for segmentation tasks, each offering its own unique advantages suited for a sepcific scenario . PSPNet excels in accurately segmenting complex scenes by capturing contextual information at multiple scales, making it ideal for tasks such as scene understanding. ICNet is preferred for real-time segmentation tasks due to its efficient cascade architecture, making it suitable for applications like video surveillance and robotics. ERFNet stands out for its lightweight design and efficient residual factorized convolutions, making it well-suited for deployment on resource-constrained devices in tasks such as traffic sign recognition and pedestrian detection. BiSeNet combines spatial and contextual information effectively, achieving high segmentation accuracy with low computational cost, making it suitable for real-time applications like image editing and video analytics. SGNet gives importance to informative gradients to improve segmentation accuracy, making it ideal for tasks such as medical image segmentation and industrial inspection. Each model offers distinct advantages and is selected based on specific requirements, including segmentation accuracy, computational efficiency and real-time performance.

3 Analysis

The primary objective of this assignment is to integrate a new segmentation module into the James robot setup. Unfortunately, several hardware-related issues have arisen, and these challenges are extensively discussed in the drawbacks section. Currently, our approach involves the implementation of five distinct neural network models to achieve segmentation. The selection of these models was not arbitrary; rather, it was based on their unique structures, capabilities, and designs, as detailed in the sections below.

Although SGNet was initially intended for implementation in the James robot setup, technical difficulties prompted a comprehensive analysis of five different segmentation models. This analysis aims to compare their effectiveness and suitability for the specific requirements of this setup. The models under consideration for this comparative study include PSPNet, ICNet, ERFNet, and BiSeNet and SGNet. Through this evaluation, we aim to identify the model that is best aligned with the needs and challenges posed by the James robot when it comes to image segmentation.

Considering the research questions it can be concluded that they delve into how to combine deep learning and traditional computer vision techniques which affect segmentation accuracy and real-time performance in dynamic object segmentation tasks. They also take into how the computational hurdles involved in achieving real-time image segmentation addresses these challenges. They also consider the influence of evaluation metrics like accuracy, precision, recall, and processing time on the comparison of image segmentation algorithms, and determine the criteria for selecting appropriate metrics. By considering the the trade-offs between segmentation precision, real-time execution speed and by exploring strategies for effectively managing these trade-offs to meet the requirements of applications demanding both accuracy and promptness in segmentation.

3.1 ERFNet

3.1.1 ERFNet Architecture

The network architecture presented in [] has been meticulously crafted in a sequential manner, wherein layers are stacked based on our innovative redesign of the residual layer. ERFNet adopts an encoder-decoder structure, eliminating the necessity for skip layers in refining the output. The encoder segment employs a more sequential architecture, generating down-sampled feature maps, followed by a decoder segment that up-samples these features to match the original input resolution.

While down-sampling introduces the challenge of reduced pixel accuracy, it concurrently offers the advantage of allowing deeper layers to capture more context, thereby reducing computation. The decoder component is responsible for up-sampling the feature maps to align with the input resolution. The down-sampler block achieves down-sampling by concatenating the parallel outputs of a 3x3 convolution and a max pooling module. Additionally, some layers incorporate dilated convolutions, enhancing classification accuracy by gathering more contextual information. This strategic combination of architectural elements contributes to the efficacy of ERFNet in semantic segmentation tasks.

3.2 ICNet

3.2.1 ICNet Architecture

The challenge of striking a delicate balance between inference accuracy and speed is highlighted in the time budget analysis conducted in [18]. Addressing this challenge, the proposed image cascade network [18], as illustrated in the accompanying figure, introduces a novel approach involving cascade image inputs. Simultaneously, it incorporates a cascade feature fusion unit and undergoes training with cascade label guidance.

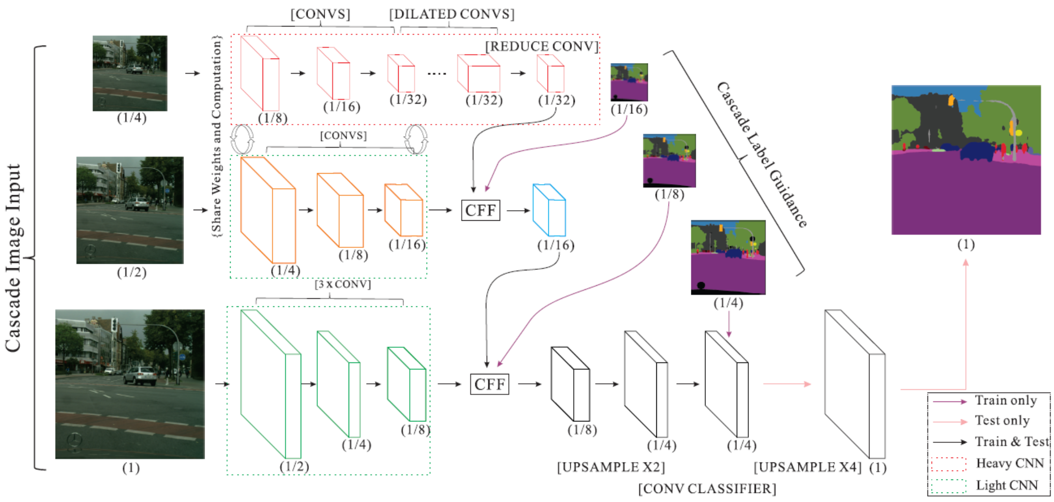


Figure 3.1: A detailed view of the architecture in ICNet

In the architectural depiction, the input image undergoes downsampling to create a cascade input for both medium and high-resolution branches. This approach enables semantic extraction using a lower resolution input, mitigating time consumption drawbacks. A 1/4th-sized image is directed to PSPNet, resulting in a 1/32 resolution feature map. The medium and high-resolution branches play a crucial role in recovering and refining the coarse prediction, ultimately achieving high-quality segmentation. The adoption of lightweight CNNs in the higher resolution branches contributes to computational efficiency.

The cascade feature fusion unit is pivotal, harmonizing the output feature maps from different branches through training guided by cascade labels. Finally, upsampling is employed to generate the ultimate segmentation map, completing the intricate process of achieving a balance between inference accuracy and speed in the image cascade network.

3.3 BiSeNet

3.3.1 BiSeNet Architecture

In the provided figure, the architecture of BiSeNet is showcased. In this design, the pretrained Xception model serves as the backbone for the context path, while three convolution layers with stride contribute to the spatial path. The final prediction is crafted by fusing the output features from these two paths, achieving a remarkable blend of real-time performance and high accuracy.

The spatial path, despite possessing a large spatial size, consists of only three convolutional layers, ensuring it remains computationally efficient. On the other hand, the context path employs a lightweight model for swift downsampling. The concurrent computation of both paths enhances overall efficiency. The feature fusion module plays a crucial role in addressing accuracy challenges encountered by both the spatial and context paths. Additionally, a loss function is employed to supervise the training of the method.

BiSeNet is meticulously designed with a primary focus on efficiency, making it particularly well-suited for real-time applications. The architecture achieves a commendable balance between computational cost and segmentation accuracy. Notably, the design of BiSeNet facilitates fast

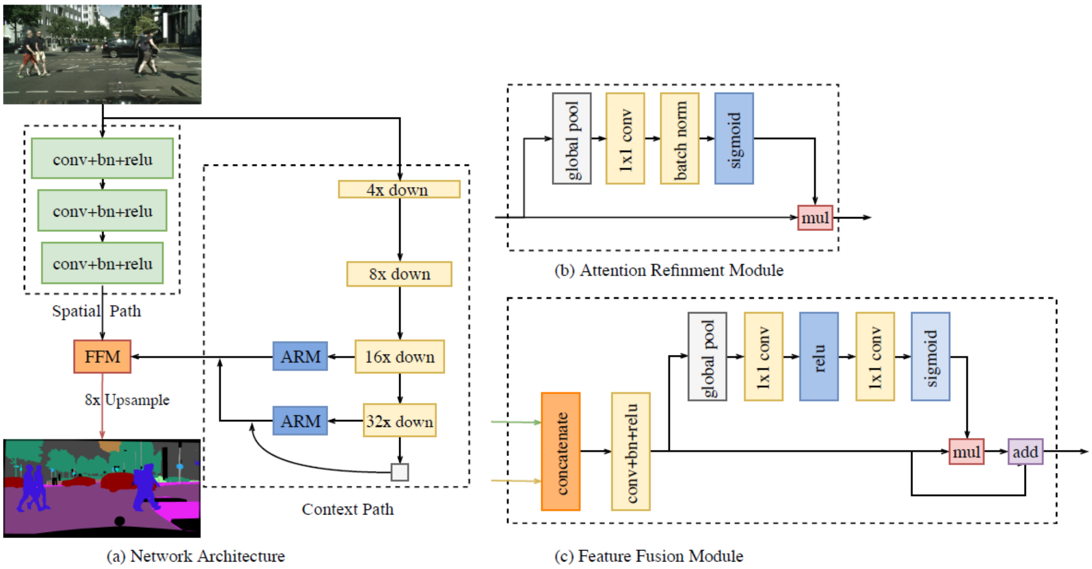


Figure 3.2: Overview of the network architecture in BiSeNet

inference, rendering it highly practical for applications where low-latency segmentation is of paramount importance.

3.4 PSPNet

3.4.1 PSPNet Architecture

In the accompanying figure, the network architecture of the Pyramid Scene Parsing Network is depicted. The PSPNet is conceptualized on the foundation of the pyramid pooling module. Initiated by a convolutional neural network, the feature map of the final convolutional layer is obtained.

Following the acquisition of the feature map, a pyramid pooling module comes into play, orchestrating the extraction of context information by capturing diverse sub-region representations. This process is succeeded by upsampling and concatenation layers, culminating in the creation of the definitive feature representation. Noteworthy is the comprehensive coverage of the entire image achieved through the implementation of a 4-layer pyramid.

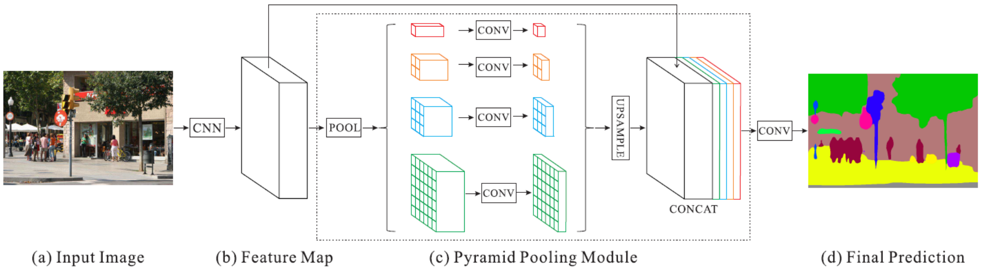


Figure 3.3: An overview of the approach implemented in PSPNet

Subsequently, this representation undergoes processing in a convolution layer, ultimately yielding the final prediction map. Significantly, the size of this final feature map is precisely 1/8 of the input image. Importantly, the computational cost of PSPNet remains consistent when compared to the original dilated FCN network.

PSPNet's efficacy is amplified by the incorporation of global context integration, furnishing a more holistic comprehension of the entire image. This integration renders the network resilient to variations in object sizes and positions by employing pooling features across multiple scales. Notably, the application of pyramid pooling plays a pivotal role in enhancing semantic segmentation performance. This enhancement is attributed to a refined understanding of object boundaries and relationships, which proves indispensable in this context.

3.5 SGNet

3.5.1 SGNet Architecture

This section provides a detailed insight into the architecture of the semantic segmentation network designed to enhance segmentation. The input to the SGNet model is an RGB image. The SGNet method is structured around a Spatial convolution backbone and a decoder.

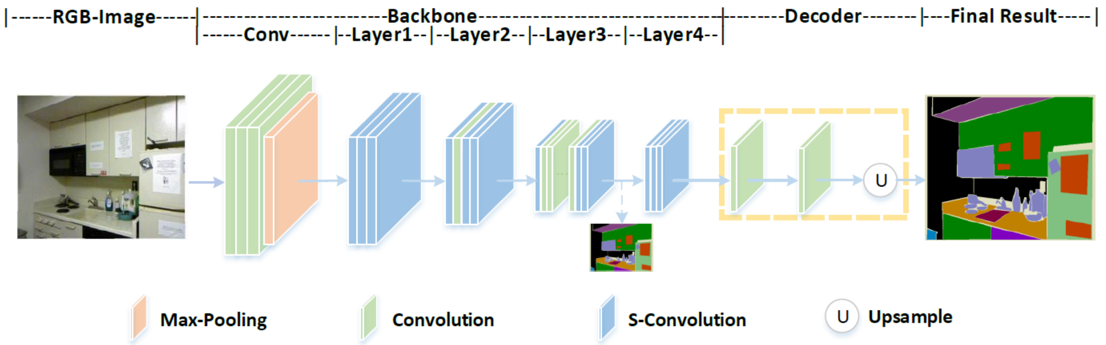


Figure 3.4: The network architecture of SGNet equipped with S-Conv for RGBD semantic segmentation

In this method, ResNet101 serves as the backbone, but with a modification wherein the two convolutions at the beginning and the end are Spatial convolutions. The rationale behind incorporating the ResNet model in SGNet has been extensively discussed in the preceding section. To further aid in network optimization, deep supervision is introduced between layer 3 and layer 4. This approach bears similarities to the architecture of the PSPNet model, as discussed in the literature review.

To obtain the final segmentation probability map, the features extracted from the series of convolutions undergo bilinear up-sampling. This up-sampling process, representing the decoder, utilizes bilinear interpolation to reconstruct the feature map to the original resolution. The output of this process is the segmented image, where each pixel is assigned a probability of belonging to a specific class. This architecture leverages the power of deep convolutional networks, particularly ResNet101, and incorporates spatial convolutions and deep supervision to improve the accuracy and performance of semantic segmentation.

4 Experimentation

4.1 Comparison of different aspects of models

When comparing all the aforementioned discussed models we have taken certain factors into account. In this section we will discuss all the differences between these models with respect to these factors. The main factors taken into consideration here are:

1. Semantic Segmentation Performance: Intersection over Union (IoU): IoU measures the overlap between the predicted segmentation masks and the ground truth masks. A higher IoU score indicates a more accurate delineation of object boundaries. It is a pivotal metric for evaluating the precision of segmentation.

   Mean Intersection over Union (mIoU): mIoU is the average IoU calculated across all classes in the dataset. It provides a single scalar value that represents the overall performance of the segmentation model across different classes. mIoU is often used as a summary metric to assess the segmentation accuracy comprehensively, taking into account the performance across all individual classes.

   Pixel Accuracy: Pixel accuracy provides a straightforward measure of overall correctness in pixel-wise predictions. It is valuable for understanding the general accuracy of the segmentation across the entire image. Both these metrics collectively reveal the models' efficacy in correctly identifying and delineating objects within images.

2. Model Efficiency: Inference Speed: Inference speed measures the time it takes for a model to process an input image and generate segmentation results. Faster inference is critical for applications requiring quick decision-making, such as autonomous vehicles or real-time video analysis.

   Computational Resources: Consider the hardware requirements, including the number of parameters and the size of the model. Smaller models with fewer parameters are generally more resource-efficient.

3. Training and Inference Setup: Training Time: Assess the time required to train the models to convergence. Faster training can be beneficial, especially when dealing with large datasets. Pretraining Requirements: Examine whether models require pretraining on external datasets like ImageNet. Training from scratch may be preferred in certain scenarios.

4. Dataset and Evaluation Metrics: Datasets Used: Ensure that the models are evaluated on similar datasets or benchmark datasets such as Cityscapes, Pascal VOC, or COCO. Consistency in datasets allows for fair comparisons. Evaluation Metrics: Besides IoU and pixel accuracy, consider other relevant metrics such as F1 score, precision, and recall to understand the models' performance from different perspectives.

5. Architectural Considerations: Network Architecture: Understand the architectural choices made in each model, such as the use of encoder-decoder structures, skip connections, or specific modules (e.g., spatial pyramid pooling). Backbone Networks: Examine the type of backbone networks used, whether they are based on popular architectures like ResNet, Xception or custom-designed.

6. Accuracy vs. Speed Trade-off: Assess how well each model balances segmentation accuracy with computational efficiency. Some models may sacrifice a bit of accuracy for significantly faster inference.

7. Qualitative Evaluation: Visual Inspection: Conduct a qualitative assessment by visually inspecting segmentation results. Check whether the models handle challenging scenarios, such as fine details, object boundaries, and diverse scenes.

Motivation for factors of Comparison

Only a few metrics were considered for evaluation here for the purpose of simplicity. Considering too many metrics would complicate the matter of choosing the most optimum suited algorithms as the simulations performed did differ in some scenarios. Metrics such as Intersection over Union (IoU) and Mean IoU (mIoU) provide insights into how well each of these models can accurately classify pixels into different semantic categories. Since semantic segmentation involves pixel-level predictions, these metrics help quantify the model's ability to segment objects accurately. Semantic segmentation models are often deployed in real-time applications where computational efficiency is crucial. The complexity of the model architecture is considered here as it can impact both performance and computational efficiency. Simpler models with fewer parameters may offer faster inference times and reduced computational overhead, making them suitable for resource-constrained environments. With regards to robustness here it refers to the model's ability to generalize well to unseen data and handle various challenging scenarios. Metrics such as accuracy on different datasets or under different environmental conditions(illumination changes for instance) can assess the robustness of segmentation models. Models that perform consistently well across diverse datasets and conditions are always preferred. In addition to overall accuracy, it's essential to evaluate the quality of semantic segmentation results qualitatively. Visual inspection of segmentation outputs can help identify any artifacts, inaccuracies, or limitations of the models, providing complementary insights to quantitative metrics.

4.2 Implementation of the models

In this chapter, the detailed implementation of prominent segmentation models, namely PSPNet, ICNet, BiSeNet, and ERFNet, clarified. The objective is to provide a comprehensive understanding of the practical realization of these models within the scope of our project. The process involves an account of configuring the models and optimizing parameters. The intricacies of the implementation are discussed to offer insights into the technical aspects of the segmentation module development. The results from these models will be discussed in the conclusions section and the most suitable method which can be used for the James robot will be recommended.

4.2.1 ERFNet

For our experimentation, we utilized the Cityscapes dataset, a comprehensive collection capturing diverse urban scenes. Widely recognized for its challenging scenarios and featuring 19 labeled classes, the Cityscapes dataset stands as a prominent benchmark for semantic segmentation tasks. The dataset consists of a training set with 2,975 images, a validation set with 500 images, and a test set with 1,525 images. While direct inspection of test labels is not available, evaluation can be performed through an online test server. Notably, the model was exclusively trained on the fine annotations of the training set, deliberately avoiding pre-training on larger datasets like ImageNet to maintain simplicity and gauge the network's inherent capacity.

During the training process, we initiated the model from scratch, incorporating simple yet effective data augmentation techniques such as random horizontal flips and translations (0-2 pixels in both axes). The assessment of pixel-level accuracy relied on the Intersection-over-Union (IoU) metric. Training was carried out with an inference resolution of 1024x512, and the output was rescaled to the original dataset resolution for evaluation using simple interpolation.

Class weighing, following the technique proposed in [9] with parameters c = 1.10, was employed to enhance results. The training strategy involved initially training the encoder segment with downsampled annotations. Subsequently, the decoder was attached to continue end-to-end training, producing segmentation with the same resolution as the input. This dual-stage setup allowed both segments to converge between 200-250 epochs. Although training the decoder directly from scratch (without separately training the encoder) is viable, our experiments indicated slightly lower performance under this approach.

Results

Results for this architecture, evaluated at a resolution of 2048x1024 on the Cityscapes dataset, demonstrate compelling performance. The architecture, trained from scratch without pretraining on external datasets like ImageNet or Pascal, achieves a mean Intersection-over-Union (IoU) of 68% across 19 classes and an impressive 86.5% IoU specifically for 7 categories. The forward time for a single Titan X (Maxwell) GPU is reported at 24 ms.

When comparing the approach with other methods, it becomes evident that the architecture strikes an optimal balance between segmentation accuracy and computational efficiency. Notably, it outperforms many efficiency-focused approaches in accuracy while maintaining competitive efficiency, allowing real-time processing on a single GPU.

Several top-accuracy methods, such as RefineNet, FRRN, and Deeplabv2, utilize complex architectures with large ResNets and multiple pipelines, demanding significant computational resources. In contrast, this architecture demonstrates efficiency without compromising accuracy, showcasing its superiority in achieving an optimal trade-off.

Furthermore, the capability of the architecture to achieve these results without relying on pretraining on additional datasets, such as ImageNet, underscores its simplicity in design and training. While pretraining on ImageNet could potentially enhance accuracy, the architecture's direct training from scratch with fine Cityscapes annotations ensures a streamlined and efficient training process.

In qualitative assessments, it becomes evident that while other networks like ENet might accurately segment the road immediately ahead of the vehicle, they may provide coarser predictions for distant objects or those requiring finer pixel-level accuracy. In contrast, the architecture consistently delivers accurate results for all classes, even for distant objects in the scene. The reported IoU metrics, albeit challenging, take into account the confusion between all classes and aim to balance the impact between small and large classes. Despite a mean IoU of 68% and a category IoU of 86.5%, the total pixel accuracy exceeds 95%, highlighting the qualitative excellence of our segmentation results.

4.2.2 ICNet

Description of Experiment

This method is specifically crafted for handling high-resolution images and is evaluated on two demanding datasets: the Cityscapes urban-scene understanding dataset, featuring images with a resolution of 1024 × 2048, and the COCO-Stuff understanding dataset, with images reaching up to 640 × 640 resolution. The implementation is performed on the Caffe platform.

To assess the forward inference time, we utilize the 'Caffe time' tool, conducting 100 iterations to minimize accidental errors. Parameters in batch normalization layers are seamlessly integrated into neighboring front convolution layers. Regarding training of the hyperparameters, the mini-batch size remains fixed at 16, with a base learning rate of 0.01, and a 'poly' learning rate policy is adopted with a power of 0.9. The maximum iteration number is set at 30K for both Cityscapes and COCO-Stuff datasets. Momentum is established at 0.9, and weight decay at 0.0001. Data augmentation includes random mirror and random resizing between 0.5 and 2.

The Cityscapes dataset poses a significant challenge due to its high-resolution images (1024 × 2048). It comprises 5,000 meticulously annotated images, divided into training, validation, and testing sets, with 2,975, 500, and 1,525 images, respectively. To enhance processing speed, three key aspects are considered: downsampling the input, downsampling features, and model compression. Downsampling the input image resolution emerges as a crucial factor influencing running speed. An initial approach involves using a smaller resolution image as input.

Experimentation includes downsampling the image with ratios of 1/2 and 1/4, feeding the resulting images into PSPNet50, and directly upsampling the prediction results to the original size. Additionally, an alternative approach involves scaling down the feature map by a considerable ratio during the inference process. Experiments were conducted using PSPNet50 with downsampling ratios of 1:8, 1:16, and 1:32.

Results

The results after downsampling are seen in the 4.1.

Down Sample Size	mIoU%	Time(ms)
8	71.7	446
16	70.3	177
32	67.2	131

Table 4.1: Results of Downsampling

This approach involves a compromise between prediction accuracy and faster inference, as processing a smaller feature map can be done more quickly. However, this trade-off leads to some loss of information, especially in the detailed content of lower-level layers. Even with the smallest resulting feature map at a 1:32 ratio, the system still requires 131ms for inference. In the analysis of cascade branches, its been established that the half-compressed PSPNet50 is used as a baseline, resulting in an inference time of 170ms with mIoU reduced to 67.9%. This indicates that relying solely on model compression has limited potential for achieving real-time performance while maintaining acceptable segmentation quality. Building on this baseline, ICNet has been evaluated on different branches.

To demonstrate the effectiveness of the proposed cascade framework, the outputs of the low, medium, and high-resolution branches are labelled as ’sub4’, ’sub24’, and ’sub124’, respectively. ’sub4’ uses only the top branch with low-resolution input, ’sub24’ combines the top two branches, and ’sub124’ involves all three branches. Testing these configurations on the Cityscapes validation set produces the results shown in the 4.2.

PSPNet50	mIoU(%)	Time(ms)	Frame(fps)
Baseline	67.9	170	5.9
Sub4	59.6	18	55.5
Sub24	66.5	25	40
Sub124	67.8	33	30.2
ICNet	69.5	33	30.3
ICNet(Fine and coarse)	70.6	33	30.3

Table 4.2: ICNet model results

Using only the low-resolution input branch (’sub4’) leads to faster processing but a decrease in result quality to 59.6%. Incorporating two and three branches (’sub24’ and ’sub124’) increases mIoU to 66.5% and 67.8%, respectively, with a slight increase in processing time. Remarkably, our segmentation quality almost matches the baseline while achieving a 5.2 times speedup. Additionally, memory consumption is significantly reduced by 5.8 times.

In the ablation study,the cascade feature fusion unit and cascade label guidance in the cascade structure has been explored. Compared to deconvolution layers with 3 × 3 and 5 × 5 kernels, the cascade feature fusion unit achieves higher mIoU performance with similar inference efficiency. Furthermore, compared to deconvolution layers with a larger kernel size of 7 × 7, the cascade feature fusion unit provides comparable mIoU performance while achieving faster processing speed.

Results show that with a scaling ratio of 0.25, although the processing time significantly decreases, the prediction map becomes coarse, missing many small yet crucial details compared to higher resolution predictions. With a scaling ratio of 0.5, the prediction recovers more information, but the processing time remains impractical for real-time systems. This highlights the trade-off between speed and detailed segmentation accuracy within the Cityscapes context.

When comparing methods, the mIoU performance and processing time of ICNet on the Cityscapes test set are listed. ICNet achieves an mIoU of 69.5%, surpassing several methods prioritizing accuracy over speed, including Enet and SQ by approximately 10 points. With training on both fine and coarse data, the mIoU performance is boosted to 70.6%. Remarkably, ICNet achieves a processing speed of 30 frames per second on 1024 × 2048 resolution images using only one TitanX GPU card.

4.2.3 BiSeNet

Experiment description

We evaluated our proposed BiSeNet on the Cityscapes and COCO datasets, employing a network architecture with three convolutions for the Spatial Path and utilizing the Xception39 model for the Context Path. The Feature Fusion Module is instrumental in combining features from these two paths to generate the final results, with the output resolution of the Spatial Path and the ultimate prediction being 1/8 of the original image.

For training, mini-batch stochastic gradient descent (SGD) was utilized with a batch size of 16, a momentum of 0.9, and weight decay of $1e^{-4}$. The "poly" learning rate strategy was applied, where the initial rate is multiplied by $1 - \frac{iter}{max_iter}^{powerofeachiteration}$, with a power of 0.9. The initial learning rate was set to $2.5e^{-2}$.

During training, augmentation techniques such as mean subtraction, random horizontal flip, and random scaling on the input images, with scales 0.75, 1.0, 1.5, 1.75, 2.0 were applied. Additionally, random cropping of the image was performed to achieve the desired size.

In the baseline approach, the Xception39 network pretrained on the ImageNet dataset served as the backbone for the Context Path. The output of this network was directly upsampled to the original input image, similar to FCN, establishing the baseline for evaluation.

Results

For the ablation study on the U-shape structure, its proposed that the Context Path with a U-shape structure, utilizing the lightweight Xception39 model for quick downsampling is done. Adopting the U-shape-8s structure, which combines features from the last two stages of the Xception39 network, significantly improved performance from 60.79% to 66.01%.

To address the challenge of lost spatial information in real-time semantic segmentation, here the Spatial Path has been introduced during the ablation study. The Spatial Path, featuring three convolutions with stride = 2, followed by batch normalization and ReLU, enhanced performance from 66.01% to 67.42%, demonstrating its effectiveness in encoding spatial information.

In the ablation study for the Attention Refinement Module (ARM), the aim was to further enhance performance. This module incorporates global average pooling, a convolutional layer, batch normalization, and ReLU to compute an attention vector. The original feature is then reweighted by this attention vector, contributing to improved global context awareness without complex up-sampling operations.

Addressing the different levels of features from the Spatial Path and Context Path, the ablation study for the Feature Fusion Module was conducted to effectively combine these features. A straightforward sum of features was compared with our proposed Feature Fusion Module, highlighting the importance of considering feature levels in the fusion process.

To enhance the receptive field of the Context Path, the ablation study for Global Average Pooling was implemented. This involved adding global average pooling at the tail of the Xception39 model. The output of global average pooling was then upsampled and summed with the output of the last stage in the Xception39 model. This design improved performance from 67.42% to 68.42%, underscoring the efficacy of this approach.

4.2.4 PSPNet

Experiment Details

The implementation of Pyramid Scene Parsing (PSPNet) has been executed on the PASVOC and Cityscapes datasets. Utilizing the Caffe platform, the implementation adheres to the poly learning rate policy, setting the base learning rate to 0.01 with a power of 0.9. To optimize model performance, the iteration number is adjusted to 150K for the ImageNet experiment, 30K for PASCAL VOC, and 90K for Cityscapes.

A robust data augmentation strategy is applied uniformly across all datasets, encompassing random mirror and random resizing between 0.5 and 2. For ImageNet and PASCAL VOC, additional augmentations, such as random rotation between -10 and 10 degrees and random Gaussian blur, are introduced. This comprehensive approach to data augmentation aims to enhance the network's resilience to overfitting.

Throughout our experiments, it has been observed that an appropriately large "cropsize" significantly contributes to achieving good performance. Additionally, the "batchsize" parameter in the batch normalization layer is of great importance for the success of the model.

In the context of the ImageNet scene parsing challenge 2016, the ADE20K dataset which is known for its complexity with up to 150 classes, diverse scenes, and 1,038 image-level labels has been leveraged. The dataset is split into 20K/2K/3K images for training, validation, and testing, respectively. ADE20K presents a unique challenge as it requires parsing both objects and stuff in the scene, setting it apart from other datasets. Evaluation metrics include pixelwise accuracy (Pixel Acc.) and the mean of class-wise intersection over union (Mean IoU).

In the ablation study for PSPNet, various settings have been explored, including different pooling types (max and average), pooling with either one global feature or four-level features, and the impact of dimension reduction after pooling and before concatenation. Results reveal that pooling with pyramid parsing outperforms global pooling, and dimension reduction further enhances performance. The optimized PSPNet achieves impressive results of 41.68/80.04 in Mean IoU and Pixel Acc. (%), surpassing global average pooling of 40.07/79.52 by 1.61/0.52.

Additionally, in the ablation study for pre-trained models, experiments involve different depths of pre-trained ResNet (50, 101, 152, 269). Increasing ResNet depth from 50 to 269 yields a notable improvement, with the score of (Mean IoU + Pixel Acc.) / 2 (%) improving from 60.86 to 62.35, demonstrating a substantial 1.49 absolute improvement.

Results

The results of the ablation study are in the 4.3.

Method	Mean IoU(%)	Pixel Accuracy (%)
PSPNet(50)	41.68	80.04
PSPNet(101)	41.96	80.64
PSPNet(152)	42.62	80.80
PSPNet(269)	43.81	80.88

Table 4.3: PSPNet model results

The ensemble submission attained a score of 57.21% on the testing set, and even the single-model submission achieved a high score of 55.38%, surpassing some multi-model ensemble submissions. The slightly lower score on the testing set compared to the validation set may be attributed to differences in data distributions between the two sets.

In the experiments on the PASCAL VOC 2012 segmentation dataset, containing 20 object categories and one background class, the model achieved an impressive accuracy of 82.6% when exclusively trained with the VOC 2012 data. This surpasses existing methods across all 20 classes. Remarkably, when pre-trained with the MS-COCO dataset, PSPNet reached an even higher accuracy of 85.4%, with 19 out of the 20 classes achieving the highest accuracy. Noteworthy is that the PSPNet, trained solely with VOC 2012 data, outperformed existing methods trained with the MS-COCO pre-trained model.

Cityscapes, a recently released dataset designed for semantic urban scene understanding, comprises 5,000 high-quality, pixel-level finely annotated images from 50 cities across different seasons. The dataset is divided into training (2,975 images), validation (500 images), and testing (1,525 images) sets, covering 19 categories encompassing both stuff and objects. Additionally, 20,000 coarsely annotated images are available for two training settings: utilizing only fine data or incorporating both fine and coarse data. PSPNet exhibited superior performance compared to other methods, showcasing a notable advantage. When trained with both fine and coarse data, our method achieved an impressive accuracy of 80.2%.

4.2.5 SGNet

The SGNet module implementation adheres to the intricacies outlined in the SGNet paper. Segmentation is conducted solely on a single dataset as per the specified instructions. It's important to note that the original intent was to deploy this model on the James robot for achieving accurate segmentation. Here first the effectiveness of S-Conv through various analyses has been assessed, including its application in different layers, ablation studies, comparisons with alternative methods, examining results using different input information for offset generation, and evaluating inference speed. Subsequently, comparison has been done on the performance of the SGNet, equipped with S-Conv, against other state-of-the-art semantic segmentation methods on the NYUDV2 dataset.

The NYUDV2 dataset comprises of 1449 RGB images with corresponding depth maps and pixel-wise labels. Of these, 795 images are allocated for training, while the remaining 654 are reserved for testing. As a backbone network for feature extraction, a dilated ResNet101 pretrained on ImageNet has been employed. For training, the SGD optimizer with a learning rate schedule following the "poly" policy has been utilized. In the ablation study, the initial learning rate is set to 5e-3, while for NYUDv2, it is adjusted to 8e-3. During testing, we down-sample the image to the training crop size (480 x 640), and its prediction map is upsampled to the original size.

Substituting Convolution with S-Conv: Here the efficacy of S-Conv has been assessed by replacing conventional convolutions (using a 3x3 filter) in different layers. Initially, the convolution in layer 3 has been replaced and subsequently extend this exploration to other layers. The findings yield two key conclusions from the results: While the baseline network demonstrates fast inference speed, its performance is subpar. However, replacing convolution with S-Conv enhances the results of the baseline network, albeit with a slight increase in parameters and computational time. The second being notably, replacing later convolutions, especially beyond the first convolution in layer 3 with a stride of 2, yields better results. This improvement is attributed to the superior ability of spatial information to guide down-sampling operations in the initial convolution.

The aforementioned experiments underscore that S-Conv significantly enhances network performance with only a marginal increase in parameters. It's essential to highlight that our network lacks a dedicated spatial information stream; instead, spatial information influences the distribution and weight of the convolution kernel.

Results

The influence of various spatial information formats on S-Conv has also been assessed, with results detailed in 4.4.

Information	Acc	mAcc	mIoU)
Depth	75.4	60.9	49.1
RGB Feature	73.9	58.5	46.2
HHA	75.7	60.8	48.8
Coordinates	75.2	61.1	48.5

Table 4.4: SGNet model results

The findings reveal that depth information yields comparable results to HHA and 3D coordinates, outperforming intermediate RGB features used by deformable convolution. Notably, the conversion of depth to HHA is time-consuming, making 3D coordinates and depth maps more suitable for real-time segmentation using SGNet. Even without spatial information input (utilizing RGB features alone), the S-Conv demonstrates over a 3.4% improvement.

To highlight the lightweight nature of S-Conv, here the inference speed of SGNet is evaluated in this context. Additionally, S-Conv with two-stream methods using an image size of 480x640 has been compared. It is evident that S-Conv incurs only a minimal additional computational cost compared to two-stream methods. Finally comparison has been done with other state of the art methods. The learning rate has been adjusted accordingly and the input image has been downsampled to 480x640. In contrast to utilizing additional networks for spatial feature extraction, SGNet (ResNet50) attains competitive performance and the fastest inference with a minimal number of parameters.

Furthermore, the SGNet (ResNet101) achieves even more competitive performance, enabling real-time inference. This is facilitated by S-Conv, which efficiently leverages spatial information with only a marginal increase in parameters and computational cost. Notably, the S-Conv delivers promising results without relying on HHA information, making it well-suited for real-time applications. This underscores the efficiency of the S-Conv in harnessing spatial information.

The proposed SGNet demonstrates superior performance compared to other methods, which incorporates multi-scale testing, HHA information, and two ResNet152 backbones. Notably, the performance of SGNet can be further enhanced by adopting multi-scale testing, a technique employed by other methods in the comparison.

4.3 Comparison Summary

From 4.5 which shows mIoU it can be seen that ERFNet demonstrates impressive performance on the Cityscapes dataset, achieving a mean IoU of 68% . It maintains a forward time of 24 ms on a single GPU, making it efficient for real-time processing. ERFNet strikes a balance between segmentation accuracy and computational efficiency, outperforming many approaches while remaining competitive in efficiency. It's suitable for tasks requiring fast inference without sacrificing accuracy.

ICNet achieves an mIoU of 70.6% on the Cityscapes dataset set with a processing speed of 30 frames per second on a single GPU card. It surpasses several accuracy-focused methods and is suitable for real-time applications where accuracy is paramount.

PSPNet demonstrates remarkable accuracy on datasets like PASCAL VOC 2012 and Cityscapes, achieving up to 82.6% accuracy when pre-trained with the MS-COCO dataset. It outperforms existing methods across multiple classes and is suitable for tasks requiring high segmentation accuracy.

SGNet efficiently leverages spatial information for segmentation tasks, demonstrating competitive performance with minimal computational cost. Models like SGNet (ResNet50) and SGNet (ResNet101) achieve real-time inference while delivering promising results without relying on depth information. They are suitable for real-time applications where computational efficiency is crucial. The 4.5 shows the Mean IoU of models considered. It shows the best results of each model based on a specific configuration.

Method	Mean IoU(%)
ICNet	70.6
PSPNet(101)	82.6
BiseNet	
Xception39	65.6
Res18	68.7
PSPNet(269)	43.81
ERFNet	68
SGNet	48.6

Table 4.5: Mean IoU comparison from the most effective configurations in each model

ErfNet, ICNet, PSPNet, BiSeNet, and SGNet, each bring something unique to the table when it comes to segmenting images. ErfNet is like the lightweight champion, designed to be fast and efficient one even on less powerful devices. ICNet, on the other hand can be called the multitasker, juggling different tasks simultaneously to get real-time results. PSPNet focuses on getting the big picture, making sure to capture all the details at different scales. BiSeNet is all about quick and accurate segmentation, ensuring speed without sacrificing accuracy. Lastly, SGNet is like the navigator, using spatial guidance to stay accurate while moving swiftly. Together, they cover a wide range of needs, offering solutions for various image segmentation challenges.

In the pursuit of understanding the impact of integrating deep learning approaches with traditional computer vision methods on dynamic object segmentation accuracy and real-time performance, the selection of ERFNet, ICNet, PSPNet, BiSeNet, and SGNet is strategic. ERFNet's architectural simplicity and impressive accuracy, achieved without pre-training on external datasets, position it as a strong candidate for assessing the fusion of deep learning with conventional methods.

Similarly, ICNet's real-time processing capabilities, coupled with its competitive accuracy, make it suitable for investigating computational challenges in image segmentation. PSPNet's exceptional accuracy on datasets like PASCAL VOC 2012 and Cityscapes, along with its flexibility in handling various evaluation metrics, aligns with the need to understand the role of machine learning techniques in precision improvement.

BiSeNet's focus on balancing accuracy and efficiency, particularly in real-time applications, addresses the trade-offs between precision and execution speed, crucial for addressing research questions related to performance trade-offs.

Lastly, SGNet's innovative use of spatial information and lightweight architecture makes it a viable candidate for exploring how different algorithms handle computational challenges and achieve real-time segmentation. Overall, the selection of these models offers a comprehensive approach to investigating the research questions while considering architectural design, computational efficiency, attention mechanisms, skip connections, and the balance between accuracy and speed in image segmentation algorithms.

Brief overview of certain factors which were taken into consideration while comparing the difference between the models:

	SGNet	PSPNet	ICNet	ERFNet	BiSeNet
Architectural Design	Uses spatial guided information in convolution	Pyramid pooling module	Cascade Architecture with 3 branches	Factorised convolution and residual connections	Uses a Dual path structure
Computational Efficiency	Efficiency depends on design and optimization	Demanding dude to pyramid pooling	Balances accuracy with computational accuracy very well	Emphasizes efficiency and is suitable for real time applications	Tries to find balance between accuracy and efficiency
Attention Mechanisms	Spatial guidance is used	Used for context mechanisms	Uses spatial attention for guiding	Used for guiding the convolutional operations	Used for guidance during convolution
Skip Connections	Depends of the specific design used	Does not use skip connections	In order to combine features from different scales	Used for feature propagation	To enhance information flow
Accuracy vs Speed	Trade off depends on the design	Emphasizes accuracy	Balances accuracy with real time processing	Emphasizes on efficiency and achieves a balance between speed and accuracy	Strives for a balance between the two

5 Conclusion and Recommendations

In summary, the thorough analysis of PSPNet, ERFNet, BiSeNet, ICNet, and SGNet yields valuable insights into their unique strengths and characteristics in the realm of semantic segmentation. These intricately crafted models, tailored to address specific challenges, exhibit diverse performances influenced by various factors. The primary goal of this assignment is to determine the most suitable network for implementation in the James robot setup. Chapter 3 provides a detailed exploration of several factors considered in comparing these methods.

PSPNet, distinguished by its emphasis on global contextual information through the pyramid pooling module, stands out as a robust solution for tasks demanding a nuanced understanding of an image's overall context. Its capacity to recognize objects within a broader perspective positions PSPNet as a compelling choice for applications requiring an in-depth contextual analysis.

ERFNet, in contrast, emerges as a standout model prioritizing real-time semantic segmentation. Leveraging efficient residual factorized convolutions, its lightweight architecture proves instrumental in scenarios where computational constraints are critical. ERFNet's ability to deliver real-time processing makes it particularly well-suited for embedded systems and environments emphasizing swift responsiveness.

BiSeNet, with its innovative two-path architecture encompassing spatial and context paths, strikes an optimal balance between accuracy and efficiency. By parallelizing computations and integrating local and global information, BiSeNet offers versatility across scenarios where a trade-off between speed and accuracy is desired.

ICNet introduces a multi-resolution cascade network architecture, optimizing efficiency and accuracy for real-time semantic segmentation tasks. The adoption of a cascade of networks with different resolutions enables ICNet to tailor its computational complexity while maintaining competitive performance, making it a suitable choice for real-time applications.

SGNet, as observed in the literature, introduces the Spatial Information-Guided Convolutional Network, showcasing notable results. Its adaptability to changes in spatial information, facilitated by the Spatial Information Guided Convolution (S-Conv), positions SGNet as a promising solution for real-time semantic segmentation tasks.

In summary, the comparison among PSPNet, ERFNet, BiSeNet, ICNet, and SGNet unveils a rich landscape of semantic segmentation models, each offering unique advantages. The selection of an ideal model hinges on a careful consideration of application-specific requirements, balancing factors such as computational resources, accuracy, and real-time performance. This nuanced evaluation highlights the diverse strengths and applicability of each model, contributing to the evolving toolbox for semantic segmentation challenges in computer vision.

A Appendix 1

During the preparation of this work the author used OpenAI's ChatGPT 3.5 in order to improve the quality of writing with the aim to enhance readability & articulation and to obtain a preliminary idea for the skeletal structure of various sections of the report.

Bibliography

[1] H. Li and K. Ngan, *Image/Video Segmentation: Current Status, Trends, and Challenges*, pp. 1–23. 10 2011.

[2] R. A. Graciela and C. M. M. I., *New Trends on Dynamic Object Segmentation in Video Sequences: A Survey.* REVISTA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y COMPUTACIÓN, 2013.

[3] A. Bewley, V. Guizilini, F. Ramos, and B. Upcroft, "Online self-supervised multi-instance segmentation of dynamic objects," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1296–1303, 2014.

[4] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[5] R. Hachiuma, C. Pirchheim, D. Schmalstieg, and H. Saito, "Detectfusion: Detecting and segmenting both known and unknown dynamic objects in real-time SLAM," *CoRR*, vol. abs/1907.09127, 2019.

[6] R. Dias, B. Cunha, E. Sousa, J. L. Azevedo, J. Silva, F. Amaral, and N. Lau, "Real-time multi-object tracking on highly dynamic environments," in *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 178–183, 2017.

[7] M. S. A. Manap, R. Sahak, A. Zabidi, I. Yassin, and N. M. Tahir, "Object detection using depth information from kinect sensor," in *2015 IEEE 11th International Colloquium on Signal Processing and Its Applications (CSPA)*, pp. 160–163, 2015.

[8] A. F. Elaraby, A. Hamdy, and M. Rehan, "A kinect-based 3d object detection and recognition system with enhanced depth estimation algorithm," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 247–252, 2018.

[9] N. Tatematsu, J. Ohya, and L. Davis, "Detection and segmentation of moving objects from dynamic rgb and depth images," vol. 8971, pp. 19–34, 09 2015.

[10] Y. Xie and J. Ohya, "A method for detecting multiple independently moving objects from the sequences acquired by active stereo cameras and estimating the cameras' egomotion," *Journal of the Institute of Image Electronics Engineers of Japan*, vol. 39, pp. 163–174, Jan. 2010.

[11] A. Mishra and Y. Aloimonos, "Visual segmentation of simple objects for robots," vol. VII, 06 2011.

[12] J. Owens, *Object Detection using the Kinect.* Army Research Laboratory, 2012.

[13] G. Ramirez and M. Chacon, "Segmentation of dynamic objects in video sequences fusing the strengths of a background subtraction model, optical flow and matting algorithms," pp. 33–36, 04 2014.

[14] Y. Tian, A. Senior, and M. Lu, "Robust and efficient foreground analysis in complex surveillance videos," *Machine Vision and Applications*, vol. 23, 09 2012.

[15] Q. Zhu and Z. Song, "Dynamic video segmentation via a novel recursive bayesian learning method," in *2010 IEEE International Conference on Image Processing*, pp. 2997–3000, 2010.

[16] D. Zeng, X. Chen, M. Zhu, M. Goesele, and A. Kuijper, "Background subtraction with real-time semantic segmentation," *IEEE Access*, vol. 7, pp. 153869–153884, 2019.

[17] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, 2015.

[18] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," *CoRR*, vol. abs/1704.08545, 2017.

[19] W. Runzhi, W. Wan, Y. Wang, and K. Di, "A new rgb-d slam method with moving object detection for dynamic indoor scenes," *Remote Sensing*, vol. 11, p. 1143, 05 2019.

[20] P. F. Alcantarilla, J. J. Yebes, J. Almazán, and L. M. Bergasa, "On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *2012 IEEE International Conference on Robotics and Automation*, pp. 1290–1297, 2012.

[21] D.-H. Kim and J.-H. Kim, "Effective background model-based rgb-d dense visual odometry in a dynamic environment," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1565–1573, 2016.

[22] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3748–3754, 2013.

[23] T. Nakamura, "Real-time 3-d object tracking using kinect sensor," in *2011 IEEE International Conference on Robotics and Biomimetics*, pp. 784–788, 2011.

[24] C. Hou, X. Zhao, and Y. Lin, "Depth estimation and object detection for monocular semantic slam using deep convolutional network," in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 256–263, 2020.

[25] L. Kalake, W. Wan, and L. Hou, "Analysis based on recent deep learning approaches applied in real-time multi-object tracking: A review," *IEEE Access*, vol. 9, pp. 32650–32671, 2021.

[26] S. Lee and H. Hong, "Use of gradient-based shadow detection for estimating environmental illumination distribution," *Applied Sciences*, vol. 8, p. 2255, 11 2018.

[27] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 433–440, 2018.

[28] J. Shin, H. Kim, D. Kim, and J. Paik, "Fast and robust object tracking using tracking failure detection in kernelized correlation filter," *Applied Sciences*, vol. 10, p. 713, 01 2020.

[29] A. Kampker, M. Sefati, A. Rachman, K. Kreisköther, and P. Campoy, "Towards multi-object detection and tracking in urban scenario under uncertainties," pp. 156–167, 01 2018.

[30] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu, "DETRAC: A new benchmark and protocol for multi-object tracking," *CoRR*, vol. abs/1511.04136, 2015.

[31] H. Wang, X. Jiang, H. Ren, Y. Hu, and S. Bai, "Swiftnet: Real-time video object segmentation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1296–1305, 2021.

[32] *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1.* Springer International Publishing, 2020.

[33] G. Marcus, "Deep learning: A critical appraisal," 2018.

[34] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2018.

[35] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," 2018.

[36] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," 2017.

[37] L.-Z. Chen, Z. Lin, Z. Wang, Y.-L. Yang, and M.-M. Cheng, "Spatial information guided convolution for real-time rgbd semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, p. 2313–2324, 2021.