ICIC Express Letters
Volume **15**, Number **10**, October **2021**

ICIC International ©2021 ISSN 1881-803X
pp. 1021–1027

BENCHMARK OF EDGE-AWARE FILTERING APPLICATION FOR VEHICLE OBJECT DETECTION

Rini Wongso[1], Meiliana[1] and Williem[1,2]

[1]Computer Science Department
School of Computer Science
Bina Nusantara University
JL. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ rwongso; williem003 }@binus.ac.id; meiliana@binus.edu

[2]Verihubs
JL. Let. Jend. S. Parman Kav. 28 SOHO CAPITAL Lt. 32, Grogol Petamburan
Jakarta Barat 11470, Indonesia
williem@verihubs.com

Received January 2021; accepted April 2021

Abstract. *This paper introduces a benchmark of various object detectors with various edge-aware filtering applications as the preprocessing methods. Traffic scenes often have various environments; thus image preprocessing is necessary to obtain good object detectors. We apply tone adjustment, L_0 smoothing and detail manipulation as the preprocessing with state-of-the-art deep edge aware filtering methods. We utilize three object detectors: Free Anchor, Guided Authoring, and NAS-FPN (Neural Architecture Search – Feature Pyramid Network). Experimental results show that the edge-aware filtering application is helpful for the object detector when the input image quality is not standard. For future work, we expect to compare more methods and preprocessing methods.*
Keywords: Object detection, Edge-aware filtering, Benchmark, Deep learning

1. **Introduction.** Object detection has gained attention from industry and academia due to its wide range of applications, such as traffic surveillance, monitoring, and robotic. For the past few years, the state-of-the-art object detectors have achieved the significant performance due to the development of deep learning technologies [1, 2, 3]. Most object detectors are trained in general dataset with many object categories, but there are also object detectors that focus on specific object category, such as pedestrian detection, vehicle detection, and face detection, to improve the accuracy even though the same object category also exists in the general object detector dataset. For example, the car object is included in both MS COCO dataset [4] and UA-DETRAC dataset [5]. As most developers could not produce a new dataset with large number of images, it is interesting to analyze how the performance of pre-trained model on general object detector dataset on the specific object detector dataset.

Edge-aware filtering is important to various image processing applications, such as detail manipulation [6], tonal adjustment [7], and L_0 smoothing [8], to preserve the major image structures. In addition, those edge-aware filtering applications can be used as the preprocessing of the input image of object detector methods. Note that it is necessary to have input image with a good quality for deep object detection. Li et al. have performed a benchmark for dehazing methods in order to improve the performance of object detection [9]. As traffic scene is often to have low quality of image, it is interesting to apply various image processing enhancement to seeing the refinement of each object detector.

DOI: 10.24507/icicel.15.10.1021

In this paper, we perform a benchmark evaluation between state-of-the-art detectors on the traffic scene object detection dataset. We also perform experiment based on analysis of what kind of object detector is more suitable on the traffic scene. In addition, we also apply various edge-aware filtering application as the preprocessing method for object detection and analyze the performance. Experimental results show the performance of edge-aware filtering applications on various use-cases.

The contributions of this paper are as follows:

- Integration framework of edge-aware filtering application and object detection;
- Analysis of the usage of edge-aware filtering to improve object detection performance for traffic scenes;
- Exhaustive benchmarking of various integration scenarios.

The next section will discuss about related works for both methods of edge-aware filtering applications and object detector. The third section describes the proposed secnarios of this research and experimental results are provided in the subsequent section. Conclusions are given in the last section.

2. **Related Works.** This paper focuses on analyzing the effect of edge-aware filtering applications on object detectors. Thus, we select the representative methods for each edge-aware filtering and object detector.

2.1. **Edge-aware filtering.** Edge-aware filtering can be categorized into two groups: local filter based approaches and optimization based approaches. The local filter based approaches [10] utilize the information of pixels inside a local filter to assign a new value to a pixel. On the other hand, the optimization based approaches [8] solve an objective function that is set based on the need. Recently, deep learning based approaches have been introduced to solve the optimization function in more efficient way [11].

2.2. **Object detection.** Object detection methods can be categorized into two groups: two-stage pipeline and single-stage pipeline. Two-stage pipeline firstly generates the region proposals and the regress the bounding box coordinates at the second stage [12]. On the other hand, the single-stage pipeline predicts the bounding box directly without any region proposal [13]. While the single-stage pipeline depends on the predefined anchors, there are some works that focus on anchor-free methods [3].

3. **Proposed Scenarios.** In this paper, we analyze the performance of object detector methods on traffic scenes. In addition, a preprocessing method is applied to the input image. The preprocessing method is based on edge-aware filtering application. Figure 1 shows the overview framework of the proposed scenarios. The proposed scenario consists of three main steps: edge-aware filtering, preprocessing, and object detector.

In edge-aware filtering phase, we select 1 out of 6 available methods. We utilize various deep guided filter methods proposed by Wu et al. [11]. The first option is no filtering (represented by None in this paper) which means that there is no preprocessing applied to the input image. The second option is Guided Filter (GF) [10] which is a non-deep learning edge-aware filtering method. It is one of the most representative methods in edge-aware filtering domain. While the original guided filter is not differentiable, the Deep Guided Filter (DGF) reformulates the guided filter so that it can be differentiable. The fourth option is the Deep Guided Filter Advanced (DGFA) which is a deep guided filter added by a task-specific guidance map in order to add task related computation. A single guided filter layer tends to fail on complex task and scene; thus there is the modification of the guided filtering layer using the convolutional layer. The modification results in the fifth and sixth options, which are Deep Convolutional Guided Filter (DCGF) and Deep Convolutional Guided Filter Advance (DCGFA). In this paper, we employ those six edge-aware filtering options as part of the image preprocessing phase.

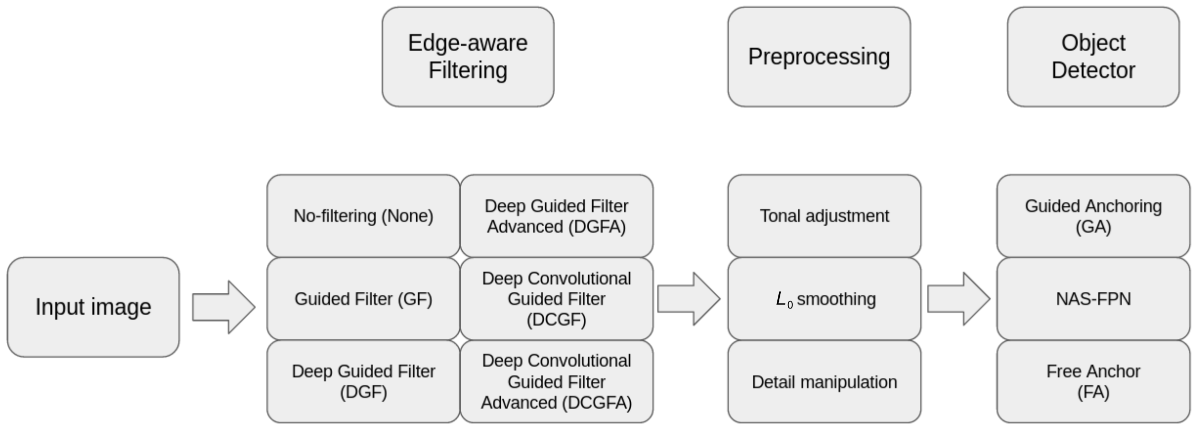

FIGURE 1. Overview of the proposed benchmark scenarios. The input image is processed by selecting each method on each step. The proposed scenarios are built in order to evaluate the effect of edge-aware filtering application on the performance of object detectors.

In preprocesing phase, we select 1 out of 3 available methods, which are tonal adjustment [7], L_0 smoothing [8], and detail manipulation [6]. The preprocessing phase has been trained together with the deep guided filter work [11] using the ground truth of the representative works in each application. The tonal adjustment is done to retouch the input image in more aesthetic way. Note that the ground truth is labeled by human experts. L_0 smoothing removes the minor edge while performs sharpening to the major edge in the image. Finally, the detail manipulation boosts the features by multiple scales in order to magnify the edges. In this paper, we analyze the effect of each image processing method on the object detector.

In the object detector phase, we select 1 out of 3 available methods, which are NAS-FPN [1], Guided Anchoring (GA) [2], and Free Anchor (FA) [3]. Note that we utilize the same framework (RetinaNet-50 [13]) for all methods. NAS-FPN applies neural architecture search to finding the best architecture design for the pyramid feature network. While NAS-FPN focuses on refining the extracted features, guided anchoring and free anchor focus on setting up the anchor of the bounding box. Guided Anchoring (GA) utilizes the semantic features which are benefit to guide the anchor; thus it results in the center position of the box, together with the scale and the aspect ratio of the bounding box. Free anchor also optimizes the anchor selection for object detection. Instead of using handcrafted anchor assignment, it formulates the detector learning as a maximum likehood estimation process which results in a bag of anchor for each possible object. In this paper, we analyze the performance of each object detector together with the preprocessing methods.

4. **Experimental Results.** This paper employs MM Detection Toolbox and Benchmark [14] which contains state-of-the-art object detectors. Note that the object detector methods in the toolbox has been trained using the same dataset, which is MS COCO [4]. The MS COCO dataset contains up to 80 general object categories labeled for object detection and instance segmentation. As we focus on the traffic scene object detection, we utilize UA-DETRAC object detection dataset which captured various traffic scenes from video surveillance [5]. The UA-DETRAC dataset consists of various vehicle categories, such as SUV, Sedan, and Bus. Note that we do not perform fine tuning process to UA-DETRAC dataset; thus the domain between the training and testing dataset is different. As the object categories are different between both datasets, we simplify the vehicle related object categories in both UA-DETRAC and MS COCO dataset into one

object category. In addition, we also remove the detected objects that are in the ignored regions of UA-DETRAC dataset. Note that UA-DETRAC is designed for vehicle tracking problem; thus there are lots of repetitive frames. Thus, we sample the testing set by taking the first frame for each 100 frames. Thus, we can reduce the similar scenes and focus on the object detection performance.

Table 1 shows the mean-Average Precision (mAP) comparison between each object detector and the edge-aware filtering applications on the UA-DETRAC testing dataset. In overall, the edge-aware filtering application does not improve the performance of object detectors. Note that UA-DETRAC testing dataset consists of various traffic scenes: daytime, nighttime, crowded scene. Thus, each application might not work well for all conditions.

Table 1. Benchmark of object detection methods with edge-aware filtering applications (mAP)

	None	GF	DGF	DGFA	DCFG	DCFGA
Free Anchor + Tonal Adjustment	0.783	0.767	0.774	0.768	0.776	0.775
Free Anchor + L_0 Smoothing	0.783	0.774	0.774	0.774	0.770	0.772
Free Anchor + Detail Manipulation	0.783	0.782	0.781	0.773	0.780	0.780
NAS-FPN + Tonal Adjustment	0.607	0.587	0.601	0.596	0.589	0.592
NAS-FPN + L_0 Smoothing	0.607	0.597	0.599	0.595	0.599	0.598
NAS-FPN + Detail Manipulation	0.607	0.592	0.594	0.589	0.588	0.593
Guided Anchoring + Tonal Adjustment	0.694	0.671	0.675	0.669	0.687	0.675
Guided Anchoring + L_0 Smoothing	0.694	0.677	0.680	0.676	0.677	0.679
Guided Anchoring + Detail Manipulation	0.694	0.688	0.686	0.668	0.683	0.674

Table 2. Benchmark of object detection methods with edge-aware filtering applications (mAP) on MVI_40775 dataset

	None	GF	DGF	DGFA	DCFG	DCFGA
Free Anchor + Tonal Adjustment	0.804	0.800	0.785	0.802	0.809	0.811
Free Anchor + L_0 Smoothing	0.804	0.786	0.786	0.777	0.795	0.794
Free Anchor + Detail Manipulation	0.804	0.787	0.811	0.812	0.778	0.804
NAS-FPN + Tonal Adjustment	0.682	0.707	0.715	0.674	0.691	0.699
NAS-FPN + L_0 Smoothing	0.682	0.666	0.699	0.691	0.682	0.691
NAS-FPN + Detail Manipulation	0.682	0.707	0.697	0.666	0.650	0.674
Guided Anchoring + Tonal Adjustment	0.707	0.699	0.699	0.682	0.764	0.723
Guided Anchoring + L_0 Smoothing	0.707	0.631	0.650	0.642	0.658	0.650
Guided Anchoring + Detail Manipulation	0.707	0.739	0.755	0.674	0.739	0.682

Table 3. Benchmark of object detection methods with edge-aware filtering applications (mAP) on MVI_40743 dataset

	None	GF	DGF	DGFA	DCFG	DCFGA
Free Anchor + Tonal Adjustment	0.851	0.859	0.851	0.832	0.871	0.859
Free Anchor + L_0 Smoothing	0.851	0.859	0.858	0.866	0.858	0.858
Free Anchor + Detail Manipulation	0.851	0.846	0.850	0.849	0.849	0.850
NAS-FPN + Tonal Adjustment	0.767	0.767	0.787	0.768	0.774	0.774
NAS-FPN + L_0 Smoothing	0.767	0.767	0.767	0.807	0.761	0.767
NAS-FPN + Detail Manipulation	0.767	0.774	0.773	0.788	0.767	0.787
Guided Anchoring + Tonal Adjustment	0.822	0.810	0.788	0.812	0.815	0.817
Guided Anchoring + L_0 Smoothing	0.822	0.827	0.808	0.829	0.801	0.815
Guided Anchoring + Detail Manipulation	0.822	0.801	0.820	0.804	0.803	0.815

To perform better analysis, we do the evaluation on several various specific conditions. Tables 2 and 3 show the mAP of specific data in UA-DETRAC dataset, which are MVI_40775 and MVI_40743. Both data are captured at night which have lack of texture and detail. With those data, we want to ensure that the edge-aware filtering application can improve the object detection performance by adjusting the tonal and manipulating the detail. It is clear that the L_0 smoothing could not refine the performance of object detector, but both tonal adjustment and detail manipulation are able to increase the mAP metric. Though there is no consistent performance of each edge-aware filtering method, it shows that the edge-aware filtering can improve the magnitude of the details, resulting in better precision.

Figure 2 shows the qualitative comparison of MV_40755 dataset with GA object detector. It shows that the edge-aware filtering operators help the object detector to detect missing objects that are not detected without any preprocessing method applied.

Figure 2. Qualitative comparison of GA object detector with edge-aware filtering applications on MV_40755 dataset. First column: Tonal adjustment; Second column: L_0 smoothing; Third column: Detail manipulation. First row: None; Second row: GF; Third row: DGF; Fourth row: DGFA; Fifth row: DCGF; Sixth row: DCGFA.

To analyze the performance of the object detectors with detail manipulation on low quality image, we downsample the image by 4 and then upsample again to the original size. Thus, the detail in the input image is missing. Table 4 and Figure 3 show the quantitative and qualitative comparison of UA-DETRAC dataset. It shows that the detail manipulation processing helps the object detector to detect more objects. However, the

TABLE 4. Benchmark of object detection methods with detail manipulation application on low-resolution UA-DETRAC dataset

	None	GF	DGF	DGFA	DCFG	DCFGA
Free Anchor + Detail Manipulation	0.641	0.682	0.686	0.652	0.685	0.663
NAS-FPN + Detail Manipulation	0.477	0.466	0.468	0.475	0.458	0.465
Guided Anchoring + Detail Manipulation	0.497	0.512	0.514	0.491	0.512	0.495

FIGURE 3. Qualitative comparison of various object detectors with detail manipulation on low quality MV_40711 dataset. First column: NAS; Second column: GA; Third column: FA. First row: None; Second row: GF; Third row: DGF; Fourth row: DGFA; Fifth row: DCGF; Sixth row: DCGFA.

final result still depends on the object detector type. Detail manipulation is suitable with free anchor and guided anchoring, but it does not refine the performance of NAS-FPN.

5. **Conclusions.** This paper performed a benchmark of various object detectors with various edge-aware filtering based image preprocessing on traffic scenes. We utilized state-of-the-art object detectors trained with neural architecture search, guided anchor, and free anchor. In addition, we applied state-of-the-art deep guided filter image which could be adapted with tonal adjustment, L_0 smoothing, or detail manipulation. Experimental results showed that the edge-aware filtering methods were not suitable for general condition, but it helped the object detectors in extreme condition, such as nighttime and low resolution.

Acknowledgements. We gratefully acknowledge the support of grant from Directorate of Higher Education with grant number: 225/SP2H/LT/DRPM/2019.

REFERENCES

[1] G. Ghiasi, T. Lin and Q. V. Le, NAS-FPN: Learning scalable feature pyramid architecture for object detection, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.7036-7045, 2019.
[2] J. Wang, K. Chen, S. Yang, C. C. Loy and D. Lin, Region proposal by guided anchoring, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.2960-2969, 2019.
[3] X. Zhang, F. Wan, C. Liu, R. Ji and Q. Ye, FreeAnchor: Learning to match anchors for visual object detection, *Neural Information Processing Systems*, vol.32, 2019.
[4] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick and P. Dollár, Microsoft COCO: Common objects in context, *arXiv Preprint*, arXiv: 1405.0312v3, 2015.
[5] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M. Yang and S. Lyu, UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking, *Computer Vision and Image Understanding*, vol.193, DOI: 10.1016/j.cviu.2020.102907, 2020.
[6] Z. Farbman, R. Fattal, D. Lischinski and R. Szeliski, Edge-preserving decompositions for multi-scale tone and detail manipulation, *ACM Trans. Graph.*, vol.27, no.3, 2008.
[7] V. Bychkovsky, S. Paris, E. Chan and F. Durand, Learning photographic global tonal adjustment with a database of input/output image pairs, *2011 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.97-104, 2011.
[8] L. Xu, C. Lu, Y. Xu and J. Jia, Image smoothing via L_0 gradient minimization, *ACM Trans. Graph.*, vol.30, no.6, 2011.
[9] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng and Z. Wang, Benchmarking single-image dehazing and beyond, *IEEE Trans. Image Processing*, vol.28, no.1, pp.492-505, 2019.
[10] K. He, J. Sun and X. Tang, Guided image filtering, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.35, no.6, pp.1397-1409, DOI: 10.1109/TPAMI.2012.213, 2013.
[11] H. Wu, S. Zheng, J. Zhang and K. Huang, Fast end-to-end trainable guided filter, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1838-1847, 2018.
[12] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.39, no.6, pp.1137-1149, DOI: 10.1109/TPAMI.2016.2577031, 2017.
[13] T. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, Focal loss for dense object detection, *2017 IEEE International Conference on Computer Vision (ICCV)*, pp.2999-3007, 2017.
[14] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy and D. Lin, MMDetection: Open MMLab detection toolbox and benchmark, *arXiv Preprint*, arXiv: 1906.07155, 2019.