Stanford | ENGINEERING
Computer Science

Tuning Video Segmentation for Creating Masks of Pedestrians

Quinn McIntyre, Yunqi Richard Gu, Aaryan Singhal {qam, yrichard, aaryan04} @ cs.stanford.edu

Summary

- Given the increasing prevalence of self-driving cars, it is useful to process visual data from vehicles for the prediction and analysis of pedestrian behavior to improve the safety of the vehicles.
- Common approaches in previous works: bounding boxes and image segmentation - while the latter is more computationally expensive, it is a more accurate/precise/coarse segmentation method and is hence more useful for downstream tasks like agent prediction
- Through our project, we want to perform pedestrian segmentation to provide downstream behavior prediction models with more information than they receive otherwise via bounding boxes
- We develop two distinct model pipelines that both take in videos as input and return a video with image segmentation masks around pedestrians
- (1) Use fine-tuned SAM to generate a mask of the first frame in a video clip and then use XMem to propagate the initial mask through the frames in the video
- (2) Generates bounding boxes around every pedestrian in the video using Pedestron and then use the per-frame bounding boxes as input for SAM to generate a video by concatenating pedestrian-segmented frames

Relevant Background

Segment Anything Model (SAM):

- The architecture consists of an image encoder which uses a pre-trained Vision Transformer (ViT) -> a prompt encoder for the possible prompts of SAM (points, boxes, and text) -> and a mask decoder which uses a modification of a transformer decoder block
- SAM can be augmented to perform video segmentation tasks by slicing a video into sequential frame, where segmentation is performed on each frame separately and an output video is generated by stitching the segmented frames

Track Anything Model (TAM):

- An architecture that uses SAM alongside XMem (a video segmentation model) - prompts the user for the object of interest -> uses SAM with the point-prompt to generate a segmentation mask of the desired targets -> provides input to XMem, which propagates the initial segmentation through the video based on the masks of previous frames.
- Performs reasonably well on general tasks, but fails to recognize when new pedestrians enter the frame.

Mask Region-Based Convolutional Neural Network (R-CNN):

- A standard architecture used in image segmentation, Mask R-CNN leverages convolutional layers to classify objects within an image: each pixel in the image is classified and this information is aggregated to produce the desired output.
- Applied to videos in a frame-by-frame manner, but produces lower accuracy masks than SAM and generalizes less well for out-of-domain tasks.

Acknowledgements & References

[1] Zhaowei Cai and Nuno Vasconcelos.
Cascade r-cnn: High quality object detection and instance segmentation, 2019.

[2] Ho Kei Cheng and Alexander G. Schwing.
Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model, 2022.

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick.
Segment anything, 2023.

Methods + Experiments

Fine-Tuning SAM:

- First, we fine-tune SAM to detect humans on a general human dataset (OCHuman dataset). Then, we use the resulting model on a dataset of just pedestrian segmentation (CityScapes dataset).
- Fine-tuning process: use as input the relevant training image and its corresponding bounding boxes, generated for the specified target of interest -> then, to fine-tune, calculate loss against a ground truth segmentation mask.

XMem + SAM:

- Use fine-tuned SAM to generate a precise segmented image for the first frame in the video and then provide that as an input seed to XMem to be propagated throughout the remaining frames in the video

R-CNN + SAM:

- Break the video into still frames, which we feed into Pedestron to generate per-frame bounding boxes -> feed each frame into the fine-tuned SAM, with the per-frame bounding boxes for pedestrians as the prompts -> concatenate the resultant frames to generate video

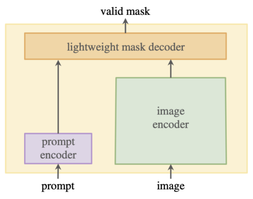

Figure 1. SAM Architecture

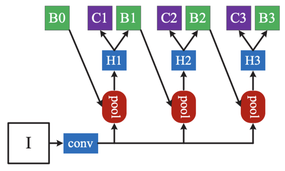

Figure 2. R-CNN Architecture

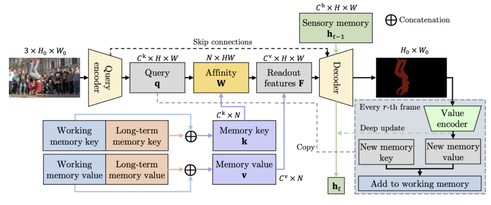

Figure 3. XMem Architecture

Experiments:

1. **Performance of Fine-tuned SAM with Varying Loss Functions:** Common loss functions used for segmentation work = focal loss, dice loss, boundary loss, and BCE loss.
2. **Fine-tuned SAM + XMem:** First frame segmented using SAM and propagated through video using XMem
3. **Fine-tuned SAM + R-CNN:** Each frame bounding-boxed using R-CNN and then segmented using SAM.

Results

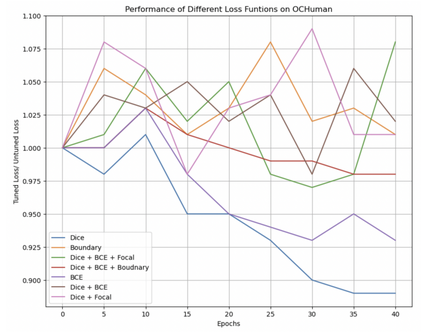

Figure 4. SAM loss function performance on the OCHuman dataset

Figure 5. Success of fine-tuned SAM (left-to-right: tuned, untuned, ground-truth)

Figure 6. Success of fine-tuned SAM (left-to-right: tuned, untuned, ground-truth)

Figure 7. Failure of fine-tuned SAM (left-to-right: tuned, untuned, ground-truth)

Figure 8. R-CNN Bounding Boxes, fed as input to SAM

Figure 9. SAM segmented image, fed as input to XMem

Category	Percentage
Track-Anything Baseline	90%
SAM + XMem	90%
R-CNN + SAM	93%

Table 1. Comparison of different model performances

Category	Count (Percentage)
Track-Anything Baseline	97 (74%)
SAM + XMem	97 (74%)
R-CNN + SAM	122 (93%)

Table 2. Comparison of different model performances

Analysis & Conclusion

- Our experiments found that using only dice loss and dice + BCE + focal are the most suitable approaches to our use-case -> we found that the latter updates the parameters much slower than the former, hence set the loss function of our fine-tuned SAM to dice loss
- We found that SAM + XMem performed identically to our Track Anything baseline, however the former pipeline did generate relatively finer masks - the input/output configuration of the former pipeline also makes it more suitable for use in AVs in the real-world
- SAM + R-CNN took longer to run than SAM + XMem, but produced higher accuracy - furthermore, the former did not fail to track new pedestrians that entered the frame, while the latter did
- **Future work**: Modify the memory structure of XMem so that it can be prompted by new pedestrians entering the frame -> through such work, we believe we can resolve failure to identify new objects entering the frame.