Challenging Roads, Smarter Segmentation: Deep Learning vs. Traditional Methods

Ahmad Faisal Mirza, Humayun Kamal, Bilal Hashmi, Huma Tahir
School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST), Pakistan
CMS: 383744, 366008, 366009, 378286
Emails: {amirza.bee21seecs, hsiddiqui.bee21seecs, bhashmi.bee21seecs, htahir.bee21seecs}@seecs.edu.pk

***Abstract*—In this paper, we will analyze semantic segmentation, a critical task in computer vision that involves dividing images into meaningful regions through pixel-level categorization. We will provide a detailed comparison of deep learning and non-deep learning methodologies for semantic segmentation. Using advanced architectures such as U-Net and DeepLab, we will explore how deep learning approaches leverage convolutional neural networks (CNNs) to achieve state-of-the-art performance across diverse applications. Simultaneously, we will examine non-deep learning techniques, including thresholding, region-based segmentation, and clustering algorithms, which offer computational efficiency but often lack the adaptability of deep learning methods. By systematically evaluating the strengths, limitations, and practical use cases of both methodologies, we will identify the trade-offs between accuracy, computational complexity, and applicability. Through this study, we will provide valuable insights to guide researchers and practitioners in selecting the most suitable techniques for their semantic segmentation challenges.**

I. Introduction

Semantic segmentation, the task of classifying each pixel in an image, is crucial for applications like autonomous driving, medical imaging, and satellite analysis. Two main methodologies address this challenge: deep learning-based approaches and non-deep learning techniques. Deep learning methods, such as U-Net, achieve high accuracy and adaptability by leveraging convolutional neural networks (CNNs). However, they require substantial computational resources and large datasets. Non-deep learning techniques, including clustering and edge detection, are computationally efficient and simpler to implement but often lack the flexibility and robustness of deep learning. This paper explores the trade-offs between these methodologies, balancing accuracy, computational cost, and practical applicability to guide optimal approach selection.

II. Methodology

A. Dataset

The Indian Driving Dataset (IDD) is a publicly available dataset specifically designed to address the unique challenges faced by autonomous driving systems in the Indian subcontinent. Unlike datasets collected in regions with well-developed road infrastructure, such as Cityscapes or KITTI, the IDD captures the complexity and variability inherent to Indian roads. It includes images that reflect diverse and underdeveloped road conditions, non-lane-driven traffic, mixed traffic types (including pedestrians, bicycles, rickshaws, and animal carts), and varied environmental conditions.

The dataset contains over 10,000 high-resolution images with pixel-level annotations for 34 semantic classes. These classes include road, vehicle, pedestrian, vegetation, and more, making it suitable for semantic segmentation tasks.

We have chosen the IDD because limited work has been done in developing autonomous driving systems tailored for the Indian subcontinent. The unique characteristics of Indian roads, such as irregular lane markings, chaotic traffic behavior, and the frequent presence of unexpected obstacles, necessitate specialized solutions. Using this dataset, we aim to contribute to filling this research gap and advancing the development of robust autonomous driving technologies for underdeveloped and diverse road environments.

B. Deep Learning Techniques for Semantic Segmentation

For the semantic segmentation of driving scenarios in the Indian context, we employ state-of-the-art deep learning techniques tailored for accuracy and efficiency. These models are specifically selected to address the challenges posed by the complex and dynamic environments depicted in the IDD. Each technique is carefully fine-tuned and evaluated for its ability to generate precise segmentation maps, as detailed below:

1) SegFormer: SegFormer is a transformer-based semantic segmentation model that combines a lightweight architecture with state-of-the-art performance. It operates as follows:

- A hierarchical transformer encoder is employed to capture multi-scale contextual information from the input image. This allows the model to understand both global and local patterns, crucial for segmenting complex road scenes.
- The encoder outputs are passed to a Multi-Layer Perceptron (MLP)-based decoder, which efficiently aggregates features across different scales to produce the final segmentation map.
- The model avoids reliance on positional embeddings, making it robust to variations in input resolution, which is advantageous for processing diverse scene sizes in the IDD.

This approach is particularly effective in cluttered and dynamic environments typical of Indian roads, where objects such as pedestrians, vehicles, and infrastructure frequently interact.

2) UNet with ResNet34 Base: UNet is a classic encoder-decoder architecture designed for segmentation tasks. In this implementation, it is combined with a ResNet34 backbone to enhance feature extraction. The process is as follows:

- The ResNet34 backbone acts as the encoder, extracting hierarchical features from the input image through its residual connections, which help mitigate the vanishing gradient problem and enable learning deep representations.
- The decoder reconstructs a high-resolution segmentation map by upsampling the encoded features and concatenating them with corresponding encoder features using skip connections. This ensures precise spatial localization of segmented objects.
- The model is well-suited for identifying diverse objects such as vehicles, pedestrians, and road features, leveraging the strong feature extraction capability of ResNet34.

This architecture is particularly effective for capturing fine-grained details in the complex environments of Indian driving scenarios.

3) Feature Pyramid Network (FPN): The Feature Pyramid Network (FPN) is a multi-scale feature representation model designed to address the challenge of segmenting objects of varying sizes. It works as follows:

- The encoder processes the input image, extracting features at multiple levels. These features represent different spatial resolutions, enabling the model to capture both coarse and fine details.
- A top-down architecture is utilized to fuse high-level semantic features with low-level detailed features. This is achieved by upsampling the high-level features and combining them with the corresponding lower-level features through lateral connections.
- The resulting feature maps are used to generate fine-grained segmentation outputs, making FPN particularly effective for identifying both small obstacles and large vehicles.

This technique is well-suited for Indian roads, where objects of varying scales—ranging from small debris to large buses—must be accurately segmented.

4) Pyramid Scene Parsing Network (PSPNet) with MobileNetV2 Base: The Pyramid Scene Parsing Network (PSPNet) is designed to capture global contextual information by employing pyramid pooling modules. With the MobileNetV2 backbone, the process includes:

- The MobileNetV2 backbone serves as a lightweight encoder, extracting features efficiently while maintaining a balance between computational cost and accuracy. Its depthwise separable convolutions significantly reduce the number of parameters.
- The pyramid pooling module aggregates features at multiple scales by dividing the feature map into regions of different sizes and applying pooling operations. This enables the model to understand both global scene structure and local details.
- The pooled features are concatenated and upsampled to produce the final segmentation map, ensuring that high-resolution details are preserved.

This combination of PSPNet and MobileNetV2 is particularly advantageous for high-resolution images in the IDD, offering a trade-off between accuracy and real-time performance, critical for autonomous driving applications.

Each of these techniques is meticulously fine-tuned to address the unique challenges posed by Indian driving conditions, such as high object density, varied scales, and dynamic scenes. By leveraging these models, we aim to identify the most effective approach for advancing semantic segmentation in autonomous driving within the subcontinent.

C. Traditional Methods for Semantic Segmentation

In addition to deep learning techniques, traditional image processing methods are explored for semantic segmentation on the IDD. These methods serve as baseline approaches, providing foundational insights into the segmentation challenges in Indian driving scenarios. Each technique employs distinct mechanisms to process image data and identify meaningful regions, as detailed below.

1) Watershed Algorithm: The Watershed algorithm is a region-based image segmentation technique that treats the grayscale intensity of an image as a topographic surface. It works as follows:

- Local intensity minima are identified and treated as catchment basins.
- The algorithm simulates flooding from these basins by incrementally raising the water level.
- As the flooding progresses, pixels are assigned to their respective catchment basins based on proximity and intensity gradients.
- To prevent basins from merging, ”dams” are constructed at points where water from different basins would meet.

This technique is especially effective for segmenting overlapping objects, such as vehicles in dense traffic, by leveraging intensity differences to separate adjacent regions. However, the algorithm's sensitivity to noise and over-segmentation can be mitigated by pre-processing the image using Gaussian blurring or morphological operations.

2) Histogram-Based Thresholding: Histogram-based thresholding, specifically Otsu's method, is a global technique for image segmentation. The process involves:

- Constructing the histogram of pixel intensities from the input image.
- Dividing the intensity values into two classes: foreground and background.
- Iteratively evaluating all possible thresholds to maximize the between-class variance, which ensures clear separation of the two classes.
- Selecting the optimal threshold and applying it to the image to create a binary segmentation mask.

This method is computationally efficient and works well for scenes with distinct foreground and background intensity distributions, such as roads and vehicles. However, its global nature makes it less effective in scenarios with non-uniform lighting or complex textures, where adaptive or local thresholding may be required.

3) k-Means Clustering: The k-Means clustering algorithm is an unsupervised segmentation method that groups pixels into clusters based on their similarity in color, intensity, or other features. The steps are as follows:

- Initialize k cluster centroids randomly or using a heuristic.
- Assign each pixel to the nearest cluster based on a distance metric, such as Euclidean distance.
- Recalculate the centroids as the mean of all pixels assigned to each cluster.
- Repeat the assignment and centroid update steps until the centroids converge or a maximum number of iterations is reached.

The result is a segmented image where pixels within the same cluster share similar characteristics. This method is simple and intuitive but requires prior knowledge of the number of clusters (k), which may vary depending on the scene complexity. It also struggles with overlapping intensity distributions, common in natural environments.

4) Minimum Spanning Tree (MST): The Minimum Spanning Tree (MST) algorithm is a graph-based approach for segmentation that models an image as a graph. The steps involved are:

- Treat each pixel as a node in the graph, and define edges between nodes based on pixel similarity (e.g., intensity or color difference).
- Assign weights to the edges, representing the similarity between connected pixels.
- Construct a spanning tree that connects all nodes with the minimum total edge weight, ensuring that regions with similar pixels are grouped together.
- Partition the MST by removing edges with high weights, separating the graph into distinct subgraphs corresponding to segmented regions.

This method is highly effective for preserving object continuity and capturing fine details. However, it relies heavily on defining appropriate similarity metrics and can be computationally expensive for large-scale images, such as those in the IDD.

III. Results

IV. Results and Discussion

In this study, the evaluation metric used for comparing the performance of the semantic segmentation techniques was the Intersection over Union (IoU) for the **road class**. While deep learning (DL) methods were trained to segment all 34 classes in the IDD, their per-class IoUs are not directly compared with the non-deep learning (Non-DL) methods, as the IoU for all classes in the Non-DL methods was left for future work. Due to limited hardware constraints, the DL methods were trained for only 10 epochs. The results for the road IoU across all methods are summarized in Table I.

TABLE I
IoU Results for the Road Class using Various Methods

Method	**Road IoU**
Deep Learning (DL) Methods	
SegFormer	0.7746
UNet with ResNet34 Base	0.7869
PSPNet with MobileNetV2 Base	0.7571
FPN with ResNet34 Base	0.7869
Non-Deep Learning (Non-DL) Methods	
Histogram-Based Thresholding	0.3436
Watershed Algorithm	0.3862
Minimum Spanning Tree (MST)	0.4434
k-Means Clustering	0.3352

A. Comparison of Methods

a) Deep Learning (DL) Methods:: Among the DL methods, UNet with ResNet34 Base and FPN with ResNet34 Base demonstrated the highest road IoU, achieving 0.7869 each. SegFormer also performed well with an IoU of 0.7746, showcasing its ability to capture global and local context efficiently. PSPNet with MobileNetV2 Base came out with a road IoU of 0.7571 despite the trade-off in the MobileNetV2 backbone, which prioritizes computational efficiency over segmentation accuracy.

b) Non-Deep Learning (Non-DL) Methods:: The Non-DL methods yielded lower IoUs compared to their DL counterparts. The MST method achieved the highest IoU of 0.4434 among Non-DL methods, followed by the Watershed algorithm with an IoU of 0.3862. Histogram-based thresholding and k-Means clustering were less effective, with IoUs of 0.3436 and 0.3352, respectively. For k-Means, multiple values of k were tested, and the value providing the best road IoU was selected. The IoUs for multiple values are plotted in Figure 1.

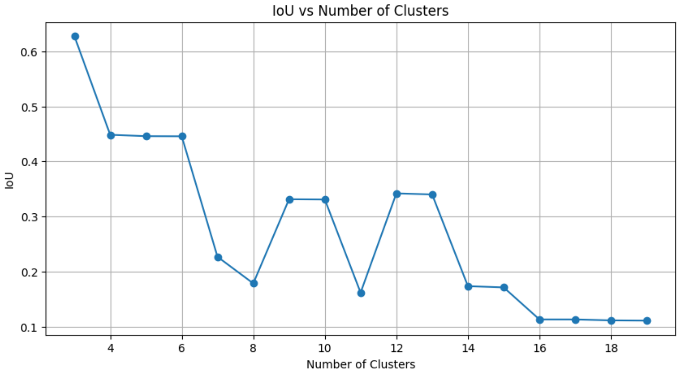

Fig. 1. Sample segmentation results UNet.

B. Evaluation Procedure

For the Non-DL methods, the road IoU was calculated by determining the class with the highest overlap with the ground-truth road class in the segmentation output. The DL methods provided IoU values directly for the road class, derived from the trained model outputs.

C. Visualization of Results

Figures 2, 3, 4, 5 and 6 illustrate the segmentation outputs, highlighting their ability to segment the road class in sample images from the IDD. Each colour represents a different class.

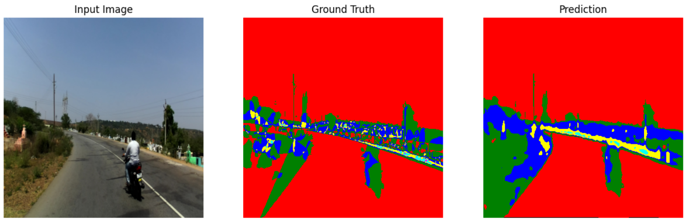

Fig. 2. Sample segmentation results UNet with the Input Image, Ground Truth and Prediction (Left to Right)

D. Discussion

The results highlight the superiority of DL methods over Non-DL methods for road segmentation in the IDD. The incorporation of hierarchical feature extraction and contextual information in DL methods significantly improves performance, as seen in the high IoUs of UNet and FPN. On the other hand, Non-DL methods struggle with the complex scene structure of Indian driving scenarios, yielding comparatively lower IoUs.

Future work will involve extending the IoU evaluation to all 34 classes for Non-DL methods, enabling a more comprehensive comparison. Additionally, fine-tuning DL methods for longer epochs and exploring other lightweight architectures could further enhance their performance, particularly for resource-constrained environments.

V. Conclusion

Semantic segmentation remains a pivotal task in computer vision, with diverse methodologies ranging from traditional techniques to advanced deep learning models. In this paper, we analyzed both paradigms, highlighting their strengths, limitations, and applicability to real-world challenges. Our exploration of IDD showcased the potential of deep learning models, such as SegFormer, U-Net, FPN, and PSPNet, in handling complex and dynamic road environments. These models excelled in accuracy and adaptability but came at the cost of higher computational requirements. On the other hand, traditional methods like Watershed, histogram-based thresholding, and k-means clustering offered computational efficiency but struggled with the intricacies of non-uniform and cluttered scenes.

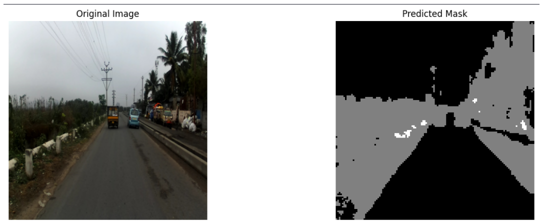

Fig. 4. Segformer Results

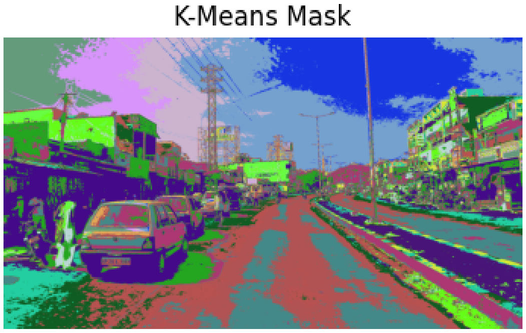

Fig. 5. Visualization of K-Means Results

The comparative analysis demonstrates that while deep learning techniques provide state-of-the-art performance for semantic segmentation tasks, traditional methods remain valuable for scenarios with limited resources or simpler segmentation needs. Future research should focus on hybrid approaches that combine the efficiency of traditional methods with the adaptability of deep learning to develop robust and resource-efficient solutions for semantic segmentation, particularly in challenging environments like those represented by IDD.

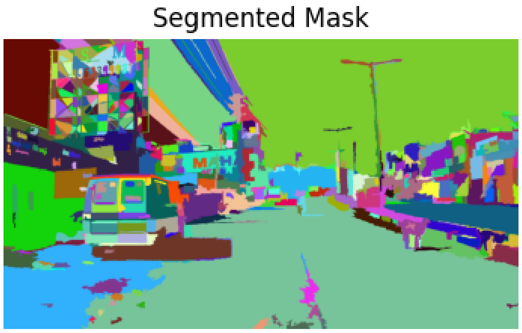

Fig. 3. Sample segmentation results MST

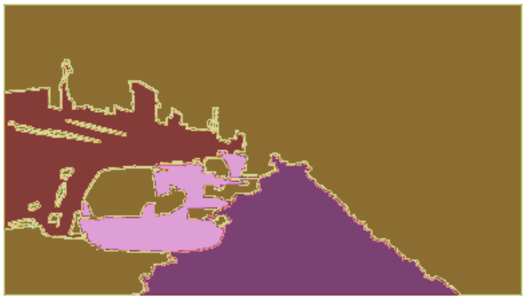

Fig. 6. Visualization of Watershed Results