VisTR

End-to-End Video Instance Segmentation with Transformers

Authors: Yuqing Wang , Zhaoliang Xu , Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, Huaxia Xia Meituan

Presented by: Sourav Karmakar, Mahendra Nandi

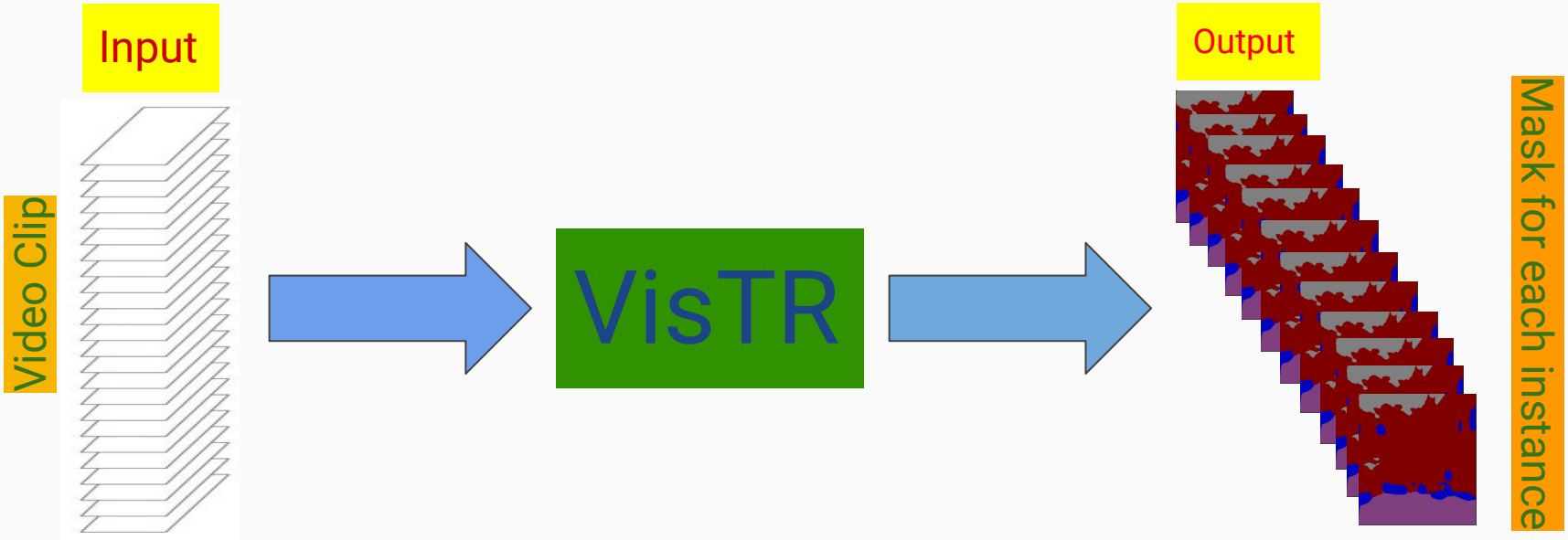
Segmentation

Two types:

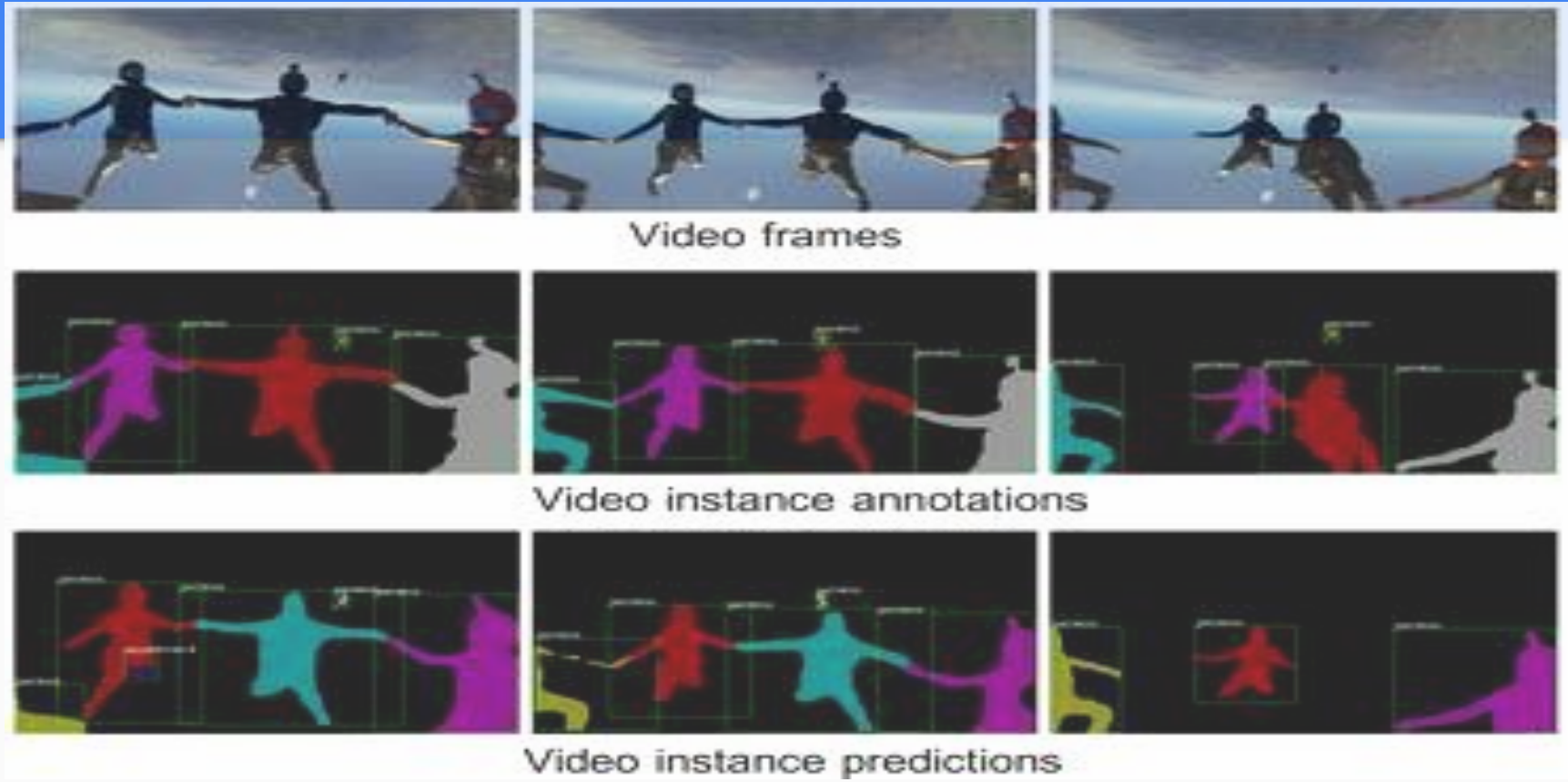
- ➔ Semantic Segmentation
- ➔ Instance Segmentation

Input <<< Video clip consisting multiple image frames

Sequence of mask for each instance in order >>> Output



What is our target ?



Video Instance Segmentation 3 internal steps

Requires simultaneously classifying, segmenting and tracking

- Instance segmentation is to learn the pixel-level similarity
- And instance tracking is to learn the similarity between instances.

Thus, it will be brilliant to solve these two sub-tasks in a single framework and benefit each other

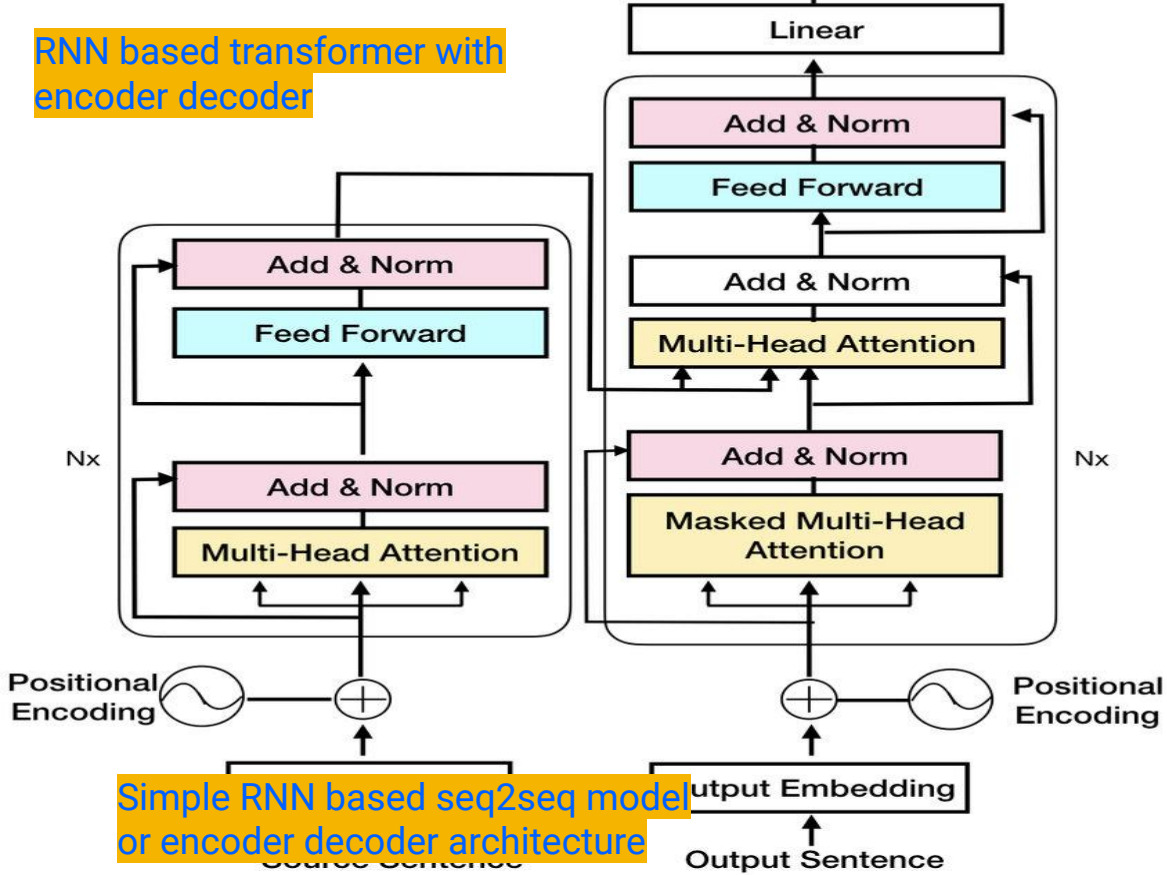
➔ State-of-the-art methods typically develop sophisticated pipelines to tackle it.

Other VIS tasks and models

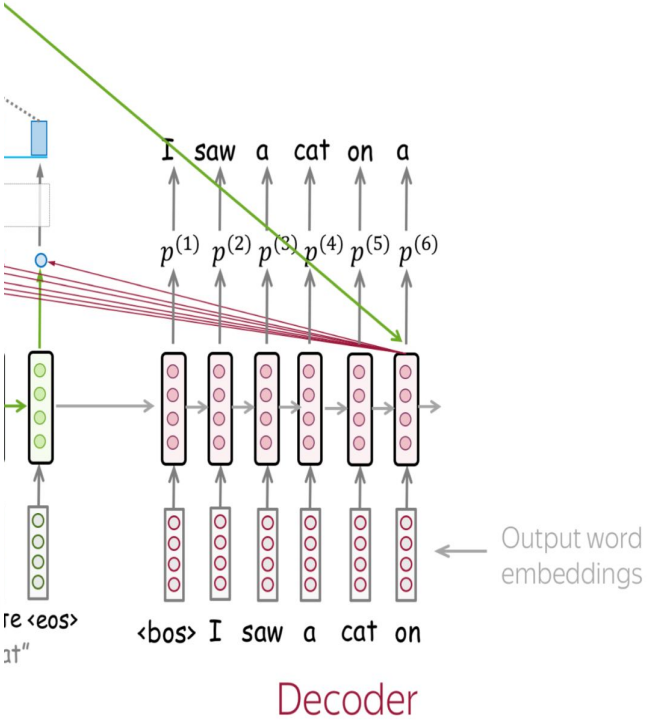
- MaskTrack R-CNN extends the Mask R-CNN with a tracking branch and external memory that saves the features of instances across multiple frames.
- Maskprop builds on the Hybrid Task Cascade Network and re-uses the predicted masks to crop the extracted features, then propagates them temporally to improve the segmentation and tracking.
- STEm-Seg proposes to model video clips as 3D spacetime volumes and then separates object instances by clustering learned embeddings.

- **The above approaches either rely on complex heuristic rules to associate the instances or require multiple steps to generate and optimize the masks iteratively.**
- **In contrast, they aimed to build a simple and end-to-end trainable VIS framework.**

RNN based transformer with encoder decoder

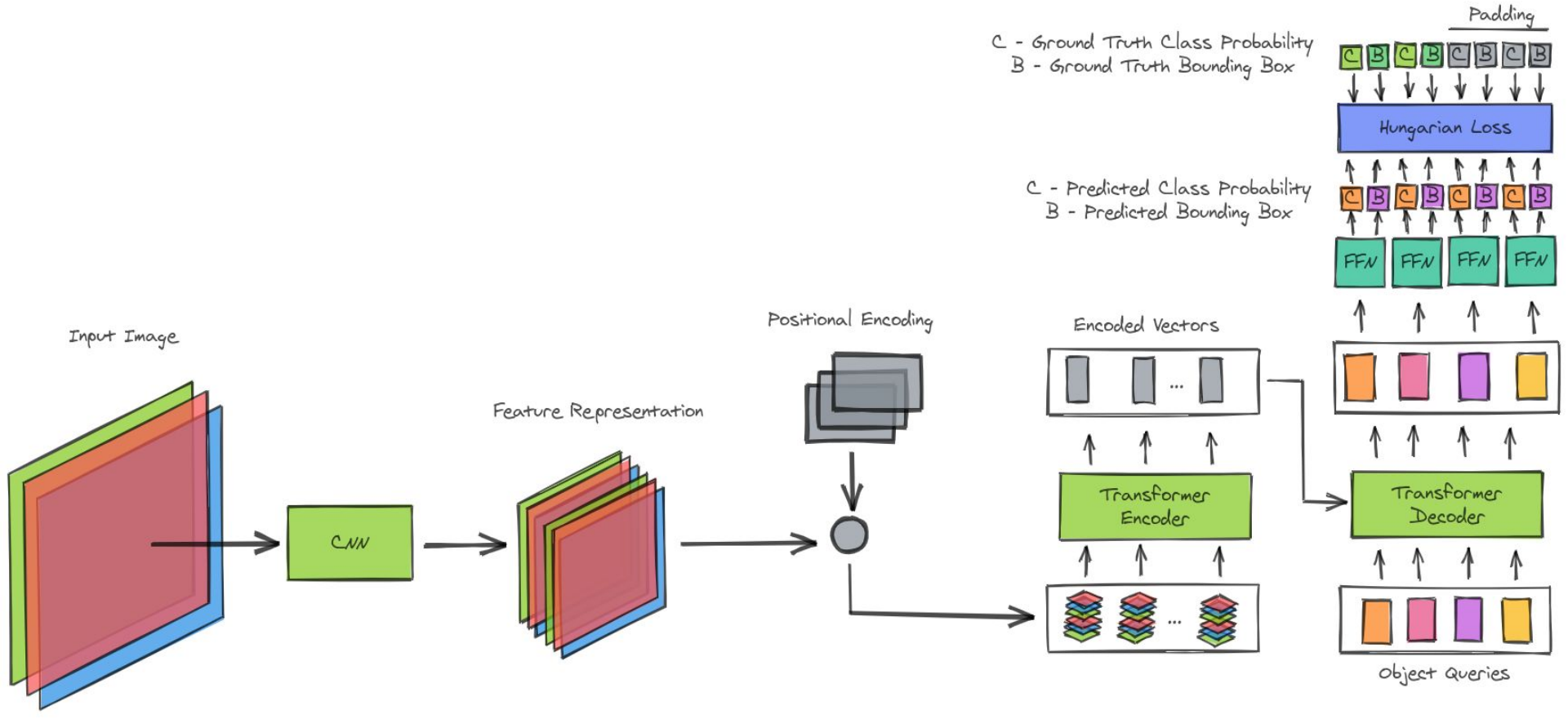


Simple RNN based seq2seq model or encoder decoder architecture

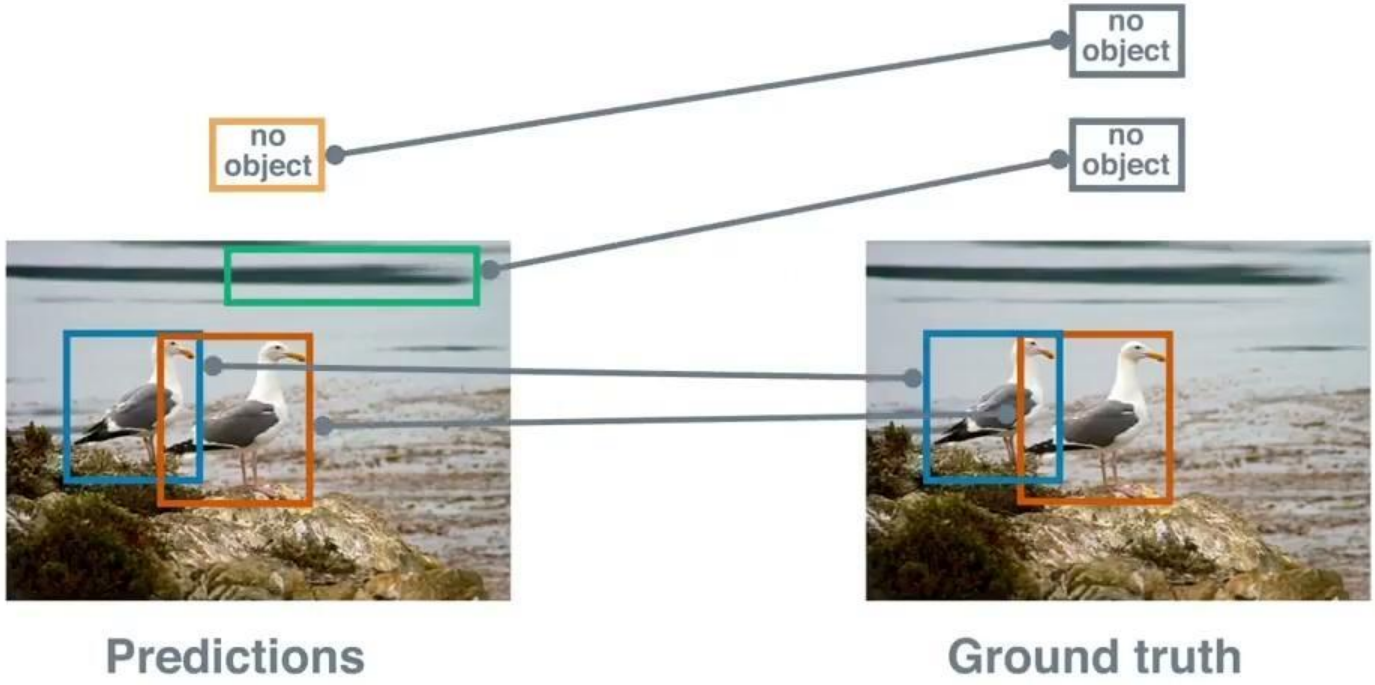


Seq2seq model with attention layer

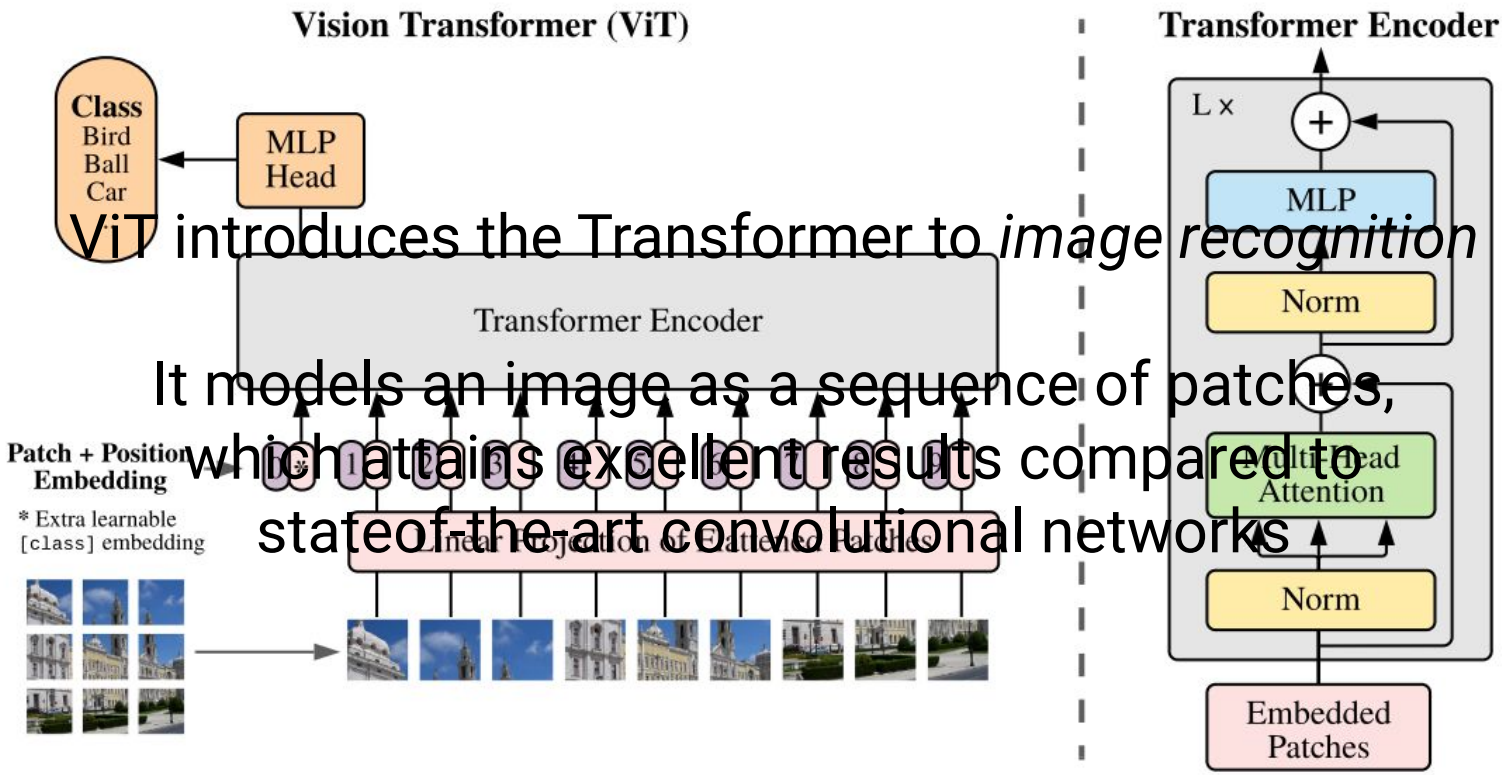
DETR (1st application of transformer in CV)



Architecture overview



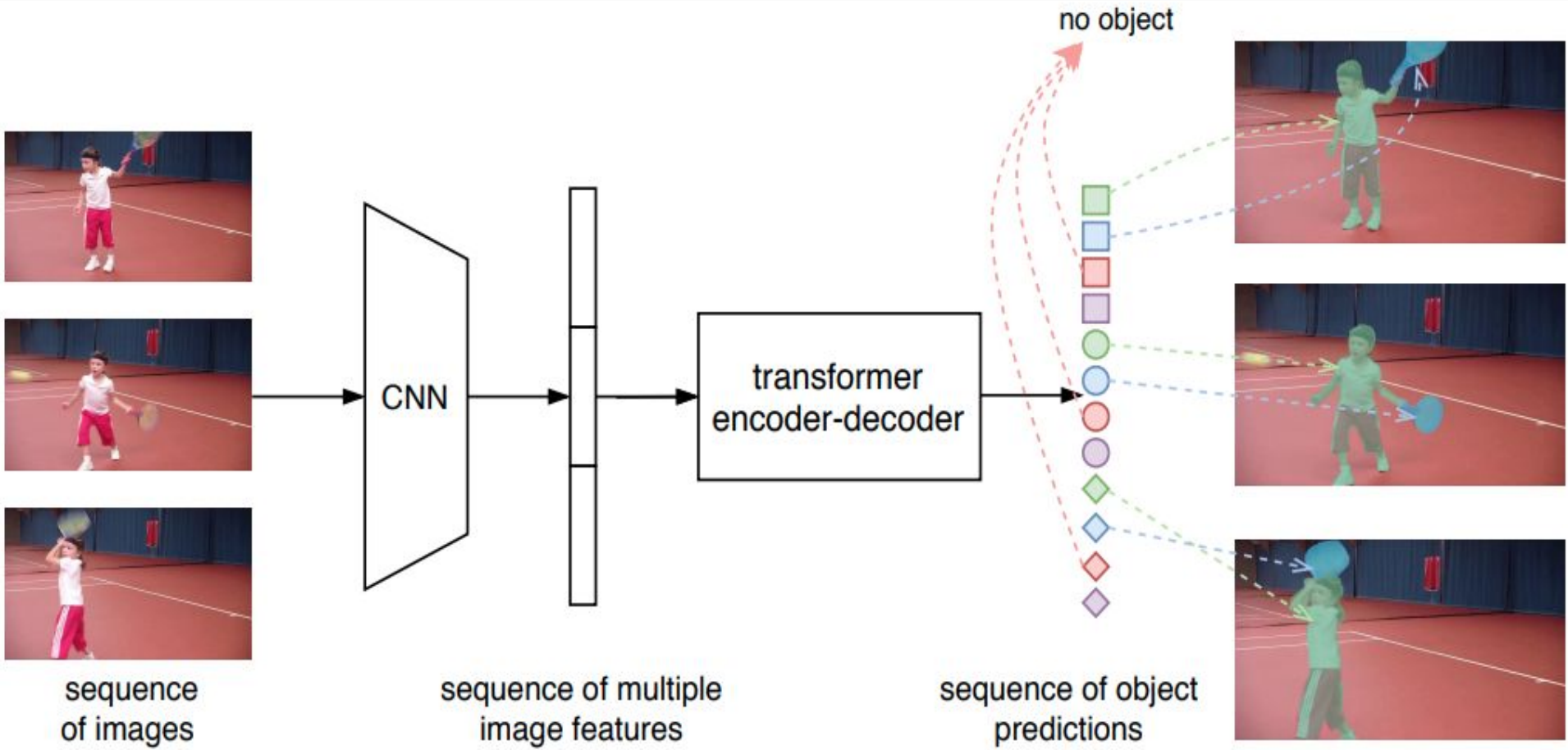
ViT (Vision Transformer)
[another application of transformer]



ViT introduces the Transformer to *image recognition*

It models an image as a sequence of patches, which attains excellent results compared to state-of-the-art convolutional networks

How does VisTR work ?



The two main challenges

1) **How to maintain the order of outputs**

Instance sequence matching strategy

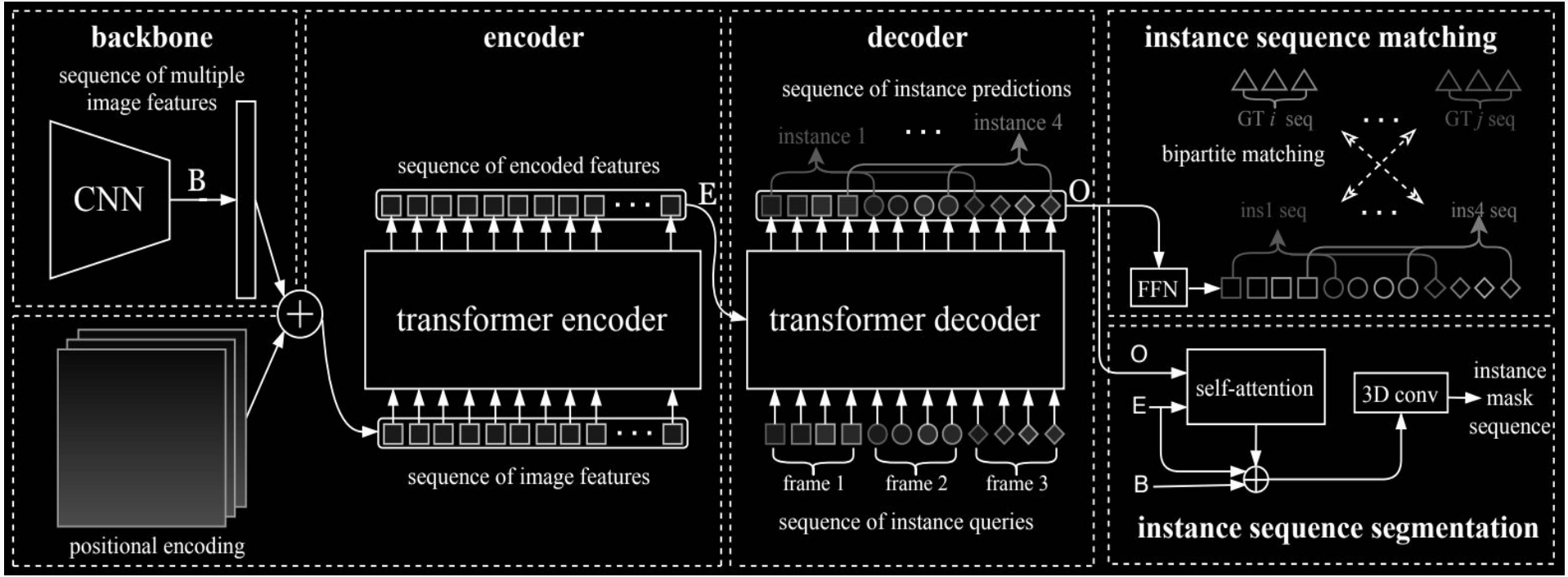
2) **How to obtain the mask sequence for each instance out of the Transformer network**

Instance sequence segmentation module.

VisTR Architecture

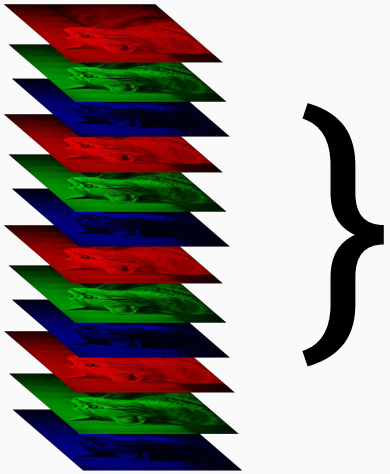
It contains four main components:

- ➔ A CNN backbone to extract compact feature representations of multiple frames
- ➔ An encoder-decoder Transformer to model the similarity of pixel-level and instance-level features
- ➔ An instance sequence matching module for supervising the model
- ➔ And an instance sequence segmentation module



CNN backbone

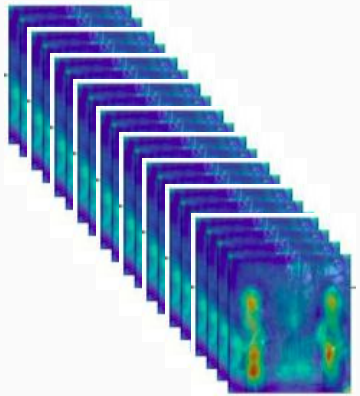
Video Clip



T * 3 * H_o * W_o

CNN

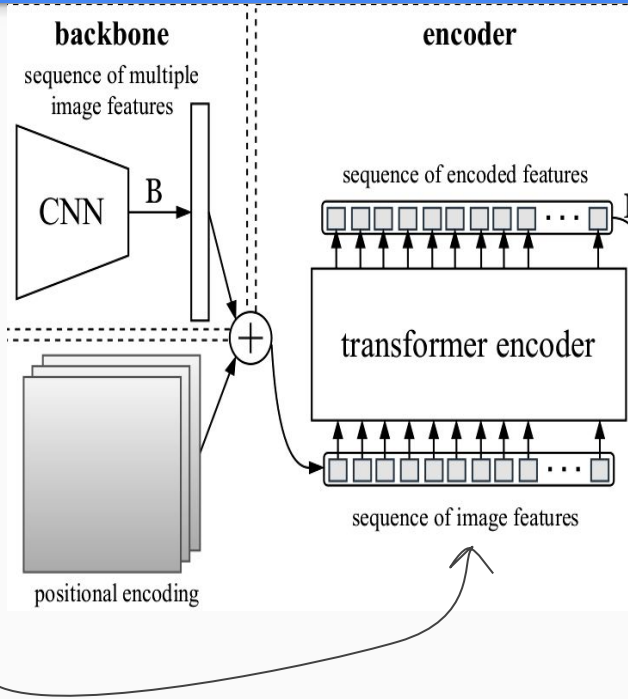
Feature maps for each frame



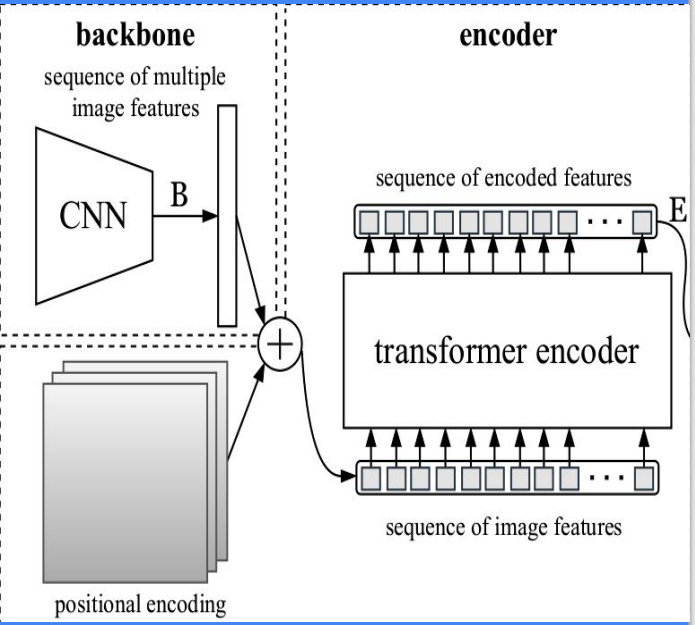
T * C * H* W

Transformer encoder

- 1*1 convolution applied to the feature map
- Dimension reduces from T x C x H x W to T x d x H x W
- Flattening temporal and spatial dimensions with initial temporal order
- Resulting 2D feature map of size d x (T*H*W)
- Each encoder layer has a multi-head self-attention module and a fully connected feed forward network which is brain of the network.



Temporal and Spatial Encoding



- Transformer architecture is permutation invariant
- Segmentation task requires positional information
- Solution to this is we supplement the features with fixed positional encodings information that contains the three dimensional(temporal,horizontal and vertical) positional information.
- Here we adapt positional encoding for our 3D case.
- For the coordinates of each dimension we independently use d/3 sine and cosine functions with different frequencies:

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

where $w_k = 1/10000^{2k/(d/3)}$ and pos is the position in the corresponding dimension.

- The positional encodings of the three dimensions should be concatenated to form the final d channel positional encoding.
- These encodings are added to the input of each attention layer.

Why Transformers?

- ★ Video frames contain richer information than single images [motion patterns and temporal consistency of instances] offering useful cues for instance segmentation, and classification.
- ★ At the same time, the better learned instance features can help tracking of instances

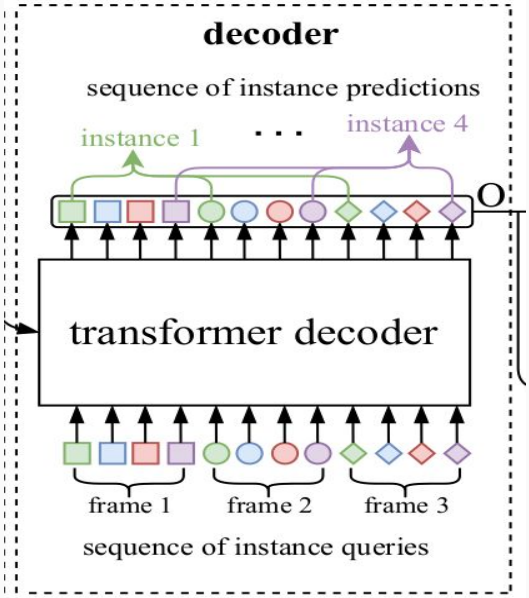
- ❖ Transformers are capable of modeling long-range dependencies, and thus can be naturally applied to video for learning temporal information across multiple frames.
- ❖ The core mechanism of Transformers, self-attention, is designed to learn and update the features based on all pairwise similarities between them. The above characteristics of Transformers make them great candidates for the VIS task.

Decoder

- Decodes the top pixel features that can represent the instances of each frame, which is called instance level feature.
- There are fixed number of input embeddings to query the instance features from pixel features, termed as instance queries.

Suppose that model decodes **n** instances each frame, then for T frames the instance query number is N= n.T

- The instance queries are learned by the model and have the same dimension with the pixel features.
- Taking the output of encoder E and N instance queries Q as input, the transformer decoder outputs N instance features, denoted by O.
- The overall predictions follow the input frame order, the order of instance predictions for different images is the same.
- Thus tracking of instances in different frames could be realized by linking the items of the corresponding indices directly.

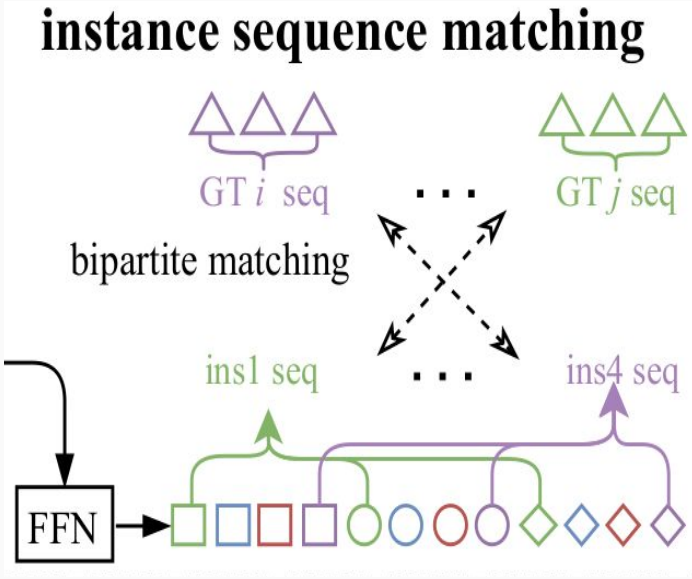


Instance Sequence Matching

- The number of instance sequence is same as decode instances each frame i.e., n.
- Assuming n is larger than the number of instances in the video clip, we consider the ground truth also a set of size n padded with 0.
- In order to find a bipartite graph matching between the two sets we search for a permutation of n elements with lowest cost:

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

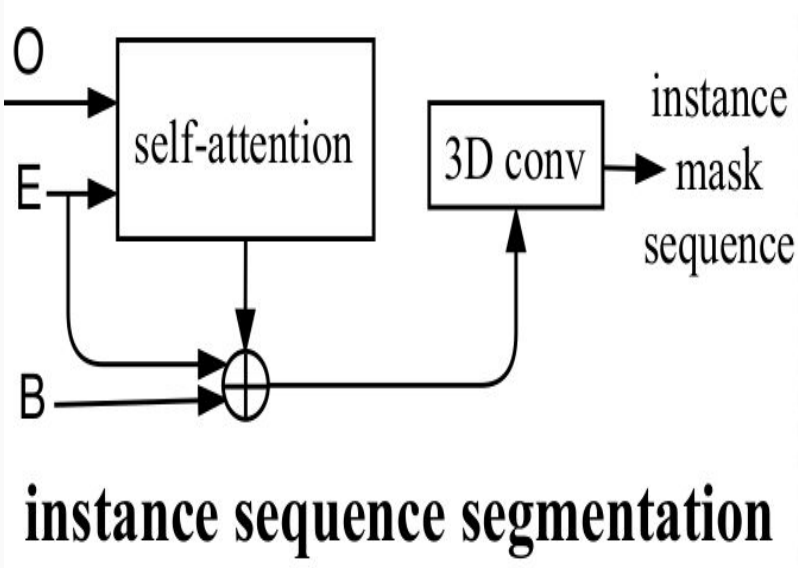
- This optimal assignment could be computed efficiently by the Hungarian algorithm.
- To obtain the box predictions, authors applied a 3-layer feed forward(FFN) network with ReLu activation function and a linear projection layer to the object predictions O of Transformer decoder.



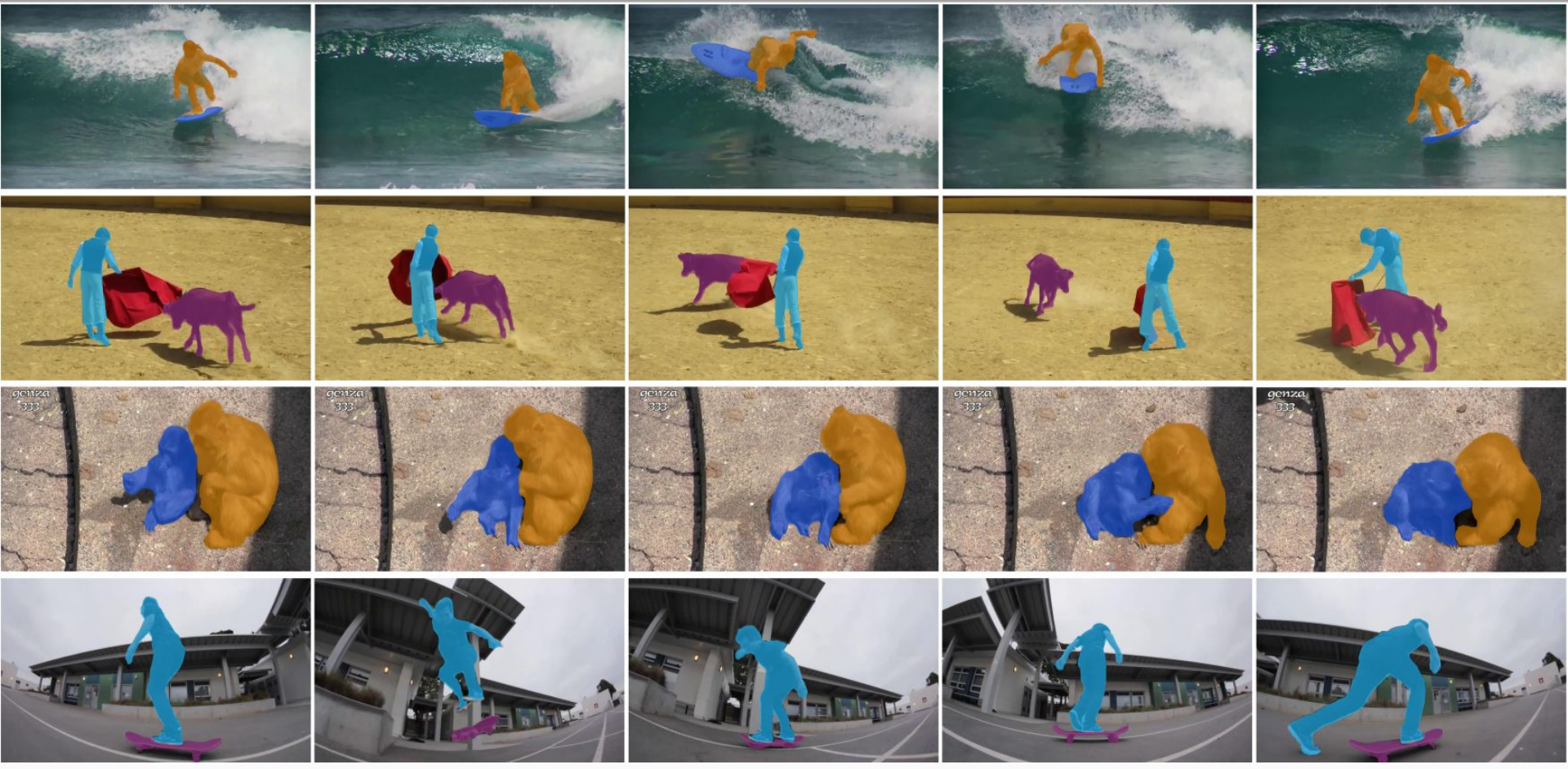
- Authors also add a “background” class to represent that no object is detected and the linear layer predicts the class label using a softmax function.

Instance Sequence Segmentation

- This module aims to predict the mask sequence for each instance
- The mask features are obtained by computing the similarity map between the object predictions O and the Transformer encoded features E.
- To simplify the calculation, we only compute with the features of its corresponding frame for each object prediction.
- For each frame, the object predictions O and the corresponding encoded feature maps E are fed into the self-attention module to obtain the initial attention maps.
- Then the attention map will be fused with initial backbone features B and he tanformed encoded features E of the corresponding frames, following similar practice with the DETR.
- The last layer of the fusion is a deformable convolution layer.



Some Results



Ablation study

Length	AP	AP50	AP75	AR1	AR10
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

Video sequence length:
The performance improves as the sequence length increases.

	AP	AP50	AP75	AR1	AR10
w/o	28.4	50.1	29.5	29.6	33.3
w	33.3	53.4	35.1	33.1	38.5

Position encoding: Position encoding brings about 5% AP gains to VisTR

	AP	AP50	AP75	AR1	AR10
w/o	33.3	53.4	35.1	33.1	38.5
w	34.4	55.7	36.5	33.5	38.9

Instance sequence segmentation module: The module with 3D convolutions brings 1.1% AP gains.

Thank You