Supplementary Material: Annotating Object Instances with a Polygon-RNN

Lluís Castrejón Kaustav Kundu Raquel Urtasun Sanja Fidler
Department of Computer Science
University of Toronto
{castrejon, kkundu, urtasun, fidler}@cs.toronto.edu

We show additional results for our approach on the Cityscapes dataset [1]. Note that in all our experiments we assume to be given a ground-truth box around the object. Our goal then is to provide a polygon outlining this object as accurately as possible and with minimal number of clicks required from the annotator. While a box around the object in principle requires two additional clicks, boxes are typically much easier and cheaper to obtain using crowd-sourcing services such as AMT. On the other hand, for most major segmentation benchmarks, polygons have been collected with high quality annotators.

In particular, in Table 1 we show an additional experiment in which we train the model on only one category (i.e., *car*) and test it on all the categories. While this result is not at the level of the model trained on all objects, it performs surprisingly well, particularly for the vehicle categories. This shows that our model essentially learns to follow boundaries, and is thus able to generalize to other classes. This is a desirable property for a generic annotation tool to facilitate labeling of various diverse, potentially not-seen-before classes.

Full image annotation in prediction mode. We show full image results from our approach without any correction (i.e., 0 clicks) in Fig. 1- 4. In the first column, we show the GT provided by the Cityscapes dataset [1], and in the second column, we show results from our approach. Below each image we show the number of clicks required to annotate the GT polygons as well as the number of bounding boxes (*i.e.* the number of instances) in the image. Note that in this experiment our method requires only the bounding boxes around the objects.

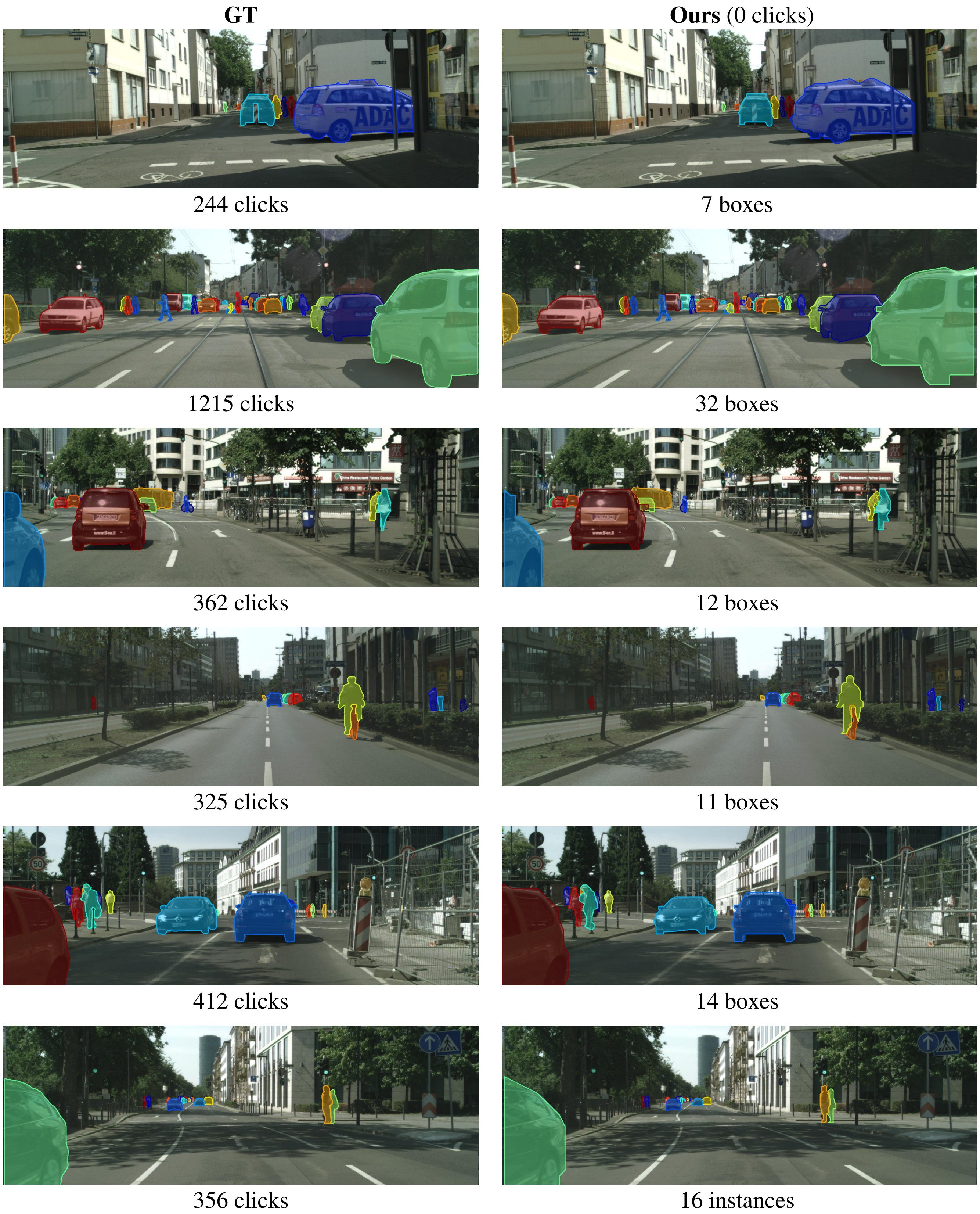
Examples of humans-in-the-loop. In Fig. 5- 8, we show visualizations of the instances inside the crop of the GT boxes. In the first column, we show the ground-truth polygon annotation, while in the second column, we show the output from SharpMask [3]. Note that Sharpmask predicts pixel labeling of the image (crop in this case). In our visualization, we draw the boundary based on the connectivity in the 8-neighborhood. In the third column, we report our predictions without any human intervention. Finally, in the fourth column, we show the "human-in-the-loop" results, where we provide our PolygonRNN with a correction of a point if the prediction deviates from the ground-truth vertex by 1 pixel. For each example, we show the number of vertices in the original GT annotation and the number of corrections needed in our model.

We can observe that our model performs well on small instances, particularly `person` or `bicycle` (rows 1, 2 and 3 of Fig. 5). Our model also performs well on `car` since they have simpler shapes (row 4 of Fig. 5 and row 1 of Fig. 6). The blocky effects on some of the objects is due to the resolution of our output (our RNN operates on a 28 × 28 grid). We plan to increase our output resolution in the future, requiring architectural changes to our RNN due to memory constraints.

In Fig. 6 rows 2, 3, 4, we show some failure cases. For big instances (such as `bus`, `truck` or `train`) both SharpMask and our model achieve a lower performance. The most likely reason for this is that there are fewer big instances in the dataset. Fig. 7 row 1 shows an instance with multiple components. Our model tends to include occlusion inside the instance segmentation. By providing a few corrections, our model can recover and provide a much better labeling (column 4). Row 2 shows a trend of SharpMask in providing worse annotations for smaller instances. While overall, the predictions of SharpMask are quite good, we can still see many examples of mistakes. From the labeling perspective and the nature of SharpMask (and similar dense pixel-labeling approaches), such examples are useless since the annotator needs to re-label them from scratch. The main advantage of our method is that the interaction with the human labeler comes very naturally, and allows the annotator to obtain a good annotation with only a few clicks.

Model	Bicycle	Bus	Person	Train	Truck	Motorcycle	Car	Rider	Mean
Square Box	35.41	53.44	26.36	39.34	54.75	39.47	46.04	26.09	40.11
Dilation10	46.80	48.35	49.37	44.18	35.71	26.97	61.49	38.21	43.89
DeepMask [2]	47.19	69.82	47.93	62.20	63.15	47.47	61.64	52.20	56.45
SharpMask [3]	52.08	**73.02**	53.63	**64.06**	65.49	51.92	65.17	56.32	60.21
Ours (trained on cars only)	41.80	68.80	44.60	51.14	66.13	50.78	**73.56**	43.50	55.03
Ours (trained on all classes)	**52.13**	69.53	**63.94**	53.74	**68.03**	**52.07**	71.17	**60.58**	**61.40**

Table 1. **Performance** (IoU in %) on all the Cityscapes classes **without the annotator in the loop**.



244 clicks

7 boxes

1215 clicks

32 boxes

362 clicks

12 boxes

325 clicks

11 boxes

412 clicks

14 boxes

356 clicks

16 instances

Figure 1. **Full image:** here we show our results for all instances in an image. We remind the reader that that our approach exploited (ground-truth) boxes to be provided as input. On the **left** we show the ground-truth labeling of the image, while on the **right** we show our polygons in the 0-click regime (running in automatic prediction mode).

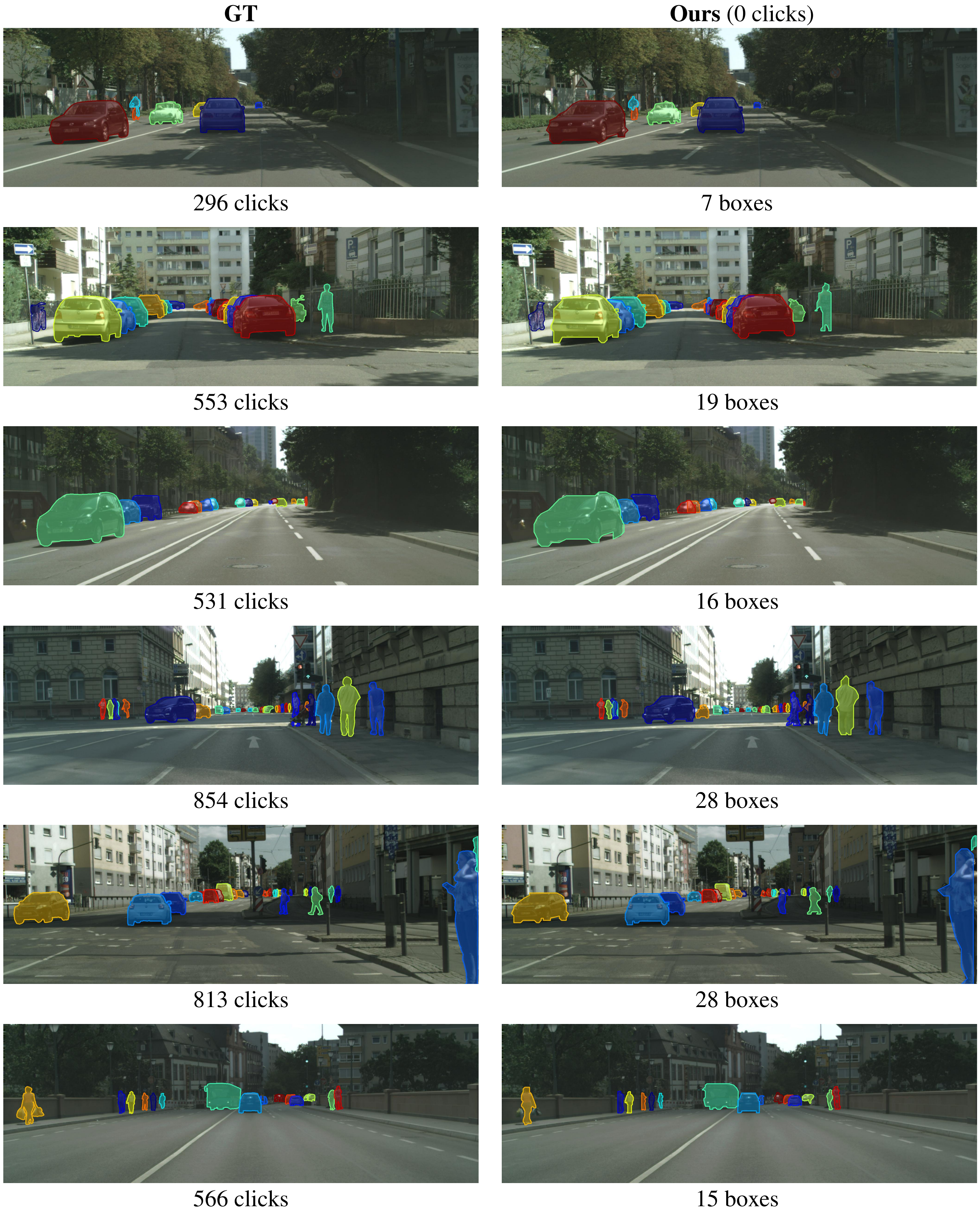


Figure 2. **Full image:** here we show our results for all instances in an image. We remind the reader that that our approach exploited (ground-truth) boxes to be provided as input. On the **left** we show the ground-truth labeling of the image, while on the **right** we show our polygons in the 0-click regime (running in automatic prediction mode).

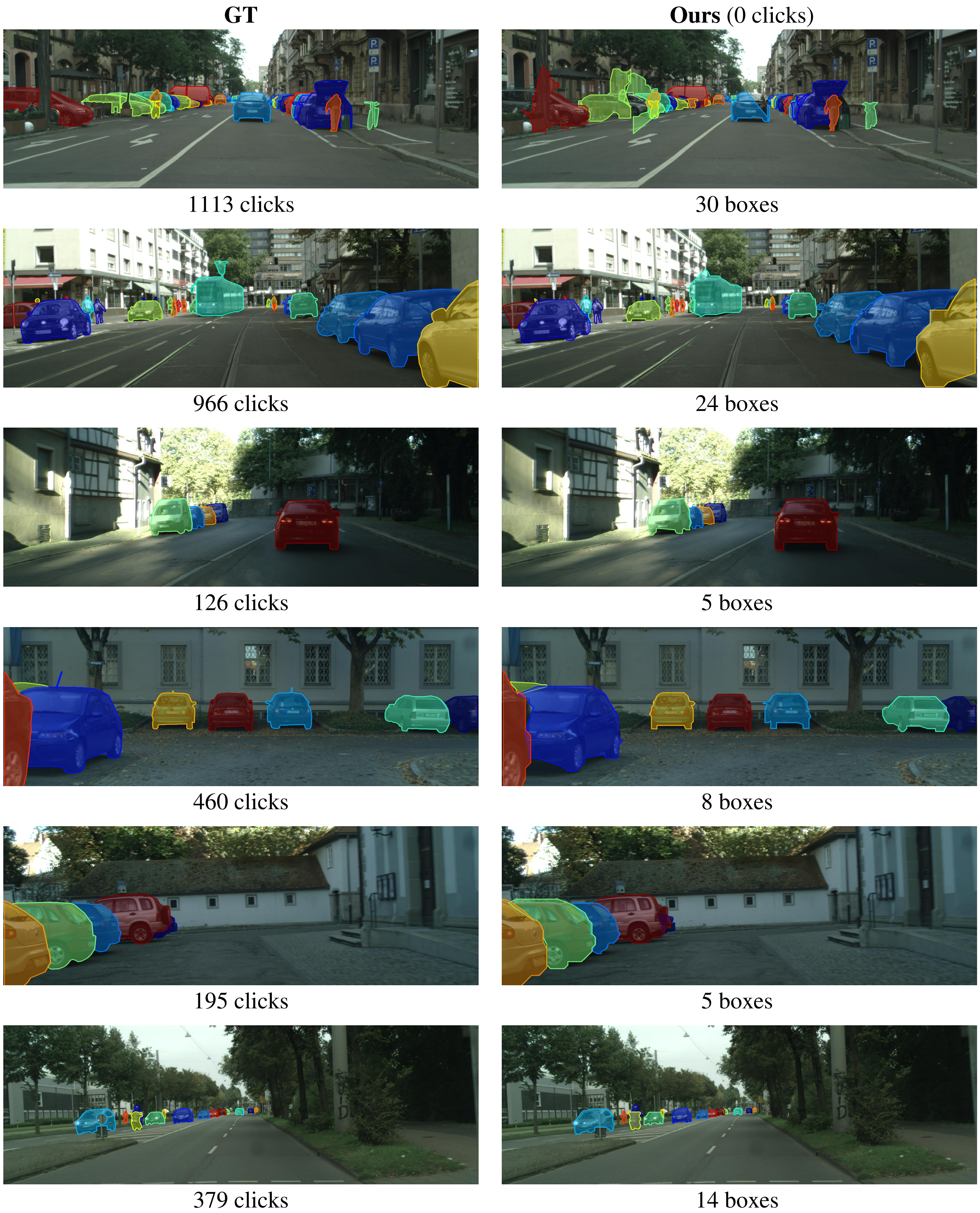
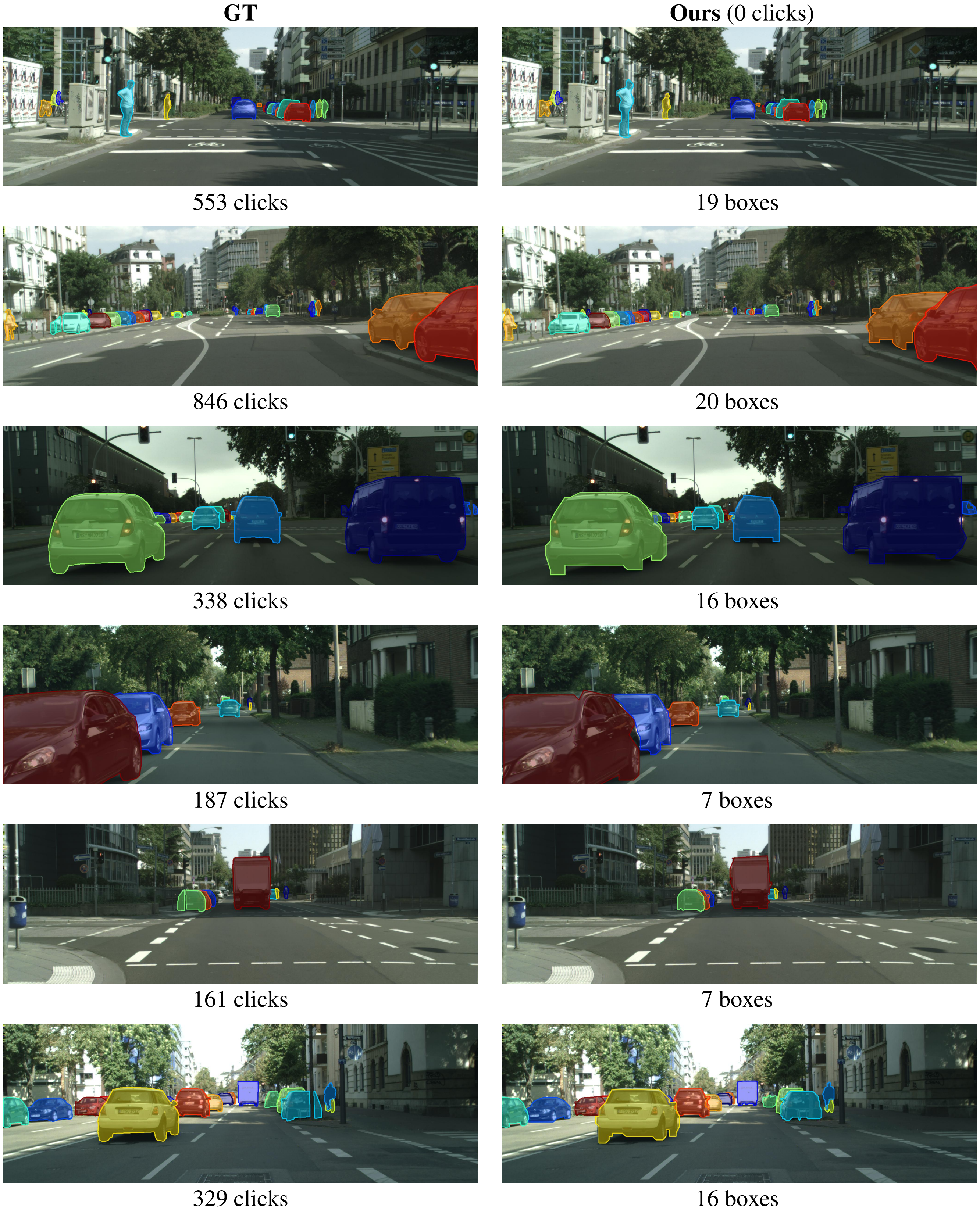


Figure 3. **Full image:** here we show our results for all instances in an image. We remind the reader that that our approach exploited (ground-truth) boxes to be provided as input. On the **left** we show the ground-truth labeling of the image, while on the **right** we show our polygons in the 0-click regime (running in automatic prediction mode).



GT

Ours (0 clicks)

553 clicks

19 boxes

846 clicks

20 boxes

338 clicks

16 boxes

187 clicks

7 boxes

161 clicks

7 boxes

329 clicks

16 boxes

Figure 4. **Full image:** here we show our results for all instances in an image. We remind the reader that that our approach exploited (ground-truth) boxes to be provided as input. On the **left** we show the ground-truth labeling of the image, while on the **right** we show our polygons in the 0-click regime (running in automatic prediction mode).

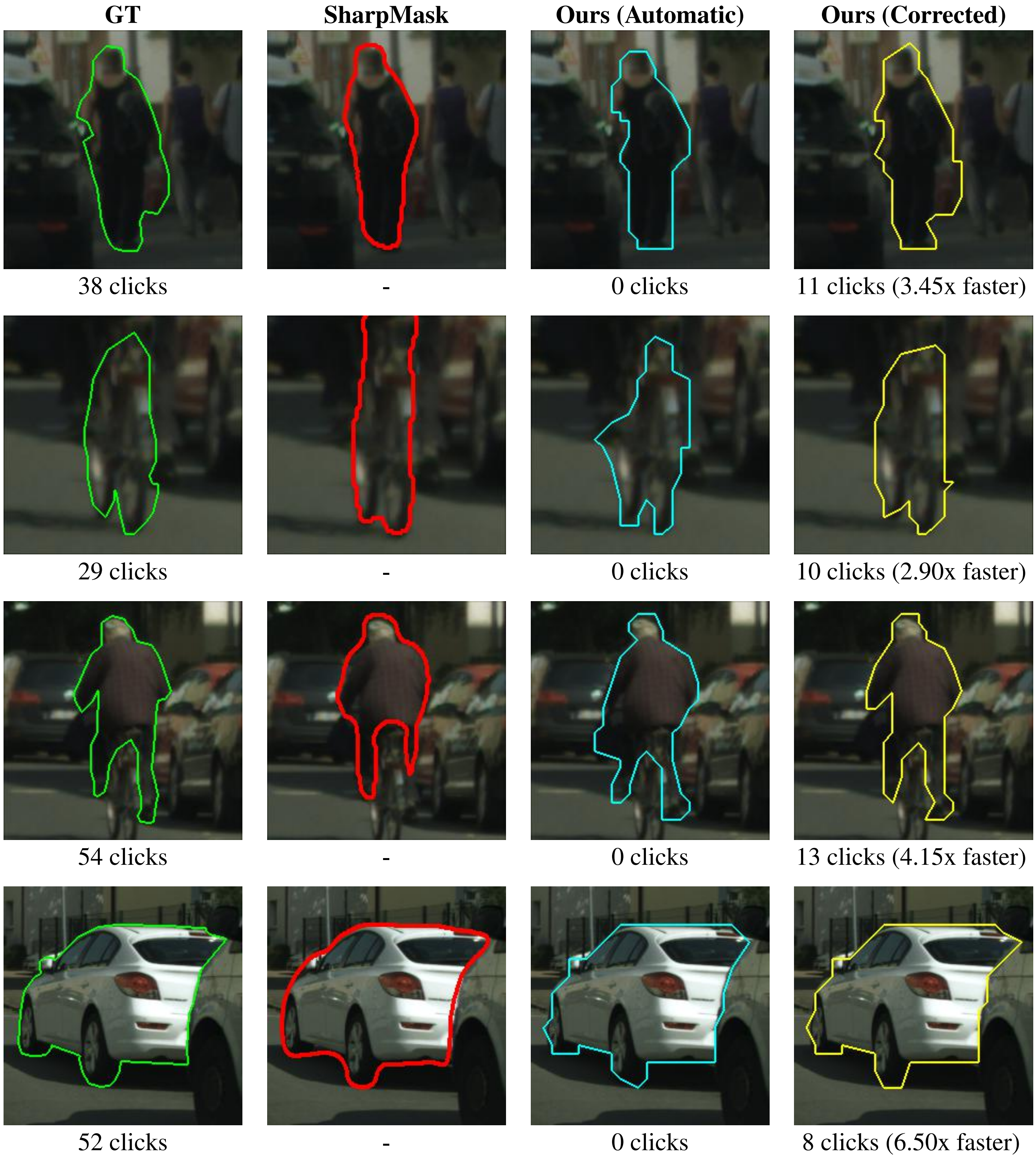


Figure 5. The **first column** we show the GT annotation, while on the **second column**, we show the output from SharpMask. On the **third column** we report the PolygonRNN prediction without human intervention. Finally, on the **fourth column** we show a corrected prediction using a distance threshold of 1, showing how we can refine our model predictions to obtain high quality annotations.

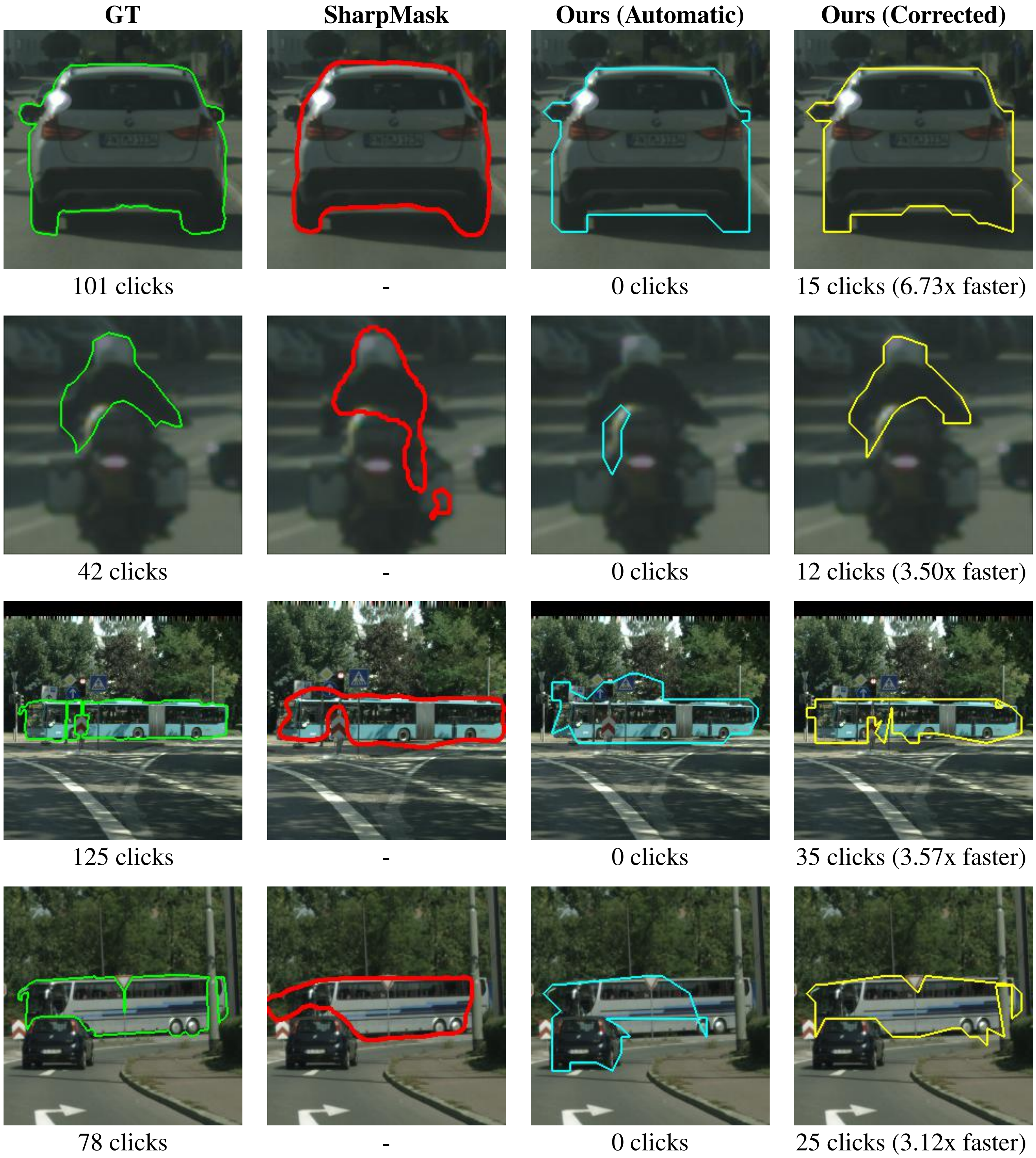


Figure 6. The **first column** we show the GT annotation, while on the **second column**, we show the output from SharpMask. On the **third column** we report the PolygonRNN prediction without human intervention. Finally, on the **fourth column** we show a corrected prediction using a distance threshold of 1, showing how we can refine our model predictions to obtain high quality annotations.

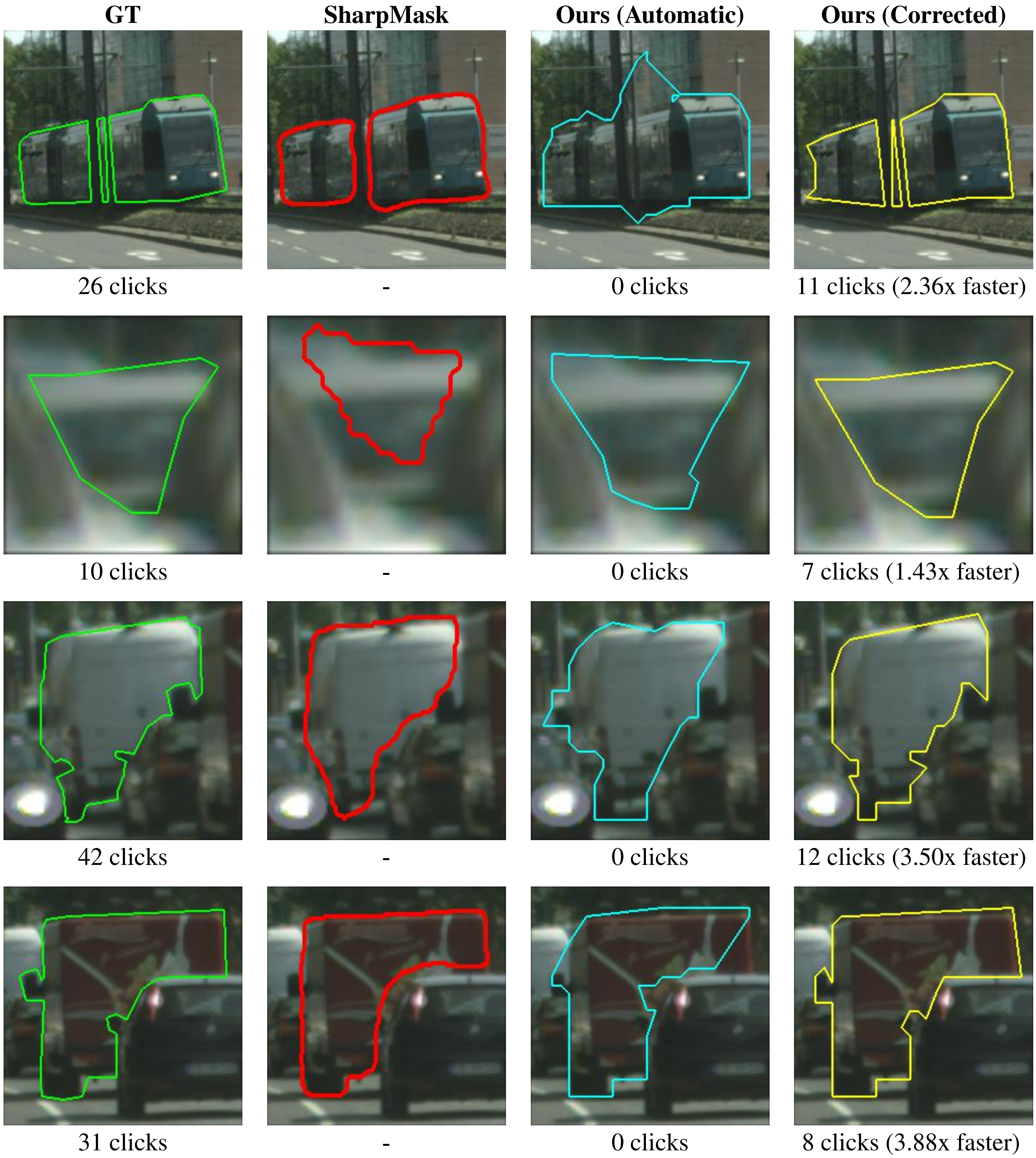


Figure 7. The **first column** we show the GT annotation, while on the **second column**, we show the output from SharpMask. On the **third column** we report the PolygonRNN prediction without human intervention. Finally, on the **fourth column** we show a corrected prediction using a distance threshold of 1, showing how we can refine our model predictions to obtain high quality annotations.

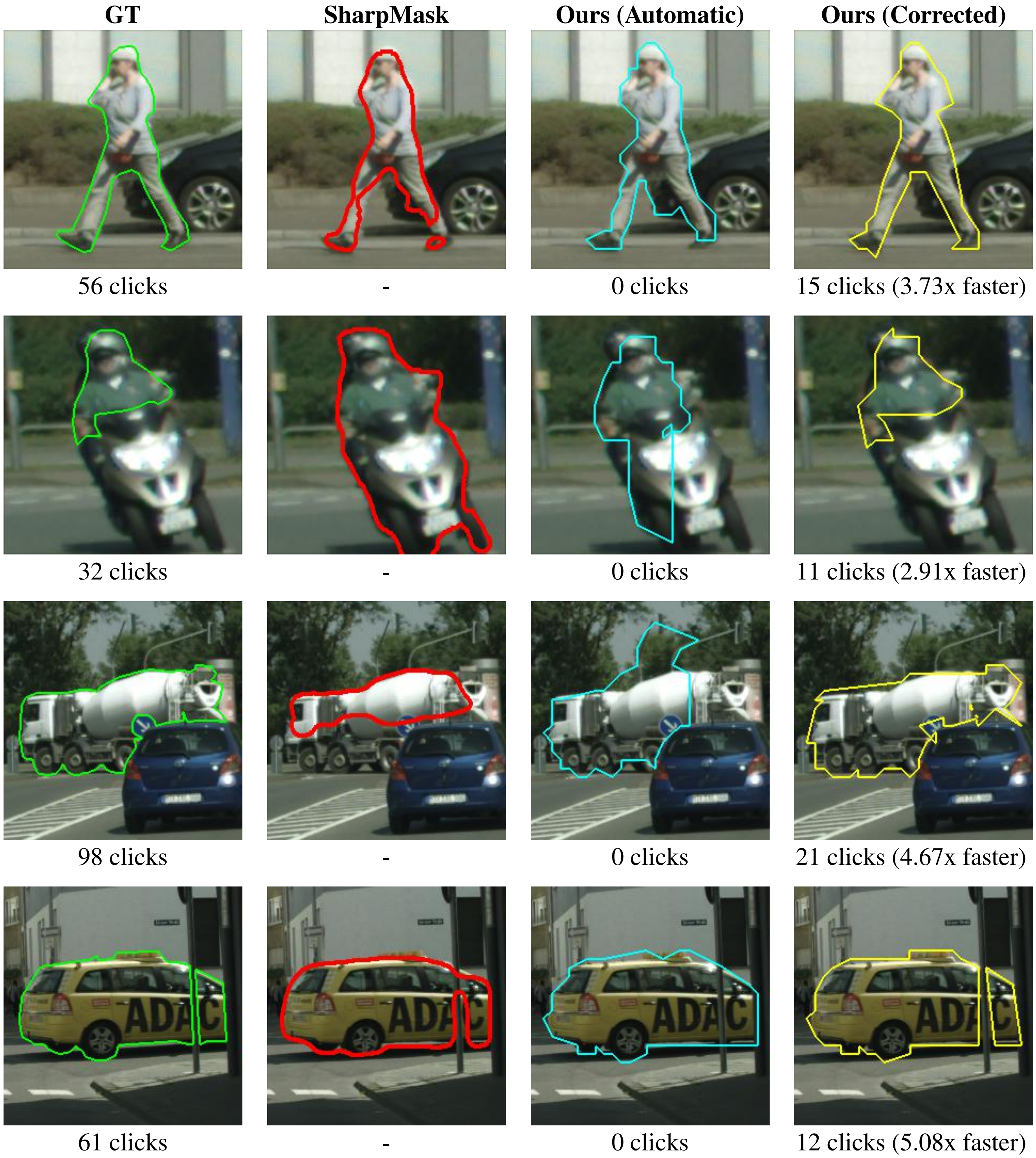


Figure 8. The **first column** we show the GT annotation, while on the **second column**, we show the output from SharpMask. On the **third column** we report the PolygonRNN prediction without human intervention. Finally, on the **fourth column** we show a corrected prediction using a distance threshold of 1, showing how we can refine our model predictions to obtain high quality annotations.

References

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1

[2] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *NIPS*, pages 1990–1998, 2015. 1

[3] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. *ECCV 2016*, 2016. 1