Human-Centric Efficiency Improvements in Image Annotation for Autonomous Driving

Martine Bertrand* [1] **Frédéric Ratle*** [1] **Loic Juillard** [1]

Abstract

We present an instance segmentation few-click annotation tool that significantly improves labelling efficiency. This is achieved by combining the well-known DEXTR approach with a raster-to-polygon conversion algorithm that yields high quality polygons whose vertices are sampled in a way that reproduces human drawing patterns. Furthermore, we demonstrate the importance of integrating the user input into the model to encourage a more constructive human-machine interaction.

1. Introduction

Building instance segmentation Deep Learning (DL) models for autonomous vehicles requires significant amount of labelled data. The use of Machine Learning (ML) for producing pre-annotations to be reviewed by human annotators, whether in an interactive setting or as a pre-processing procedure, is a very popular approach for scaling up labelling while controlling the costs. However, few studies have approached ML integration from a human-centric perspective, i.e. what interactions are most desirable and how best to present the output of the model to annotators?

In this work, we implement an instance segmentation model based on the Deep Extreme Cut (DEXTR) approach (Maninis et al., 2018) and some clever contouring algorithm that delivers high quality polygonal annotations derived from a few clicks provided by human annotators. We show:

- That interactive polygonal annotation is significantly faster than its full manual counterpart while preserving quality.
- The importance of proper human/machine interactions for improving annotation efficiency.

*Equal contribution [1]Samasource, Montréal, Canada. Correspondence to: Frédéric Ratle <fratle@samasource.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

2. Related work

As image and video annotation is a time-consuming and costly process, much research effort has been dedicated to image pre-annotation, and few-click interactive annotation.

We will primarily focus on **instance segmentation** from images, a task we identified as being time-consuming for annotators. Several works such as (Russakovsky et al., 2015) and (Papadopoulos et al., 2017) present interactive annotation schemes for 2D bounding boxes, and we will address this annotation output in a future study.

Multiple approaches have been suggested for machine-assisted instance segmentation. These typically consist of a DL-based segmentation of the object(s) integrated into a human-in-the-loop system. The human can interact with the system by correcting the model output, initializing the model with one or several clicks, or a combination of those steps. Examples of such systems include Polygon-RNN++ (Acuna et al., 2018), DELSE (Wang et al., 2019), DEXTR (Maninis et al., 2018), and CurveGCN (Ling et al., 2019). For video, a method for spatio-temporal segmentation is presented in (Jain & Grauman, 2016). In (Andriluka et al., 2018), authors provide, in addition, an in-depth analysis of the proposed system (Fluid Annotation) with respect to the number of actions required for humans and the precision/recall of the pixel-wise label agreement. Those systems all present good results, but there are common caveats:

- Do the findings hold if a higher, production-level accuracy is required, as when working for a customer project? In the present study, we are using realistic labelling guidelines.
- Does the choice of annotation tool influence the results? The gains to be made by using ML depend on how difficult it is for humans to draw polygons in the provided UI. We use a proprietary optimized drawing tool for polygons in this work.

3. Methodology

We explore the effectiveness of interactive few-click instance segmentation by training a DL model as part of a

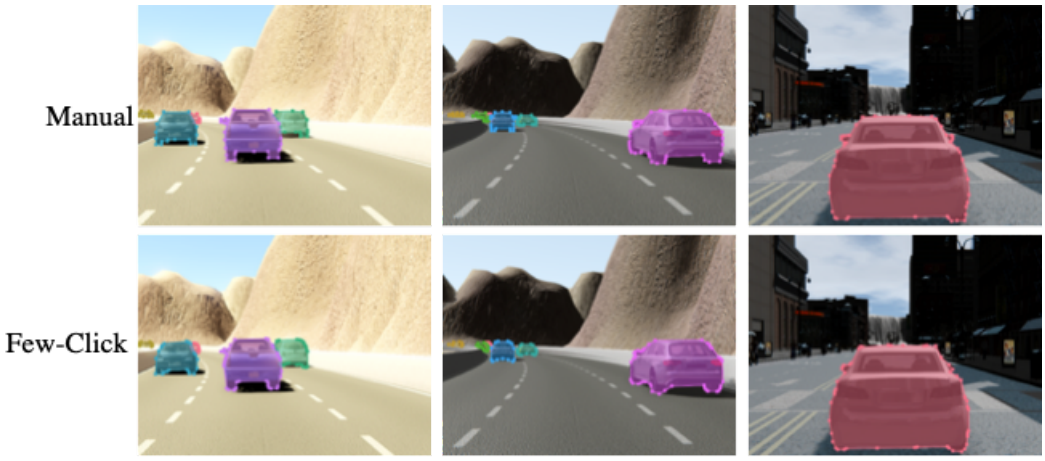

Figure 1. Comparing polygonal annotations of motor vehicles: (top) polygons drawn from scratch; (bottom) polygons fine-tuned from the output of our few-click annotation tool triggered by the annotator's extreme clicks.

system capable of assisting human annotators on a task regarding motor vehicle segmentation and by performing rigorous A/B testing of the said system in a production setting.

3.1. Data

We used the colour images and instance masks within the train portion of the SYNTHIA-AL synthetic automotive dataset (Zolfaghari Bengar et al., 2019) in order to compute IoUs with respect to exact ground truths. The dataset's images and corresponding annotations were generated from video streams at 25 FPS. To train our few-click model, we used a 80/20 train/validation split at the stream level. For our A/B testing experiments, we selected a diverse set of 50 images containing motor vehicles from the validation portion of our data.

3.2. ML System

We wish to assist human annotators in an instance segmentation task. As such, we've developed a Machine Learning system comprised of a simple few-click segmentation model (custom DEXTR) and a post-processing procedure that converts the produced raster mask in a high quality sparse polygon.

3.2.1. Few-Click Segmentation Model

We built a segmentation model based on the DEXTR (Maninis et al., 2018) approach summarized below:

1. Derive a heatmap either from simulated extreme clicks around a desired object instance at train time or from a user input at inference time.
2. Concatenate the heatmap to the image of the said instance to make an input.
3. Pass the input through a segmentation model of your choice to get a prediction of the raster mask.
4. If training, compare with ground truth (IoU), calculate loss, and backpropagate.

Our segmentation model is a custom UNet (Ronneberger et al., 2015) based on an EfficientNet B-4 backbone (Tan & Le, 2019) and trained on the data described in Section 3.1.

3.2.2. Raster-to-Polygon

In our experience, for human annotators to produce high quality instance segmentation masks efficiently, a polygon annotation tool should be used. As such, we needed to convert the raster masks produced by our model to high quality polygons. To add to the challenge, humans tend to produce sparse polygons, adding vertices only when curvature significantly changes. We thus devised the following raster-to-polygon procedure:

1. Blur the edges of the raster mask using a Gaussian kernel.
2. Find an iso-contour C using the Marching Squares algorithm.
3. Output a polygon by sparsifying C based on local curvature κ, *i.e.* discard vertices where $\kappa \to 0$.

3.3. A/B Testing

To assess the usefulness of our interactive few-click polygonal segmentation system, we have set up the following A/B experiments:

1. A comparison of manual annotation (from scratch) of motor vehicles with our few-click annotation tool. This is aimed at confirming the efficiency gains that can be made while preserving a realistic accuracy criterion.
2. A comparison of the few-click annotation results with a form of pre-annotation using *simulated* clicks. This is achieved by simply selecting the extreme top, bottom, left and right points on the ground truth mask. This is **not** a real pre-annotation experiment, as those points are not available in a real scenario. What we want to measure is the influence of giving the annotator the flexibility of picking those points on the annotation *adjustment* time. We cannot compare the overall annotation time in this case, as the simulated clicks are automatically inserted, yielding an unfair advantage over the few-click baseline.

In the setup of our A/B experiments, we followed this protocol:

- A minimum of 10 annotators should be working on each variant.
- No annotator should work on more than one variant.
- In each of the groups, we should have a similar representation of (i) level of skill and experience, (ii) type of workstation, and (iii) type of shift (night/day).
- Annotators should not be told more details than necessary about the experiment. Only the guidelines specific to their variant are needed.

3.4. Labelling guidelines

The following instructions have been distributed to all annotators:

- **Label classes**: Only label Motor Vehicles, using polygons. Each vehicle gets its own polygon.
- **Pixel-level thresholds**: Any object with less than 10 visible pixels (height or width) will be ignored.
- **Accuracy requirement**: Polygon needs to be within 2 pixels of edge of vehicle.

3.5. Variant-specific guidelines

Annotators working on different experimental variants have been given instructions as follows:

1. Manual annotation (*Manual*)
   (a) Draw polygons around all motor vehicles in the images, following the guidelines specified in 3.4.
2. Few-click annotation (*FewClick*)
   (a) To draw a mask on the object of your choice, select the FewClick tool, click on 4 extreme points that lie on the object boundary.
   (b) Edit the obtained polygon if needed.
   (c) If the obtained polygon fulfills the minimum accuracy criterion, there is no need to edit it. Only edit the polygon if you think it does not pass the accuracy criterion.
3. Simulated few-click annotation (*SimFewClick*)
   (a) Edit the pre-loaded polygons to make them fit the quality requirements specified in 3.4.
   (b) If you see a polygon that you would have not drawn, you should not edit it, but rather delete it.
   (c) If the pre-loaded polygon fulfills the minimum quality criterion, there is no need to edit it. You should only edit the polygon if you think it does not pass the quality criterion.

4. Results

Human annotators take on average $\bar{t}_{\text{total}} = 106.2$ seconds to produce a polygonal outline per given object instance (Table 1) and only 31.2 seconds on average using the few-click tool, out of which ~ 7.4 seconds are for the initial four clicks, in line with the 7.2 seconds reported in (Papadopoulos et al., 2017). This represents a significant improvement in efficiency while maintaining a high level of quality as testified by the respective mean IoUs in Table 1. Figure 2 (top) shows that few-click assisted annotation also significantly reduces the variability in $\bar{t}_{\text{total}}$ over annotators. We hypothesize that this is a result of reducing the effect of the "skill level" covariate.

Our results also hint at the importance of preserving human annotators control over the machine. Indeed, in our third experiment where we simulated the few-click part and asked annotators to correct the output polygons, we observe an increase in average adjustment time $\bar{t}_{\text{adjust}}$ per object instance (from 23.8 to 29.1 seconds) with a marginally better mean IoU when compared to annotators that used the few-click system as intended. We hypothesize that the act of drawing the initial four points makes annotators more accepting of the machine's output.

Variant	$\bar{t}_{\text{total}}$ (s/object)	$\overline{\text{IoU}}$
FewClick	31.2	0.951
Manual	106.2	0.957

Table 1. Average total time spent annotating (sec) and IoU for *Manual* vs *FewClick*.

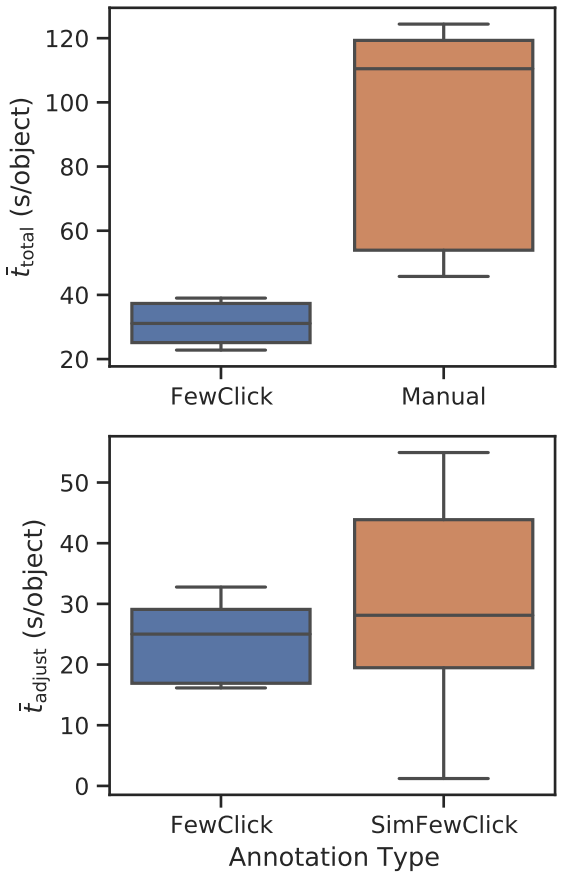

Figure 2. Distribution of average annotation time per object instance (in seconds) across all human annotators and for the three labelling variants. Included is the average time for drawing adjustments.

Variant	$\bar{t}_{\text{adjust}}$ (s/object)	$\overline{\text{IoU}}$
FewClick	23.8	0.951
SimFewClick	29.1	0.955

Table 2. Average time spent adjusting shapes (sec) and IoU for *FewClick* vs *SimFewClick*.

5. Conclusion

We demonstrated that using a group of annotators and production-level quality requirements, few-click segmentation increases efficiency by 3-4X while maintaining the same quality. An interesting avenue to further explore is that part of this efficiency improvement seems to arise from the annotators' involvement in the process; having them select the extreme points appears to make them more confident in the model prediction, and perhaps less likely to edit the output for subjective reasons. We've also introduced a simple raster-to-polygon procedure to enable a more efficient editing of the model prediction.

References

Acuna, D., Ling, H., Kar, A., and Fidler, S. Efficient annotation of segmentation datasets with polygon-rnn++. In *CVPR*, 2018.

Andriluka, M., Uijlings, J. R., and Ferrari, V. Fluid annotation: a human-machine collaboration interface for full image annotation. In *ACM Multimedia*, 2018.

Jain, S. D. and Grauman, K. Click carving: Segmenting objects in video with point clicks. In *AAAI Conference on Human Computation and Crowdsourcing*, 2016.

Ling, H., Gao, J., Kar, A., Chen, W., and Fidler, S. Fast interactive object annotation with curve-gcn. In *CVPR*, 2019.

Maninis, K.-K., Caelles, S., Pont-Tuset, J., and Van Gool, L. Deep extreme cut: From extreme points to object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Papadopoulos, D., Uijlings, J., Keller, F., and Ferrari, V. Extreme clicking for efficient object annotation. In *ICCV*, 2017.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Russakovsky, O., Li, L.-J., and Li, F. F. Best of both worlds: Human-machine collaboration for object annotation. In *CVPR*, 2015.

Tan, M. and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019. URL http://arxiv.org/abs/1905.11946.

Wang, Z., Acuna, D., Ling, H., Kar, A., and Fidler, S. Object instance annotation with deep extreme level set evolution. In *CVPR*, 2019.

Zolfaghari Bengar, J., Gonzalez-Garcia, A., Villalonga, G., Raducanu, B., Aghdam, H. H., Mozerov, M., Lopez, A. M., and van de Weijer, J. Temporal coherence for active learning in videos. *arXiv preprint arXiv:1908.11757*, 2019.