End-to-End Video Instance Segmentation with Transformers

Charlie Arleth and Connor Vines

Problem Overview

Video Instance Segmentation (VIS)

- Normal Object Segmentation only considers a single frame
- VIS - Requires simultaneous classification, segmentation, and tracking

- They compare the problem to Similarity Learning
  - Instance Segmentation ~ learn pixel-level similarity
  - Instance Tracking ~ learn instance-level similarity

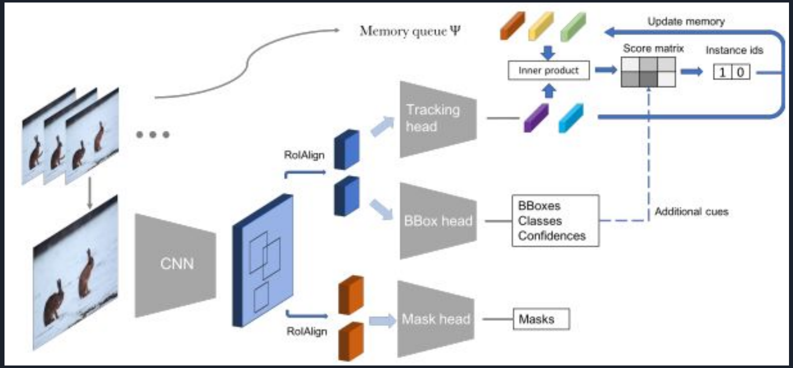
Motivation

- Vast applications
- Complexity
- Bring Transformers into the field of VIS
- Solve the VIS problem in a single framework
- “The framework needs to be simple and achieve strong performance without whistles and bells”

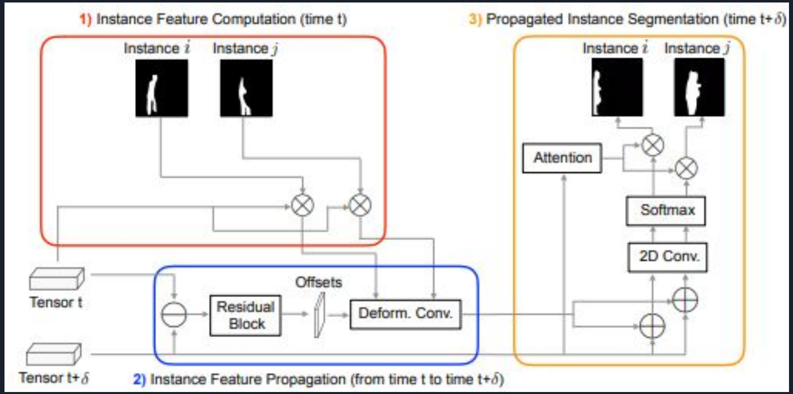
Prior Works

- Top Down - tracking-by-detection
- Bottom Up - separate instances with clustering

- MaskTrack R-CNN
- Maskprop



MaskTrack R-CNN

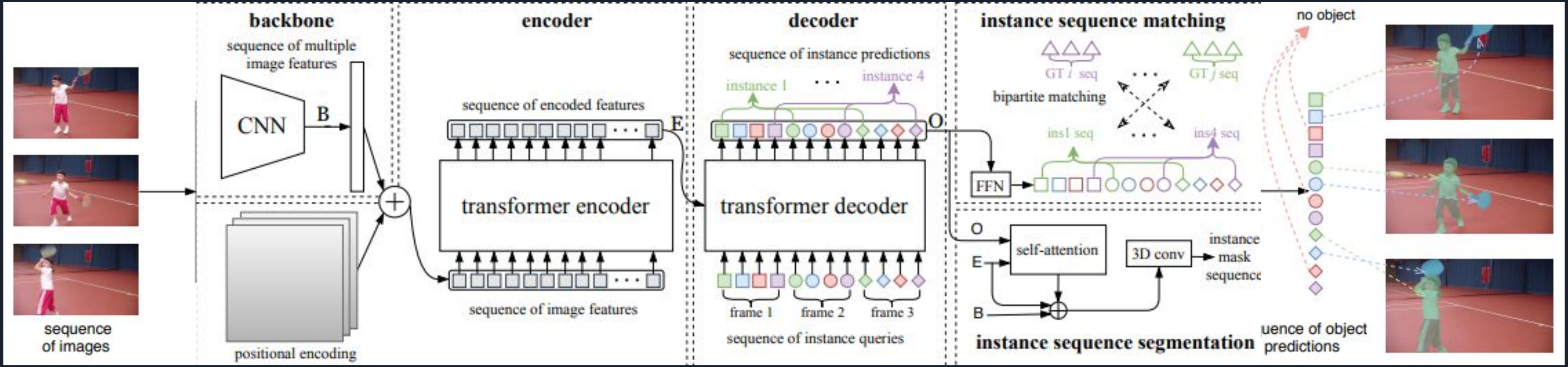


Maskprop

VisTR - Video Instance Segmentation TRansformer

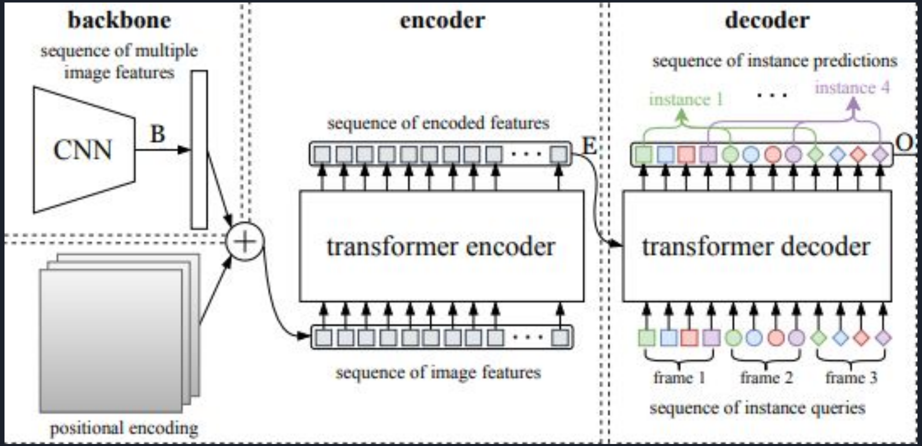
Overview

- Input: Sequence of frames
- Output: Sequence of masks for each instance in the video (in order directly)
- Main Components:
  - CNN Backbone, Transformer Encoder/Decoder, Instance Sequence Matching and Segmentation
- The Hungarian Loss is used to train the whole framework



Backbone and Transformer

- Backbone - extracts pixel-level feature sequence of input video clip
- Temporal and Spatial Positional Encoding
- Transformer - models the similarity of pixel and instance-level features



$$\mathrm{PE}(\mathrm{pos}, i) = \begin{cases} \sin(\mathrm{pos} \cdot \omega_k), & \text{for } i = 2k, \\ \cos(\mathrm{pos} \cdot \omega_k), & \text{for } i = 2k+1; \end{cases}$$

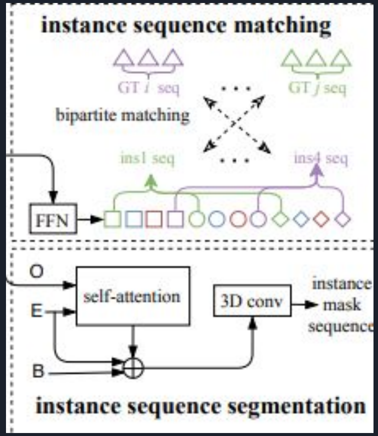
$$\omega_k = 1/10000^{2k/\frac{d}{3}}$$

Challenges

- Overlapping instances
- Changes of relative positions between instance
- Instances in various poses

Two main goals:

- Maintaining output order
    - How to consistently track and segment the same image between frames
    - Addressed with the instance sequence matching strategy
- Obtaining mask sequence for each instance out of the Transformer network
    - Addressed with instance sequence segmentation



Instance Sequence Matching

Tries to maintain the relative positions of objects in video

Instance Sequence Matching:

- Creates a set of n instances, if less than n instances the set is padded with null
- Finds a mapping between predictions and ground truth
- Computes the lowest matching cost between sets (arg min)
- Hungarian loss is the log likelihood + bounding box loss + mask loss

$$\hat{\sigma} = \underset{\sigma \in S_n}{\arg\min} \sum_{i}^{n} \mathcal{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right)$$

$$\mathcal{L}_{\text{Hung}}(y, \hat{y}) = \sum_{i=1}^{N} \Big[\left(-\log \hat{p}_{\hat{\sigma}(i)}(c_i)\right) + \mathcal{L}_{\text{box}}\left(b_i, \hat{b}_{\hat{\sigma}}(i)\right) + \mathcal{L}_{\text{mask}}\left(m_i, \hat{m}_{\hat{\sigma}}(i)\right)\Big]. \quad (7)$$

c_i is ground truth class, b_i is predicted box sequence, m_i is mask features

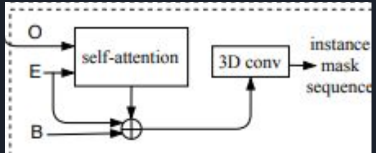
$$\mathcal{L}_{\text{box}}\left(b_i, \hat{b}_{\sigma(i)}\right) = \frac{1}{T} \sum_{t=1}^{T} \Big[\lambda_{\text{iou}} \cdot \mathcal{L}_{\text{iou}}\left(b_{i,t}, \hat{b}_{\sigma(i),t}\right) + \lambda_{\text{L1}} \left\| b_{i,t} - \hat{b}_{\sigma(i),t} \right\|_1 \Big].$$

Instance Sequence Segmentation

Used to predict the mask sequence for each individual instance

Instance Sequence Segmentation:

- Masks are obtained through the similarity map between objects (O) and encoded features (E)
- Attention output is fused with backbone through DETR
- Mask features are put through a 3D convolutional network
- Tensor G_i holds features of instance i across all frames
- Mask loss is computed through *Dice* loss and *Focal* loss (Cross-Entropy)



instance sequence segmentation

$$\mathcal{L}_{\text{mask}}\left(m_i, \hat{m}_{\sigma(i)}\right) = \lambda_{\text{mask}} \frac{1}{T} \sum_{t=1}^{T} \Big[\mathcal{L}_{\text{Dice}}(m_{i,t}, \hat{m}_{\sigma(i),t}) + \mathcal{L}_{\text{Focal}}(m_{i,t}, \hat{m}_{\sigma(i),t}) \Big]. \quad (9)$$

Experiments

Implementation Details

- 8 V100 GPUs - frame sizes downsampled to fit in GPU mem
- 8 Attention heads, 6 encoder and decoder layers
- Assumes a default video length of 36 frames because that's the longest in the YT database
- Model can track 10 objects per frame
- Trained for 18 epochs

Ablations

Length	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
18	29.7	50.4	31.1	29.5	34.4
24	30.5	47.8	33.0	29.5	34.4
30	31.7	53.2	32.8	31.3	36.0
36	33.3	53.4	35.1	33.1	38.5

(a) Video sequence length. The performance improves as the sequence length increases.

	#	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
video level	1	8.4	13.2	9.5	20.0	20.8
frame level	36	13.7	23.3	14.5	30.4	35.1
ins. level	10	32.0	52.8	34.0	31.6	37.2
pred. level	360	33.3	53.4	35.1	33.1	38.5

(b) Instance query embedding. Instance-level query is only 1.3% lower in AP than the prediction-level query with 36× fewer embeddings.

time order	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
random	32.3	52.1	34.3	33.8	37.3
in order	33.3	53.4	35.1	33.1	38.5

(c) Video sequence order. Sequence in time order is 1.0% better in AP than sequence in random order.

	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
w/o	28.4	50.1	29.5	29.6	33.3
w	33.3	53.4	35.1	33.1	38.5

(d) Position encoding. Position encoding brings about 5% AP gains to VisTR.

	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
CNN	32.0	54.5	31.5	31.6	37.7
Transformer	33.3	53.4	35.1	33.1	38.5

(e) CNN-encoded feature vs. Transformer-encoded feature for mask prediction. The transformer improves the feature quality.

	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
w/o	33.3	53.4	35.1	33.1	38.5
w	34.4	55.7	36.5	33.5	38.9

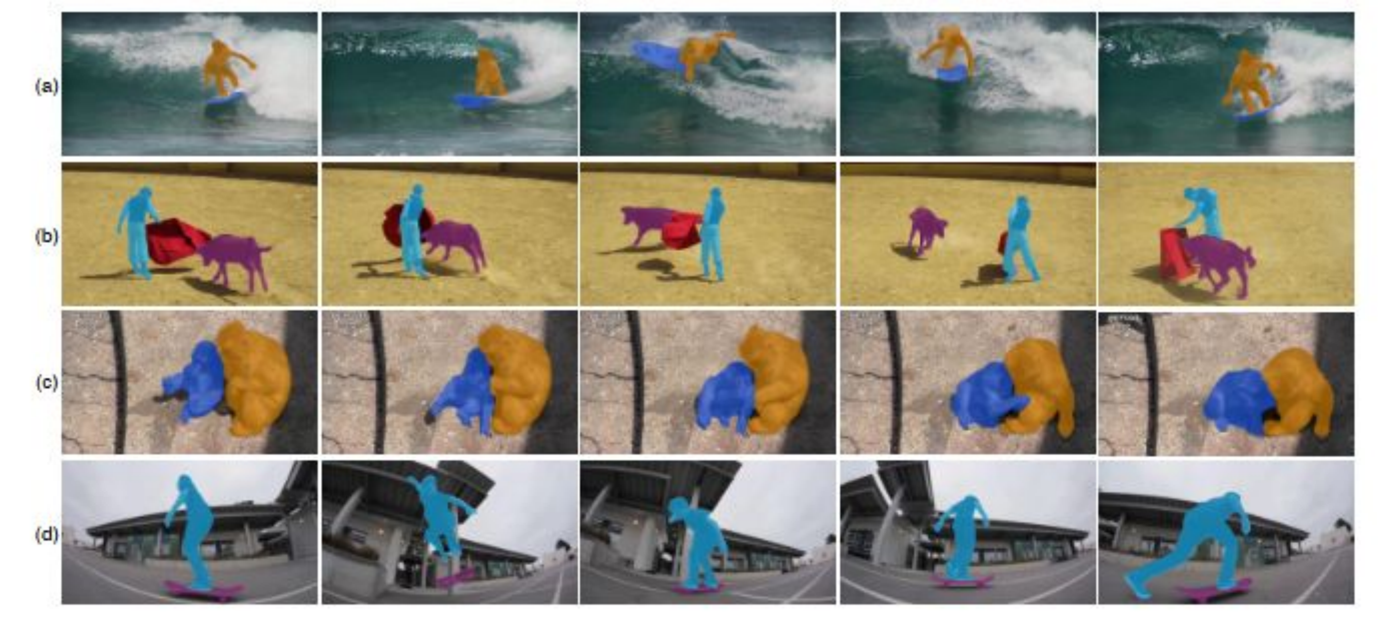
(f) Instance sequence segmentation module. The module with 3D convolutions brings 1.1% AP gains.

Main results

- 40.1% is mask mAP at speed of 57.7 FPS on YouTube-VIS
  - Best/Fastest among all single-model methods
- Achieves fastest speeds even with being slowed by loading image in serial
  - Image loading can be parallelized to achieve a speed more similar to FPS w/o image loading
- MaskProp barely beats it due to combining multiple models
- Simplest version of this model, can be improved

Method	backbone	FPS	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
DeepSORT [28]	ResNet-50	-	26.1	42.9	26.1	27.8	31.3
FEELVOS [24]	ResNet-50	-	26.9	42.0	29.7	29.9	33.4
OSMN [31]	ResNet-50	-	27.5	45.1	29.1	28.6	33.1
MaskTrack R-CNN [30]	ResNet-50	20.0	30.3	51.1	32.6	31.0	35.5
STEm-Seg [1]	ResNet-50	-	30.6	50.7	33.5	31.6	37.1
STEm-Seg [1]	ResNet-101	2.1	34.6	55.8	37.9	34.4	41.6
MaskProp [2]	ResNet-50	-	40.0	-	42.9	-	-
MaskProp [2]	ResNet-101	-	42.5	-	45.6	-	-
VisTR	ResNet-50	30.0/69.9	36.2	59.8	36.9	37.2	42.4
VisTR	ResNet-101	27.7/57.7	40.1	64.0	45.0	38.3	44.9

Visualization (Validation set)



Summary

- Introduces Transformers into a field dominated by Convolutional Methods
- Reshapes VIS task as direct end-to-end parallel sequence decoding/prediction
- SOTA AP and FPS for single model VIS
- Performs well in challenging situations