Supplementary material

This supplementary document summarizes the following experimental results:

- Detailed description of the synthetic test set (section 1).
- Ablation study of the proposed system (section 2).
- Evaluation on real testing downloaded from the internet (section 3).
- Additional visual comparisons (section 4).

1. Test Set Details

Our synthetic test set is composed of 12,000 images synthesized with clean natural images and graphics patterns drastically different from training set. We used 4 types of patterns: stickers, lines, text, and logos. For each type of pattern, we synthesize test images at 3 different size levels: large, medium, and small. Sizes are defined slightly different across different patterns as it is very difficult to stipulate a common meaningful definition of effective size taken by a pattern for all 4 patterns (e.g. number of pixels overtaken by the rendered pattern in an image is a good measure of size for stickers and logos but not for text, since to achieve same amount of area as stickers/logos, a text box would occupy much larger part of image). During test set synthesis, we do not perform random attribute alignment between graphics patterns and image. We list the exact definition of size for each category and statistics used during test set synthesis for each category and size level below. Example images are as shown in figure 1

Stickers/logos The *effective size* of stickers/logos pattern is defined as the ratio between number of pixels overwritten by patterns in an image and the total number of pixels of that image. During synthesis, we render a random number of patterns onto image such that the total effective area is within range for that size level.

- **Small**: size ˜ Uniform(0.001, 0.016), number of patterns $\in [1, 2]$
- **Medium**: size ˜ Uniform(0.016, 0.064), number of patterns $\in [1, 4]$
- **Large**: size ˜ Uniform(0.064, 0.4), number of patterns $\in [1, 12]$

Lines The *effective size* of lines is defined as the ratio between width of a line and the length of shorter side of an image. Note although we use term line, the pattern rendered includes both line segment as well as free-form curves.

- **Small**: size ˜ Uniform(0.008, 0.02), number of patterns $\in [1, 10]$
- **Medium**: size ˜ Uniform(0.02, 0.06), number of patterns $\in [1, 10]$
- **Large**: size ˜ Uniform(0.06, 0.15), number of patterns $\in [1, 6]$

Text The *effective size* of text is dictated by both the size of a glyph (roughly, width of a single character relative to image width) and the total area of bounding box of text in the image since we could have a very long string of small font text that occupies entire image or a very large single character. We used following number during synthesis

- **Small**: glyph size ˜ Uniform(0.05, 0.1), bounding box size $\in [0.002, 0.016]$
- **Medium**: glyph size ˜ Uniform(0.1, 0.2), bounding box size $\in [0.016, 0.25]$
- **Large**: glyph size ˜ Uniform(0.15, 0.4), bounding box size $\in [0.25, 0.6]$

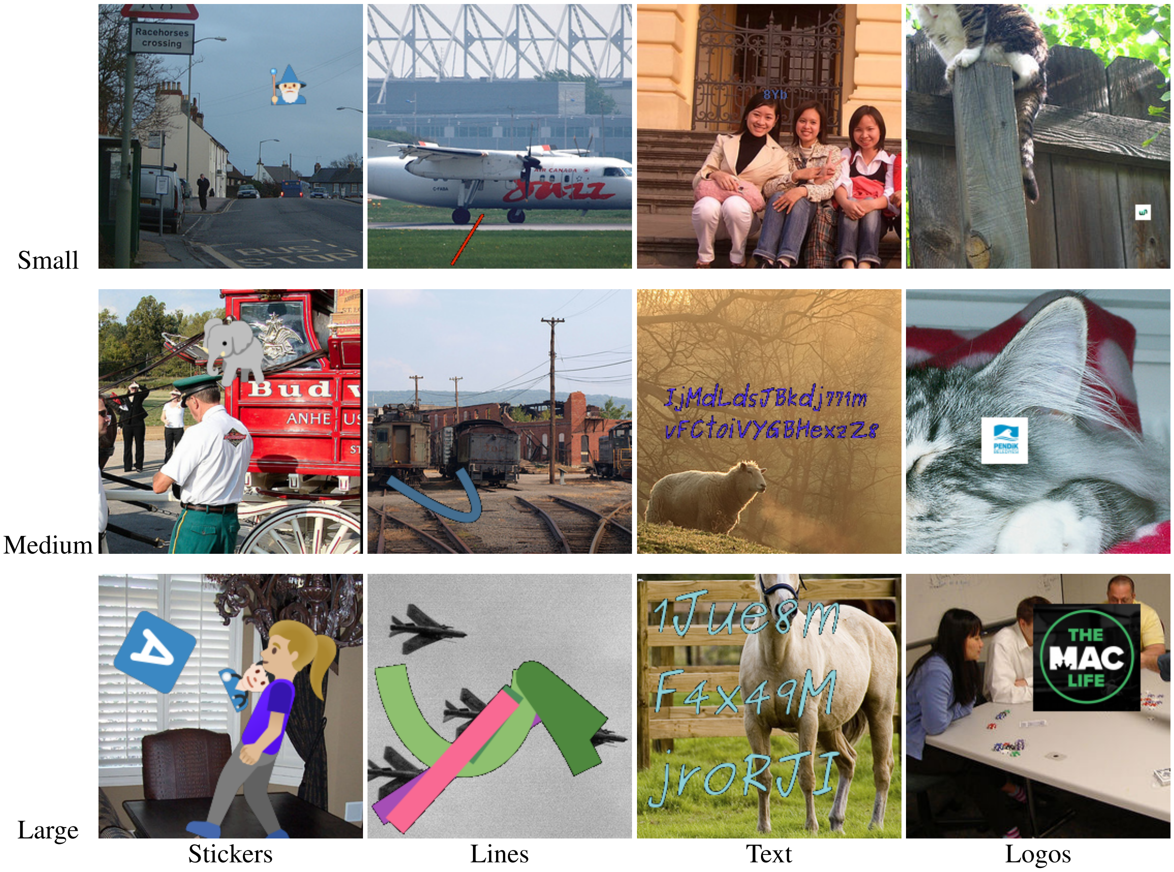


Figure 1. Example test images from each category and size level

2. Ablation Study of Proposed Solution

We perform ablation study on each element of our proposed solution and show the effectiveness of each proposed element in boosting the performance.

2.1. Proposed Attribute Randomization

For data synthesis, we compare our random graphics pattern attribute blending scheme with 1) a simple and straightforward attribute random perturbation, 2) no attribute perturbation. In the simple perturbation scheme, we randomly adjust pattern attributes (e.g. make pattern brighter/darker, less/more saturated, etc.) irrespective of the corresponding local/global attribute at the target location in the image

	Overall Performance on Test Set			
	mIoU	MAE	$F_{0.3}$	F_2
Proposed	**0.745**	**0.022**	**0.873**	**0.863**
Simple	0.667	0.031	0.843	0.793
None	0.606	0.042	0.799	0.743

Table 1. Quantitative comparison of our proposed attribute randomization versus a simple randomization scheme and no randomization

2.2. Impact of JPEG Compression

Albeit already discussed in other scenarios (e.g. Photoshop face manipulation detection by Wang et al. [5]), models trained for our problem heavily relies on low-level image features, and therefore we emphasize the necessity of applying JPEG compression to training data for achieving reasonable generalization to real online images. To this end, we show in

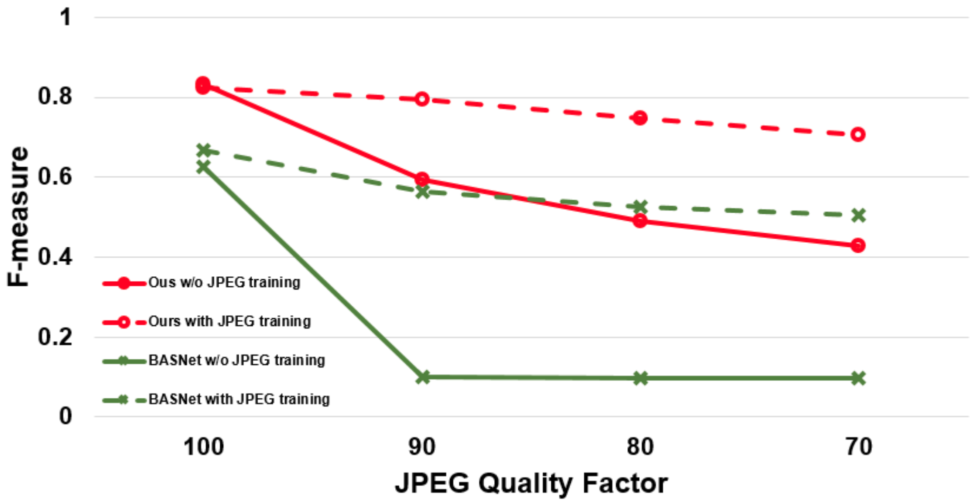


Figure 2. $F_{0.3}$ score as a function of the JPEG compression quality factor. A lower quality factor means less quantization levels on DCT coefficient and therefore stronger blocking artifacts and poorer image quality. Only performance degradation of our proposed model and BASNet is shown for figure clarity, but all models are heavily influenced.

figure 2 the performance (F-measure) of different models, trained with/without JPEG compression, on test set under different JPEG compression settings. As can be seen, without JPEG compression during training, model performance is severely impaired by blocking artifacts introduced by JPEG compression even at quality level 90.

2.3. Network Architecture

We investigate the effectiveness of cascading scheme by training variants of our multi-scale cascade network with 1-level (i.e. no cascade), multi-scale input 2/3/4-level cascade, and single scale input 3-level cascade (SS 3-level). For fair comparison, we keep total size (number of parameters) of feature extractors of each network roughly the same. Specifically, we used 12-resblocks for 1-level network feature extractor, 6-resblocks for each of two feature extractors of 2-level network, 4-resblocks for each of three feature extractors of 3-level network, and 3-resblocks for each of four feature extractors of 4-level network. As shown in table 2, although 1-level network demonstrates on-par/slightly better performance on large patterns, our usage of cascading scheme improves consistency of performance across different pattern sizes. The 3-level cascade network that only uses single scale input (input at original resolution of 256x256) achieves similar performance as compared to its multi-scale input counterpart. but it should be noted that: 1) multi-scale version still has better consistency across patterns sizes, 2) single scale version generates very large feature maps at all hidden layers and the computation budget required is much higher at both training and inference time. The extra computation budget limits the possibility of using a larger backbone at each cascade level or cascading more levels.

	Performance on Test Set							
	Small		Medium		Large		Gap between Large and Small	
	mIoU	$F_{0.3}$	mIoU	$F_{0.3}$	mIoU	$F_{0.3}$	$\lvert\Delta mIoU\rvert$	$\lvert\Delta F_{0.3}\rvert$
1-level	0.545	0.695	0.735	0.856	0.809	0.923	0.262	0.228
2-level	0.582	0.732	0.790	0.910	0.801	0.920	0.219	0.188
3-level	**0.664**	**0.809**	**0.796**	**0.913**	0.782	0.902	0.118	0.093
4-level	0.616	0.782	0.757	0.893	0.724	0.869	**0.108**	**0.087**
SS 3-level	0.645	0.788	0.781	0.906	**0.820**	**0.926**	0.175	0.138

Table 2. Quantitative comparison of models with different number of cascade levels. Note that 3-level model performs best on small and medium size patterns and has smallest performance gap between performance on large and small pattern

2.4. Multi-stage training

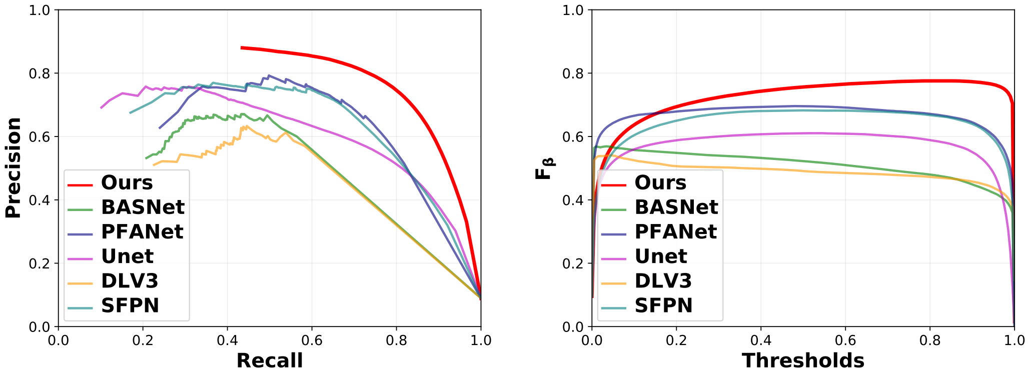
Deep supervision has been demonstrated to be effective in training segmentation networks and is widely adopted by numerous works [2, 3, 4, 6, 1]. Our multi-stage training can be perceived as a special form of deep supervision. Therefore, we compare our training scheme with training entire multi-scale cascade network jointly with supervision on each sub-network. Table 3 shows that our multi-stage training scheme enables our cascade network to achieve better performance.

	Overall Performance on Test Set			
	mIoU	MAE	$F_{0.3}$	F_2
Multi-stage	**0.745**	**0.022**	**0.873**	**0.863**
Joint	0.565	0.047	0.755	0.722

Table 3. Quantitative comparison of multi-stage training versus simultaneous joint training

3. Wild Data Test

We compare performance of our proposed model and competing models on 100 images collected from popular social media website using same metric as in the main text. Collected images are manually labeled. As shown in figure 3 (a, b) and table 4, our proposed model displays best performance on these wild images. Some visual comparison examples are shown in figure 4. Note that the overall performance is poorer as compared with synthetic test set because some images are heavily corrupted by blur and noise.



(a) Precision-Recall Curve

(b) F-measure vs. Thresholds

Figure 3. Precision-recall and F-measure curve of each competing methods evaluated on images collected from popular social media websites. The models here are the same as the ones in the main result section of the paper (i.e. no extra training or additional data used)

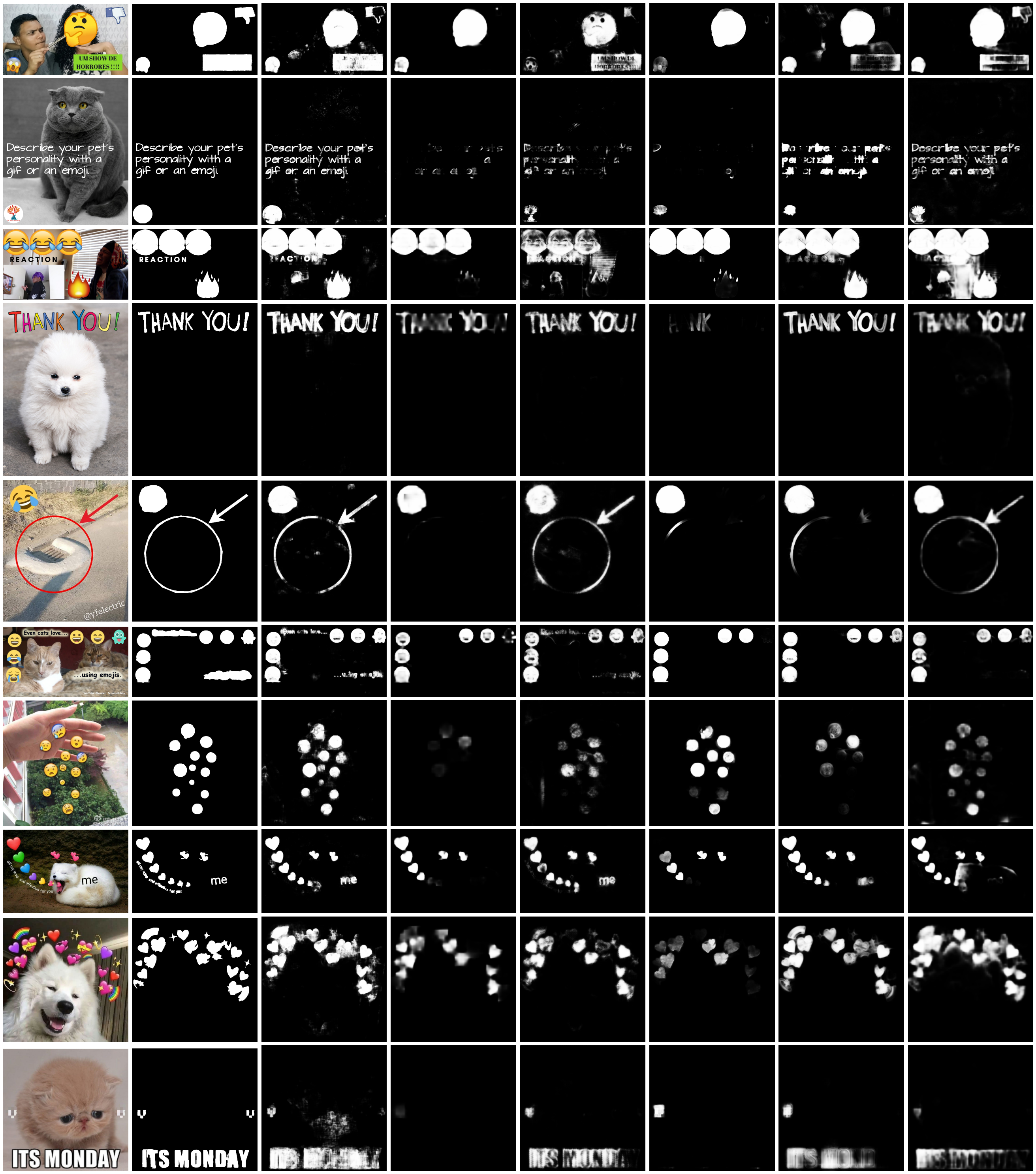
	Overall Performance on Test Set			
	mIoU	MAE	$F_{0.3}$	F_2
Ours	**0.631**	**0.044**	**0.776**	**0.793**
Unet	0.445	0.071	0.610	0.685
DLV3	0.410	0.052	0.541	0.546
BASNet	0.410	0.063	0.569	0.541
PFANet	0.541	**0.044**	0.696	0.707
SFPN	0.540	0.050	0.683	0.700

Table 4. Quantitative comparison of competing methods on data from wild

4. Additional Visual Comparisons

Additional results of the visual comparisons are shown in Figure 5, Figure 7, Figure 6, and Figure 8. For each category, we show the perturbations of different sizes, and compare the performance with the competing methods.

Real images downloaded from the internet



(a) Image (b) GT (c) Ours (d) DLV3 (e) UNet (f) BASNet (g) PFANet (h) SFPN

Figure 4. Visual comparison of the proposed method and the competing methods on wild data collected from popular social media websites

More Test Set Visual Comparisons (1): Stickers

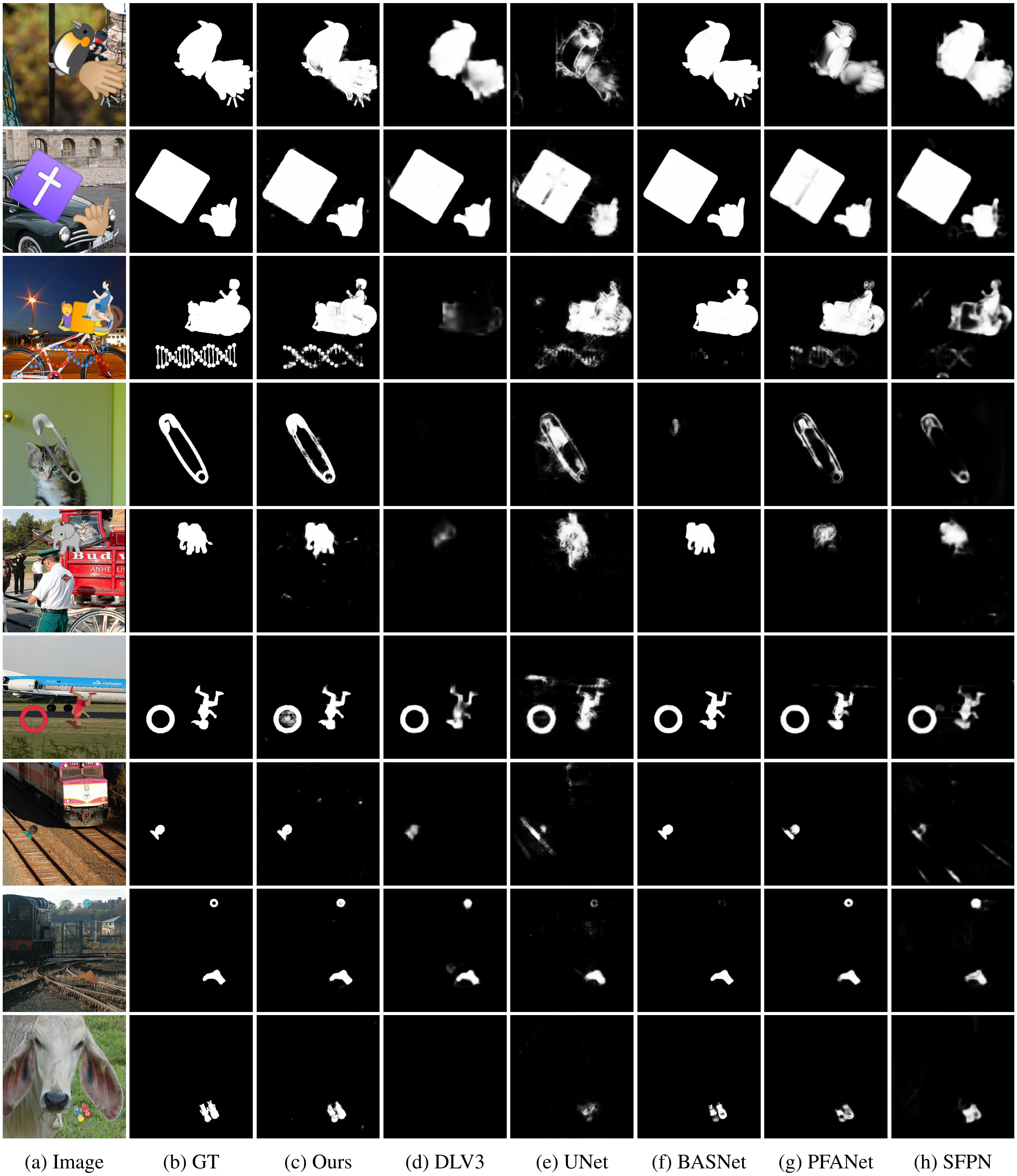


Figure 5. Visual comparison of the proposed method and the competing methods on stickers

More Test Set Visual Comparisons (2): Lines

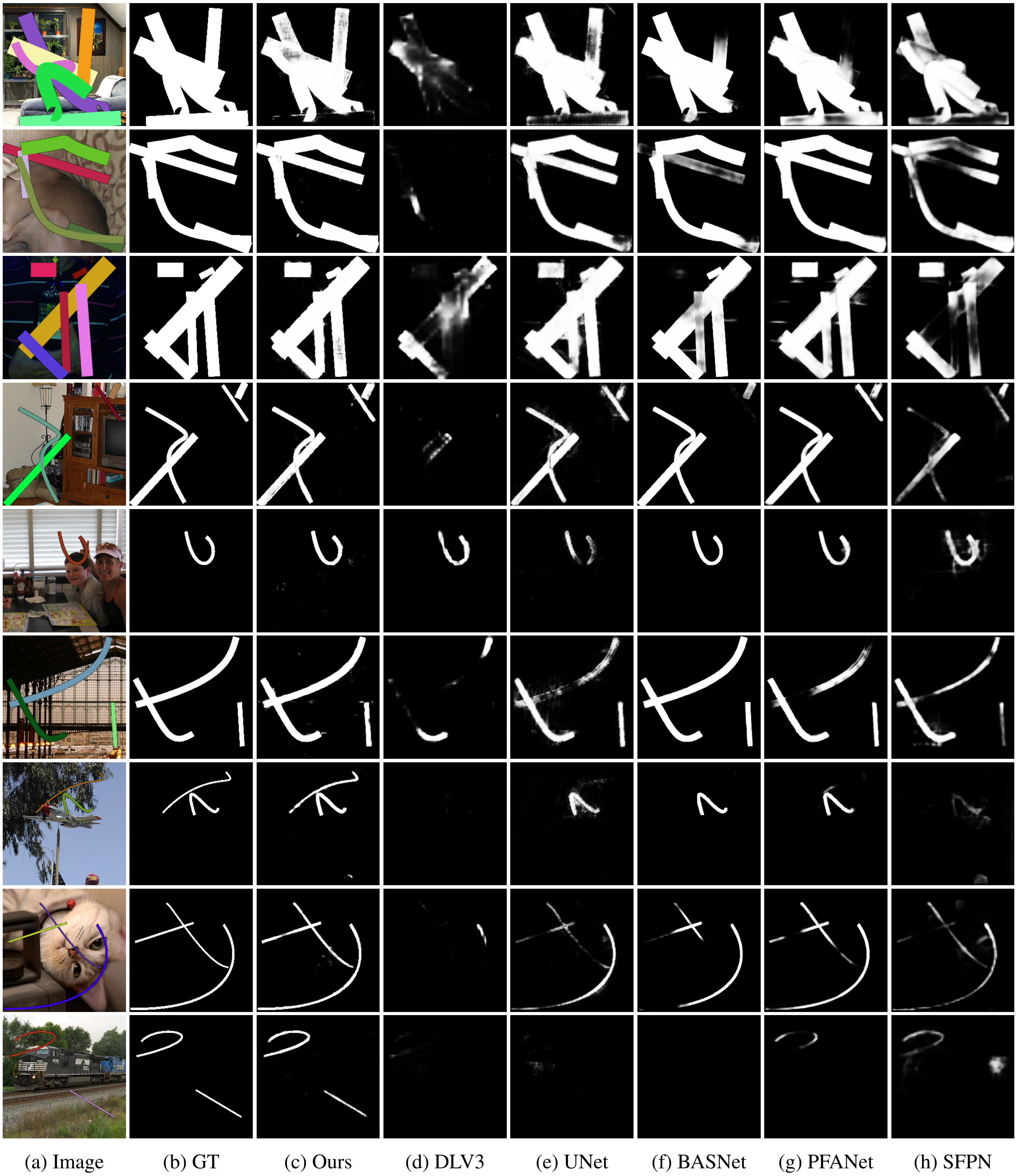
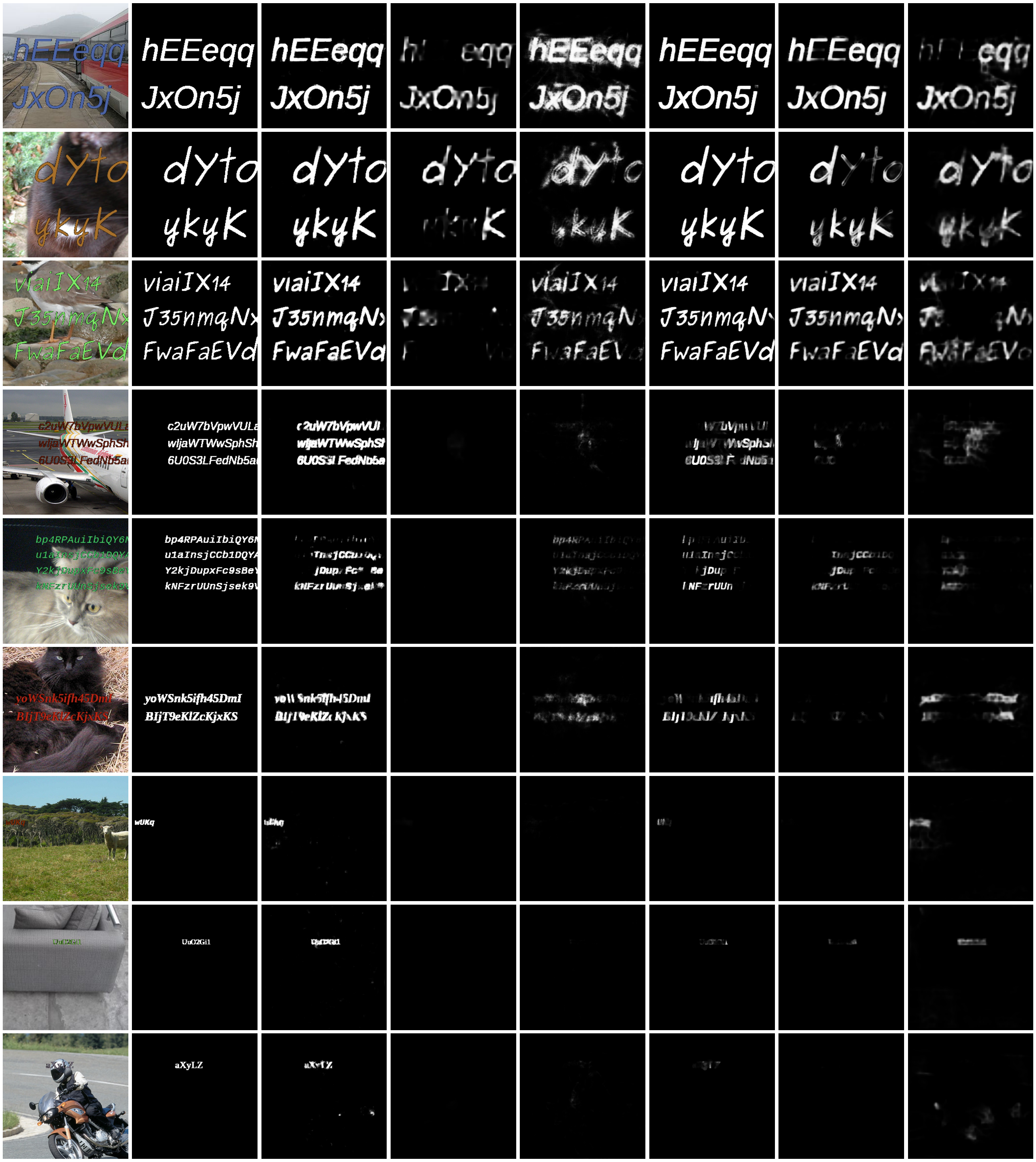


Figure 6. Visual comparison of the proposed method and the competing methods on lines

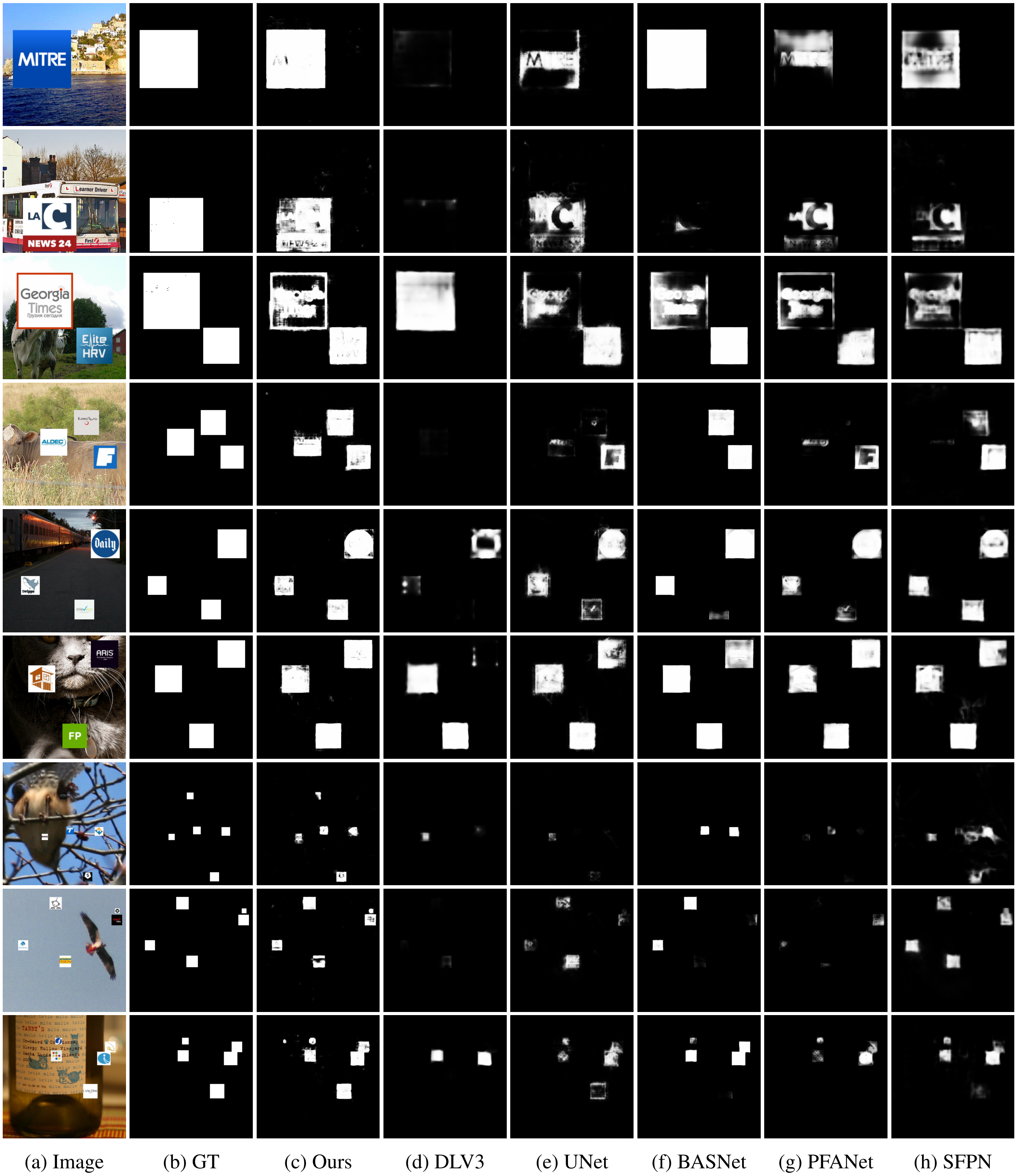
More Test Set Visual Comparisons (3): Text



(a) Image (b) GT (c) Ours (d) DLV3 (e) UNet (f) BASNet (g) PFANet (h) SFPN

Figure 7. Visual comparison of the proposed method and the competing methods on text

More Test Set Visual Comparisons (4): Logo



(a) Image (b) GT (c) Ours (d) DLV3 (e) UNet (f) BASNet (g) PFANet (h) SFPN

Figure 8. Visual comparison of the proposed method and the competing methods on logos

References

[1] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *CVPR*, 2019. 3

[2] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(4):815–828, 2019. 3

[3] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. BASNet: Boundary-aware salient object detection. In *CVPR*, 2019. 3

[4] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Salient object detection with recurrent fully convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(7):1734–1746, 2019. 3

[5] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, and Alexei A Efros. Detecting photoshopped faces by scripting photoshop. In *ICCV*, 2019. 2

[6] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, 2017. 3