Saliency Guided 2D-Object Annotation for Instrumented Vehicles

Venkatesh G M*
Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland
venkatesh.gurummunirathnam@insight-centre.org

Feiyan Hu*
Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland
feiyan.hu@dcu.ie

Noel E. O'Connor
Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland
noel.oconnor@dcu.ie

Alan F. Smeaton
Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland
alan.smeaton@dcu.ie

Zhen Yang
IoV Innovation Center
Huawei
Shanghai, China
yang.zhen@huawei.com

Suzanne Little
Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland
suzanne.little@dcu.ie

***Abstract*—Instrumented vehicles can produce huge volumes of video data per vehicle per day that must be analysed automatically, often in real time. This analysis should include identifying the presence of objects and tagging these as semantic concepts such as car, pedestrian, etc. An important element in achieving this is the annotation of training data for machine learning algorithms, which requires accurate labels at a high-level of granularity. Current practise is to use trained human annotators who can annotate only a limited volume of video per day. In this paper, we demonstrate how a generic human saliency classifier can provide visual cues for object detection using deep learning approaches. Our work is applied to datasets for autonomous driving. Our experiments show that utilizing visual saliency improves the detection of small objects and increases the overall accuracy compared with a standalone single shot multibox detector.**

***Index Terms*—deep learning, visual saliency, object detection, data annotation, autonomous vehicles**

I. Large Scale Annotation

Advanced Driver Assistance Systems (ADAS) (e.g., parking assistance, lane detection, collision avoidance, etc.) are generally built on annotations of sensor data recordings of road traffic objects, events and scenes. To develop and evaluate such applications, extremely large volumes of automotive video and other sensor data is gathered using specially instrumented vehicles that are directed to capture data for specific scenarios. These cars may generate in excess of 8TB of video data each per day (∼8hrs) of operation. This data is recovered and then requires exhaustive, pixel-level annotation of each frame to be used for training and evaluation of computer-based driver assistance systems. This is non-trivial and has very high costs of manual labelling and there are many methodologies proposed to improve the efficiency of such human-in-the-loop annotation [1]–[3].

As advances in instrumented vehicles develop, the data generated increases exponentially and if we desire dense data annotations of this data, it could require many years of human annotation. This will not scale quickly enough to support advances in deep visual learning. Clearly, we must introduce certain pre-processing steps to aid the annotators to pre-localise objects in a scene. Saliency is an estimate of the regions that draw a viewer's attention and models are generally trained based on human experiments tracking eye gaze and attention on generic images. Visual saliency could be a option in aiding annotators to examine the most important segments in an image and to correct the position of the objects' bounding boxes in the image or to label objects which might be missed by an object detection module. In this paper, we propose visual saliency as a possible pre-processing step before passing to human annotators or to an object detection module for object localisation and recognition in the image scene.

The paper is structured as following. Section II describes related research in saliency prediction, object detection and automatic annotation of instrumented vehicle video. Section III briefly introduces the KITTI and VOC datasets on which experiments are conducted. Section IV has details of how we constructed our processing pipeline while Section V presents our experimental results. Section VI include conclusions and future work.

II. Related Work

A. Visual Saliency

Image saliency detection methods can be classified into bottom-up models [4]–[10] and top-down models [11]–[15]. Bottom-up models rely heavily on prior knowledge about human visual perception. Bottom-up methods normally have the following: processing, pre-processing, feature extraction, and saliency estimation based on saliency cues. In feature

The authors acknowledge the contributions and advice of Bonita Liu, Huawei.

*equal contributions.

978-1-7281-4673-7/19/$31.00 ©2019 IEEE

extraction, normally low-level features such as color, texture, SIFT, etc. are computed depending on which saliency cues are used. There are many theories about how the human visual system derives saliency cues such as contrast prior [4], background prior [7]–[9], and compactness prior [16]. In addition to methods that use saliency clues, there are also approaches that use sparse representation [5], cellular automata [6], random walks [9], low-rank recovery [10]. To produce a final saliency map, multiple saliency cues are normally combined. Modern top-down model normally use convolutional neural networks to generate saliency map for images [12]–[14] and videos [15].

One of the main advantages of bottom-up methods is interpretability; for example if more than just saliency cues contribute to the final prediction of saliency maps, we could determine how much contribution each saliency cue made by examining the weights. However, as these approaches require the engineering of selection and combination saliency cues, the capacity to generalize is limited to the scenario in which the saliency cues are designed to work.

Top-down models are results-driven, which require the annotation of saliency images/videos. The annotation process is normally dependent on humans to label each pixel in the input media. A computational model is then constructed to make predictions based on the input media. As the difference between annotations between human input and prediction by computational models should be minimized, weights of the computational models are updated based on the difference between human labels and computer model predictions. Top-down models are less dependent on prior human knowledge of human visual systems, so these approaches are normally associated with statistical models, machine learning, and more recently, deep learning. In the current state-of-the-art, deep learning shows high performance in saliency detection tasks.

B. Object Detection

Object detection is a process of identifying and localizing multiple semantic objects of a certain class in images. There are datasets that have been released for object detection challenges and specific performance metrics have been developed to take into account the spatial position of detected objects and the accuracy of the predicted categories. Deep learning techniques have emerged as powerful methods for learning feature representations automatically from data. In particular, these techniques have provided significant improvement for object detection, a problem that has attracted enormous attention in recent years. Object detectors can be organized into two main categories:

Two stage detection frameworks, which includes a pre-processing step for region proposal such as RCNN [17], SPPNet [18], Fast RCNN [19], Faster RCNN [20], RFCN [21], Mask RCNN [22] and Light Head RCNN [23], making the overall pipeline two stages. In a two stage detection framework, category-independent region proposals are generated from an image, CNN [24] features are extracted from these regions, and then category-specific classifiers are used to determine the category labels.

One stage detection frameworks, or region proposal free frameworks like DetectorNet [25], OverFeat [26], YOLO [27] and SSD [28], are single pass methods that do not separate detection proposals, making the detection pipeline single stage. A single stage detection framework refers to an architecture that directly predicts class probabilities and bounding box offsets from full images with a single feed-forward CNN network in a monolithic setting that does not involve region proposal generation or post classification. The approach is simple because it completely eliminates region proposal generation and the subsequent pixel or feature re-sampling stages. In our experiments, we used the SSD300 pre-trained on MS-COCO dataset with 20 classes for the object detection task in the automatic annotation pipeline.

C. Automatic Annotation of Instrumented Vehicle Video

Object detection and recognition using deep learning based approaches require a huge number of data samples covering different scenarios for training and testing. Generating annotated labels for training the deep network requires both time and skilled annotators. To address this annotation requirement, web-based generic image annotators are proposed in [29] and more recent DNN based [30]–[32] There are public available tools such as LabelImg[1], VGG Image Annotator[2], Humans-in-the-Loop[3], Anno-Mage[4].

III. Datasets

The **KITTI** [33] benchmark suite is a dataset covering many types of data in the context of autonomous driving. All data provided in KITTI was captured on a vehicle platform during driving. The main sub-tasks for creating the dataset are 2D/3D object detection, 2D/3D object tracking, object orientation estimation, optical flow estimation using data such as RGB images, depth images, optical flow, odometry and so on. Labels in the KITTI dataset include Car, Truck, Van, Pedestrian, Cyclist, Traffic Light, Pole, Bus, Train, Motorcycle. In this paper we focus on the 2D object detection sub-task which consists of 7,481 training images and 7,518 test images, comprising a total of 80,256 labelled objects.

The **VOC** dataset is a popular object detection benchmark. We conduct experiments on the testing set realised on 2007, since this is the only VOC testing set that releases annotations for us to evaluate the effect of adding saliency guidance in object annotation pipeline.

IV. Object Annotation Pipeline

There is much research on saliency and object detection, yet little has focused on combining saliency prediction and object detection in the context of ADAS, especially in the case of annotation for instrumented vehicles. In the paper we propose a pipeline that incorporates pre-trained saliency prediction

[1]https://github.com/tzutalin/labelImg
[2]http://www.robots.ox.ac.uk/ vgg/software/via/
[3]https://humansintheloop.org/
[4]https://github.com/virajmavani/semi-auto-image-annotation-tool

and object detection algorithms, given the hypothesis that the human attention/perception mechanism plays a role in vehicle driving. We set out to examine how a saliency map algorithm trained to predict human attention could affect object detection in the context of ADAS.

The proposed pipeline for object annotation is shown in Fig. 2. The input to the pipeline is an RGB image and the output is annotated objects in the image. The pipeline consists of two main blocks, a visual saliency prediction block which is used to compute the salience map from the input frame and an object detection block to localise and classify objects in the scene. We use SalGAN [14] and Single Shot MultiBox Detector(SSD) [28] trained on the MS-COCO [34] dataset for annotation of objects in the KITTI [33] dataset. In this work, we have considered cars and pedestrians (person) only for performance evaluation. The following sub-section gives a brief outline of different modules employed in the pipeline.

A. Saliency map generation

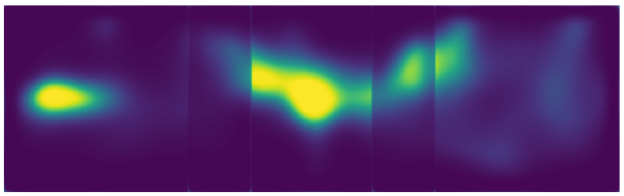
We use a pre-trained SalGAN on SALICON [35] to generate a saliency map for KITTI and VOC2007 testing. Since SalGAN is originally trained using resized images with size 192×256 and the input of the KITTI dataset are quite different with size 375×1242, we segment each KITTI image into 3 images with equal width and height. Neighbouring segments have same size of overlapping regions. In our experiments, this overlapping width is 66 pixels. Then all segments of images are used to compute a saliency map, and saliency maps of image segments are combined to generate a final saliency map with the same size as the KITTI input images. Overlapping parts are computed as the average of two contributing neighbouring saliency map. Figure 1a shows an example of a generated saliency map on the KITTI dataset. We can observe overlapping part being distorted, but the distortion are rectified in the binaraized saliency map thus have little effect on the generation of bounding boxes. All VOC2007 images use their original size to generate the saliency map. Each saliency map takes about 30ms to generate.

B. ROIs generation

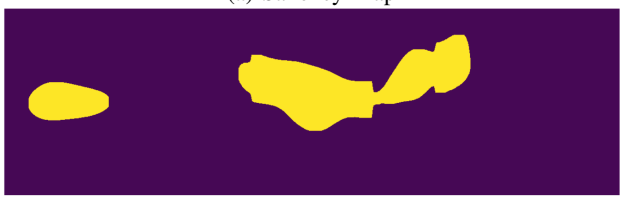
Saliency has been shown to be effective in image cropping [36], [37] around objects of interest so, in this paper, we generate regions of interest (ROIs) based on the heatmap from saliency prediction. To do this, we first erode and then dilate the saliency map with a 5×5 kernel. The image binarization uses Otsu's threshold [38], which is determined automatically. The erosion, dilation and thresholding uses code implemented in OpenCV. Figure 1b shows the binarized maps based on the saliency map in Figure 1a while Figure 1c shows generated bounding boxes of ROIs.

C. SSD detection within ROIs

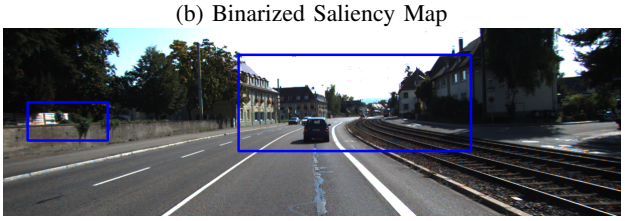
This is the final stage of the processing pipeline. As mentioned earlier, we use off-the-shelf existing implementations to build the pipeline for aiding the annotation framework. More details about SSD can be found in [28]. We used SSD300 pre-trained on the MSCOCO dataset with 20 classes. The input to the module is all possible salient regions generated by the ROI generation module. The regions are resized to 300x300 before passing to the SSD300 module for object detection. The bounding box co-ordinates of detected objects are mapped to the original image for final evaluation by mean Average Precision (mAP). Figure 3 shows two examples of detected objects using proposed pipeline and standard SSD benchmark.



(a) Saliency Map



(b) Binarized Saliency Map



(c) Generated Saliency ROIs

Fig. 1: Generation of Saliency ROIs.

V. Experimental Results

In this section, we describe and analyse experimental results on the training set of KITTI 2D object detection, and the testing set of VOC2007.

A. Implementation

The pipeline runs on a single GPU (GeForce GTX TITAN X, 12GB RAM) using SSD implemented in Keras[5] with a Tensorflow[6] backend. SSD is pre-trained on MSCOCO as an off-the-shelf module. We use SalGAN implemented in Pytorch[7] pre-trained on SALICON. For the evaluation, we report mean Average Precision (mAP) as reported in VOC. During object detection inference, all input images of SSD are resized to 300×300. No fine tuning is conducted on either the SSD or SalGAN models.

[5]https://github.com/fchollet/keras
[6]https://www.tensorflow.org
[7]https://pytorch.org

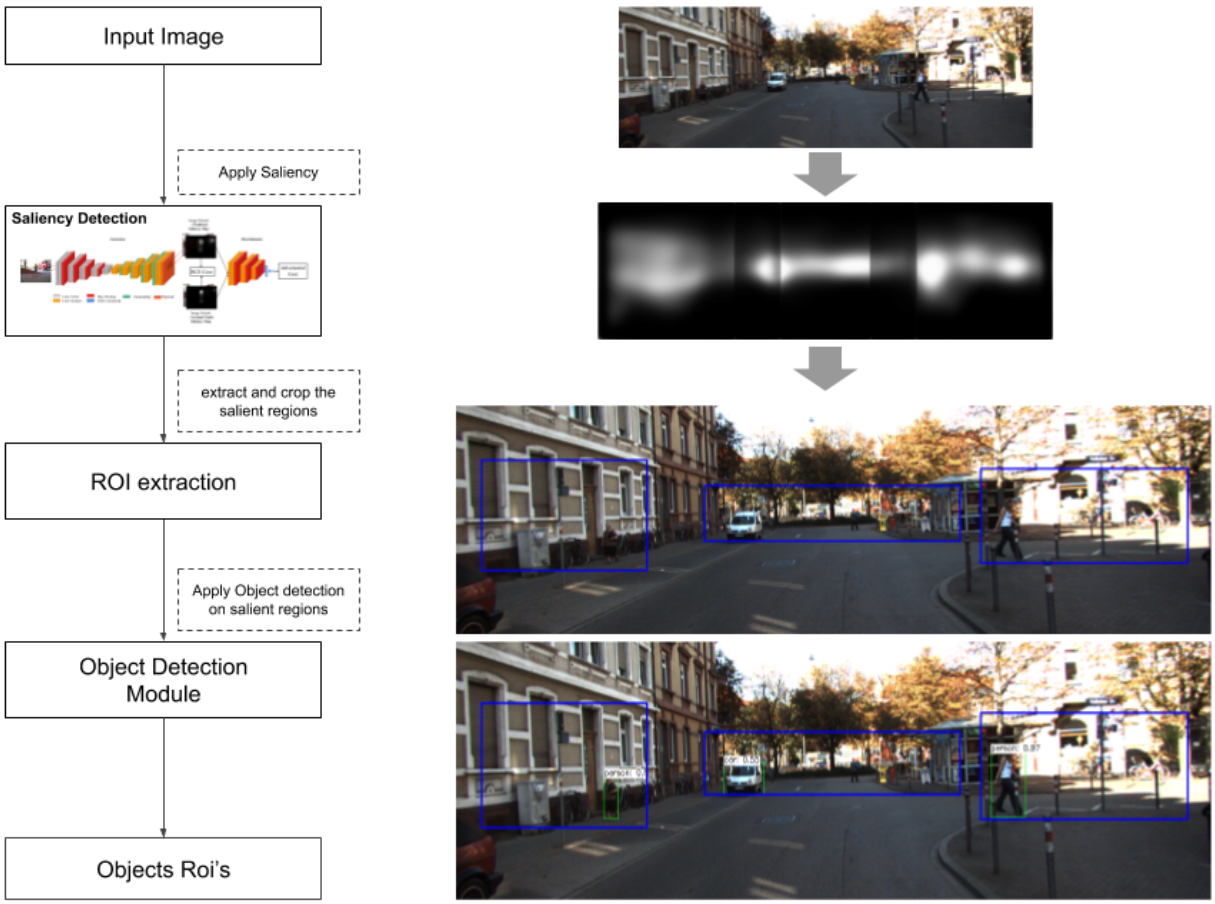


Fig. 2: Object annotation pipeline

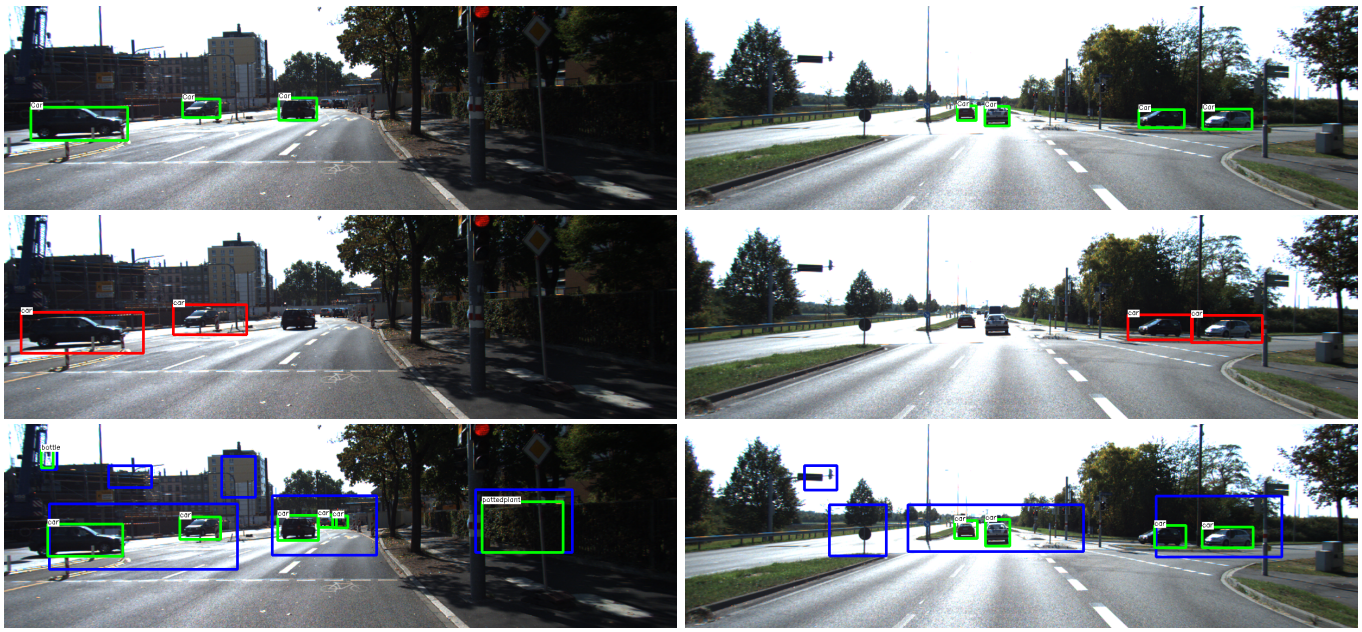


Fig. 3: Annotation of objects, 1^{st} row: ground truth images, 2^{nd} row: SSD detection results and 3^{rd} row: outputs of the proposed pipeline for automatic object annotation.

B. Results

Since we apply object detection only in the region where salient regions are identified by the SalGAN module, we computed the ratio of overall salient region to the area of the image for comparison of salient region occupancy in the image while applying object detection. This ratio indicated the percentage of images area process compared to the full image size. In the VOC2007 test data set, we employed the salient ROIs in two ways, one uses all extracted ROIs and applies object detection and the other is to consider the union of all extracted ROIs and applies object detection. The union of all extracted ROIs contains all the different bounding boxes thus some non salient regions might be included. In the case of union of salient ROIs, we obtained an average saliency map occupancy of 0.663 of the full image. The minimum area ratio is 0.015 and maximum is 0.995. In the case of the individual salient ROIs, we obtained an average saliency map occupancy of 0.374 , the minimum area ratio is 0.015 and maximum area ratio is 1.0.

Results on VOC2007 testing are reported in Table I. We report mAP of each class as well as all classes. The results achieved using saliency are compared against standard SSD. All saliency ROIs are all the salient region bounding boxes as per methods in the KITTI training set which is detailed in Section IV. In Table I, uSal is the Union of all the generated bounding boxes and iSal represents the individual salient regions.

Overall we use 66.3% of original image information and achieve comparable results with c.2% less than the SSD benchmark. We still observe that on some object classes such as *Aeroplane*, *Bird*, *Boat*, *Bottle*, *Cow* and *Sheep* we achieved better mAP with less image information. Using the union of all salient regions performs overwhelmingly better than combining all smaller salient regions, which indicates that objects in this dataset occupy a large percentage of image areas. We hypothesise that large relative object size makes saliency detection less effective than images with smaller object sizes such as KITTI. The results in Table II has confirmed the hypothesis, as we can see with the increasing of salient ROIs, the mAP is improving as well. We achieve better than SSD benchmark with area c.18% increase.

The first thing we noticed is that on the KITTI dataset, generated saliency regions are much smaller compared to the original ones in terms of area. To compare quantitatively we compute saliency ROIs to image area ratio. On average, saliency map are 0.29 of full image area. The maximum area ratio is 0.819 and minimum is 0.024. In other words, on average we are processing c.70% less data than the original image input.

Results on the KITTI training set are reported in Table III. With saliency guidance, the overall mAP of bounding boxes with all sizes increases c.12% compared with the one without saliency ROIs. If we break down by the class of objects, Person increased c.8% and Car increased c.16%. If we look at the result for objects that have an area more than 15×15 pixels, it is slightly better than all sizes. The results is aligned with all sizes. However if we look at objects with an area smaller than 75×75, the pipeline with saliency guidance greatly improve the SSD benchmark by almost 24%. The KITTI experiments demonstrate that off-the-shelf SSD seems to improve quite a lot with saliency added as guidance and saliency is specially useful for smaller objects.

	SSD	uSal	uSal +25%	iSal
Aeroplane	75.77	**81.21**	80.71	52.21
Bicycle	82.07	79.08	**82.87**	55.00
Bird	75.10	76.29	**76.83**	52.60
Boat	63.14	66.25	**68.22**	48.22
Bottle	35.15	**36.42**	36.09	25.34
Bus	88.71	86.35	87.74	61.03
Car	78.53	77.20	**79.84**	54.80
Cat	93.84	85.61	90.16	60.31
Chair	60.12	52.81	59.22	39.47
Cow	80.82	**81.40**	79.27	64.89
Diningtable	79.87	77.53	79.84	46.95
Dog	92.68	87.16	91.42	64.80
Horse	89.10	84.80	**89.20**	66.97
Motorbike	84.72	81.97	83.87	53.76
Person	66.75	60.09	64.90	42.33
Pottedplant	47.52	44.34	**47.60**	30.74
Sheep	69.58	72.14	**73.69**	62.41
Sofa	92.38	86.78	89.81	66.29
Train	89.80	88.75	89.79	59.00
Tvmonitor	78.06	76.73	**78.88**	55.81
mAP	76.19	74.15	**76.50**	53.15

TABLE I: Using SSD with Saliency on VOC2007 test dataset

% increase	0%	10%	15%	20%	25%	30%
mAP	74.15	75.87	76.18	76.38	76.50	76.35
% Sal-ROI	66%	77%	81%	84%	86%	88%

TABLE II: Variation of mAP with increase in salient ROI size

	All sizes		$>15^2$ px		$<75^2$ px	
	SSD	sal-SSD	SSD	sal-SSD	SSD	sal-SSD
Person	8.00	15.39	8.15	15.68	0.13	7.54
Cars	36.37	52.40	36.53	52.62	2.56	40.50
mAP	22.19	33.90	22.34	34.15	1.34	24.02

TABLE III: Results using SSD and with saliency on KITTI 2D object detection training set. Here px represents pixels

VI. Conclusions

In this paper we propose a annotation pipeline for data generated by instrumented vehicles. The proposed method incorporates pre-trained saliency map prediction and object detection algorithms. Popular dataset VOC2007 testing set and KITTI 2D object detection training set are used for experiments with pre-trained SalGAN and SSD. On VOC2007 we observe that, with saliency guidance, we only need c.66% of an image area to achieve almost the same mAP that the SSD benchmark achieves which uses all image information. This shows that human saliency could be exploited to discard redundant information for object detection.

In the VOC2007 experiments with saliency guidance using c.86% of image information, we slightly outperform the SSD benchmark. We hypothesise that this is because VOC2007 is a dataset for generic object detection purposes and objects in VOC2007 take a large percentage of the image areas. As objects occupy smaller portions of images in KITTI, saliency maps could better localize those objects and result in improved object detection and classification accuracy. This has been confirmed in KITTI experiments in which we have achieved c.13% better performance than the SSD benchmark on all objects sizes. On smaller objects namely object areas that are smaller than 75×75 we improve SSD benchmark almost 24% from 1.34%. This demonstrates human saliency is a helpful cue to facilitate object annotation of instrumented vehicle video, especially for smaller objects that standard benchmarks could miss.

Our future work includes fine-tuning the saliency map prediction algorithms that use methods of co-saliency in particular focusing on vehicle and/or pedestrians which are the dominant objects of interest in self-driving traffic scenarios.

ACKNOWLEDGMENT

This work has been sponsored by Huawei HIRP. The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289.

REFERENCES

[1] J. Deng, O. Russakovsky, J. Krause, M. S. Bernstein, A. Berg, and L. Fei-Fei, "Scalable multi-label annotation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2014, pp. 3099–3102.

[2] O. Russakovsky, L.-J. Li, and L. Fei-Fei, "Best of both worlds: human-machine collaboration for object annotation," in *Proc of the IEEE conference on CVPR*, 2015, pp. 2121–2131.

[3] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," *arXiv preprint arXiv:1501.02876*, 2015.

[4] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on PAMI*, vol. 37, no. 3, pp. 569–582, 2015.

[5] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *Proceedings of the IEEE ICCV*, 2013, pp. 2976–2983.

[6] Y. Qin, H. Lu, Y. Xu, and H. Wang, "Saliency detection via cellular automata," in *Proceedings of the IEEE Conference on CVPR*, 2015, pp. 110–119.

[7] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proceedings of the IEEE conference on CVPR*, 2014, pp. 2814–2821.

[8] Z. Wang, D. Xiang, S. Hou, and F. Wu, "Background-driven salient object detection," *IEEE Transactions on multimedia*, vol. 19, no. 4, pp. 750–762, 2017.

[9] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. Dagan Feng, "Robust saliency detection via regularized random walks ranking," in *Proceedings of the IEEE conference on CVPR*, 2015, pp. 2710–2717.

[10] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, and S. J. Maybank, "Salient object detection via structured matrix decomposition," *IEEE Transactions on PAMI*, vol. 39, no. 4, pp. 818–832, 2017.

[11] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Transactions on PAMI*, vol. 33, no. 2, pp. 353–367, 2011.

[12] G. Lee, Y.-W. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," in *Proceedings of the IEEE Conference on CVPR*, 2016, pp. 660–668.

[13] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Proceedings of the IEEE ICCV*, 2017, pp. 212–221.

[14] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," in *Proceedings of the IEEE Conference on CVPR, SUNw: Scene Understanding Workshop*, 2017.

[15] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, "Deepvs: A deep learning based video saliency prediction approach," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 602–617.

[16] L. Zhou, Z. Yang, Q. Yuan, Z. Zhou, and D. Hu, "Salient region detection via integrating diffusion-based compactness and local contrast," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3308–3320, 2015.

[17] 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 CVPR*, 2014, pp. 580–587.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on PAMI*, vol. 37, no. 9, pp. 1904–1916, 2015.

[19] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE ICCV*, 2015, pp. 1440–1448.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[21] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE ICCV*, 2017, pp. 2961–2969.

[23] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head r-cnn: In defense of two-stage object detector," *arXiv preprint arXiv:1711.07264*, 2017.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[25] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in neural information processing systems*, 2013, pp. 2553–2561.

[26] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[27] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*, 2016, pp. 21–37.

[29] A. Sorokin and D. Forsyth, "Utility data annotation with amazon mechanical turk," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2008, pp. 1–8.

[30] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 859–868.

[31] X. Ke, J. Zou, and Y. Niu, "End-to-end automatic image annotation based on deep cnn and multi-label data augmentation," *IEEE Transactions on Multimedia*, 2019.

[32] Y. Ma, Y. Liu, Q. Xie, and L. Li, "Cnn-feature based automatic image annotation method," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3767–3780, 2019.

[33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European conference on computer vision*, 2014, pp. 740–755.

[35] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *Proceedings of the IEEE conference on CVPR*, 2015, pp. 1072–1080.

[36] W. Wang and J. Shen, "Deep cropping via attention box prediction and aesthetics assessment," in *Proceedings of the IEEE ICCV*, 2017, pp. 2186–2194.

[37] F. Hu and A. F. Smeaton, "Image aesthetics and content in selecting memorable keyframes from lifelogs," in *International Conference on Multimedia Modeling*. Springer, 2018, pp. 608–619.

[38] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.