# Attention Mechanisms in Computer Vision: A Survey

Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R. Martin, Ming-Ming Cheng, *Senior Member, IEEE,* Shi-Min Hu, *Senior Member, IEEE,*

**Abstract**—Humans can naturally and effectively find salient regions in complex scenes. Motivated by this observation, attention mechanisms were introduced into computer vision with the aim of imitating this aspect of the human visual system. Such an attention mechanism can be regarded as a dynamic weight adjustment process based on features of the input image. Attention mechanisms have achieved great success in many visual tasks, including image classification, object detection, semantic segmentation, video understanding, image generation, 3D vision, multi-modal tasks and self-supervised learning. In this survey, we provide a comprehensive review of various attention mechanisms in computer vision and categorize them according to approach, such as channel attention, spatial attention, temporal attention and branch attention; a related repository https://github.com/MenghaoGuo/Awesome-Vision-Attentions is dedicated to collecting related work. We also suggest future directions for attention mechanism research.

**Index Terms**—Attention, Transformer, Survey, Computer Vision, Deep Learning, Salience.

## 1 Introduction

Methods for diverting attention to the most important regions of an image and disregarding irrelevant parts are called attention mechanisms; the human visual system uses one [1], [2], [3], [4] to assist in analyzing and understanding complex scenes efficiently and effectively. This in turn has inspired researchers to introduce attention mechanisms into computer vision systems to improve their performance. In a vision system, an attention mechanism can be treated as a dynamic selection process that is realized by adaptively weighting features according to the importance of the input. Attention mechanisms have provided benefits in very many visual tasks, e.g. image classification [5], [6], object detection [7], [8], semantic segmentation [9], [10], face recognition [11], [12], person re-identification [13], [14], action recognition [15], [16], few-show learning [17], [18], medical image processing [19], [20], image generation [21], [22], pose estimation [23], super resolution [24], [25], 3D vision [26], [27], and multi-modal task [28], [29].

In the past decade, the attention mechanism has played an increasingly important role in computer vision; Fig. 3, briefly summarizes the history of attention-based models in computer vision in the deep learning era. Progress can be coarsely divided into four phases. The first phase begins from RAM [31], pioneering work that combined deep neural networks with attention mechanisms. It recurrently predicts the important region and updates the whole network in an end-to-end manner through a policy gradient. Later, various works [21], [35] adopted a similar strategy for attention in vision. In this phase, recurrent neural networks(RNNs) were necessary tools for an attention mechanism. At the start of the second phase, Jaderberg et al. [32] proposed the STN which introduces a sub-network to predict an affine transformation used to to select important regions in the input. Explicitly predicting discriminatory input features is the major characteristic of the second phase; DCNs [7], [36] are representative works. The third phase began with SENet [5]

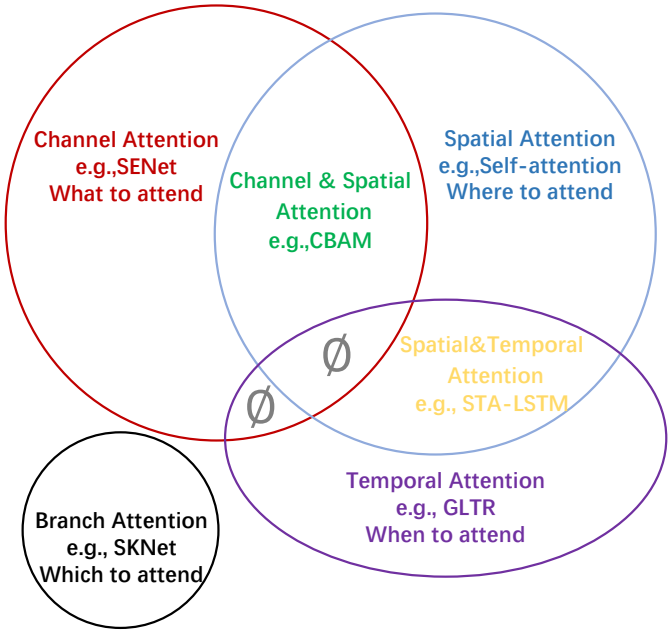



Fig. 1. Attention mechanisms can be categorised according to data domain. These include four fundamental categories of channel attention, spatial attention, temporal attention and branch attention, and two hybrid categories, combining channel & spatial attention and spatial & temporal attention. ∅ means such combinations do not (yet) exist.


- *M.H.Guo, T.X.Xu, Z.N.Liu, T.J.Mu, S.H.Zhang and S.M.Hu are with the BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.*
- *J.J.Liu, P.T.Jiang and M.M.Cheng are with TKLNDST, College of Computer Science, Nankai University, Tianjin 300350, China.*
- *R.R.Martin was with the School of Computer Science and Informatics, Cardiff University, UK.*
- *S.M.Hu is the corresponding author. E-mail: shimin@tsinghua.edu.cn.*

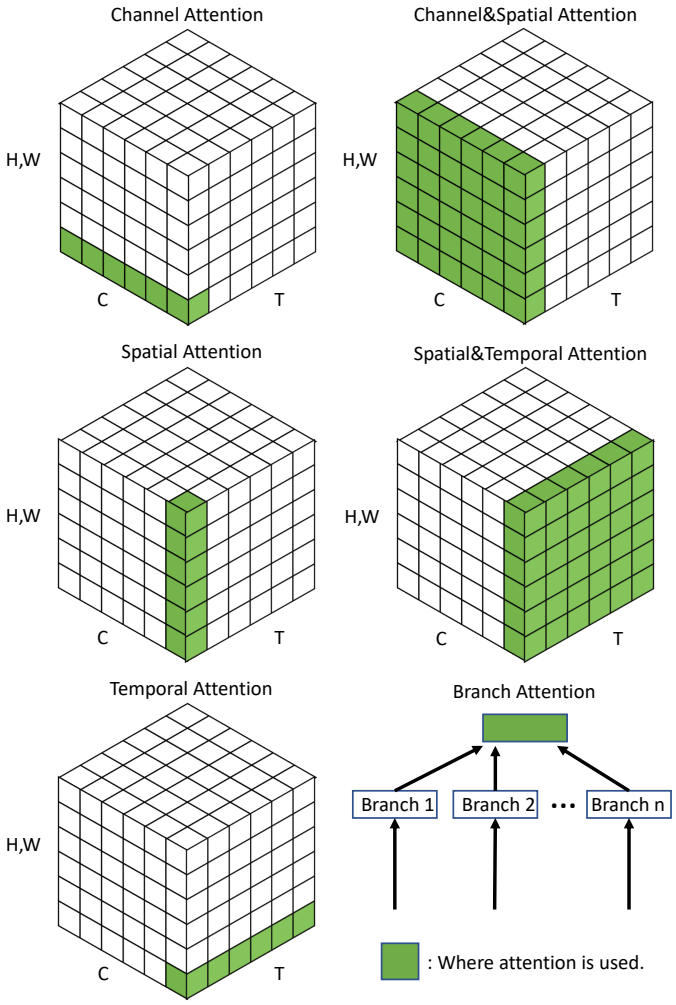



Fig. 2. Channel, spatial and temporal attention can be regarded as operating on different domains. $C$ represents the channel domain, $H$ and $W$ represent spatial domains, and $T$ means the temporal domain. Branch attention is complementary to these. Figure following [30].

that presented a novel channel-attention network which implicitly and adaptively predicts the potential key features. CBAM [6] and ECANet [37] are representative works of this phase. The last phase is the self-attention era. Self-attention was firstly proposed in [33] and rapidly provided great advances in the field of natural language processing [33], [38], [39]. Wang et al. [15] took the lead in introducing self-attention to computer vision and presented a novel non-local network with great success in video understanding and object detection. It was followed by a series of works such as EMANet [40], CCNet [41], HamNet [42] and the Stand-Alone Network [43], which improved speed, quality of results, and generalization capability. Recently, various pure deep self-attention networks (visual transformers) [27], [34], [44], [45], [46], [47], [48], [49] have appeared, showing the huge potential of attention-based models. It is clear that attention-based models have the potential to replace convolutional neural networks and become a more powerful and general architecture in computer vision.

The goal of this paper is to summarize and classify current attention methods in computer vision. Our approach

TABLE 1
Key notation in this paper. Other minor notation is explained where used.

| Symbol | Description |
|---|---|
| $X$ | input feature map, $X \in \mathbb{R}^{C\times H\times W}$ |
| $Y$ | output feature map |
| $W$ | learnable kernel weight |
| FC | fully-connected layer |
| Conv | convolution |
| GAP | global average pooling |
| GMP | global max pooling |
| $[\ ]$ | concatenation |
| $\delta$ | ReLU activation [51] |
| $\sigma$ | sigmoid activation |
| $\tanh$ | tanh activation |
| Softmax | softmax activation |
| BN | batch normalization [52] |
| Expand | expan input by repetition |

is shown in Fig. 1 and further explained in Fig. 2: it is based around data domain. Some methods consider the question of *when* the important data occurs, or others *where* it occurs, etc., and accordingly try to find key times or locations in the data. We divide existing attention methods into six categories which include four basic categories: channel attention (*what to pay attention to* [50]), spatial attention (*where to pay attention*), temporal attention (*when to pay attention*) and branch channel (*which to pay attention to*), along with two hybrid combined categories: channel & spatial attention and spatial & temporal attention. These ideas are further briefly summarized together with related works in Tab. 2.

The main contributions of this paper are:

- a systematic review of visual attention methods, covering the unified description of attention mechanisms, the development of visual attention mechanisms as well as current research,
- a categorisation grouping attention methods according to their data domain, allowing us to link visual attention methods independently of their particular application, and
- suggestions for future research in visual attention.

Sec. 2 considers related surveys, then Sec. 3 is the main body of our survey. Suggestions for future research are given in Sec. 4 and finally, we give conclusions in Sec. 5.

## 2 Other Surveys

In this section, we briefly compare this paper to various existing surveys which have reviewed attention methods and visual transformers. Chaudhari et al. [140] provide a survey of attention models in deep neural networks which concentrates on their application to natural language processing, while our work focuses on computer vision. Three more specific surveys [141], [142], [143] summarize the development of visual transformers while our paper reviews attention mechanisms in vision more generally, not just self-attention mechanisms. Wang et al. [144] present a survey of attention models in computer vision, but it only considers RNN-based attention models, which form just a part of our survey. In addition, unlike previous surveys, we provide a classification which groups various attention methods according to their data domain, rather than according to their field of application. Doing so allows us to concentrate

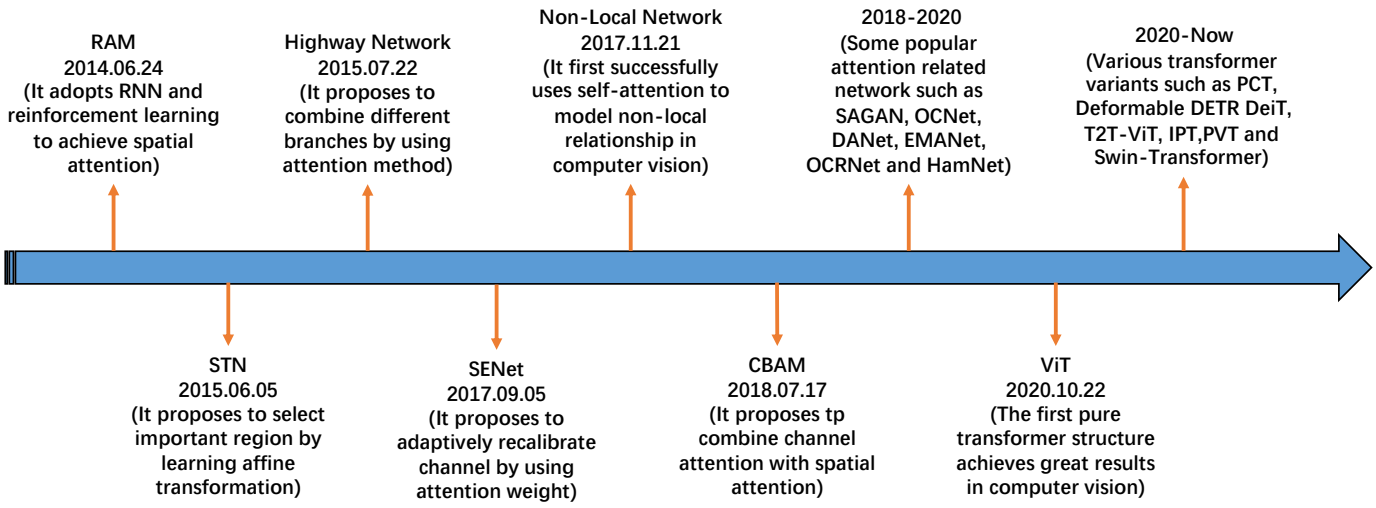



Fig. 3. Brief summary of key developments in attention in computer vision, which have loosely occurred in four phases. Phase 1 adopted RNNs to construct attention, a representative method being RAM [31]. Phase 2 explicitly predicted important regions, a representative method being STN [32]. Phase 3 implicitly completed the attention process, a representative method being SENet [5]. Phase 4 used self-attention methods [15], [33], [34].

TABLE 2
Brief summary of attention categories and key related works.

| Attention category | Description | Related work |
|---|---|---|
| Channel attention | Generate attention mask across the channel domain and use it to select important channels. | [5], [37], [53], [54], [55], [56], [57], [58] [25], [59], [60] |
| Spatial attention | Generate attention mask across spatial domains and use it to select important spatial regions (e.g. [15], [61]) or predict the most relevant spatial position directly (e.g. [7], [31]). | [8], [9], [15], [21], [31], [32], [34], [35] , [22], [26], [62], [63], [64], [65], [66], [67] , [41], [68], [69], [70], [71], [72], [73], [74] , [8], [34], [42], [43], [75], [76], [77], [78] , [27], [44], [45], [46], [79], [80], [81], [82] , [61], [83], [84], [85], [86], [87], [88], [89] , [47], [90], [91], [92], [93], [94], [95], [96] , [97], [98], [99], [100], [101], [102], [103], [104] , [20], [105], [106], [107], [108], [109] |
| Temporal attention | Generate attention mask in time and use it to select key frames. | [110], [111], [112] |
| Branch attention | Generate attention mask across the different branches and use it to select important branches. | [113], [114], [115], [116] |
| Channel & spatial attention | Predict channel and spatial attention masks separately (e.g. [6], [117]) or generate a joint 3-D channel, height, width attention mask directly (e.g. [118], [119]) and use it to select important features. | [6], [50], [117], [119], [120], [121], [122], [10], [101], [118], [123], [124], [125], [126] , [13], [14], [127], [128], [129] |
| Spatial & temporal attention | Compute temporal and spatial attention masks separately (e.g. [16], [130]), or produce a joint spatiotemporal attention mask (e.g. [131]), to focus on informative regions. | [130], [132], [133], [134], [135], [136], [137], [138], [139] |

on the attention methods in their own right, rather than treating them as supplementary to other tasks.

## 3 Attention Methods in Computer Vision

In this section, we first sum up a general form for the attention mechanism based on the recognition process of human visual system in Sec. 3.1. Then we review various categories of attention models given in Fig. 1, with a subsection dedicated to each category. In each, we tabularize representative works for that category. We also introduce that category of attention strategy more deeply, considering its development in terms of motivation, formulation and function.

### 3.1 General form

When seeing a scene in our daily life, we will focus on the discriminative regions, and process these regions quickly. The above process can be formulated as:

$$Attention = f(g(x), x) \tag{1}$$

Here $g(x)$ can represent to generate attention which corresponds to the process of attending to the discriminative regions. $f(g(x), x)$ means processing input $x$ based on the attention $g(x)$ which is consistent with processing critical regions and getting information.

With the above definition, we find that almost all existing attention mechanisms can be written into the above formulation. Here we take self-attention [15] and squeeze-and-

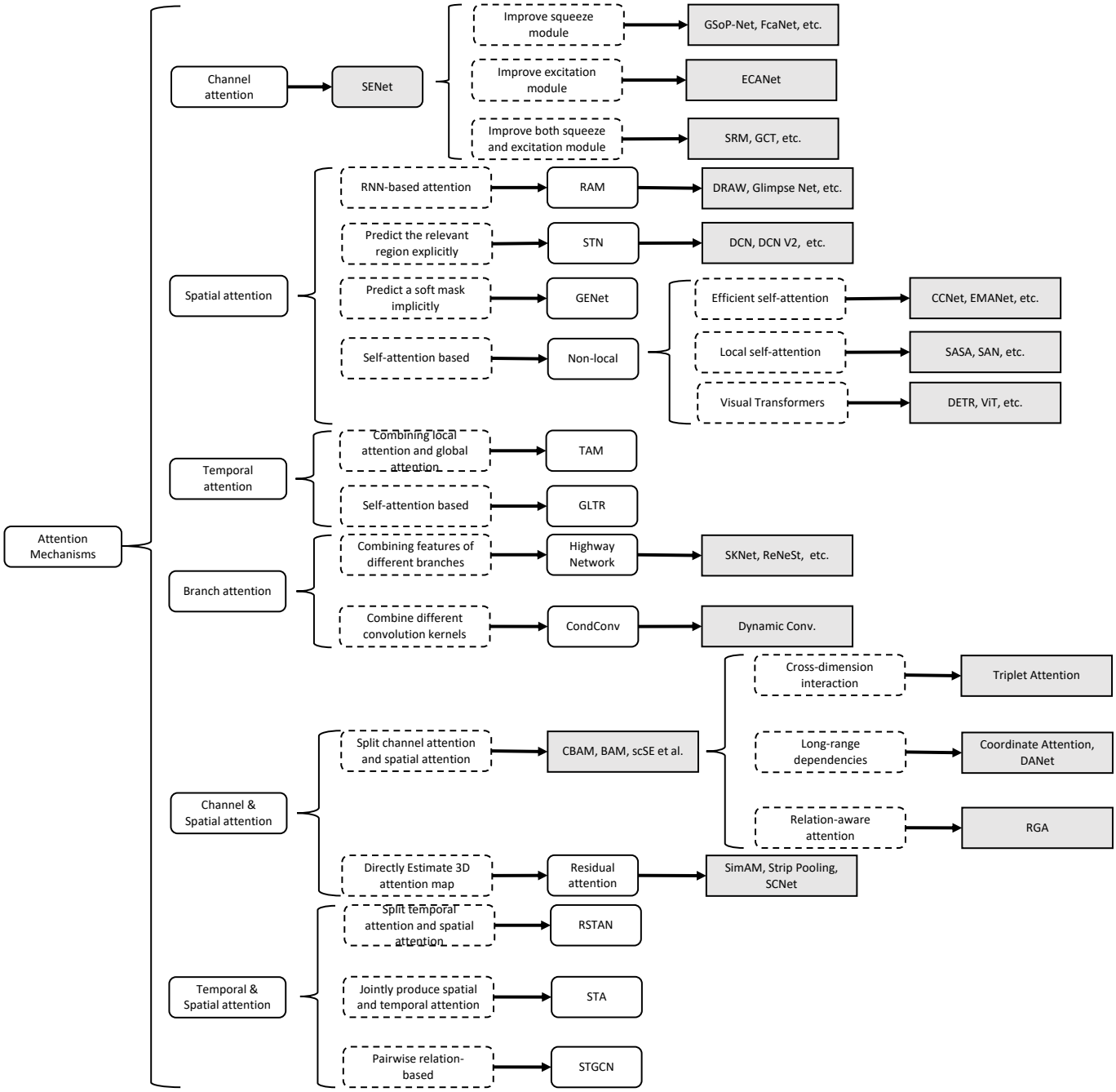



Fig. 4. Developmental context of visual attention.

excitation(SE) attention [5] as examples. For self-attention, $g(x)$ and $f(g(x), x)$ can be written as

$$Q, K, V = \text{Linear}(x) \tag{2}$$
$$g(x) = \text{Softmax}(QK) \tag{3}$$
$$f(g(x), x) = g(x)V \tag{4}$$
$$\tag{5}$$

For SE, $g(x)$ and $f(g(x), x)$ can be written as

$$g(x) = \text{Sigmoid}(\text{MLP}(\text{GAP}(x))) \tag{6}$$
$$f(g(x), x) = g(x)x \tag{7}$$

In the following, we will introduce various attention mechanisms and specify them to the above formulation.

### 3.2 Channel Attention

In deep neural networks, different channels in different feature maps usually represent different objects [50]. Channel attention adaptively recalibrates the weight of each channel, and can be viewed as an object selection process, thus determining *what to pay attention to*. Hu et al. [5] first proposed the concept of channel attention and presented SENet for this purpose. As Fig. 4 shows, and we discuss

shortly, three streams of work continue to improve channel attention in different ways.

In this section, we first summarize the representative channel attention works and specify process $g(x)$ and $f(g(x), x)$ described as Eq. 1 in Tab. 3 and Fig. 5. Then we discuss various channel attention methods along with their development process respectively.

#### 3.2.1 SENet

SENet [5] pioneered channel attention. The core of SENet is a *squeeze-and-excitation* (SE) block which is used to collect global information, capture channel-wise relationships and improve representation ability.

SE blocks are divided into two parts, a squeeze module and an excitation module. Global spatial information is collected in the squeeze module by global average pooling. The excitation module captures channel-wise relationships and outputs an attention vector by using fully-connected layers and non-linear layers (ReLU and sigmoid). Then, each channel of the input feature is scaled by multiplying the corresponding element in the attention vector. Overall, a squeeze-and-excitation block $F_{\text{se}}$ (with parameter $\theta$) which takes $X$ as input and outputs $Y$ can be formulated as:

$$s = F_{\text{se}}(X, \theta) = \sigma(W_2 \delta(W_1 \text{GAP}(X))) \tag{8}$$

$$Y = sX \tag{9}$$

SE blocks play the role of emphasizing important channels while suppressing noise. An SE block can be added after each residual unit [145] due to their low computational resource requirements. However, SE blocks have shortcomings. In the squeeze module, global average pooling is too simple to capture complex global information. In the excitation module, fully-connected layers increase the complexity of the model. As Fig. 4 indicates, later works attempt to improve the outputs of the squeeze module (e.g. GSoP-Net [54]), reduce the complexity of the model by improving the excitation module (e.g. ECANet [37]), or improve both the squeeze module and the excitation module (e.g. SRM [55]).

#### 3.2.2 GSoP-Net

An SE block captures global information by only using global average pooling (i.e. first-order statistics), which limits its modeling capability, in particular the ability to capture high-order statistics.

To address this issue, Gao et al. [54] proposed to improve the squeeze module by using a *global second-order pooling* (GSoP) block to model high-order statistics while gathering global information.

Like an SE block, a GSoP block also has a squeeze module and an excitation module. In the squeeze module, a GSoP block firstly reduces the number of channels from $c$ to $c'$ ($c' < c$) using a $1x1$ convolution, then computes a $c' \times c'$ covariance matrix for the different channels to obtain their correlation. Next, row-wise normalization is performed on the covariance matrix. Each $(i, j)$ in the normalized covariance matrix explicitly relates channel $i$ to channel $j$.

In the excitation module, a GSoP block performs row-wise convolution to maintain structural information and output a vector. Then a fully-connected layer and a sigmoid function are applied to get a $c$-dimensional attention vector. Finally, it multiplies the input features by the attention vector, as in an SE block. A GSoP block can be formulated as:

$$s = F_{\text{gsop}}(X, \theta) = \sigma(W \text{RC}(\text{Cov}(\text{Conv}(X)))) \tag{10}$$

$$Y = sX \tag{11}$$

Here, $\text{Conv}(\cdot)$ reduces the number of channels, $\text{Cov}(\cdot)$ computes the covariance matrix and $\text{RC}(\cdot)$ means row-wise convolution.

By using second-order pooling, GSoP blocks have improved the ability to collect global information over the SE block. However, this comes at the cost of additional computation. Thus, a single GSoP block is typically added after several residual blocks.

#### 3.2.3 SRM

Motivated by successes in style transfer, Lee et al. [55] proposed the lightweight *style-based recalibration module* (SRM). SRM combines style transfer with an attention mechanism. Its main contribution is style pooling which utilizes both mean and standard deviation of the input features to improve its capability to capture global information. It also adopts a lightweight *channel-wise fully-connected* (CFC) layer, in place of the original fully-connected layer, to reduce the computational requirements.

Given an input feature map $X \in \mathbb{R}^{C \times H \times W}$, SRM first collects global information by using style pooling ($\text{SP}(\cdot)$) which combines global average pooling and global standard deviation pooling. Then a channel-wise fully connected ($\text{CFC}(\cdot)$) layer (i.e. fully connected per channel), batch normalization BN and sigmoid function $\sigma$ are used to provide the attention vector. Finally, as in an SE block, the input features are multiplied by the attention vector. Overall, an SRM can be written as:

$$s = F_{\text{srm}}(X, \theta) = \sigma(\text{BN}(\text{CFC}(\text{SP}(X)))) \tag{12}$$

$$Y = sX \tag{13}$$

The SRM block improves both squeeze and excitation modules, yet can be added after each residual unit like an SE block.

#### 3.2.4 GCT

Due to the computational demand and number of parameters of the fully connected layer in the excitation module, it is impractical to use an SE block after each convolution layer. Furthermore, using fully connected layers to model channel relationships is an implicit procedure. To overcome the above problems, Yang et al. [56] propose the *gated channel transformation* (GCT) to efficiently collect information while explicitly modeling channel-wise relationships.

Unlike previous methods, GCT first collects global information by computing the $l_2$-norm of each channel. Next, a learnable vector $\alpha$ is applied to scale the feature. Then a competition mechanism is adopted by channel normalization to interact between channels. Like other common normalization methods, a learnable scale parameter $\gamma$ and bias $\beta$ are applied to rescale the normalization. However, unlike previous methods, GCT adopts tanh activation to control the attention vector. Finally, it not only multiplies the input

by the attention vector but also adds an identity connection. GCT can be written as:

$$s = F_{\text{gct}}(X, \theta) = \tanh(\gamma CN(\alpha \text{Norm}(X)) + \beta) \quad (14)$$

$$Y = sX + X, \quad (15)$$

where $\alpha$, $\beta$ and $\gamma$ are trainable parameters. Norm$(\cdot)$ indicates the $L2$-norm of each channel. $CN$ is channel normalization.

A GCT block has fewer parameters than an SE block, and as it is lightweight, can be added after each convolutional layer of a CNN.

#### 3.2.5 ECANet

To avoid high model complexity, SENet reduces the number of channels. However, this strategy fails to directly model correspondence between weight vectors and inputs, reducing the quality of results. To overcome this drawback, Wang et al. [37] proposed the *efficient channel attention* (ECA) block which instead uses a 1D convolution to determine the interaction between channels, instead of dimensionality reduction.

An ECA block has similar formulation to an SE block including a squeeze module for aggregating global spatial information and an efficient excitation module for modeling cross-channel interaction. Instead of indirect correspondence, an ECA block only considers direct interaction between each channel and its $k$-nearest neighbors to control model complexity. Overall, the formulation of an ECA block is:

$$s = F_{\text{eca}}(X, \theta) = \sigma(\text{Conv1D}(\text{GAP}(X))) \quad (16)$$

$$Y = sX \quad (17)$$

where Conv1D$(\cdot)$ denotes 1D convolution with a kernel of shape $k$ across the channel domain, to model local cross-channel interaction. The parameter $k$ decides the coverage of interaction, and in ECA the kernel size $k$ is adaptively determined from the channel dimensionality $C$ instead of by manual tuning, using cross-validation:

$$k = \psi(C) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{\text{odd}} \quad (18)$$

where $\gamma$ and $b$ are hyperparameters. $|x|_{\text{odd}}$ indicates the nearest odd function of $x$.

Compared to SENet, ECANet has an improved excitation module, and provides an efficient and effective block which can readily be incorporated into various CNNs.

#### 3.2.6 FcaNet

Only using global average pooling in the squeeze module limits representational ability. To obtain a more powerful representation ability, Qin et al. [57] rethought global information captured from the viewpoint of compression and analysed global average pooling in the frequency domain. They proved that global average pooling is a special case of the discrete cosine transform (DCT) and used this observation to propose a novel *multi-spectral channel attention*.

Given an input feature map $X \in \mathbb{R}^{C \times H \times W}$, multi-spectral channel attention first splits $X$ into many parts $x^i \in \mathbb{R}^{C' \times H \times W}$. Then it applies a 2D DCT to each part $x^i$. Note that a 2D DCT can use pre-processing results to reduce computation. After processing each part, all results are concatenated into a vector. Finally, fully connected layers, ReLU activation and a sigmoid are used to get the attention vector as in an SE block. This can be formulated as:

$$s = F_{\text{fca}}(X, \theta) = \sigma(W_2 \delta(W_1[(\text{DCT}(\text{Group}(X)))])) \quad (19)$$

$$Y = sX \quad (20)$$

where Group$(\cdot)$ indicates dividing the input into many groups and DCT$(\cdot)$ is the 2D discrete cosine transform.

This work based on information compression and discrete cosine transforms achieves excellent performance on the classification task.

#### 3.2.7 EncNet

Inspired by SENet, Zhang et al. [53] proposed the *context encoding module* (CEM) incorporating *semantic encoding loss* (SE-loss) to model the relationship between scene context and the probabilities of object categories, thus utilizing global scene contextual information for semantic segmentation.

Given an input feature map $X \in \mathbb{R}^{C \times H \times W}$, a CEM first learns $K$ cluster centers $D = \{d_1, \ldots, d_K\}$ and a set of smoothing factors $S = \{s_1, \ldots, s_K\}$ in the training phase. Next, it sums the difference between the local descriptors in the input and the corresponding cluster centers using soft-assignment weights to obtain a permutation-invariant descriptor. Then, it applies aggregation to the descriptors of the $K$ cluster centers instead of concatenation for computational efficiency. Formally, CEM can be written as:

$$e_k = \frac{\sum_{i=1}^{N} e^{-s_k ||X_i - d_k||^2} (X_i - d_k)}{\sum_{j=1}^{K} e^{-s_j ||X_i - d_j||^2}} \quad (21)$$

$$e = \sum_{k=1}^{K} \phi(e_k) \quad (22)$$

$$s = \sigma(We) \quad (23)$$

$$Y = sX \quad (24)$$

where $d_k \in \mathbb{R}^C$ and $s_k \in \mathbb{R}$ are learnable parameters. $\phi$ denotes batch normalization with ReLU activation. In addition to channel-wise scaling vectors, the compact contextual descriptor $e$ is also applied to compute the SE-loss to regularize training, which improves the segmentation of small objects.

Not only does CEM enhance class-dependent feature maps, but it also forces the network to consider big and small objects equally by incorporating SE-loss. Due to its lightweight architecture, CEM can be applied to various backbones with only low computational overhead.

#### 3.2.8 Bilinear Attention

Following GSoP-Net [54], Fang et al. [146] claimed that previous attention models only use first-order information and disregard higher-order statistical information. They thus proposed a new *bilinear attention block* (bi-attention) to capture local pairwise feature interactions within each channel, while preserving spatial information.

Bi-attention employs the *attention-in-attention* (AiA) mechanism to capture second-order statistical information: the outer point-wise channel attention vectors are computed from the output of the inner channel attention. Formally,

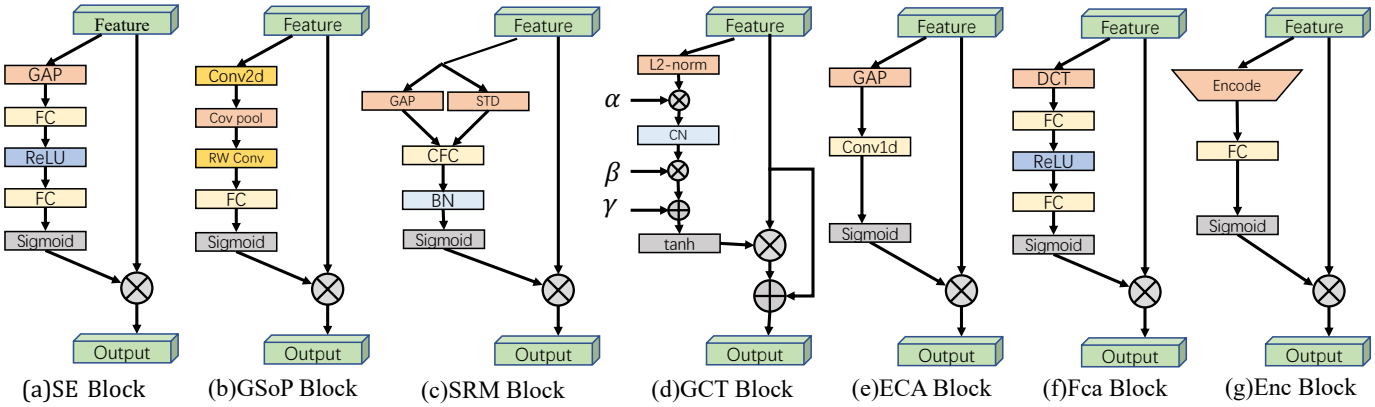



Fig. 5. Various channel attention mechanisms. GAP=global average pooling, GMP=global max pooling, FC=fully-connected layer, Cov pool=Covariance pooling, RW Conv=row-wise convolution, CFC=channel-wise fully connected, CN=channel normalization, DCT=discrete cosine transform.

TABLE 3
Representative channel attention mechanisms ordered by category and publication date. Their key aims are to emphasize important channels and capture global information. Application areas include: Cls = classification, Det = detection, SSeg = semantic segmentation, ISeg = instance segmentation, ST = style transfer, Action = action recognition.$g(x)$ and $f(g(x), x)$ are the attention process described by Eq. 1. Ranges means the ranges of attention map. S or H means soft or hard attention.(A) channel-wise product. (I) emphasize important channels, (II) capture global information.

| Category | Method | Publication | Tasks | $g(x)$ | $f(g(x), x)$ | Ranges | S or H | Goals |
|---|---|---|---|---|---|---|---|---|
| Squeeze-and-excitation network | SENet [5] | CVPR2018 | Cls, Det | global average pooling -> MLP -> sigmoid. | (A) | (0,1) | S | (I),(II) |
| Improve squeeze module | EncNet [53] | CVPR2018 | SSeg | encoder -> MLP -> sigmoid. | (A) | (0,1) | S | (I),(II) |
| | GSoP-Net [54] | CVPR2019 | Cls | 2nd-order pooling -> convolution & MLP -> sigmoid | (A) | (0,1) | S | (I),(II) |
| | FcaNet [57] | ICCV2021 | Cls, Det, ISeg | discrete cosine transform -> MLP -> sigmoid. | (A) | (0,1) | S | (I),(II) |
| Improve excitation module | ECANet [37] | CVPR2020 | Cls, Det, ISeg | global average pooling -> conv1d -> sigmoid. | (A) | (0,1) | S | (I),(II) |
| Improve both squeeze and excitation module | SRM [55] | arXiv2019 | Cls, ST | style pooling -> convolution & MLP -> sigmoid. | (A) | (0,1) | S | (I),(II) |
| | GCT [56] | CVPR2020 | Cls, Det, Action | compute $L2$-norm on spatial -> channel normalization -> tanh. | (A) | (-1,1) | S | (I),(II) |

given the input feature map $X$, bi-attention first uses bilinear pooling to capture second-order information

$$\widetilde{x} = \mathrm{Bi}(\phi(X)) = \mathrm{Vec}(\mathrm{UTri}(\phi(X)\phi(X)^T)) \tag{25}$$

where $\phi$ denotes an embedding function used for dimensionality reduction, $\phi(x)^T$ is the transpose of $\phi(x)$ across the channel domain, $\mathrm{Utri}(\cdot)$ extracts the upper triangular elements of a matrix and $\mathrm{Vec}(\cdot)$ is vectorization. Then bi-attention applies the inner channel attention mechanism to the feature map $\widetilde{x} \in \mathbb{R}^{\frac{c'(c'+1)}{2} \times H \times W}$

$$\widehat{x} = \omega(\mathrm{GAP}(\widetilde{x}))\varphi(\widetilde{x}) \tag{26}$$

Here $\omega$ and $\varphi$ are embedding functions. Finally the output feature map $\widehat{x}$ is used to compute the spatial channel attention weights of the outer point-wise attention mechanism:

$$s = \sigma(\widehat{x}) \tag{27}$$

$$Y = sX \tag{28}$$

The bi-attention block uses bilinear pooling to model the local pairwise feature interactions along each channel, while preserving the spatial information. Using the proposed AiA, the model pays more attention to higher-order statistical information compared with other attention-based models. Bi-attention can be incorporated into any CNN backbone to improve its representational power while suppressing noise.

### 3.3 Spatial Attention

Spatial attention can be seen as an adaptive spatial region selection mechanism: *where to pay attention*. As Fig. 4 shows, RAM [31], STN [32], GENet [61] and Non-Local [15] are representative of different kinds of spatial attention methods. RAM represents RNN-based methods. STN represents those use a sub-network to explicitly predict relevant regions. GENet represents those that use a sub-network implicitly to predict a soft mask to select important regions. Non-Local represents self-attention related methods. In this subsection,

we first summarize representative spatial attention mechanisms and specify process $g(x)$ and $f(g(x), x)$ described as Eq. 1 in Tab. 4, then discuss them according to Fig. 4.

#### 3.3.1 RAM

Convolutional neural networks have huge computational costs, especially for large inputs. In order to concentrate limited computing resources on important regions, Mnih et al. [31] proposed the *recurrent attention model* (RAM) that adopts RNNs [147] and reinforcement learning (RL) [148] to make the network learn where to pay attention. RAM pioneered the use of RNNs for visual attention, and was followed by many other RNN-based methods [21], [35], [88].

As shown in Fig. 6, the RAM has three key elements: (A) a glimpse sensor, (B) a glimpse network and (C) an RNN model. The glimpse sensor takes a coordinate $l_{t-1}$ and an image $X_t$. It outputs multiple resolution patches $\rho(X_t, l_{t-1})$ centered on $l_{t-1}$. The glimpse network $f_g(\theta(g))$ includes a glimpse sensor and outputs the feature representation $g_t$ for input coordinate $l_{t-1}$ and image $X_t$. The RNN model considers $g_t$ and an internal state $h_{t-1}$ and outputs the next center coordinate $l_t$ and the action $a_t$, e.g. the softmax result in an image classification task. Since the whole process is not differentiable, it applies reinforcement learning strategies in the update process.

This provides a simple but effective method to focus the network on key regions, thus reducing the number of calculations performed by the network, especially for large inputs, while improving image classification results.

#### 3.3.2 Glimpse Network

Inspired by how humans perform visual recognition sequentially, Ba et al. [88] proposed a deep recurrent network, similar to RAM [31], capable of processing a multi-resolution crop of the input image, called a glimpse, for multiple object recognition task. The proposed network updates its hidden state using a glimpse as input, and then predicts a new object as well as the next glimpse location at each step. The glimpse is usually much smaller than the whole image, which makes the network computationally efficient.

The proposed deep recurrent visual attention model consists of a context network, glimpse network, recurrent network, emission network, and classification network. First, the context network takes the down-sampled whole image as input to provide the initial state for the recurrent network as well as the location of the first glimpse. Then, at the current time step $t$, given the current glimpse $x_t$ and its location tuple $l_t$, the goal of the glimpse network is to extract useful information, expressed as

$$g_t = f_{\text{image}}(X) \cdot f_{\text{loc}}(l_t) \tag{29}$$

where $f_{\text{image}}(X)$ and $f_{\text{loc}}(l_t)$ are non-linear functions which both output vectors having the same dimension, and $\cdot$ denotes element-wise product, used for fusing information from two branches. Then, the recurrent network, which consists of two stacked recurrent layers, aggregates information gathered from each individual glimpse. The outputs of the recurrent layers are:

$$r_t^{(1)} = f_{\text{rec}}^{(1)}(g_t, r_{t-1}^{(1)}) \tag{30}$$

$$r_t^{(2)} = f_{\text{rec}}^{(2)}(r_t^{(1)}, r_{t-1}^{(2)}) \tag{31}$$

Given the current hidden state $r_t^{(2)}$ of the recurrent network, the emission network predicts where to crop the next glimpse. Formally, it can be written as

$$l_{t+1} = f_{\text{emis}}(r_t^{(2)}) \tag{32}$$

Finally, the classification network outputs a prediction for the class label $y$ based on the hidden state $r_t^{(1)}$ of the recurrent network

$$y = f_{\text{cls}}(r_t^{(1)}) \tag{33}$$

Compared to a CNN operating on the entire image, the computational cost of the proposed model is much lower, and it can naturally tackle images of different sizes because it only processes a glimpse in each step. Robustness is additionally improved by the recurrent attention mechanism, which also alleviates the problem of over-fitting. This pipeline can be incorporated into any state-of-the-art CNN backbones or RNN units.

#### 3.3.3 Hard and soft attention

To visualize where and what an image caption generation model should focus on, Xu et al. [35] introduced an attention-based model as well as two variant attention mechanisms, *hard attention* and *soft attention*.

Given a set of feature vectors $\boldsymbol{a} = \{a_1, \ldots, a_L\}, a_i \in \mathbb{R}^D$ extracted from the input image, the model aims to produce a caption by generating one word at each time step. Thus they adopt a long short-term memory (LSTM) network as a decoder; an attention mechanism is used to generate a contextual vector $z_t$ conditioned on the feature set $\boldsymbol{a}$ and the previous hidden state $h_{t-1}$, where $t$ denotes the time step. Formally, the weight $\alpha_{t,i}$ of the feature vector $a_i$ at the $t$-th time step is defined as

$$e_{t,i} = f_{\text{att}}(a_i, h_{t-1}) \tag{34}$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{L} \exp(e_{t,k})} \tag{35}$$

where $f_{\text{att}}$ is implemented by a multilayer perceptron conditioned on the previous hidden state $h_{t-1}$. The positive weight $\alpha_{t,i}$ can be interpreted either as the probability that location $i$ is the right place to focus on (hard attention), or as the relative importance of location $i$ to the next word (soft attention). To obtain the contextual vector $z_t$, the hard attention mechanism assigns a multinoulli distribution parametrized by $\{\alpha_{t,i}\}$ and views $z_t$ as a random variable:

$$p(s_{t,i} = 1 | \boldsymbol{a}, h_{t-1}) = \alpha_{t,i} \tag{36}$$

$$z_t = \sum_{i=1}^{L} s_{t,i} a_i \tag{37}$$

On the other hand, the soft attention mechanism directly uses the expectation of the context vector $z_t$,

$$z_t = \sum_{i=1}^{L} \alpha_{t,i} a_i \tag{38}$$

The use of the attention mechanism improves the interpretability of the image caption generation process by allowing the user to understand what and where the model is focusing on. It also helps to improve the representational capability of the network.

TABLE 4
Representative spatial attention mechanisms sorted by category and date. Application areas include: Cls = classification, FGCls = fine-grained classification, Det = detection, SSeg = semantic segmentation, ISeg = instance segmentation, ST = style transfer, Action = action recognition, ICap = image captioning.$g(x)$ and $f(g(x), x)$ are the attention process described by Eq. 1. Ranges means the ranges of attention map. S or H means soft or hard attention.(A) choose region according to the prediction. (B) element-wise product, (C)aggregate information via attention map. (I)focus the network on discriminative regions, (II) avoid excessive computation for large input images, (III) provide more transformation invariance, (IV) capture long-range dependencies, (V) denoise input feature map (VI) adaptively aggregate neighborhood information, (VII) reduce inductive bias.

| Category | Method | Publication | Tasks | $g(x)$ | $f(g(x), x)$ | Ranges | S or H | Goals |
|---|---|---|---|---|---|---|---|---|
| RNN-based methods | RAM [31] | NIPS2014 | Cls | use RNN to recurrently predict important regions | (A) | (0,1) | H | (I), (II). |
| | Hard and soft attention [35] | ICML2015 | ICap | a)compute similarity between visual features and previous hidden state -> interpret attention weight. | (C) | (0,1) | S, H | (I). |
| Predict the relevant region explictly | STN [32] | NIPS2015 | Cls, FG-Cls | use sub-network to predict an affine transformation. | (A) | (0,1) | H | (I), (III). |
| | DCN [7] | ICCV2017 | Det, SSeg | use sub-network to predict offset coordinates. | (A) | (0,1) | H | (I), (III). |
| Predict the relevant region implictly | GENet [61] | NIPS2018 | Cls, Det | average pooling or depth-wise convolution -> interpolation -> sigmoid | (B) | (0,1) | S | (I). |
| | PSANet [87] | ECCV2018 | SSeg | predict an attention map using a sub-network. | (C) | (0,1) | S | (I), (IV). |
| Self-attention based methods | Non-Local [15] | CVPR2018 | Action, Det, ISeg | Dot product between query and key -> softmax | (C) | (0,1) | S | (I), (IV), (V) |
| | SASA [43] | NeurIPS2019 | Cls, Det | Dot product between query and key -> softmax. | (C) | (0,1) | S | (I), (VI) |
| | ViT [34] | ICLR2021 | Cls | divide the feature map into multiple groups -> Dot product between query and key -> softmax. | (C) | (0,1) | S | (I),(IV), (VII). |

#### 3.3.4 Attention Gate

Previous approaches to MR segmentation usually operate on particular regions of interest (ROI), which requires excessive and wasteful use of computational resources and model parameters. To address this issue, Oktay et al. [19] proposed a simple and yet effective mechanism, the *attention gate* (AG), to focus on targeted regions while suppressing feature activations in irrelevant regions.

Given the input feature map $X$ and the gating signal $G \in \mathbb{R}^{C' \times H \times W}$ which is collected at a coarse scale and contains contextual information, the attention gate uses additive attention to obtain the gating coefficient. Both the input $X$ and the gating signal are first linearly mapped to an $\mathbb{R}^{F \times H \times W}$ dimensional space, and then the output is squeezed in the channel domain to produce a spatial attention weight map $S \in \mathbb{R}^{1 \times H \times W}$. The overall process can be written as

$$S = \sigma(\varphi(\delta(\phi_x(X) + \phi_g(G)))) \tag{39}$$

$$Y = SX \tag{40}$$

where $\varphi$, $\phi_x$ and $\phi_g$ are linear transformations implemented as $1 \times 1$ convolutions.

The attention gate guides the model's attention to important regions while suppressing feature activation in unrelated areas. It substantially enhances the representational power of the model without a significant increase in computing cost or number of model parameters due to its lightweight design. It is general and modular, making it simple to use in various CNN models.

#### 3.3.5 STN

The property of translation equivariance makes CNNs suitable for processing image data. However, CNNs lack other transformation invariance such as rotational invariance, scaling invariance and warping invariance. To achieve these attributes while making CNNs focus on important regions, Jaderberg et al. [32] proposed *spatial transformer networks* (STN) that use an explicit procedure to learn invariance to translation, scaling, rotation and other more general warps, making the network pay attention to the most relevant regions. STN was the first attention mechanism to explicitly predict important regions and provide a deep neural network with transformation invariance. Various following works [7], [36] have had even greater success.

Taking a 2D image as an example, a 2D affine transformation can be formulated as:

$$\begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} = f_{\text{loc}}(U) \tag{41}$$

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}. \tag{42}$$

Here, $U$ is the input feature map, and $f_{\text{loc}}$ can be any differentiable function, such as a lightweight fully-connected network or convolutional neural network. $x_i^s$ and $y_i^s$ are

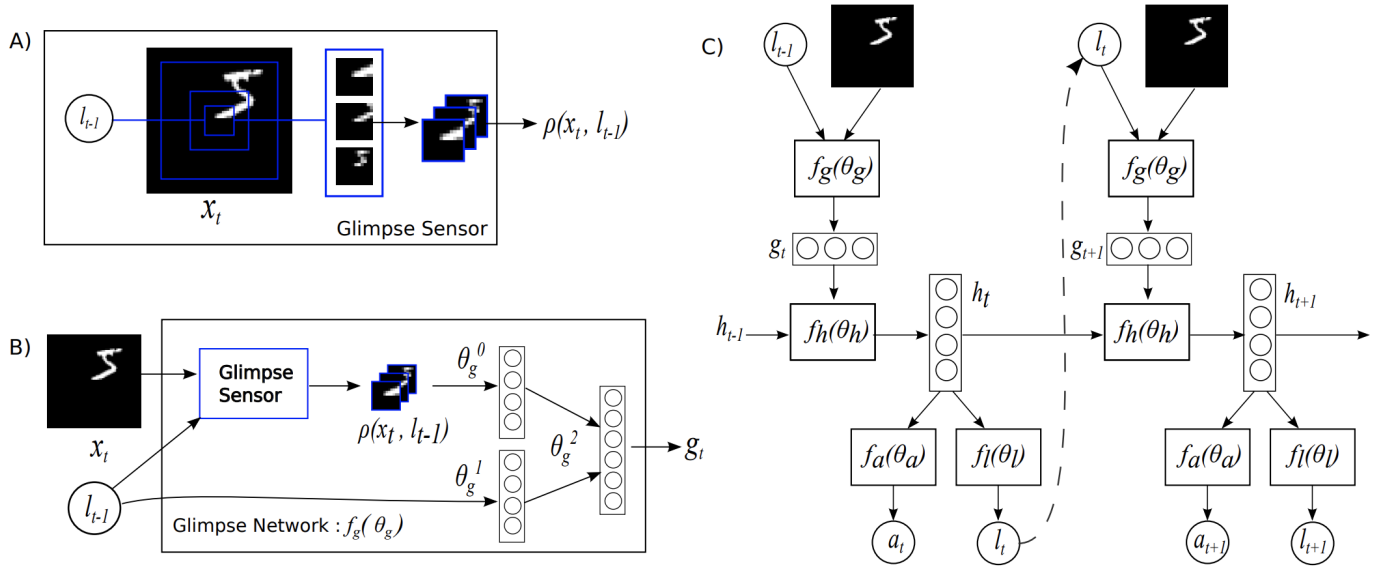



Fig. 6. Attention process in RAM [31]. (A): a glimpse sensor takes image and center coordinates as input and outputs multiple resolution patches. (B): a glimpse network includes a glimpse sensor, taking image and center coordinates as input and outputting a feature vector. (C) the entire network recurrently uses a glimpse network, outputting the predicted result as well as the next center coordinates. Figure is taken from [31].

coordinates in the output feature map, while $x_i^t$ and $y_i^t$ are corresponding coordinates in the input feature map and the $\theta$ matrix is the learnable affine matrix. After obtaining the correspondence, the network can sample relevant input regions using the correspondence. To ensure that the whole process is differentiable and can be updated in an end-to-end manner, bilinear sampling is used to sample the input features

STNs focus on discriminative regions automatically and learn invariance to some geometric transformations.

#### 3.3.6 Deformable Convolutional Networks

With similar purpose to STNs, Dai et al. [7] proposed *deformable convolutional networks* (deformable ConvNets) to be invariant to geometric transformations, but they pay attention to the important regions in a different manner.

Specifically, deformable ConvNets do not learn an affine transformation. They divide convolution into two steps, firstly sampling features on a regular grid $\mathcal{R}$ from the input feature map, then aggregating sampled features by weighted summation using a convolution kernel. The process can be written as:

$$Y(p_0) = \sum_{p_i \in \mathcal{R}} w(p_i) X(p_0 + p_i) \tag{43}$$

$$\mathcal{R} = \{(-1,-1),(-1,0),\ldots,(1,1)\} \tag{44}$$

The deformable convolution augments the sampling process by introducing a group of learnable offsets $\Delta p_i$ which can be generated by a lightweight CNN. Using the offsets $\Delta p_i$, the deformable convolution can be formulated as:

$$Y(p_0) = \sum_{p_i \in \mathcal{R}} w(p_i) X(p_0 + p_i + \Delta p_i). \tag{45}$$

Through the above method, adaptive sampling is achieved. However, $\Delta p_i$ is a floating point value unsuited to grid sampling. To address this problem, bilinear interpolation is used. Deformable RoI pooling is also used, which greatly improves object detection.

Deformable ConvNets adaptively select the important regions and enlarge the valid receptive field of convolutional neural networks; this is important in object detection and semantic segmentation tasks.

#### 3.3.7 Self-attention and variants

Self-attention was proposed and has had great success in the field of *natural language processing* (NLP) [33], [38], [39], [149], [150], [151], [152]. Recently, it has also shown the potential to become a dominant tool in computer vision [8], [15], [34], [78], [153]. Typically, self-attention is used as a spatial attention mechanism to capture global information. We now summarize the self-attention mechanism and its common variants in computer vision.

Due to the localisation of the convolutional operation, CNNs have inherently narrow receptive fields [154], [155], which limits the ability of CNNs to understand scenes globally. To increase the receptive field, Wang et al. [15] introduced self-attention into computer vision.

Taking a 2D image as an example, given a feature map $F \in \mathbb{R}^{C \times H \times W}$, self-attention first computes the queries, keys and values $Q, K, V \in \mathbb{R}^{C' \times N}, N = H \times W$ by linear projection and reshaping operations. Then self-attention can be formulated as:

$$A = (a)_{i,j} = \text{Softmax}(QK^T), \tag{46}$$

$$Y = AV, \tag{47}$$

where $A \in \mathbb{R}^{N \times N}$ is the attention matrix and $\alpha_{i,j}$ is the relationship between the $i$-th and $j$-th elements. The whole process is shown in Fig. 7(left). Self-attention is a powerful tool to model global information and is useful in many visual tasks [9], [22], [26], [62], [63], [64], [65], [66], [67].

However, the self-attention mechanism has several shortcomings, particularly its quadratic complexity, which limit its

applicability. Several variants have been introduced to alleviate these problems. The *disentangled non-local* approach [74] improves self-attention's accuracy and effectiveness, but most variants focus on reducing its computational complexity.

CCNet [41] regards the self-attention operation as a graph convolution and replaces the densely-connected graph processed by self-attention with several sparsely-connected graphs. To do so, it proposes *criss-cross attention* which considers row attention and column attention recurrently to obtain global information. CCNet reduces the complexity of self-attention from $O(N^2)$ to $O(N\sqrt{N})$.

EMANet [40] views self-attention in terms of expectation maximization (EM). It proposes *EM attention* which adopts the EM algorithm to get a set of compact bases instead of using all points as reconstruction bases. This reduces the complexity from $O(N^2)$ to $O(NK)$, where $K$ is the number of compact bases.

ANN [68] suggests that using all positional features as key and vectors is redundant and adopts spatial pyramid pooling [156], [157] to obtain a few representative key and value features to use instead, to reduce computation.

GCNet [69] analyses the attention map used in self-attention and finds that the global contexts obtained by self-attention are similar for different query positions in the same image. Thus, it first proposes to predict a single attention map shared by all query points, and then gets global information from a weighted sum of input features according to this attention map. This is like average pooling, but is a more general process for collecting global information.

$A^2$Net [70] is motivated by SENet to divide attention into feature gathering and feature distribution processes, using two different kinds of attention. The first aggregates global information via second-order attention pooling and the second distributes the global descriptors by soft selection attention.

GloRe [71] understands self-attention from a graph learning perspective. It first collects $N$ input features into $M \ll N$ nodes and then learns an adjacency matrix of global interactions between nodes. Finally, the nodes distribute global information to input features. A similar idea can be found in LatentGNN [72], MLP-Mixer [158] and ResMLP [159].

OCRNet [73] proposes the concept of *object-contextual representation* which is a weighted aggregation of all object regions' representations in the same category, such as a weighted average of all car region representations. It replaces the key and vector with this object-contextual representation leading to successful improvements in both speed and effectiveness.

The *disentangled non-local* approach was motivated by [15], [69]. Yin et al [74] deeply analyzed the self-attention mechanism resulting in the core idea of decoupling self-attention into a pairwise term and a unary term. The pairwise term focuses on modeling relationships while the unary term focuses on salient boundaries. This decomposition prevents unwanted interactions between the two terms, greatly improving semantic segmentation, object detection and action recognition.

HamNet [42] models capturing global relationships as a low-rank completion problem and designs a series of white-box methods to capture global context using matrix decomposition. This not only reduces the complexity, but increases the interpretability of self-attention.

EANet [75] proposes that self-attention should only consider correlation in a single sample and should ignore potential relationships between different samples. To explore the correlation between different samples and reduce computation, it makes use of an external attention that adopts learnable, lightweight and shared key and value vectors. It further reveals that using softmax to normalize the attention map is not optimal and presents double normalization as a better alternative.

In addition to being a complementary approach to CNNs, self-attention also can be used to replace convolution operations for aggregating neighborhood information. Convolution operations can be formulated as dot products between the input feature $X$ and a convolution kernel $W$:

$$Y^c_{i,j} = \sum_{a,b\in\{0,\dots,k-1\}} W_{a,b,c} X_{\hat{a},\hat{b}} \tag{48}$$

where

$$\hat{a} = i + a - \lfloor k/2 \rfloor, \qquad \hat{b} = j + b - \lfloor k/2 \rfloor, \tag{49}$$

$k$ is the kernel size and $c$ indicates the channel. The above formulation can be viewed as a process of aggregating neighborhood information by using a weighted sum through a convolution kernel. The process of aggregating neighborhood information can be defined more generally as:

$$Y_{i,j} = \sum_{a,b\in\{0,\dots,k-1\}} \text{Rel}(i,j,\hat{a},\hat{b}) f(X_{\hat{a},\hat{b}}) \tag{50}$$

where $\text{Rel}(i,j,\hat{a},\hat{b})$ is the relation between position (i,j) and position $(\hat{a},\hat{b})$. With this definition, local self-attention is a special case.

For example, SASA [43] writes this as

$$Y_{i,j} = \sum_{a,b\in\mathcal{N}_k(i,j)} \text{Softmax}_{ab}(q_{ij}^T k_{ab} + q_{ij} r_{a-i,b-j}) v_{ab} \tag{51}$$

where $q$, $k$ and $v$ are linear projections of input feature $x$, and $r_{a-i,b-j}$ is the relative positional embedding of $(i,j)$ and $(a,b)$.

We now consider several specific works using local self-attention as basic neural network blocks

SASA [43] suggests that using self-attention to collect global information is too computationally intensive and instead adopts local self-attention to replace all spatial convolution in a CNN. The authors show that doing so improves speed, number of parameters and quality of results. They also explores the behavior of positional embedding and show that relative positional embeddings [160] are suitable. Their work also studies how to combinie local self-attention with convolution.

LR-Net [76] appeared concurrently with SASA. It also studies how to model local relationships by using local self-attention. A comprehensive study probed the effects of positional embedding, kernel size, appearance composability and adversarial attacks.

SAN [77] explored two modes, pairwise and patchwise, of utilizing attention for local feature aggregation. It proposed a novel vector attention adaptive both in content and

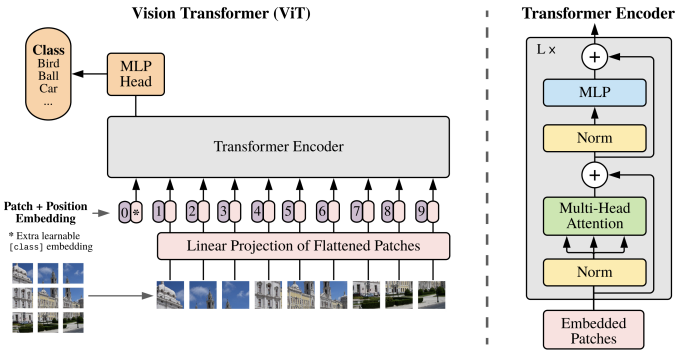



Fig. 7. Vision transformer [34]. Left: architecture. Vision transformer first splits the image into different patches and projects them into feature space where a transformer encoder processes them to produce the final result. Right: basic vision transformer block with multi-head attention core. Figure is taken from [34].

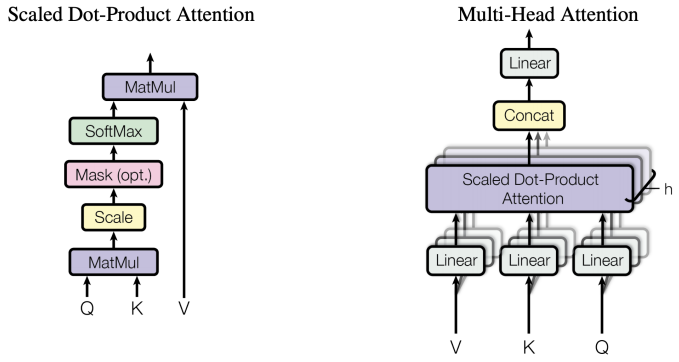



Fig. 8. Left: Self-attention. Right: Multi-head self-attention. Figure from [33].

channel, and assessed its effectiveness both theoretically and practically. In addition to providing significant improvements in the image domain, it also has been proven useful in 3D point cloud processing [80].

### 3.3.8 Vision Transformers

Transformers have had great success in natural language processing [33], [38], [149], [150], [152], [161]. Recently, iGPT [78] and DETR [8] demonstrated the huge potential for transformer-based models in computer vision. Motivated by this, Dosovitskiy et al [34] proposed the *vision transformer* (ViT) which is the first pure transformer architecture for image processing. It is capable of achieving comparable results to modern convolutional neural networks.

As Fig 7 shows, the main part of ViT is the multi-head attention (MHA) module. MHA takes a sequence as input. It first concatenates a class token with the input feature $F \in \mathcal{R}^{N\times C}$, where $N$ is the number of pixels. Then it gets $Q, K \in \mathcal{R}^{N\times C'}$ and $V \in \mathcal{R}^{N\times C}$ by linear projection. Next, $Q$, $K$ and $V$ are divided into $H$ heads in the channel domain and self-attention separately applied to them. The MHA approach is shown in Fig. 8. ViT stacks a number of MHA layers with fully connected layers, layer normalization [162] and the GELU [163] activation function.

ViT demonstrates that a pure attention-based network can achieve better results than a convolutional neural network especially for large datasets such as JFT-300 [164] and ImageNet-21K [165].

Following ViT, many transformer-based architectures such as PCT [27], IPT [79], T2T-ViT [44], DeepViT [166], SETR [81], PVT [45], CaiT [167], TNT [82], Swin-transformer [46], Query2Label [83], MoCoV3 [84], BEiT [85], SegFormer [86], FuseFormer [168] and MAE [169] have appeared, with excellent results for many kind of visual tasks including image classification, object detection, semantic segmentation, point cloud processing, action recognition and self-supervised learning.

A detailed survey of vision transformers is omitted here as other recent surveys [141], [142], [143], [170] comprehensively review the use of transformer methods for visual tasks.

### 3.3.9 GENet

Inspired by SENet, Hu et al. [61] designed GENet to capture long-range spatial contextual information by providing a recalibration function in the spatial domain.

GENet combines part gathering and excitation operations. In the first step, it aggregates input features over large neighborhoods and models the relationship between different spatial locations. In the second step, it first generates an attention map of the same size as the input feature map, using interpolation. Then each position in the input feature map is scaled by multiplying by the corresponding element in the attention map. This process can be described by:

$$g = f_{\text{gather}}(X), \tag{52}$$

$$s = f_{\text{excite}}(g) = \sigma(\text{Interp}(g)), \tag{53}$$

$$Y = sX. \tag{54}$$

Here, $f_{\text{gather}}$ can take any form which captures spatial correlations, such as global average pooling or a sequence of depth-wise convolutions; $\text{Interp}(\cdot)$ denotes interpolation.

The gather-excite module is lightweight and can be inserted into each residual unit like an SE block. It emphasizes important features while suppressing noise.

### 3.3.10 PSANet

Motivated by success in capturing long-range dependencies in convolutional neural networks, Zhao et al. [87] presented the novel PSANet framework to aggregate global information. It models information aggregation as an information flow and proposes a bidirectional information propagation mechanism to make information flow globally.

PSANet formulates information aggregation as:

$$z_i = \sum_{j\in\Omega(i)} F(x_i, x_j, \Delta_{ij})x_j \tag{55}$$

where $\Delta_{ij}$ indicates the positional relationship between $i$ and $j$. $F(x_i, x_j, \Delta_{ij})$ is a function that takes $x_i$, $x_j$ and $\Delta_{ij}$ into consideration to controls information flow from $j$ to $i$. $\Omega_i$ represents the aggregation neighborhood of position $i$; if we wish to capture global information, $\Omega_i$ should include all spatial positions.

Due to the complexity of calculating function $F(x_i, x_j, \Delta_{ij})$, it is decomposed into an approximation:

$$F(x_i, x_j, \Delta_{ij}) \approx F_{\Delta_{ij}}(x_i) + F_{\Delta_{ij}}(x_j) \tag{56}$$

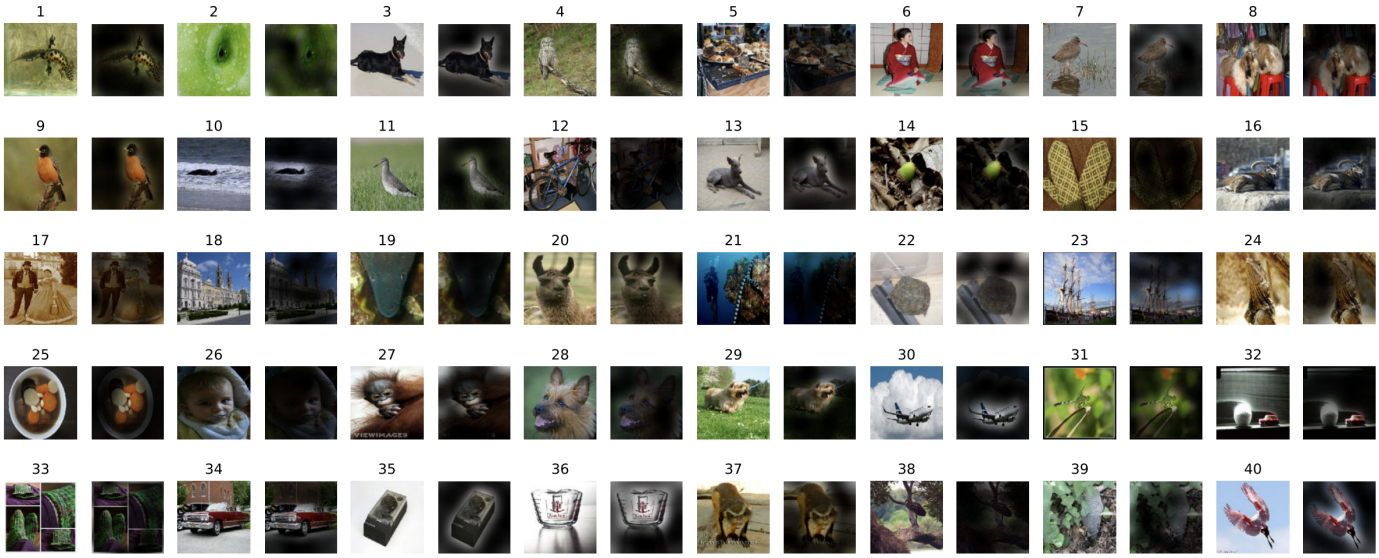


Fig. 9. Attention map results from [34]. The network focuses on the discriminative regions of each image. Figure from [34].

whereupon Eq. 55 can be simplified to:

$$z_i = \sum_{j \in \Omega(i)} F_{\Delta_{ij}}(x_i) x_j + \sum_{j \in \Omega(i)} F_{\Delta_{ij}}(x_j) x_j. \tag{57}$$

The first term can be viewed as collecting information at position $i$ while the second term distributes information at position $j$. Functions $F_{\Delta_{ij}}(x_i)$ and $F_{\Delta_{ij}}(x_j)$ can be seen as adaptive attention weights.

The above process aggregates global information while emphasizing the relevant features. It can be added to the end of a convolutional neural network as an effective complement to greatly improve semantic segmentation.

### 3.4 Temporal Attention

Temporal attention can be seen as a dynamic time selection mechanism determining *when to pay attention*, and is thus usually used for video processing. Previous works [171], [172] often emphasise how to capture both short-term and long-term cross-frame feature dependencies. Here, we first summarize representative temporal attention mechanisms and specify process $g(x)$ and $f(g(x), x)$ described as Eq. 1 in Tab. 5, and then discuss various such mechanisms according to the order in Fig. 4.

#### 3.4.1 Self-attention and variants

RNN and temporal pooling or weight learning have been widely used in work on video representation learning to capture interaction between frames, but these methods have limitations in terms of either efficiency or temporal relation modeling.

To overcome them, Li et al. [171] proposed a *global-local temporal representation* (GLTR) to exploit multi-scale temporal cues in a video sequence. GLTR consists of a *dilated temporal pyramid* (DTP) for local temporal context learning and a *temporal self attention* module for capturing global temporal interaction. DTP adopts dilated convolution with dilatation rates increasing progressively to cover various temporal ranges, and then concatenates the various outputs to aggregate multi-scale information. Given input frame-wise features $F = \{f_1, \dots f_T\}$, DTP can be written as:

$$\{f_1^{(r)}, \dots, f_T^{(r)}\} = \text{DConv}^{(r)}(F) \tag{58}$$

$$f'_t = [f_t^{(1)}; \dots f_t^{(2^{n-1})} \dots; f_t^{(2^{N-1})}] \tag{59}$$

where $\text{DConv}^{(r)}(\cdot)$ denotes dilated convolution with dilation rate $r$. The self-attention mechanism adopts convolution layers followed by batch normalization and ReLU activation to generate the query $Q \in \mathbb{R}^{d \times T}$, the key $K \in \mathbb{R}^{d \times T}$ and the value $V \in \mathbb{R}^{d \times T}$ based on the input feature map $F' = \{f'_1, \dots f'_T\}$, which can be written as

$$F_{\text{out}} = g(V \text{Softmax}(Q^T K)) + F' \tag{60}$$

where $g$ denotes a linear mapping implemented by a convolution.

The short-term temporal contextual information from neighboring frames helps to distinguish visually similar regions while the long-term temporal information serves to overcome occlusions and noise. GLTR combines the advantages of both modules, enhancing representation capability and suppressing noise. It can be incorporated into any state-of-the-art CNN backbone to learn a global descriptor for a whole video. However, the self-attention mechanism has quadratic time complexity, limiting its application.

#### 3.4.2 TAM

To capture complex temporal relationships both efficiently and flexibly, Liu et al. [172] proposed a *temporal adaptive module* (TAM). It adopts an adaptive kernel instead of self-attention to capture global contextual information, with lower time complexity than GLTR [171].

TAM has two branches, a local branch and a global branch. Given the input feature map $X \in \mathbb{R}^{C \times T \times H \times W}$, global spatial average pooling GAP is first applied to the feature map to ensure TAM has a low computational cost. Then the local branch in TAM employs several 1D convolutions with ReLU

TABLE 5
Representative temporal attention mechanisms sorted by date. ReID = re-identification, Action = action recognition. Ranges means the ranges of attention map. S or H means soft or hard attention. $g(x)$ and $f(g(x), x)$ are the attention process described by Eq. 1. (A) aggregate information via attention map. (I) exploit multi-scale short-term temporal contextual information (II)capture long-term temporal feature dependencies (III) capture local temporal contexts

| Category | Method | Publication | Tasks | $g(x)$ | $f(g(x), x)$ | Ranges | S or H | Goals |
|---|---|---|---|---|---|---|---|---|
| Self-attention based methods | GLTR [171] | ICCV2019 | ReID | dilated 1D Convs -> self-attention in temporal dimension | (A) | (0,1) | S | (I), (II). |
| Combine local attention and global attention | TAM [172] | Arxiv2020 | Action | a)local: global spatial average pooling -> 1D Convs, b) global: global spatial average pooling -> MLP -> adaptive convolution | (A) | (0,1) | S | (II), (III). |

nonlinearity across the temporal domain to produce location-sensitive importance maps for enhancing frame-wise features. The local branch can be written as

$$s = \sigma(\text{Conv1D}(\delta(\text{Conv1D}(\text{GAP}(X))))) \tag{61}$$

$$X^1 = sX. \tag{62}$$

Unlike the local branch, the global branch is location invariant and focuses on generating a channel-wise adaptive kernel based on global temporal information in each channel. For the $c$-th channel, the kernel can be written as

$$\Theta_c = \text{Softmax}(\text{FC}_2(\delta(\text{FC}_1(\text{GAP}(X)_c)))) \tag{63}$$

where $\Theta_c \in \mathbb{R}^K$ and $K$ is the adaptive kernel size. Finally, TAM convolves the adaptive kernel $\Theta$ with $X^1_{\text{out}}$:

$$Y = \Theta \otimes X^1 \tag{64}$$

With the help of the local branch and global branch, TAM can capture the complex temporal structures in video and enhance per-frame features at low computational cost. Due to its flexibility and lightweight design, TAM can be added to any existing 2D CNNs.

### 3.5 Branch Attention

Branch attention can be seen as a dynamic branch selection mechanism: *which to pay attention to*, used with a multi-branch structure. We first summarize representative branch attention mechanisms and specify process $g(x)$ and $f(g(x), x)$ described as Eq. 1 in Tab. 6, then discuss various ones in detail.

#### 3.5.1 Highway networks

Inspired by the *long short term memory* network, Srivastava et al. [113] proposed *highway networks* that employ adaptive gating mechanisms to enable information flows across layers to address the problem of training very deep networks.

Supposing a plain neural network consists of $L$ layers, and $H_l(X)$ denotes a non-linear transformation on the $l$-th layer, a highway network can be expressed as

$$Y_l = H_l(X_l)T_l(X_l) + X_l(1 - T_l(X_l)) \tag{65}$$

$$T_l(X) = \sigma(W_l^T X + b_l) \tag{66}$$

where $T_l(X)$ denotes the transform gate regulating the information flow for the $l$-th layer. $X_l$ and $Y_l$ are the inputs and outputs of the $l$-th layer.

The gating mechanism and skip-connection structure make it possible to directly train very deep highway networks using simple gradient descent methods. Unlike fixed skip-connections, the gating mechanism adapts to the input, which helps to route information across layers. A highway network can be incorporated in any CNN.

#### 3.5.2 SKNet

Research in the neuroscience community suggests that visual cortical neurons adaptively adjust the sizes of their receptive fields (RFs) according to the input stimulus [174]. This inspired Li et al. [114] to propose an automatic selection operation called *selective kernel* (SK) convolution.

SK convolution is implemented using three operations: split, fuse and select. During split, transformations with different kernel sizes are applied to the feature map to obtain different sized RFs. Information from all branches is then fused together via element-wise summation to compute the gate vector. This is used to control information flows from the multiple branches. Finally, the output feature map is obtained by aggregating feature maps for all branches, guided by the gate vector. This can be expressed as:

$$U_k = F_k(X) \qquad k = 1, \ldots, K \tag{67}$$

$$U = \sum_{k=1}^{K} U_k \tag{68}$$

$$z = \delta(\text{BN}(W\text{GAP}(U))) \tag{69}$$

$$s_k^{(c)} = \frac{e^{W_k^{(c)} z}}{\sum_{k=1}^{K} e^{W_k^{(c)} z}} \quad k = 1, \ldots, K, \quad c = 1, \ldots, C \tag{70}$$

$$Y = \sum_{k=1}^{K} s_k U_k \tag{71}$$

Here, each transformation $F_k$ has a unique kernel size to provide different scales of information for each branch. For efficiency, $F_k$ is implemented by grouped or depthwise convolutions followed by dilated convolution, batch normalization and ReLU activation in sequence. $t^{(c)}$ denotes the $c$-th element of vector $t$, or the $c$-th row of matrix $t$.

SK convolutions enable the network to adaptively adjust neurons' RF sizes according to the input, giving a notable improvement in results at little computational cost. The gate mechanism in SK convolutions is used to fuse information from multiple branches. Due to its lightweight design, SK

TABLE 6
Representative branch attention mechanisms sorted by date. Cls = classification, Det=Object Detection. $g(x)$ and $f(g(x), x)$ are the attention process described by Eq. 1.Ranges means the ranges of attention map. S or H means soft or hard attention. (A) element-wise product. (B) channel-wise product. (C) aggergate information via attention. (I)overcome the problem of vanishing gradient (II) dynamically fuse different branches. (III) adaptively select a suitable receptive field (IV) improve the performance of standard convolution (be) dynamically fuse different convolution kernels.

| Category | Method | Publication | Tasks | $g(x)$ | $f(g(x), x)$ | Ranges | S or H | Goals |
|---|---|---|---|---|---|---|---|---|
| Combine different branches | Highway Network [113] | ICML2015W | Cls | linear layer -> sigmoid | (A) | (0,1) | S | (I), (II). |
| | SKNet [114] | CVPR2019 | Cls | global average pooling -> MLP -> softmax | (B) | (0,1) | S | (II), (III) |
| Combine different convolution kernels | CondConv [173] | NeurIPS2019 | Cls, Det | global average pooling -> linear layer -> sigmoid | (C) | (0,1) | S | (IV), (V). |

convolution can be applied to any CNN backbone by replacing all large kernel convolutions. ResNeSt [115] also adopts this attention mechanism to improve the CNN backbone in a more general way, giving excellent results on ResNet [145] and ResNeXt [175].

#### 3.5.3 CondConv

A basic assumption in CNNs is that all convolution kernels are the same. Given this, the typical way to enhance the representational power of a network is to increase its depth or width, which introduces significant extra computational cost. In order to more efficiently increase the capacity of convolutional neural networks, Yang et al. [173] proposed a novel multi-branch operator called CondConv.

An ordinary convolution can be written

$$Y = W * X \tag{72}$$

where $*$ denotes convolution. The learnable parameter $W$ is the same for all samples. CondConv adaptively combines multiple convolution kernels and can be written as:

$$Y = (\alpha_1 W_1 + \cdots + \alpha_n W_n) * X \tag{73}$$

Here, $\alpha$ is a learnable weight vector computed by

$$\alpha = \sigma(W_r(\text{GAP}(X))) \tag{74}$$

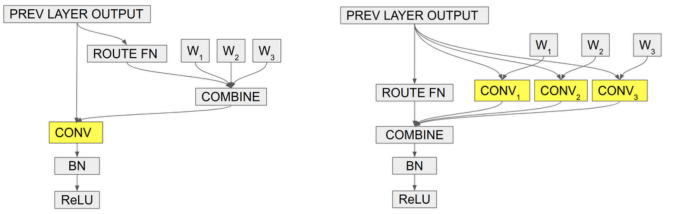
This process is equivalent to an ensemble of multiple experts, as shown in Fig. 10.

CondConv makes full use of the advantages of the multi-branch structure using a branch attention method with little computing cost. It presents a novel manner to efficiently increase the capability of networks.

#### 3.5.4 Dynamic Convolution

The extremely low computational cost of lightweight CNNs constrains the depth and width of the networks, further decreasing their representational power. To address the above problem, Chen et al. [116] proposed *dynamic convolution*, a novel operator design that increases representational power with negligible additional computational cost and does not change the width or depth of the network in parallel with CondConv [173].

Dynamic convolution uses $K$ parallel convolution kernels of the same size and input/output dimensions instead of one kernel per layer. Like SE blocks, it adopts a squeeze-and-excitation mechanism to generate the attention weights for the different convolution kernels. These kernels are




(a) CondConv: $(\alpha_1 W_1 + \ldots + \alpha_n W_n) * x$ (b) Mixture of Experts: $\alpha_1(W_1 * x) + \ldots + \alpha_n(W_n * x)$

Fig. 10. CondConv [173]. (a) CondConv first combines different convolution kernels and then uses the combined kernel for convolution. (b) Mixture of experts first uses multiple convolution kernels for convolution and then merges the results. While (a) and (b) are equivalent, (a) has much lower computational cost. Figure is taken from [173].

then aggregated dynamically by weighted summation and applied to the input feature map $X$:

$$s = \text{softmax}(W_2 \delta(W_1 \text{GAP}(X))) \tag{75}$$

$$\text{DyConv} = \sum_{i=1}^{K} s_k \text{Conv}_k \tag{76}$$

$$Y = \text{DyConv}(X) \tag{77}$$

Here the convolutions are combined by summation of weights and biases of convolutional kernels.

Compared to applying convolution to the feature map, the computational cost of squeeze-and-excitation and weighted summation is extremely low. Dynamic convolution thus provides an efficient operation to improve representational power and can be easily used as a replacement for any convolution.

### 3.6 Channel & Spatial Attention

Channel & spatial attention combines the advantages of channel attention and spatial attention. It adaptively selects both important objects and regions [50]. The *residual attention network* [119] pioneered the field of channel & spatial attention, emphasizing the importance of informative features in both spatial and channel dimensions. It adopts a bottom-up structure consisting of several convolutions to produce a 3D (height, width, channel) attention map. However, it has high computational cost and limited receptive fields.

To leverage global spatial information later works [6], [117] enhance discrimination of features by introducing global average pooling, as well as decoupling channel attention and spatial channel attention for computational

efficiency. Other works [10], [101] apply self-attention mechanisms for channel & spatial attention to explore pairwise interaction. Yet further works [120], [124] adopt the spatial-channel attention mechanism to enlarge the receptive field.

Representative channel & spatial attention mechanisms and specific process $g(x)$ and $f(g(x), x)$ described as Eq. 1 are in given Tab. 7; we next discuss various ones in detail.

#### 3.6.1 Residual Attention Network

Inspired by the success of ResNet [145], Wang et al. [119] proposed the very deep convolutional *residual attention network* (RAN) by combining an attention mechanism with residual connections.

Each attention module stacked in a residual attention network can be divided into a mask branch and a trunk branch. The trunk branch processes features, and can be implemented by any state-of-the-art structure including a pre-activation residual unit and an inception block. The mask branch uses a bottom-up top-down structure to learn a mask of the same size that softly weights output features from the trunk branch. A sigmoid layer normalizes the output to $[0, 1]$ after two $1 \times 1$ convolution layers. Overall the residual attention mechanism can be written as

$$s = \sigma(\text{Conv}_2^{1\times1}(\text{Conv}_1^{1\times1}(h_{\text{up}}(h_{\text{down}}(X))))) \tag{78}$$
$$X_{out} = sf(X) + f(X) \tag{79}$$

where $h_{\text{up}}$ is a bottom-up structure, using max-pooling several times after residual units to increase the receptive field, while $h_{\text{down}}$ is the top-down part using linear interpolation to keep the output size the same as the input feature map. There are also skip-connections between the two parts, which are omitted from the formulation. $f$ represents the trunk branch which can be any state-of-the-art structure.

Inside each attention module, a bottom-up top-down feedforward structure models both spatial and cross-channel dependencies, leading to a consistent performance improvement. Residual attention can be incorporated into any deep network structure in an end-to-end training fashion. However, the proposed bottom-up top-down structure fails to leverage global spatial information. Furthermore, directly predicting a 3D attention map has high computational cost.

#### 3.6.2 CBAM

To enhance informative channels as well as important regions, Woo et al. [6] proposed the *convolutional block attention module* (CBAM) which stacks channel attention and spatial attention in series. It decouples the channel attention map and spatial attention map for computational efficiency, and leverages spatial global information by introducing global pooling.

CBAM has two sequential sub-modules, channel and spatial. Given an input feature map $X \in \mathbb{R}^{C\times H\times W}$ it sequentially infers a 1D channel attention vector $s_c \in \mathbb{R}^C$ and a 2D spatial attention map $s_s \in \mathbb{R}^{H\times W}$. The formulation of the channel attention sub-module is similar to that of an SE block, except that it adopts more than one type of pooling operation to aggregate global information. In detail, it has two parallel branches using max-pool and avg-pool operations:

$$F_{\text{avg}}^c = \text{GAP}^s(X) \tag{80}$$
$$F_{\text{max}}^c = \text{GMP}^s(X) \tag{81}$$
$$s_c = \sigma(W_2\delta(W_1F_{\text{avg}}^c) + W_2\delta(W_1F_{\text{max}}^c)) \tag{82}$$
$$M_c(X) = s_cX \tag{83}$$

where $\text{GAP}^s$ and $\text{GMP}^s$ denote global average pooling and global max pooling operations in the spatial domain. The spatial attention sub-module models the spatial relationships of features, and is complementary to channel attention. Unlike channel attention, it applies a convolution layer with a large kernel to generate the attention map

$$F_{\text{avg}}^s = \text{GAP}^c(X) \tag{84}$$
$$F_{\text{max}}^s = \text{GMP}^c(X) \tag{85}$$
$$s_s = \sigma(\text{Conv}([F_{\text{avg}}^s; F_{\text{max}}^s])) \tag{86}$$
$$M_s(X) = s_sX \tag{87}$$

where $\text{Conv}(\cdot)$ represents a convolution operation, while $\text{GAP}^c$ and $\text{GMP}^c$ are global pooling operations in the channel domain. [] denotes concatenation over channels. The overall attention process can be summarized as

$$X' = M_c(X) \tag{88}$$
$$Y = M_s(X') \tag{89}$$

Combining channel attention and spatial attention sequentially, CBAM can utilize both spatial and cross-channel relationships of features to tell the network *what* to focus on and *where* to focus. To be more specific, it emphasizes useful channels as well as enhancing informative local regions. Due to its lightweight design, CBAM can be integrated into any CNN architecture seamlessly with negligible additional cost. Nevertheless, there is still room for improvement in the channel & spatial attention mechanism. For instance, CBAM adopts a convolution to produce the spatial attention map, so the spatial sub-module may suffer from a limited receptive field.

#### 3.6.3 BAM

At the same time as CBAM, Park et al. [117] proposed the *bottleneck attention module* (BAM), aiming to efficiently improve the representational capability of networks. It uses dilated convolution to enlarge the receptive field of the spatial attention sub-module, and build a *bottleneck structure* as suggested by ResNet to save computational cost.

For a given input feature map $X$, BAM infers the channel attention $s_c \in \mathbb{R}^C$ and spatial attention $s_s \in \mathbb{R}^{H\times W}$ in two parallel streams, then sums the two attention maps after resizing both branch outputs to $\mathbb{R}^{C\times H\times W}$. The channel attention branch, like an SE block, applies global average pooling to the feature map to aggregate global information, and then uses an MLP with channel dimensionality reduction. In order to utilize contextual information effectively, the

TABLE 7
Representative channel & spatial attention mechanisms sorted by date. Cls = classification, ICap = image captioning, Det = detection, Seg = segmentation, ISeg = instance segmentation, KP = keypoint detection, ReID = re-identification.$g(x)$ and $f(g(x), x)$ are the attention process described by Eq. 1. Ranges means the ranges of attention map. S or H means soft or hard attention. (A) element-wise product. (B) aggregate information via attention map.(I) focus the network on the discriminative region, (II) emphasize important channels, (III) capture long-range information, (IV) capture cross-domain interaction between any two domains.

| Category | Method | Publication | Tasks | $g(x)$ | $f(g(x), x)$ | Ranges | S or H | Goals |
|---|---|---|---|---|---|---|---|---|
| Jointly predict channel & spatial attention map | Residual Attention [119] | CVPR2017 | Cls | top-down network -> bottom down network -> 1×1 Convs -> Sigmoid | (A) | (0,1) | S | (I), (II) |
| | SCNet [120] | CVPR2020 | Cls, Det, ISeg, KP | top-down network -> bottom down network -> identity add -> sigmoid | (A) | (0,1) | S | (II), (III) |
| | Strip Pooling [124] | CVPR2020 | Seg | a)horizontal/vertical global pooling -> 1D Conv -> point-wise summation -> 1 × 1 Conv -> Sigmoid | (A) | (0,1) | S | (I), (II), (III) |
| Separately predict channel & spatial attention maps | SCA-CNN [50] | CVPR2017 | ICap | a)spatial: fuse hidden state -> 1 × 1 Conv -> Softmax, b)channel: global average pooling -> MLP -> Softmax | (A) | (0,1) | S | (I), (II), (III) |
| | CBAM [6] | ECCV2018 | Cls, Det | a)spatial: global pooling in channel dimension-> Conv -> Sigmoid, b)channel: global pooling in spatial dimension -> MLP -> Sigmoid | (A) | (0,1) | S | (I), (II), (III) |
| | BAM [6] | BMVC2018 | Cls, Det | a)spatial: dilated Convs, b)channel: global average pooling -> MLP, c)fuse two branches | (A) | (0,1) | S | (I), (II), (III) |
| | scSE [123] | TMI2018 | Seg | a)spatial: 1 × 1 Conv -> Sigmoid, b)channel: global average pooling -> MLP -> Sigmoid, c)fuse two branches | (A) | (0,1) | S | (I), (II), (III) |
| | Dual Attention [10] | CVPR2019 | Seg | a)spatial: self-attention in spatial dimension, b)channel: self-attention in channel dimension, c) fuse two branches | (B) | (0,1) | S | (I), (II), (III) |
| | RGA [101] | CVPR2020 | ReID | use self-attention to capture pairwise relations -> compute attention maps with the input and relation vectors | (A) | (0,1) | S | (I), (II), (III) |
| | Triplet Attention [121] | WACV2021 | Cls, Det | compute attention maps for pairs of domains -> fuse different branches | (A) | (0,1) | S | (I), (IV) |

spatial attention branch combines a bottleneck structure and dilated convolutions. Overall, BAM can be written as

$$s_c = \text{BN}(W_2(W_1\text{GAP}(X) + b_1) + b_2) \quad (90)$$

$$s_s = \text{BN}(\text{Conv}_2^{1\times1}(\text{DC}_2^{3\times3}(\text{DC}_1^{3\times3}(\text{Conv}_1^{1\times1}(X))))) \quad (91)$$

$$s = \sigma(\text{Expand}(s_s) + \text{Expand}(s_c)) \quad (92)$$

$$Y = sX + X \quad (93)$$

where $W_i$, $b_i$ denote weights and biases of fully connected layers respectively, $\text{Conv}_1^{1\times1}$ and $\text{Conv}_2^{1\times1}$ are convolution layers used for channel reduction. $\text{DC}_i^{3\times3}$ denotes a dilated convolution with $3 \times 3$ kernel, applied to utilize contextual information effectively. Expand expands the attention maps $s_s$ and $s_c$ to $\mathbb{R}^{C\times H\times W}$.

BAM can emphasize or suppress features in both spatial and channel dimensions, as well as improving the representational power. Dimensional reduction applied to both channel and spatial attention branches enables it to be integrated with any convolutional neural network with little extra computational cost. However, although dilated convolutions enlarge the receptive field effectively, it still fails to capture long-range contextual information as well as encoding cross-domain relationships.

#### 3.6.4 scSE

To aggregate global spatial information, an SE block applies global pooling to the feature map. However, it ignores pixel-wise spatial information, which is important in dense prediction tasks. Therefore, Roy et al. [123] proposed *spatial and channel SE blocks* (scSE). Like BAM, spatial SE blocks are used, complementing SE blocks, to provide spatial attention weights to focus on important regions.

Given the input feature map $X$, two parallel modules, spatial SE and channel SE, are applied to feature maps to encode spatial and channel information respectively. The channel SE module is an ordinary SE block, while the spatial SE module adopts $1 \times 1$ convolution for spatial squeezing. The outputs from the two modules are fused. The overall process can be written as

$$s_c = \sigma(W_2\delta(W_1\text{GAP}(X))) \tag{94}$$
$$X_{\text{chn}} = s_c X \tag{95}$$
$$s_s = \sigma(\text{Conv}^{1\times1}(X)) \tag{96}$$
$$X_{\text{spa}} = s_s X \tag{97}$$
$$Y = f(X_{\text{spa}}, X_{\text{chn}}) \tag{98}$$

where $f$ denotes the fusion function, which can be maximum, addition, multiplication or concatenation.

The proposed scSE block combines channel and spatial attention to enhance features as well as capturing pixel-wise spatial information. Segmentation tasks are greatly benefited as a result. The integration of an scSE block in F-CNNs makes a consistent improvement in semantic segmentation at negligible extra cost.

#### 3.6.5 Triplet Attention

In CBAM and BAM, channel attention and spatial attention are computed independently, ignoring relationships between these two domains [121]. Motivated by spatial attention, Misra et al. [121] proposed *triplet attention*, a lightweight but effective attention mechanism to capture cross-domain interaction.

Given an input feature map $X$, triplet attention uses three branches, each of which plays a role in capturing cross-domain interaction between any two domains from $H$, $W$ and $C$. In each branch, rotation operations along different axes are applied to the input first, and then a Z-pool layer is responsible for aggregating information in the zeroth dimension. Finally, a standard convolution layer with kernel size $k \times k$ models the relationship between the last two domains. This process can be written as

$$X_1 = \text{Pm}_1(X) \tag{99}$$
$$X_2 = \text{Pm}_2(X) \tag{100}$$
$$s_0 = \sigma(\text{Conv}_0(\text{Z-Pool}(X))) \tag{101}$$
$$s_1 = \sigma(\text{Conv}_1(\text{Z-Pool}(X_1))) \tag{102}$$
$$s_2 = \sigma(\text{Conv}_2(\text{Z-Pool}(X_2))) \tag{103}$$
$$Y = \frac{1}{3}(s_0X + \text{Pm}_1^{-1}(s_1X_1) + \text{Pm}_2^{-1}(s_2X_2)) \tag{104}$$

where $\text{Pm}_1$ and $\text{Pm}_2$ denote rotation through 90° anti-clockwise about the $H$ and $W$ axes respectively, while $\text{Pm}_i^{-1}$ denotes the inverse. Z-Pool concatenates max-pooling and average pooling along the zeroth dimension.

$$Y = \text{Z-Pool}(X) = [\text{GMP}(X); \text{GAP}(X)] \tag{105}$$

Unlike CBAM and BAM, triplet attention stresses the importance of capturing cross-domain interactions instead of computing spatial attention and channel attention independently. This helps to capture rich discriminative feature representations. Due to its simple but efficient structure, triplet attention can be easily added to classical backbone networks.

#### 3.6.6 SimAM

Yang et al. [118] also stress the importance of learning attention weights that vary across both channel and spatial domains in proposing SimAM, a simple, parameter-free attention module capable of directly estimating 3D weights instead of expanding 1D or 2D weights. The design of SimAM is based on well-known neuroscience theory, thus avoiding need for manual fine tuning of the network structure.

Motivated by the spatial suppression phenomenon [176], they propose that a neuron which shows suppression effects should be emphasized and define an energy function for each neuron as:

$$e_t(w_t, b_t, y, x_i) = (y_t - \hat{t})^2 + \frac{1}{M-1}\sum_{i=1}^{M-1}(y_o - \hat{x}_i) \tag{106}$$

where $\hat{t} = w_t t + b_t$, $\hat{x}_i = w_t x_i + b_t$, and $t$ and $x_i$ are the target unit and all other units in the same channel; $i \in 1, \ldots, N$, and $N = H \times W$.

An optimal closed-form solution for Eq. 106 exists:

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda} \tag{107}$$

where $\hat{\mu}$ is the mean of the input feature and $\hat{\sigma}^2$ is its variance. A sigmoid function is used to control the output range of the attention vector; an element-product is applied to get the final output:

$$Y = \text{Sigmoid}\left(\frac{1}{E}\right)X \tag{108}$$

This work simplifies the process of designing attention and successfully proposes a novel 3-D weight parameter-free attention module based on mathematics and neuroscience theories.

#### 3.6.7 Coordinate attention

An SE block aggregates global spatial information using global pooling before modeling cross-channel relationships, but neglects the importance of positional information. BAM and CBAM adopt convolutions to capture local relations, but fail to model long-range dependencies. To solve these problems, Hou et al. [129] proposed *coordinate attention*, a novel attention mechanism which embeds positional information into channel attention, so that the network can focus on large important regions at little computational cost.

The coordinate attention mechanism has two consecutive steps, coordinate information embedding and coordinate attention generation. First, two spatial extents of pooling kernels encode each channel horizontally and vertically. In the second step, a shared $1 \times 1$ convolutional transformation function is applied to the concatenated outputs of the two pooling layers. Then coordinate attention splits the resulting tensor into two separate tensors to yield attention vectors

with the same number of channels for horizontal and vertical coordinates of the input $X$ along. This can be written as

$$z^h = \text{GAP}^h(X) \tag{109}$$
$$z^w = \text{GAP}^w(X) \tag{110}$$
$$f = \delta(\text{BN}(\text{Conv}_1^{1\times1}([z^h; z^w]))) \tag{111}$$
$$f^h, f^w = \text{Split}(f) \tag{112}$$
$$s^h = \sigma(\text{Conv}_h^{1\times1}(f^h)) \tag{113}$$
$$s^w = \sigma(\text{Conv}_w^{1\times1}(f^w)) \tag{114}$$
$$Y = Xs^hs^w \tag{115}$$

where $\text{GAP}^h$ and $\text{GAP}^w$ denote pooling functions for vertical and horizontal coordinates, and $s^h \in \mathbb{R}^{C\times1\times W}$ and $s^w \in \mathbb{R}^{C\times H\times1}$ represent corresponding attention weights.

Using coordinate attention, the network can accurately obtain the position of a targeted object. This approach has a larger receptive field than BAM and CBAM. Like an SE block, it also models cross-channel relationships, effectively enhancing the expressive power of the learned features. Due to its lightweight design and flexibility, it can be easily used in classical building blocks of mobile networks.

#### 3.6.8 DANet

In the field of scene segmentation, encoder-decoder structures cannot make use of the global relationships between objects, whereas RNN-based structures heavily rely on the output of the long-term memorization. To address the above problems, Fu et al. [10] proposed a novel framework, the *dual attention network* (DANet), for natural scene image segmentation. Unlike CBAM and BAM, it adopts a self-attention mechanism instead of simply stacking convolutions to compute the spatial attention map, which enables the network to capture global information directly.

DANet uses in parallel a position attention module and a channel attention module to capture feature dependencies in spatial and channel domains. Given the input feature map $X$, convolution layers are applied first in the position attention module to obtain new feature maps. Then the position attention module selectively aggregates the features at each position using a weighted sum of features at all positions, where the weights are determined by feature similarity between corresponding pairs of positions. The channel attention module has a similar form except for dimensional reduction to model cross-channel relations. Finally the outputs from the two branches are fused to obtain final feature representations. For simplicity, we reshape the feature map $X$ to $C \times (H \times W)$ whereupon the overall process can be written as

$$Q, \quad K, \quad V = W_qX, \quad W_kX, \quad W_vX \tag{116}$$
$$Y^{\text{pos}} = X + V\text{Softmax}(Q^TK) \tag{117}$$
$$Y^{\text{chn}} = X + \text{Softmax}(XX^T)X \tag{118}$$
$$Y = Y^{\text{pos}} + Y^{\text{chn}} \tag{119}$$

where $W_q, W_k, W_v \in \mathbb{R}^{C\times C}$ are used to generate new feature maps.

The position attention module enables DANet to capture long-range contextual information and adaptively integrate similar features at any scale from a global viewpoint, while the channel attention module is responsible for enhancing useful channels as well as suppressing noise. Taking spatial and channel relationships into consideration explicitly improves the feature representation for scene segmentation. However, it is computationally costly, especially for large input feature maps.

#### 3.6.9 RGA

Unlike coordinate attention and DANet, which emphasise capturing long-range context, in *relation-aware global attention* (RGA) [101], Zhang et al. stress the importance of global structural information provided by pairwise relations, and uses it to produce attention maps.

RGA comes in two forms, *spatial RGA* (RGA-S) and *channel RGA* (RGA-C). RGA-S first reshapes the input feature map $X$ to $C \times (H \times W)$ and the pairwise relation matrix $R \in \mathbb{R}^{(H\times W)\times(H\times W)}$ is computed using

$$Q = \delta(W^QX) \tag{120}$$
$$K = \delta(W^KX) \tag{121}$$
$$R = Q^TK \tag{122}$$

The relation vector $r_i$ at position $i$ is defined by stacking pairwise relations at all positions:

$$r_i = [R(i,:); R(:,i)] \tag{123}$$

and the spatial relation-aware feature $y_i$ can be written as

$$Y_i = [g^c_{\text{avg}}(\delta(W^\varphi x_i)); \delta(W^\phi r_i)] \tag{124}$$

where $g^c_{\text{avg}}$ denotes global average pooling in the channel domain. Finally, the spatial attention score at position $i$ is given by

$$a_i = \sigma(W_2\delta(W_1y_i)) \tag{125}$$

RGA-C has the same form as RGA-S, except for taking the input feature map as a set of $H \times W$-dimensional features.

RGA uses global relations to generate the attention score for each feature node, so provides valuable structural information and significantly enhances the representational power. RGA-S and RGA-C are flexible enough to be used in any CNN network; Zhang et al. propose using them jointly in sequence to better capture both spatial and cross-channel relationships.

#### 3.6.10 Self-Calibrated Convolutions

Motivated by the success of group convolution, Liu et at [120] presented *self-calibrated convolution* as a means to enlarge the receptive field at each spatial location.

Self-calibrated convolution is used together with a standard convolution. It first divides the input feature $X$ into $X_1$ and $X_2$ in the channel domain. The self-calibrated convolution first uses average pooling to reduce the input size and enlarge the receptive field:

$$T_1 = \text{AvgPool}_r(X_1) \tag{126}$$

where $r$ is the filter size and stride. Then a convolution is used to model the channel relationship and a bilinear interpolation operator $Up$ is used to upsample the feature map:

$$X_1' = \text{Up}(\text{Conv}_2(T_1)) \tag{127}$$

Next, element-wise multiplication finishes the self-calibrated process:

$$Y_1' = \text{Conv}_3(X_1)\sigma(X_1 + X_1') \tag{128}$$

Finally, the output feature map of is formed:

$$Y_1 = \text{Conv}_4(Y_1') \tag{129}$$
$$Y_2 = \text{Conv}_1(X_2) \tag{130}$$
$$Y = [Y_1; Y_2] \tag{131}$$

Such self-calibrated convolution can enlarge the receptive field of a network and improve its adaptability. It achieves excellent results in image classification and certain downstream tasks such as instance segmentation, object detection and keypoint detection.

#### 3.6.11 SPNet

Spatial pooling usually operates on a small region which limits its capability to capture long-range dependencies and focus on distant regions. To overcome this, Hou et al. [124] proposed *strip pooling*, a novel pooling method capable of encoding long-range context in either horizontal or vertical spatial domains.

Strip pooling has two branches for horizontal and vertical strip pooling. The horizontal strip pooling part first pools the input feature $F \in \mathcal{R}^{C\times H\times W}$ in the horizontal direction:

$$y^1 = \text{GAP}^w(X) \tag{132}$$

Then a 1D convolution with kernel size 3 is applied in $y$ to capture the relationship between different rows and channels. This is repeated $W$ times to make the output $y_v$ consistent with the input shape:

$$y_h = \text{Expand}(\text{Conv1D}(y^1)) \tag{133}$$

Vertical strip pooling is performed in a similar way. Finally, the outputs of the two branches are fused using element-wise summation to produce the attention map:

$$s = \sigma(\text{Conv}^{1\times 1}(y_v + y_h)) \tag{134}$$
$$Y = sX \tag{135}$$

The strip pooling module (SPM) is further developed in the mixed pooling module (MPM). Both consider spatial and channel relationships to overcome the locality of convolutional neural networks. SPNet achieves state-of-the-art results for several complex semantic segmentation benchmarks.

#### 3.6.12 SCA-CNN

As CNN features are naturally spatial, channel-wise and multi-layer, Chen et al. [50] proposed a novel *spatial and channel-wise attention-based convolutional neural network* (SCA-CNN). It was designed for the task of image captioning, and uses an encoder-decoder framework where a CNN first encodes an input image into a vector and then an LSTM decodes the vector into a sequence of words. Given an input feature map $X$ and the previous time step LSTM hidden state $h_{t-1} \in \mathbb{R}^d$, a spatial attention mechanism pays more attention to the semantically useful regions, guided by LSTM hidden state $h_{t-1}$. The spatial attention model is:

$$a(h_{t-1}, X) = \tanh(\text{Conv}_1^{1\times 1}(X) \oplus W_1 h_{t-1}) \tag{136}$$
$$\Phi_s(h_{t-1}, X) = \text{Softmax}(\text{Conv}_2^{1\times 1}(a(h_{t-1}, X))) \tag{137}$$

where $\oplus$ represents addition of a matrix and a vector. Similarly, channel-wise attention aggregates global information first, and then computes a channel-wise attention weight vector with the hidden state $h_{t-1}$:

$$b(h_{t-1}, X) = \tanh((W_2\text{GAP}(X) + b_2) \oplus W_1 h_{t-1}) \tag{138}$$
$$\Phi_c(h_{t-1}, X) = \text{Softmax}(W_3(b(h_{t-1}, X)) + b_3) \tag{139}$$

Overall, the SCA mechanism can be written in one of two ways. If channel-wise attention is applied before spatial attention, we have

$$Y = f(X, \Phi_s(h_{t-1}, X\Phi_c(h_{t-1}, X)), \Phi_c(h_{t-1}, X)) \tag{140}$$

and if spatial attention comes first:

$$Y = f(X, \Phi_s(h_{t-1}, X), \Phi_c(h_{t-1}, X\Phi_s(h_{t-1}, X))) \tag{141}$$

where $f(\cdot)$ denotes the modulate function which takes the feature map $X$ and attention maps as input and then outputs the modulated feature map $Y$.

Unlike previous attention mechanisms which consider each image region equally and use global spatial information to tell the network where to focus, SCA-Net leverages the semantic vector to produce the spatial attention map as well as the channel-wise attention weight vector. Being more than a powerful attention model, SCA-CNN also provides a better understanding of where and what the model should focus on during sentence generation.

#### 3.6.13 GALA

Most attention mechanisms learn where to focus using only weak supervisory signals from class labels, which inspired Linsley et al. [122] to investigate how explicit human supervision can affect the performance and interpretability of attention models. As a proof of concept, Linsley et al. proposed the *global-and-local attention* (GALA) module, which extends an SE block with a spatial attention mechanism.

Given the input feature map $X$, GALA uses an attention mask that combines global and local attention to tell the network where and on what to focus. As in SE blocks, global attention aggregates global information by global average pooling and then produces a channel-wise attention weight vector using a multilayer perceptron. In local attention, two consecutive $1 \times 1$ convolutions are conducted on the input to produce a positional weight map. The outputs of the local and global pathways are combined by addition and multiplication. Formally, GALA can be represented as:

$$s_g = W_2\delta(W_1\text{GAP}(x)) \tag{142}$$
$$s_l = \text{Conv}_2^{1\times 1}(\delta(\text{Conv}_1^{1\times 1}(X))) \tag{143}$$
$$s_g^* = \text{Expand}(s_g) \tag{144}$$
$$s_l^* = \text{Expand}(s_l) \tag{145}$$
$$s = \tanh(a(s_g^* + s_l^*) + m \cdot (s_g^* s_l^*)) \tag{146}$$
$$Y = sX \tag{147}$$

where $a, m \in \mathbb{R}^C$ are learnable parameters representing channel-wise weight vectors.

Supervised by human-provided feature importance maps, GALA has significantly improved representational power and can be combined with any CNN backbone.

### 3.7 Spatial & Temporal Attention

Spatial & temporal attention combines the advantages of spatial attention and temporal attention as it adaptively selects both important regions and key frames. Some works [16], [130] compute temporal attention and spatial attention separately, while others [131] produce joint spatiotemporal attention maps. Further works focusing on capturing pairwise relations [177]. Representative spatial & temporal attention attentions and specific process $g(x)$ and $f(g(x), x)$ described as Eq. 1 are summarised in Tab. 8. We next discuss specific spatial & temporal attention mechanisms according to the order in Fig. 4.

#### 3.7.1 STA-LSTM

In human action recognition, each type of action generally only depends on a few specific kinematic joints [130]. Furthermore, over time, multiple actions may be performed. Motivated by these observations, Song et al. [130] proposed a joint spatial and temporal attention network based on LSTM [147], to adaptively find discriminative features and keyframes. Its main attention-related components are a spatial attention sub-network, to select important regions, and a temporal attention sub-network, to select key frames. The spatial attention sub-network can be written as:

$$s_t = U_s \tanh(W_{xs} X_t + W_{hs} h^s_{t-1} + b_{si}) + b_{so} \quad (148)$$

$$\alpha_t = \text{Softmax}(s_t) \quad (149)$$

$$Y_t = \alpha_t X_t \quad (150)$$

where $X_t$ is the input feature at time $t$, $U_s$, $W_{hs}$, $b_{si}$, and $b_{so}$ are learnable parameters, and $h^s_{t-1}$ is the hidden state at step $t-1$. Note that use of the hidden state $h$ means the attention process takes temporal relationships into consideration.

The temporal attention sub-network is similar to the spatial branch and produces its attention map using:

$$\beta_t = \delta(W_{xp} X_t + W_{hp} h^p_{t-1} + b_p). \quad (151)$$

It adopts a ReLU function instead of a normalization function for ease of optimization. It also uses a regularized objective function to improve convergence.

Overall, this paper presents a joint spatiotemporal attention method to focus on important joints and keyframes, with excellent results on the action recognition task.

#### 3.7.2 RSTAN

To capture spatiotemporal contexts in video frames, Du et al. [16] introduced *spatiotemporal attention* to adaptively identify key features in a global way.

The spatiotemporal attention mechanism in RSTAN consists of a spatial attention module and a temporal attention module applied serially. Given an input feature map $X \in \mathbb{R}^{D \times T \times H \times W}$ and the previous hidden state $h_{t-1}$ of an RNN model, spatiotemporal attention aims to produce a spatiotemporal feature representation for action recognition. First, the given feature map $X$ is reshaped to $\mathbb{R}^{D \times T \times (H \times W)}$, and we define $X(n,k)$ as the feature vector for the $k$-th location of the $n$-th frame. At time $t$, the spatial attention mechanism aims to produce a global feature $l_n$ for each frame, which can be written as

$$\alpha_t(n,k) = w_\alpha \tanh(W_h h_{t-1} + W_x X(n,k) + b_\alpha) \quad (152)$$

$$\alpha^*_t(n,k) = e^{\gamma_\alpha \alpha_t(n,k)} / \sum_{j=1}^{W \times H} e^{\gamma_\alpha \alpha_t(n,k)} \quad (153)$$

$$l_n = \sum_{k=1}^{H \times W} \alpha^*_t(n,k) X(n,k) \quad (154)$$

where $\gamma_\alpha$ is introduced to control the sharpness of the location-score map. After obtaining frame-wise features $\{l_1, \ldots, l_T\}$, RSTAN uses a temporal attention mechanism to estimate the importance of each frame feature

$$\beta_t(n) = w_\beta \tanh(W'_h h_{t-1} + W_l l(n) + b_\beta) \quad (155)$$

$$\beta^*_t(n) = e^{\gamma_\beta \beta_t(n)} / \sum_{j=1}^{T} e^{\gamma_\beta \beta_t(n)} \quad (156)$$

$$\phi_t = \sum_{n=1}^{T} \beta^*_t(n) l(n) \quad (157)$$

The spatiotemporal attention mechanism used in RSTAN identifies those regions in both spatial and temporal domains which are strongly related to the prediction in the current step of the RNN. This efficiently enhances the representation power of any 2D CNN.

#### 3.7.3 STA

Previous attention-based methods for video-based person re-identification only assigned an attention weight to each frame and failed to capture joint spatial and temporal relationships. To address this issue, Fu et al. [131] propose a novel *spatiotemporal attention* (STA) approach, which assigns attention scores for each spatial region in different frames without any extra parameters.

Given the feature maps of an input video $\{X_n | X_n \in \mathbb{R}^{C \times H \times W}\}_{n=1}^{N}$, STA first generates frame-wise attention maps by using the $l_2$ norm on the squares sum in the channel domain:

$$g_n(h,w) = \frac{|| \sum_{c=1}^{C} X_n(c,h,w)^2 ||_2}{\sum_{h=1}^{H} \sum_{w=1}^{W} || \sum_{c=1}^{C} X_n(c,h,w)^2 ||_2} \quad (158)$$

Then both the feature maps and attention maps are divided into $K$ local regions horizontally, each of which represents one part of the person. The spatial attention score for region $k$ is obtained using

$$s_{n,k} = \sum_{(i,j) \in \text{Region}_k} ||g_n(i,j)||_1 \quad (159)$$

To capture the relationships between regions in different frames, STA applies $l_1$ normalization to the attention scores in the temporal domain, using

$$S(n,k) = \frac{s_{n,k}}{\sum_{n=1}^{N} ||s_{n,k}||_1} \quad (160)$$

TABLE 8
Representative spatial & temporal attentions sorted by date. Action=action recognition, ReID = re-identification. Ranges means the ranges of attention map. S or H means soft or hard attention. $g(x)$ and $f(g(x), x)$ are the attention process described by Eq. 1. (A)element-wise product.(B) aggregate information via attention map.(I) emphasize key points in both spatial and temporal domains, (II)capture global information.

| Category | Method | Publication | Tasks | $g(x)$ | $f(g(x), x)$ | Ranges | S or H | Goals |
|---|---|---|---|---|---|---|---|---|
| Separately predict spatial & temporal attention | STA-LSTM [130] | AAAI2017 | Action | a)spatial: fuse hidden state -> MLP -> Softmax, b)temporal: fuse hidden state -> MLP -> ReLU | (A) | (0,1), (0,+$\infty$) | S | (I) |
| | RSTAN [16] | TIP2018 | Action | a)spatial: fuse hidden state -> MLP -> Softmax, b)temporal: fuse hidden state -> MLP -> Softmax | (B) | (0,1) | S | (I) (II) |
| Jointly predict spatial & temporal attention | STA [131] | AAAI2019 | ReID | a) tenporal: produce per-frame attention maps using $l_2$ norm b) spatial: obtain spatial scores for each patch by summation using $l_1$ norm. | (B) | (0,1) | S | (I) |
| Pairwise relation-based method | STGCN [177] | CVPR2020 | ReID | construct a patch graph using pairwise similarity | (B) | (0,1) | S | (I) |

Finally, STA splits the input feature map $X_i$ into $K$ regions $\{X_{n,1}, \ldots, X_{n,K}\}$ and computes the output using

$$Y^1 = [X_{\arg\max_n S(n,1),1}; \ldots; X_{\arg\max_n S(n,K),K}] \quad (161)$$

$$Y^2 = [\sum_{n=1}^{N} S(n,1)X_{n,1}; \ldots; \sum_{n=1}^{N} S(n,K)X_{n,K}] \quad (162)$$

$$Y = [Y^1; Y^2] \quad (163)$$

Instead of computing spatial attention maps frame by frame, STA considers spatial and temporal attention information simultaneously, fully using the discriminative parts in both dimensions. This reduces the influence of occlusion. Because of its non-parametric design, STA can tackle input video sequences of variable length; it can be combined with any 2D CNN backbone.

#### 3.7.4 STGCN

To model the spatial relations within a frame and temporal relations across frames, Yang et al. [177] proposed a novel *spatiotemporal graph convolutional network* (STGCN) to learn a discriminative descriptor for a video. It constructs a patch graph using pairwise similarity, and then uses graph convolution to aggregate information.

STGCN includes two parallel GCN branches, the temporal graph module and the structural graph module. Given the feature maps of a video, STGCN first horizontally partitions each frame into $P$ patches and applies average pooling to generate patch-wise features $x_1, \ldots, x_N$, where the total number of patches is $N = TP$. For the temporal module, it takes each patch as a graph node and construct a patch graph for the video, where the adjacency matrix $\widehat{A}$ is obtained by normalizing the pairwise relation matrix $E$, defined as

$$E(i,j) = (W^{\phi}x_i)^T W^{\phi}x_j \quad (164)$$

$$A(i,j) = E^2(i,j) / \sum_{j=1}^{N} E^2(i,j) \quad (165)$$

$$\widehat{A} = D^{-\frac{1}{2}}(A+I)D^{-\frac{1}{2}} \quad (166)$$

where $D(i,i) = \sum_{j=1}^{N}(A+I)(i,j)$. Given the adjacency matrix $\widehat{A}$, the $m$-th graph convolution can be found using

$$X^m = \widehat{A}X^{m-1}W^m + X^{m-1} \quad (167)$$

where $X \in \mathbb{R}^{N \times c}$ represents the hidden features for all patches and $W^m \in \mathbb{R}^{c \times c}$ denotes the learnable weight matrix for the $m$-th layer. For the spatial module, STGCN follows a similar approach of adjacency matrix and graph convolution, except for modeling the spatial relations of different regions within a frame.

Flattening spatial and temporal dimensions into a sequence, STGCN applies the GCN to capture the spatiotemporal relationships of patches across different frames. Pairwise attention is used to obtain the weighted adjacency matrix. By leveraging spatial and temporal relationships between patches, STGCN overcomes the occlusion problem while also enhancing informative features. It can used with any CNN backbone to process video.

## 4 Future Directions

We present our thoughts on potential future research directions.

### 4.1 Necessary and sufficient condition for attention

We find the Eq. 1 is a necessary condition but not a necessary and sufficient condition. For instance, GoogleNet [178] conforms to the above formula, but does not belong to the attention mechanisms. Unfortunately, we find it difficult to find a necessary and sufficient condition for all attention mechanisms. The necessary and sufficient conditions for the attention mechanism are still worth exploring which can promote our understanding of attention mechanisms.

### 4.2 General attention block

At present, a special attention mechanism needs to be designed for each different task, which requires considerable effort to explore potential attention methods. For instance, channel attention is a good choice for image classification,

while spatial attention is well-suited to dense prediction tasks such as semantic segmentation and object detection. Channel attention focuses on *what to pay attention to* while spatial attention considers *where to pay attention*. Based on this observation, we encourage consideration as to whether there could be a general attention block that takes advantage of all kinds of attention mechanisms. For example, a soft selection mechanism (branch attention) could choose between channel attention, spatial attention and temporal attention according to the specific task undertaken.

### 4.3 Characterisation and interpretability

Attention mechanisms are motivated by the human visual system and are a step towards the goal of building an interpretable computer vision system. Typically, attention-based models are understood by rendering attention maps, as in Fig. 9. However, this can only give an intuitive feel for what is happening, rather than precise understanding. However, applications in which security or safety are important, such as medical diagnostics and automated driving systems, often have stricter requirements. Better characterisation of how methods work, including modes of failure, is needed in such areas. Developing characterisable and interpretable attention models could make them more widely applicable.

### 4.4 Sparse activation

We visualize some attention map and obtains consistent conclusion with ViT [34] shown in Fig. 9 that attention mechanisms can produce sparse activation. There phenomenon give us a inspiration that sparse activation can achieve a strong performance in deep neural networks. It is worth noting that sparse activation is similar with human cognition. Those motivate us to explore which kind of architecture can simulate human visual system.

### 4.5 Attention-based pre-trained models

Large-scale attention-based pre-trained models have had great success in natural language processing [85], [179]. Recently, MoCoV3 [84], DINO [180], BEiT [85] and MAE [169] have demonstrated that attention-based models are also well suited to visual tasks. Due to their ability to adapt to varying inputs, attention-based models can deal with unseen objects and are naturally suited to transferring pretrained weights to a variety of tasks. We believe that the combination of pre-training and attention models should be further explored: training approach, model structures, pre-training tasks and the scale of data are all worth investigating.

### 4.6 Optimization

SGD [181] and Adam [182] are well-suited for optimizing convolutional neural networks. For visual transformers, AdamW [183] works better. Recently, Chen et al. [184] significantly improved visual transformers by using a novel optimizer, the *sharpness-aware minimizer* (SAM) [185]. It is clear that attention-based networks and convolutional neural networks are different models; different optimization methods may work better for different models. Investigating new optimzation methods for attention models is likely to be worthwhile.

### 4.7 Deployment

Convolutional neural networks have a simple, uniform structure which makes them easy to deploy on various hardware devices. However, it is difficult to optimize complex and varied attention-based models on edge devices. Nevertheless, experiments in [46], [47], [48] show that attention-based models provide better results than convolutional neural networks, so it is worth trying to find simple, efficient and effective attention-based models which can be widely deployed.

## 5 Conclusions

Attention mechanisms have become an indispensable technique in the field of computer vision in the era of deep learning. This survey has systematically reviewed and summarized attention mechanisms for deep neural networks in computer vision. We have grouped different attention methods according to their domain of operation, rather than by application task, and show that attention models can be regarded as an independent topic in their own right. We have concluded with some potential directions for future research. We hope that this work will encourage a variety of potential application developers to put attention mechanisms to use to improve their deep learning results. We also hope that this survey will give researchers a deeper understanding of various attention mechanisms and the relationships between them, as a springboard for future research.

## Acknowledgments

This work was supported by the Natural Science Foundation of China (Project 61521002, 62132012). We would like to thank Cheng-Ze Lu, Zhengyang Geng, Shilong liu, He Wang, Huiying Lu and Chenxi Huang for their helpful discussions and insightful suggestions.

## References

[1] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[2] M. Hayhoe and D. Ballard, "Eye movements in natural behavior," *Trends in cognitive sciences*, vol. 9, no. 4, pp. 188–194, 2005.

[3] R. A. Rensink, "The dynamic representation of scenes," *Visual cognition*, vol. 7, no. 1-3, pp. 17–42, 2000.

[4] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature reviews neuroscience*, vol. 3, no. 3, pp. 201–215, 2002.

[5] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," 2019.

[6] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, ser. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11211. Springer, 2018, pp. 3–19. [Online]. Available: https://doi.org/10.1007/978-3-030-01234-2_1

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," 2017.

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020.

[9] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.

[10] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua, "Neural aggregation network for video face recognition," 2017.

[12] Q. Wang, T. Wu, H. Zheng, and G. Guo, "Hierarchical pyramid diverse attention networks for face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[13] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2285–2294.

[14] B. Chen, W. Deng, and J. Hu, "Mixed high-order attention network for person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 371–381.

[15] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1347–1360, 2018.

[17] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, p. 1487–1500, Mar 2018. [Online]. Available: http://dx.doi.org/10.1109/TIP.2017.2774041

[18] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," 2017.

[19] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[20] Q. Guan, Y. Huang, Z. Zhong, Z. Zheng, L. Zheng, and Y. Yang, "Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification," *arXiv preprint arXiv:1801.09927*, 2018.

[21] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," 2015.

[22] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2019.

[23] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," 2017.

[24] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 057–11 066.

[25] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," 2018.

[26] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional shapecontextnet for point cloud recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[27] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, p. 187–199, Apr 2021. [Online]. Available: http://dx.doi.org/10.1007/s41095-021-0229-5

[28] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," 2020.

[29] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," 2017.

[30] Y. Wu and K. He, "Group normalization," 2018.

[31] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," 2014.

[32] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," 2016.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[35] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*. PMLR, 2015, pp. 2048–2057.

[36] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[37] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[39] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," 2020.

[40] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *International Conference on Computer Vision*, 2019.

[41] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. S. Huang, "Ccnet: Criss-cross attention for semantic segmentation," 2020.

[42] Z. Geng, M.-H. Guo, H. Chen, X. Li, K. Wei, and Z. Lin, "Is attention better than matrix decomposition?" in *International Conference on Learning Representations*, 2021.

[43] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," 2019.

[44] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," *arXiv preprint arXiv:2101.11986*, 2021.

[45] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," 2021.

[46] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.

[47] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *International Conference on Computer Vision (ICCV)*, Oct. 2021.

[48] L. Yuan, Q. Hou, Z. Jiang, J. Feng, and S. Yan, "Volo: Vision outlooker for visual recognition," 2021.

[49] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," 2021.

[50] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T. Chua, "SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 6298–6306. [Online]. Available: https://doi.org/10.1109/CVPR.2017.667

[51] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on Machine Learning*. Omnipress, 2010, pp. 807–814.

[52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[53] H. Zhang, K. J. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 7151–7160. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Context_Encoding_for_CVPR_2018_paper.html

[54] G. Zilin, X. Jiangtao, W. Qilong, and L. Peihua, "Global second-order pooling convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[55] H. Lee, H.-E. Kim, and H. Nam, "Srm : A style-based recalibration module for convolutional neural networks," 2019.

[56] Z. Yang, L. Zhu, Y. Wu, and Y. Yang, "Gated channel transformation for visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 794–11 803.

[57] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: Frequency channel attention networks," 2021.

[58] A. Diba, M. Fayyaz, V. Sharma, M. M. Arzani, R. Yousefzadeh, J. Gall, and L. V. Gool, "Spatio-temporal channel correlation networks for action classification," 2019.

[59] Z. Chen, Y. Li, S. Bengio, and S. Si, "You look twice: Gaternet for dynamic filter selection in cnns," 2019.

[60] H. Shi, G. Lin, H. Wang, T.-Y. Hung, and Z. Wang, "Spsequencenet: Semantic segmentation network on 4d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4574–4583.

[61] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," 2019.

[62] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5589–5598.

[63] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," 2018.

[64] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[65] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[66] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks," 2019.

[67] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, "Spatial pyramid based graph reasoning for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[68] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *International Conference on Computer Vision*, 2019. [Online]. Available: http://arxiv.org/abs/1908.07678

[69] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," *arXiv preprint arXiv:1904.11492*, 2019.

[70] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "$a^2$-nets: Double attention networks," 2018.

[71] Y. Chen, M. Rohrbach, Z. Yan, S. Yan, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," 2018.

[72] S. Zhang, S. Yan, and X. He, "Latentgnn: Learning efficient non-local relations for visual recognition," 2019.

[73] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation," 2021.

[74] M. Yin, Z. Yao, Y. Cao, X. Li, Z. Zhang, S. Lin, and H. Hu, "Disentangled non-local neural networks," 2020.

[75] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, "Beyond self-attention: External attention using two linear layers for visual tasks," 2021.

[76] H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," 2019.

[77] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[78] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1691–1703. [Online]. Available: https://proceedings.mlr.press/v119/chen20s.html

[79] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," 2021.

[80] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," 2020.

[81] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *CVPR*, 2021.

[82] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," 2021.

[83] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, "Query2label: A simple transformer way to multi-label classification," 2021.

[84] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," 2021.

[85] H. Bao, L. Dong, and F. Wei, "Beit: Bert pre-training of image transformers," 2021.

[86] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," 2021.

[87] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[88] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," 2015.

[89] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," 2016.

[90] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," 2017.

[91] Z. Li, E. Gavves, M. Jain, and C. G. M. Snoek, "Videolstm convolves, attends and flows for action recognition," 2016.

[92] K. Yue, M. Sun, Y. Yuan, F. Zhou, E. Ding, and F. Xu, "Compact generalized non-local network," 2018.

[93] X. Liu, Z. Han, X. Wen, Y.-S. Liu, and M. Zwicker, "L2g auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 989–997.

[94] A. Paigwar, O. Erkent, C. Wolf, and C. Laugier, "Attentional pointnet for 3d-object detection in point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[95] X. Wen, Z. Han, G. Youk, and Y.-S. Liu, "Cf-sis: Semantic-instance segmentation of 3d point clouds by context fusion with self-attention," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1661–1669.

[96] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian, "Modeling point clouds with self-attention and gumbel subset sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3323–3332.

[97] J. Xu, R. Zhao, F. Zhu, H. Wang, and W. Ouyang, "Attention-aware compositional network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2119–2128.

[98] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3492–3506, 2017.

[99] Z. Zheng, L. Zheng, and Y. Yang, "Pedestrian alignment network for large-scale person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3037–3045, 2018.

[100] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9215–9223.

[101] Z. Zhang, C. Lan, W. Zeng, X. Jin, and Z. Chen, "Relation-aware global attention for person re-identification," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2020, pp. 3186–3195.

[102] B. N. Xia, Y. Gong, Y. Zhang, and C. Poellabauer, "Second-order non-local attention networks for person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3760–3769.

[103] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan, "Diversified visual attention networks for fine-grained object classification," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1245–1256, 2017.

[104] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5209–5217.

[105] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4438–4446.

[106] Anonymous, "DAB-DETR: Dynamic anchor boxes are better queries for DETR," in *Submitted to The Tenth International Conference on Learning Representations*, 2022, under review. [Online]. Available: https://openreview.net/forum?id=oMI9PjOb9Jl

[107] G.-Y. Yang, X.-L. Li, R. R. Martin, and S.-M. Hu, "Sampling equivariant self-attention networks for object detection in aerial images," 2021.

[108] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," in *Proceedings of the IEEE/CVF*

*Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5012–5021.

[109] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 3744–3753. [Online]. Available: http://proceedings.mlr.press/v97/lee19d.html

[110] S. Xu, Y. Cheng, K. Gu, Y. Yang, S. Chang, and P. Zhou, "Jointly attentive spatial-temporal pooling networks for video-based person re-identification," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4733–4742.

[111] R. Zhang, J. Li, H. Sun, Y. Ge, P. Luo, X. Wang, and L. Lin, "Scan: Self-and-collaborative attention network for video person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4870–4882, 2019.

[112] D. Chen, H. Li, T. Xiao, S. Yi, and X. Wang, "Video person re-identification with competitive snippet-similarity aggregation and co-attentive snippet embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1169–1178.

[113] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *arXiv preprint arXiv:1507.06228*, 2015.

[114] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 510–519.

[115] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "Resnest: Split-attention networks," 2020.

[116] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 030–11 039.

[117] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," 2018.

[118] L. Yang, R.-Y. Zhang, L. Li, and X. Xie, "Simam: A simple, parameter-free attention module for convolutional neural networks," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 11 863–11 874. [Online]. Available: http://proceedings.mlr.press/v139/yang21o.html

[119] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156–3164.

[120] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[121] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, "Rotate to attend: Convolutional triplet attention module," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3139–3148.

[122] D. Linsley, D. Shiebler, S. Eberhardt, and T. Serre, "Learning what and where to attend," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=BJgLg3R9KQ

[123] A. G. Roy, N. Navab, and C. Wachinger, "Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks," *IEEE transactions on medical imaging*, vol. 38, no. 2, pp. 540–549, 2018.

[124] Q. Hou, L. Zhang, M.-M. Cheng, and J. Feng, "Strip Pooling: Rethinking spatial pooling for scene parsing," in *CVPR*, 2020.

[125] H. You, Y. Feng, R. Ji, and Y. Gao, "Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1310–1318.

[126] Q. Xie, Y.-K. Lai, J. Wu, Z. Wang, Y. Zhang, K. Xu, and J. Wang, "Mlcvnet: Multi-level context votenet for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 447–10 456.

[127] C. Wang, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Mancs: A multi-task attentional network with curriculum sampling for person re-identification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 365–381.

[128] T. Chen, S. Ding, J. Xie, Y. Yuan, W. Chen, Y. Yang, Z. Ren, and Z. Wang, "Abd-net: Attentive but diverse person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8351–8361.

[129] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 713–13 722.

[130] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *AAAI Conference on Artificial Intelligence*, 2017, pp. 4263–4270.

[131] Y. Fu, X. Wang, Y. Wei, and T. Huang, "Sta: Spatial-temporal attention for large-scale video-based person re-identification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, p. 8287–8294, Jul 2019. [Online]. Available: http://dx.doi.org/10.1609/aaai.v33i01.33018287

[132] L. Gao, X. Li, J. Song, and H. T. Shen, "Hierarchical lstms with adaptive attention for visual captioning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1112–1131, 2019.

[133] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai, "Stat: Spatial-temporal attention mechanism for video captioning," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 229–241, 2020.

[134] L. Meng, B. Zhao, B. Chang, G. Huang, W. Sun, F. Tung, and L. Sigal, "Interpretable spatio-temporal attention for video action recognition," 2019.

[135] B. He, X. Yang, Z. Wu, H. Chen, S.-N. Lim, and A. Shrivastava, "Gta: Global temporal attention for video action understanding," 2021.

[136] S. Li, S. Bak, P. Carr, and X. Wang, "Diversity regularized spatiotemporal attention for video-based person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 369–378.

[137] Z. Zhang, C. Lan, W. Zeng, and Z. Chen, "Multi-granularity reference-aided attentive feature aggregation for video-based person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 407–10 416.

[138] M. Shim, H.-I. Ho, J. Kim, and D. Wee, "Read: Reciprocal attention discriminator for image-to-video re-identification," in *European Conference on Computer Vision*. Springer, 2020, pp. 335–350.

[139] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, "Decoupled spatial-temporal transformer for video inpainting," 2021.

[140] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An attentive survey of attention models," 2021.

[141] Y. Xu, H. Wei, M. Lin, Y. Deng, K. Sheng, M. Zhang, F. Tang, W. Dong, F. Huang, and C. Xu, "Transformers in computational visual media: A survey," *Computational Visual Media*, vol. 8, no. 1, pp. 33–62, 2022.

[142] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on visual transformer," 2021.

[143] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," 2021.

[144] F. Wang and D. M. J. Tax, "Survey on the attention based rnn model and its applications in computer vision," 2016.

[145] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[146] P. Fang, J. Zhou, S. K. Roy, L. Petersson, and M. Harandi, "Bilinear attention networks for person retrieval," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8030–8039.

[147] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[148] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[149] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[150] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," 2017.

[151] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," 2019.

[152] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller, "Rethinking attention with performers," 2021.
[153] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," 2021.
[154] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," 2015.
[155] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters – improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[156] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 6230–6239. [Online]. Available: https://doi.org/10.1109/CVPR.2017.660
[157] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Lecture Notes in Computer Science*, p. 346–361, 2014. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10578-9_23
[158] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "Mlp-mixer: An all-mlp architecture for vision," 2021.
[159] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, and H. Jégou, "Resmlp: Feedforward networks for image classification with data-efficient training," 2021.
[160] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018.
[161] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.
[162] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.
[163] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2020.
[164] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," 2017.
[165] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.
[166] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng, "Deepvit: Towards deeper vision transformer," 2021.
[167] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," 2021.
[168] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, "Fuseformer: Fusing fine-grained information in transformers for video inpainting," 2021.
[169] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021.
[170] M.-H. Guo, Z.-N. Liu, T.-J. Mu, D. Liang, R. R. Martin, and S.-M. Hu, "Can attention enable mlps to catch up with cnns?" 2021.
[171] J. Li, J. Wang, Q. Tian, W. Gao, and S. Zhang, "Global-local temporal representations for video person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3958–3967.
[172] Z. Liu, L. Wang, W. Wu, C. Qian, and T. Lu, "Tam: Temporal adaptive module for video recognition," *arXiv preprint arXiv:2005.06803*, 2020.
[173] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," 2020.
[174] L. Spillmann, B. Dresp-Langley, and C.-H. Tseng, "Beyond the classical receptive field: the effect of contextual stimuli," *Journal of Vision*, vol. 15, no. 9, pp. 7–7, 2015.
[175] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2017.
[176] W. BS, D. NT, S. SG, T. C, and L. P, "Early and late mechanisms of surround suppression in striate cortex of macaque," 2005.
[177] J. Yang, W.-S. Zheng, Q. Yang, Y.-C. Chen, and Q. Tian, "Spatial-temporal graph convolutional network for video-based person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3289–3299.
[178] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.
[179] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.
[180] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021.
[181] N. Qian, "On the momentum term in gradient descent learning algorithms." *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999. [Online]. Available: http://dblp.uni-trier.de/db/journals/nn/nn12.html#Qian99
[182] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.
[183] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.
[184] X. Chen, C.-J. Hsieh, and B. Gong, "When vision transformers outperform resnets without pretraining or strong data augmentations," 2021.
[185] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," 2021.