Actor-Critic Instance Segmentation
– Supplemental Material –

Nikita Araslanov* Constantin A. Rothkopf†,§ Stefan Roth*,§
*Dept. of Computer Science †Institute of Psychology §Centre for Cognitive Science
TU Darmstadt

A. Ordering of Prediction

As we discussed in the main text, previous work suggests that the overall accuracy of a recurrent model is not invariant to the prediction order [20, 37]. To confirm this experimentally for instance segmentation, we decouple the localisation and the segmentation aspect of a recurrent model in an oracle experiment. Reserving the location of the ground-truth segments as the oracle knowledge (*e.g.*, with a bounding box) allows us to control the prediction order and study its role for the quality of the pixelwise prediction.

The setup for our oracle experiment is illustrated in Fig. 5. We supply the image and context patches (4 input channels) to a segmentation network in random order, while at the same time accumulating its predictions into the (global) context mask. We repeat this procedure 20 times for every image, with a different random order each. We use the CVPPP train/val split from our ablation study, and report the results on the 25 images in the validation set. The architecture of the network is the same as the segmentation module used by Ren and Zemel [31].

Figure 6a shows the statistic of the results (mean and standard deviation) for each of the 25 images. We observe that there is a tangible difference between the different segmentation orderings. The mean of the maximum and the minimum Dice across the images are 86.5 and 78.1 Dice, respectively, which corresponds to a potential gain of 8.4 mean Dice for the optimal prediction order. Notably, the deviation between the runs varies depending on the image. To investigate this further, we show two examples from the validation set with the highest (index #1, Fig. 6b) and the lowest (index #19, Fig. 6c) deviation. Example #1 appears to contain many more occlusions and more complex segment shapes (*e.g.*, a single stalk) compared to example #19. This aligns well with our expectation: the benefit of the context should be more pronounced in more complex scenes.

B. Architecture

The architecture of the actor network and its parameters depending on the dataset used are shown in Tables 5 and 6.

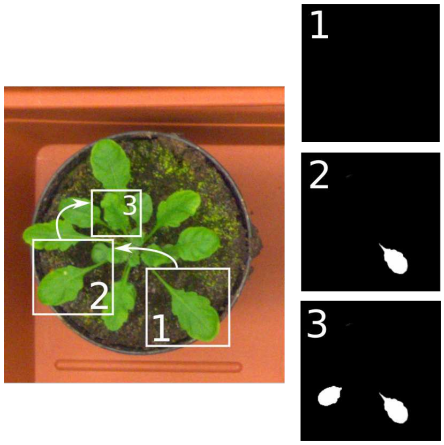


Figure 5. *Oracle experiment:* We extract patches of the image and the context mask centred on the ground-truth instances and supply only those to the segmentation network. The oracle knowledge about the object location allows us to focus exclusively on the pixelwise prediction within the bounding box, given a specific context of occlusions and the previously predicted segments.

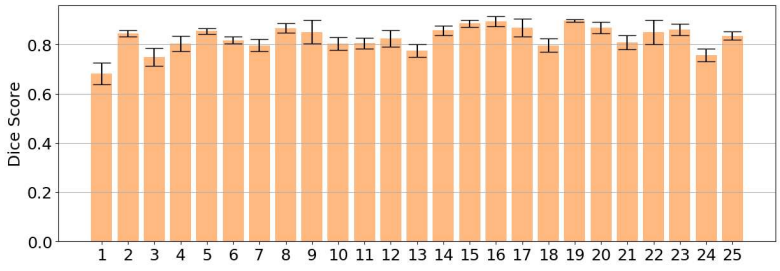
The critic has a similar architecture, but makes use of batch normalisation (BN) [46]. Additionally, one fully-connected layer (FC) from the last convolution is replaced by global maximum pooling. Further details are summarised in Table 7.

We use a vanilla LSTM [44] without peephole connections to represent the hidden state due to its established performance [45] and more economic use of memory compared to convolutional LSTMs [32, 39].

The pre-processing network, which predicts the foreground and the angle quantisation of instances, is an FCN [25] with the same architecture as used by Ren and Zemel [31].

C. Training

To train our actor-critic model, we empirically found that gradually increasing the maximum sequence length leads to faster convergence than training on the complete length



(a) Mean Dice for 25 examples from CVPPP (val)

(b) Example #1

(c) Example #19

Figure 6. *Prediction of ground-truth segments in different order by using oracle bounding boxes centred on each of the target segments: (a)* Mean Dice and standard deviation (error bars) of sequential mask prediction from 20 random permutations of oracle bounding boxes. The best and the worst prediction sequence averaged across the images yield 86.5 and 78.1 Dice, which implies the potential benefit of the optimal prediction ordering. *(b)* Example from CVPPP with the highest standard deviation, *i.e.* which can particularly benefit from the optimal ordering. *(c)* Example from CVPPP with the lowest standard deviation, *i.e.* where ordering of prediction has little effect on Dice. Note how the benefits of choosing an optimal ordering particularly occur in scenes with occlusions.

from the start. Specifically, we train the actor to predict n remaining masks, where n starts with 1 and is gradually extended until the maximum number of instances in the dataset plus the terminations state (21 for CVPPP and 16 for KITTI) is reached. For example, with $n = 1$ the model learns to predict the last instance, *i.e.* the initial state s_1 input to the network is an accumulation of all but one ground-truth masks on a given image, which we choose randomly. We extend n by 5 once the segmentation accuracy on the validation set (either Dice or IoU) stops improving for several epochs.

We decrease the learning rate by a factor of 10 if we observe either no improvement or high variance of the results on the validation set. To trade-off the critic's gradient, we used a constant scalar $\beta_{\text{act}} = 10^{-3}$ for the KL-divergence loss. The actor and critic networks use weight decay regularisation of 10^{-5} and 10^{-4}, respectively. We use Adam [15] for optimisation, as our experiments with SGD lead to considerably slower convergence. To manage training time, we downscale the original images with bilinear interpolation. The ground-truth masks are scaled down using nearest-neighbour interpolation. For CVPPP, we reduce the original resolution from 530×500 to 224×224, and for KITTI from the original 1242×375 to 768×256.

We implemented our method in Lua Torch [43]. Both pre-training and training were performed on a single Nvidia Titan X GPU. The code is available at https://github.com/visinf/acis/.

D. Qualitative Examples

We provide qualitative segment-by-segment visualisations in Figs. 7 and 8 from the CVPPP and KITTI validation sets. Supporting our analysis in the main paper, the order of prediction exhibits a consistent pattern. Furthermore, we observe that our model copes well with inaccuracies of intermediate predictions – a common failure mode of recurrent networks [30].

References

[43] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A Matlab-like environment for machine learning. In *NIPS*2011*. 2

[44] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. 1

[45] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2017. 1

[46] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 1

Section	Type	(Kernel) size	Stride	# of channels
Encoder	Conv → ReLU	3×3	1	32
	MaxPooling	2×2	2	–
	Conv → ReLU	3×3	1	48
	MaxPooling	2×2	2	–
	Conv → ReLU	3×3	1	64
	MaxPooling	2×2	2	–
	Conv → ReLU	3×3	1	96
	MaxPooling	2×2	2	–
	Conv → ReLU	3×3	1	128
	MaxPooling	2×2	2	–
Bottleneck	FC → LeakyReLU	$128 \times h \times w$	–	size(h_t)
	LSTM	size(h_t)	–	size(h_t)
	FC → LeakyReLU	size(h_t)	–	z
	FC (μ, σ)	z	–	$2 \times l$
	FC → LeakyReLU	l	–	z
	FC → LeakyReLU	z	–	$128 \times h \times w$
Decoder	Deconv → ReLU	3×3	2	128 + SP_5
	Deconv → ReLU	3×3	2	96 + SP_4
	Deconv → ReLU	3×3	2	64 + SP_3
	Deconv → ReLU	3×3	2	48 + SP_2
	Deconv → ReLU	3×3	2	32 + SP_1
	Deconv → ReLU	3×3	1	1 + SP_0

Table 5. *Architecture of the actor network.* SP_m denotes the additional channels in the State Pyramid provided at resolution scaled by 2^m.

Dataset	$h \times w$	$size(h_t)$	l	z
CVPPP	7×7	512	16	256
KITTI	8×24	512	64	512

Table 6. *Parameter values for the actor-critic.*

Type	(Kernel) size	Stride	# of output channels
Conv → BN → ReLU	3×3	1	32
MaxPooling	2×2	2	–
Conv → BN → ReLU	3×3	1	64
MaxPooling	2×2	2	–
Conv → BN → ReLU	3×3	1	128
MaxPooling	2×2	2	–
Conv → BN → ReLU	3×3	1	256
MaxPooling	2×2	2	–
Conv → BN → ReLU	3×3	1	512
Global MaxPooling	$h \times w$	1	512
FC → LeakyReLU	512	–	1024
FC → LeakyReLU	1024	–	1024
FC → LeakyReLU	1024	–	512
FC	512	–	1

Table 7. *Architecture of the critic network.*

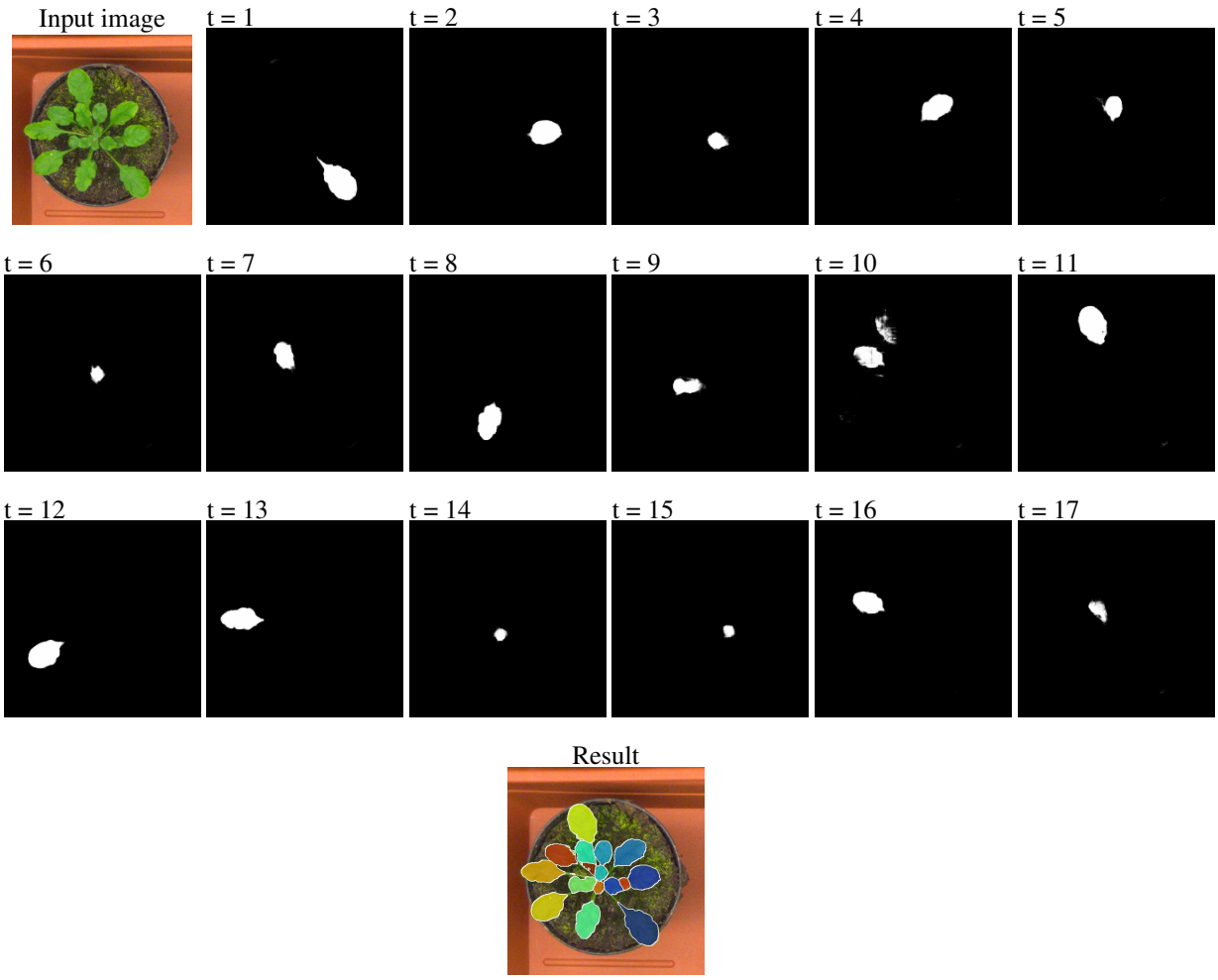


Figure 7. *Visualisation of individual masks as they are predicted at each timestep t on CVPPP val.* Interestingly, our model continued to predict quality masks despite an inaccurate prediction at timestep $t = 10$.

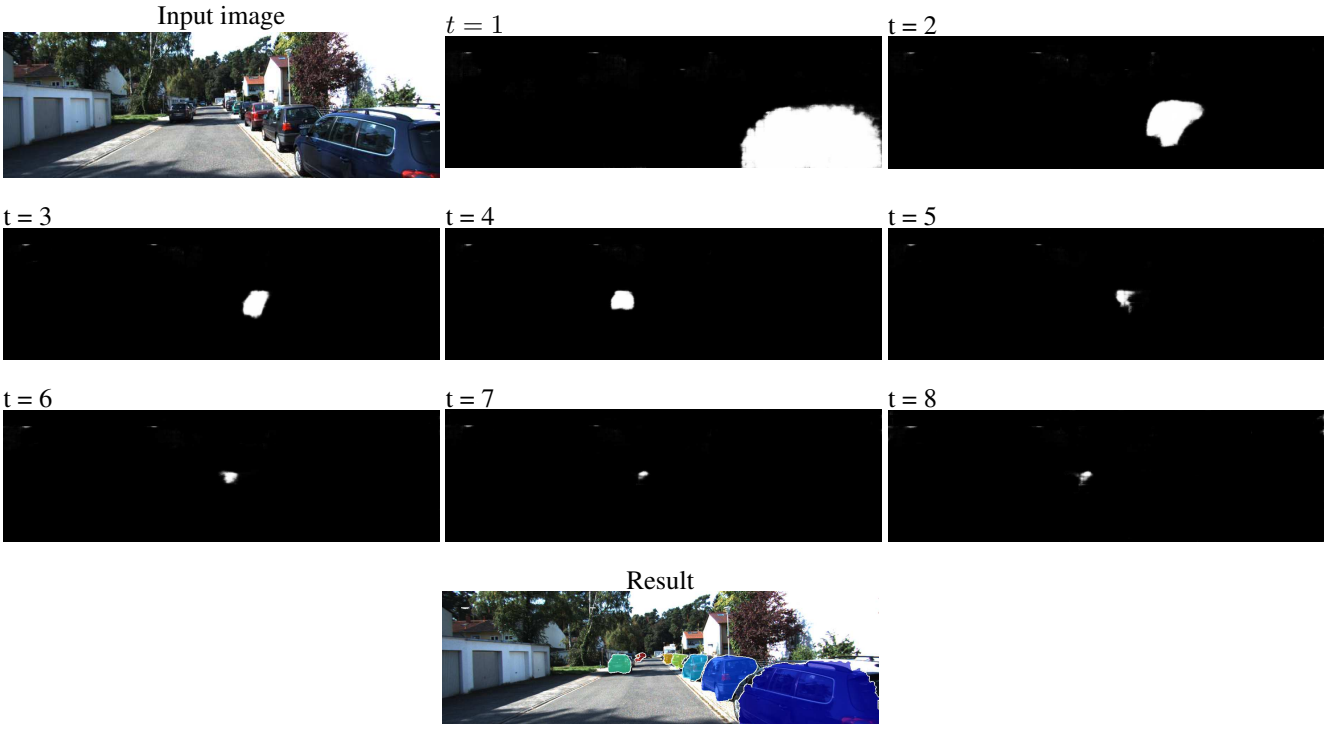


Figure 8. *Visualisation of individual masks as they are predicted at each timestep t on KITTI val.* We observe that the prediction order in this case strongly correlates with the vicinity of the vehicles to the camera.