"Looking at the right stuff" - Guided semantic-gaze for autonomous driving: Supplementary Material

Anwesan Pal, Sayan Mondal, and Henrik I. Christensen

Additional material including code and videos are available here.

Appendix A: Derivation of β for F-score

For the purpose of saliency prediction in driving, False Negatives (FN) are more of a concern as compared to False Positives (FP). This is because it is probably still fine to detect a pedestrian, even if they are not crossing the road anytime soon. On the contrary, it is a much bigger cost to not detect a person crossing. Thus, we need to tune our metrics in order to penalize FN more in comparison to FP. As discussed in the paper, D_{KL} and CC already do that. Here, we provide the derivation of F-score in terms of its hyper-parameter β. We know that:

$$\text{Precison} = \frac{\text{True Positive } (TP)}{\text{True Positive } (TP) + \text{False Positive } (FP)} \tag{1}$$

and,

$$\text{Recall} = \frac{\text{True Positive } (TP)}{\text{True Positive } (TP) + \text{False Negative } (FN)} \tag{2}$$

Now, the F-score is given by:

$$\text{F-score } (\beta) = \frac{(1+\beta^2) * \text{Precision} * \text{Recall}}{\beta^2 * \text{Precision} + \text{Recall}} \tag{3}$$

Replacing for Precision and Recall from 1 and 2 respectively,

$$\text{F-score } (\beta) = \frac{(1+\beta^2) * \frac{TP}{TP+FP} * \frac{TP}{TP+FN}}{\beta^2 * \frac{TP}{TP+FP} + \frac{TP}{TP+FN}} \tag{4}$$

$$= \frac{(1+\beta^2) * TP}{\beta^2 * (TP+FN) + (TP+FP)} \tag{5}$$

$$= \frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + \beta^2 * FN + FP} \tag{6}$$

In equation 6, we clearly see that the numerator has no FP or FN terms and they are present only in the denominator in the additive form. Thus we conclude that with increase in FP or FN, F-score (β) value decreases. That is,

$$\text{F-score } (\beta) \downarrow = \frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + \beta^2 * FN \uparrow + FP} \tag{7}$$

$$\text{F-score } (\beta) \downarrow = \frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + \beta^2 * FN + FP \uparrow} \tag{8}$$

Also in equation 6, we see that FN has a weight of β^2 and FP has a weight of 1. Thus, when β^2 is lower than 1, FN has smaller influence on F-score (β) compared to FP , and when β^2 is greater than 1, FN has greater influence on F-score (β) compared to FP. As discussed in the paper and above, FN is more dangerous compared to FP for autonomous driving task and thus the value of β^2 must NOT be chosen lower than 1. We chose to give equal weightage to FN and FP by taking β^2 equals to 1.

Appendix B: Algorithm description

Here, we mention the hyperparameters in Table 1 and the architecture components in Table 2 of each of the four algorithms we considered. All the four models were trained on both the gaze-only data and our proposed SAGE data. The hyperparameters were kept the same during both the training process.

Parameters	**DR(eye)VE** [4]	**BDD-A** [5]	**ML-Net** [1]	**PiCANet** [2]
Input image size	448×448	576×1024	480×640	224×224
Initial Learning rate	0.0001	0.001	0.001	0.001
Learning rate decay	-	-	0.0005/step	0.1/7000 steps
Non-Linearity in feedforward network	ReLU, Leaky ReLU (α=0.001)	ReLU	ReLU	ReLU
Training Loss function	K-L Divergence	Cross-Entropy	K-L Divergence	Binary Cross-Entropy
Optimizer	Adam	Adam	SGD	SGD
Batchsize	8	10	8	4
#Training Epochs	20	20	32	20

Table 1: Summary of Hyperparameters

Algorithms	**Model Architecture**
DR(eye)VE [4]	COARSE: 6 layer 3D ConvNet (C3D architecture) with Bilinear Upsampling REFINE: 5 layer 2D ConvNet (for resized input), 1 layer 2D ConvNet (for cropped input)
BDD-A [5]	AlexNet feature extractor + Upsampling + 3 layer 2D ConvNet (visual processing) + Conv2D-LSTM (temporal processing)
ML-Net [1]	Feature Extraction Network: 13 layer Fully Convolutional Network (FCN) Encoder Network: 1 layer 2D ConvNet Decoder Network: Bilinear Upsampling
PiCANet [2]	Encoder Network: 16 layer FCN (VGG-16) Decoder Network: 6 layer Deconvolution with bilinear interpolation

Table 2: Network Architectures

Appendix C: Miscellaneous

In the main paper, in Figure 5, we showed a cross-evaluation where we considered two variants of SAGE and compared it with the respective gaze-only groundtruths. In Table 3, we show a similar result where we consider two variants of the gaze-only results along with that of SAGE. As seen from the results, SAGE outperforms the former in almost every case.

	Fixation-centric metrics						Semantic-centric metrics					
	D_{KL}			CC			F_1 score			MAE		
Dataset	DR(eye)VE gt	BDD-A gt	**SAGE gt**	DR(eye)VE gt	BDD-A gt	**SAGE gt**	DR(eye)VE gt	BDD-A gt	**SAGE gt**	DR(eye)VE gt	BDD-A gt	**SAGE gt**
DR(eye)VE	2.02±0.47	2.26±0.55	**1.67±0.41**	0.48±0.1	0.45±0.11	**0.55±0.11**	0.17±0.09	0.13±0.05	**0.36±0.09**	0.07±0.03	0.07±0.03	0.07±0.03
BDDA	1.74±0.43	1.28±0.43	**0.73±0.38**	0.42±0.14	0.58±0.13	**0.75±0.13**	0.09±0.06	0.1±0.06	**0.37±0.14**	0.12±0.06	0.11±0.06	**0.08±0.05**

Table 3: Comparison of SAGE with two variants of the gaze truth.

Real-time applicability of SAGE-Net - In Figure 1 we compare fps rate and F_1 score of the SAGE-trained algorithms with the DR(eye)VE multi-branch network [3], which included optical-flow and semantic segmentation branches, along with the raw image prediction.

Table 4 further shows a comparison of the number of trainable parameters between our approach and [3]. In addition to the computational inefficiency observed, optical flow, by itself, also does not provide information regarding a salient object's *absolute* velocity, which we think is vital for driving.

Added results on some real-world video data are shown in the attached video.

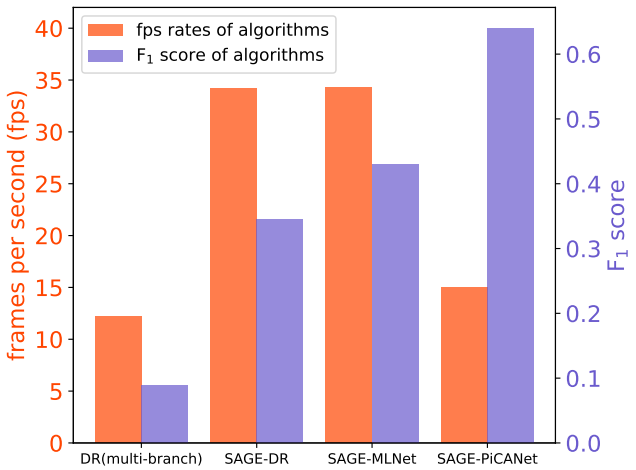

Figure 1: Comparison of fps rate and F_1 score.

	F_1 score	#params	$\frac{F_1 \text{ score}}{\#\text{params}} \times 10^9$
DR(multi-branch) [3]	0.09	40,578,441	2.22
SAGE-DR	0.34	13,515,395	25.16

Table 4: Comparison of F_1 score and #trainable parameters

References

[1] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR)*, 2016.

[2] Nian Liu, Junwei Han, and Ming-Hsuan Yang. PiCANet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3089–3098, 2018.

[3] Andrea Palazzi, Davide Abati, Francesco Solera, Rita Cucchiara, et al. Predicting the driver's focus of attention: the dr (eye) ve project. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1720–1733, 2018.

[4] Andrea Palazzi, Francesco Solera, Simone Calderara, Stefano Alletto, and Rita Cucchiara. Learning where to attend like a human driver. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 920–925. IEEE, 2017.

[5] Ye Xia, Danqing Zhang, Jinkyu Kim, Ken Nakayama, Karl Zipser, and David Whitney. Predicting driver attention in critical situations. In *Asian Conference on Computer Vision*, pages 658–674. Springer, 2018.