iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 Issue: VIII Month of publication: Aug 2023

DOI: https://doi.org/10.22214/ijraset.2023.54991

Drivable Road Region Segmentation in Real Time with High Precision using Deep Learning

Dr. J. Sreerambabu[1], Mr. N. Santhosh[2], Mr. D. Rajkumar[3], Ms. K. Dharshini[4]
[1]Head of the Department, [2, 3]Assistant Professor, [4]PG Scholar

Abstract: This paper presents a novel approach that addresses the challenging task of real-time drivable road region extraction in computer vision. Semantic segmentation, which involves accurately identifying and segmenting objects in real-time data, is a complex problem. However, deep learning has proven to be a powerful technique for achieving semantic segmentation by automatically identifying patterns without the need for explicit programming. To tackle this task, the paper proposes a fusion of the YOLO algorithm and UNET architecture, leveraging their respective strengths. The YOLO algorithm enables high-speed object detection, while the UNET architecture provides advantages in global location utilization, contextual understanding, and performance, even with limited training samples. Importantly, the proposed method is lightweight, making it suitable for deployment on embedded systems with limited computational power. To optimize memory usage and capture context at different scales, the system employs dilated convolutions for efficient feature extraction. The algorithm exhibits exceptional performance in accurately segmenting irregular objects and handles diverse input data types, including images and videos, in real-time. Overall, this paper contributes significantly to the advancement of computer vision technologies and offers a valuable solution for real-time drivable road region extraction. Its potential applications include addressing driving challenges and enhancing safety in autonomous vehicles and intelligent transportation systems.

Keywords: Computer vision, Object detection, UNET architecture, YOLO algorithm, Image and video processing, Autonomous vehicles.

I. INTRODUCTION

Semantic segmentation, a challenging task in computer vision, involves partitioning an image into segments that correspond to specific objects or regions of interest. Recent advancements in deep learning have revolutionized this field, providing effective solutions. Semantic segmentation offers a detailed understanding of scenes by assigning labels to individual pixels, surpassing simple image classification approaches that focus on main objects.

This paper introduces a groundbreaking dataset with pixel-level annotations, specifically targeting drivable road regions. Deep learning algorithms leverage visual features such as color, shape, and text to detect and analyze objects in images, benefiting from their ability to automatically learn and recognize patterns without explicit programming. Semantic segmentation assigns a class label to each pixel, making it akin to "pixelwise classification."

The ongoing progress in computer vision and deep learning has significantly enhanced the speed and accuracy semantic segmentation algorithms.

The paper proposes a comprehensive system for semantic segmentation, incorporating essential components like UNET, pre-processing and post-processing modules, a training dataset, and evaluation modules. The system is designed to handle various input types, including images and videos, and has the capability to learn from data, continually improving accuracy in different lighting and environmental conditions.

By capturing both low-level and high-level features, the system effectively trains UNET to accurately segment input images into the desired semantic classes. The pre-processing module optimizes images through color correction and other necessary steps, ensuring their suitability for semantic segmentation.

Moreover, the system employs edge detection, smoothing techniques, and morphological operations to refine segmentation results and eliminate unwanted noise from the input. In conclusion, this paper presents a comprehensive study on the advancements in deep learning for semantic segmentation, addressing the challenges associated with detailed object recognition in images. The proposed system combines cutting-edge algorithms, carefully designed datasets, and efficient processing techniques to achieve highly accurate and efficient segmentation results.

The experimental evaluation showcases the system's performance and robustness, paving the way for future research and development in this exciting field of computer vision.

II. RESEARCH GAPS

Several existing systems utilize Convolutional Neural Network (CNN) algorithms for semantic segmentation of images. Among these systems, the Fully Convolutional Network (FCN) model is widely adopted for this purpose. The FCN model employs a deep encoder-decoder architecture based on CNN algorithms to perform semantic segmentation on input images. However, the conventional FCN approach faces challenges when dealing with small objects or objects with fine details, resulting in imprecise segmentation masks.

The FCN model has been trained on a comprehensive image dataset, such as the PASCAL VOC dataset, and its accuracy has been thoroughly evaluated and validated in real-world scenarios. Nevertheless, the existing system lacks proper implementation of the test-train split ratio during the training and testing phases.

Additionally, the use of a limited number of images in the existing system leads to inefficient dataset training. Moreover, the existing system does not consider the temporal coherence of video frames, leading to the presence of flickering and jittering segmentation masks. It produces inaccurate segmentation masks for certain objects. Another limitation of the FCN model is its high computational power and memory requirements, making it unsuitable for real-time applications. This limitation hinders the deployment of these methods on low-end hardware, restricting their practical use. Consequently, these limitations result in inaccurate segmentation masks and can pose challenges in object detection tasks.

III. PROPOSED METHODOLOGY

The proposed system integrates the YOLO algorithm and UNET architecture for performing semantic segmentation on diverse input data types, encompassing images and videos. It produces high-quality segmentation masks with smooth boundaries while maintaining a lightweight nature suitable for deployment on resource-constrained embedded systems. The system employs multi-scale feature extraction and skip connections to enhance segmentation accuracy and effectively handle objects of varying sizes and shapes.

The proposed system incorporates dilated convolutions for feature extraction, optimizing memory utilization and enhancing contextual understanding at different scales. This enables real-time operation on low-end hardware, making it an appropriate choice for deployment in resource-limited environments. Additionally, the system features an advanced computer vision algorithm based on the UNET architecture, which is well-suited for semantic segmentation tasks.

The components of the proposed system comprise a training dataset, a pre-processing module, a post-processing module, and an evaluation module. The training dataset consists of labeled images. The pre-processing module is responsible for preparing input images or videos for segmentation, encompassing tasks such as image enhancement, color correction, and other processing steps. The post-processing module refines the segmentation results to improve accuracy. The evaluation module quantifies the accuracy of the semantic segmentation algorithm.

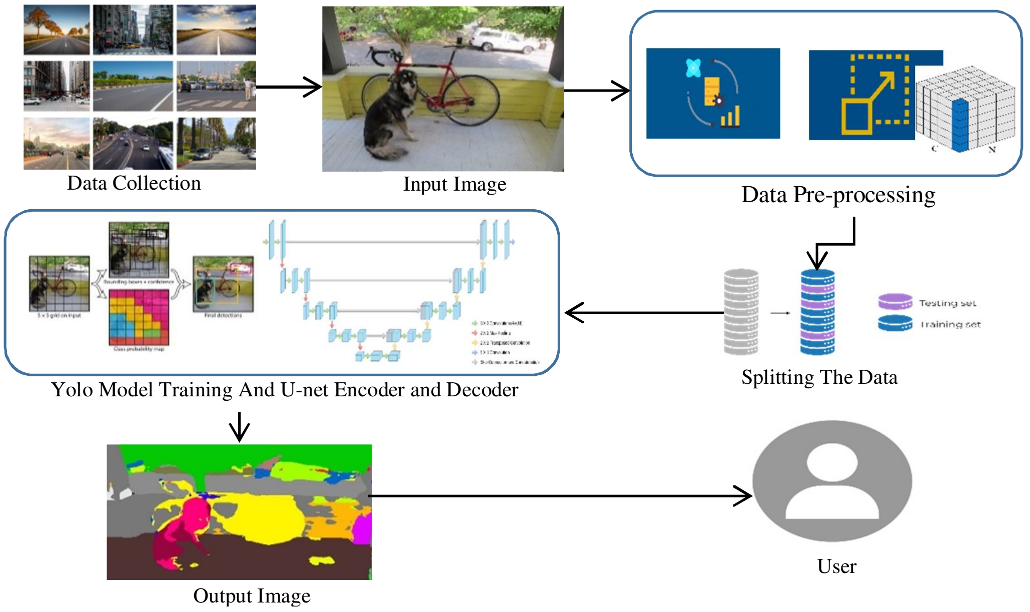

Fig 1. Workflow Diagram

IV. METHODS

A. *Data Pre-processing*

Data pre-processing stands a critical phase in machine learning process, particularly for semantic segmentation
Undertakings wherein the objective is to categorize each pixel within an image into a specific class.

The acquisition of a substantial and varied dataset holds paramount importance in constructing a resilient semantic segmentation model. This dataset must encompass images encompassing diverse scenes, viewpoints, lighting conditions, and object sizes. The greater the diversity in the dataset, the higher the potential for the model to exhibit robust generalization capabilities towards previously unseen data.

Data augmentation is a methodology employed to artificially enhance the dataset size by generating novel images derived from the pre-existing ones. This technique aids in mitigating overfitting and enhances the model's ability to generalize. Random cropping, flipping, rotation, and scaling are conventional data augmentation techniques utilized in semantic segmentation.

The resizing process can involve reducing the dimensions for computational efficiency or increasing them for enhanced model accuracy. It is imperative to resize the images to a predetermined size to guarantee the model's ability to process inputs consistently.

The normalization of pixel values within a designated range is of utmost importance to facilitate meaningful feature learning by the model. Typically, pixel values are normalized to reside within the interval of 0 to 1 or -1 to 1.

TYPICAL RANGE: 0 to 1 or -1 to 1

In label encoding, each pixel in an image is assigned a specific class label. The labels can be encoded as either one-hot vectors or integer values. While one-hot encoding is favored due to its superior performance, integer encoding can be employed for expedited processing.

B. *Module Training using YOLO*

The dataset for the YOLO model will be downloaded and preprocessed before being utilized in model construction. The YOLO model will be imported, and its parameters and backbone structure will be defined. If the existing network architectures are inadequate, YOLO allows for the creation of a custom architecture and anchor configuration. To facilitate this, a custom weights configuration file must be created.

1) *Defining Parameters:* The default model offered and trained on the COCO dataset, which is widely used for object recognition and segmentation. This dataset consists of images capturing various everyday environments and includes pre-defined classes. When selecting the pre-trained COCO model in the 'weights' parameter, the model will be initialized with the corresponding weights, which will be automatically downloaded for immediate use. Model depth and model width play crucial roles in developing scalable models. The depth multiplier parameter is responsible for adding more layers to the neural network, increasing its depth. Conversely, the width multiplier parameter enhances the number of filters within the layers, thus expanding the channels in the layer outputs. These multipliers are widely embraced by the research community as an effective methodology for constructing scalable models.

TABLE I
VALUES OF PARAMETERS

Parameter	**Value**
Number of classes	80
Depth multiplier	0.33
Width multiplier	0.50

2) *Creating Backbone:* The integration of an object segmentation algorithm in a prediction system to outline object boundaries and classify them is a novel approach. The incorporation of class labels within the differentiable network process, enabling end-to-end prediction of bounding boxes, represents a significant breakthrough in object detection. The YOLO (You Only Look Once) model stands as the pioneering object detector to embrace this advancement.

There are three components of the YOLO network:

- *Back bone:* The backbone is a convolutional neural network that gathers and creates visual features at various granularities.
- *Neck:* A number of layers that integrate and mix visual features before sending them on to prediction.
- *Head:* Consumes neck-related features and performs class and box prediction processes.

During this stage, we establish the foundation of the model by incorporating the Focus Layer, convolution layer, and Bottleneck CSP layer. The purpose of the focus layer can be likened to a spatial-to-depth transformation.

In traditional backbones like ResNet, the initial stem layer employs a Conv2d kernel with a size of 7x7 and a stride of 2 to reduce spatial dimensions (resolution) while simultaneously increasing depth (channel count).

In the case of YOLO, our objective is to optimize the computational cost of Conv2d operations while leveraging tensor reshaping techniques to reduce spatial requirements (resolution) and amplify depth (number of channels).

In YOLO, the focus layer performs a spatial-to-depth transformation. It takes a tensor of shape (B, C, H, W) as input and produces an output tensor of shape (B, C × 4, H/2, W/2) by rearranging and concatenating the spatial information.

DenseNet serves as the foundational architecture for the CSP models, which aim to address the vanishing gradient problem, enhance feature propagation, promote feature reuse, and reduce network parameters.

In CSPResNext50 and CSPDarknet53, modifications have been made to isolate the feature map from the base layer. This involves duplicating the feature map, passing one copy through the dense block while directly transmitting the other copy to the subsequent stage.

The objectives of CSPResNext50 and CSPDarknet53 projects are to preserve the original feature map for improved learning and alleviate computational bottlenecks present in DenseNet.

The YOLO layer, located in the model head, plays a crucial role in performing the final detection process. By incorporating anchor boxes onto features, it generates output vectors comprising class probabilities and bounding boxes.

The model head in YOLO V5 shares similarities with its predecessors, YOLO V3 and V4, employing a multiscale prediction approach. This involves constructing feature maps at three different scales (small, medium, and large) to enhance the model's capability to detect objects of various sizes.

TABLE II
BACKBONE CONFIGURATION

#	From	Number	Module	Args
0	-1	1	Focus	[64, 3]
1	-1	1	Conv	[128, 3, 2]
2	-1	3	C3	[128]
3	-1	1	Conv	[256, 3, 2]
4	-1	9	C3	[256]
5	-1	1	Conv	[512, 3, 2]
6	-1	9	C3	[512]
7	-1	1	Conv	[1024, 3, 2]
8	-1	1	Spp	[1024, [5, 9, 13]]
9	-1	3	C3	[1024, False]

C. Model training using UNET

The Unet architecture, a dedicated CNN for image segmentation, comprises an Encoder and a Decoder. The Encoder uses convolutional layers to create a feature map from the input image, while the Decoder employs up-convolutional layers to generate a segmentation map. Skip connections connect the Encoder and Decoder, enabling access to high-level features. The loss function plays a vital role in the training process by quantifying the disparity between the predicted segmentation map and the ground truth segmentation map. In semantic segmentation tasks, the commonly used loss function is cross-entropy loss. It measures the dissimilarity between the predicted and ground truth segmentation maps, calculated pixel-wise and then averaged over all pixels in the map. The loss function guides the model's parameter updates, aiding in the optimization process. The training process involves iteratively updating the Unet model's parameters to minimize the loss function.

The training process typically involves the following steps:

1) *Initialization:* The parameters of the Unet model are initialized randomly, preparing the model for the training process.
2) *Forward Propagation:* The input image is passed through the encoder, which produces a feature map. Subsequently, the feature map is forwarded through the decoder to generate a segmentation map.
3) *Loss Computation:* The input image is passed through the encoder, which produces a feature map. Subsequently, the feature map is forwarded through the decoder to generate a segmentation map.
4) *Backward Propagation:* The gradients of the loss function are computed through the process of backpropagation, considering the model parameters.
5) *Parameter Update:* The model parameters are adjusted by applying an optimizer algorithm, such as stochastic gradient descent (SGD), to optimize their values.
6) *Repeat:* Steps b-e are repeated for a fixed number of epochs or until the loss function reaches a minimum value.

D. Test Semantic Segmentation Model

To test a semantic segmentation model, the following steps can be followed:

1) *Evaluate the Model:* After training, the model can be assessed on a distinct set of test images that were not used during training. Evaluation metrics such as Intersection over Union (IoU), Dice coefficient, and Pixel Accuracy can be employed to measure the performance of the model.
2) *Visualize the Result:* To gain insights into the model's performance, visualizing the segmentation results on a selection of sample images from the test set can be valuable. This visualization aids in identifying the model's strengths and areas that require improvement.
3) *Iterate:* Upon reviewing the evaluation results and visualizations, the model can undergo fine-tuning and refinement. This iterative process of training, evaluation, and visualization can be repeated until desired and satisfactory outcomes are achieved.

E. CLI Based Program

CLI, an abbreviation for command-line interface, is a text-based user interface used for executing programs, managing computer files, and interacting with a computer system. It is also known as console user interface, character user interface, or command-line user interface.

CLIs receive commands entered through the keyboard, which are then executed by the computer at the command prompt. Upon booting up and operating, the command line interface (CLI) of a computer system opens on a blank screen with a command prompt.

In this project, we will develop a CLI-based Python program to capture user input from images or videos and generate segmented photos and videos as the output, using a dataset.

V. IMAGE AND VIDEO SEGMENTATION

During the processing phase, each frame of the input data is individually handled. The loaded YOLO and Unet models are used to perform a forward pass on the frame, resulting in segmentation of the input. This step ensures that each frame is effectively segmented and ready for further analysis and visualization.

Taking these factors into account, we have the choice of utilizing either videos or images as input data. Once the decision is made, the video file is processed to generate an annotated output video. The initial steps involve loading labels, generating colours, and loading the YOLO and Unet models while specifying the output layer names. With these preparations in place, we are ready to commence frame-by-frame processing.

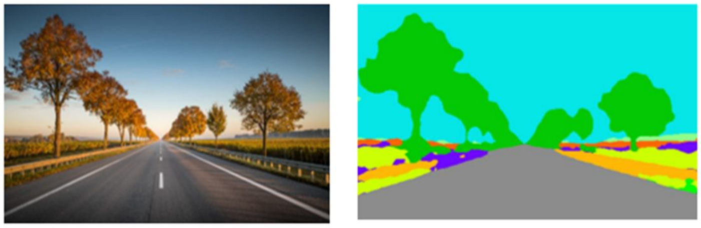

Fig 2. Image Segmentation

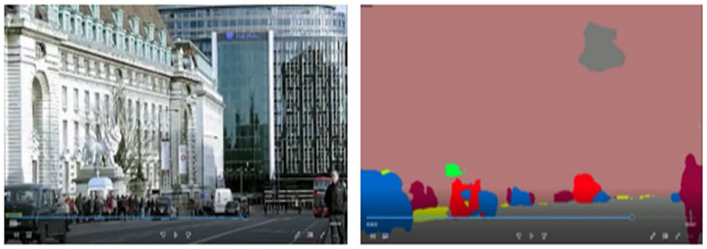

Fig 3. Video Segmentation

VI. CONCLUSION AND FUTURE WORK

The primary goal of the project is to develop a highly accurate and efficient segmentation solution specifically tailored for drivable road areas. Real-time segmentation of images and videos plays a vital role in self-driving applications, ensuring system safety and effectiveness. The ability to reduce the computational load of semantic segmentation is crucial for enabling its implementation in embedded systems and autonomous driving scenarios.

Semantic segmentation itself aims to assign semantic class labels to each pixel in an image, providing a comprehensive understanding of the scene's composition. The models discussed in this context offer practical solutions to meet the real-time requirements of critical applications, including road scene analysis, environmental awareness, and self-driving guidelines.

The future prospects of this project involve deploying it in a cloud-based environment and enhancing algorithm accuracy through additional training epochs. Additionally, exploring more efficient few-shot learning techniques for semantic segmentation is an intriguing avenue. These techniques enable models to learn how to segment images from new categories using only a limited number of training examples, which is particularly valuable in applications where acquiring labelled training data is costly or challenging.

VII. ACKNOWLEDGEMENT

I acknowledge our Head of the Department Mr Dr. J. Sreerambabu, M.E., Ph.D., PDF, FIE and our mentor Mr N. Santhosh, MCA, who provided insight and expertise that greatly helped the research, for suggestions that greatly improved this manuscript. Special thanks, to our supervisor Mr M. Mohammed Riyaz, MCA, for the support in this research work.

REFERENCES

[1] Ruturaj Kulkarni, Shruti Dhavalikar, Sonal Bangar, “Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning”, Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2019

[2] Ze Liu, Yingfeng Cai, Hai Wang, Long Chen, Hongbo Gao, Yunyi Jia, Yicheng Li, “Robust Target Recognition and Tracking of Self-Driving Cars With Radar and Camera Information Fusion Under Severe Weather Conditions”, IEEE Transactions on Intelligent Transportation Systems (Volume: 23, Issue: 7), 2022

[3] Ramin Nabati, Hairong Qi, “RRPN: Radar Region Proposal Network for Object Detection in Autonomous Vehicles”, IEEE International Conference on Image Processing (ICIP), 2019

[4] Zhenchao Ouyang, Jianwei Niu, Yu Liu, Mohsen Guizani, “Deep CNN-Based Real-Time Traffic Light Detector for Self-Driving Vehicles”, IEEE Transactions on Mobile Computing (Volume: 19, Issue: 2), 2020

[5] Rikuya Takehara, Tad Gonsalves, “Autonomous Car Parking System using Deep Reinforcement Learning”, 2nd International Conference on Innovative and Creative Information Technology (ICITech), 2021

[6] Malvi Mungalpara, Priyanka Goradia, Trisha Baldha, Yanvi Soni, “Deep Convolutional Neural Networks for Scene Understanding: A Study of Semantic Segmentation Models”, International Conference on Artificial Intelligence and Machine Vision (AIMV), 2022

[7] A.A. Mahersatillah, Z. Zainuddin, Y. Yusran, “Unstructured Road Detection and Steering Assist Based on HSV Color Space Segmentation for Autonomous Car”, 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2021

[8] Jack Stelling, Amir Atapour-Abarghouei, “‘Just Drive’: Colour Bias Mitigation for Semantic Segmentation in the Context of Urban Driving”, IEEE International Conference on Big Data (Big Data), 2022

[9] Tuan Pham, “Semantic Road Segmentation using Deep Learning”, Applying New Technology in Green Buildings (ATiGB), 2021

[10] Sanchit Gautam, Tarosh Mathuria, Shweta Meena, “Image Segmentation for Self-Driving Car”, 2nd International Conference on Intelligent Technologies (CONIT), 2022