2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)

2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI) | 978-1-6654-3881-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CEI52496.2021.9574462

Transformer in Computer Vision

Jiarui Bi†
Faculty of Electrical Engineering and Computer Science
University of California, Irvine
92697, Irvine, California, United States
Jiaruib@uci.edu

Qinglong Meng†
Faculty of Radiology
Shandong First Medical University
271000, Shandong, China
mqlforce@163.com

Zengliang Zhu†, *
Faculty of Information Technology
Macau University of Science and Technology
Macau, China
* Corresponding author: 1909853si011001@student.must.edu.mo
†These authors contributed equally.

***Abstract*—Transformer is widely used in Natural Language Processing (NLP), in which numerous papers have been proposed. Recently, the transformer has been borrowed for many computer vision tasks. However, there are few papers to give a comprehensive survey on the vision-based transformer. To this end, we give an in-depth review of the vision-based transformer. We conclude 15 articles covering transformers on image object detection, multiple object tracking, action classification, and visual segmentation. Furthermore, we summarize 6 related datasets for corresponding tasks as well as their metrics. We also provide a comprehensive experimental comparison to validate the strength of transformer-based methods. We provide a brief introduction to the transformer and its applications on computer vision tasks, which can help beginners to recognize it.**

Keywords-component; Transformer network; Object detection; Multiple object tracking; Action classification; Visual segmentation; Transformer survey

I. INTRODUCTION

In 2017, Vaswani et al. from google proposed a later well-renowned paper entitled Attention Is All You Need [1], giving birth to the first transformer architecture. Although it was designed for natural language processing (nlp), this architecture has also proven effective in computer vision. Studies of vision transformers have become so popular within the last few months that some people call the trend "The vision transformer explosion".

What is a transformer? A Transformer is a sequence-to-sequence (input a sequence and out another) deep learning model based on attention operation, which means that it focuses on various parts of an input sequence depending on various objects in the query list. As mention above, this model was initially designed for sequence modeling in natural language processing tasks. Traditionally, these tasks were handled using recurrent neural networks (RNN), which is shown in Figure 1, such as Long short-term memory (LSTM) and Gated recurrent unit network (GRU).

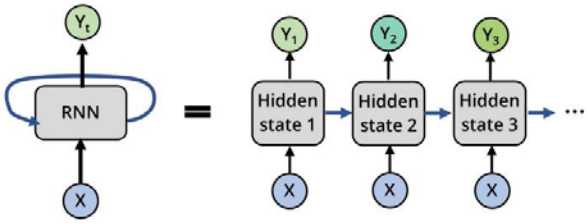

Figure 1. Conventional Recurrent Neural Network.

These RNN-based methods were trained sequentially, which means that words in a sentence are generated by checking the memory of previous generations (or hidden states) during training. The RNN-based designs come with two significant flaws. First, they have difficulty modeling relations between distant tokens (e.g., words from different parts of a sentence) because the long-term memories in hidden states tend to fade and change through iterations. Second, the time complexity of training grows with the length of the input sequence, resulting from the fact that an output token has to wait for the former ones. The transformer addressed these issues through "Scaled Dot-Product Attention", which is illustrated in Figure 2.

What attention operation does is that it transforms input vectors (Ys and Xs in Figure 2) into corresponding query vectors "Q", key vectors "K", and value vector "V". An intuitive illustration for these vectors is that one query vector is like the words you type into a web browser, key vectors are like the words in web pages, and value vectors are like the links of web pages. In attention operations, query objects (Ys in Figure 2) use query vectors to match the key vectors of non-query objects (Xs in Figure2). The value vectors of non-query objects will be taken and summed up accordingly concerning how well their key vectors match the query vector (Attention score in Figure 2). The summed vectors will be the final output of the attention block.

978-0-7381-4649-2/21/$31.00 ©2021 IEEE

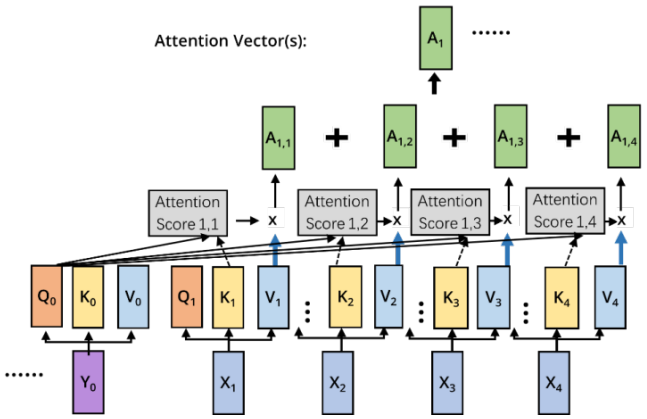

Figure 2. An illustration of Dot-Product Attention. Note that when Q is taken from X vectors instead of Y, this forms a special case called "Self-Attention".

It turns out that dot products of matrices can represent the operation mentioned above:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

In this function, d_k means the dimension of key and query vector, while *softmax* function normalizes the arguments and maps them into an interval of [0,1].

The transformer architectures rely a lot on attention operations, as is shown in Figure 3.

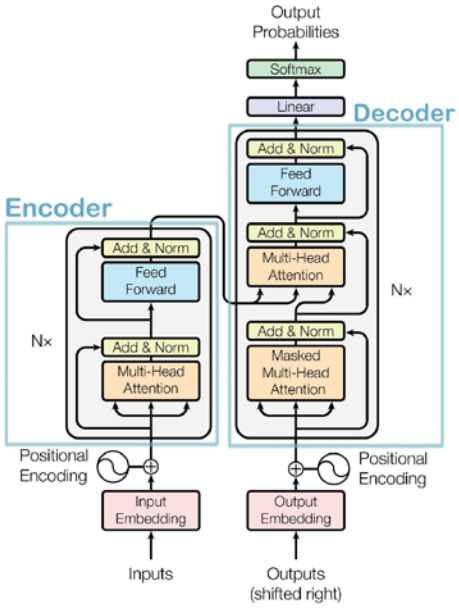

Figure 3. The original transformer architecture.

The conventional transformer can be divided into two parts. The left is called the encoder, which uses Multi-head self-attention to model the relation between the tokens within the input sequence. The one on the right is called decoder, which performs attention on the encoder's output (Xs in figure 2) using embedded query objects (Ys in figure 2). The query object can be previous outputs for NLP tasks (autoregressive) or a learned anchor-like matrix for some vision transformers like DETR (non-autoregressive). What is worth mentioning is that some vision transformers built for classifying tasks only include an encoder and exclude the decoder. This is because the encoder alone can summarize the whole input sequence by adding a learnable classification token [CLS] in the inputs.

Transformers have been adopted to replace several conventional methods in the fields of computer vision. For instance, Vision Transformer (ViT) [2] managed to replace convolutional network (CNN), Detection Transformer (DETR) managed to replace anchor and non-maximum suppression, and Video Vision Transformer (ViViT) managed to replace 3D convolution, etc. Using transformers in computer vision tasks can bring several benefits. First, transformers have less inductive bias, which means that the models have fewer assumptions on what method will work. This leads to potentially better performance. For instance, Fast-RCNN uses a manually designed anchor mechanism to predict the center of the bounding boxes. However, in Detection Transformer (DETR), it is replaced by learned objected queries, which will be feed into decoders. While both training on the large-scale dataset, the transformer can outperform the CNN-based methods, shown in Figure 4.

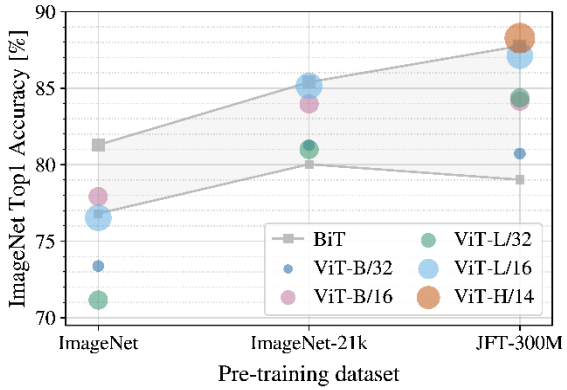

Figure 4. Comparison between ViT and CNN-based BiT (Big Transfer) ResNet on accuracy using small-scale and large-scale training set.

Second, transformers have bigger respective fields. In CNN, we assume that a hierarchy of local features can be good at representing an image. The truth is that it does work in a quite efficient way but at the cost of limited respective fields. The convolution-based methods can only learn to grasp the "big picture" at comparatively deep layers of the network, making it different to train these abilities and limit the models' performance. Transformers, however, replace sliding windows with learnable global weights, which is provided by the attention mechanism. As shown in Figure 4, ViT can focus on high-level features at the beginning of the training regime, making it powerful not only in both image and video understanding.

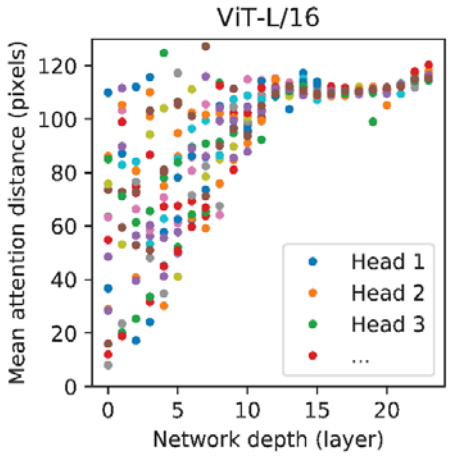

Figure 5. Mean attention distance for each layer (ViT).

This paper gives an in-depth survey on the vision-based transformer, which concluded several transformer-based methods on four fields of computer vision and included 15 papers. We also put together their performance on 6 datasets, including COCO2017, MOT17, Kinect, Something-Something V2, Cityscape, YouTube-VIS, and listed their evaluating indicators. We also analyze their performance for specific applications, which can be used to handle particular problems. We hope our work can provide beginners with a rough idea of transformers and their applications in different vision fields.

II. Review on Vision Transformer

A. *Transformers on image object detection*

DETR. Nicolas Carion et al. [1] present a new method that views object detection as a direct set prediction problem. This approach streamlines the detection pipeline, effectively removing. The main ingredients of the new framework, called DEtection TRansformer or DETR, are a set-based global loss that forces unique predictions via bipartite matching and a transformer encoder-decoder architecture.

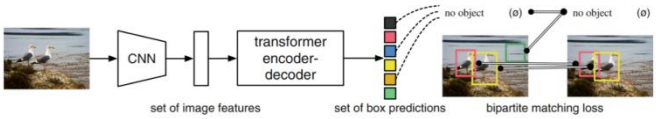

Figure 6. The proposed DETR framework.

DEtection TRansformer (DETR, shown in Figure 6) predicts all objects at once and is trained end-to-end with a set of loss function which performs bipartite matching between predicted and ground-truth objects. Compared to most previous works, the main features of DETR are the conjunction of the bipartite matching loss and transformers with (non-autoregressive) parallel decoding. And DETR demonstrates significantly better performance on large objects. However, lower performances on small objects. The first author et al. Present a new method that views object detection as a direct set prediction problem. This approach streamlines the detection pipeline.

Deformable DETR. DETR suffers from slow convergence and limited feature spatial resolution due to the limitation of Transformer attention modules in processing image feature maps. Deformable DETR can solve these problems. In this paper [2], Deformable DETR was proposed, which mitigates the slow convergence and high complexity issues of DETR. They propose the deformable attention module, which can be naturally extended to aggregating multi-scale features, without the help of FPN(Feature pyramid networks). In DETR, Multi-scale features are usually used to solve small targets which are not friendly to small targets. However, high-resolution feature maps greatly increase the complexity of DETR, and the training cycle is long, and DETR is 10-20 times slower than Faster RCNN.

Compared with DETR, for each query, now Deformable DETR only pays attention to the positions considered to contain more local information by more meaningful networks to alleviate the problem of large computation caused by large feature graphs. Secondly, the Deformable Attention Module is extended into a multi-scale feature map to solve small objects problems.

Up-DETR. Inspired by the great success of pre-training transformers in natural language processing, Zhigang Dai1et.al [5] propose a pretext task named random query patch detection to Unsupervised Pre-train DETR (UP-DETR) for object detection. Specifically, Zhigang Dai1 et al. randomly crop patches from the given image and then feed them as queries to the decoder: the model is pre-trained to detect these query patches from the original image. After the experiment, UP-DETR transfers well with state-of-the-art performance on one-shot detection and panoptic segmentation. In ablations, Zhigang Dai1 et al. find that freezing the pre-training CNN backbone is the most important procedure to preserve the feature discrimination during the pre-training. Unlike DETR, to trade off classification and localization preferences in the pretext task, UP-DETR freezes the CNN backbone and proposes a patch feature reconstruction branch that is jointly optimized with patch detection.

B. *Transformers on multiple object tracking*

TrackFormer. Tim Meinhardt et al. [6] present TrackFormer, an end-to-end multi-object tracking and segmentation model based on an encoder-decoder Transformer architecture. This approach introduces track query embeddings which follow objects through a video sequence in an autoregressive fashion. Their approach includes the novel concept of track queries that follow an object in space and time over the course of a video sequence in an autoregressive fashion. It achieves track association implicitly with attention and requires no additional matching, optimization, or modeling of motion and appearance.

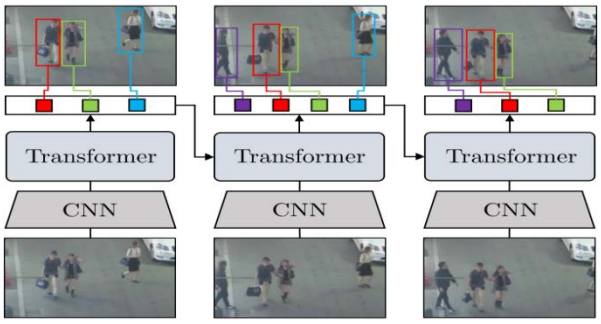

Figure 7. Framework of TrackFormer [6].

TransTrack. The query-key mechanism in single-object tracking(SOT), which tracks the object of the current frame by object feature of the previous frame, has great potential to set up a simple joint-detection-and-tracking MOT (Multiple-Object Tracking) paradigm. Nonetheless, the query-key method is seldom studied due to its inability to detect newcoming objects.

In this work, Peize Sun et al. [7] propose TransTrack, a baseline for MOT with Transformer. It takes advantage of the query-key mechanism and introduces learned object queries into the pipeline to detect new-coming objects. TransTrack has three main advantages: (1) It is an online joint-detection-and-tracking pipeline based on a query-key mechanism. Complex and multi-step components in the previous methods are simplified. (2) It is a brand-new architecture based on Transformer. The learned object query detects objects in the current frame. The object feature query from the previous frame associates those current objects with the previous ones. (3) For the first time. Peize Sun et al. demonstrate a much simple and effective method based on query-key mechanism and Transformer architecture that could achieve a competitive 65.8% MOTA on the MOT17 challenge dataset.

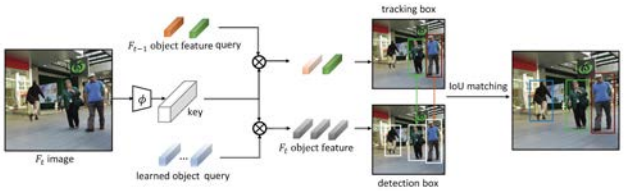

Figure 8. Framework of TransTrack [7].

C. *Transformers on action classification*

ActionTransformer. Rohit Girdhar et al. [8] developed a method to recognize the action of a person in each video clip base on their interaction with other people in the scene. First, they used a region proposal network (RPN) to locate the human targets and the initial layer of the Inflated 3D convolutional (I3D) model to extract frame-level information. After that, they fed the feature maps generated from the I3D backbone into their Action Transformer and attended the located human targets to each feature map. In this way, the model will analyze the interaction between human targets and their environment. This approach is limited to human action classification.

VTN. In this paper, Neimark et al. [9] proposed a transformer-based model for understanding a long video with multiple modals. It comprises three parts: a 2D spatial feature extraction model (that can be any off-the-shelf one), an encoder with Longformer self-attention model, and an MLP head for classification. The transformer does not include a decoder and will take the [CLS]-related state as the input of the Multilayer Perceptron.

The Longformer model, or long document transformer, was initially designed to process large-scale text sequences. It adopted a method similar to convolution called sliding window attention, limiting the receptive field of the attention process so that each token can only attend to the ones near them.

The benefit of using Longformer is that it significantly reduces the computational cost when processing long video clips. However, this approach is a trade-off between computational cost and accuracy since it provides an indirect approach for long-term modeling relations.

Experiment on the Kinetics dataset shows that the combination of ViT model and Longformer can achieve impressive accuracy in action classification, making it the first pure transformer-based action classifier. It is also worth noticing that the Kinetics dataset contains only short videos, not what this model is designed for. Unfortunately, since there is no long video dataset available, it is unclear how well the model will perform long video footage.

A^2-Nets. Chen et al. [10] proposed a "double attention block" integrated into existing architecture. This paper is believed to be the first trial of using the attention-based model in video classification tasks. To resolve the inefficiency of CNN models on capturing long-range relations, the authors developed an approach to model the correlation of two arbitrary feature maps from one or multiple frames. The model is illustrated in Figure 9.

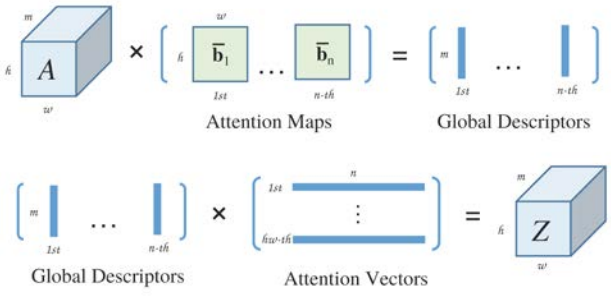

Figure 9. Double Attention Networks.

Similar to the self-attention approach using a query, key, and value attributes (but with slightly different ways of using softmax function), the model performs outer product of two layers to produce a "descriptor" matrix. Again performs the outer product of the "descriptor" matrix with the set feature vectors from each layer. In this way, the classifier will attend to different features from a long sequence of frames. Experiments show that this module can achieve a state-of-the-art result on the Kinetics dataset, which is quite impressive at its time.

TimeSformer. Inspired by ViT, Bertasius et al. [11] developed a complete transformer-based video classification model that includes no convolutional operation, and they call their model 'TimeSformer'. Experimenting with various time-space transformer designs concluded that the "Divided Space-Time Attention" model (shown in Figure 10) exhibits the best outcome regarding the accuracy and computational cost. This "Divided Space-Time Attention" model first performs self-attention on all the patches at the same special location in the temporal domain. The encoded result will attend to each patch on their respective frames in a spatial manner. Experiments suggest that the TimeSformer model outperforms all traditional action detectors on Kinetics-400, Kinetics-600, Diving-48, and HowTo100M, especially in long-term video modeling, which makes it one of the best long-term video action detector in the field. In addition, this outcome can be achieved with less training time comparing to 3D CNNs like SlowFast and I3D. It is also worth mentioning that the model did not achieve a state-of-the-art result on a something-something v2 training set, indicating that there remains more dedicated effort to improve its performance temporally-heavy datasets.

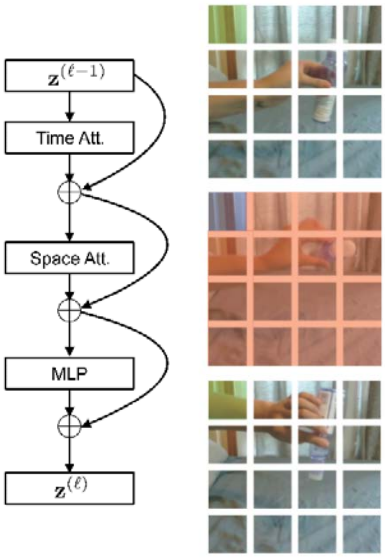

Figure 10. Divided Space-Time Attention. The query patch is denoted in blue. Time-attention neighbors are denoted in green. Space-attention neighbors are denoted in red.

STAM. Sharir et al. [12] proposed another pure transformer-based model named Space Time Attention Model, or STAM, similar to the TimeSformer model aforementioned. However, there is one thing that the STAM model differs from TimeSformer. As is illustrated in Figure 11, the temporal transformer only takes the first embedding vector from each frame, instead of all the outputs in TimeSformer's case. This approach helps reduce the tokens needed for temporal self-attention, which led to lower computational cost. The issue of this model is that although it managed to match convolution-based methods on the accuracy, it failed to excel the existing top-record. As a result, the issue of how to design a transformer that is both efficient and effective is still on the table.

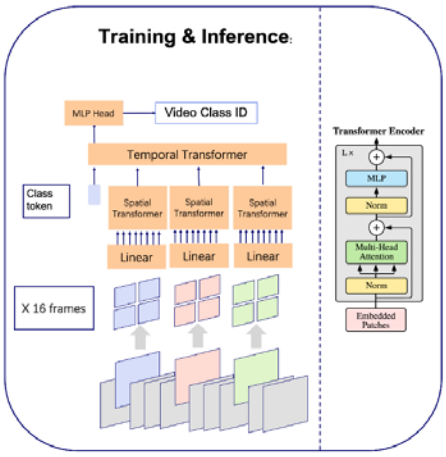

Figure 11. Space Time Attention Model.

ViViT. Several weeks after the release of the world's first complete transformer-based video classifier, known as 'TimeSformer', Arnab et al. [13] brought out another pure transformer-based architecture for video classification, named 'Video Vision Transformer' or 'ViViT'. Their new model outperformed both traditional CNN-related methods and the 'TimeSformer' model on the 'Kinetics' dataset and achieved state-of-the-art results on the Something-Something v2 dataset, which is something that 'TimeSformer' has failed to manage. There are several major ways in which ViViT differs from TimeSformer. First, they used different ways to embed a video. Apart from TimeSformer, which embeds each frame independently, ViViT patches a video jointly in the spatial and temporal domain. Doing so, separate a video into smaller spatio-temporal "capsules" or "tubes", to think intuitively. They then embed each "capsules" and concatenate the tokens together, as is shown in Figure 12.

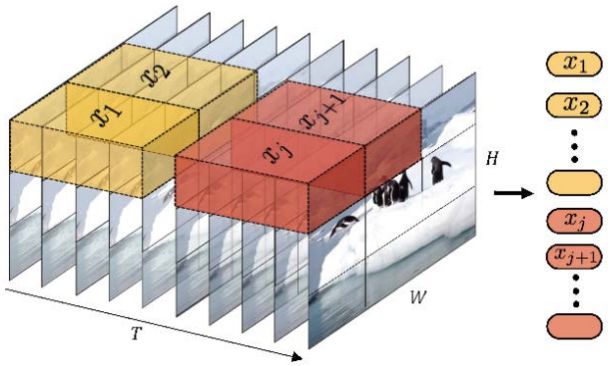

Figure 12. Tubelet embedding

When training on large datasets like Kinetics and Moments in time, the authors combined this Tubelet embedding with Joint Space-Time self-attention, which simply feeds all the tokens into the transformer encoder. It turned out that this combination can achieve a state-of-the-art result on a large dataset when spatially downsizing the "Tubelet" and thus increasing spatial resolution. Second, the authors proposed a new encoder structure called Factorised encoder, which turned out to be surprisingly effective on smaller datasets like Epic Kitchens and Something-Something v2. The Factorised

encoder is more like a tweaked version of TimeSformer's Divided Space-Time attention model. It first feeds the tokens from each frame into a ViT encoder for special self-attention, similar to the first spatial encoder block of TimeSformer but with an additional classification prepended to each frame-level token. Then, instead of feeding the whole produced sequence into the temporal attention module, ViViT takes the first token of each frame-level produced sequence and feeds into the temporal self-attention module. It follows the architecture of the BERT model developed for natural language processing tasks. The architecture of ViViT's factorised encoder model is shown in Figure 13.

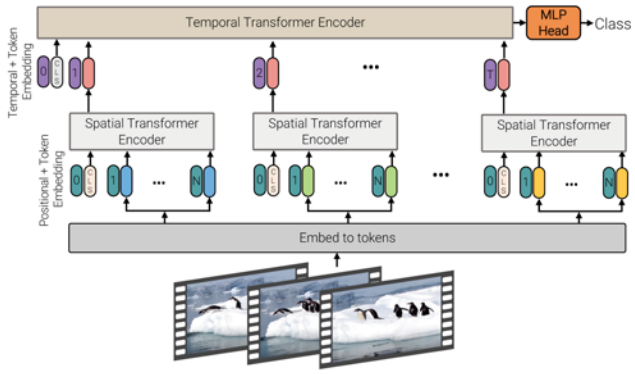

Figure 13. The factorised encoder of ViViT.

The ViViT variation using Factorised Encoder managed to achieve a state-of-the-art result and outcompete the concurrent video transformer on the smaller dataset, which shows the potential of this encoder model.

One interesting thing is that the authors used the pre-trained image classifier (ViT on JFT), which is trained on Google's JFT large-scale dataset for the image-level task, to initialize the model when training, which improved the accuracy by about 5%. This training approach provides a solution to the issue of the poor dataset in video classification.

D. Transformers on visual segmentation

PolyTransform. Liang and his team [14] designed an instance segmentation algorithm called PolyTransform. By combining popular segmentation methods and polygon mathematics, PloyTransform can generate geometry-based masks in object identification. The first step of PloyTransform is to create a segmentation network to produce instance masks. It then converts the masks into several polygons as initialization. And finally, PolyTransform will apply a deforming network to better suit the instance outlines. PolyTransform is proved to perform well in the Cityscapes dataset, and it can effectively promote the backbone instance segmentation network. It also shows great potential in the interactive annotation.

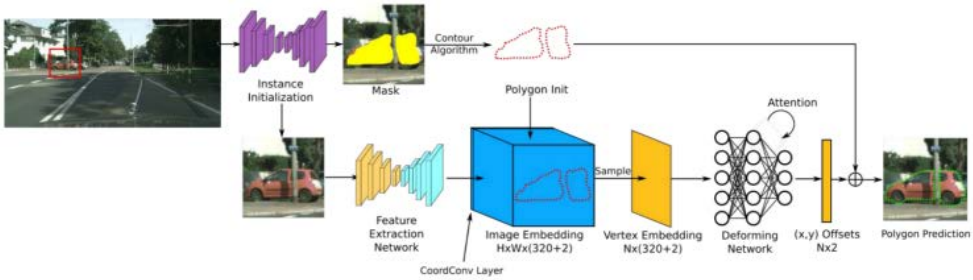

Figure 14. Overview of PolyTransform system

PolyTransform contains four parts: instance initialization, feature extraction network, deforming network, and learning module. The instance initialization module is designed to provide a good polygon initialization for each object, which will first set a model to generate a mask for each instance in the scene. The feature extraction network is expected to learn multiple-scale object boundary features. The deforming network can effectively predict the offset of each vertex to make the polygon snaps better fit the object boundaries. The learning module is to minimize the weighted sum of two losses: the first one penalizes the model for when the vertices deviate from the ground truth. The second one regularizes the edges of the polygon to prevent overlap and unstable movement of the vertices.

TeTrIS. Lee et al. [15] introduced and compared different methods on applying shape prior data into deep-based image segmentation, which is shown in figure 15. They used a template transformer network to deform shape templates to fit the underlying structure of interesting images via an end-to-end transformer network. They also introduced an effective approach to incorporate priors into pixel-wise classification, in which the shape templates were given as an extra input channel.

.**VisTR.** In this paper, Wang and her team [16] illustrated a new video instance segmentation framework based on transformers which are named VisTR. VisTR breaks the video instance segmentation tasks into direct end-to-end sequence decoding tasks. VisTR works on video inputs that consist of multiple image frames and then output a sequence of masks for each instance of the input video. The main promotion of VisTR is that it focuses on tracking the same aspects of similarity learning and can dramatically simplify the overall pipelines.

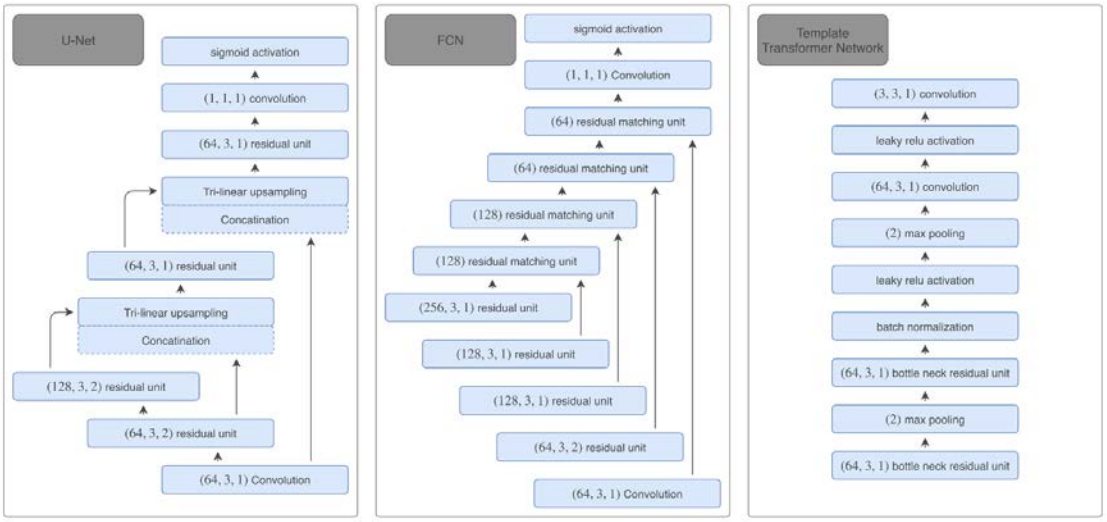

Figure 15. Graphical representation of the three different models: from left to right, the U-Net, FCN, and the black box model used to produce deformation parameters for TeTrIS.

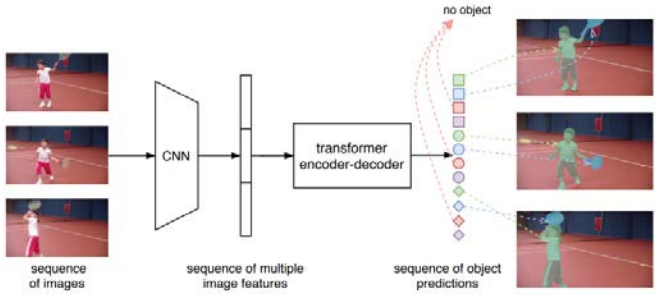

Figure 16. Overall Pipeline of VisTR

The architecture of VisTR has consisted of four parts: (i) a CNN backbone which extracts a sequence of multiple image features; (ii) an encoder transformer that similarities among all the pixel level features in the clip; (iii) a decoder transformer which decodes the top pixel features that can represent the instances of each frame, which is called instance level features; (iv) an instance sequence matching module; (v) an instance sequence segmentation module that outputs the final mask sequences.

III. EXPERIMENTS

A. Dataset & Metrics

COCO 2017. COCO is a data set provided by the Microsoft team that can be used for image recognition. Coco collects data through extensive use of Amazon Mechanical Turk. COCO data sets now have three annotation types: Object Instances, Object Keypoints, and Image Captions, which are stored by JSON files. We summarize the metrics as follow.

- **AP:** the average of all the categories (80). This is traditionally referred to as average accuracy.
- **AP50:** It has the same meaning as Mean Average Precision (MAP) in Pascal VOC. MAP is the average of all the APs of the categories.
- **AP75:** It's the same as the MAP in Pascal VOC, but the IOU threshold has been raised to 0.75, which is stricter and less accurate.
- **APS/APM/APL:** Measurement criteria were proposed for three images of different sizes (small, medium, and large), and COCO contained about 41% of small targets (Area<32×32), 34% of medium targets (32×32 lt; area<96×96), and 24% of the big targets (Area>96×96). Small targets are hard to improve.

MOT17. The video in MOT7 is exactly the same as the video in MOT16, but it's relatively fair with three detectors. It's the main data set in papers now. MOT16 dataset was put forward in 2016 to measure the standard of multi-target tracking detection and tracking methods, and it was especially used for pedestrian tracking. There are 14 videos, seven for the training set and seven for the test set, each of which is different. Here are the metrics for MOT17.

- **MOTA:** Multiple Object Tracking Accuracy. This measure combines three error sources: false positives, missed targets, and identity switches.
- **IDF1:** ID F1 Score. The ratio of correctly identified detections over the average number of ground-truth and computed detections.
- **MT:** Mostly tracked targets. The ratio of ground-truth trajectories covered by a track hypothesis for at least 80% of their respective life span.
- **ML:** Mostly lost targets. The ratio of ground-truth trajectories covered by a track hypothesis for at most 20% of their respective life span.
- **FP:** The total number of false positives.
- **FN:** The total number of false negatives (missed targets).
- **ID Sw:** The total number of identity switches.

Kinetics. The Kinetics datasets are series of large-scale datasets for human action recognition in videos. They consist of video clips collected from YouTube, lasting around 10 seconds. Each clip is provided with a single action class label for further training and testing. There are different variations of the Kinetics datasets. For Kinetics-400, 400 human action classes are covered. On the contrary, Kinetics-600 contains 600 action classes.

Something-Something V2 (SSV2). The something-something dataset contains video clips of human actions with a duration ranging from 2 to 6 seconds. Each clip from the dataset is labeled with descriptions in various templets such as "Dropping [something] into [something]", which is to be classified by classification models. The dataset provides more than 100,000 videos across 174 action classes in the format mentioned above.

Cityscapes Dataset. The Cityscapes Dataset presents a new large-scale dataset that contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, with high quality pixel-level annotations of 5 000 frames in addition to a larger set of 20 000 weakly annotated frames. The dataset is thus an order of magnitude larger than similar previous attempts. The Cityscapes Dataset is intended for assessing the performance of vision algorithms for major tasks of semantic urban scene understanding: pixel-level, instance-level, and panoptic semantic labeling; supporting research that aims to exploit large volumes of (weakly) annotated data, e.g., for training deep neural networks.

YouTube-VIS Dataset. YouTube-VIS is based on our initial YouTube-VOS dataset, which collected the first large-scale dataset for video instance segmentation. Video Instance Segmentation (VIS) extends the image instance segmentation task from the image domain to the video domain. The new problem aims at simultaneous detection, segmentation, and tracking of object instances in videos. Given a test video, the task requires the masks of all instances of a predefined category set to be labeled and the instance identities across frames to be associated.

B. Experimental results

Kinetics-400. In action classification, we usually take top-1 accuracy and top-5 accuracy as the evaluating indicators. Top-1 accuracy means how well the Top-1 outputs (top 1 prediction with the highest probability) of the model match the ground-truth labels. Top-5 accuracy means how well the Top-5 outputs (top 5 predictions with the highest probabilities) match the ground-truth labels.

From the results in Table 1, we can observe that the large-scale pretrained ViViT exhibits the best accuracy on both top-1 and top-5 accuracy (top-1: 84.8%; top-5: 95.8%). However, it is unfair to compare large-scale pretrained ViViT to other models because they used an unreleased dataset called JFT-300M. When omitting that model, we observe that the normal version of ViViT also obtained the highest accuracy on both top-1 and top-5 accuracy (top-1: 81.3%; top-5: 94.7%).

TABLE I. COMPARISON RESULTS ON KINETICS-400.

Method	Kinetics -400	
	Top-1 Accuracy	Top-5 Accuracy
VTN	78.6%	93.7%
A^2-Net	74.6%	91.5%
TimeSformer	80.7%	94.7%
STAM	79.20%	-
ViViT	81.3%	94.7%
ViViT (with large-scale pretraining)	84.8%	95.8%

Kinetics-600. As is shown in Table II, both versions of ViViT again achieved the highest performance. The normal version of ViViT obtained 83% top-1 accuracy and 95.7% top-5 accuracy, while large-scale pretrained ViViT achieved 83% top-1 accuracy and 95.7% top-5 accuracy.

TABLE II. COMPARISON RESULTS ON KINETICS-600.

Method	Kinetics -600	
	Top-1 Accuracy	Top-5 Accuracy
TimeSformer	82.4%	96.0%
ViViT	83.0%	95.7%
ViViT (with large-scale pretraining)	85.8%	96.5%

Something-Something V2. Among our summarized papers, only TimeSformer and ViViT evaluate on SSV2 dataset. Thus, we only show their performances for comparison, which is shown in Table III. ViViT shows superior results than TimeSformer on the SSV2 dataset. The results of ViViT are 65.4%, while TimeSformer obtains 62.5%.

TABLE III. THE COMPARISON BETWEEN TIMESFORMER AND VIVIT ON SSV2.

Method	Something-Something V2
	Top-1 Accuracy
TimeSformer	62.5%
ViViT	65.4%

COCO 2017. From the results, we can observe that the two-stage Deformable DETR exhibits the best accuracy on AP, AP50, AP75and APS, DETR-DC5-R101 exhibits the best accuracy on APM, DETR-R101 exhibits the best accuracy on APL, and DETR has the top speed. Table IV shows the results on COCO with other methods. With 150 epochs schedule, UP-DETR outperforms DETR by 0.8 AP and achieves comparable performance compared with Faster R-CNN-FPN (3 × schedules). With 300 epochs schedule, UP-DETR obtains 42.8

AP on COCO, which is 0.7 AP better than DETR (SwAV CNN) and 0.8 AP better than Faster R-CNN-FPN (9 × schedules). Overall, UP-DETR comprehensively outperforms DETR to detect small, medium, and large objects with both short and long training schedules. Regrettably, UP-DETR is still slightly lagging behind Faster R-CNN in APS because of the lacking of FPN-like architecture and the high-cost attention operation.

TABLE IV. THE EXPERIMENTAL RESULTS ON THE COCO DATASET.

Model	FPS	AP	AP50	AP75	APS	APM	APL
RetinaNet	18	**38.7**	**58.0**	**41.5**	**23.3**	**42.3**	**50.3**
Faster RCNN-DC5	16	**39.0**	**60.5**	**42.3**	**21.4**	**43.5**	**52.5**
Faster RCNN-FPN	26	**40.2**	**61.0**	**43.8**	**24.2**	**43.5**	**52.0**
Faster RCNN-R101-FPN	20	**42.0**	**62.5**	**45.9**	**25.2**	**45.6**	**54.6**
RetinaNet+	18	**41.1**	**60.4**	**43.7**	**25.6**	**44.8**	**53.6**
Faster RCNN-DC5+	16	**41.1**	**61.4**	**44.3**	**22.9**	**45.9**	**55.0**
Faster RCNN-FPN+	26	**42.0**	**62.1**	**45.5**	**26.6**	**45.4**	**53.4**
Faster RCNN-R101-FPN+	20	**44.0**	**63.9**	**47.8**	**27.2**	**48.1**	**56.0**
DETR	28	**42.0**	**62.4**	**44.2**	**20.5**	**45.8**	**61.1**
DETR-DC5	12	**43.3**	**63.1**	**45.9**	**22.5**	**47.3**	**61.1**
DETR-R101	20	**43.5**	**63.8**	**46.4**	**21.9**	**48.0**	**61.8**
DETR-DC5-R101	10	**44.9**	**64.7**	**47.7**	**23.7**	**49.5**	**62.3**
Deformable DETR	19	**43.8**	**62.6**	**47.7**	**26.4**	**47.1**	**58.0**
+ iterative bounding box refinement	19	**45.4**	**64.7**	**49.0**	**26.8**	**48.3**	**61.7**
++ two-stage Deformable DETR	19	**46.2**	**65.2**	**50.0**	**28.8**	**49.2**	**61.7**
Faster R-CNN		**40.2**	**61.0**	**43.8**	**24.2**	**43.5**	**52.0**
DETR (Supervised CNN)		**39.5**	**60.3**	**41.4**	**17.5**	**43.0**	**59.1**
DETR (SwAV CNN)		**39.7**	**60.3**	**41.7**	**18.5**	**43.8**	**57.5**
UP-DETR（3x）		**40.5**	**60.8**	**42.6**	**19.0**	**44.4**	**60.0**
Faster R-CNN		**42.0**	**62.1**	**45.5**	**26.6**	**45.4**	**53.4**
DETR (Supervised CNN)		**40.8**	**61.2**	**42.9**	**20.1**	**44.5**	**60.3**
DETR (SwAV CNN)		**42.1**	**63.1**	**44.5**	**19.7**	**46.3**	**60.9**
UP-DETR(6x)		**42.8**	**63.0**	**45.3**	**20.8**	**47.1**	**61.7**

MOT17. Under MOTA standards, CenterTrack has the best performance. Under IDF1 standards, Lif T has the best performance. Under MT standards, TrackFormer has the best performance. Under ML standards, TubeTK has the best performance. Under FP standards, Tracktor++ has the best performance. Under FN standards, CenterTrack has the best performance. Under ID Sw.standards, TT has the best performance.

TABLE V. COMPARISON RESULTS ON THE MOT17 DATASET.

Method	MOTA↑	IDF1 ↑	MT (%) ↑	ML (%) ↓	FP ↓	FN ↓	ID Sw. ↓
OFFLINE							
MHT DAM	50.7	47.2	20.8	36.9	22875	252889	2314
jCC	51.2	54.5	20.9	37.0	25937	247822	1802
FWT	51.3	47.6	21.4	35.2	24101	247921	2648
eHAF	51.8	54.7	23.4	37.9	33212	236772	1834
TT	54.9	63.1	24.4	38.0	20236	233295	1088
MPNTrack	58.8	61.7	28.8	33.5	17413	213594	1185
Lif T	60.5	65.6	27.0	33.5	14966	206619	1189
MHT-bLSTM	47.5	51.9	18.2	41.7	25981	268042	2069
EDMT	50.0	51.3	21.6	36.6	32279	247297	2264
JCC	51.2	54.5	20.9	37.0	25937	247822	1802
FWT	51.3	47.6	21.4	35.2	24101	247921	2648
ONLINE							
MOTDT	50.9	52.7	17.5	35.7	24069	250768	2474
FAMNet	52.0	48.7	19.1	33.4	14138	253616	3072
Tracktor++	56.3	55.1	21.1	35.2	8866	235449	1987
GSM Tracktor	56.4	57.8	22.2	34.5	14379	230174	1485
CenterTrack	61.5	59.6	26.3	31.9	14076	200672	2583
TrackFormer	61.8	59.8	35.4	21.0	35226	177270	2982
DMAN	48.2	55.7	19.3	38.3	26218	263608	2194
MOTDT	50.9	52.7	17.5	35.7	24069	250768	2474
Tracktor	53.5	52.3	19.5	36.6	12201	248047	2072
Tracktor+CTdet	54.4	56.1	25.7	29.8	44109	210774	2574
DeepSORT	60.3	61.2	31.5	20.3	36111	185301	2442
TubeTK	63.0	58.6	31.2	19.9	27060	177483	4137
CenterTrack	67.8	64.7	34.6	24.6	18489	160332	3039
ChainedTracker	66.6	57.4	32.2	24.2	22284	160491	5529
TransTrack	65.8	56.9	32.2	21.8	24000	163683	5355

Cityscapes dataset. As shown in Table VI, PolyTransform has better overall average precision and 50% overlap average precision. In test sets including person, rider, car, truck, bus, train, and motorcycle, PolyTransform performed the best; but it is proven to perform relatively weak when testing bicycle models.

YOUTUBE-VIS DATASET. The test result of VisTR was compared against some state-of-the-art methods in video instance segmentation in terms of both accuracy and speed. For the accuracy measured by AP, VisTR achieves the best result among methods using a single model without any bells and whistles. Using the same backbone of resnet50, VisTR achieves about 5 points higher in AP than the masktrack R-CNN and the recently proposed STEm-Seg method. The result is shown in Table VII.

TABLE VI. COMPARISON RESULTS ON CITYSCAPES DATASET. THE RESULTS WERE BORROWED FROM [12].

Methods	Training data	APval	AP	AP50	person	rider	car	truck	bus	train	mcycle	bcycle
DWT	fine	21.2	19.4	35.3	15.5	14.1	31.5	22.5	27	22.9	13.9	8
Kendakk et al.	fine	~	21.6	39	19.2	21.4	36.6	18.8	26.8	15.9	19.4	14.5
Arnab et al.	fine	~	23.4	45.2	21	18.4	31.7	22.8	31.1	31	19.6	11.7
SGN	fine+coarse	29.2	25	44.9	21.8	20.1	39.4	24.8	33.2	30.8	17.7	12.4
PolygonRNN++	fine	~	25.5	45.5	29.4	21.8	48.3	21.2	32.3	23.7	13.6	13.6
Mask R-CNN	fine	31.5	26.2	49.9	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16
BShapeNet+	fine	~	27.3	50.4	29.7	23.4	46.7	26.1	33.3	24.8	20.3	14.1
GMIS	fine+coarse	~	27.3	45.6	31.5	25.2	42.3	21.8	37.2	28.9	18.8	12.8
Neven et al.	fine	~	27.6	50.9	34.5	26.1	52.4	21.7	31.2	16.4	20.1	18.9
PANet	fine	36.5	31.8	57.1	36.8	30.4	54.8	27	36.3	25.5	22.6	20.8
Mask R-CNN	fine+COCO	36.4	32	58.1	34.8	27	49.1	30.1	40.9	30.9	24.1	18.7
AdaptIS	fine	36.3	32.5	52.5	31.4	29.1	50	31.6	41.7	39.4	24.7	12.1
SSAP	fine	37.3	32.7	51.8	35.4	25.5	55.9	33.2	43.9	31.9	19.5	16.2
BShapeNet+	fine+COCO	~	32.9	58.8	36.6	24.8	50.4	33.7	41	33.7	25.4	17.8
UPSNet	fine+COCO	37.8	33	59.7	35.9	27.4	51.9	31.8	43.1	31.4	23.8	19.1
PANet	fine+COCO	41.4	36.4	63.1	41.5	33.6	58.2	31.8	45.3	28.74	28.2	24.1
PolyTransform	fine+COCO	44.6	40.1	65.9	42.4	34.8	58.5	39.8	50	41.3	30.9	23.4

TABLE VII. THE DETAILED COMPARISON ON YOUTUBE-VIS DATASET. THE RESULTS WERE BORROWED FROM [14].

Length	AP	AP50	AP75	AR1	AR10
18	29.7	50.4	31.1	29.5	34.4
24	30.5	47.8	33	29.5	34.4
30	31.7	53.2	32.8	31.3	36
36	33.3	53.4	35.1	33.1	38.5

(a) Video sequence length.

Levels	#	AP	AP50	AP75	AR1	AR10
video level	1	8.4	13.2	9.5	20	20.8
frame level	36	13.7	23.3	14.5	30.4	35.1
ins. Level	10	32	52.8	34	31.6	37.2
pred. level	360	33.3	53.4	35.1	33.1	38.5

(b) Instance query embedding.

time order	AP	AP50	AP75	AR1	AR10
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.

	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

(d) Position encoding.

	AP	AP50	AP75	AR1	AR10
CNN	32	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.

	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

(f) Instance sequence segmentation module.

IV. CONCLUSION

This paper has provided a brief introduction to the transformer and its applications on computer vision tasks. We summarized 15 articles covering transformers on image object detection, multiple object tracking, action classification, and visual segmentation. We gave a short presentation of the method in each paper. In addition, we summarized 6 related datasets and metrics used to test and evaluate these methods. Hopefully, this article would be helpful for beginners to set up a general idea of the application of transformers in the computer vision field.

REFERENCES

[1] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In arXiv preprint arXiv:1706.03762, 2017.

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929, 2020.

[3] Carion N., Massa F., Synnaeve G., Usunier N., Kirillov A., Zagoruyko S. (2020) End-to-End Object Detection with Transformers. In: Vedaldi A., Bischof H., Brox T., Frahm JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, vol 12346. Springer, Cham. https://doi.org/10.1007/978-3-030-58452-8_13

[4] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, Jifeng Dai, DEFORMABLE DETR: DEFORMABLE TRANSFORMERS FOR END-TO-END OBJECT DETECTION In arXiv preprint arXiv: 2010.04159,2010.

[5] Zhigang Dai1, Bolun Cai, Yugeng Lin, Junying Chen, UP-DETR: Unsupervised Pre-training for Object Detection with Transformers In arXiv preprint arXiv: 2011.09094,2011.

[6] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, Christoph Feichtenhofer, TrackFormer: Multi-Object Tracking with Transformers In arXiv preprint arXiv:2101.02702,2021.

[7] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie , Jinkun Cao , Xinting Hu, Tao Kong , Zehuan Yuan , Changhu Wang , Ping Luo TransTrack: Multiple-Object Tracking with Transformer In arXiv preprint arXiv: 2012.15460, 2012.

[8] Girdhar, Rohit, João Carreira, Carl Doersch, and Andrew Zisserman. Video Action Transformer Network. In arXiv preprint arXiv: 1812.02707, 2018.

[9] Neimark, Daniel, Omri Bar, Maya Zohar, and Dotan Asselmann. Video Transformer Network. In arXiv preprint arXiv: 2102.00719, 2021.

[10] Chen, Yunpeng, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A2-Nets: Double Attention Networks. In arXiv preprint arXiv:1810.11579, 2018.

[11] Bertasius, Gedas, Heng Wang, and Lorenzo Torresani. Is Space-Time Attention All You Need for Video Understanding? In arXiv preprint arXiv:2102.05095, 2021.

[12] Sharir, Gilad, Asaf Noy, and Lihi Zelnik-Manor. An Image Is Worth 16x16 Words, What Is a Video Worth? In arXiv preprint arXiv:2103.13915, 2021.

[13] Arnab, Anurag, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A Video Vision Transformer. In arXiv preprint arXiv: 2103.15691, 2021.

[14] J. Liang, N. Homayounfar, W.-C. Ma, Y. Xiong, R. Hu, and R. Urtasun, "PolyTransform: Deep Polygon Transformer for Instance Segmentation," http://openaccess.thecvf.com/, 01-Jan-1970.

[15] M. C. H. Lee, K. Petersen, N. Pawlowski, B. Glocker and M. Schaap, "TeTrIS: Template Transformer Networks for Image Segmentation With Shape Priors," in IEEE Transactions on Medical Imaging, vol. 38, no. 11, pp. 2596-2606, Nov. 2019, doi: 10.1109/TMI.2019.2905990.

[16] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-End Video Instance Segmentation with Transformers," arXiv.org, 24-Mar-2021. [Online]. Available: https://arxiv.org/abs/2011.14503. [Accessed: 25-Apr-2021]