Exploiting saliency for object segmentation from image level labels
Supplementary material

1. Content

This document contains the following additional details:

- GAP
  - Architecture details for `GAP-LowRes`, `GAP-HighRes`, `GAP-DeepLab`, `GAP-ROI`.
  - Training details.
  - Qualitative results.
- CRF experiments and CRF parameters used.
- More qualitative results of our saliency model.
- Details of $\mathcal{G}_2$ guide labeller rules, and more qualitative examples of $\mathcal{G}_0$, $\mathcal{G}_1$, and $\mathcal{G}_2$ strategies.
- Convnet training details for Seeder, Classifier, and Segmenter networks.
- Additional qualitative examples like figure 7 in the main paper.

2. GAP

2.1. Network details

See table 3 for the details of the GAP networks used. For `GAP-ROI`, we insert the GAP layer *after* the final linear layer, instead of after the penultimate layer as suggested by [5]. We note that the resulting functions are identical: GAP is a linear sum over the spatial dimensions, and the final layer performs a linear combination over the channel dimensions, so they can be swapped without changing the function. Also in practice, we find that there is only negligible difference in performance between the variants. *We suggest for the future practitioners using GAP to include the pooling layer after the final layer at training time, and extract heatmaps from the final layer at test time.*

2.2. Training GAP

All GAP network variants are trained with stochastic gradient descent (SGD) with minibatch size 15, momentum 0.9, weight decay 5×10^{-4}, and base learning rate 0.001, decreased by the factor of 10 at every 2 000 iterations. The training stops at 8 000 iterations.

2.3. Qualitative examples

See figure 1 for qualitative examples. We observe that `GAP-LowRes`, `GAP-HighRes`, and `GAP-ROI` show qualitatively similar results, while `GAP-DeepLab` has significantly low quality with repeating patterns in the output. The output suggests that the learned filters have repeating patterns modulo ≈ 12 output pixels, which is the width of the dilated filters in DeepLab-LargeFOV [1] on `conv5` features.

3. CRF

See table 1 for the segmenter performance after applying different combinations of CRF units, `crf-seed`, `crf-loss`, and `crf-postproc`. Combination of `crf-loss` and `crf-postproc` on the `GAP-HighRes` seed gives 50.4 mIoU, giving 12.9 mIoU boost over the vanilla seed. However, we do not see such a gain when either the CRF parameters or the seed type is changed. When CRF parameters are changed from v_1 to v_2, both of which are reasonable choices (see §3.1), we lose 5.2 mIoU. When the seed type is changed from `GAP-HighRes` to `GAP-ROI`, we lose 5.4 mIoU. The 12.9 mIoU boost thus seems fragile.

Our saliency-based model, on the other hand, gives a consistent ≥ 4 mIoU gain over the best CRF combination, regardless of the seed type used, showing superiority over CRF both in terms of performance and stability. It is possible to combine `crf-loss` and saliency, but our preliminary experiments show that it hurts the performance of the saliency-only case. Thus, `crf-loss` is excluded from our final model. See table 2 for all the combinations considered in our experiments.

3.1. CRF parameters

Throughout the main paper, we use the CRF parameters from the DeepLab-LargeFOV model [1], unless stated otherwise. The parameters are given by $w^{(1)} = 4$, $\theta_\alpha = 121$, and $\theta_\beta = 5$ for the appearance kernel, and $w^{(2)} = 3$ and $\theta_\gamma = 3$ for the smoothness kernel, following the notation of equation 3 in [3]. We always use the Potts model $\mu(x_i, x_j) = 1_{x_i = x_j}$ for compatibility function.

For some experiments, we also use parameters from [2],

Table 1. Results of the CRF variants on Pascal 2012 validation. v_1, v_2: CRF parameters from [2] and [1] respectively.

Seed method	`crf`			val. set	
	`-seed`	`-loss`	`-postproc`	mIoU	ΔmIoU
`GAP-HighRes`	✗	✗	✗	37.5	-12.9
	✗	$\checkmark_{v1}$	$\checkmark_{v1}$	50.4	0
	✗	$\checkmark_{v2}$	$\checkmark_{v2}$	45.2	-5.2
		Saliency: $\mathcal{G}_2$		55.2	$+4.8$
`GAP-ROI`	✗	✗	✗	37.6	-12.8
	✗	$\checkmark_{v1}$	$\checkmark_{v1}$	45.0	-5.4
		Saliency: $\mathcal{G}_2$		54.6	$+4.2$

Table 2. Extension of table 1 showing all the combinations considered.

Seed method	`crf`			val. set	
	`-seed`	`-loss`	`-postproc`	mIoU	ΔmIoU
`GAP-HighRes`	✗	✗	✗	37.5	-12.9
	✗	$\checkmark_{v1}$	$\checkmark_{v1}$	50.4	0
	✗	$\checkmark_{v1}$	✗	46.4	-4.0
	✗	✗	$\checkmark_{v1}$	45.5	-4.9
	✗	$\checkmark_{v2}$	$\checkmark_{v2}$	45.2	-5.2
	$\checkmark_{v1}$	✗	✗	33.0	-17.4
		Saliency: $\mathcal{G}_2$		55.2	$+4.8$
`GAP-ROI`	✗	✗	✗	37.6	-12.8
	✗	$\checkmark_{v1}$	$\checkmark_{v1}$	45.0	-5.4
	✗	$\checkmark_{v2}$	$\checkmark_{v2}$	44.2	-6.2
		Saliency: $\mathcal{G}_2$		54.6	$+4.2$

which uses $w^{(1)} = 10$, $\theta_\alpha = 80$, and $\theta_\beta = 13$ for the appearance kernel, and $w^{(2)} = 3$ and $\theta_\gamma = 3$ for the smoothness kernel.

4. Saliency

See figure 2 for more examples of the MSRA training samples for our weakly supervised saliency model. Samples corresponding to Pascal categories are excluded from the training.

See figure 3 for qualitative examples of our saliency model on the Pascal images. We observe that the saliency model does fail in examples usually when the central salient object is not Pascal category, or when the scene is cluttered.

5. $\mathcal{G}_2$ guide labeller algorithm

We introduce details of the algorithm for $\mathcal{G}_2$ strategy of combining seed and saliency signals (§5.3 of the main paper). As mentioned in the main paper, we follow five simple ideas:

1. We treat seeds as reliable small size point predictors of each object instance.
2. We assume the saliency might trigger on objects that are not part of the classes of interest.
3. If a seed touches a connected component R_i^{fg},it should take the label of the seed.
4. If two (or more) seeds touch the same foreground component, then we want to propagate all the seed labels inside it.
5. When in doubt, mark as ignore.

The detailed procedure is given as follows.

We compute the set of connected components of the saliency foreground mask with area $\geq 1\%$ of the image size, $\{R_i^{fg}\}_i$, and similarly for the set of connected components of the seeds, $\{R_j^s\}_j$. For each R_i^{fg}, we assign a ground truth label on it depending on how many foreground seed categories it intersects with:

- 0 category: R_i^{fg} is then either a false positive from the saliency (e.g. salient object that is not part of the classes of interest), or a false negative from the seeds. We don't commit to any of those cases by marking with "ignore" label.

- 1 category: R_i^{fg} is delineating the full extent of the instance for the seed. Put the class label from the seed.
- $\geq$ 2 categories: R_i^{fg} is a combination of instances from multiple classes. Use dense CRF inference inside R_i^{fg}, with unaries set by the seed(s), to assign precise pixel-wise labels in R_i^{fg}.

After assigning pixel-wise labels on each R_i^{fg}, we perform the following operations regarding the seed connected components R_j^s:

- When a seed R_j^s intersects with some R_i^{fg}, but is not strictly covered by R_i^{fg}, we put "ignore" labels on the seed region bleeding out of R_i^{fg}, assuming that the saliency mask provides a better delineation of the object.
- If a seed R_j^s touches two or more foreground regions, it will propagate its label to all of them.
- Whenever there is an isolated seed R_j^s not intersecting with any R_i^{fg}, we treat it as a reliable foreground prediction missed by saliency, and include it in the final guide labelling.

See figure 4 for the qualitative examples of guide labelling strategies, $\mathcal{G}_0$, $\mathcal{G}_1$, and $\mathcal{G}_2$. Note that $\mathcal{G}_2$ produces much more precise labelling with the access to rich localisation information from the seeds. We will publish the code.

6. Convnet training details

Saliency. The network is `DeepLab-v2` ResNet, and follows the training procedure for `DeepLab-v2` ResNet in [1].

Segmenter. The network is `DeepLab-v1`, and is trained with stochastic gradient descent (SGD) with minibatch size 15, momentum 0.9, weight decay 5×10^{-4}, and base learning rate 0.001, decreased by the factor of 10 at every $2\,000$ iterations. The training stops at $8\,000$ iterations.

Classifiers. All classifiers discussed in the paper are `VGG-16` trained with stochastic gradient descent (SGD) with minibatch size 40, momentum 0.9, weight decay 5×10^{-4}, and base learning rate 0.001, decreased by the factor of 10 at every $5\,000$ iterations. The training stops at $30\,000$ iterations.

7. Qualitative examples

See figure 5 and 6 for more qualitative examples of the seeds, saliency, $\mathcal{G}_2$ guide labeller output, and Guided Segmentation trained results on the training set. Seeds have high precision and low recall. The saliency foreground mask gives pixel-wise class-agnostic object extent information. $\mathcal{G}_2$ guide labeller combines both sources to generate an accurate class-wise guide labelling. The generated guide labelling can still be noisy especially if the saliency quality is low. However, the segmenter convnet averages out the noisy supervision to produce more precise predictions. CRF post-processing further refines the predictions.

References

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 1, 2, 3, 4

[2] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision (ECCV)*. Springer, 2016. 1, 2, 4

[3] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*. 2011. 1

[4] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[5] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016. 1, 4

Layers	VGG-16 [4]	GAP-LowRes [5]	GAP-HighRes [2]	GAP-ROI	GAP-DeepLab [1]
input (C, H, W)	$(3, 224, 224)$	$(3, 321, 321)$	$(3, 321, 321)$	$(3, 321, 321)$	$(3, 321, 321)$
$2 \times$ conv1	$(64, 3, 3)$ pad $= 1$	Same as VGG-16	Same as VGG-16	Same as VGG-16	Same as VGG-16
pool1	(-, $2, 2$) pad $= 0$	Same as VGG-16	Same as VGG-16	Same as VGG-16	(-, $3, 3$) st $= 2$, pad $= 1$
$2 \times$ conv2	$(128, 3, 3)$ pad $= 1$	Same as VGG-16	Same as VGG-16	Same as VGG-16	Same as VGG-16
pool2	(-, $2, 2$) pad $= 0$	Same as VGG-16	Same as VGG-16	Same as VGG-16	(-, $3, 3$) st $= 2$, pad $= 1$
$3 \times$ conv3	$(256, 3, 3)$ pad $= 1$	Same as VGG-16	Same as VGG-16	Same as VGG-16	Same as VGG-16
pool3	(-, $2, 2$) pad $= 0$	Same as VGG-16	Same as VGG-16	Same as VGG-16	(-, $3, 3$) st $= 1$, pad $= 1$
$3 \times$ conv4	$(512, 3, 3)$ pad $= 1$	Same as VGG-16	Same as VGG-16	Same as VGG-16	Same as VGG-16
pool4	(-, $2, 2$) pad $= 0$	Same as VGG-16	None	None	(-, $3, 3$) st $= 1$, pad $= 1$
$3 \times$ conv5	$(512, 3, 3)$ pad $= 1$	Same as VGG-16	Same as VGG-16	Same as VGG-16	$(512, 3, 3)$ dil $= 2$, pad $= 2$
pool5	(-, $2, 2$) pad $= 0$	None	None	ROI-pool 3×3 windows	None
fc6	$(4096, 7, 7)$ pad $= 0$	$(1024, 3, 3)$ pad $= 1$	$(1024, 3, 3)$ pad $= 1$	$(1024, 1, 1)$ pad $= 0$	$(1024, 3, 3)$ dil $= 12$, pad $= 12$
fc7	$(4096, 1, 1)$ pad $= 0$	None	$(1024, 3, 3)$ pad $= 1$	$(1024, 1, 1)$ pad $= 0$	$(1024, 1, 1)$ pad $= 0$
GAP	None	GAP	GAP	Used after fc8	GAP
fc8	$(20, 1, 1)$ pad $= 0$	Same as VGG-16	Same as VGG-16	Same as VGG-16	Same as VGG-16
output heatmap	$(20, 1, 1)$	$(20, 21, 21)$	$(20, 41, 41)$	$(20, 41, 41)$	$(20, 41, 41)$

Table 3. Detailed architecture of the GAP networks. Triplets denote (channel, height, width) for `input` and `output`; for layers, triplets denote (output channel dim, kernel height, kernel width). `st` =stride and `dil` =width of dilated convolution, with default values 1 for both, unless otherwise stated.

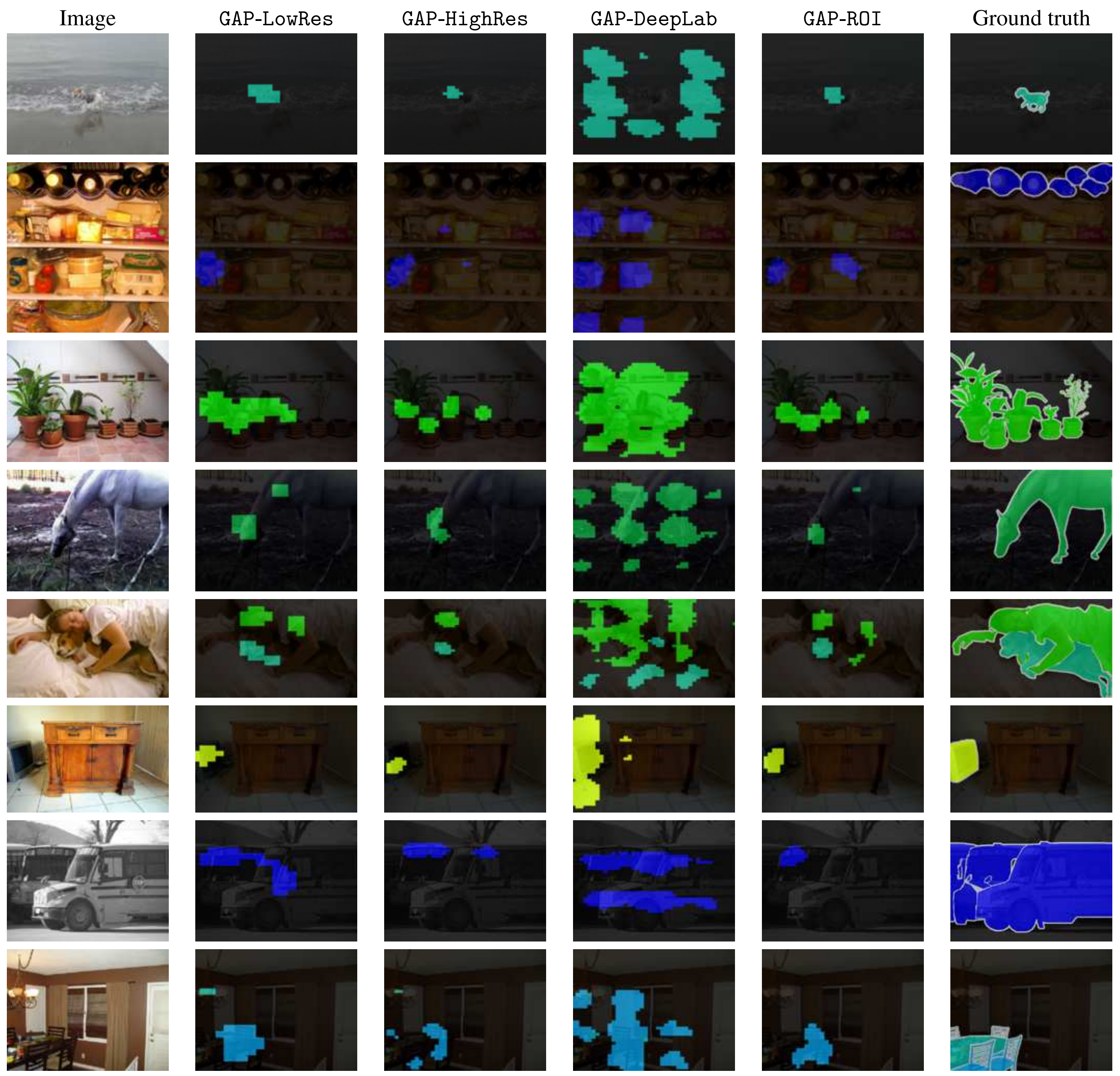

Figure 1. Qualitative examples of GAP output for `GAP-LowRes`, `GAP-HighRes`, `GAP-DeepLab`, and `GAP-ROI`. Note that all of them, except for `GAP-DeepLab`, are qualitatively similar. For `GAP-DeepLab`, we observe repeating patterns of certain stride. Examples are chosen at random.

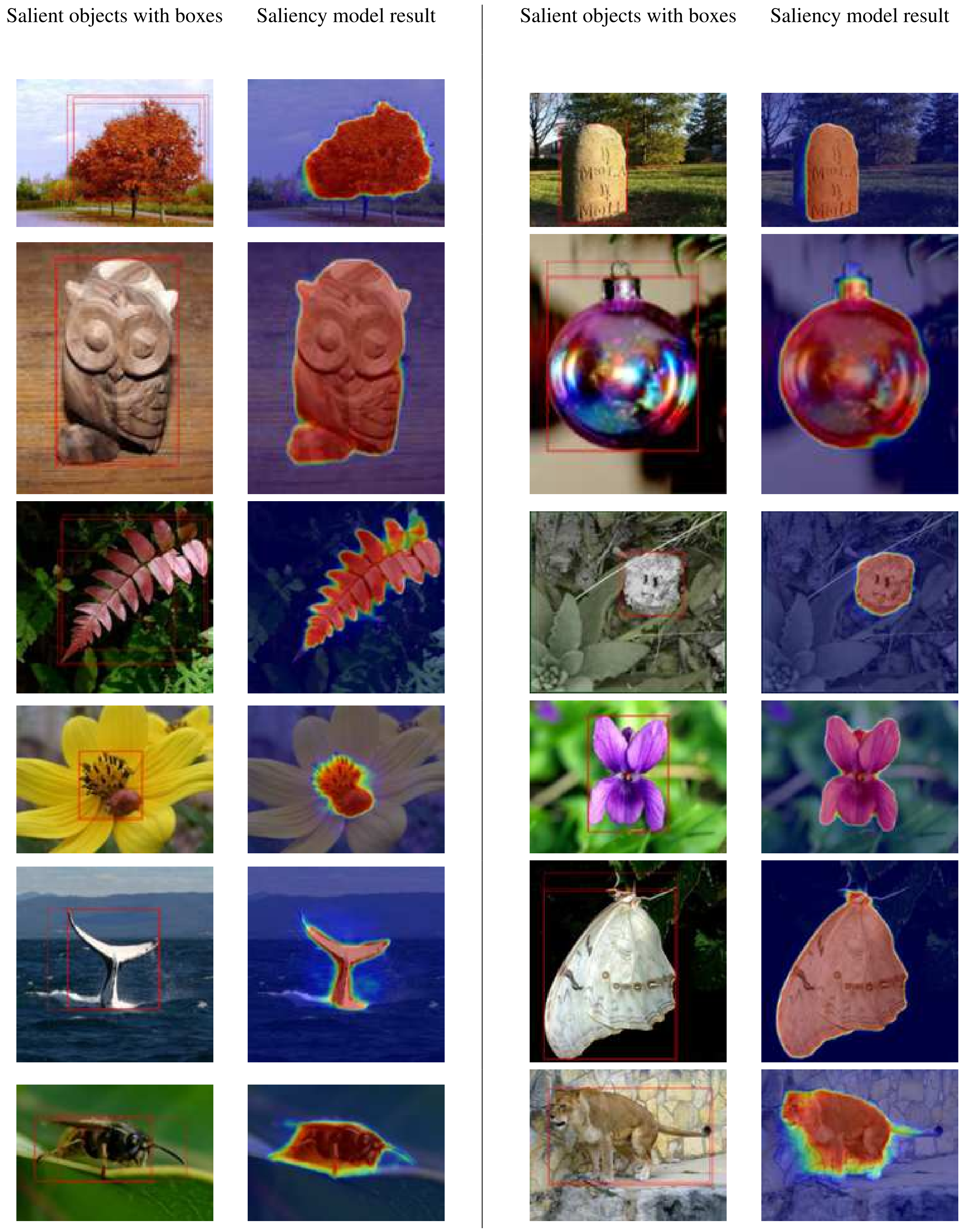

Figure 2. Extension of figure 6 in the main paper. Examples of saliency results on its training data. We use MSRA box annotations to train a weakly supervised saliency model. Note that the MSRA subset employed is not biased towards the Pascal categories. Examples are chosen at random.

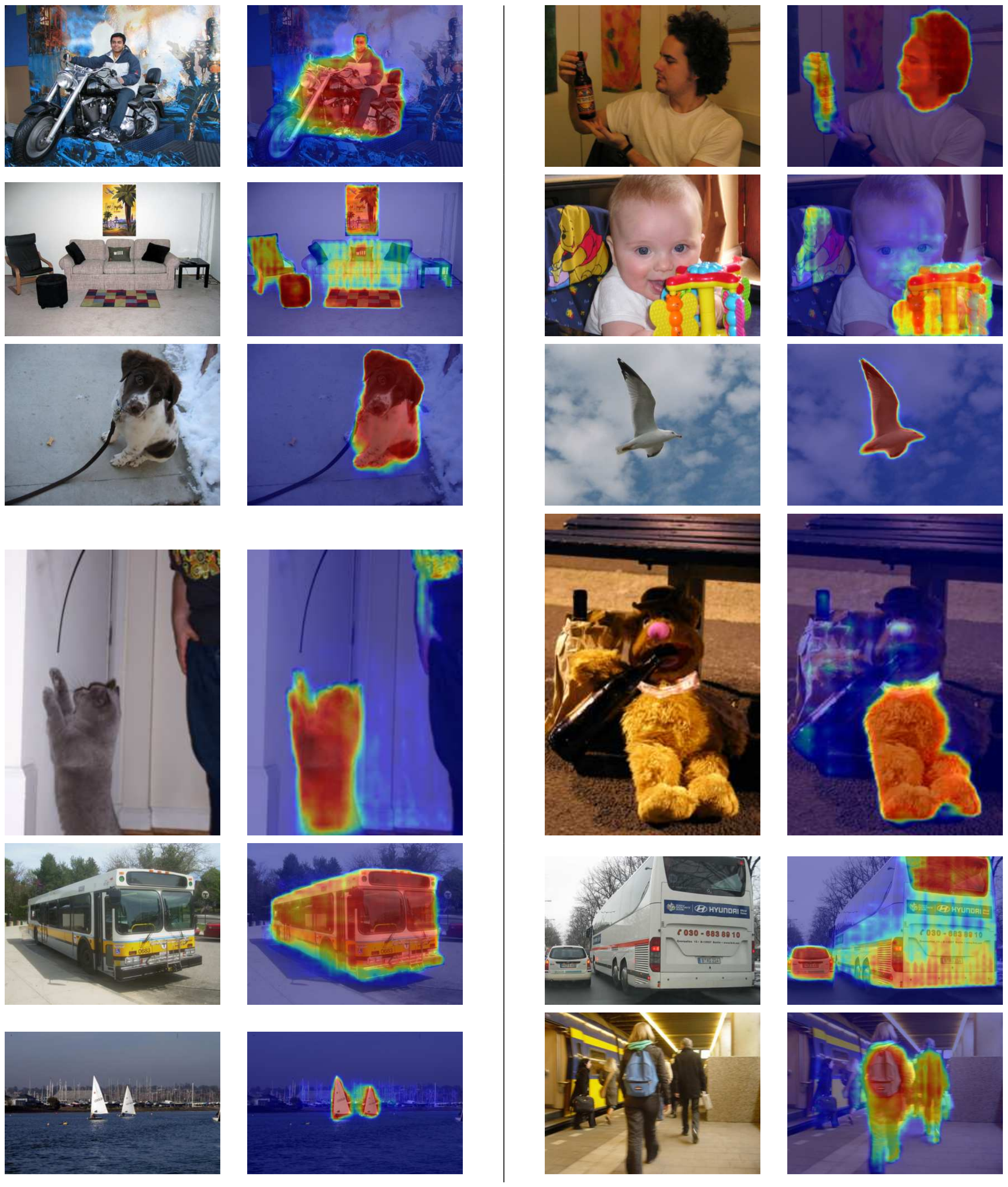

Figure 3. Extension of figure 4 in the main paper. Example of saliency results on Pascal images. We note that the saliency often fails when the central, salient objects are non-Pascal or when the scene is cluttered. Examples are chosen at random.

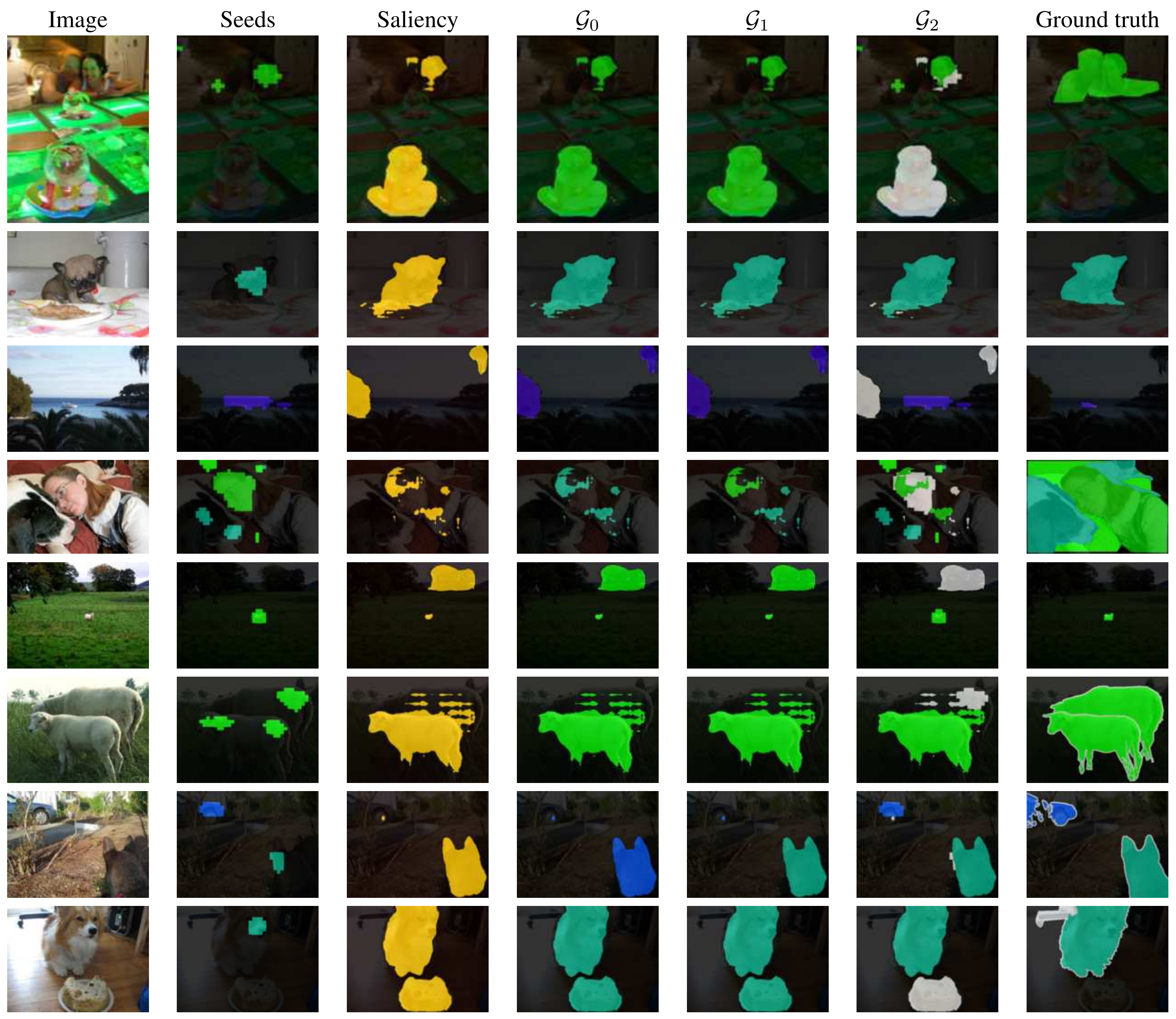

Figure 4. Extension of figure 5 in the main paper. Example results for three different guide labelling strategies, $\mathcal{G}_0$, $\mathcal{G}_1$, and $\mathcal{G}_2$. The image, its image labels, seeds, and saliency map are their input. White labels indicate “ignore” regions. Note that $\mathcal{G}_0$ and $\mathcal{G}_1$ give qualitatively similar results, while $\mathcal{G}_2$ produces much more precise labelling by exploiting rich localisation information from the seeds. Examples are chosen at random.

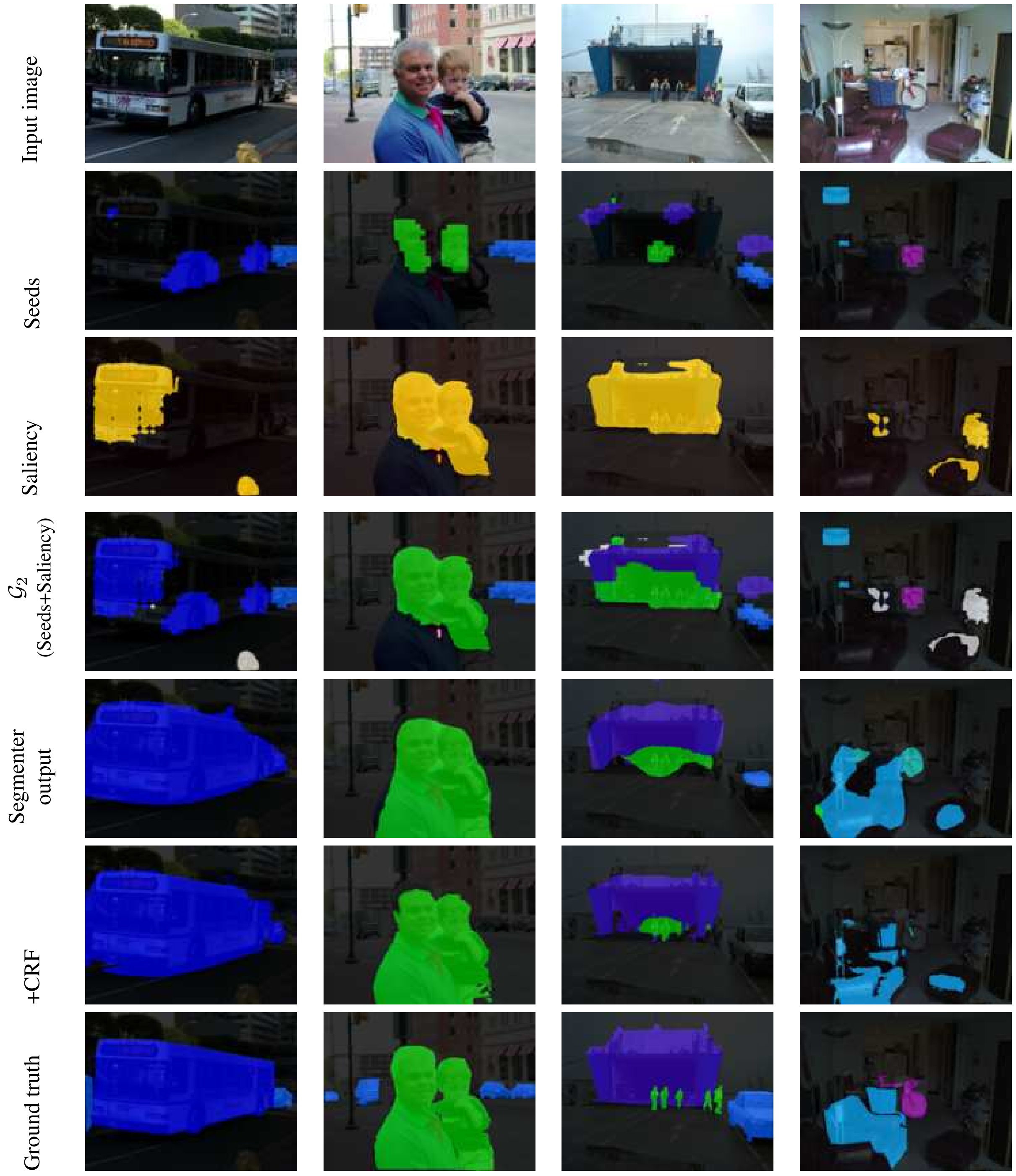

Figure 5. Extension of figure 7 in the main paper. Qualitative examples of the different stages of the Guided Segmentation system on the training images. White labels are "ignore" regions. Seeds have high precision and low recall; combined with saliency foreground mask using $\mathcal{G}_2$ guide labeller, object extents are recovered. The generated guide labelling can still be noisy; however, the segmenter convnet can average out the noise to produce more precise predictions. CRF post-processing further refines the predictions.

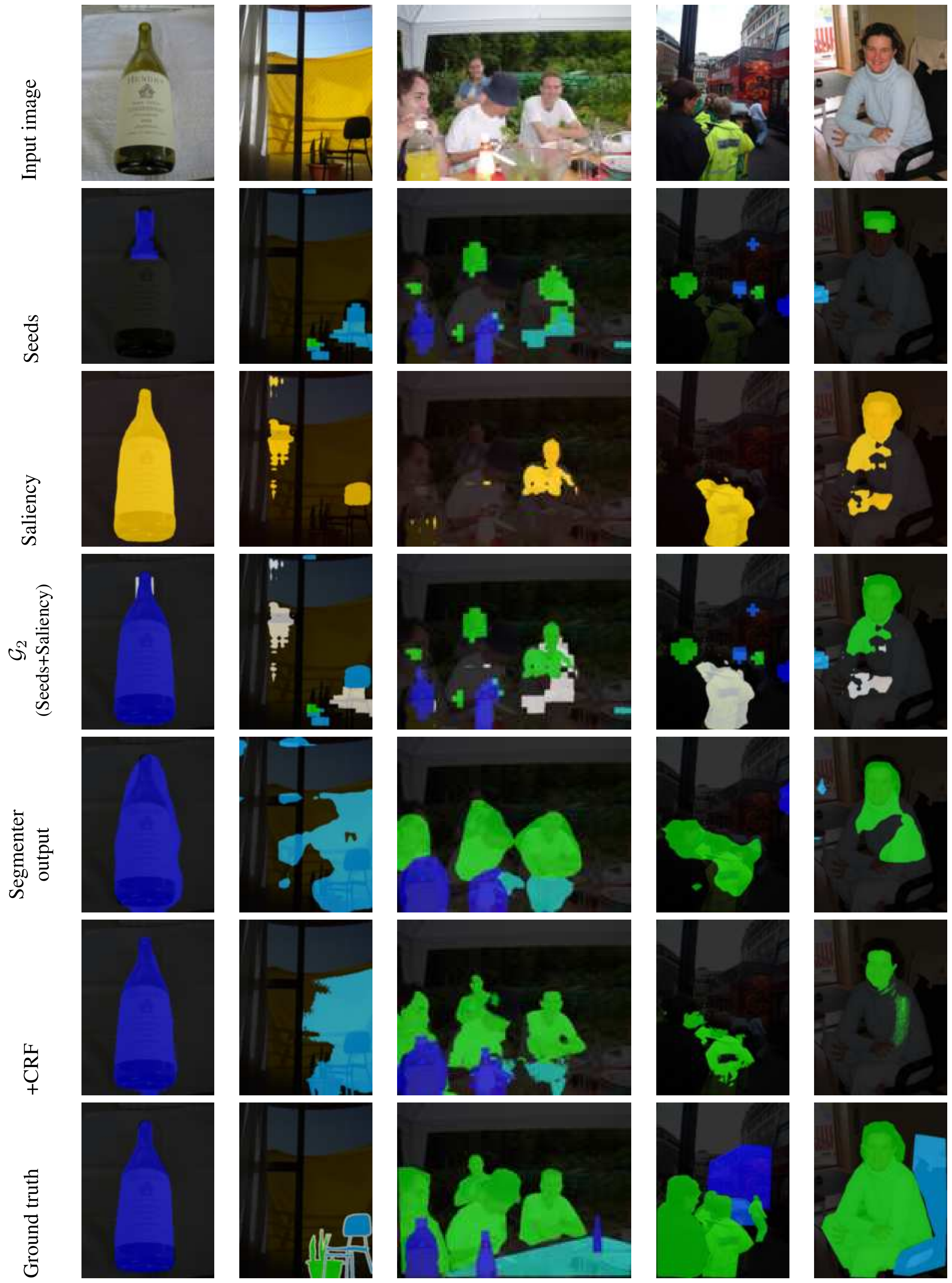

Figure 6. Extension of figure 7 in the main paper. More qualitative examples of the different stages of the Guided Segmentation system on the training images. White labels are "ignore" regions. Seeds have high precision and low recall; combined with saliency foreground mask using $\mathcal{G}_2$ guide labeller, object extents are recovered. The generated guide labelling can still be noisy; however, the segmenter convnet can average out the noise to produce more precise predictions. CRF post-processing further refines the predictions.