A Survey on Visual Transformer

Kai Han[1], Yunhe Wang[1*], Hanting Chen[1,2], Xinghao Chen[1], Jianyuan Guo[1], Zhenhua Liu[1,2], Yehui Tang[1,2], An Xiao[1], Chunjing Xu[1], Yixing Xu[1], Zhaohui Yang[1,2], Yiman Zhang[1], Dacheng Tao[3*]
[1]Noah's Ark Lab, Huawei Technologies
[2]Peking University [3]University of Sydney
{kai.han,yunhe.wang,xinghao.chen,jianyuan.guo,xiaoan1,xuchunjing,yixing.xu,zhangyiman1}@huawei.com
{htchen,liu-zh,yhtang,zhaohuiyang}@pku.edu.cn, dacheng.tao@sydney.edu.au

Abstract

Transformer is a type of deep neural network mainly based on self-attention mechanism which is originally applied in natural language processing field. Inspired by the strong representation ability of transformer, researchers propose to extend transformer for computer vision tasks. Transformer-based models show competitive and even better performance on various visual benchmarks compared to other network types such as convolutional networks and recurrent networks. In this paper we provide a literature review of these visual transformer models by categorizing them in different tasks and analyze the advantages and disadvantages of these methods. In particular, the main categories include the basic image classification, high-level vision, low-level vision and video processing. Self-attention in computer vision is also briefly revisited as self-attention is the base component in transformer. Efficient transformer methods are included for pushing transformer into real applications. Finally, we give a discussion about the further research directions for visual transformer.

1. Introduction

Deep neural networks have become the fundamental infrastructure in modern artificial intelligence system. There have been various network types proposed for addressing different tasks. Multi-layer perception (MLP) or say fully connected (FC) network is the classical neural network which is stacked of multiple linear layers and nonlinear activations [104, 105]. Convolutional neural networks (CNNs) introduce convolutional layers and pooling layers for processing shift-invariant data such as images [68, 65]. Recurrent neural networks (RNNs) utilize the recurrent cells to process sequential data or time series data [106, 49]. Transformer is a kind of newly proposed neural network which mainly utilizes self-attention mechanism [5, 90] to extract intrinsic feature [123]. Among these networks, transformer is a recently invented neural network and show great potential for extensive artificial intelligence applications.

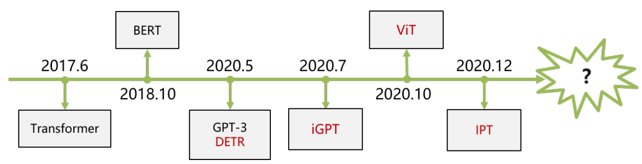

Figure 1. Milestones of transformer. The visual transformer models are in red.

Transformer is originally applied on natural language processing (NLP) tasks and brings in significant improvement [123, 29, 10]. For example, Vaswani *et al*. [123] first propose the transformer based solely on attention mechanisms for machine translation and English constituency parsing tasks. Devlin *et al*. [29] introduce a new language representation model called BERT, which pre-trains a transformer from unlabeled text by jointly conditioning on both left and right context. BERT obtains state-of-the-art results on eleven NLP tasks of the time. Brown *et al*. [10] pre-train a gigantic transformer based model GPT-3 with 175 billion parameters on 45TB compressed plaintext data and achieve strong performances on different types of downstream natural language tasks without fine-tuning. These Transformer based models show strong representation capacity and have obtained breakthrough in NLP area.

Inspired by the power of transformer in NLP, recently researchers extend transformer for computer vision (CV) tasks. CNNs used to be the fundamental component in vision applications [47, 103], but transformer is showing its ability as an alternative of CNN. Chen *et al*. [18] train a sequence transformer to auto-regressively predict pixels and achieve competitive results with CNNs on image classification task. ViT is a recently proposed vision transformer model by Dosovitskiy *et al*. [31] which applies a pure trans-

[*]Corresponding authors. All authors are in alphabetical order of last name (except the first and the corresponding authors).

Table 1. Representative works of visual transformers.

Subject	Secondary Subject	Method	Keypoints	Publication
Image classification	Image classification	iGPT [18]	Pixel prediction self-supervised learning, GPT model	ICML 2020
		ViT [31]	Image patches, standard transformer	arXiv 2020
High-level vision	Object detection	DETR [14]	Set-based prediction, bipartite matching, transformer	ECCV 2020
		Deformable DETR [155]	DETR, deformable attention module	arXiv 2020
		ACT [153]	Adaptive clustering transformer	arXiv 2020
		UP-DETR [28]	Unsupervised pre-training, random query patch detection	arXiv 2020
		TSP [117]	New bipartite matching, encoder-only transformer	arXiv 2020
	Segmentation	Max-DeepLab [126]	PQ-style bipartite matching, dual-path transformer	arXiv 2020
		VisTR [129]	Instance sequence matching, instance sequence segmentation	arXiv 2020
Low-level vision	Image enhancement	IPT [17]	Multi-task, ImageNet pre-training, transformer model	arXiv 2020
		TTSR [135]	Texture transformer, RefSR	CVPR 2020
	Image generation	Image Transformer [92]	Pixel generation using transformer	ICML 2018
Video processing	Video Inpainting	STTN [144]	Spatial-temporal adversarial loss	ECCV 2020
	Video Captioning	Masked Transformer [154]	Masking network, event proposal	CVPR 2018
Efficient transformer	Decomposition	ASH [85]	Number of heads, importance estimation	NeurIPS 2019
	Distillation	TinyBert [62]	Various losses for different modules	EMNLP Findings 2020
	Quantization	FullyQT [97]	Fully quantized transformer	EMNLP Findings 2020
	Architecture Design	ConvBert [61]	Local fependence, dynamic vonvolution	NeurIPS 2020

former directly to sequences of image patches and attains state-of-the-art performance on multiple image recognition benchmarks. Apart from the basic image classification, transformer has been utilized to address more computer vision problems such as object detection [14, 155], semantic segmentation, image processing and video understanding. Due to the excellent performance, more and more transformer-based models are proposed for improving various visual tasks.

The transformer-based vision models are springing up like mushrooms, which leads to difficulty to keep pace with the rate of new progress. Thus, a survey of the existing works is agent and can be beneficial for the community. In this paper, we focus on providing a comprehensive overview of the recent advances in visual transformers and discuss about the potential directions for further improvement. To have a better archive and be convenient to researchers on different topics, we category the transformer models by their application scenarios as shown in Table 1. In particular, the main subjects include basic image classification, high-level vision, low-level vision and video processing. High-level vision deals with the interpretation and use of what is seen in the image [121] such as object detection, segmentation and lane detection. There have been a number of transformer models addressing these high-level vision tasks, such as DETR [14], deformable DETR [155] for object detection and Max-DeepLab [126] for segmentation. Low level image processing is mainly concerned with extracting descriptions from images (that are usually represented as images themselves) [35], whose typical applications include super-resolution, image denoising and style transfer. Few works [17, 92] in low-level vision use transformers and more investigation is required. Video processing is an import part in computer vision in addition to image-based tasks. Transformer can be applied in video naturally [154, 144] due to the sequential property of video. Transformer is beginning to show competitive performance on these tasks compared to the conventional CNNs or RNNs. Here we give a survey on these works of transformer-based visual models to keep pace with the progress in this field. The development timeline of visual transformer is shown in Figure 1 and we believe more and more excellent works will be engraved in the milestones.

The rest of the paper is organized as follows. Section 2 first formulates the self-attention mechanism and the standard transformer. We describe the methods of transformer in NLP in Section 3 as the research experience may be beneficial for vision tasks. Next, Section 4 is the main part of the paper, in which we summarize the visual transformer models on image classification, high-level vision, low-level vision and video tasks. We also briefly revisit self-attention mechanism for CV and efficient transformer methods as they are closely related to our main topic. Finally, we give a conclusion and discuss about several research directions and challenges.

2. Formulation of Transformer

Transformer [123] was firstly applied on the machine translation task in neural language processing (NLP). As shown in Fig. 2, it consists of an encoder module and a decoder module with several encoders/decoders of the same architecture. Each encoder is composed of a self-attention layer and a feed-forward neural network, while each decoder is composed of a self-attention layer, an encoder-decoder attention layer and a feed-forward neural network. Before translating sentences with transformer, each word in the sentence will be embedded into a vector with $d_{model} =$

512 dimensions.

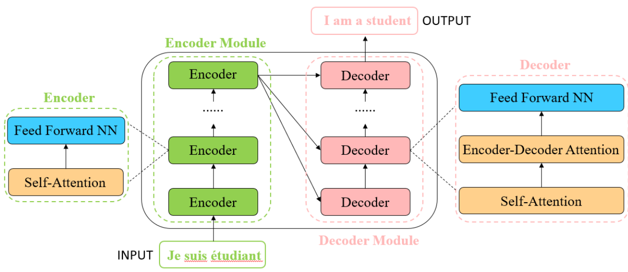

Figure 2. Pipeline of vanilla transformer.

2.1. Self-Attention Layer

In self-attention layer, the input vector is firstly transformed into three different vectors, *i.e.*, the query vector q, the key vector k and the value vector v with dimension $d_q = d_k = d_v = d_{model} = 512$. Vectors derived from different inputs are then packed together into three different matrices Q, K and V. After that, the attention function between different input vectors is calculated with the following steps (as shown in Fig. 3 left):

- **Step 1:** Compute scores between different input vectors with $S = Q \cdot K^\top$;
- **Step 2:** Normalize the scores for the stability of gradient with $S_n = S/\sqrt{d_k}$;
- **Step 3:** Translate the scores into probabilities with softmax function $P = softmax(S_n)$;
- **Step 4:** Get the weighted value matrix with $Z = V \cdot P$.

The process can be unified into a single function:

$$Attention(Q, K, V) = softmax(\frac{Q \cdot K^\top}{\sqrt{d_k}}) \cdot V. \quad (1)$$

The intuition behind Eq. 1 is simple. Step 1 calculates scores between two different vectors, and the score is to determine the degree of attention that we put on other words when encode the word at the current position. Step 2 normalize the scores to have more stable gradients for better training, and step 3 transfer the scores into probabilities. Finally, each value vector is multiplied by the sum-upped probability, and vectors with larger probabilities will be focused on more by the following layers.

The encoder-decoder attention layer in the decoder module is almost the same as self-attention layer in the encoder-module, except that the key matrix K and value matrix V are derived from the encoder module and the query matrix Q is derived from the previous layer.

Note that the above process is irrelevant to the position of each word, thus the self-attention layer lacks the ability of capturing the positional information of the words in a sentence. To address this, a positional encoding with dimension d_{model} is added to the original input embedding to get the final input vector of the word. Specifically, the position is encoded with the following equation:

$$PE(pos, 2i) = sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}}); \quad (2)$$

$$PE(pos, 2i+1) = cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}}), \quad (3)$$

in which pos denotes the position of the word in a sentence, and i represents the current dimension of the positional encoding.

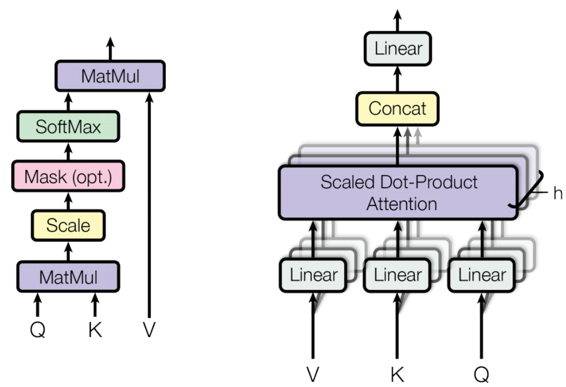

Figure 3. (Left) The process of self-attention. (Right) Multi-head attention. (The image is from [123])

2.2. Multi-Head Attention

The self-attention layer is further improved by adding a mechanism called multi-head attention in order to boost the performance of the vanilla self-attention layer. Note that for a given reference word, we often want to focus on several other words when going through the sentence. Thus, a single-head self-attention layer limits the ability of focusing on a specific position (or several specific positions) while does not influence the attention on other positions that is equally important at the same time. This is achieved by giving attention layers different representation subspace. Specifically, different query, key and value matrices are used for different heads, and they can project the input vectors into different representation subspace after training due to the random initialization.

In detail, given an input vector and the number of heads h, the input vector is firstly transformed into three different groups of vectors, *i.e.*, the query group, the key group and the value group. There are h vectors in each group with dimension $d_{q'} = d_{k'} = d_{v'} = d_{model}/h = 64$. Then, vectors derived from different inputs are packed together into three different groups of matrices $\{Q_i\}_{i=1}^h$, $\{K_i\}_{i=1}^h$ and $\{V_i\}_{i=1}^h$. Then, the process of multi-head attention is shown as follow:

$$MultiHead(Q', K', V') = Concat(head_1, \cdots, head_h)W^o,$$
$$\text{where } head_i = Attention(Q_i, K_i, V_i), \quad (4)$$

where Q' is the concatenation of $\{Q_i\}_{i=1}^h$ (so is K' and V') and $W^o \in \mathbb{R}^{d_{model} \times d_{model}}$ is the linear projection matrix.

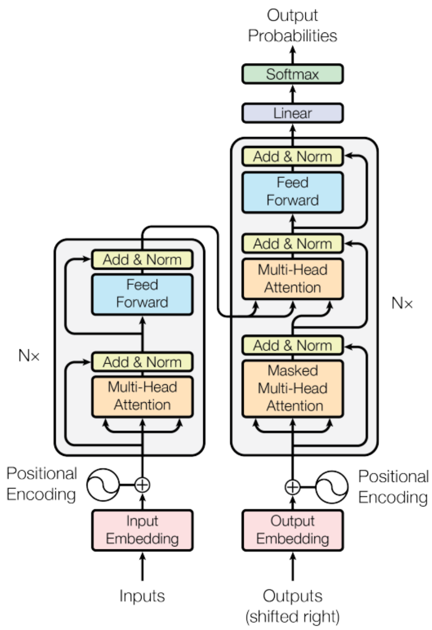

Figure 4. Detailed structure of transformer. (The image is from [123])

2.3. Other Parts in Transformer

Residual in the encoder and decoder. As shown in Fig. 4, a residual connection is added in each sub-layer in the encoder and decoder in order to strengthen the flow of information and get a better performance. A layer-normalization [4] is followed afterward. The output of operations mentioned above can be described as:

$$LayerNorm(X + Attention(X)). \quad (5)$$

Note that X is used as the input of self-attention layer here, since the query, key and value matrices Q, K and V are all derived from the same input matrix X.

Feed-forward neural network. A feed-forward NN is applied after the self-attention layers in each encoder and decoder. Specifically, the feed-forward NN is consist of two linear transformation layers and a ReLU activation function within them, which can be denoted as the following function:

$$FFNN(X) = W_2\sigma(W_1X), \quad (6)$$

where W_1 and W_2 are the two parameter matrices of the two linear transformation layers, and σ represents the ReLU activation function. The dimensionality of the hidden layer is $d_h = 2048$.

Final layer in decoder. The final layer in decoder aims to turn the stack of vectors back into a word. This is achieved by a linear layer followed by a softmax layer. The linear layer project the vector into a logits vector with d_{word} dimensions in which d_{word} is the number of words in the vocabulary. Then, a softmax layer is used to transform the logits vector into probabilities.

Most of the transformers used in the computer vision tasks utilize the encoder module of the original transformer. In short, it can be treated as a new feature selector that is different from the convolutional neural networks (CNNs) and the recurrent neural networks (RNNs). Compared to CNN which only focuses on local characteristics, transformer is able to capture long distance characteristics which means that the global information can be easily derived by transformer. Compared to RNN whose hidden state must be computed in sequence, transformer is much efficient since the output of the self-attention layer and the fully-connected layers can be computed in parallel and accelerated easily. Thus, it is meaningful to further study the application of transformer in the area of not only NLP but also computer vision.

3. Revisiting Transformers for NLP

Before the advent of Transformer, recurrent neural networks (*e.g.*, GRU [26] and LSTM [50]) with added attention empower most of the state-of-the-art language models. However, in RNNs, the information flow needs to be processed sequentially from the previous hidden states to the next one, which precludes the acceleration and parallelization during training, and thus hinders the potential of RNNs to process longer sequences or build larger models. In 2017, Vaswani *et al.* [123] proposes Transformer, a novel encoder-decoder architecture solely built on multi-head self-attention mechanisms and feed-forward neural networks, aiming to solve seq-to-seq natural language tasks (*e.g.*, machine translation) with acquiring global dependencies at ease. The success of Transformer demonstrates that leveraging attention mechanisms alone can achieve comparable performances with attentive RNNs. Moreover, the architecture of Transformer favors massively parallel computing, enabling training on larger datasets and thus leading to the burst of large pre-trained models (PTM) for natural language processing.

BERT [29] and its variants (*e.g.*, SpanBERT [63], RoBERTa [82]) are a series of PTMs built on the multi-layer Transformer encoder architecture. Two tasks are conducted on BookCorpus [156] and English Wikipedia datasets at the pre-training stage of BERT: 1) Masked language modeling (MLM) via first randomly masking out some tokens in the input and then train the model to predict; 2) Next sentence prediction using paired sentences as input and predicting whether the second sentence is the original one in the document. After pre-training, BERT can be fine-tuned by adding one output layer alone on a wide range of downstream tasks. More specifically, When performing sequence level tasks (*e.g.*, sentiment analysis), BERT uses the representation of the first token for classification; while for token-level tasks (*e.g.*, Name Entity Recognition), all tokens are fed into the softmax layer for classification. At the time of release, BERT achieves the state-of-the-art results on 11 natural language processing task, setting up a milestone in pre-trained language models. Generative Pre-Trained Transformer series (*e.g.*, GPT [99], GPT-2 [100]) are another type of pre-trained models based on the Transformer decoder architecture, which uses masked self-attention mechanisms. The major difference between GPT series and BERT lies in the way of pre-training. Unlike BERT, GPT series are one-directional language models pre-trained by Left-to-Right (LTR) language modeling. Besides, sentence separator ([SEP]) and classifier token ([CLS]) are only involved in the fine-tuning stage of GPT but BERT learns those embeddings during pre-training. Because the one-directional pre-pretraining strategy of GPT, it shows superiority in many natural language generation tasks. More recently, a gigantic transformer-based model, GPT-3, with incredibly 175 billion parameters has been introduced [10]. By pre-training on 45TB compressed plaintext data, GPT-3 claims the ability to directly process different types of downstream natural language tasks without fine-tuning, achieving strong performances on many NLP datasets, including both natural language understanding and generation. Besides the aforementioned transformer-based PTMs, many other models have been proposed since the introduction of Transformer. For this is not the major topic in our survey, we simply list a few representative models in Table 2 for interested readers.

Apart from the PTMs trained on large corpora for general natural language processing tasks, transformer-based models have been applied in many other NLP related domains or multi-modal tasks. ***BioNLP Domain.*** Transformer-based models have outperformed many traditional biomedical methods. BioBERT [69] uses Transformer architecture for biomedical text mining tasks; SciBERT [7] is developed by training Transformer on 114M scientific articles covering biomedical and computer science field, aiming to execute NLP tasks related to scientific domain more precisely; Huang *et al.* [55] proposes ClinicalBERT utilizing Transformer to develop and evaluate continuous representations of clinical notes and as a side effect, the attention map of ClinicalBERT can be used to explain predictions and thus discover high-quality connections between different medical contents. ***Multi-Modal Tasks.*** Owing to the success of Transformer across text-based NLP tasks, many researches are committed to exploiting the potential of Transformer to process multi-modal tasks (*e.g.*, video-text, image-text and audio-text). VideoBERT [115] uses a CNN-based module pre-processing the video to get the representation tokens, based on which a Transformer encoder is trained to learn the video-text representations for downstream tasks, such as video caption. VisualBERT [72] and VL-BERT [114] propose single-stream unified Transformer to capture visual elements and image-text relationship for downstream tasks like visual question answering (VQA) and visual commonsence reasoning (VCR). Moreover, several studies such as SpeechBERT [24] explore the possibility of encoding audio and text pairs with a Transformer encoder to process auto-text tasks like Speech Question Answering (SQA).

Table 2. List of representative language models built on Transformer.

Models	Architecture	Params	Fine-tuning
GPT [99]	Transformer Dec.	117M	Yes
GPT-2 [100]	Transformer Dec.	117M$\sim$1542M	No
GPT-3 [10]	Transformer Dec.	125M$\sim$175B	No
BERT [29]	Transformer Enc.	110M$\sim$340M	Yes
RoBERTa [82]	Transformer Enc.	355M	Yes
XLNet [136]	Two-Stream Transformer Enc.	$\approx$ BERT	Yes
ELECTRA [27]	Transformer Enc.	335M	Yes
UniLM [30]	Transformer Enc.	340M	Yes
BART [70]	Transformer	110% of BERT	Yes
T5 [101]	Transfomer	220M$\sim$11B	Yes
ERNIE (THU) [149]	Transform Enc.	114M	Yes
KnowBERT [94]	Transformer Enc.	253M$\sim$523M	Yes

[1] Transformer is the standard encoder-decoder architecture. Transfomer Enc. and Dec. represent the encoder and decoder part of standard Transformer. Decoder uses mask self-attention to prevent attending to the future tokens.
[2] The data of the Table is from [98].

The rapid development of transformer-based models on varieties of natural language processing as well as NLP-related tasks demonstrates its structural superiority and versatility. This empowers Transformer to become a universal module in many other AI fields beyond natural language processing. The following part of this survey will focus on the applications of Transformer in a wide range of computer vision tasks emerged in the past two years.

4. Visual Transformer

In this section, we provide a comprehensive review of the transformer-based models in computer vision, including the applications in image classification, high-level vision, low-level vision and video processing. We also briefly summarize the applications of self-attention mechanism and model compression methods for efficient transformer.

4.1. Image Classification

Inspired by great success of transformer on natural language processing, some researchers have tried to examine whether similar models can learn useful representations for images. As a higher dimensional, noiser, and more redundant modality than text, images are believed to be difficult for generative modeling. iGPT [18] and ViT [31] are the two works purely using transformer for image classification.

4.1.1 iGPT

It has been a long time since the original wave of generative pre-training methods for images, Chen *et al*. [18] re-examine this class of methods and combine with the recent progress of self-supervised methods. The approach consists of a pre-training stage followed by a fine-tuning stage. In pre-training, auto-regressive and BERT objectives are explored. Besides, sequence transformer architecture is applied to predict pixels instead of language tokens in NLP. Pre-training can be viewed as a favorable initialization or as a regularizer when used in combination with early stopping. During fine-tuning, they adds a small classification head to the model, which is used to optimize a classification objective and adapts all weights.

Given an unlabeled dataset X consisting of high dimensional data $x = (x_1, \cdots, x_n)$. They train the model by minimizing the negative log-likelihood of the data:

$$L_{AR} = \mathop{\mathbb{E}}_{x \sim X} [-log\, p(x)] \tag{7}$$

where $p(x)$ is the density of the data of images, which can be modeled as:

$$p(x) = \prod_{i=1}^{n} p(x_{\pi_i} | x_{\pi_1}, \cdots, x_{\pi_{i-1}}, \theta) \tag{8}$$

where the identity permutation $\pi_i = i$ is adopted for $1 \leqslant i \leqslant n$, also known as raster order. They also consider the BERT objective, which samples a sub-sequence $M \subset [1, n]$ such that each index i independently has probability 0.15 of appearing in M. The M is called the BERT mask, and the model is trained by minimizing the negative log-likelihood of the "masked" elements x_M conditioned on the "unmasked" ones $x_{[1,n]\backslash M}$:

$$L_{BERT} = \mathop{\mathbb{E}}_{x \sim X} \mathop{\mathbb{E}}_{M} \sum_{i \in M} [-log\, p(x_i | x_{[1,n]\backslash M})] \tag{9}$$

In pre-training, they pick one of L_{AR} or L_{BERT} and minimize the loss over the pre-training dataset.

They use the GPT-2 [100] formulation of the transformer decoder block. In particular, layer norms precede both the attention and Multi-Layer Perceptron (MLP) operations, and all operations lie strictly on residual paths. The only mixing across sequence elements occurs in the attention operation, and to ensure proper conditioning when training the AR objective, they apply the standard upper triangular mask to the $n \times n$ matrix of attention logits. When using the BERT objective, no attention logit masking is required: after applying content embeddings to the input sequence, they zero out the positions.

Following the final transformer layer, they apply a layer norm and learn a projection from the output to logits parameterizing the conditional distributions at each sequence element. When training BERT, they simply ignore the logits at unmasked positions.

During fine-tuning, they average pool the output of the final layer norm n^L across the sequence dimension to extract a d-dimensional vector of features per example:

$$f^L = \langle n_i^L \rangle_i \tag{10}$$

They learn a projection from f_L to class logits, which is utilized to minimize a cross entropy loss L_{CLF}. In practice, they find empirically that the joint objective $L_{GEN} + L_{CLF}$ works even better, where $L_{GEN} \in \{L_{AR}, L_{BERT}\}$.

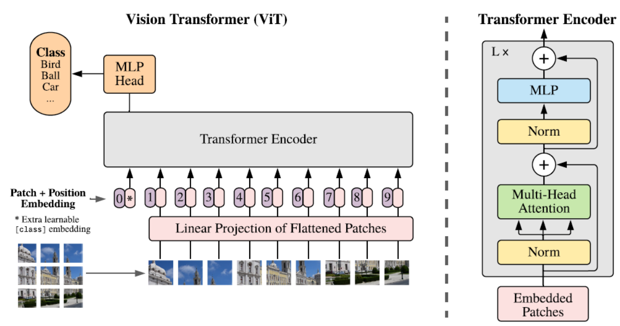

Figure 5. The framework of the Vision Transformer (The image is from [31]).

4.1.2 ViT

Recently, Dosovitskiy et al. [31] propose a pure transformer, namely Vision Transformer (ViT), which performs well on image classification tasks when applied directly to sequences of image patches. They follow as closely as possible the design of the original Transformer. Figure 5 shows the framework of Vision Transformer.

To handle 2D images, the image $x \in \mathbb{R}^{H \times W \times C}$ is reshaped into a sequence of flattened 2D patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$. (H, W) is the resolution of the original image and (P, P) is the resolution of each image patch. $N = HW/P^2$ is then the effective sequence length for the transformer. Since the transformer uses constant widths through all of its layers, a trainable linear projection maps each vectorized path to the model dimension D, the output of which they refer to as patch embeddings.

Similar to BERT's $[class]$ token, a learnable embedding is employed to the sequence of embedding patches, whose state at the output of the transformer encoder serves as the image representation. During both pre-training and fine-tuning, the classification head is attached to the same size. In addition, 1D position embeddings are added to the patch embeddings to retain positional information. They have explored different 2D-aware variants of position embeddings, which does not obtain significant gains over standard 1D position embeddings. The joint embeddings severs as input to the encoder. It is worth noting that the Vision Transformer only employs the encoder of the standard transformer and a MLP head is followed by the output of the transformer encoder.

Typically, the ViT is firstly pre-trained on large datasets, and fine-tune to smaller downstream tasks. For this, the pre-trained prediction head is removed and a zero-initialized $D \times K$ feedforward layer is attached, where K is the number of downstream classes. It is often beneficial to fine-tune at higher resolution than pre-training. When feeding images of higher resolution, the patch size is kept the same, which results in a larger effective sequence length. The Vision Transformer can handle arbitrary sequence lengths, however, the pre-trained position embeddings may no longer be meaningful. The authors therefore perform 2D interpolation of the pre-trained position embeddings according to their location in the original image. Note that this resolution adjustment and patch extraction are the only points at which an inductive bias about the 2D structure of the images is manually injected into the Vision Transformer.

Such models yield modest results when trained on mid-sized datasets such as ImageNet, achieving accuracies of a few percentage points below ResNets of comparable size. Transformers lack some inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data. However, the picture changes if the models are trained on large datasets (14M-300M images). The authors find that large scale training trumps inductive bias. Transformers attain excellent results when pre-trained at sufficient scale and transferred to tasks with fewer datapoints. The Vision Transformer, pretrained on the JFT-300M dataset, approaches or beats state of the art on multiple image recognition benchmarks, reaching accuracy of 88.36% on ImageNet, 99.50% on CIFAR-10, 94.55% on CIFAR-100, and 77.16% on the VTAB suite of 19 tasks. The detail results of iGPT and ViT are shown in table 3.

In conclusion, iGPT recalls the generative pre-training method and combine it with self-supervised methods, the results is not very satisfied. ViT has achieves much better results especially when it utilizes a larger dataset (JFT-300). However, the structure of ViT is basically the same as the transformer in NLP, how to explicit the correlations in the intra-patch and inter-patch is still a challenging problem. Besides, the patches with the same size are treated equal in ViT. As we all know, the complexity of each patch is different and this characteristic has not been fully employed now.

4.2. High-level Vision

Recently there have been increasing interests for adopting transformer for high-level computer vision tasks, such as object detection [15, 155, 23], lane detection [81] and segmentation [129, 126]. In this section we make a review for these methods.

4.2.1 Object Detection

According to the modules that adopt transformer architecture, transformer-based methods for object detection could be coarsely categorized into neck-based, head-based and framework-based methods.

Multi-scale feature fusion modules (as known as *neck* in modern detection framework) like Feature Pyramid Network (FPN) [77] have been widely used in object detection for better detection performance. Zhang *et al*. [145] suggest that traditional methods failed to interact the cross-scale features, and thus propose Feature Pyramid Transformer (FPT) to fully exploit feature interactions across both space and scales. FPT consists of three types of transformer, *i.e*., self-transformer, grounding transformer and rendering transformer, which encode the information of self-level, top-down and bottom-up paths of the feature pyramid, respectively. FPT basically utilizes the self-attention module in transformer to enhance the feature fusion for feature pyramid networks.

The prediction head plays an important role for object detectors. Prior detection methods usually exploit a single visual representation (*e.g*., bounding box and corner point) for predicting the final results. Chi *et al*. [23] propose the bridging visual representations (BVR) to combine different heterogeneous representations into a single one via a multi-head attention module. Specifically, the master representation is treated as the query input and the auxiliary representation is regarded as the key input. Through an attention module similar as that in transformer, the enhanced features for the master representation can be obtained, which bridge the information from the auxiliary representation and benefit the final detection performance.

Unlike the above methods that utilize transformer the enhance the specific modules for modern detectors, Carion [15] redesign the framework of object detection and proposed detection transformer (DETR), which is a simple and fully end-to-end object detector. DETR treats the object detection task as an intuitive set prediction problem and get rids of traditional hand-crafted components like

Table 3. Experimental results comparison between iGPT and ViT. The results are from [18] and [31].

Method	Network	Pre-training Policy	Pre-training data	Dataset	Params (M)	Acc (Top-1)
iGPT	Transformer	Self-supervised	ImageNet	CIFAR-10	1362	99.00
				CIFAR-100	1362	91.70
ViT	Transformer	Supervised	ImageNet	CIFAR-10	86	98.13
				CIFAR-100	86	87.13
				ImageNet	86	77.91
			JFT-300M	CIFAR-10	86	99.00
				CIFAR-100	86	91.87
				ImageNet	86	84.15
				CIFAR-10	632	99.50
				CIFAR-100	632	94.55
				ImageNet	632	88.36
BiT-L	CNN	Supervised	JFT-300M	CIFAR-10	60	99.37
				CIFAR-100	60	93.51
				ImageNet	60	87.54

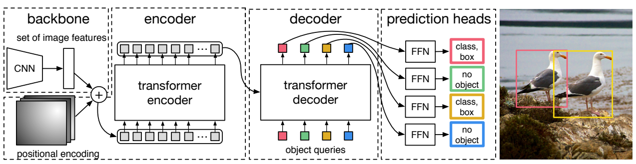

Figure 6. The overall architecture of DETR (Figure from [15]).

anchor generation and non-maximum suppression (NMS) post-processing. As shown in Fig. 6, DETR starts with an CNN backbone to extract features from the input image. To supplement the image features with position information, the fixed positional encodings are added to the flatten features before fed into the encode-decoder transformer. The transformer decoder consumes the embeddings from encoder along with N learnt positional endcodings (object queries), and produces N output embeddings, where N is a predefined parameter and is typically larger than the number of objects in an image. The final predictions are computed with simple feed-forward networks (FFN), which include the bounding box coordinates and class labels to indicate the specific class of the object or no object. Unlike original transformer that produce predictions sequentially, DETR decodes N objects in parallel at the same time. DETR employs an bipartite matching algorithm to assign the predicted and groud-truth objects. As shown in Eq. (11), the Hungarian loss is exploited to compute the loss function for all matched pairs of objects.

$$\mathcal{L}_{\text{Hungarian}}(y,\hat{y}) = \sum_{i=1}^{N}\left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i\neq\varnothing\}}\mathcal{L}_{\text{box}}(b_i,\hat{b}_{\hat{\sigma}}(i))\right], \tag{11}$$

where y and $\hat{y}$ are the ground truth and prediction of objects respectively, $\hat{\sigma}$ is the optimal assignment, c_i and $\hat{p}_{\hat{\sigma}(i)}(c_i)$ are the target class label and predicted label, b_i and $\hat{b}_{\hat{\sigma}}(i))$ are ground truth and predicted bounding box, respectively. DETR shows impressive performance on object detection, with comparable accuracy and speed with the popular and well-established Faster R-CNN baseline on COCO benchmark.

DETR is a new design for object detection framework based on transformer and sheds light on the community to develop fully end-to-end detectors. However, the vanilla DETR also comes with several challenges, *e.g.*, longer training schedule and poor performance for small objects. Deformable DETR proposed by Zhu *et al.* [155] is an popular method for address the above issues and greatly improves the detection performance. Instead of looking over all spatial locations on image feature maps by original multi-head attention in transformer, the deformable attention module is proposed to attend to a small set of key positions around a reference point. In this way, the computational complexity is greatly reduced and also benefit to fast convergence. More importantly, the deformable attention module can be easily applied for fusing multi-scale features. Deformable DETR achieves better performance than DETR with $10\times$ less training cost and $1.6\times$ faster inference speed. Some additional improvements are also applied for Deformable DETR, including an effective iterative bounding box refinement method and a two-stage scheme, which result in further performance gain.

Aiming at the high computation complexity problem of DETR, Zheng *et al.* [153] propose a Adaptive Clustering Transformer (ACT) to reduce the computation cost of the pre-trained DETR without any training process. ACT adaptively clusters the query features using a locality sensitivity hashing method and broadcast the attention output to the queries represented by the selected prototypes. By replacing the self-attention module of the pre-trained DETR model with the proposed ACT without any re-training, the computation cost can be considerably reduced with suffering little accuracy degradation. Additionally, the performance drop can be further reduced by utilizing a multi-task knowledge distillation (MTKD) method, which exploit the original transformer to distill the ACT module with a few epochs of fine-tuning.

Sun *et al*. [117] investigate the slow convergence problem of DETR model and reveal that the cross-attention module in the transformer decoder is the main reason behind. To this end, an encoder-only version of DETR is proposed and achieves considerably improvement for the detection accuracy and training convergence. Moreover, a new bipartite matching scheme is designed for more stable training and faster convergence. Two transformer-based set prediction models are proposed to improve encoder-only DETR with feature pyramids, namely TSP-FCOS and TSP-RCNN, which achieve better performance than the original DETR model.

Inspired by the pre-training transformer scheme in natural language processing, Dai *et al*. [28] propose a method for unsupervisedly pre-train DETR (UP-DETR) for object detection. Specifically, a novel unsupervised pretext task named random query patch detection is proposed to pre-train the DETR model. With this scheme, UP-DETR improve the detection accuracy on a relatively small dataset, *i.e*., PASCAL VOC, by a large margin. On the COCO benchmark with sufficient training data, UP-DETR still outperform DETR, which demonstrate the effectiveness of unsupervised pre-training scheme.

4.2.2 Segmentation

DETR [15] can be naturally extended for panoptic segmentation task by appending a mask head on the decoder and achieve competitive results. Wang *et al*. [126] propose Max-DeepLab to directly predict panoptic segmentation results with a mask transformer, without surrogate sub-tasks like box detection. Similar with DETR, Max-DeepLab streamlines the panoptic segmentation tasks in an end-to-end fashion and directly predicts a set of non-overlapping masks and corresponding labels. A panoptic quality (PQ) style loss is utilized for training the model. Additionally, unlike prior methods that stack transformer on top of a CNN backbone, Max-DeepLab employs a dual-path framework for better combining CNN with transformer.

Wang *et al*. [129] propose a transformer-based video instance segmentation (VisTR) model, which takes a sequence of images as input and produces corresponding instance prediction results. An instance sequence matching strategy is proposed to assign the predictions with ground truth. To obtain the mask sequence for each instance, VisTR utilizes the instance sequence segmentation module to accumulate the mask features from multiple frames and segment the mask sequence with 3D CNN.

There is also a attempt to use transformer for cell instance segmentation [95], which is based on the DETR panoptic segmentation model. The proposed Cell-DETR additionally adds skip connections to bridge feature from backbone CNN and the CNN decoder in segmentation head to obtain better fused features. Cell-DETR shows state-of-the-art performance for cell instance segmentation from microscopy imagery.

Zhao *et al*. [150] design a novel transformer architecture (Point Transformer) for processing point clouds. The proposed self-attention layer is invariant to the permutation of the point set and thus is suitable for the point set processing tasks. Point Transformer shows strong performance for semantic segmentation task from 3D point cloud.

4.2.3 Lane Detection

Based on PolyLaneNet [119], Liu *et al*. [81] propose to improve performance of curve lane detection by learning the global context with transformer network. Similar to PolyLaneNet, the proposed method (LSTR) regards the lane detection as a task of fitting lanes with polynomials and uses neural networks to predict the parameters of polynomials. To capture long and thin structures for lanes and the global context, LSTR introduces transformer network into the architecture to process low-level features extracted by convolutional neural networks. Besides, LSTR uses Hungarian Loss to optimize network parameters. As is shown in [81], LSTR achieves 2.82% higher accuracy and $3.65\times$ FPS than PolyLaneNet with only $0.2\times$ parameters. The combination of transformer network, convolutional neural network and Hungarian Loss realizes a tiny, fast and precise lane detection framework.

4.3. Low-level Vision

Besides high-level vision tasks, few works apply transformers on low-level vision fields, such as image super-resolution, generation, *etc*. Compared with classification, segmentation and detection whose outputs are labels or boxes, the low-level tasks often take images as outputs (*e.g.*, high-resolution or denoised images), which is more challenging.

Parmar *et al*. [92] take the first step at generalizing the transformer model to formulate image translation and generation task and propose Image Transformer. The Image Transformer consists of two parts: an encoder for extracting image representation and a decoder to generate pixels. For each pixel with value $0-255$, a $256 \times d$ dimensional embedding is learned for encoding each value into a d dimensional vector, which is taken as the input of the encoder. The architecture of encoder and decoder are the same as that in [123]. The detailed structure of each layer in decoder is

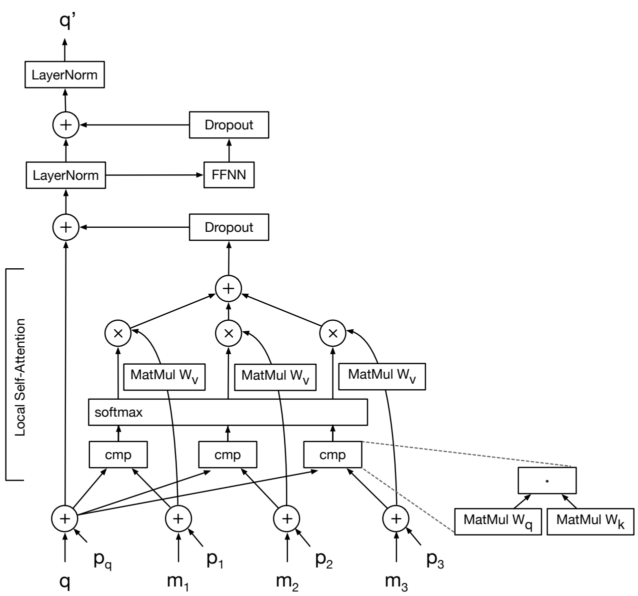

Figure 7. Diagram of one layer in the decoder of Image Transformer. Each output pixel q' is generated by previously generated pixels $m_1, m_2, ...$ and the corresponding input pixel q. (Figure from [92])

shown in Figure 7. Each output pixel q' is generated by calculating self-attention between the input pixel q and previously generated pixels $m_1, m_2, ...$ with position embedding $p_1, p_2, ...$ For image-conditioned generation, such as super-resolution and inpainting, an encoder-decoder architecture is used, where the input of encoder is the low-resolution images or the corrupted images. For unconditional and class-conditional generation (*i.e.*, noise to image), only decoder is used for inputting noise vectors. Since the input for the decoder is the former generated pixels which will bring large computation cost when producing high-resolution images, a local self-attention scheme is proposed, which only use the closest generated pixels as inputs for the decoder. As a result, the Image Transformer can achieve competitive performance with the CNN-based models on image generation and translation tasks, which shows the effectiveness of transformer-based models on low-level vision task.

Compared with using each pixel as inputs of transformer models, recent works use patches (set of pixels) as inputs. Yang *et al*. [135] propose Texture Transformer Network for Image Super-Resolution (TTSR). They use the transformer architecture in the references-based image super-resolution problem, which aims to transfer relevant textures from reference images to the low-resolution images. Taking the low-resolution image and reference image as the query Q and key K, the relevance $r_{i,j}$ is calculated between each patch q_i in Q and k_i in K by

$$r_{i,j} = < \frac{q_i}{\|q_i\|}, \frac{k_i}{\|k_i\|} > . \tag{12}$$

Then a hard-attention module is proposed to select high-resolution features V according to the reference image to match the low-resolution image by using the relevance. The hard-attention map is calculated by

$$h_i = \arg\max_j r_{i,j} \tag{13}$$

Then the most relevant reference patch from is $t_i = v_{h_i}$, where t_i in T is the transferred features. After that, a soft-attention module is used to transfer V to the low-resolution feature F. The soft-attention can be calculated by

$$s_i = \max_j r_{i,j}. \tag{14}$$

Therefore, the equation to transfer the high-resolution texture images to the low-resolution images can be formulated as:

$$F_{out} = F + Conv(Concat(F, T)) \odot S, \tag{15}$$

where F_{out} and F denotes the output and input features of the low-resolution image, S is the soft-attention and T is the transferred features from the high-resolution texture image. By introducing the transformer based architecture, TTSR can successfully transfer the texture information from the high resolution referenced images to the low-resolution image for super-resolution task.

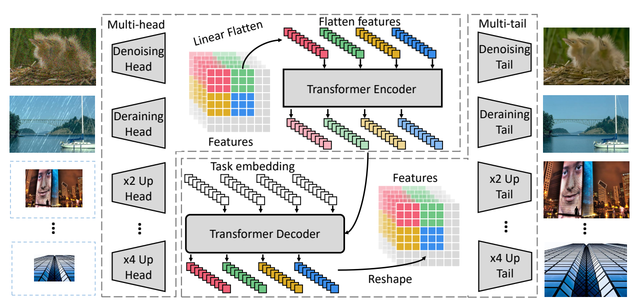

Figure 8. Diagram of the architecture of IPT. They use multi-head and multi-tail structure to handling different image processing tasks. (Figure from [17])

The above methods use transformer models on single task, while Chen *et al*. [17] propose Image Processing Transformer (IPT) to fully utilize the advantages of transformers by using large scale pre-training and achieve the state-of-the-art performance in several image processing tasks including super-resolution, denoising and deraining. As shown in Figure 8, IPT consists of multi-heads, an encoder, a decoder and multi-tails. The multi-head and multi-tail structure and task embeddings is introduced for different image processing tasks. The features are divided into patches to put into the encoder-decoder architecture and then the outputs are reshape to features with the same size. As transformer models show advantages on large scale pre-training, IPT use the ImageNet dataset for pre-training. Specifically, Images from ImageNet dataset are downgraded to generated corrupted images by manually adding noising, rainstreaks or downsampling. The degraded images are then taken as the inputs of IPT and the clean images are taken as the optimization goal of the outputs. A self-supervised method is also introduced to enhance the generalization ability of IPT model. The trained model is then fine-tuned on each task using the corresponding head, tail and task embedding. IPT largely improve the performance on the image processing tasks (*e.g.*, 2dB in image denoising task), which demonstrate the huge potential of transformer-based model on low-level vision fields.

4.4. Video Processing

The transformer performs surprisingly well on sequence-based tasks, especially on NLP tasks. In computer vision, spatial and temporal dimension information is favored in video tasks. Therefore, the transformer is applied to a number of video tasks, such as frame synthesis [83], action recognition [41], and video retrieval [80].

4.4.1 High-level Video Processing

Human Action Recognition. Video human action task refers to identify and localize human actions in videos. The contextual stuff plays a critical role in recognizing human actions. Rohit *et al.* proposes the action transformer [41] to model the underlying relationship between the human of interest and the surrounding stuff. In specific, the I3D is used as the backbone to extract high-level feature maps. The extracted features from intermediate feature maps by ROIPooling is viewed as the query (Q). The key (K), values (V) are calculated from the intermediate features. The self-attention mechanism is conducted on three components and outputs the classification and regressions predictions. Lohit *et al.* [84] proposes an interpretable differentiable module, named temporal transformer network to reduce the intra-class variance and increases the inter-class variance. Fayyaz and Gall propose a temporal transformer to perform the action recognition task under weakly supervised settings.

Face Alignment. The video-based face alignment task aims to localize the facial landmarks. The temporal dependency and spatial information are important to the final performance. However, the former methods failed to capture both the temporal information across consecutive frames and the complementary spatial information on a still frame. Therefore, Liu *et al.* [80] uses the two-stream transformer networks to separately learn the temporal and spatial features. Two streams are jointly optimized in an end-to-end manner and the features are weighted to get the final predictions.

Video Retrieval. The key to content-based video retrieval is to find the similarity between videos. To overcome the shortcomings by leveraging only the image-level of video-level features, Shao *et al.* [110] suggests to use the transformer to model the longe-range semantic dependency. Moreover, the supervised contrastive learning strategy is introduced to perform hard negative mining. The results on benchmark datasets demonstrate the performance and speed advantages. Gabeur *et al.* [39] presents a multi-modal transformer to learn different cross-modal cues so as to represent the video.

Activity Recognition. The activity recognition refers to identify the activity of a human of a group. The former methods to solve this problem are based on the individual actors' location. Gavrilyuk *et al.* proposes an actor-transformer [40] architecture to learn the representation. The actor-transformer takes the static and dynamic representations generated by the 2D and 3D networks as input. The output of the transformer is the predicted activity.

Video Object Detection. To detect the objects from a video, global and local information are required. Chen *et al.* introduces the memory enhanced global-local aggregation (MEGA) [19] to capture more content. The representative features enhance the overall performance and address the *ineffective* and *insufficient* problems. Yin *et al.* [138] proposes a spatiotemporal transformer to aggregate the spatial and temporal information. Together with another spatial feature encoding component, these two components perform well on 3D video object detection task.

Multi-task Learning. The untrimmed video usually contains many frames irrelevant to the target tasks. Therefore, it is crucial to mine the relevant information and remove the redundant information. To cope with the multi-task learning on untrimmed videos, Seong *et al.* adopts the video multi-task transformer network [109] to extract the information. For the CoVieW dataset, the tasks are scene recognition, action recognition and importance score prediction. Two pre-trained networks on ImageNet and Places365 extract the scene features and object features. The Multi-task transformers are stacked to fuse the features with the help of the class conversion matrix (CCM).

4.4.2 Low-level Video Processing

Frame/Video Synthesis. The frame synthesis task refers to synthesize the frames between two continuing frames or after a frame sequence. The video synthesis task aims to synthesize a video. Liu *et al.* proposes the ConvTransformer [83] which comprises five components: feature embedding, position encoding, encoder, query decoder, and the synthesis feed-forward network. Compared with the LSTM based works, the ConvTransformer achieves superior results with more parallelizable architecture. Schatz *et al.* [108] uses a recurrent transformer network to synthesis human actions from novel views.

Video Inpainting. The video inpainting task aims to complete the missing regions in a frame. This challenging task requires to merge information along the spatial and temporal dimensions. Zeng *et al.* proposes a spatial-temporal transformer network [144] for this task. All the input frames are taken as input and they are filled in parallel. The spatial-temporal adversarial loss is used to optimize the transformer network.

4.4.3 Multimodality

Video Captioning/Summarization. The goal of the video caption task is to generate text for the untrimmed videos. The event detection and description modules are two main parts. Zhou *et al.* [154] proposes an end-to-end optimized transformer to solve the dense video captioning task. The encoder transforms the video into representations. The proposal decoder generates event proposals from the encoding. The caption decoder masks the coding with the proposal and outputs the description. Bilkhu *et al.* [9] uses the C3D and I3D network to extract features and use a transformer to generate the predictions. The algorithm performs well on both single and dense summarization tasks. Li *et al.* [71] exploits the transformer based on the EnTangled Attention (ETA) module to tackle the image captioning task. Sun *et al.* [29] proposes a visual-linguistic framework to learn the representations without supervision. The model can be applied to a number of tasks, including video captioning, action classification and *etc*.

4.5. Self-attention for Computer Vision

In the above sections, we have reviewed the methods using transformer architecture for visual tasks. Self-attention is the key part of transformer. In this section, we delve deep into the self-attention based methods for challenging tasks in computer vision, *e.g.*, semantic segmentation, instance segmentation, object detection, keypoint detection and depth estimation. We start with formulating the algorithm of self-attention in Section 4.5.1 and summarize the existing applications using self-attention for computer vision in Section 4.5.2.

4.5.1 General Formulation of Self-attention

Self-attention module [123] for machine translation computes the responses at one position in a sequence by attending to all positions and weighted summing them up accordingly in an embedding space, which can be viewed as a form of non-local filtering operations [128, 11] that is applicable in computer vision. We follow the convention [128] to formulate the self-attention module. Given an input signal (*e.g.*, image, sequence, video and feature) $X \in \mathbb{R}^{n\times c}$, where $n = h \times w$ indicates the number of pixels in feature, and c is the number of channels, the output signal is generated as:

$$y_i = \frac{1}{C(x_i)} \sum_{\forall j} f(x_i, x_j) g(x_j), \tag{16}$$

where $x_i \in \mathbb{R}^{1\times c}$ and $y_i \in \mathbb{R}^{1\times c}$ indicate the i^{th} position (*e.g.*, space, time and spacetime) of the input signal X and output signal Y, respectively. Subscript j is the index that enumerates all positions. A pairwise function $f(\cdot)$ computes a representing relationship such as affinity between i and all j. The function $g(\cdot)$ computes a representation of the input signal at the position j. The response is normalized by a factor $C(x_i)$.

Note that there are many choices for the pairwise function $f(\cdot)$, for example, a simple extension of the Gaussian function can be used to compute the similarity in an embedding space, thus the function $f(\cdot)$ can be formulated as:

$$f(x_i, x_j) = e^{\theta(x_i)\phi(x_j)^T} \tag{17}$$

where $\theta(\cdot)$ and $\phi(\cdot)$ can be any embedding layers. If we consider the $\theta(\cdot), \phi(\cdot), g(\cdot)$ in the form of linear embedding: $\theta(X) = XW_\theta$, $\phi(X) = XW_\phi, g(X) = XW_g$ where $W_\theta \in \mathbb{R}^{c\times d_k}, W_\phi \in \mathbb{R}^{c\times d_k}, W_g \in \mathbb{R}^{c\times d_v}$, and set the normalization factor as $C(x_i) = \sum_{\forall j} f(x_i, x_j)$, the E.q. 16 can be rewritten as:

$$y_i = \frac{e^{x_i w_{\theta,i} w_{\phi,j}^T x_j^T}}{\sum_j e^{x_i w_{\theta,i} w_{\phi,j}^T x_j^T}} x_j w_{g,j}, \tag{18}$$

where $w_{\theta,i} \in \mathbb{R}^{c\times 1}$ is the i^{th} row of the weight matrix W_θ. For a given index i, $\frac{1}{C(x_i)} f(x_i, x_j)$ becomes the softmax output along the dimension j, thus the formulation can be further rewritten as:

$$Y = softmax(XW_\theta W_\phi^T X) g(X), \tag{19}$$

where $Y \in \mathbb{R}^{n\times c}$ is the output signal of the same size as X. Compared to the query, key and value representations $Q = XW_q, K = XW_k, V = XW_v$ from translation module, once $W_q = W_\theta, W_k = W_\phi, W_v = W_g$, E.q. 19 can be formulated as:

$$Y = softmax(QK^T)V = Attention(Q, K, V), \tag{20}$$

where the self-attention module [**?**] proposed for machine translation is exactly the same as above non-local filtering operations proposed for computer vision.

Generally, the final output signal of self-attention module for computer vision will be wrapped as:

$$Z = YW_Z + X \tag{21}$$

where Y is generated through E.q. 19. If W_Z is initialized as zero, this self-attention module can be inserted to any existing model without breaking its initial behavior.

4.5.2 Applications on Visual Tasks

Self-attention module is considered a building block of the convolutional neural network architectures, which have low scaling properties concerning the large receptive fields. The building block is always used on top of the networks to capture long-range interactions for computer vision tasks. In what follows, we review the proposed self-attention based methods for image based tasks, such as image classification, semantic segmentation and object detection.

Image Classification. Trainable attention for classification consists of two main streams: hard attention [3, 87, 134] regarding the use of an image region and soft attention [125, 60, 43, 102] generating non-rigid feature maps. Ba *et al*. [3] first proposes the visual attention term for the image classification task, and use attention to select relevant regions and locations within the input image, which can also reduce the computational complexity of proposed model w.r.t. the size of input image. AG-CNN [42] proposes to crop a sub-region from global image by the attention heat map for medical image classification. Instead of using hard attention and recalibrating the crop of feature maps, SENet [54] proposes soft self-attention to reweight the channel-wise responses of the convolutional features. Jetley *et al*. [60] uses attention maps generated by corresponding estimators to reweight intermediate features in deep neural networks. Han *et al*. [43] utilize the attribute-aware attention to enhance the representation of CNNs.

Semantic Segmentation. PSANet [151], OCNet [139], DANet [38] and CFNet [147] are the first works that introduce the self-attention module to semantic segmentation task, which consider and augment the relation and similarity [146, 74, 46, 89, 130] between the contextual pixels. DANet [38] simultaneously leverages self-attention module on spatial and channel dimensions. A^2Net [20] proposes to group the pixels into a set of regions, and then augment the pixel representations by aggregating the region representations with the generated attention weights. To alleviate the huge amount of parameters brought by calculating pixel similarities in self-attention module, several works [140, 59, 58, 75, 66] are proposed to improve the efficiency of the self-attention module for semantic segmentation. For example, CGNL [140] applies the Taylor series of the RBF kernel function to approximate the pixel similarities. CCNet [59] approximates the original self-attention scheme via two consecutive criss-cross attention. ISSA [58] proposes to factorize the dense affinity matrix as the product of two sparse affinity matrices. There are other related works using attention based graph reasoning module [76, 21, 75] to enhance both the local and global representations.

Object Detection. Ramachandran *et al*. [102] proposes an attention-based layer to build a fully attentional model that outperforms the convolutional RetinaNet [78] on COCO [79] benchmark. GCNet [13] finds that the global contexts modeled by non-local operations are almost the same for different query positions within an image, and proposes to unify the simplified formulation and SENet [54] into a general framework for global context modeling [73, 52, 34, 93]. Vo *et al*. [124] designs a bidirectional operation to gather and distribute the information from a query position to all possible positions. Hu *et al*. [53] proposes a self-attention based relation module to process a set of objects simultaneously through interaction between their appearance feature. Cheng *et al*. proposes RelationNet++ [23] with an attention-based decoder module to bridge other representations into a typical object detector built on a single representation format.

Other Vision Tasks. Zhang *et al*. [148] proposes a resolution-wise attention module to learn enhanced resolution-wise feature maps for precise pose estimation. Huang *et al*. [57] proposes a transformer based network [56] for 3D hand-object pose estimation. Chang *et al*. [16] improves the accuracy and accelerates the convergence of keypoint detection model with the help of attention mechanism based feature fusion block.

4.6. Efficient Transformer

Though Transformer models has achieve success in various tasks, high memory and computing resource are still required, which block the implementation on resource-limited devices (*e.g*., mobile phones). In this section, we review the researches about compressing & accelerating transformer models for efficient implementation, including network pruning, low-rank decomposition, knowledge distillation, network quantization, compacting architecture design. Table 4 lists a few representative works for compressing transformer-based models.

Table 4. List of representative compressed transformer-based models. The data of the Table is from [98].

Models	Compress Type	#Layer	Params	Speed Up
BERT_{BASE} [29]	Baseline	12	110M	×1
ALBERT [67]	Decomposition	12	12M	×5.6
BERT-of-Theseus [133]	Architecture design	6	66M	×1.94
Q-BERT [111]	Quantization	12	-	-
Q8BERT [142]		12		
TinyBERT [62]	Distillation	4	14.5M	×9.4
DistilBERT [107]		6	6.6m	×1.63
BERT-PKD [116]		3∼6	45.7∼67M	×3.73∼1.64
MobileBERT [118]		24	25.3M	×4.0
PD [120]		6	67.5M	×2.0

4.6.1 Pruning & Decomposition

In transformer based pre-trained models (*e.g*., BERT), multiple attention operations are paralleled to independently model the relationship between different tokens [123, 29], while no all the heads are necessary for a specific task. Michel *et al*. [85] empirically observe that a large percentage of attention heads can be removed at test time without significantly impacting performance. The required numbers of heads vary across different layers and even one head is enough for some layers. Considering the redundancy on attention heads, importance scores are defined to estimate the influence of each head on the final output in [85], and unimportant heads can be removed for efficient deployment. Dalvi *et al*. [96] further analyzes the redundancy in pretrained transformer models from two perspectives, *i.e*., general redundancy and task-specific redundancy. Following the lottery ticket hypothesis *et al*. [36], Prasanna *et al*. [96] analyzes the lotteries in BERT and show that good sub-networks also exist in Transformer-based models. Both the FFN layers and attention heads are reduced in [96] to achieve high compression rates.

Besides the width of transformer models, the depth, *i.e*., the number of layers can also be reduced to accelerate the inference process [32]. Different from that different attention heads in transformer models can be computed in parallel, different layers have to be calculated sequentially due to that the input of the next layer depends on the output of previous layers. Fan *et al*. [32] proposes a layer-wisely dropping strategy to regularize the training of models, and then the whole layers are removed together at the test phase. Considering that the available resources in different devices may vary, Hou *et al*. [51] propose to adaptively reduce the width and depth of the pre-defined transformer models, and obtains multiple models with different sizes simultaneously. Important attention heads and neurons are shared across different sub-networks via a rewiring mechanism.

Beyond the pruning methods that directly discarding part modules in transformer models, matrix decomposition aims to approximate the large matrices with multiple small matrices based on the low-rank assumption. For example, Wang *et al*. [131] decompose the standard matrix multiplication in the transformer models and achieve more efficient inference.

4.6.2 Knowledge Distillation

Knowledge distillation aims to train student networks by transferring knowledge from giant teacher networks [48, 12, 2]. Compared to the teacher networks, student networks usually have thinner and shallower architectures, which are easier to be deployed on resource-limited resources. Both the output and intermediate features of neural networks can also be used to transfer effective information from teacherd to students. Focused on Transform models, Mukherjee *et al*. [88] use the pre-trained BERT [29] as the teacher to guide the training of small models, with the help of large

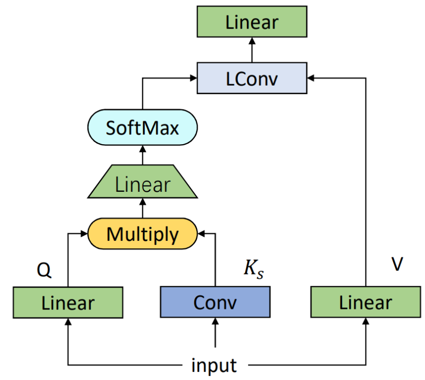

Figure 9. Span-based dynamic convolution in [61]. LConv denotes the light-weight depth-wise convolution.

amounts of unlabeled data. Wang *et al*. [127] train the student networks to mimick outputs of self-attention layers in the pre-trained teacher models. The dot-product between values is introduced as a new form of knowledge for guiding students. A teacher assistant [86] is also introduced in [127], which reduces the gap between the large pre-trained Transformer models and compact student networks to make the mimicking takes more easily. Considering various types of layers in the transformer model (*i.e*., self-attention layer, embedding layer, prediction layers), Jiao *et al*. [62] design different objective functions to transfer knowledge from teachers to students. For examples, the outputs of the embedding layers of the students models are to imitate the those of teachers via MSE losses. A learnable linear transformation is also imposed to map different features into a same space. For the output of prediction layers, the KL-divergence is adopted to measure the difference between different models.

4.6.3 Quantization

Quantization aim to reduce the number of bits to represent network weight or intermediate features [122, 137]. Quantization methods for general neural networks have been well discussed and achieve performance comparable to the original networks [91, 37, 6]. Recently, how to specially quantize transformer models has been attracted much attention [8, 33]. Shridhar *et al*. [112] suggest to embed the input into binary high-dimensional vectors, and then the binary input representation are used to train the binary neural networks. Cheong *et al*. [22] represent the weights in the transformer models by low-bit (*e.g*., 4-bit) representation. Zhao *et al*. [152] empirically investigate various quantization methods and show that k-means quantization has a huge development potential. Aimed at the machine translation task, Prato *et al*. [97] proposes a fully quantized transformer, which is the first 8-bit quality model without any loss in translation quality as claimed in the paper.

4.6.4 Compact Architecture Design

Beyond compressing pre-defined transformer models to small ones, some works try to design compact models directly [132, 61]. Jiang *et al*. [61] simplify the calculation of self-attention by proposing a new module named as span-based dynamic convolution, which combing the full-connected layers and convolutional layers, as shown in Figure 9. The local dependence between representation from different tokens is calculated by convolution operations, which is much efficient than the dense full-connected layers in the standard transformers. Depth-wise convolution is also used to further reduce the computing cost. Interesting hamburger layers are proposed in [1], which use matrix decomposition to substitute the original self-attention layers. Matrix decomposition can be calculated more efficiently than the standard self-attention operations while reflect the dependence between different tokens well.

The self-attention operation in Transformer models calculate the dot product between representations from different input tokens in a given sequence (patches in image recognition task [31]), whose complexity is $O(N)$, where N is the length of the sequence. Recently, massive methods focus on reducing the complexity to $O(N)$ to make transformer models scalable to long sequences. For example, Katharopoulos *et al*. [64] approximate the self-attention as a linear dot-product of kernel feature maps and reveal the relationship between tokens via recurrent neural networks. Zaheer *et al*. [143] treat each token as a vertex in a graph and the inner product calculation between two tokens is denoted an edge. Inspired graph theories [113, 25], various sparse graph are combined together to approximate the dense graph in the transformer model, which also achieve $O(N)$ complexity. From a theoretical perspective, Yun *et al*. [141] prove that a sparse transformer with $O(N)$ complexity is enough to reflect any kind of relationship between tokens and can make universal approximation, which provides theoretical guarantees for further research about transformer with $O(N)$ complexity.

5. Conclusions and Future Prospects

Transformer is becoming a hot topic in computer vision area due to its competitive performance and tremendous potential compared to convolutional neural networks. To discover and utilize the power of transformer, as summarized in the survey, a number of solutions have been proposed in recent years. These methods show excellent performance on a wide range of visual tasks, including basic image classification, high-level vision, low-level vision and video processing. Nevertheless, the potential of transformer for computer vision is yet not fully explored and several challenges are remaining to be solved.

Although researchers have proposed many transformer-based models to tackle computer vision tasks, these works are the initiatory solutions and have much room for improvement. For example, the transformer architecture in ViT [31] follows the standard transformer for NLP [123]. The improved version specially for CV remains to be explored. Moreover, the applications of transformer on more tasks in addition to those mentioned above are also required.

Besides, most of the existing visual transformer models are designed to handle single task. Many NLP models such as GPT-3 [10] have shown the ability of transformer to deal with multiple tasks in one model. IPT [17] in CV area is also able to process multiple low-level vision tasks, such as super-resolution, image denoising and deraining. We believe that more tasks can be involved in only one model.

Last but not the least, developing efficient transformer models for CV is also an open problem. Transformer models are usually huge and computation-expensive, *e.g*., the base ViT model [31] requires 18B FLOPs to process an image. As a contrast, the lightweight CNN model GhostNet [44, 45] can achieve similar performance with only about 600M FLOPs. Although several methods have been proposed to compress transformer, their complexities are still large. And these methods which are originally designed for NLP may not be suitable for CV. Thus, efficient transformer models are agent to deploy visual transformer on resource-limited devices.

References

[1] Anonymous. Is attention better than matrix decomposition? In *Submitted to International Conference on Learning Representations*, 2021. under review. 15

[2] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27:2654–2662, 2014. 14

[3] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. In *International Conference on Learning Representations*, 2014. 13

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1

[6] Yu Bai, Yu-Xiang Wang, and Edo Liberty. Proxquant: Quantized neural networks via proximal operators. *arXiv preprint arXiv:1810.00861*, 2018. 14

[7] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019. 5

[8] Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. Efficient 8-bit quantization of transformer neural machine language translation model. *arXiv preprint arXiv:1906.00532*, 2019. 14

[9] Manjot Bilkhu, Siyang Wang, and Tushar Dobhal. Attention is all you need for videos: Self-attention based video summarization using universal transformers. *arXiv preprint arXiv:1906.02792*, 2019. 12

[10] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1, 5, 15

[11] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 60–65, 2005. 12

[12] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 14

[13] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 13

[14] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 2

[15] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 7, 8, 9

[16] Yuan Chang, Zixuan Huang, and Qiwei Shen. The same size dilated attention network for keypoint detection. In *International Conference on Artificial Neural Networks*, pages 471–483, 2019. 13

[17] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020. 2, 10, 15

[18] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 1, 2, 6, 8

[19] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10337–10346, 2020. 11

[20] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. *Advances in neural information processing systems*, pages 352–361, 2018. 13

[21] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 433–442, 2019. 13

[22] Robin Cheong and Robel Daniel. transformers. zip: Compressing transformers with pruning and quantization. Technical report, tech. rep., Stanford University, Stanford, California, 2019. 14

[23] Cheng Chi, Fangyun Wei, and Han Hu. Relationnet++: Bridging visual representations for object detection via transformer decoder. *Advances in Neural Information Processing Systems*, 2020. 7, 13

[24] Yung-Sung Chuang, Chi-Liang Liu, and Hung-Yi Lee. Speechbert: Cross-modal pre-trained language model for end-to-end spoken question answering. *arXiv preprint arXiv:1910.11559*, 2019. 5

[25] Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002. 15

[26] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 4

[27] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020. 5

[28] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UP-DETR: unsupervised pre-training for object detection with transformers. *arXiv preprint arXiv:2011.09094*, 2020. 2, 9

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019. 1, 4, 5, 12, 13, 14

[30] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13063–13075, 2019. 5

[31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 6, 8, 15

[32] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019. 14

[33] Chaofei Fan. Quantized transformer. Technical report, Technical report, Stanford University, Stanford, California, 2019. URL https 14

[34] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2020. 13

[35] Robert B Fisher. Cvonline: The evolving, distributed, non-proprietary, on-line compendium of computer vision. *Re-*

trieved January 28, 2006 from http://homepages. inf. ed. ac. uk/rbf/CVonline, 2008. 2

[36] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 14

[37] Joshua Fromm, Meghan Cowan, Matthai Philipose, Luis Ceze, and Shwetak Patel. Riptide: Fast end-to-end binarized neural networks. *Proceedings of Machine Learning and Systems*, 2:379–389, 2020. 14

[38] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 13

[39] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision (ECCV)*, pages 214–229, 2020. 11

[40] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 839–848, 2020. 11

[41] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019. 11

[42] Qingji Guan, Yaping Huang, Zhun Zhong, Zhedong Zheng, Liang Zheng, and Yi Yang. Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification. In *arXiv preprint arXiv:1801.09927*, 2018. 13

[43] Kai Han, Jianyuan Guo, Chao Zhang, and Mingjian Zhu. Attribute-aware attention model for fine-grained representation learning. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 2040–2048, 2018. 13

[44] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020. 15

[45] Kai Han, Yunhe Wang, Qiulin Zhang, Wei Zhang, Chunjing Xu, and Tong Zhang. Model rubik's cube: Twisting resolution, depth and width for tinynets. *Advances in Neural Information Processing Systems*, 33, 2020. 15

[46] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7519–7528, 2019. 13

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016. 1

[48] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 14

[49] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[50] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[51] Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33, 2020. 14

[52] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *Advances in Neural Information Processing Systems*, pages 2725–2734, 2019. 13

[53] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 13

[54] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 13

[55] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019. 5

[56] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation. In *European Conference on Computer Vision*, pages 17–33, 2020. 13

[57] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan. Hot-net: Non-autoregressive transformer for 3d hand-object pose estimation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3136–3145, 2020. 13

[58] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Interlaced sparse self-attention for semantic segmentation. *arXiv preprint arXiv:1907.12273*, 2019. 13

[59] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 603–612, 2019. 13

[60] Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip HS Torr. Learn to pay attention. In *International Conference on Learning Representations*, 2018. 13

[61] Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 14, 15

[62] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Nov. 2020. 2, 13, 14

[63] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. 4

[64] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 15

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 1

[66] Saumya Kumaar, Ye Lyu, Francesco Nex, and Michael Ying Yang. Cabinet: Efficient context aggregation network for low-latency semantic segmentation. *arXiv preprint arXiv:2011.00993*, 2020. 13

[67] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. 13

[68] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[69] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020. 5

[70] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. 5

[71] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8928–8937, 2019. 12

[72] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 5

[73] Wei Li, Kai Liu, Lizhe Zhang, and Fei Cheng. Object detection based on an adaptive attention mechanism. *Scientific Reports*, pages 1–13, 2020. 13

[74] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9167–9176, 2019. 13

[75] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. *Advances in Neural Information Processing Systems*, pages 9225–9235, 2018. 13

[76] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. *Advances in Neural Information Processing Systems*, pages 1853–1863, 2018. 13

[77] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 7

[78] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 13

[79] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755, 2014. 13

[80] Hao Liu, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Two-stream transformer networks for video-based face alignment. *IEEE transactions on pattern analysis and machine intelligence*, 40(11):2546–2554, 2017. 11

[81] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *WACV*, 2021. 7, 9

[82] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 4, 5

[83] Zhouyong Liu, Shun Luo, Wubin Li, Jingben Lu, Yufan Wu, Chunguo Li, and Luxi Yang. Convtransformer: A convolutional transformer network for video frame synthesis. *arXiv preprint arXiv:2011.10185*, 2020. 11

[84] Suhas Lohit, Qiao Wang, and Pavan Turaga. Temporal transformer networks: Joint learning of invariant and discriminative time warping. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12426–12435, 2019. 11

[85] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024, 2019. 2, 14

[86] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020. 14

[87] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. *Advances in neural information processing systems*, pages 2204–2212, 2014. 13

[88] Subhabrata Mukherjee and Ahmed Hassan Awadallah. Xtremedistil: Multi-stage distillation for massive multilingual models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2221–2234, 2020. 14

[89] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018. 13

[90] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, 2016. 1

[91] Eunhyeok Park and Sungjoo Yoo. Profit: A novel training method for sub-4-bit mobilenet models. In *European Conference on Computer Vision*, pages 430–446. Springer, 2020. 14

[92] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *ICML*, 2018. 2, 9, 10

[93] Hughes Perreault, Guillaume-Alexandre Bilodeau, Nicolas Saunier, and Maguelonne Héritier. Spotnet: Self-attention multi-task network for object detection. In *2020 17th Conference on Computer and Robot Vision (CRV)*, pages 230–237, 2020. 13

[94] Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*, 2019. 5

[95] Tim Prangemeier, Christoph Reich, and Heinz Koeppl. Attention-based transformers for instance segmentation of cells in microstructures. *arXiv preprint arXiv:2011.09763*, 2020. 9

[96] Sai Prasanna, Anna Rogers, and Anna Rumshisky. When bert plays the lottery, all tickets are winning. *arXiv preprint arXiv:2005.00561*, 2020. 14

[97] Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. Fully quantized transformer for machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1–14, 2020. 2, 14

[98] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *arXiv preprint arXiv:2003.08271*, 2020. 5, 13

[99] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. 5

[100] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 5, 6

[101] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 5

[102] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 13

[103] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[104] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957. 1

[105] FRANK ROSENBLATT. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961. 1

[106] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985. 1

[107] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 13

[108] Kara Marie Schatz, Erik Quintanilla, Shruti Vyas, and Yogesh Singh Rawat. A recurrent transformer network for novel view action synthesis. In *ECCV (27)*, pages 410–426, 2020. 11

[109] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Video multitask transformer network. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 11

[110] Jie Shao, Xin Wen, Bingchen Zhao, and Xiangyang Xue. Temporal context aggregation for video retrieval with contrastive learning. 11

[111] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. In *AAAI*, pages 8815–8821, 2020. 13

[112] Kumar Shridhar, Harshil Jain, Akshat Agarwal, and Denis Kleyko. End to end binarized neural networks for text classification. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 29–34, 2020. 14

[113] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011. 15

[114] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 5

[115] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7464–7473, 2019. 5

[116] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019. 13

[117] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection. *arXiv preprint arXiv:2011.10881*, 2020. 2, 9

[118] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*, 2020. 13

[119] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixão, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. *arXiv preprint arXiv:2004.10924*, 2020. 9

[120] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: The impact of student initialization on knowledge distillation. *arXiv preprint arXiv:1908.08962*, 2019. 13

[121] Shimon Ullman et al. *High-level vision: Object recognition and visual cognition*, volume 2. MIT press Cambridge, MA, 1996. 2

[122] Vincent Vanhoucke, Andrew Senior, and Mark Z Mao. Improving the speed of neural networks on cpus. 2011. 14

[123] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017. 1, 2, 3, 4, 9, 12, 14, 15

[124] Xuan-Thuy Vo, Lihua Wen, Tien-Dat Tran, and Kang-Hyun Jo. Bidirectional non-local networks for object detection. In *International Conference on Computational Collective Intelligence*, pages 491–501, 2020. 13

[125] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017. 13

[126] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan L. Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. *arXiv preprint arXiv:2012.00759*, 2020. 2, 7, 9

[127] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*, 2020. 14

[128] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 12

[129] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020. 2, 7, 9

[130] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12275–12284, 2020. 13

[131] Ziheng Wang, Jeremy Wohlwend, and Tao Lei. Structured pruning of large language models. *arXiv preprint arXiv:1910.04732*, 2019. 14

[132] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. *arXiv preprint arXiv:2004.11886*, 2020. 15

[133] Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. Bert-of-theseus: Compressing bert by progressive module replacing. *arXiv preprint arXiv:2002.02925*, 2020. 13

[134] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 13

[135] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5791–5800, 2020. 2, 10

[136] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019. 5

[137] Zhaohui Yang, Yunhe Wang, Kai Han, Chunjing Xu, Chao Xu, Dacheng Tao, and Chang Xu. Searching for low-bit weights in quantized neural networks. *arXiv preprint arXiv:2009.08695*, 2020. 14

[138] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11495–11504, 2020. 11

[139] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 13

[140] Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. In *Advances in Neural Information Processing Systems*, pages 6510–6519, 2018. 13

[141] Chulhee Yun, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. $o(n)$ connections are expressive enough: Universal approximability of sparse transformers. *arXiv preprint arXiv:2006.04862*, 2020. 15

[142] Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*, 2019. 13

[143] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*, 2020. 15

[144] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *European Conference on Computer Vision*, pages 528–543. Springer, 2020. 2, 12

[145] Dong Zhang, Hanwang Zhang, Jinhui Tang, Meng Wang, Xiansheng Hua, and Qianru Sun. Feature pyramid transformer. In *ECCV*, 2020. 7

[146] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6798–6807, 2019. 13

[147] Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-occurrent features in semantic segmentation. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 548–557, 2019. 13

[148] Kun Zhang, Peng He, Ping Yao, Ge Chen, Rui Wu, Min Du, Huimin Li, Li Fu, and Tianyao Zheng. Learning enhanced resolution-wise features for human pose estimation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2256–2260, 2020. 13

[149] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019. 5

[150] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. *arXiv preprint arXiv:2012.09164*, 2020. 9

[151] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018. 13

[152] Zihan Zhao, Yuncong Liu, Lu Chen, Qi Liu, Rao Ma, and Kai Yu. An investigation on different underlying quantization schemes for pre-trained language models. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 359–371. Springer, 2020. 14

[153] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020. 2, 8

[154] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018. 2, 12

[155] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 7, 8

[156] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015. 4