khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Region Enhanced Edge-Based Multi-Class Object Proposal for Self-Driving Vehicles

Muhamad Amirul Haq [1]*, Muhammad Ridlwan [2], Le Nam Quoc Huy [3]

[1]Department of Computer Science
Universitas Muhammadiyah Surabaya
Surabaya
[2]Department of Indonesian Language and Literature
Universitas Muhammadiyah Surabaya
Surabaya
[3]Department of Mechanical Engineering
National Taiwan University of Science and Technology
Taipei City
*correspondence: amirulhaq@ft.um-surabaya.ac.id

Abstract- On-road object detection is a fundamental element for the safety and reliability of autonomous driving systems. A primary challenge is developing object detection algorithms that are both fast and robust. This paper introduces a novel object proposal algorithm, named Region Enhanced Edge-Based (REEB) proposal, designed to accelerate object detection by significantly reducing the number of candidate regions requiring evaluation by a subsequent classification network. REEB leverages edge-map cues to score and rank initial proposals. To further enhance both detection quality and processing speed, the algorithm integrates efficient complementary techniques: image entropy is used to guide proposal generation density in relevant image regions, and road segmentation aids in refining proposal scores by differentiating road from non-road areas. Experimental evaluations on the KITTI dataset demonstrate that REEB achieves an average recall rate of 72.1% across four classes (pedestrian, cyclist, car, and truck) with an average processing time of 15 milliseconds per image. These results indicate strong performance when compared to other traditional, non-deep learning object proposal algorithms.

Keywords: on-road object detection, object proposals, edge detection, autonomous driving assistance.

Article info: *submitted March 30, 2024, revised May 30, 2025, accepted June 24, 2025*

1. Introduction

In recent decades, autonomous driving has become the major goal of various institutes and researches around the world [1]. Yet a series of self-driving car accidents causing losses of human lives and property damage raised questions about its safety [2]. Hence, a reliable on-road object detection system to prevent such accidents is necessary.

Various research using different sensing modalities such as radar, lidar [3], and optical sensors [4] have been developed in order to achieve a reliable on-road object detection. Compared to other sensing modalities, optical sensors have several advantages including higher spatial resolution, more extensive information in images, lower device cost, and easier maintenance [5]. However, on-road object detection based on optical sensors also pose their own problems mainly due to variability in object appearance (e.g., size, color, pose, and occlusion) and complex outdoor environments (e.g., extreme illumination, cluttered backgrounds, and so on) [1]. Nevertheless, on-road object detection based on the optical sensor has gained immense popularity due to the achievement of the convolutional neural network (CNN) [6], [7].

CNN is one of the most popular neural network architectures, which is extremely successful in image processing, including object detection. In CNN, most object detection algorithms work by using a classifier network to classify proposals that are generated by the well-known sliding window algorithm [8]. In modern detection datasets, the number of proposals required to achieve good detection quality in one image could range from 10^6 – 10^7 proposals [9]. Additionally, the advancement of the classifier network has led to significant improvement of detection quality at the cost of increasingly complex and computationally expensive networks [9]. This increasing complexity combined with the high number of proposals led to the speed degradation of the overall object detection. On the other hand, on-road object detection requires faster detection than other applications since the vehicle speed is directly affected by the detection speed [1]. For on-road object detection in autonomous vehicles, real-time performance is critical; inference times ideally should be well under 50 milliseconds to allow for timely decision-making, a standard that

many complex, high-accuracy models struggle to meet without specialized hardware [1].

To overcome the drawback of high detection quality on the detection speed, one could use either object proposal, region proposal network (RPN) [10], feature pyramid network (FPN) [11], *You Only Look Once* (YOLO) [12], or *Single-shot Detector* (SSD) [13]. However, RPN, YOLO, and SSD need large amount of training data and tend to generalize poorly on unseen object [14]. On the other hand, object proposal uses the assumption that all objects of interest share common visual properties that distinguish them from the background [9]. Object proposal algorithm that generates significantly fewer high-quality proposals than sliding window algorithm can reduce the inference time of the overall detection algorithm.

Based on how object proposals generate the initial proposals, they can be divided into two categories [9]: grouping methods and windows scoring methods. Grouping methods generate multiple segments that likely correspond to objects and group those segments based on a diverse set of parameters including superpixel shape, color, and boundary estimates [15], [16], [17], [18]. On the other hand, windows scoring methods use dense sliding windows to generate large numbers of proposals, then cancel and rank them efficiently using their unique scoring algorithm [14], [19], [20], [21]. Each category has its own merits and drawbacks. For example, grouping methods can produce a detailed segmentation mask of the object, whereas windows scoring methods only return rectangular bounding boxes. Nevertheless, windows scoring methods are generally faster than grouping methods [9], making them more suitable for applications requiring real-time processing. Naturally, a good balance between high speed and quality is needed, although the algorithm should still achieve real-time requirement.

To achieve high quality and real-time requirement, we propose windows scoring object proposal algorithm that combines scores from edge density cues with inexpensive combinatorial geometry constraints and probability models. Additionally, we utilize a flexible multi-regional division approach to improve the algorithm's ability to detect a relatively small object of interest on the image while improving the overall speed of the algorithm. This paper aims to provide a fast and robust object proposal algorithm with a high recall rate for on-road object detection.

This paper introduces a novel window scoring object proposal algorithm, termed Region Enhanced Edge-Based (REEB) proposal, specifically designed to achieve a strong balance between high recall rates and fast inference times for on-road object detection. The main contributions of this work are:

a. A fast window scoring mechanism that primarily uses edge density cues, complemented by inexpensive geometric constraints and regional likelihood based on image entropy.

b. A flexible multi-regional image division strategy that leverages hardware setup knowledge (e.g., camera parameters) and entropy to adaptively control proposal density, improving the detection of smaller objects and overall processing speed.

c. The integration of a simple yet effective road segmentation technique to further refine proposal scores and reduce false positives on road surfaces.

Our proposed REEB algorithm aims to provide a robust and computationally efficient front-end for on-road object detection systems. Experimental results on the KITTI dataset demonstrate that our method achieves an average recall of 72.1% at 2000 proposals with an inference time of 15 milliseconds per image, outperforming several other traditional object proposal methods in terms of speed and achieving competitive recall rates.

2. Related Works

Object detection is a cornerstone of computer vision, with significant advancements made in recent years. Broadly, detection frameworks can be categorized into two-stage and one-stage detectors. Two-stage detectors, such as R-CNN [22] and its successors like Fast R-CNN [23] and Faster R-CNN (which introduced RPNs [10]), first generate a sparse set of object proposals and then classify each proposal. These methods often achieve high accuracy but can be computationally intensive. One-stage detectors, like YOLO [12] and SSD [13], perform detection in a single pass, directly predicting bounding boxes and class probabilities from image features. They are generally faster but sometimes lag in accuracy compared to two-stage methods, especially for small objects. Both paradigms heavily rely on deep learning and typically require substantial labeled datasets for training and can be resource-intensive.

In contrast, traditional object proposal algorithms, which predate many deep learning detectors or serve as components within them, aim to identify a manageable number of candidate regions in an image that likely contain objects, without performing full classification. These methods are often based on low-level image cues and heuristics. As noted by Hosang et al. [9], object proposal methods can be broadly categorized into grouping methods and window scoring methods.

1. *Grouping-based Object Proposal Methods*

Grouping-based methods typically start by segmenting an image into superpixels or initial regions. These regions are then hierarchically merged based on similarity cues such as color, texture, size, and boundary continuity to form object proposals. Prominent examples include Selective Search [16], which explores various grouping strategies across different color spaces and similarity measures. Other notable grouping approaches are Constrained Parametric Min-Cuts (CPMC) [15] and Multiscale Combinatorial Grouping (MCG) [17].

The advantages of grouping methods can generate proposals that adhere well to object boundaries and can even produce segmentation masks. They often achieve high recall rates. Meanwhile, the hierarchical grouping process can be computationally expensive, making many of these methods relatively slow for real-time applications, e.g., Selective Search can take several seconds per image [9].

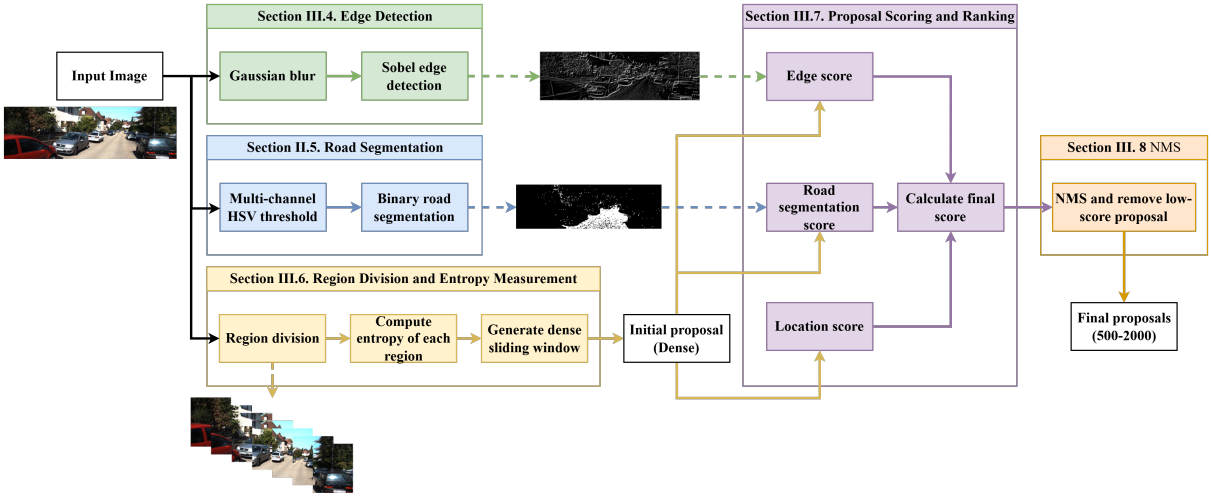

Figure 1. **Overall architecture of the proposed Region Enhanced Edge-Based (REEB) object proposal algorithm for self-driving vehicles. The process begins with an input image, followed by parallel streams for (A) edge detection (Gaussian blur and Sobel edge detection)**

2. *Windows Scoring Object Proposal Methods*

Windows scoring methods generate a large number of candidate windows, often through a dense sliding window approach across various scales and aspect ratios. Each window is then assigned an "objectness" score, which quantifies the likelihood of that window containing an object of any class. Proposals with low objectness scores are discarded. Edge Boxes [20] is a well-known example that scores proposals based on the number of edge contours wholly contained within a box. BING [14] uses simple gradient features and a linear SVM to achieve very fast objectness estimation.

Window scoring methods are generally much faster than grouping methods, with some, like BING, capable of processing hundreds of frames per second [14]. They are conceptually simpler and often easier to implement. However, the proposals are typically rectangular and may not tightly fit non-rectangular objects. The quality of proposals heavily depends on the effectiveness of the scoring function and the initial window sampling strategy. They might require a larger number of proposals to achieve the same recall as grouping methods.

Several works have focused on enhancing the accuracy of objectness scores by combining multiple cues or learning more discriminative features [18], [19], [21], [22], [23], [24]. While these can improve detection quality, combining multiple scoring mechanisms often increases the computational cost, potentially negating the speed advantage crucial for real-time systems. The challenge lies in finding a balance between discriminative power and efficiency. Our work builds upon the window scoring paradigm, aiming to improve both speed and quality by incorporating efficient complementary cues like regional entropy and road segmentation, specifically tailored for the autonomous driving context.

Table 1. Inference time of each component in our algorithm

Input: Image I
Edge_Map = Sobel(Gaussian_Blur(I)) //Sec. III.4
Road_Mask = Generate_Road_Segmentation(I, HSV) //Sec. III.5
Regions = Divide_Image_Into_Regions(I, Camera_Params) //Sec. III.6
Initial_Proposals = []
for each Region R_i in Regions do:
    H_i = Calculate_Shannon_Entropy(R_i) //eq. (2)
    Num_Proposals = Determine_Num_Proposals(H_i, $\sigma_{entropy}$) //eq. (9)
    Window_Params = Define_Sliding_Window_Parameters(R_i)
    Dense_Proposals_ R_i = Generate_Sliding_Windows()
    Add Dense_Proposals_ R_i to Initial_Proposals
End for loop
Scored_Proposals = []
for each Proposal P_j in Initial_Proposals do:
    S_j = Calculate_Edge_Score(P_j, Clustered_Edges) // Eq. (1)
    if S_j < Min_Edge_Score_Threshold then:
        continue // Discard proposal if edge score is too low
    End if
    R_j = Calculate_Road_Score(P_j, Road_Mask) // Eq. (10)
    L_j = Calculate_Location_Score(P_j, Image_Height) // Eq. (11)
    V_j = S_j * (α * L_j + β * R_j) // Eq. (12)
    Add (P_j, V_j) to Scored_Proposals
End for loop
Sort_Proposals_By_Score(Scored_Proposals) // Sec. III.8
Final_Proposals = NMS(Sorted_Proposals, NMS_Threshold) // Eq. (13)

Output: Return Final_Proposals (ranked list of object proposals)

3. Methods

In this study, an edge-based multi-class object proposal algorithm is proposed to detect on-road objects in a self-driving dataset as illustrated in Figure 1. The algorithm uses windows scoring method to significantly reduce the number of proposals generated by a dense sliding-windows method. At first, the algorithm will generate an edge map and binary road segmentation which will later be used to score a proposal. The proposed algorithm divides the image into several regions based on the hardware setup and the target object's size constraint. In those regions, the dense sliding-windows will generate a large number of proposals depending on the region's entropy. After that, the aforementioned scoring schemes, namely edges, location, and road segmentation will score all of the generated proposals. Finally, the proposals will be sorted, and the lower-ranked proposals will be removed until only a few thousands of top-ranked proposals remain. The remainder of this section will describe each of the process shown in Figure 1 and Table 1 in more details.

1. *Objectness score*

Objectness score is a measure of how likely a proposal contains an object [19]. It allows for ranking proposals from highest to lowest, enabling the algorithm to control the number of generated proposals by taking only the few hundred or few thousand top-ranked proposals. An algorithm that is able to calculate objectness score accurately will be more reliable at a fewer number of candidate proposals. Attempts in combining multiple objectness scores to improve its accuracy have been done in recent years [18], [24], [25], [26]. However, combining several objectness scores means that it will take the combined time of the involved algorithms. While the aforementioned works manage to successfully improve the detection quality, the total inference time is too high for real-time applications. In this paper, the scores are calculated using edge density cues while complemented by the inexpensive geometric constraint score and regional likelihood based on the region entropy. The complementary methods used in this work are inexpensive and implemented in such a way that is able to improve the quality of object proposals both in terms of speed and quality, exceeding current state-of-the-art methods.

2. *Edge-based object proposals*

In this work, we adopt Edge Box scoring method [20] which uses edges density cues as the base score for candidate proposals. The structured edge detector (SE) [27] used by Edge Box, is a state-of-the-art edge detector that can generate a high-quality edge map. The high-quality edge map helps the scoring algorithm get a more accurate objectness score. However, according to our experiment which will be further discussed in section IV, although using the edge map SE can help achieve acceptable detection quality, we found out that Sobel edge detection can produce comparable results while significantly improving the speed of the algorithm. Nevertheless, the scoring method based on edge density is highly efficient and able to produce promising results. The proposal score from edge density can be expressed as:

$$S = \frac{\sum_n w_n \cdot m_n - \sum_c m_c}{2 \cdot area^{\gamma}} \quad (1)$$

Figure 2. Efficient road segmentation on KITTI dataset

where S represents the edge density score of a proposal. The value of w_n is either 0 or 1, depending on whether an edge group n is wholly enclosed within a proposal or not. This binary scoring provides an extremely efficient computation since the value of m_n which represents the total magnitude of edges within an edge group n does not matter if an edge is not wholly enclosed within a proposal. The sum of m_n is then subtracted by the sum of edges magnitude located in the center of the proposal. The area in the center of a proposal is denoted as c and it has half the width and height of the proposal. The edges inside it are less important than those near the proposal's boundary. The score of a proposal is effectively equal to the sum of edges magnitude near the proposal's boundary divided by the center area of the proposal. The γ itself is constantly used to compensate for the fact that the bigger proposals have more edges than the smaller ones.

Despite the efficiency of Edge Box scoring scheme, the only way to improve its quality is by increasing the density of candidate proposals. By covering more parts of the image, it increases the probability of an object is covered in at least one of the candidate proposals. However, it also means that more candidate proposals are needed, which contradicts its original purpose. Thus, complementary methods are needed to provide a clue about which part of the image is more likely to contain an object.

3. *Image Entropy*

Image entropy is a cost-efficient method to measure the amount of information contained inside an image. The most common method to approximate the entropy of an image is Shannon entropy, which can be expressed as:

$$H = -\sum_{k=0}^{M-1} P_k \, log_2(P_k) \quad (2)$$

where P_k denotes the non-empty probability density of the k-th pixel level in an image with M number of pixel level. However, Shannon entropy does not incorporate any location information, which makes it less useful for an object proposals algorithm that needs to provide localization for an object detection algorithm. Other methods have been developed to integrate location information with spatial entropy [28], [29]. Then, the entropy information can be used to detect in which part of the image is more likely to contain an object. While these methods have been able to successfully provide high-quality location information, it is computationally expensive and not suitable for real-time applications. Thus, the proposed method is developed to provide a low-cost spatial entropy to estimate the most probable object locations and increase proposal density in those locations.

4. Image denoising and edge detection

Owing to the complex environment on roads, images captured by cameras are consequently degraded by noise. Therefore, image denoising plays a crucial role to reduce noise without losing image features. Gaussian blur is one of the inexpensive methods that can be used to reduce noise in the edge map. The reduced edges can improve inference time since the number of edges needed to be evaluated is decreased. After performing denoising, our algorithm uses Sobel edge detection on the image.

Sobel is chosen because it is highly resistant to contour and noise. SE is also a viable alternative, but it takes a significant amount of time compared with Sobel. After edge detection, further pre-processing is performed by removing small or isolated edges. Our algorithm will automatically remove edges that have single-pixel or an isolated edge with no neighboring edges in the adjacent pixels.

5. Road segmentation

A computationally inexpensive method was employed to generate segmentation for the road pixels. The method to identify pixels into the road or background was based on HSV color space, which undertook a thresholding operation.

First of all, the algorithm will take hundreds of random pixel values on the image part which are likely classified as the road in the analysis. Analysis of those random pixel values was used to find the median value of those sample pixels. Secondly, thresholding operation is performed. The pixel value in range with the aforementioned median value will be given maximum value (white), otherwise, it will be given the minimum value (black). This method will provide an efficient early rejection to bounding box which correspond the road.

The scoring principles of output pixels are such as the following:

$$O_n(x,y) = \begin{cases} 255, & \text{if } I_n(x,y) \in R_n \\ 0, & otherwise, \end{cases} \quad (3)$$

where $I_n(x, y)$ denotes the pixel value at the source image cartesian coordinates whereas $O_n(x, y)$ is the corresponded pixels at the output image. As mentioned earlier, the source image is a 3-channel HSV image, yet the output image is a single channel binary image. Thus, the algorithm will apply equation (3) in each channel of the image, then it will combine those channels into one binary image using OR operation that can be written as the following:

$$O_{binary} = O_1 \wedge O_2 \wedge O_3. \quad (4)$$

The resulting binary image is illustrated in Figure 2 which shows the road pixels as white and the non-road pixels as black. Binary images will also be used for scoring the generated proposals.

Figure 4. The image is divided into 7 regions. The width of each region is 25 percent of the image's width, and the stride is 12.5

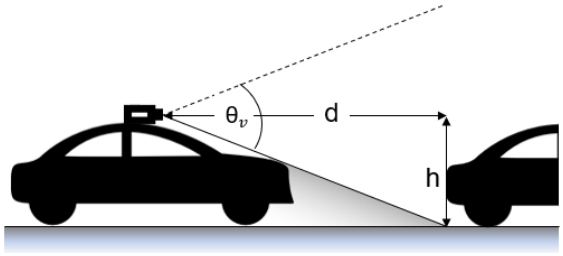

Figure 3. Parameters for region division.

6. Region division and entropy measurement

Our region division methods basically use the same principle as applying size constrain on the proposal. However, instead of specifying the maximum size of a proposal intuitively, our algorithm uses the known information from the hardware setup and exploit this method to improve the speed and proposal distribution on the image. The image is divided into overlapping regions with the same size which control the initial number of proposals based on the entropy of said region while also limit the maximum size of each proposal. Each region will be processed simultaneously by utilizing the multithreading process on CPU in order to speed up the algorithm.

Much of the previous research on object proposal, the size constraint is also based on the prior knowledge of the dataset that it wants to be tested on. Based on our observation and the characteristics of most self-driving dataset especially KITTI, most target object does not exceed the certain size that corresponds to the camera's opening angle. The number of regions needed in the KITTI dataset images can be acquired by first determining the minimum distance between vehicle d using the following formula:

$$tan\left(\frac{\theta_v}{2}\right)^0 = \frac{h}{d}, \quad (5)$$

The known parameters are h = 1.65 m and θ_v = 35 which represent the position of the camera from the ground and the vertical opening angle respectively. Also, a similar formula written in (6) will be applied for the horizontal opening angle represented as θ_h of the KITTI dataset camera which has a value of 90° to acquire the value of M. M itself represents the field of view at distance d.

$$tan\left(\frac{\theta_h}{2}\right)^0 = \frac{M}{d}. \quad (6)$$

Then we determine the width of each region which is denoted as w_r using the following formula:

$$w_r = \frac{W}{2 \cdot M} \cdot 100, \quad (7)$$

where W value is equal to 2.55 meters. It represents the maximum width of a vehicle allowed in the place in which the KITTI dataset took place. Thus, the number of regions in one image denoted as n can be calculated as:

$$n = \frac{100}{(w_r \cdot overlap)} - 100, \quad (8)$$

To make sure no object is missed or partially detected, the minimum region's overlap or stride should be at least 50 percent of the region's width. Figure 5 shows the divided region of our algorithm based on the KITTI dataset parameter. There are seven regions where each region's width is 25 percent of the original image. This means that there is a 50 percent overlap among each region. For example, region number 1 has an overlapping area with whereas region number 2 has overlapping areas with region 1 and

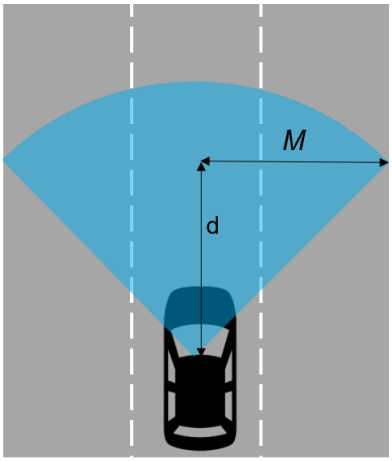

Figure 5. **Region division for improving speed and recall rate of the algorithm.**

3, and so on. Using all the known parameters, the value of n in a single image of the KITTI dataset is roughly seven regions. Note that we have rounded up the value of w_r from 24.70 percent to 25 percent of the image's width. It is important to note that one can easily determine the value of w_r and n without having prior knowledge of the hardware setup as long as the maximum size of the target object in the image is known. However, the aforementioned formula can provide insight into how to determine the size of the region.

Our algorithm utilized window scoring proposal methods for generating object proposals. In this step, our algorithm will generate many proposals using dense sliding window mechanism. The initial proposal does not contain any score and densely distributed depending on the corresponding region entropy

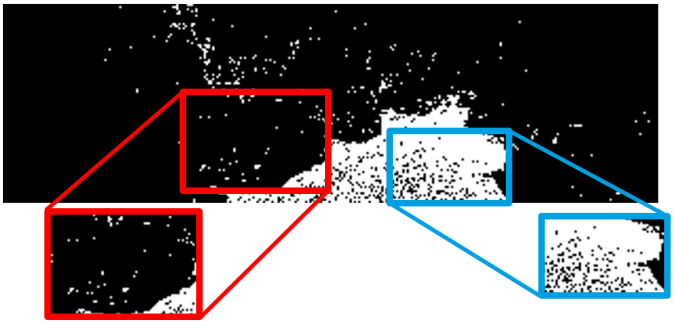

Figure 6. **The number of white pixels which represent the road can be used to improve the ranking of proposal, where proposal containing more white pixels will be ranked**

written in (2). The number of proposals that will be generated in each region can be determined using the following equation:

$$\%P = \left(1 - \frac{|\sigma + H|}{\sigma}\right) \cdot 100\%, \quad \text{if } |\sigma + H| \leq \sigma, \quad (9)$$

where P represents the percentage from the maximum number of proposals that the user wants to generate in a region. The value of σ is a constant which varies for each dataset. For KITTI dataset, the optimum value is empirically set to 4, whereas H denotes the entropy for the current region which is explained as in (2). In a case where the specified condition is not satisfied then the value of P would be equal to 0, which means that no proposals would be generated in the said region.

7. *Proposal scoring and ranking*

Using prior knowledge of hardware setup and dataset characteristics, the proposals from the object proposal generation can be reduced and better ranked to get the ones which most likely contain an object. Thus, we implement additional ranking for the generated proposal based on parameters such as size and location.

Firstly, the proposals will be evaluated based on the edge contained insides. The number of edges inside a proposal has been known to be an effective measure to score a proposal. To improve the calculation of edges in the scoring algorithm, the edges in edge map are clustered into groups. The clustering algorithm will group pixels adjacent to each other as one edge group. After using sliding window mechanism to generate initial proposals, the proposals are scored based on how many edges are inside which corresponds to the equation (1).

Aside from the edge intensity scores, it is necessary to include scores from other cues. Thus, the proposals are also evaluated based on the road segmentation algorithm, which is based on the binary image acquired by applying equations (3) and (4) on the RGB images. The example of the resulting image can be seen in Figure 6, which also shows example of two proposals; the red one is dominated by black pixels whereas the blue one is dominated by white pixels.

$$R = \frac{1}{b_w \cdot b_h} \cdot \sum_{x,y} I_{(x,y)}, \quad (10)$$

The L score for each proposal can be written as shown in (11). A proposal located at the bottom of the image is expected to have bigger area than an object located at the top. Thus, the score L can be formulated as follows:

$$L = \frac{b_y}{b_w \cdot b_h} \cdot image\ width, \quad (11)$$

,

where b_y denotes the distance from the top of an image to the bottom of the proposal measured in pixels, whereas b_w and b_h represent the width and height of a proposal also measured in pixels.The overall score of each proposal can be calculated as follow:

$$V = S \cdot (\alpha \cdot L + \beta \cdot R), \quad (12)$$

where α and β are constant values denoting t. The value of α=0.6 and β=0.4 are ideal for KITTI dataset. However, R and L do not

need to be calculated in the case where S = 0 and the corresponding proposal will be automatically removed. The algorithm also specifies the minimum value of S to be 0.01, the algorithm will eliminate any proposals lower than those values to improve inference time.

8. Non-maximum suppression

This step is used to reduce the number of overlapping proposals using Jaccard index to calculate the overlap area denoted as J between proposals A and B, as written in (13). The non-maximum suppression (NMS) threshold is set to 70 percent. When the overlap value between two proposals exceeds the threshold, the proposal with less score will be removed.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \quad (13)$$

4. Result and Discussion

For benchmarking purposes, the algorithm is run on Windows PC with Intel 8750H and 24 GB of RAM. The algorithm is written in C++ using Visual Studio IDE. The test will be run on KITTI training dataset which consists of 7801 total images. Since the proposed algorithm does not need any training whatsoever, there is no need to split the dataset and all the images will be used for testing. Before comparing our algorithm with others, we will compare the results of our algorithm with and without region enhancement. Because of the region enhancement is applied after edge detection, the inference time on edge detection part will always be the same.

1. Evaluation Metrics

The most important evaluation metrics of object proposals are speed and recall rate [9], [20]. Object proposal algorithm should be fast enough so that it will not affect the overall object detection performance. It should also have the lowest false negative or the highest recall rate because any object missed by object proposal will also be missed by the classifier network. Edge detection inference time already includes the image denoising algorithm and the preprocessing. The proposals scoring speed depend on the number of edges in edge detection. Therefore, applying a denoising algorithm before edge detection is important because it can reduce the number of edges.

Table 2. Inference time of each component in our algorithm

Process	Inference Time (ms)			
	With Region		Without Region	
	Sobel	SE	Sobel	SE
Edge Detection	3	17	3	17
Proposal Scoring	12	8	37	35
Total	**15**	**25**	**40**	**52**

2. Ablation study

To demonstrate the effectiveness of the proposed region enhancement strategy (which encompasses region division, entropy-based proposal density modulation, and associated scoring refinements), we compare the performance of our REEB algorithm with and without this component.

Table 2 provides a comparison of the inference time for key components. The "With Region" column represents the full REEB algorithm, while "Without Region" represents a baseline where proposals are generated more uniformly across the entire image without the regional adaptation.Table 2 provides a comparison for the algorithm to process a single image before and after the implementation of the region enhancement method. The results show that the implementation of the method can improve the speed considerably because the number and magnitude of edges are significantly less in each region compared to the whole image.

Using Sobel for edge detection also yields better performance since Sobel is faster than SE. However, an algorithm that uses SE has shown to be faster in the proposal scoring stage. This is because the number of edges generated by SE is less than Sobel. Yet, the increase in performance is minuscule and the overall inference time is better when Sobel is used. Before the implementation of the region enhancement method, it took 40 milliseconds for the algorithm which utilize Sobel edge detection. Whereas after using region enhancement method it took only 15 milliseconds to process a single image.

Aside from the improvement in speed, it is important to note that one of the challenges in self-driving dataset is its relatively small target object compared to the background. Meanwhile, the region enhancement could improve the detection performance on smaller objects by dividing the image into several smaller images. Thus, making the target objects come into focus. In Figure 7, the result of detection without region enhancement shows that the proposal mostly encloses a bigger but irrelevant object. On the other hand, with region enhancement, our algorithm is more sensitive to a smaller object. Also note that without region enhancement, the entropy measurement is not applicable. Therefore, decimating the distribution of proposals on the image. The poor distribution of proposal can reduce the recall rate of the algorithm since many proposals are overlapping in the same location, as shown in Figure 7. On the contrary, Figure 8 shows that the proposal distribution is more diverse, thus more locations can be covered by the proposals. The improvement in recall rate is further proven by the results shown in Table 2 and 3, where our method which uses region enhancement has better results in all categories.

3. Comparison with state-of-the-art

The experiments compare our REEB method (using Sobel with region enhancement) against several well-established traditional (non-deep learning) object proposal algorithms: BING [14], Selective Search [16], and Edge Boxes [20]. These baselines were chosen as they represent prominent approaches in window scoring (BING, Edge Boxes) and grouping methods (Selective Search), providing a relevant context for evaluating our algorithm's

Table 3. Comparison with other methods at various number of proposals in KITTI dataset

Methods	Recall Rate (%)									Inference time (ms)
	500 proposals			1000 proposals			2000 proposals			
	IoU 0.3	IoU 0.5	IoU 0.7	IoU 0.3	IoU 0.5	IoU 0.7	IoU 0.3	IoU 0.5	IoU 0.7	
BING [14]	55.5	41.5	33.1	73.3	58.3	49.8	78.7	66.0	56.5	30
Selective Search [16]	**73.0**	47.5	30.0	**84.0**	61.1	42.2	**89.7**	73.0	54.8	4,300
Edge Boxes [20]	66.6	**55.8**	**48.6**	68.4	57.0	49.0	87.6	79.4	70	27
Ours*	54.9	45.0	37.8	68.5	57.5	49.8	80.9	71.0	63.4	40
Ours**	67.3	53.0	43.1	75.2	**61.3**	**56.0**	89.5	**81.0**	**72.1**	**15**

** without region enhancement, ** with region enhancement*

Table 4. Comparison with other methods at 1000 proposals in ITRI dataset

Methods	Recall Rate		
	IoU 0.3	IoU 0.5	IoU 0.7
BING [14]	54	44	32
Selective Search [16]	84	60	34
Edge Boxes [20]	92	86	77
Ours*	87	73	64
Ours**	**98**	**93**	**86**

** without region enhancement, ** with region enhancement*

performance. It is important to note that this comparison focuses

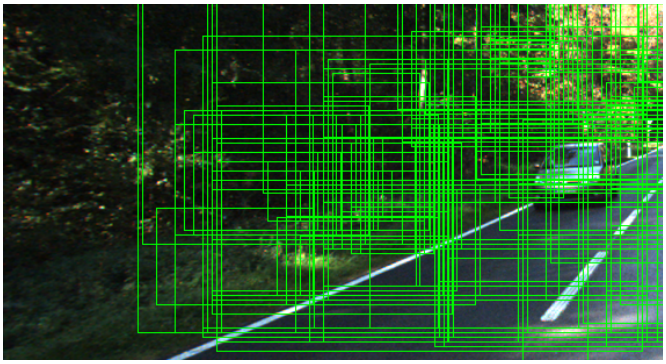

Figure 7. Without region enhancement the car is not detected, the generated proposal also does not distribute very well on the target object.

on the domain of traditional object proposal techniques, rather than contrasting with end-to-end deep learning detectors like YOLO or RPNs, which have different architectural assumptions and typically require extensive training data.

Tables II and III show that our algorithm is faster and has higher recall rate in most categories than the other state-of-the-art algorithm. Our algorithm is also the most consistent as evident in the results of both KITTI and ITRI datasets. The proposed methods perform especially well at the highest IoU, achieving up to 72.1% recall rate on KITTI dataset and 86 % on ITRI dataset.

Selective Search and Edge Box shows comparable results and even exceed our method recall rate in several categories. However, Selective Search is the slowest among the tested methods, therefore not applicable for on-road object detection. Edge Box shows a promising result in both KITTI and ITRI dataset. Nevertheless, the proposed method which utilizes region enhancement is faster and has better recall rate at higher number of proposals.

However, there are several limitations that we would like to address. Firstly, the proposed algorithm is highly dependent on the knowledge of the hardware setup. The most important are the camera placement and its opening angle. Therefore, we provide some insight to adjust those parameters and a default value for dataset with unknown parameters. Secondly, even though region enhancement and bounding box scoring that we propose could highly capitalize on the self-driving dataset characteristic by increasing both speed and the sensitivity on smaller target, it might need further adjustment on another dataset such as Microsoft COCO dataset.

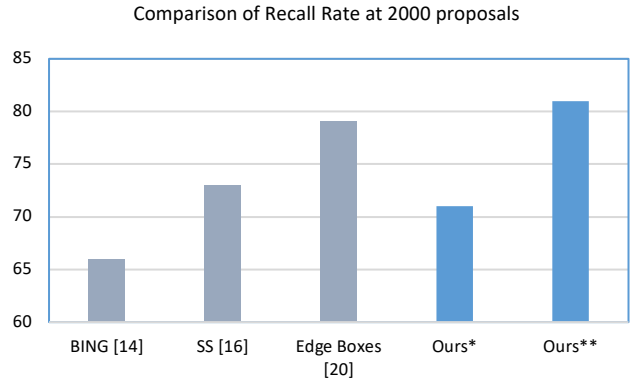

Figure 8. Qualitative comparison of our proposed method against prior object proposal methods. Our proposed method is superior in terms of accuracy and speed.

5. Conclusion

This paper presented the Region Enhanced Edge-Based (REEB) proposal algorithm, a novel window scoring method for fast and robust on-road object proposal generation. REEB effectively combines edge density cues with regional entropy analysis to guide proposal generation and leverages inexpensive road segmentation and proposal location scoring to refine results. Specifically, road segmentation is utilized to penalize proposals likely to be on the road surface, while location scoring favors proposals that adhere to typical perspective constraints in driving

scenes, thereby improving the quality and relevance of the generated proposals.

Experimental results on the KITTI dataset demonstrate the efficacy of REEB. The algorithm achieves an average processing time of 15 milliseconds per image and a recall rate of 72.1% for 2000 proposals at an IoU threshold of 0.7. This performance is notably faster than prominent traditional object proposal methods like Edge Boxes and BING, while maintaining competitive recall rates, particularly at stricter IoU thresholds. Furthermore, REEB operates without requiring any training phase, which reduces its deployment footprint and makes it suitable for embedded systems in autonomous vehicles.

The low computational cost and high recall make REEB a strong candidate for a front-end proposal generator in a two-stage object detection pipeline. For future work, we plan to integrate REEB with lightweight Convolutional Neural Network (CNN) classifiers, such as variants of MobileNet or SqueezeNet, to develop a complete, real-time on-road object detection system. We hypothesize that the high-quality proposals generated by REEB can significantly enhance the performance of these efficient classifiers. Additionally, we intend to explore the implementation of simple bounding box regression techniques, potentially a lightweight regression head, to further refine the localization accuracy (IoU) of the generated proposals, aiming to improve object delineation without substantially compromising the algorithm's speed.

Reference

[1] Zehang Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, May 2006, doi: 10.1109/TPAMI.2006.104.

[2] M. Ryan, "The Future of Transportation: Ethical, Legal, Social and Economic Impacts of Self-driving Vehicles in the Year 2025," *Sci Eng Ethics*, Sep. 2019, doi: 10.1007/s11948-019-00130-2.

[3] J. Cheng, Z. Xiang, T. Cao, and J. Liu, "Robust vehicle detection using 3D Lidar under complex urban environment," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 691–696. doi: 10.1109/ICRA.2014.6906929.

[4] S. Sivaraman and M. M. Trivedi, "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013, doi: 10.1109/TITS.2013.2266661.

[5] A. Mukhtar, L. Xia, and T. B. Tang, "Vehicle Detection Techniques for Collision Avoidance Systems: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2318–2338, Oct. 2015, doi: 10.1109/TITS.2015.2409109.

[6] S. Sivaraman and M. M. Trivedi, "A review of recent developments in vision-based vehicle detection," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2013, pp. 310–315. doi: 10.1109/IVS.2013.6629487.

[7] Y. Kang, H. Yin, and C. Berger, "Test Your Self-Driving Algorithm: An Overview of Publicly Available Driving Datasets and Virtual Testing Environments," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 171–185, Jun. 2019, doi: 10.1109/TIV.2018.2886678.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010, doi: 10.1109/TPAMI.2009.167.

[9] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What Makes for Effective Detection Proposals?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 814–830, Apr. 2016, doi: 10.1109/TPAMI.2015.2465908.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[11] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125. doi: 10.1109/CVPR.2017.106.

[12] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6517–6525. doi: 10.1109/CVPR.2017.690.

[13] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[14] M.-M. Cheng, Y. Liu, W.-Y. Lin, Z. Zhang, P. L. Rosin, and P. H. S. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," *Computational Visual Media*, vol. 5, no. 1, pp. 3–20, Mar. 2019, doi: 10.1007/s41095-018-0120-1.

[15] J. Carreira and C. Sminchisescu, "CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1312–1328, Jul. 2012, doi: 10.1109/TPAMI.2011.231.

[16] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, Sep. 2013, doi: 10.1007/s11263-013-0620-5.

[17] J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marques, and J. Malik, "Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 128–140, Jan. 2017, doi: 10.1109/TPAMI.2016.2537320.

[18] X. Yuan, S. Su, and H. Chen, "A Graph-Based Vehicle Proposal Location and Detection Algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3282–3289, Dec. 2017, doi: 10.1109/TITS.2017.2676182.

[19] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the Objectness of Image Windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012, doi: 10.1109/TPAMI.2012.28.

[20] C. L. Zitnick and P. Dollár, "Edge Boxes: Locating Object Proposals from Edges," in *Computer Vision – ECCV 2014*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 391–405. doi: 10.1007/978-3-319-10602-1_26.

[21] Q. Wu, H. Li, F. Meng, and K. N. Ngan, "Generic Proposal Evaluator: A Lazy Learning Strategy Toward Blind Proposal Quality Assessment," *IEEE Transactions on Intelligent Transportation*

Systems, vol. 19, no. 1, pp. 306–319, Jan. 2018, doi: 10.1109/TITS.2017.2750070.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587. doi: 10.48550/arXiv.1311.2524.

[23] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

[24] H. Kuang, L. Chen, L. L. H. Chan, R. C. C. Cheung, and H. Yan, "Feature Selection Based on Tensor Decomposition and Object Proposal for Night-Time Multiclass Vehicle Detection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 71–80, Jan. 2019, doi: 10.1109/TSMC.2018.2872891.

[25] X. Yuan, X. Cao, X. Hao, H. Chen, and X. Wei, "Vehicle Detection by a Context-Aware Multichannel Feature Pyramid," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1348–1357, Jul. 2017, doi: 10.1109/TSMC.2016.2587483.

[26] Y. Xu, X. Cao, and H. Qiao, "An Efficient Tree Classifier Ensemble-Based Approach for Pedestrian Detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 107–117, Feb. 2011, doi: 10.1109/TSMCB.2010.2046890.

[27] P. Dollár and C. L. Zitnick, "Structured Forests for Fast Edge Detection," in *2013 IEEE International Conference on Computer Vision*, Dec. 2013, pp. 1841–1848. doi: 10.1109/ICCV.2013.231.

[28] Y. Song, Y. Ju, K. Du, W. Liu, and J. Song, "Online Road Detection under a Shadowy Traffic Image Using a Learning-Based Illumination-Independent Image," *Symmetry*, vol. 10, no. 12, p. 707, Dec. 2018, doi: 10.3390/sym10120707.

[29] F. Hermosillo-Reynoso, D. Torres-Roman, J. Santiago-Paz, and J. Ramirez-Pacheco, "A Novel Algorithm Based on the Pixel-Entropy for Automatic Detection of Number of Lanes, Lane Centers, and Lane Division Lines Formation," *Entropy*, vol. 20, no. 10, p. 725, Oct. 2018, doi: 10.3390/e20100725.