Supplementary Material: Class-Agnostic Segmentation Loss and Its Application to Salient Object Detection and Segmentation

Angira Sharma
University of Oxford
angira.sharma@cs.ox.ac.uk

Naeemullah Khan
University of Oxford
naeemullah.khan@eng.ox.ac.uk

Muhammad Mubashar
LUMS
21100158@lums.edu.pk

Ganesh Sundaramoorthi
KAUST
ganesh.sundaramoorthi@kaust.edu.sa

Philip Torr
University of Oxford
philip.torr@eng.ox.ac.uk

A. Gradient Calculation for CAS loss

Here we describe the calculation of gradient of the class-agnostic segmentation loss function with respect to the weights ω of the network:

The class-agnostic segmentation loss is defined as:

$$CAS = \sum_{i=1}^{N} \int_{r_i} \underbrace{\frac{\alpha||\mathbf{s}(x) - \hat{\mathbf{s}}(r_i)||_2^2}{|r_i|} dx}_{\text{Uniformer}} - \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} (1-\alpha) \underbrace{||\hat{\mathbf{s}}(r_i) - \hat{\mathbf{s}}(r_j)||_2^2}_{\text{Discriminator}} \tag{1}$$

where N is the number of regions in the ground truth mask; $r_1, ..., r_i, r_j, ..., r_N$ denotes the regions of the ground truth mask (a particular segment); $|r_i|$ denotes the number of pixels in the region r_i; $\mathbf{s} = \{s^1, ..., s^m, ..., s^M\}$ is a vector of output descriptor components (or softmax output) of the network; $m \in \{1, ..., M\}$ where M denotes the number of output (softmax) channels i.e., number of units in the last layer of the network; $\alpha \in [0, 1]$ is a scalar, a weighing hyper-parameter which assigns weight to each term; for a region r we have that, $\hat{\mathbf{s}}(r) = \{\hat{s}(r)^1, ..., \hat{s}^m(r), ..., \hat{s}(r)^M\}$ is a vector containing channel-wise mean of the descriptor values; where for a channel m, $\hat{s}^m(r) = \frac{1}{|r|} \int_r s^m(x)\, dx$. In our formulation $\hat{s}^m(r)$ acts as a proxy for class label.

We compute the derivative of the loss with respect to the weights ω of the neural network,

$$\nabla_\omega \sum_{i=1}^{N} \int_{r_i} \alpha \frac{||\mathbf{s}(x) - \hat{\mathbf{s}}(r_i)||_2^2}{|r_i|} dx - \nabla_\omega (1-\alpha) ||\hat{\mathbf{s}}(r_i) - \hat{\mathbf{s}}(r_j)||_2^2 \tag{2}$$

using Leibniz rule, we can interchange the order of integral and gradient, we get,

$$\sum_{i=1}^{N} \int_{r_i} \nabla_\omega \alpha \frac{||\mathbf{s}(x) - \hat{\mathbf{s}}(r_i)||_2^2}{|r_i|} dx - \nabla_\omega (1-\alpha) ||\hat{\mathbf{s}}(r_i) - \hat{\mathbf{s}}(r_j)||_2^2 \tag{3}$$

Next, we simply apply chain rule to get,

$$\nabla_{\boldsymbol{\omega}} CAS = \sum_{i=1}^{N} \int_{r_i} 2 \frac{\alpha(\mathbf{s}(x) - \hat{\mathbf{s}}(r_i))(\nabla_{\boldsymbol{\omega}} \mathbf{s}(x) - \nabla_{\boldsymbol{\omega}} \hat{\mathbf{s}}(r_i))}{|r_i|} dx - \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} 2(1-\alpha)(\hat{\mathbf{s}}(r_i) - \hat{\mathbf{s}}(r_j))(\nabla_{\boldsymbol{\omega}} \hat{\mathbf{s}}(r_i) - \nabla_{\boldsymbol{\omega}} \hat{\mathbf{s}}(r_j))$$

From Equation (1) we've, $\nabla_{\boldsymbol{\omega}} \hat{s}^m(r_i) = \frac{1}{|r_i|} \int_{r_i} \nabla_{\boldsymbol{\omega}} s^m(x) dx$, where we have $\nabla_{\boldsymbol{\omega}} s^m(x)$ from the backpropagation of the network as $s^m(x)$ is the softmax output of the network.

B. Detailed Results for Properties of CAS loss

B.1. Sparsity

To prove the property of sparsity of the network when using CAS loss, we show the application of loss function in binary segmentation case where we have two outputs from the softmax channel. If we fix the value of α, the problem simplifies to,

$$\min - ||\mathbf{a} - \mathbf{b}||_2^2 = \max \quad ||\mathbf{a} - \mathbf{b}||_2^2 \tag{4}$$

subject to,

$$\sum_{j=1}^{2} \mathbf{a}(j) = 1 \qquad \sum_{j=1}^{2} \mathbf{b}(j) = 1 \qquad \mathbf{a}(j) \geq 0 \qquad \mathbf{b}(j) \geq 0 \qquad \forall j \tag{5}$$

Here, for simplicity of notation we are using a and b to denote the average descriptors on foreground and background respectively. We show the profile for loss function in Figure 1. Notice that the minima for loss function exists at the the points where output descriptors are sparse and different.

Writing the Lagrangian for the system we get,

$$\begin{aligned} &L(a_0, a_1, b_0, b_1, \lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3, \mu_4) = (a_0 - b_0)^2 + \\ &(a_1 - b_1)^2 - \lambda_1(a_0 + a_1 - 1) \\ &- \lambda_2(b_0 + b_1 - 1) + \mu_1 a_0 + \mu_2 a_1 + \mu_3 b_0 \\ &+ \mu_4 b_1 \end{aligned} \tag{6}$$

where λ are the Lagrange multipliers for equality constraints and μ are the Lagrange multipliers for the inequality constraints.

$$\begin{aligned} \nabla_{a_0} L &= 2(a_0 - b_0) - \lambda_1 + \mu_1 = 0 \\ \nabla_{a_1} L &= 2(a_1 - b_1) - \lambda_1 + \mu_2 = 0 \\ \nabla_{b_0} L &= -2(a_0 - b_0) - \lambda_2 + \mu_3 = 0 \\ \nabla_{b_1} L &= -2(a_1 - b_1) - \lambda_2 + \mu_4 = 0 \\ a_0 + a_1 &= 1 \\ b_0 + b_1 &= 1 \\ \mu_1 \cdot a_0 &= 0 \\ \mu_2 \cdot a_1 &= 0 \\ \mu_3 \cdot b_0 &= 0 \\ \mu_4 \cdot b_1 &= 0 \end{aligned} \tag{7}$$

For a point to be maximum of the above constraint optimization it has to satisfy the KKT conditions. Here we write the KKT conditions and show that the sparse and different descriptors satisfy the KKT conditions. Sparse and different descriptors $a_0 = 1, a_1 = 0, b_0 = 0, b_1 = 1$, $\lambda_1 = \lambda_2 = 2, \mu_2 = \mu_3 = 4$ satisfy the KKT conditions. Similarly, we can show the same for the other sparse and different solution. The same can be extended to multiple dimension with the only condition being that the number of softmax channels should be greater than or equal to the number of classes.

The network using CAS loss has sparse solutions which can be visualised as sparse outputs as shown in Figure 2 and 3.

B.2. Class-Imbalance

The empirical results which show robustness to class imbalance are shown in Fig. 5 To test the accuracy of the class-agnostic segmentation loss in tackling class imbalance, we test on an artificially generated toy example. This allows us to specifically focus on the class agnostic property of the loss function while eliminating other factors. We generate data for two classes in 2D where class one is centred around point $(1, 0)$ and class two is centred around point $(0, 1)$. The samples for both classes have random Gaussian noise added to each component and hence are scattered around the class centres with variance value of 0.2. The two classes are also highly unbalanced with class 1 having 10000 data points and class 2 having only 10 points (see Figure 4).

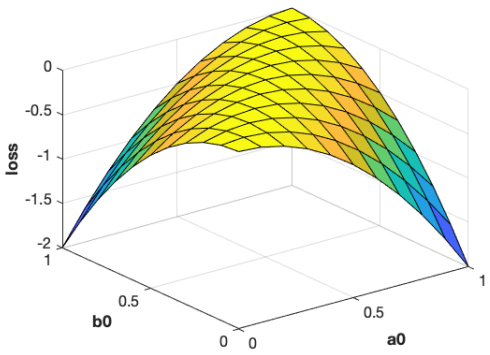

Figure 1. Profile of Loss Function

We generate 2 sets of data for training and testing respectively. We train a 2 layer fully connected networks with 10 hidden units and test on testing data. The results are summarized in Table 1. The CE loss fails to perform in this case where data is highly unbalanced and assigns all output labels to belong to class 1. On the other hand, CAS loss is immune to this class imbalance and performs better.

B.3. Boundedness

Figure 6 shows the training loss for training with 3000 images. Notice that the loss in the Figure 6 is well bounded within $(\alpha N_i, -(1-\alpha)N_i]$.

C. Low-fidelity data setting explanation

An example of what the low-fidelity data looks like is shown in Figure 7, where 50% of the data was normal and half was flipped.

D. Dataset-wise models' Results for salient object detection

In Table 2 the quantitative results of all our models trained on different datasets and tested on 7 saliency datasets are shown, along with comparison with state-of-the-art methods.

As seen in Table 2, all the models, even the state-of-the-art ones, perform better when tested on the datasets belonging to the training set domain. This is the anticipated issue of dataset bias [8], which is a shortcoming of the breadth of the various datasets.

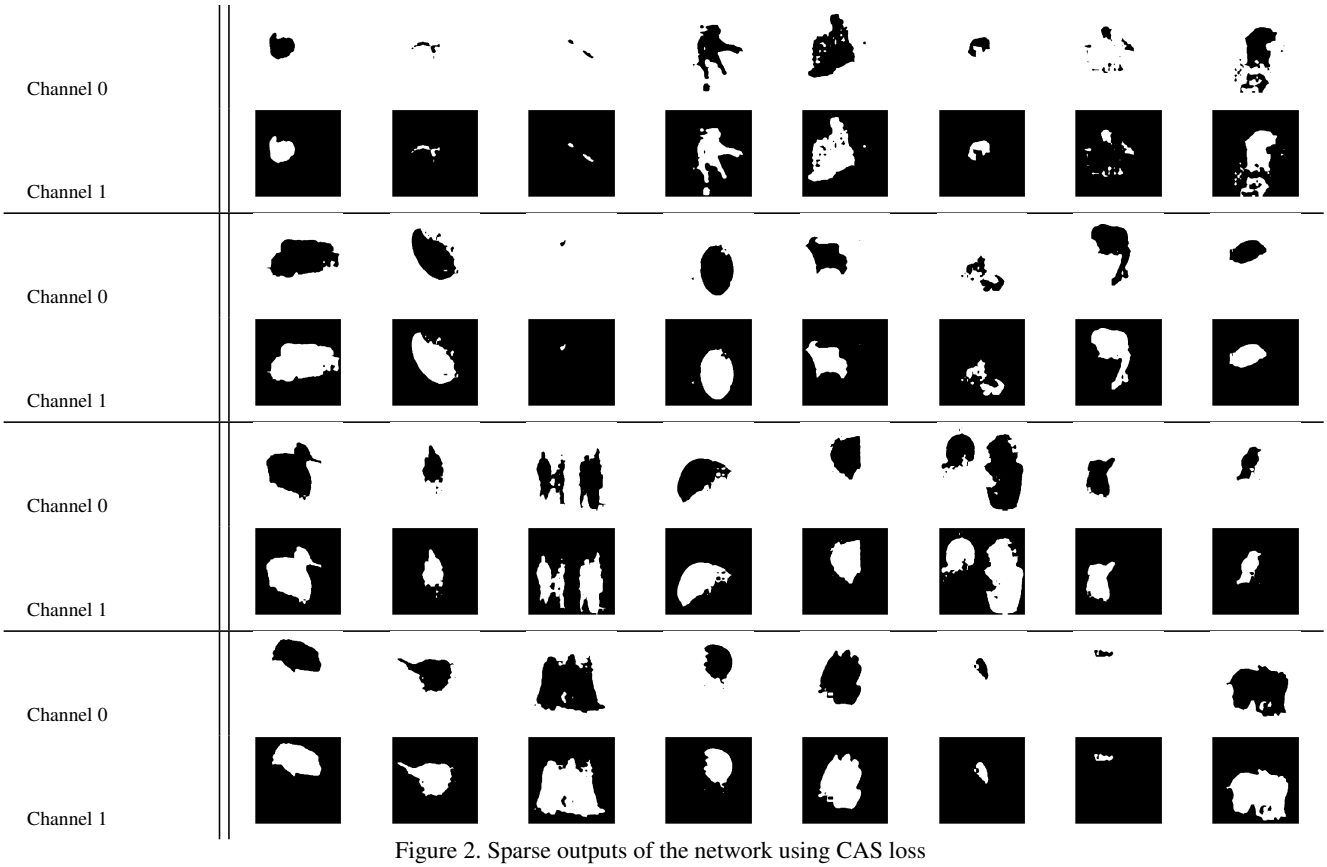

Figure 2. Sparse outputs of the network using CAS loss

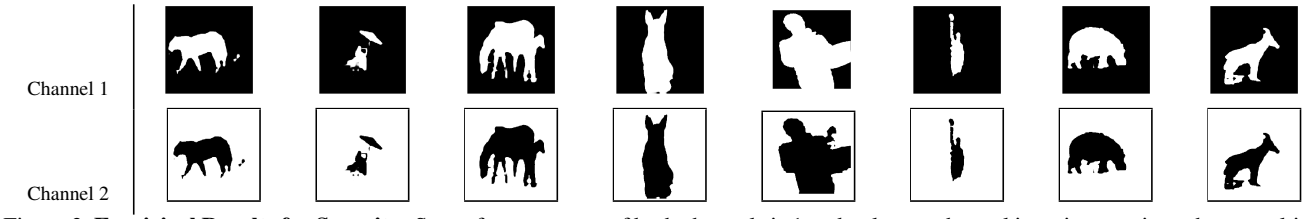

Figure 3. **Empirical Results for Sparsity**: Sum of components of both channels is 1 and only one channel is active at a time, thus, resulting in sparse output descriptors

Table 1. CE and CAS comparison

CE Results

output \ label	class 1	class 2
class 1	10000	10
class 2	0	0

CAS Results

output \ label	class 1	class 2
class 1	9899	0
class 2	101	10

E. Pre and Post-processing for Multi Object Segmentation

We have performed multi object segmentation on BSDS500 and Pascal VOC2012 datasets. The numbers of segments (objects) in each image in unknown apriori. In the pre-processing step, we resize all images to 256×256 and normalize them to zero mean and unit variance. For post processing we cluster the descritpors (outpouts of DNN) in 20 regions and then smooth out the regions with less then 2% of the total pixels in the image using conditional random fields (we used pydensecrf library [1]).

[1] https://github.com/lucasb-eyer/pydensecrf

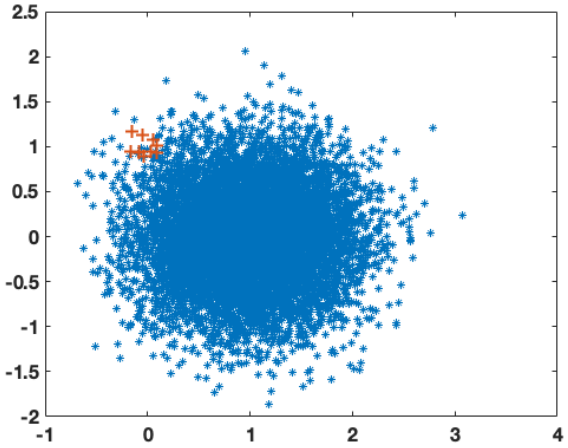

Figure 4. Data Sample

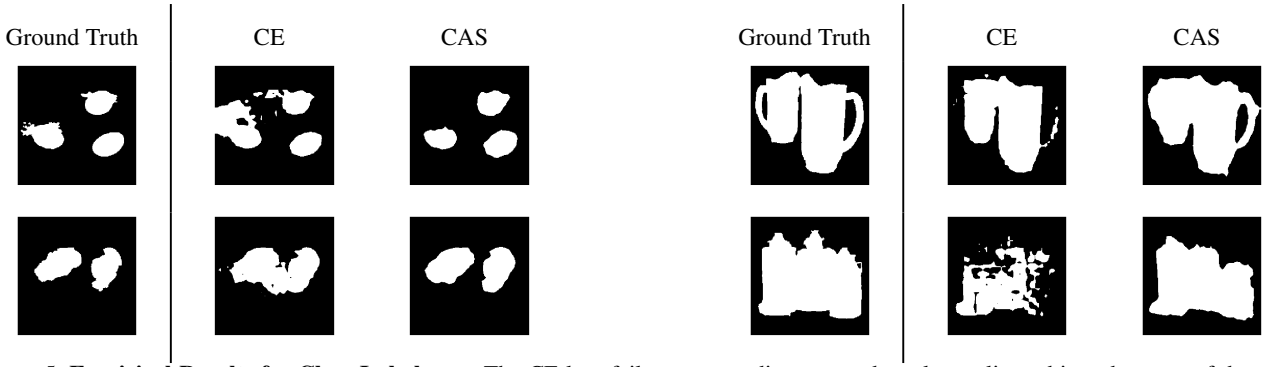

Figure 5. **Empirical Results for Class Imbalance**: The CE loss fails to output disconnected or clear salient objects because of the size bias of salient objects, whereas, the CAS loss is robust to such size bias or class imbalance

F. Visual Results for Multi Object Segmentation

The visual results for multi object segmentation are shown in Figure 10 for BSDS500 dataset and in Figure 11 for PASCALVOC2012 dataset.

G. Visual Results for DeepLab model for Salient Object Detection and Texture Segmentation

The visual results for DeepLab-CE model i.e., DeepLab-v3 architecture with cross-entropy loss function and DeepLab-CAS model i.e., DeepLab-v3 architecture with class-agnostic segmentation loss are shown in Figure 8 for salient object detection and in Figure 9 for texture segmentation.

All these results concur with those performed on FCN-ResNet-101 architecture. These verify our claims empirically, about the working of our CAS loss function with any neural network. Also, the performance of the CAS loss is comparable and majority of times better than the cross-entropy loss function, for both the tasks of salient object detection and segmentation.

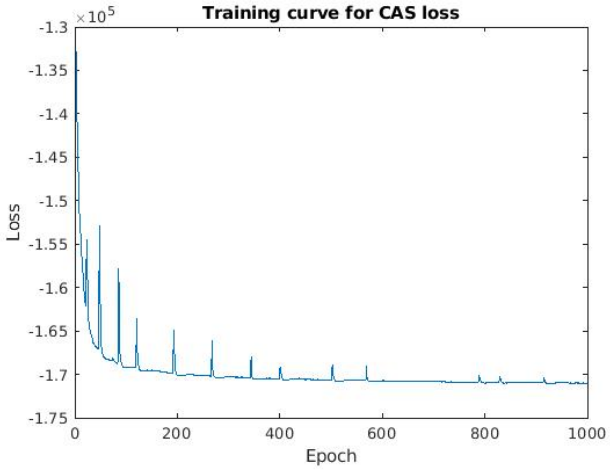

Figure 6. Training curve for CAS loss

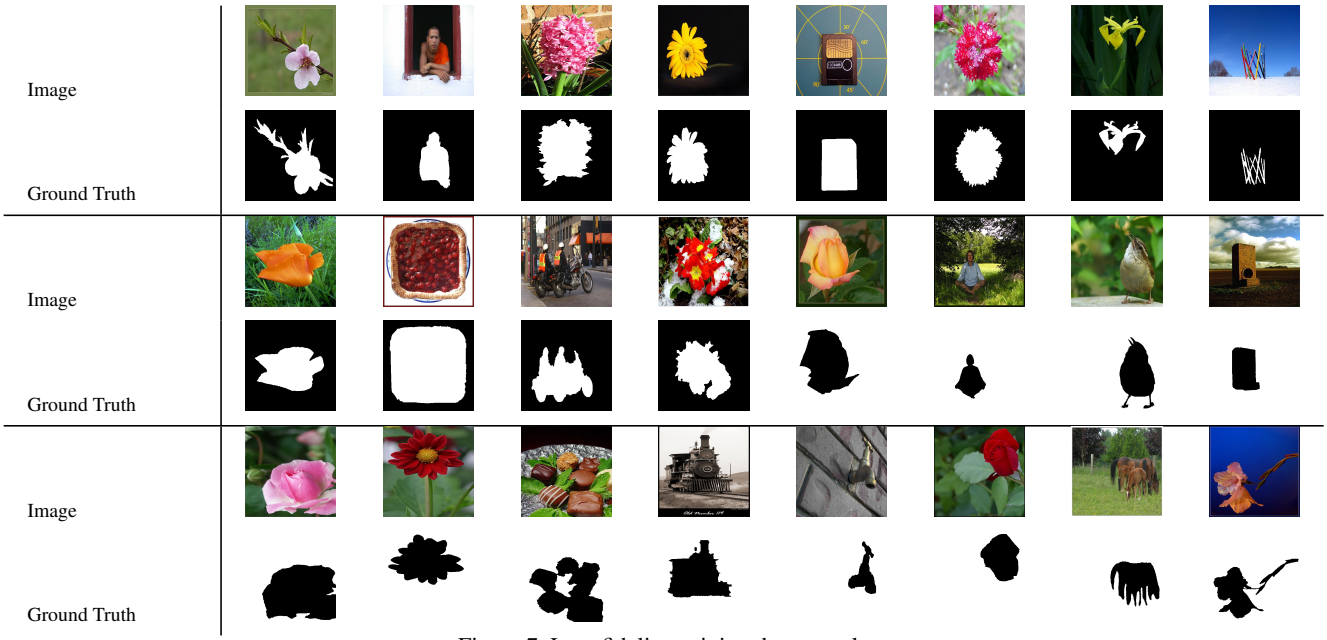

Figure 7. Low-fidelity training data sample

Model	Training Set	MSRA-B		DUTS-TE		ECSSD		PASCAL-S		HKU-IS		THUR15k		DUT-OMRON	
		$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$
ResNet-pre-CAS (ours)	MSRA-B	0.985	0.010	0.836	0.088	0.874	0.071	0.818	0.112	0.887	0.056	0.891	0.075	0.876	0.066
ResNet-m-CE (ours)	MSRA-B	0.958	0.030	0.910	0.067	0.905	0.068	0.868	0.100	0.921	0.053	0.928	0.065	0.920	0.059
ResNet-m-CAS (ours)	MSRA-B	0.944	0.037	0.853	0.091	0.876	0.079	0.811	0.124	0.931	0.057	0.930	0.075	0.863	0.080
ResNet-d-CE (ours)	DUTS-TR	0.947	0.0.65	0.919	0.055	0.867	0.077	0.876	0.091	0.928	0.044	0.935	0.057	0.867	0.062
ResNet-d-CAS (ours)	DUTS-TR	0.932	0.046	0.871	0.071	0.888	0.075	0.840	0.121	0.939	0.050	0.931	0.073	0.875	0.071
DeepLab-CAS (ours)	DUTS-TR	0.931	0.040	0.850	0.070	0.864	0.072	0.800	0.111	0.882	0.054	0.888	0.069	0.865	0.060
DeepLab-CE (ours)	DUTS-TR	0.928	0.039	0.847	0.070	0.867	0.069	0.805	0.110	0.880	0.052	0.881	0.070	0.856	0.061
BAS-Net [5]	DUTS-TR	-	-	0.860	0.047	0.942	0.037	0.854	0.076	0.921	0.039	-	-	0.805	0.056
PoolNet [3]	MSRA-B + HKU-IS	-	-	0.892	0.036	0.945	0.038	0.880	0.065	0.935	0.030	-	-	0.833	0.053
CPSNet [10]	COCO+DUT	-	-	-	-	0.878	0.096	0.790	0.134	-	-	-	-	0.718	0.114
PFAN [11]	DUTS-TR	-	-	0.870	0.040	0.931	0.032	0.892	0.067	0.926	0.032	-	-	0.855	0.041
PAGENET+CRF[9]	THUS10k	-	-	0.817	0.047	0.926	0.035	0.835	0.074	0.920	0.030	-	-	0.770	0.063
PAGENET[9]	THUS10k	-	-	0.815	0.051	0.924	0.042	0.835	0.078	0.918	0.037	-	-	0.770	0.066
HED [2]	MSRA-B	0.927	0.028	-	-	0.915	0.052	0.830	0.080	0.913	0.039	-	-	0.764	0.070
DNA [4]	DUTS-TR	-	-	0.873	0.040	0.938	0.040	-	-	0.934	0.029	0.796	0.068	0.805	0.056

-pre- represents model pre-trained using cross-entropy loss and then trained using CAS loss; -m- represents the model trained on MSRA-B dataset; -d- represents the model trained on DUTS-TR dataset ;

red represents the best score value on the dataset; blue represents the second best score on the dataset; - represents the dataset was not tested by the method

Table 2. Numerical Results on High-Fidelity Data for all the models trained on different datasets

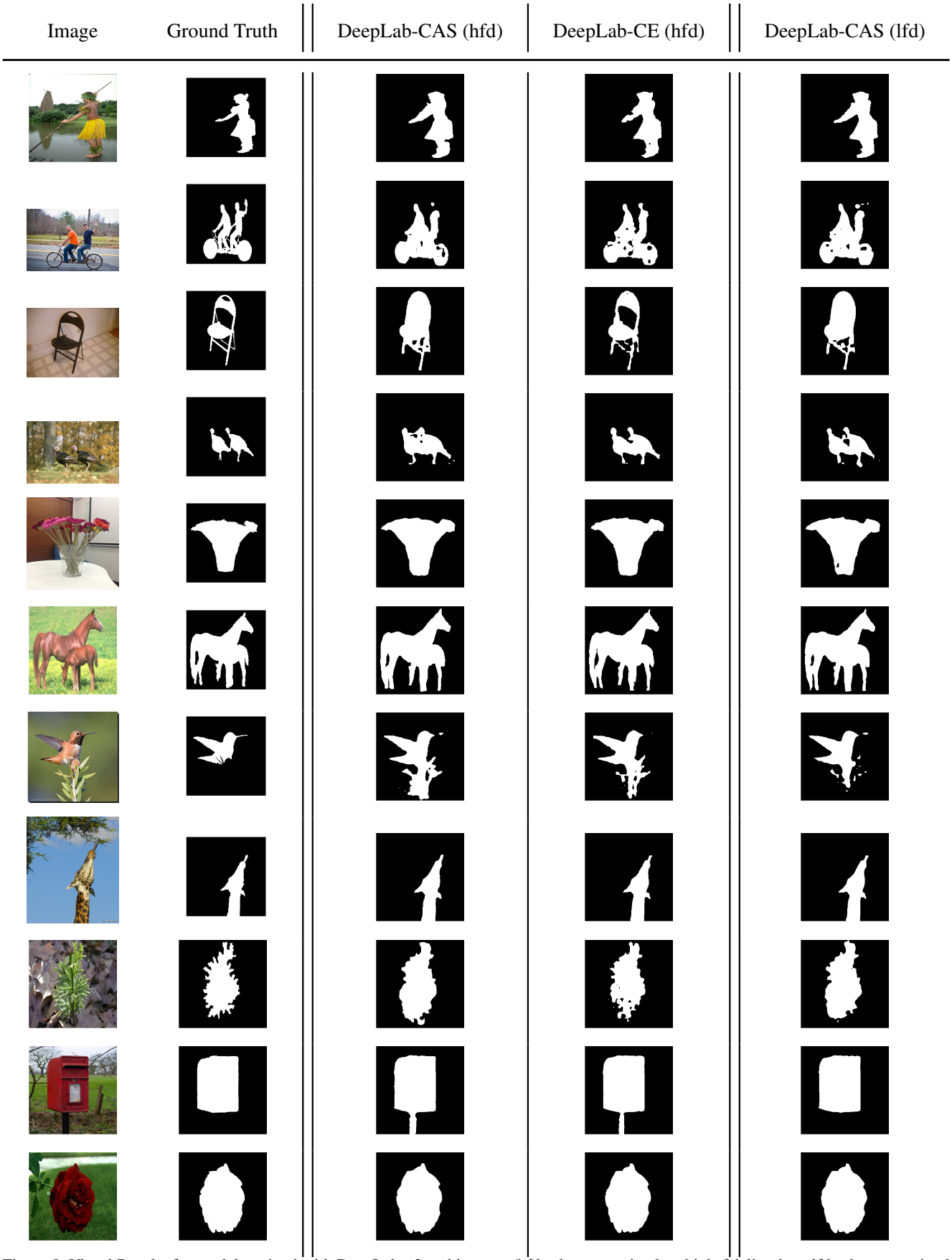

Figure 8. Visual Results for models trained with DeepLab-v3 architecture (hfd - denotes trained on high-fidelity data; lfd - denotes trained on low-fidelity data)

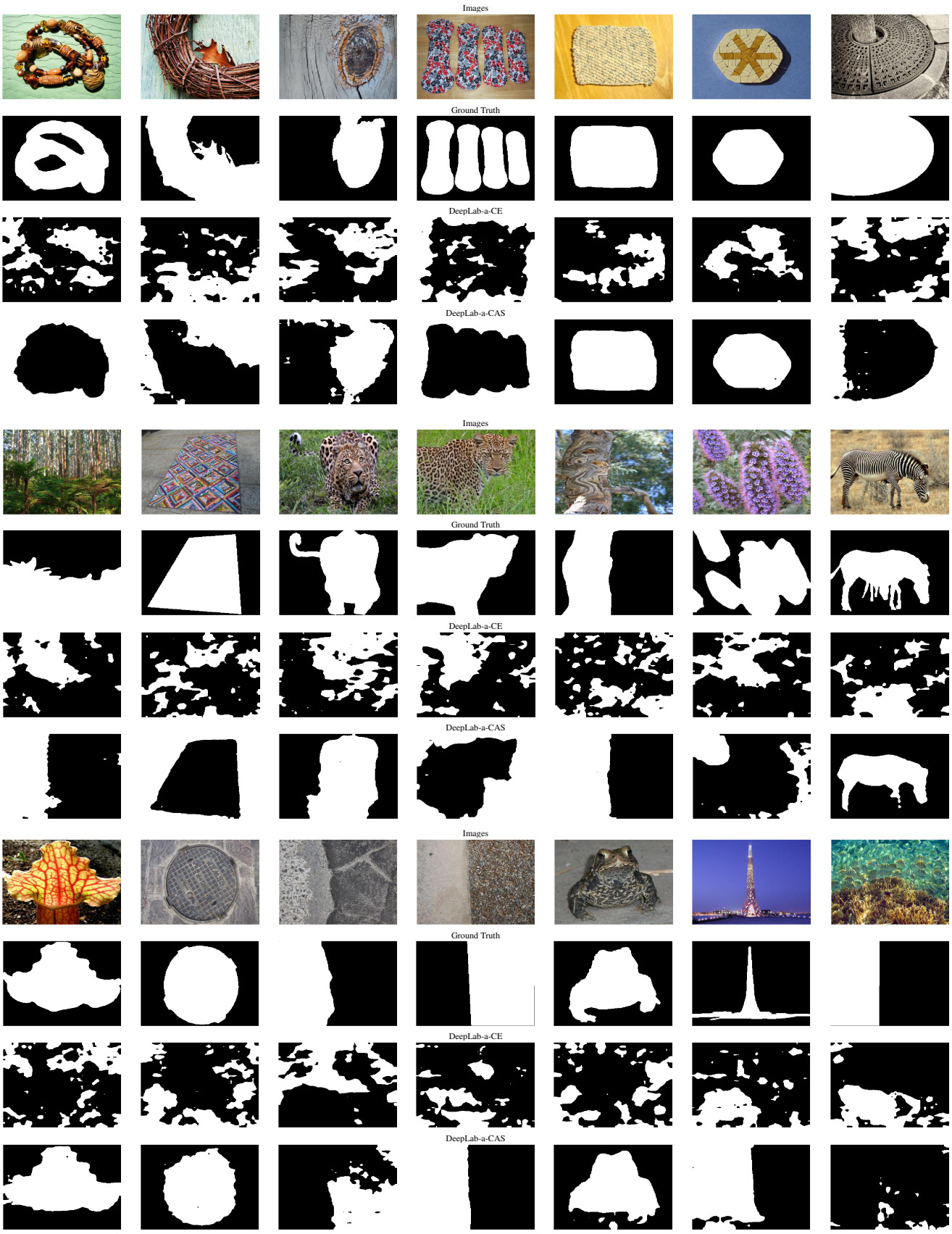

Figure 9. **Sample representative results on Real-World Texture Dataset**: Visual results for texture segmentation experiments, using DeepLab architecture; -a- denotes trained on the 7 saliency datasets and texture dataset

Figure 10. Visual Results for BSDS500

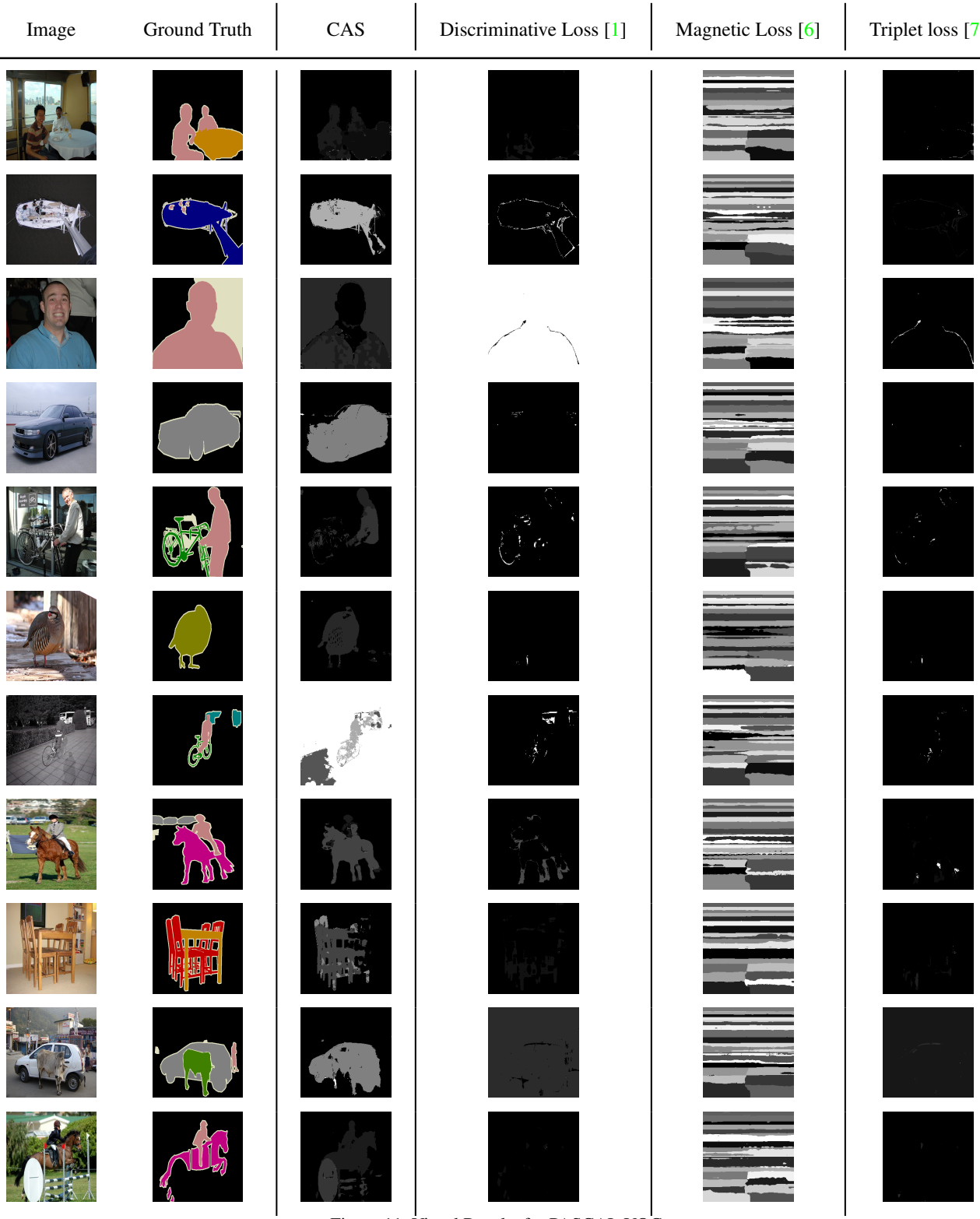

Figure 11. Visual Results for PASCAL VOC

References

[1] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic Instance Segmentation with a Discriminative Loss Function. Technical report, 2017. 9, 10

[2] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections, 2017. 6

[3] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A Simple Pooling-Based Design for Real-Time Salient Object Detection. In *CVPR*, 2019. 6

[4] Yun Liu, Deng-Ping Fan, Guang-Yu Nie, Xinyu Zhang, Vahan Petrosyan, and Ming-Ming Cheng. DNA: Deeply-supervised Nonlinear Aggregation for Salient Object Detection. 2019. 6

[5] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. BASNet: Boundary-Aware Salient Object Detection. In *CVPR*, 2019. 6

[6] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric Learning with Adaptive Density Discrimination. 2016. arXiv: 1511.05939. 9, 10

[7] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, volume 07-12-June, pages 815–823, 2015. 9, 10

[8] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011. 2

[9] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven C H Hoi, and Ali Borji. Salient Object Detection with Pyramid Attention and Salient Edges. *CVPR*, 1(c):1448–1457, 2019. 6

[10] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, Lihe Zhang, Mingyang Qian, and Yizhou Yu. Multi-source weak supervision for saliency detection. *CVPR*, 2019. 6

[11] Ting Zhao and Xiangqian Wu. Pyramid Feature Attention Network for Saliency detection. *CVPR*, 2019. 6