*electronics*

MDPI

*Article*

# A Lightweight Context-Aware Feature Transformer Network for Human Pose Estimation

**Yanli Ma †, Qingxuan Shi * and Fan Zhang †**

Hebei Machine Vision Engineering Research Center, Hebei University, Baoding 071002, China; 20218019006@stumail.hbu.edu.cn (Y.M.); zhangfan2@stumail.hbu.edu.cn (F.Z.)

* Correspondence: qingxuanshi@hbu.edu.cn

† These authors contributed equally to this work.

**Abstract:** We propose a Context-aware Feature Transformer Network (CaFTNet), a novel network for human pose estimation. To address the issue of limited modeling of global dependencies in convolutional neural networks, we design the Transformerneck to strengthen the expressive power of features. Transformerneck directly substitutes $3 \times 3$ convolution in the bottleneck of HRNet with a Contextual Transformer (CoT) block while reducing the complexity of the network. Specifically, the CoT first produces keys with static contextual information through $3 \times 3$ convolution. Then, relying on query and contextualization keys, dynamic contexts are generated through two concatenated $1 \times 1$ convolutions. Static and dynamic contexts are eventually fused as an output. Additionally, for multi-scale networks, in order to further refine the features of the fusion output, we propose an Attention Feature Aggregation Module (AFAM). Technically, given an intermediate input, the AFAM successively deduces attention maps along the channel and spatial dimensions. Then, an adaptive refinement module (ARM) is exploited to activate the obtained attention maps. Finally, the input undergoes adaptive feature refinement through multiplication with the activated attention maps. Through the above procedures, our lightweight network provides powerful clues for the detection of keypoints. Experiments are performed on the COCO and MPII datasets. The model achieves a 76.2 AP on the COCO val2017 dataset. Compared to other methods with a CNN as the backbone, CaFTNet has a 72.9% reduced number of parameters. On the MPII dataset, our method uses only 60.7% of the number of parameters, acquiring similar results to other methods with a CNN as the backbone.

**Keywords:** human pose estimation; expressive power of features; feature refinement; global dependencies



**Citation:** Ma, Y.; Shi, Q.; Zhang, F. A Lightweight Context-Aware Feature Transformer Network for Human Pose Estimation. *Electronics* **2024**, *13*, 716. https://doi.org/10.3390/electronics13040716

Academic Editor: Jenhui Chen

Received: 7 January 2024
Revised: 3 February 2024
Accepted: 7 February 2024
Published: 9 February 2024



## 1. Introduction

Human pose estimation aims to localize human anatomical keypoints in an image. It has extensive applications in the field of computer vision, for instance, human action recognition [1–4], human pose tracking [5–9], 3D human pose estimation [10–13], and so on.

CNNs have obtained praiseworthy accomplishments in human pose estimation [14–18] in recent times. However, the convolution receptive field is confined, which means that the CNNs are unable to capture the dependence of remote interaction information. Recently, different methods [19–23] have been presented to remedy the shortcomings of the convolution limitation problem. A typical solution is to expand the receptive field to learn global dependency information, for example, by increasing the network depth [24–26]. However, deepening the network will lead to a sharp increase in the number of parameters. Recently, the Transformer [27] with self-attention has become a novel choice for a variety of visual tasks [28–30] for its ability to capture interactions between any pairwise positions. For human pose estimation, we aim to leverage the global dependencies captured by self-attention to provide contextual clues for occluded keypoints. Because the body keypoints themselves have certain connections,

as shown in Figure 1, global dependencies are able to improve the ability to locate difficult keypoints depending on easily detectable keypoints, thereby enhancing the performance of the overall network. There have been some recent works on CNNs [31,32] that directly model global dependencies with self-attention instead of convolution. Traditional self-attention design effectively limits the number of parameters and the operation speed of network models. The CoT [32] uses group convolution, which not only encodes context information into self-attention modules, improving feature representation, but also replaces $3 \times 3$ convolution in ResNet [33] while retaining a favorable parameter.

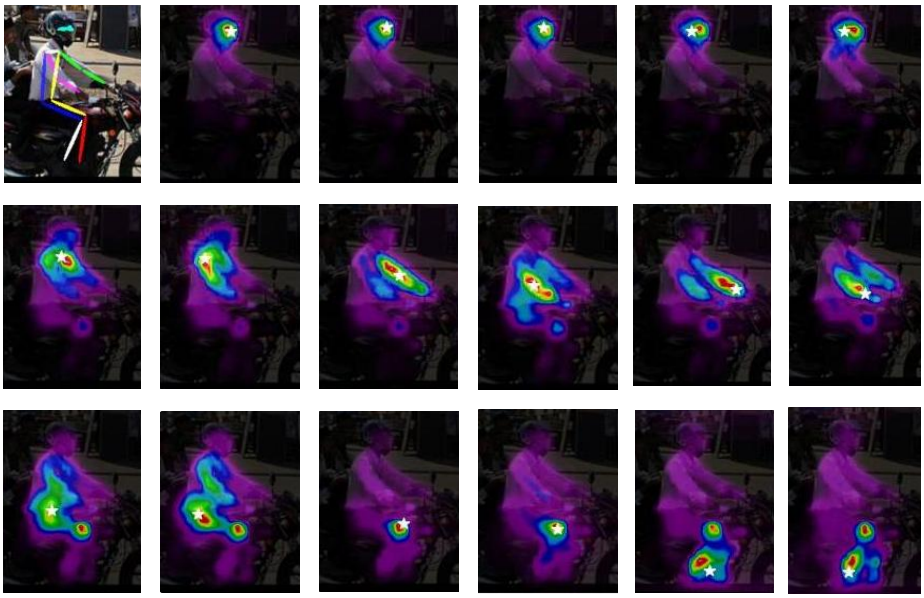


**Figure 1.** Attention map for the position of each predicted keypoint. We can see that the motorcycle covers the person's left ankle. The left ankle is predicted by relying on contextual information around the knee and the right leg joint.

In order to fully leverage the advantages of CNNs and self-attention mechanisms, some researchers have combined [34,35] them to extract features. The model design of TRPose [36] consists of sending the features extracted by a CNN with different resolutions into Encode for encoding, carrying out feature fusion, and finally carrying out downsampling to output key points. We believe that it would be better to fuse features of different resolutions before Encode because there are still some drawbacks in multi-scale networks [37–39]. Each subnetwork of multi-scale neural networks has a different resolution in order to exchange information between multiple resolution representations during feature fusion. High-resolution features with more detailed information can precisely locate the position information of keypoints. Low-resolution features with a larger receptive field can capture global information about the human pose. In feature fusion, the accuracy of keypoint detection will be enhanced if the model can fully exploit the benefits of high and low resolution. However, some existing methods [40,41] ignore the differences between features at different resolutions, resulting in the undesired fusion of noise features. To bridge the differences between features at different resolutions, an effective approach is to utilize the attention mechanism. Because attention can make the network highlight or restrain information through learning, the network can better grasp information we need to pay attention to. Recently, some scholars have conducted relevant research [42–44]. For example, CBAM [45] considers the channel and spatial dimensions and finally generates spatial attention maps. Therefore, we also expect our model to have the ability to learn information in both the channel and spatial dimensions.

Based on the above studies, in this article, we put forth a Lightweight [46] Context-aware Feature Transformer Network (CaFTNet) based upon HRNet to improve the network efficacy by enhancing the localization accuracy of occluded keypoints. Firstly, to strengthen

the semantic features of contextual information [47,48], we design a Transformerneck structure. Transformerneck directly replaces 3 × 3 convolution in the bottleneck with a Contextual Transformer (CoT) block while reducing the complexity of the network. Then, inspired by the CBAM, to further refine the features of the fusion output, we design an Attention Feature Aggregation Module (AFAM). Due to the diversity of human poses, CBAM is still insufficient for spatial processing as it only employs a 7 × 7 convolution filter for feature fusion, while spatial attention is decided by the value of each pixel, not the region of 7 × 7. So, we propose an ARM to activate the obtained features. Therefore, our method further reinforces the feature fusion in multi-scale networks and ameliorates the output features. On the COCO dataset [49], our model achieves better results than other methods with a CNN as the backbone. Furthermore, notably, the model has a 72.9% reduced number of parameters. On the MPII dataset [50], our method uses 60.7% of the number of parameters, acquiring similar results to other methods with a CNN as the backbone. In summary, our contributions are (1) We propose a lightweight network architecture that can predict the keypoints of two-dimensional human posture from input images. (2) We evaluate the impact of our method on human pose estimation data. (3) We demonstrate that our method can achieve better pose estimation results compared to directly using CNN output sequences for keypoint encoding.

## 2. Related Work

### *2.1. Human Pose Estimation*

CNNs have achieved tremendous success in the field of human pose estimation [51]. Hourglass [36] belongs to the hourglass type of network structure, which can perceive more global information. The CPN [52] has two stages, GlobalNet and RefineNet, which can alleviate the keypoint detection problem. Simple baseline [24] adds some transpose convolutional layers to restore the resolution. It highlights the importance of high-resolution feature maps. HRNet [14] is a network with high-resolution representations through the whole process, which repeats multi-scale fusion to improve the representation power of feature maps. Accordingly, HRNet achieves impressive results on multiple benchmark datasets. However, HRNet still falls within the category of CNNs, facing the issue of limited receptive fields. Therefore, global information needs to be ameliorated.

### *2.2. Attention-Enhanced Convolution*

Convolution is dependent on a fixed convolution kernel to gather information, which leads to the inability of CNNs to establish global dependencies. Multiple existing approaches to image attention can compensate for the problem of restriction of the convolution receptive field. Therefore, many scholars have explored the application of attention to improve the capability of CNNs. SENet [43] models the interactions between the channels by using global mean pooling and two fully connected layers. On the basis of SENet, ECANet [44] was proposed. A local cross-channel interaction strategy, without decreasing the dimensions, was designed, which further improves the performance. CBAM [45] calculates attention maps in the channel and spatial directions to better learn useful information in feature maps.

Recently, with the introduction of self-attention in Transformers, the interest of researchers has been aroused due to its powerful global dependence modeling ability. Some works [53,54] have shown that self-attention modules can be proposed as individual blocks which can wholly substitute for the convolutions in HRNet. Self-attention can effectively capture interactions between any paired position; however, pairwise query–key relationships are learned individually from isolated query–key pairs without taking into account the abundant contextual information between them during the learning process. This seriously restricts the self-attention learning ability of two-dimensional feature maps for visual representation learning. Most recently, ref. [31] replaced 3 × 3 convolutions with self-attention in the final stage of the network. Ref. [32] replaced 3 × 3 convolution in each

bottleneck by leveraging CoT blocks, which can take full advantage of the context of the query–key pair to model global dependencies.

### *2.3. HRNet*

HRNet [14] utilizes a stem to rapidly downsample the input features. As shown in Figure 2, HRNet can be segmented into four stages. The first stage mainly consists of a high-resolution subnetwork. Starting from the second stage, a low-resolution subnetwork is added to each stage. The resolution of the new subnetwork is half of the lowest resolution of the previous stage. Each stage will interact with information through multi-resolution blocks.

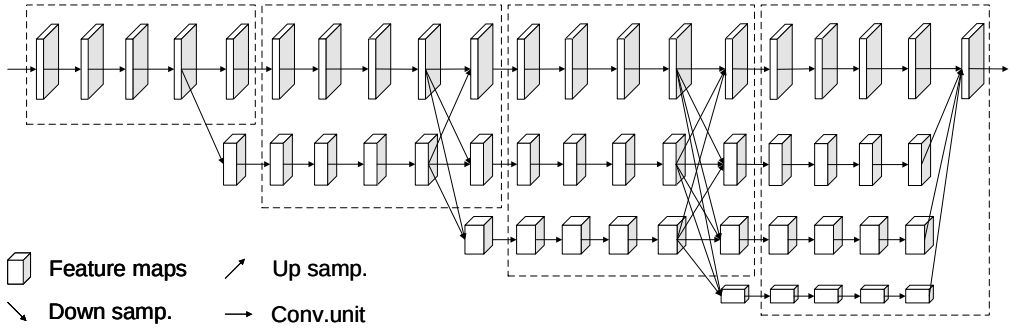



**Figure 2.** The architecture of HRNet.

HRNet has achieved remarkable success as a feature extractor. The problem of the limited receptive field that is inherent in the convolution operation needs to be solved. HRNet is unable to establish long-term dependencies, resulting in incorrect estimation of some human poses. For this reason, this paper proposes a Lightweight Context-aware Feature Transformer Network (CaFTNet). CaFTNet firstly capitalizes on the CoT block to enhance the expressiveness of features. Then, in feature fusion, CaFTNet exploits an AFAM to enhance the representative power of the output feature maps. Our final results are also better.

## 3. Methods

In this section, we put forward CaFTNet to better perform feature extraction. Figure 3 depicts the framework of our presented model. To begin with, we briefly review the framework of our CaFTNet. Then, we introduce Transformerneck and the AFAM in detail.

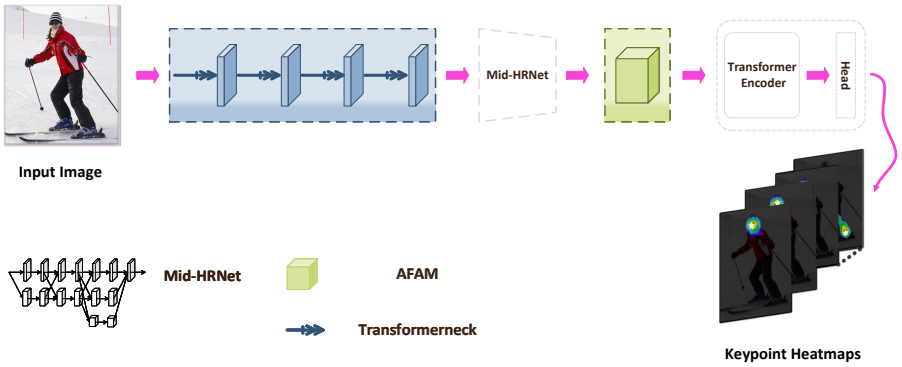



**Figure 3.** Overview of CaFTNet. Firstly, Transformerneck is used to extract preliminary input features with contextual information. Secondly, the input features continue to encode the feature information through Mid-HRNet. Then, the AFAM further refines the contextual features. Next, a Transformer Encoder Layer encodes the position representation of keypoints. Finally, a head predicts keypoint heatmaps. Mid-HRNet refers to the second and third stages of the HRNet.

### *3.1. Context-Aware Feature Transformer Network (CaFTNet)*

The purpose of this paper is to enhance the representational ability of the feature maps and decrease the network model size in pose estimation. The overall architecture of CaFTNet is revealed in Figure 3 in order to show the whole process of our method more clearly. The entire flow diagram of CaFTNet is shown in Figure 4. CaFTNet uses HRNet as the backbone and enhances it with the presented Transformerneck and Attention Feature Aggregation Module (AFAM).

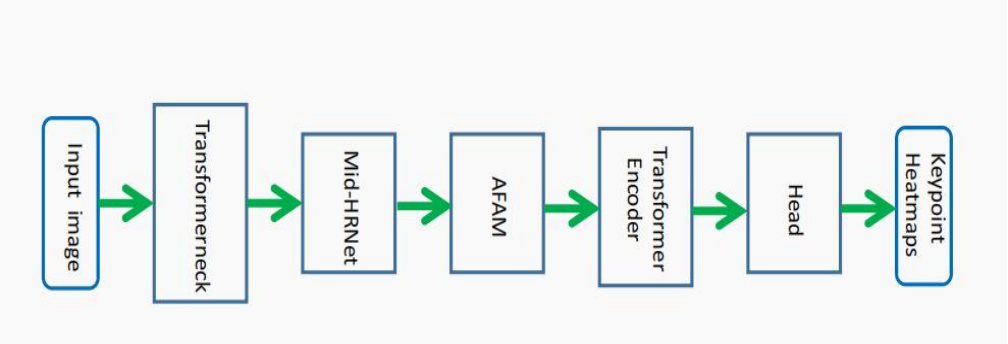



**Figure 4.** The entire flow diagram of CaFTNet.

First, the proposed Transformerneck is used to extract preliminary input features with contextual information. It is represented by the box with a blue dashed line. Transformerneck replaces $3 \times 3$ convolution with a CoT while keeping the bottleneck framework unchanged. Secondly, these input features continue to encode the feature information through Mid-HRNet. Then, we place an AFAM on the head of the neural network to further refine the enriched contextual features. The AFAM is represented by a box with a green dashed line. The AFAM successively determines attention maps in the channel and spatial dimensions. An adaptive refinement module (ARM) is exploited to activate the obtained attention maps. The input undergoes adaptive feature refinement through multiplication with the activated attention maps. Next, the output of the AFAM goes through a Transformer Encoder Layer to encode the position representation of keypoints. Finally, a head is attached to the Transformer Encoder output to predict keypoint heatmaps.

### *3.2. Transformerneck*

For a middle input $X \in R^{H \times W \times C}$, an output $H$ is first obtained through a $1 \times 1$ convolution and a nonlinear activation layer. The output $H$ is sent into the CoT (as shown in the green rectangle enclosed in Figure 5). $H$ is represented by:

$$H = ReLU(BN(Conv_{1\times1}(X))). \tag{1}$$

$H$ will then be defined by $K$, $Q$ and $V$ along three different paths. $K$ first produces contextualized $K_1$ through $3 \times 3$ convolutions. The formula of $K_1$ is described as follows:

$$K_1 = Conv_{3\times3}(K). \tag{2}$$

Then, $K_1$ and $Q$ are concatenated and the result of this operation generates an attention map $M$ by two series of $1 \times 1$ convolutions. The formula for $M$ is:

$$M = Conv_{1\times1}(ReLU(Conv_{1\times1}(Concat(K_1, Q)))). \tag{3}$$

Next, $V$ first passes through $1 \times 1$ convolution to obtain $V_1$, and the feature map $K_2$ can be computed as follows:

$$V_1 = Conv_{1\times1}(V), \tag{4}$$

$$K_2 = F(V_1 \otimes M), \tag{5}$$

where $F(\otimes)$ denotes a matrix multiplication operation. The final output $Z$ of the CoT is thus calculated as the fusion of $K_1$ and $K_2$. $Z$ continues to produce $T$ through a nonlinear activation layer and a $1 \times 1$ convolution layer. $T$ and a shortcut connection are added element-wise to produce $Y$ with context relations. Finally, $Y$ is sent to the next module via the Relu activation function. See Algorithm 1 for the overall process.

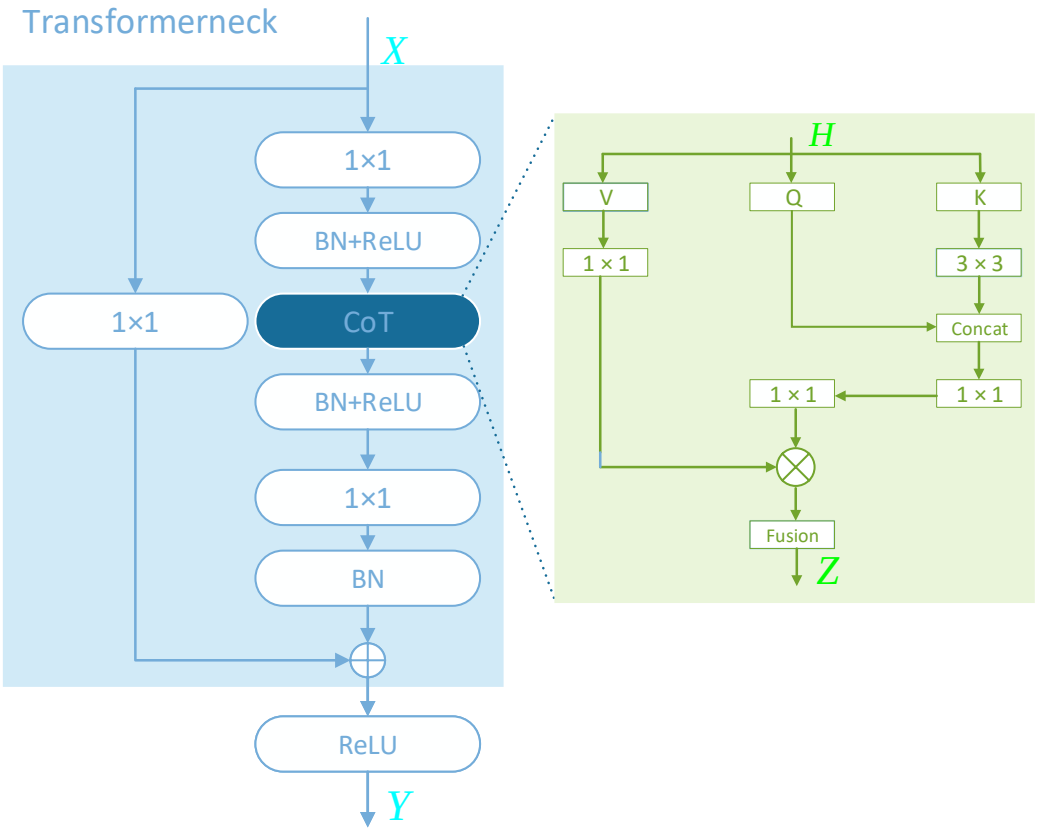



**Figure 5.** The overall structure of Transformerneck.

**Algorithm 1:** Transformerneck

**Input:** feature map
**Output:** feature map
1: $X$ <- an intermediate feature map
2: $X$ <- $K$
3: $X$ <- $Q$
4: $X$ <- $V$
5: $K_1$ <- use Equation (1)
6: $M$ <- use Equation (2)
7: $V_1$ <- use Equation (3)
8: $K_2$ <- use Equation (4)
9: $Y$ <- $K_1$ + $K_2$
10: **return** $Y$

### *3.3. Attention Feature Aggregation Module (AFAM)*

To begin with, we consider feature map $F \in R^{H \times W \times C}$ as an input in Figure 6. $F$ passes through a CS module, generating the spatial attention map $F_s$ that we require. This process can be described in the two steps in Figure 7. The first step, $F_c$, can be described as:

$$F_c = Sigmoid(MLP(AvgPool(F)) + MLP(MaxPool(F))) \otimes F. \quad (6)$$

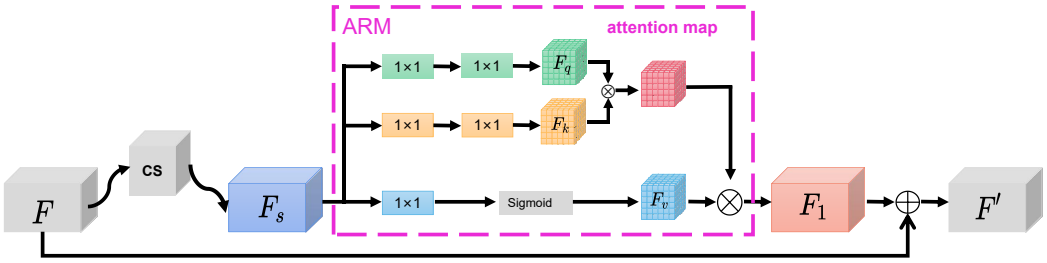



**Figure 6.** The overall structure of the Attention Feature Aggregation Module. CS: Channel Attention Module, Spatial Attention Module.

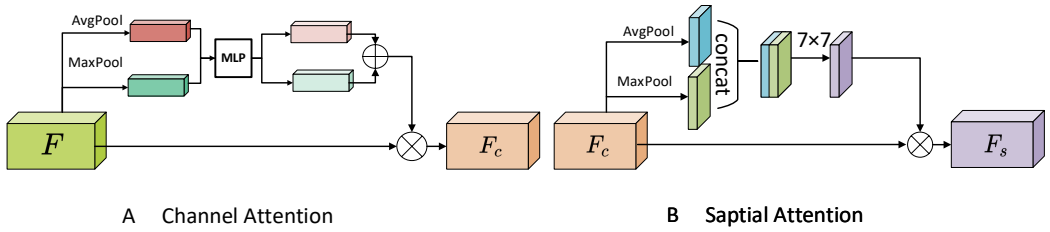



**Figure 7.** The overall structure of the Convolution Block Attention Module.

In the second step, $F_c$ is fed to the spatial attention model to obtain $F_s$. $F_s$ is adopted as:

$$F_s = Sigmoid(Conv_{7\times7}([AvgPool(F_c); MaxPool(F_c)])) \otimes F_c. \quad (7)$$

Next, $F_s$ is reshaped to feature sequences $F_q$, $F_k$ and $F_v$ in order to model the spatial context relations of the corresponding features. The detailed description of this process is as follows:

(1) $F_s$ obtains the spatial context feature $F_v$ through a $1 \times 1$ convolution and a sigmoid layer in the last row. $F_v$ is represented as:

$$F_v = Sigmoid(Conv_{1\times1}(F_s)). \quad (8)$$

(2) $F_s$ rearranges the spatially related context features together, respectively, through two $1 \times 1$ convolutions and a non-linear activation layer to obtain $F_q$ and $F_k$. $F_q$ and $F_k$ are represented as:

$$F_q = Con_{1\times1}(ReLU(Conv_{1\times1}(F_s))), \quad (9)$$

$$F_k = Con_{1\times1}(ReLU(Conv_{1\times1}(F_s))). \quad (10)$$

(3) $F_q$ and $F_k$ are multiplied element-wise to obtain an attention map with contextual relationships, which is subsequently applied to the features to recalibrate the output features $F_1$. $F_1$ is represented as:

$$F_1 = F_v \otimes Sigmoid(F_q \otimes F_k). \quad (11)$$

Finally, $F_1$ and $F$ are added element-wise to achieve $F'$. $F'$ is expressed as:

$$F' = F_1 \oplus F. \quad (12)$$

See Algorithm 2 for the overall process.

**Algorithm 2:** AFAM

**Input:** feature map
**Output:** feature map
1: $F$ <- an intermediate feature map
2: $M_c$ <- use Equation (5)
3: $F_c$ <- use Equation (6)
4: $M_s$ <- use Equation (7)
5: $F_s$ <- use Equation (8)
6: $F_v$ <- use Equation (9)
7: $F_q$ <- use Equation (10)
8: $F_k$ <- use Equation (11)
9: $F_1$ <- use Equation (12)
10: $F^{'}$ <- $F_1 + F$
11: **return** $F^{'}$

## 4. Experiments

### *4.1. Model Variants*

Based on HRNet, we present a Lightweight Context-aware Feature Transformer Network. In our structure, there are two different depths of CNNs to extract the input features. The detailed setup information is presented in Table 1. The network utilized by CaFTNet-R is ResNet. The backbone network utilized by CaFTNet-H4 is HRNet-W48. From Table 2, we can see that the model achieves the best result when the network employs CaFTNet-H4.

**Table 1.** Parameter configuration information for the different CaFTNet models.

| Model | Backbone | Layers | Heads | Flops | Params |
|---|---|---|---|---|---|
| CaFTNet-R | ResNet | 4 | 8 | 5.29 G | 5.55 M |
| CaFTNet-H3 | HRNet-W32 | 4 | 1 | 8.46 G | 17.03 M |
| CaFTNet-H4 | HRNet-W48 | 4 | 1 | 8.73 G | 17.30 M |

**Table 2.** Ablation study with different backbones.

| Model | Backbone | AP | AR | Flops | Params |
|---|---|---|---|---|---|
| CaFTNet-R | ResNet | 73.7 | 79.0 | 5.29 G | 5.55 M |
| CaFTNet-H3 | HRNet-W32 | 75.6 | 80.9 | 8.46 G | 17.03 M |
| CaFTNet-H4 | HRNet-W48 | 76.2 | 81.2 | 8.73 G | 17.30 M |

### *4.2. Technical Details*

Our model takes advantage of a top-down [55–57] approach. The experimental environment configuration is as follows: Two RTX 2080s are used. The Python Version is 3.7. The framework is PyTorch 1.10.0. The network model was optimized utilizing the Adam [58] optimizer during training, with an initial learning rate of 0.001 and 0.00001 at 220 rounds. The network was trained for 230 rounds with a batch size of 16 for each GPU. Because the sizes of the pictures in the dataset are different, the pictures were modified by image pre-processing. Here, images were cropped to 256 × 192 in the COCO dataset and 256 × 256 in the MPII dataset.

### *4.3. Results on the COCO Dataset*

#### 4.3.1. Dataset and Evaluation Metrics

The COCO dataset [49] has more than 200,000 images and 250,000 instances, each containing up to 17 human keypoints. The network model was trained on the train2017 dataset, and the network model was verified and tested on val2017 (including 5000 images)

and test-dev2017 (including 20,000 images) datasets. Our model was evaluated using the Object Keypoint Similarity (*OKS* ) on the COCO dataset. *OKS* defines the similarity between different human keypoints, $AP^{50}$ indicates the accuracy of the keypoints at *OKS* = 0.5 and $AP^{75}$ is the accuracy of the keypoints at *OKS* = 0.75. *mAP* is defined as the mean accuracy value of the predicted keypoints at 10 thresholds of *OKS* = 0.50, 0.55 . . . 0.90, 0.95. $AP^M$ is utilized to describe the accuracy of the detection of medium-sized keypoints, and $AP^L$ represents the accuracy of large-sized keypoint detection. The formula for the *OKS* is:

$$OKS = \frac{\sum_i exp\left(-d_i^2/2s^2k_i^2\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}, \tag{13}$$

where $d_i$ is the Euclidean distance between the $i$-th predicted keypoint coordinate and the corresponding groundtruth, $v_i$ is the visibility flag of the keypoint, $s$ is the object scale, and $k_i$ is a keypoint-specific constant.

#### 4.3.2. Quantitative Results

The models were compared for their performance on the COCO val2017 dataset, and the results are shown in Table 3. The numbers of parameters and GFLOPs were calculated from the human pose estimation network model. The experimental results show that in terms of the number of parameters and GFLOPs, the CaFTNet model achieves a better performance with a fewer number of parameters and GFLOPs, at 17.3 M and 8.73 G, respectively. CaFTNet-H4 acquires an AP score of 76.2 with an input size of 256 × 192, better than other models with the same input size. In contrast to TransPose-R-A4 [59], CaFTNet-R has an 8.3% lower number of parameters, but the AP score is increased by 1.1. In contrast to ResNet-152 [33], our CaFTNet-R model exhibits a better performance, utilizing only 7.2% of the model parameters. Comparing the complex network model of HRNet-W48 [14], CaFTNet acquires a good AP score with a much lower complexity. Table 4 exhibits the results of our approach and other approaches on the COCO test-dev2017 dataset. Our CaFTNet-H4 achieves an AP of 75.5. Due to effective perceptual context and semantic information, CaFTNet achieves a good balance between accuracy and complexity.

**Table 3.** Comparison results with different other methods on the COCO val2017 dataset. CaFTNet-R and CaFTNet-H achieve good results in terms of parameter numbers and calculation speeds.

| Model | Input Size | AP | AR | Flops | Params |
|---|---|---|---|---|---|
| ResNet-50 [33] | 256 × 192 | 70.4 | 76.3 | 8.9 G | 34.0 M |
| ResNet-101 [33] | 256 × 192 | 71.4 | 76.3 | 12.4 G | 53.0 M |
| ResNet-152 [33] | 256 × 192 | 72 | 77.8 | 35.3 G | 68.6 M |
| TransPose-R-A3 [59] | 256 × 192 | 71.7 | 77.1 | 8.0 G | 5.2 M |
| TransPose-R-A4 [59] | 256 × 192 | 72.6 | 78.0 | 8.9 G | 6.0 M |
| CaFTNet-R | 256 × 192 | 73.7 | 79.0 | 5.29 G | 5.55 M |
| HRNet-W32 [14] | 256 × 192 | 74.7 | 79.8 | 7.2 G | 28.5 M |
| HRNet-W48 [14] | 256 × 192 | 75.1 | 80.4 | 14.6 G | 63.6 M |
| TransPose-H-A4 [59] | 256 × 192 | 75.3 | 80.3 | 17.5 G | 17.3 M |
| TransPose-H-A6 [59] | 256 × 192 | 75.8 | 80.8 | 21.8 G | 17.5 M |
| TokenPose-L/D6 [60] | 256 × 192 | 75.4 | 80.4 | 9.1 G | 20.8 M |
| TokenPose-L/D24 [60] | 256 × 192 | 75.8 | 80.9 | 11.0 G | 27.5 M |
| CaFTNet-H3 | 256 × 192 | 75.6 | 80.9 | 8.46 G | 17.03 M |
| CaFTNet-H4 | 256 × 192 | 76.2 | 81.2 | 8.73 G | 17.30 M |

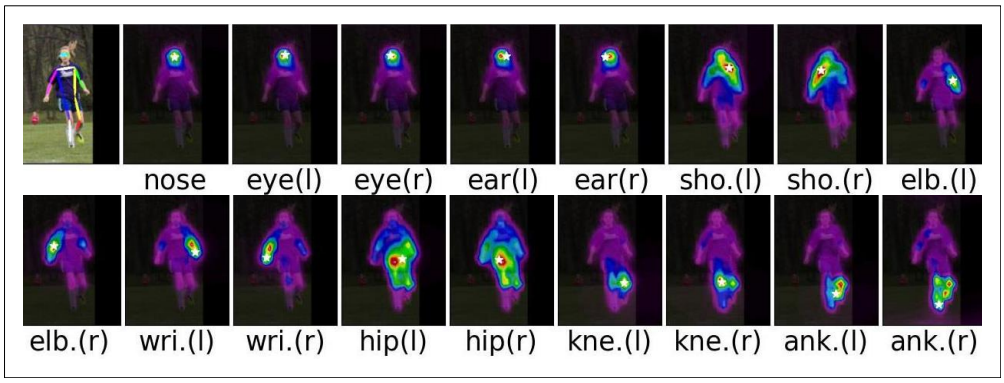
**Table 4.** Comparison results with different other methods on the COCO test-dev2017 dataset. CaFTNet-R and CaFTNet-H achieve good results in terms of parameter numbers and calculation speeds.

| Model | Input Size | AP | $AP^{50}$ | $AP^{75}$ | $AP^{m}$ | $AP^{l}$ | Params |
|---|---|---|---|---|---|---|---|
| G-RMI [18] | 357 × 257 | 64.9 | 85.5 | 71.3 | 62.3 | 70.0 | 42.6 M |
| Integral [61] | 256 × 256 | 67.8 | 88.2 | 74.8 | 63.9 | 74.0 | 45.0 M |
| CPN [52] | 384 × 288 | 72.1 | 91.4 | 80.0 | 68.7 | 77.2 | 58.8 M |
| RMPE [16] | 320 × 256 | 72.3 | 89.2 | 79.1 | 68.0 | 78.6 | 28.1 M |
| SimpleBaseline [24] | 384 × 288 | 73.7 | 91.9 | 81.8 | 70.3 | 80.0 | 68.6 M |
| HRNet-W32 [14] | 384 × 288 | 74.9 | 92.5 | 82.8 | 71.3 | 80.9 | 28.5 M |
| HRNet-W48 [14] | 256 × 192 | 74.2 | 92.4 | 82.4 | 70.9 | 79.7 | 63.6 M |
| TransPose-H-A4 [59] | 256 × 192 | 74.7 | 91.6 | 82.2 | 71.4 | 80.7 | 17.3 M |
| TransPose-H-A6 [59] | 256 × 192 | 75.0 | 92.2 | 82.3 | 71.3 | 81.1 | 17.5 M |
| TokenPose-L/D6 [60] | 256 × 192 | 74.9 | 90.0 | 81.8 | 71.8 | 82.4 | 20.8 M |
| TokenPose-L/D24 [60] | 256 × 192 | 75.1 | 90.3 | 82.5 | 72.3 | 82.7 | 27.5 M |
| CaFTNet-H3 | 256 × 192 | 75.0 | 90.0 | 82.0 | 71.5 | 82.5 | 17.03 M |
| CaFTNet-H4 | 256 × 192 | 75.5 | 90.4 | 82.8 | 72.5 | 83.3 | 17.30 M |

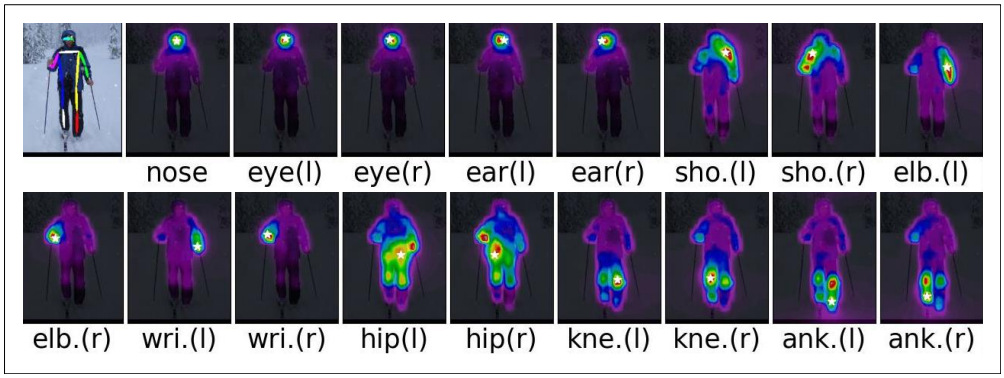
#### 4.3.3. Qualitative Comparisons

**Different keypoints rely on different regions.** The attention mechanism mimics human cognitive awareness of a particular piece of information, zooming in on key details to pay more attention to essential aspects of the data. Self-attention encodes high-level interaction and context information by extracting relationships between input sequence markers. It mainly calculates the similarity between two graphs (K, Q) of the same input. Traditional self-attention measures the attention matrix using only isolated query key pairs, but rich context information is left between the keys. For our proposed Transformerneck, first, the keys represent the upper and lower cultures via performing a 3 × 3 convolution on all adjacent keys in a 3 × 3 grid. This reflects the static context between local neighbors. After that, we input the connections between the key features of the context into two continuous convolutions to produce an attention matrix. The relationship between each query and all keywords is then used as a guide to predict the final output. We find that for the keypoints of the head like the nose, eyes, etc., the positioning depends mainly on the interdependencies between them, and it is worth noting that the prediction of the wrist or knees depends on favorable cues around them. For instance, the prediction of the right knee depends on the left knee and the right lower limb. A closer look shows that our network has the ability to derive useful information from its relevant parts for keypoints to predict targets. In this way, we can understand why the model can predict the occluded keypoints (e.g., the occluded right knee in Figure 8a).

**Visualization.** The results are compared and visualized for the COCO dataset in Figure 9. The source image is displayed at the top of the picture, the middle row displays the HRNet results, and our results are displayed in the bottom row. The objects (enclosed in red circles) were not detected by HRNet in the first and second columns of images likely due to occlusion by other objects. HRNet may have treated the undetected objects as background during the detection process. In comparison, the proposed AFAM in this paper can weight the features during information fusion, allowing for a better prediction of occluded objects. The final result shows the advantages of our method for occluded objects. Everything else on the subject is comparable, but for occluded objects, our model demonstrates its advantage. As shown in images in the third column, our approach can accurately detect occluded keypoints. This is because our model introduces a CoT, which allows for better capturing of contextual information, providing beneficial cues for detecting occluded keypoints. As a result, our approach achieves superior results.




(**a**)




(**b**)

**Figure 8.** Visualization of heatmaps predicting keypoint locations and their dependent regions for different input pictures according to the CaFTNet-R model. (**a**) Visualization of image 1. (**b**) Visualization of image 2.

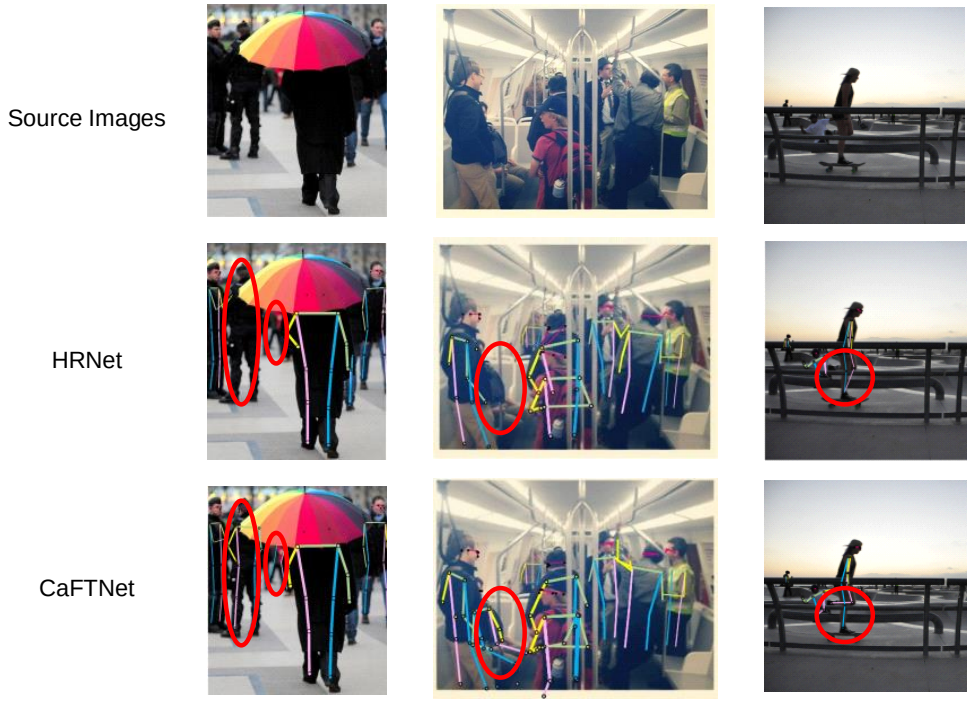



**Figure 9.** Qualitative comparisons using the COCO dataset.

### *4.4. Results on the MPII Dataset*

#### 4.4.1. Dataset and Evaluation Metrics

The MPII dataset [49] is a single-person pose estimation dataset that captures the whole-body pose of people in real scenes, and includes 28,821 training images and 11,701 test images; it is a benchmark dataset for single-person pose estimation. The training and validation sets contain 22,246 and 2958 images, respectively. The standard evaluation index of the MPII dataset is *PCKh* (head-normalized percentage of correct keypoints), using the head segment length as the normalization reference. *PCKh* is expressed as:

$$PCKh_i = \frac{\sum_p \delta\left(\frac{d_{pi}}{L_p^{head}} \leq 0.5\right)}{\sum_p 1}, \tag{14}$$

$$PCKh_{mean} = \frac{\sum_p \sum_i \delta\left(\frac{d_{pi}}{L_p^{head}} \leq 0.5\right)}{\sum_p \sum_i 1}, \tag{15}$$

where $i$ represents the $i$-th keypoint, $p$ represents the $p$-th pedestrian, $d_{pi}$ is the Euclidean distance between the $p$-th individual's $i$-th predicted keypoint coordinate and the corresponding groundtruth, $\delta(\cdot)$ represents the indicator function, and $L_p^{head}$ indicates the $p$-th head segment length. We report the $PCKh@0.5(\alpha = 0.5)$ score for a fair comparison with other methods.

#### 4.4.2. Quantitative Results

Table 5 presents the results of the different approaches on the MPII dataset. In terms of the mean, we can see that the result of our method is 90.4, and the other terms are also comparable. In more detail, we can see from Table 5 that our final results are only higher by 0.1 compared to the baseline method. In particular, for ankle detection, our method is 0.3 higher than TokenPose-L/D24. Additionally, each test result of our method outperforms the baseline method and thus proves that our method achieves a better performance on this dataset. More importantly, our method uses only 61 percent of the total number of baseline method parameters. Our results are better compared to SimpleBaseline-Res152, and our number of parameters is decreased by 74.8%.

**Table 5.** Results on the MPII dataset validation set (PCKh@0.5).

| Model | Hea | Sho | Elb | Wri | Hip | Kne | Ank | Mean | Params |
|---|---|---|---|---|---|---|---|---|---|
| SimpleBaseline-Res50 [24] | 96.4 | 95.3 | 89.0 | 83.2 | 88.4 | 84.0 | 79.6 | 88.5 | 34.0 M |
| SimpleBaseline-Res101 [24] | 96.9 | 95.9 | 89.5 | 84.4 | 88.4 | 84.5 | 80.7 | 89.1 | 53.0 M |
| SimpleBaseline-Res152 [24] | 97.0 | 95.9 | 90.0 | 85.0 | 89.2 | 85.3 | 81.3 | 89.6 | 68.6 M |
| HRNet-W32 [14] | 97.1 | 95.9 | 90.3 | 86.4 | 89.1 | 87.1 | 83.3 | 90.3 | 28.5 M |
| TokenPose-L/D24 [60] | 97.1 | 95.9 | 90.4 | 86.0 | 89.3 | 87.1 | 82.5 | 90.2 | 28.1 M |
| CaFTNet-H4 | 97.2 | 96.1 | 90.5 | 86.5 | 89.3 | 86.9 | 82.8 | 90.4 | 17.3 M |

#### 4.4.3. Qualitative Comparisons

We reveal some contrasting results on the MPII dataset in Figure 10. The source image is displayed at the top of the picture, the middle row displays the HRNet results, and our results are displayed in the last row. As shown by the visualization results of the third line, our method can correctly detect the occluded keypoints not detected by HRNet. We found above from the results in Table 5 that our method has an advantage in the detection of ankle joints, at 0.3 higher than the baseline method, mainly because our model can better capture the contextual information and provide favorable clues for blocked keypoints. Thus, our approach achieves better results.

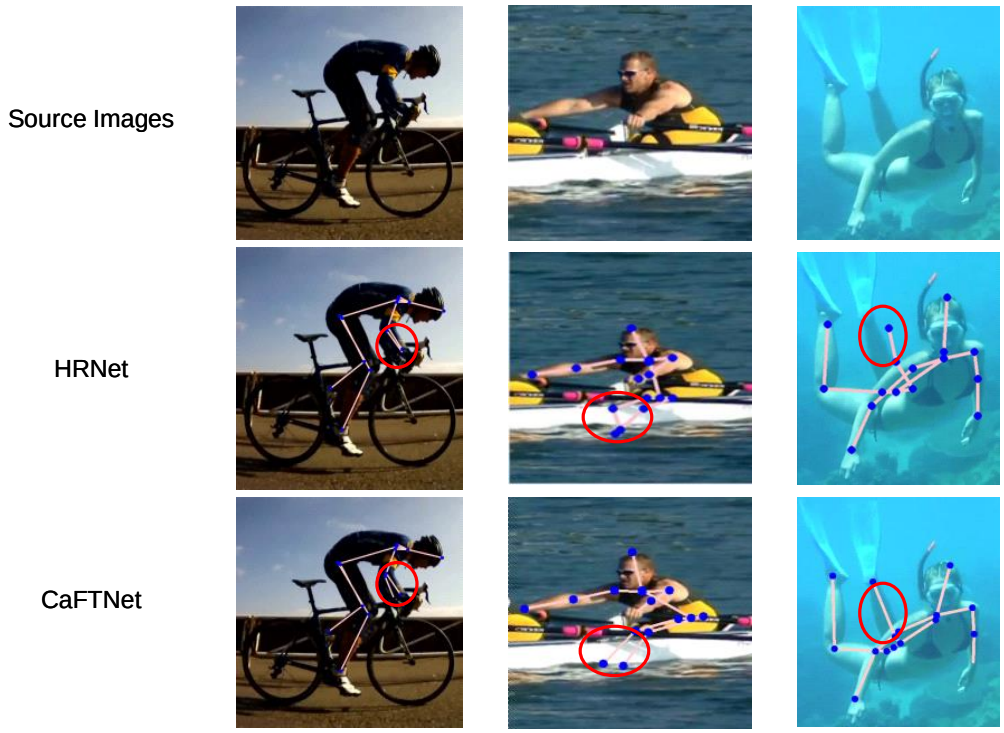



**Figure 10.** Qualitative comparisons on the MPII dataset.

### *4.5. Ablation Experiments*

Ablation experiments were performed for training validation on the COCO dataset, considering the role of Transformerneck and the AFAM in the network model.

#### 4.5.1. Transformerneck

In this paper, two sets of ablation experiments are devised to verify the effect of employing a bottleneck or a Transformerneck separately based on our different network models. When implementing our network model with CaFTNet-H, we replaced the bottleneck structure with our proposed Transformerneck while keeping the other structures unchanged. The structures using a Transformerneck achieved an AP of 75.7, see Table 6. We also report the results of replacing the bottleneck with Transformerneck when employing CaFTNet-R. The results with the Transformerneck yield a value that is 0.6 higher than the results employing a bottleneck. The results highlight the utility of exploiting contextual information for decoding subsequent features.

**Table 6.** The effects of the CoT for different models on the COCO dataset.

| Model | Bottleneck | Transformerneck | AP |
|---|---|---|---|
| CaFTNet-R | ✓ | | 72.6 |
| CaFTNet-R | | ✓ | 73.2 |
| CaFTNet-H | ✓ | | 75.3 |
| CaFTNet-H | | ✓ | 75.7 |

#### 4.5.2. Attention Feature Aggregation Module (AFAM)

We investigated the effects of different attention mechanisms on the experimental results, for example, (i) SENet; (ii) ECANet; (iii) CBAM; (iv) AFAM. Due to their different use of the feature map, their influence on the results of the experiment is also different. SENet [43] models the interactions between the channels by using global mean pooling and two fully connected layers. On the basis of SENet, ref. [44] proposed ECANet. A

local cross-channel interaction strategy, without decreasing the dimensions, was designed, which further improves the performance. Ref. [45] proposed CBAM to focus more on spatial attention maps. The AFAM compensates for the lack of spatial processing in CBAM, and it keeps the network focused on more desirable features. Table 7 presents the results from our different experiments. Although the difference between them is small, we are conscious that our proposed AFAM results in a 0.5 higher than SE. The results expose that more spatial information is needed when solving feature fusion problems.

**Table 7.** Contrasting results for the COCO dataset under different attention mechanisms.

| Model | Baseline | SE | ECA | CBAM | AFAM | AP |
|---|---|---|---|---|---|---|
| CaFTNet-R | ✓ | | | | | 72.6 |
| CaFTNet-R | ✓ | ✓ | | | | 72.7 |
| CaFTNet-R | ✓ | | ✓ | | | 72.8 |
| CaFTNet-R | ✓ | | | ✓ | | 73.0 |
| CaFTNet-R | ✓ | | | | ✓ | 73.2 |

#### 4.5.3. Complexity Analysis

We used the number of model parameters to evaluate the spatial complexity. The maximum number of parameters in our network is 17.3M. As can be seen from Table 3, the final AP result of our method is 76.2, which is 0.4 higher than that of currently popular methods. In terms of time complexity, the time consumption of our model is mainly reflected in self-attention in the Transformer, that is, the quadratic complexity. However, our method mainly focuses on the modification of a CNN, so we only analyzed GFLOPs in terms of the time complexity above. Because our model uses a top-down mode to scale all cropped images to a fixed size, we mainly trained our model with a size of 256 $\times$ 192. For input resolution, the sequence length of the Transformer in the CaFTNet-R and CaFTNet-H models is 768 and 3072, respectively. For our current model, a higher input resolution (e.g., 384 $\times$ 288) not only results in high computational costs in the self-focusing layer due to the quadratic complexity, but also decreases the scalability and efficiency. In order to perform a fair comparison with other methods, we only conducted experiments with an input image resolution of 256 $\times$ 192.

## 5. Conclusions

In this article, we put forth a Lightweight Context-aware Feature Transformer Network (CaFTNet) for enhancing the efficacy of human pose estimation models. Since CNNs cannot capture long-range dependencies between global regions, we devise the Transformerneck. Furthermore, to bolster the representation power of the fusion output feature maps, we design an Attention Feature Aggregation Module (AFAM). Extensive experiments carried out on the COCO and MPII datasets corroborate the applicability of the proposed approach.

However, our method has some limitations in terms of accuracy. Due to some design defects of the model, our method has the problem of inaccurate positioning when dealing with complex data, which means that the model is unable to obtain the best attitude estimation results at present. Therefore, in future work, we aim to further optimize our human pose estimation model and design a lightweight pose estimator that is more suitable for the current task in order to improve the accuracy of pose estimation.

**Author Contributions:** Methodology, Q.S. and Y.M.; software, Y.M.; validation, Y.M., F.Z. and Q.S.; formal analysis, Y.M.; investigation, Y.M., F.Z. and Q.S.; writing—original draft preparation, Y.M.; writing—review and editing, Y.M., F.Z. and Q.S.; visualization, Y.M. and F.Z.; supervision, Q.S.; funding acquisition, Q.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by The Natural Science Foundation of Hebei Province (F2019201451).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data is contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References


1. Si, C.; Chen, W.; Wang, W.; Wang, L.; Tan, T. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1227–1236.
2. Yang, C.; Xu, Y.; Shi, J.; Dai, B.; Zhou, B. Temporal pyramid network for action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 591–600.
3. Rahnama, A.; Esfahani, A.; Mansouri, A. Adaptive Frame Selection In Two Dimensional Convolutional Neural Network Action Recognition. In Proceedings of the 2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), Mazandaran, Iran, 28–29 December 2022; IEEE: New York, NY, USA, 2022; pp. 1–4.
4. Sun, Z.; Ke, Q.; Rahmani, H.; Bennamoun, M.; Wang, G.; Liu, J. Human action recognition from various data modalities: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 3200–3225. [CrossRef] [PubMed]
5. Snower, M.; Kadav, A.; Lai, F.; Graf, H.P. 15 keypoints is all you need. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6738–6748.
6. Ning, G.; Pei, J.; Huang, H. Lighttrack: A generic framework for online top-down human pose tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 1034–1035.
7. Wang, M.; Tighe, J.; Modolo, D. Combining detection and tracking for human pose estimation in videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11088–11096.
8. Rafi, U.; Doering, A.; Leibe, B.; Gall, J. Self-supervised keypoint correspondences for multi-person pose estimation and tracking in videos. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XX 16; Springer: New York, NY, USA, 2020; pp. 36–52.
9. Kwon, O.H.; Tanke, J.; Gall, J. Recursive bayesian filtering for multiple human pose tracking from multiple cameras. In Proceedings of the Asian Conference on Computer Vision, Kyoto, Japan, 30 November–4 December 2020.
10. Kocabas, M.; Athanasiou, N.; Black, M.J. Vibe: Video inference for human body pose and shape estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5253–5263.
11. Chen, H.; Guo, P.; Li, P.; Lee, G.H.; Chirikjian, G. Multi-person 3d pose estimation in crowded scenes based on multi-view geometry. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part III 16; Springer: New York, NY, USA, 2020; pp. 541–557.
12. Kolotouros, N.; Pavlakos, G.; Black, M.J.; Daniilidis, K. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2252–2261.
13. Qiu, H.; Wang, C.; Wang, J.; Wang, N.; Zeng, W. Cross view fusion for 3d human pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4342–4351.
14. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. *arXiv* **2019**, arXiv:1902.09212.
15. Cai, Y.; Wang, Z.; Luo, Z.; Yin, B.; Du, A.; Wang, H.; Zhang, X.; Zhou, X.; Zhou, E.; Sun, J. Learning delicate local representations for multi-person pose estimation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part III 16; Springer: New York, NY, USA, 2020; pp. 455–472.
16. Fang, H.S.; Xie, S.; Tai, Y.W.; Lu, C. Rmpe: Regional multi-person pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2334–2343.
17. Newell, A.; Huang, Z.; Deng, J. Associative embedding: End-to-end learning for joint detection and grouping. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
18. Papandreou, G.; Zhu, T.; Kanazawa, N.; Toshev, A.; Tompson, J.; Bregler, C.; Murphy, K. Towards accurate multi-person pose estimation in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4903–4911.
19. Wei, S.E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional pose machines. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4724–4732.
20. Yang, W.; Li, S.; Ouyang, W.; Li, H.; Wang, X. Learning feature pyramids for human pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1281–1290.
21. Jiang, W.; Jin, S.; Liu, W.; Qian, C.; Luo, P.; Liu, S. PoseTrans: A Simple Yet Effective Pose Transformation Augmentation for Human Pose Estimation. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Proceedings, Part V; Springer: New York, NY, USA, 2022; pp. 643–659.

22. Tang, W.; Yu, P.; Wu, Y. Deeply learned compositional models for human pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 190–206.
23. Ren, F. Distilling Token-Pruned Pose Transformer for 2D Human Pose Estimation. *arXiv* **2023**, arXiv:2304.05548.
24. Xiao, B.; Wu, H.; Wei, Y. Simple baselines for human pose estimation and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 466–481.
25. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
26. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
28. Raaj, Y.; Idrees, H.; Hidalgo, G.; Sheikh, Y. Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4620–4628.
29. Luvizon, D.C.; Picard, D.; Tabia, H. Multi-task deep learning for real-time 3D human pose estimation and action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 2752–2764. [CrossRef] [PubMed]
30. Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; Hoi, S.C. Deep learning for person re-identification: A survey and outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 2872–2893. [CrossRef] [PubMed]
31. Srinivas, A.; Lin, T.Y.; Parmar, N.; Shlens, J.; Vaswani, A. Bottleneck Transformers for Visual Recognition. *arXiv* **2021**, arXiv:2101.11605.
32. Li, Y.; Yao, T.; Pan, Y.; Mei, T. Contextual Transformer Networks for Visual Recognition. *arXiv* **2021**, arXiv:2107.12292.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
34. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
35. Pan, X.; Ge, C.; Lu, R.; Song, S.; Chen, G.; Huang, Z.; Huang, G. On the integration of self-attention and convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 815–825.
36. Wang, D.; Xie, W.; Cai, Y.; Li, X.; Liu, X. Transformer-based rapid human pose estimation network. *Comput. Graph.* **2023**, *116*, 317–326. [CrossRef]
37. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning. PMLR, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.
38. Pfister, T.; Charles, J.; Zisserman, A. Flowing convnets for human pose estimation in videos. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1913–1921.
39. Chu, X.; Yang, W.; Ouyang, W.; Ma, C.; Yuille, A.L.; Wang, X. Multi-context attention for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1831–1840.
40. Cheng, B.; Xiao, B.; Wang, J.; Shi, H.; Huang, T.S.; Zhang, L. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5386–5395.
41. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
42. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-excite: Exploiting feature context in convolutional neural networks. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018; Volume 31.
43. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
44. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
45. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
46. Chen, H.; Jiang, X.Y. Shift Pose: A Lightweight Transformer-like Neural Network for Human Pose Estimation. *Sensors* **2022**, *22*, 7264. [CrossRef] [PubMed]
47. Peng, J.; Wang, H.; Yue, S.; Zhang, Z. Context-aware co-supervision for accurate object detection. *Pattern Recognit.* **2022**, *121*, 108199. [CrossRef]
48. Zhang, J.Q.Z.; Jiang, X. Spatial Context-Aware Object-Attentional Network for Multi-Label Image Classification. *IEEE Trans. Image Process.* **2023**, *32*, 3000–3012. [CrossRef] [PubMed]

49. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: New York, NY, USA, 2014; pp. 740–755.
50. Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B. 2d human pose estimation: New benchmark and state of the art analysis. In Proceedings of the IEEE Conference on computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3686–3693.
51. Samkari, E.M.; Alghamdi, M. Human Pose Estimation Using Deep Learning: A Systematic Literature Review. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1612–1659. [CrossRef]
52. Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; Sun, J. Cascaded pyramid network for multi-person pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7103–7112.
53. Zhao, H.; Jia, J.; Koltun, V. Exploring self-attention for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10076–10085.
54. Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; Shlens, J. Stand-alone self-attention in vision models. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
55. Huang, J.; Zhu, Z.; Guo, F.; Huang, G. The devil is in the details: Delving into unbiased data processing for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5700–5709.
56. Zhang, F.; Zhu, X.; Dai, H.; Ye, M.; Zhu, C. Distribution-aware coordinate representation for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7093–7102.
57. Li, W.; Wang, Z.; Yin, B.; Peng, Q.; Du, Y.; Xiao, T.; Yu, G.; Lu, H.; Wei, Y.; Sun, J. Rethinking on multi-stage networks for human pose estimation. *arXiv* **2019**, arXiv:1901.00148.
58. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
59. Yang, S.; Quan, Z.; Nie, M.; Yang, W. Transpose: Keypoint localization via transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 11802–11812.
60. Li, Y.; Zhang, S.; Wang, Z.; Yang, S.; Yang, W.; Xia, S.T.; Zhou, E. Tokenpose: Learning keypoint tokens for human pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 11313–11322.
61. Sun, X.; Xiao, B.; Wei, F.; Liang, S.; Wei, Y. Integral human pose regression. In Proceedings of the European conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 529–545.