agriculture

MDPI

Article

DepthFormer: A High-Resolution Depth-Wise Transformer for Animal Pose Estimation

Sicong Liu [1], Qingcheng Fan [1], Shanghao Liu [1] and Chunjiang Zhao [1,2,*]

[1] College of Information Engineering, Northwest A&F University, Yangling, Xianyang 712100, China
[2] Research Center of Information Technology, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China
* Correspondence: zhaocj@nercita.org.cn

Abstract: Animal pose estimation has important value in both theoretical research and practical applications, such as zoology and wildlife conservation. A simple but effective high-resolution Transformer model for animal pose estimation called DepthFormer is provided in this study to address the issue of large-scale models for multi-animal pose estimation being problematic with limited computing resources. We make good use of a multi-branch parallel design that can maintain high-resolution representations throughout the process. Along with two similarities, i.e., sparse connectivity and weight sharing between self-attention and depthwise convolution, we utilize the delicate structure of the Transformer and representative batch normalization to design a new basic block for reducing the number of parameters and the amount of computation required. In addition, four PoolFormer blocks are introduced after the parallel network to maintain good performance. Benchmark evaluation is performed on a public database named AP-10K, which contains 23 animal families and 54 species, and the results are compared with the other six state-of-the-art pose estimation networks. The results demonstrate that the performance of DepthFormer surpasses that of other popular lightweight networks (e.g., Lite-HRNet and HRFormer-Tiny) when performing this task. This work can provide effective technical support to accurately estimate animal poses with limited computing resources.

Keywords: animal pose estimation; depthformer; multi-resolution representations; depthwise convolution

Citation: Liu, S.; Fan, Q.; Liu, S.; Zhao, C. DepthFormer: A High-Resolution Depth-Wise Transformer for Animal Pose Estimation. *Agriculture* **2022**, *12*, 1280. https://doi.org/10.3390/agriculture12081280

Academic Editors: Hao Li, Xiaoshuai Wang, Qianying Yi and Kai Liu

Received: 12 July 2022
Accepted: 16 August 2022
Published: 22 August 2022

1. Introduction

Ethology heavily relies on the observation and analysis of animal behavior in its natural state [1]. For research into how social animals think, a precise estimation of animal pose information is necessary [2]. Estimating an animal's poses can assist in tracking, protecting, and re-identifying the species [3]. The assessment of an animal's pose and the subsequent processing are also helpful in assessing an animal's health and identifying probable ailments [4]. Although extensive natural behavior recordings of mammalian animals may be made with modern multi-media capabilities, pose estimation from these recordings would be challenging to complete manually [1].

Deep learning has been used in recent years to estimate animal poses. ResNets and deconvolutional layers are the two networks that DeepLabCut [5] cascades. One is used to create spatial probability densities for the locations of body parts while the other is pretrained on ImageNet. Stacked DenseNet was suggested by Graving et al. [6] as a simple but effective software toolbox to increase robustness and speed up processing. A brand-new, sizable open source dataset of keypoint labels for macaques in the field was created by Rollyn et al. [7]. There is only one species in it, and it is used to train and test deep learning models for macaque pose estimation. AP-10K [8], a large-scale benchmark containing 23 animal families and 54 species for general mammal pose estimation, was

proposed. The existing large networks such as HRNet [9] and SimpleBaseline [10] show state-of-the-art performance on the large dataset.

Two-dimensional pose estimation is a difficult pattern recognition task in computer vision. There are two types of recent primary pose estimation paradigms: bottom-up methods and top-down methods. The bottom-up paradigm regresses joint positions from the same body directly [11–13]. The steps are as follows: first, locate all of the joints and then assign them to the appropriate body [14]. The top-down paradigm first uses an object detector to find the body instance, then uses the bounding box region to get the body's joints [15,16]. With the use of bounding boxes, the method is typically less sensitive to the scale variance of objects than the bottom-up paradigm [10,17]. The second method is selected as our process because of the superior accuracy.

The Transformer, a powerful network model described by Ashish et al. [18], has attained state-of-the-art performance in numerous Natural Language Processing applications. When compared to convolution networks, the Vision Transformer (ViT) [19] applies the Transformer to the computer vision task and has achieved significant progress in the classification task. By leveraging Transformer, DETR [20] has made significant progress in target detection by adjusting on the basis of ViT. In order to further reduce the computational load of the Swin Transformer [21], focal self-attention [22] was designed, which includes local fine-grained and coarse-grained global interactions. However, MetaFormer [23] emphasizes how the power of Transformer stems from its well-designed structure rather than self-attention. The study indicates that, while *Pooling* or even *Identity* alters self-attention, the model still performs admirably.

There is a strong correlation between self-attention mechanism and convolution operation [24]. It can be proved theoretically that self-attention mechanism can represent any convolution layer [25]. Both dynamic convolution and lightweight convolution have achieved performance equivalent to self-attention mechanism with shorter running time on WMT '14 English German translation and WMT'17 Chinese-English translation [26]. Convolution operation also has more advantages in pretraining [27]. Based on these works in the field of NLP, Qi et al. [28] further explored the similarity between depthwise convolution and self-attention mechanisms in computer vision tasks. This work provides a guide for us to design a lightweight model.

Aiming at the pose estimation task, there are a number of works achieving state-of-the-art performance, such as Hourglass [17], HRNet [9], HRFormer [29], PRTR [15], and UniFormer [30]. These methods have shown a strong ability among large networks in position-sensitive tasks, such as pose estimation and semantic segmentation. It is also worth considering not only accuracy but also computation cost and the number of parameters. In practical applications, models often run on platforms (e.g., hardware) with limited computing resources. In order to deal with this issue, a series of lightweight networks (e.g., MobileNet [31,32], ShuffleNet [33,34], Lite-HRNet [35]) have been proposed.

The keypoints in an image must be recognized in the pose recognition task. High-resolution representation is helpful in enhancing the task's effect. There are two widely used methods for obtaining high-resolution representations. The first method involves a cascaded encoder–decoder structure, which cascades various up-sampling and down-sampling blocks to obtain high-resolution representations, such as Hourglass Net and U-Net [36]. Second, throughout the entire process, the network maintains high-resolution representation. There are numerous parallel branches in the network, each with a distinct resolution. The parallel branches are permitted to interact after each stage. In this manner, the low-resolution semantic data is combined with the high-resolution representation, which may be more accurate in terms of space (e.g., HRNet, HRFormer, Lite-HRNet, FastNet [37]). The first strategy involves frequent up- and down-sampling, which results in information loss and poor performance. As a result, we use the second strategy, which is to maintain high-resolution representation throughout the process.

Existing lightweight networks can be divided into two categories. One perspective is to directly apply the networks used for classification tasks to pose estimation [31–34].

The other is to mitigate the dilemma of spatial resolution and real-time computation speed, such as in feature pyramid networks [38] and BiSeNet [39].

These studies are mainly related to human pose estimation, and a few studies focus on animal pose estimation because there is a lack of datasets. Moreover, the fewer pay attention to lightweight networks for the pose estimation task. Lite-HRNet maintains a lightweight architecture and keeps high-resolution representations all the way through. In light of this approach and combining the core of MetaFormer [23], we designed a novel network for animal pose estimation.

Our main contributions are the following:

- We propose DepthFormer, which is designed by utilizing three-stage parallel branches with the idea of multi-scale representation fusion. The network keeps rich spatial information from start to end.
- Based on two similarities (sparse connectivity and weight sharing) between self-attention and depthwise convolution, we rely on a structure with Transformer that has strong power and representative batch normalization, which could strengthen the representation of specific instances to design a new basic block. The purpose of this is to reduce the quantity of parameters and calculation.
- For balancing the contradiction between the computation and performance of the model, we add four PoolFormer blocks after the parallel network, improving performance with minimal additional computing costs.
- DepthFormer is state-of-the-art in terms of computing cost and performance tradeoff on the AP-10K benchmark, outperforming ShuffleNet, HRFormer-T, and Lite-HRNet.

2. Materials and Methods

2.1. AP-10K Dataset

The AP-10K dataset [8] consists of 10,015 images, including a total of 13,028 instances from 23 families and 54 species of animals. Each labeled instance contains 17 keypoints that are defined to illustrate animal poses, and the number of keypoints is the same as COCO [40]. Specifically, the annotation format of AP-10K is also similar to COCO. The detailed information is as shown in Table 1. The dataset is full of challenges for the many kinds of mammalian animals with complicated backgrounds in wild environments. We follow the official division of AP-10K, which is split into three non-overlapping parts, consisting of the train, validation, and test sets, containing 7010, 1002 and 2003 images, respectively.

Table 1. Illustrating the definition of animal joints.

Keypoint	Definition	Keypoint	Definition
1	Left Eye	10	Right Elbow
2	Right Eye	11	Right Front Paw
3	Nose	12	Left Hip
4	Neck	13	Left Knee
5	Root of Tail	14	Left Back Paw
6	Left Shoulder	15	Right Hip
7	Left Elbow	16	Right Knee
8	Left Front Paw	17	Right Back Paw
9	Right Shoulder		

2.2. Methods

This work concentrates on animal pose estimation among wild animals with high accuracy under situations in which there are limited computing resources. The whole structure of our network is shown in Figure 1. It adapts a top-down paradigm to detect the joints of animals and maintains high-resolution representation at every stage.

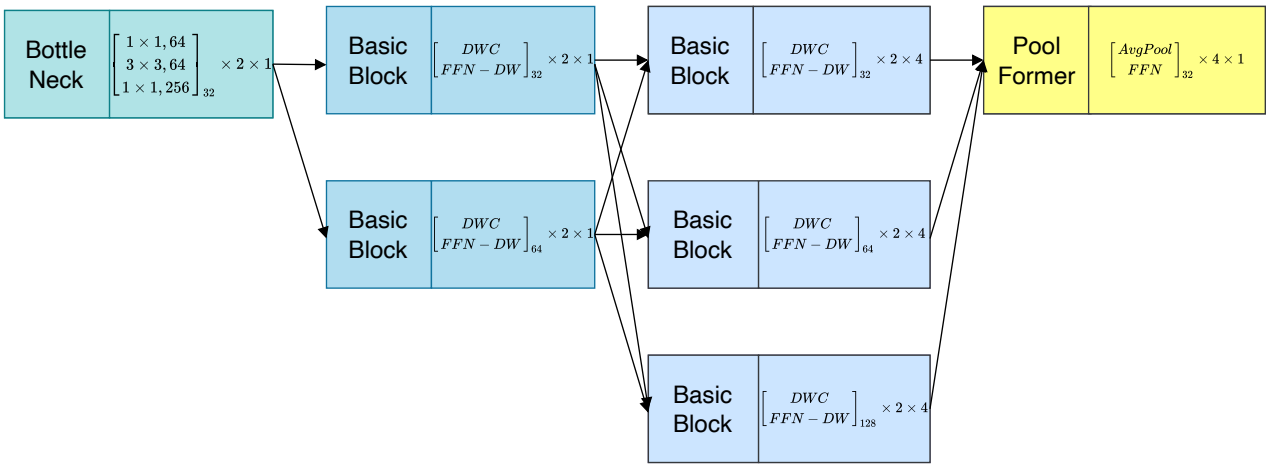


Figure 1. The illustration of the DepthFormer architecture.

2.2.1. Multi-Branch Neural Network

We employ a parallel structure, which is similar to HRNet, where the entire process is divided into three phases and new branches with lower resolution are inserted into each stage. Branches with various resolutions are dispersed in various phases concurrently. While low-resolution representation includes more semantic information, which is crucial for pose estimation tasks, high-resolution representation has advantages in recording spatial information. To combine features from distinct branches between each of the two stages, up- and down-sampling are utilized. This results in a high-resolution representation that incorporates more semantic information when recording spatial features.

In order to make full use of the representation under different resolutions, the feature fusion mechanism in $HRNetV2$ [9] is adopted. Up-sampling is used at the end of the parallel structure to combine two low-resolution representations into the high-resolution representation, which helps the model perform better. It's important to note that we exclude the fourth stage instead of duplicating the four stages structure. Following calculation and comparison, it is determined that roughly 60% of the parameters for the entire model are contained in the fourth stage. We chose a three-stage structure to reduce the size of the model. The expense of the tailoring must be paid, and the model's capacity for representation should deteriorate. On the expectation that the model's parameters and calculation quantity are kept at a low level, four PoolFormer blocks (a yellow part in Figure 1) are added to the tail of the parallel structure in order to partially bridge this performance gap. In Section 2.2.4, the PoolFormer block's structure will be presented.

2.2.2. Depthwise Convolution

Depthwise convolution, proposed in 2014 by Laurent Sifre [41] and further developed by Xception [42], has become more popular since it achieved a great performance. For the above reasons, depthwise convolution is widely used in modern backbones such as MobileNet, ShuffleNet and Horizontal Shortcut Connections (HSC) [43].

In the operator, every depth level of the input feature map $X \in \mathbb{R}^{H\times W\times C}$ is extracted by a two-dimensional filter applying a channel process only on one input channel [41]. The operator of each input channel can be described as:

$$X'_{DWC} = Concat(K_n \cdot X_n), \tag{1}$$

where K_n is the filter of the n_{th} layer and X_n is the n_{th} channel of the input X. Xception network [42] uses a structure made of residual connections, including layers of depthwise convolution, which allows the efficient component to be implementated in practice and shows it could be a cornerstone of deep learning.

2.2.3. DepthFormer Block

There are two residual sub-blocks of every single block. The input $X \in \mathbb{R}^{H \times W}$ is first processed by $Norm()$, selecting representative batch normalization [44] (RBN) as our choice. The specific function is:

$$Y = X + TokenMixer(Norm(X)). \quad (2)$$

According to the traditional approach, $TokenMixer()$ is always set to self-attention [15,20,21,30,45]. The Swin Transformer [21] uses the idea of the sliding window in convolutional neural networks to reduce the time complexity of the self-attention mechanism to a linear proportion of the image size. Based on this idea, HRFormer [29] applies the Swin Transformer to a high-resolution network structure. Figure 2 shows the structure of the HRFormer block where the green part is the local-window self-attention (W-MSA). The feature map X is cut into several parts and runs multi-head self-attention in each part independently. Moreover, these parts do not overlap with each other in an image. Each part includes $M \times M$ patches; the time complexity of a window based on $h \times w$ patches of an image is [21]:

$$\Omega(W\text{-}MSA) = 4hwC^2 + 2M^2hwC \quad (3)$$

where the value of W is normally set as seven, hw is the number of patches and C is the channels of a feature map. The algorithm is complicated and sophisticated, bringing about huge difficulties when it is trained and worked.

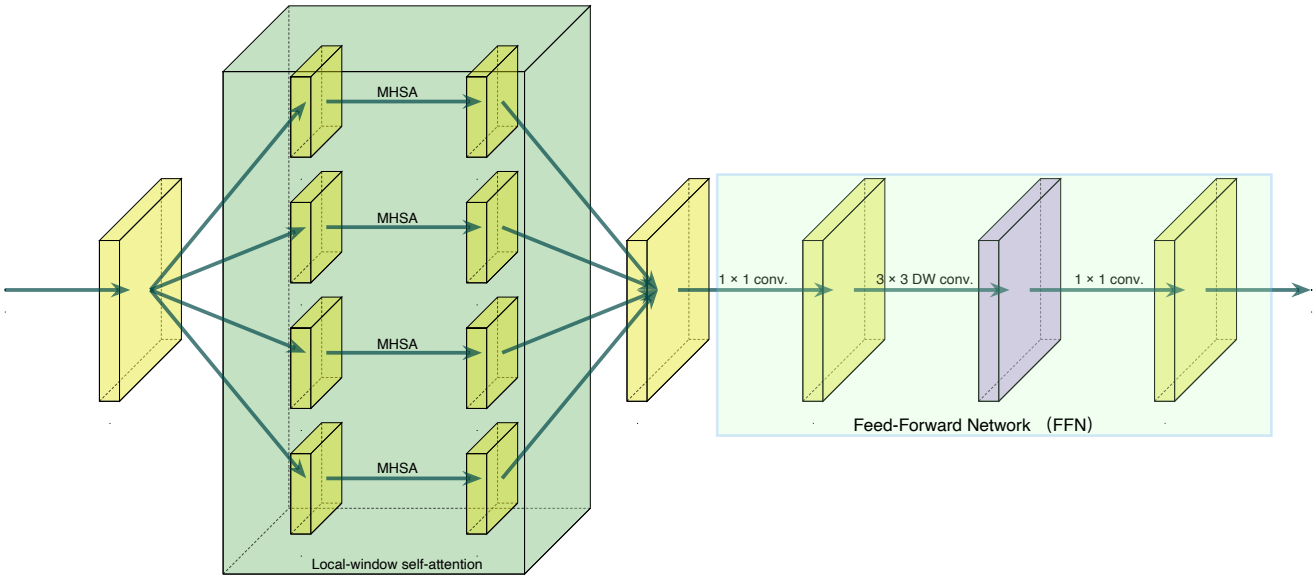


Figure 2. Illustrating the structure of the HRFormer block.

Qi et al. [28] suggest that depthwise convolution and self-attention have strong similarities. Self-attention and depthwise convolution have two things in common. The first is sparse connectivity, where each channel in depthwise convolution has a separate convolution kernel. They are sparsely connected on the channel and locally linked in space because different channels are independent of one another. The second is weight sharing: depthwise convolution (DWC) uses distinct aggregation weights for each channel while using convolution kernels with the same weight for feature aggregation at each point [28]. In the operator, the time complexity is:

$$\Omega(DWC) = CK^2HW, \quad (4)$$

where C is the number of channels and K is the kernel size. Specifically, due to using a basic design for the Transformer, there is no down-sampling in each basic block, which maintains

an identical feature map size of the input and ouput. For the purpose of designing a lightweight model, we select depthwise convolution as our $TokenMixer()$, since it has small parameters and is easy to train.

Then, in the FFN part, following the previous operation, the point convolution is used to further integrate the depth information from the depth direction. After that, a set of depthwise and pointwise convolutions is added to extract more information [32]. The blocks behind the second one of Figure 3 show the process of how 3×3 depthwise convolution and pointwise convolution are used to update features.

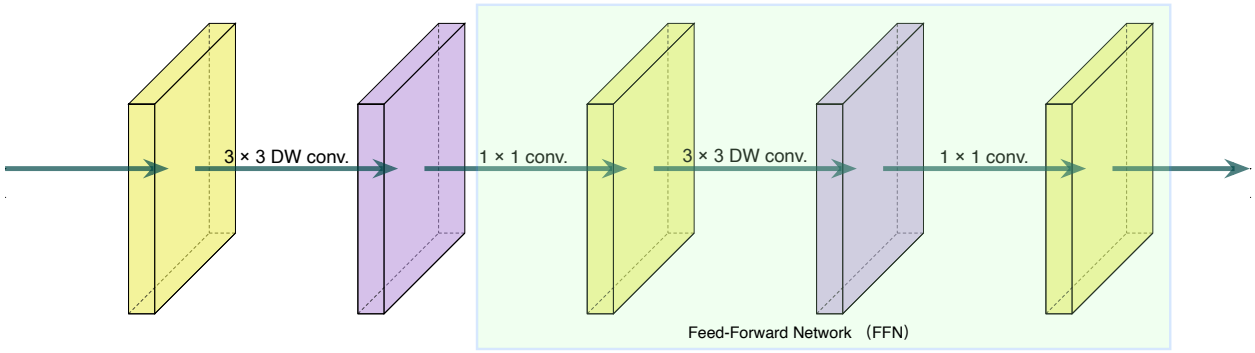


Figure 3. Illustrating the structure of our basic block.

2.2.4. PoolFormer Block

In order to balance the relationship between the effect and parameter quantity, four PoolFormer blocks are added behind the parallel structure.

First the input $X \in \mathbb{R}^{H \times W \times C}$ is reshaped into several flattened 2D patches $x_p \in \mathbb{R}^{p^2 \times C}$, where $(H \times W)$ is the original resolution of the feature map and $(p \times p)$ is the resolution of every single image patch.

$$X' = Embedding(X) \tag{5}$$

Then, the embedding output is transferred into four repeated PoolFormer blocks. Compared with the block introduced in the previous section, $TokenMixer()$ processing is substituted by a simple $AvgPooling$ operator. The structure is shown in Figure 4. The red component is the $Pooling$ part. The time complexity is as follows:

$$\Omega(AvgPooling) = HWCK^2 \tag{6}$$

where K is the size of a kernel in the $Pooling$ operator. $Pooling$ with the invariance of translation, rotation and scale could keep the main features of the input and prevent overfitting. The PoolFormer block removes the depthwise convolution operation in FFN in order not to increase the amount of computation. How many blocks need to be added will be discussed in the ablation study.

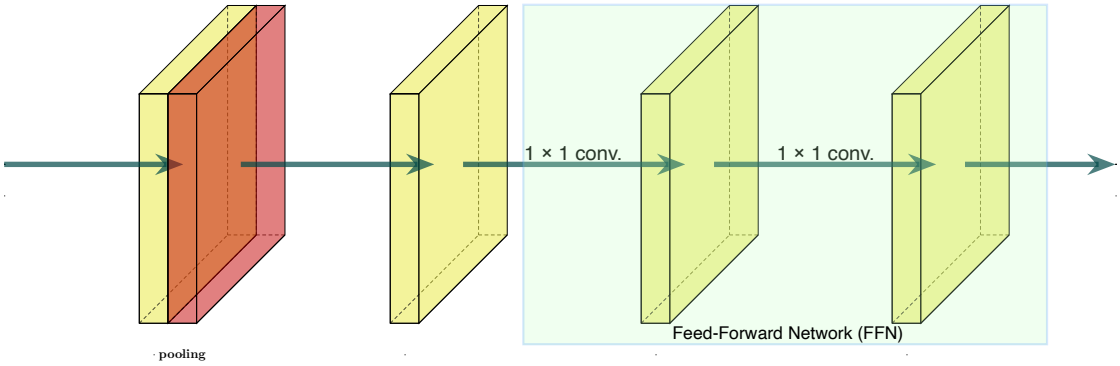


Figure 4. Illustrating the structure of a PoolFormer block.

2.2.5. Representative Batch Normalization

The dataset used for this task includes numerous animal families with various visual characteristics. All of the images are separated into various batches for the training process. We anticipate that effective reservation of the sample-specific instance representations will be possible. Layer normalization is substituted in our network by representative batch normalization (RBN) [44]. It has an easy-to-use feature calibration solution to improve the instance-specific representations. The ability of the network model to represent features is improved and the performance impact of inaccurate information is decreased when the RBN calibration is centered. Through a scaling calibration, we achieve a feature distribution that is more stable. RBN performs better than layer normalization, and the comparison of the outcomes will be displayed in the ablation study.

2.2.6. Instantiation

We demonstrate the whole structure information of DepthFormer in Figure 1. The number of blocks and modules of four stages are (2, 2, 2, 1) and (1, 1, 4, 4) correspondingly. Additionally, the channels of the four stages are shown as (32, 64, 128, 32). The first stage still follows the original structure of HRNet, with several bottleneck blocks in the residual network [16,29]. The blocks of other stages maintain the basic design of the Transformer [21,23].

2.3. Evaluation Indicators and Experimental Environment

2.3.1. Evaluation Metric

In our work, the standard evaluation metric based on Object Keypoint Similarity (OKS) is confirmed:

$$OKS = \frac{\sum_i e^{\frac{-d_i^2}{2s^2k_i^2}} \delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{7}$$

where d_i is defined by the L_2 norm, which describes the distance between a predicted joint and the ground truth of itself; v_i illustrates whether the joint could be seen, s is the scale of the object, and a constant k_i is described to control the decay of each keypoint. We report the standard average precision and recall scores: AP (the mean of AP scores at ten positions, OKS = 0.50, 0.55,. . . , 0.90, 0.95), AP^{50} (AP at OKS = 0.50), AP^{75} and AR [14].

2.3.2. Experimental Settings

We use data augmentation which mainly consists of randomly flipping animal keypoints horizontally, randomly keeping only the upper body or the lower body, as well as scaling $[0.5, 1.5]$ and rotating $[-80°, 80°]$. In the selection of the input size, we use a square box instead of a general rectangular box, because different animals have large body differences, and the square box can better adapt to various situations. The input size of each image is 192×192 or 256×256, and the heat map size is 48×48 or 64×64 correspondingly. We change the optimizer from Adam to AdamW and set the learning rate as 5×10^{-4} with the weight decay set as 0.01. In order to compare the differences between models, the same settings are used for the keypoint head and loss functions in all experiments. All the experiments start from scratch. Each experiment runs on a NVIDIA GeForce RTX 3090 (24G) GPU under a Pytorch 1.8.0 framework with a batch size of 58 and 210 epochs.

3. Results

3.1. Results on the Val Set

Table 2 summarizes the performance on the AP-10K val set. DepthFormer arrives at an AP of 65.4, which surpasses all the backbone networks. We select SimpleBaseline as the benchmark, using the main structure of ResNet and adding several connections of the cross-scale. Additionally, the depth of the net is 18. Although it is a simple net, it is always efficient and has a great performance. The result of our work outnumbers that of SimpleBaseline by 3.0 AP and 6.9 AP with input sizes of 256×256 and 192×192,

respectively. Note that the parameters of the baseline are approximately five times as many as that of ours. HRFormer-Tiny uses four-stage high-resolution architecture combining multi-scale outputs and makes good use of the Swin Transformer block, which makes it achieve good results in many vision tasks.

Table 2. Comparison on the AP-10K val set. The volume of parameters and FLOPs for the animal pose estimation network are measured without animal detection and the keypoint head.

Methods	Input Size	#param.	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
SimpleBaseline-18	192 × 192	11.17 M	1.33	56.6	87.2	59.6	45.4	56.9	60.3
SimpleBaseline-18	256 × 256	11.17 M	2.37	62.4	89.3	66.4	37.0	62.9	66.6
ShuffleNetV2	192 × 192	1.25 M	0.10	47.1	82.6	45.3	33.2	47.5	52.0
ShuffleNetV2	256 × 256	1.25 M	0.19	53.2	85.3	54.0	38.3	53.5	59.1
MobileNetV2	192 × 192	2.22 M	0.22	49.5	84.0	48.0	40.3	49.8	53.9
MobileNetV2	256 × 256	2.22 M	0.40	56.1	86.4	56.6	50.0	56.4	61.3
Lite-HRNet	192 × 192	1.76 M	0.30	58.0	89.0	61.1	52.1	58.2	62.3
Lite-HRNet	256 × 256	1.76 M	0.54	60.7	90.1	63.3	53.7	61.0	65.5
HRFormer-Tiny	192 × 192	2.49 M	1.03	61.7	90.9	65.9	55.8	62.0	65.6
HRFormer-Tiny	256 × 256	2.49 M	1.89	62.2	91.6	67.8	47.8	62.6	67.1
DepthFormer	192 × 192	2.27 M	1.50	**63.5**	91.0	67.5	51.5	63.8	67.3
DepthFormer	256 × 256	2.27 M	2.67	**65.4**	92.2	70.6	52.9	65.8	70.0

In this challenging dataset, our network shows a more powerful ability to handle the task. Compared with HRFormer-Tiny, DepthFormer overtakes it by 3.2 and 1.8 score of AP with input sizes of 256 × 256 and 192 × 192, respectively, at the same level of parameters. The table also reports other state-of-the-art lightweight networks. The performance of DepthFormer shows great advantages in both AP and AR.

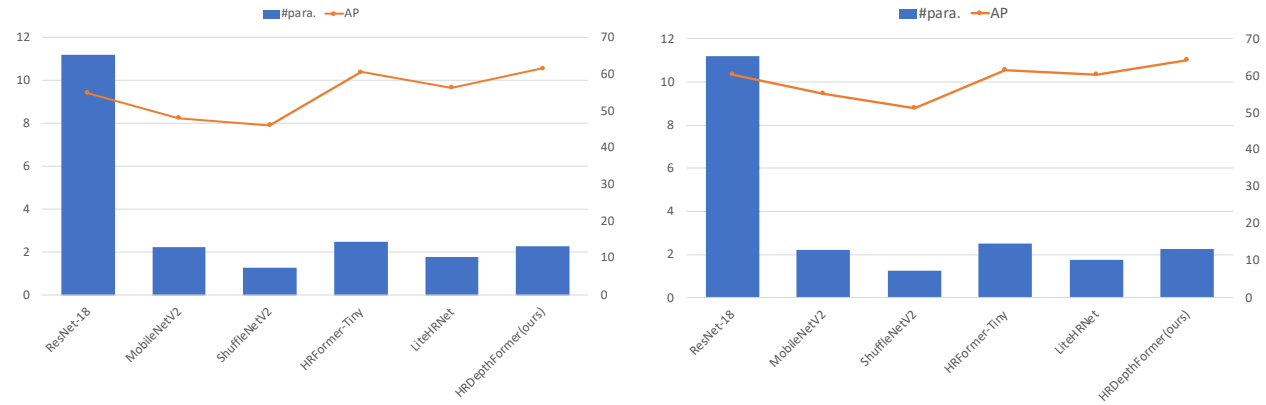
3.2. Results on the Test Set

Table 3 reports the results of the AP-10K test set. Our method is significantly better than low-resolution networks such as MobileNet and ShuffleNet. Our network can benefit from increasing the input sizes from 192 × 192 to 256 × 256, which gains 2.5 AP, and others also show the same trend. Although the computation of DepthFormer is larger than other models with higher resolution at the same input size, our performance is eye-catching. It is worth noting that the performance of our model under an input size of 192 × 192 exceeds that of other models under an input size of 256 × 256. In other words, DepthFormer achieves better results with smaller input sizes and smaller GFLOPs.

Table 3. Comparison on the AP-10K test set. The volume of parameters and FLOPs for the animal pose estimation network are measured without animal detection and the keypoint head.

Methods	Input Size	#param.	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
SimpleBaseline-18	192 × 192	11.17 M	1.33	54.9	86.2	56.6	38.7	55.5	59.4
SimpleBaseline-18	256 × 256	11.17 M	2.37	60.3	88.5	63.0	48.1	61.1	65.2
ShuffleNetV2	192 × 192	1.25 M	0.10	46.2	79.1	45.1	36.7	46.8	51.0
ShuffleNetV2	256 × 256	1.25 M	0.19	51.3	83.7	50.3	38.7	52.0	57.4
MobileNetV2	192 × 192	2.22 M	0.22	48.1	81.0	47.6	40.3	48.6	53.0
MobileNetV2	256 × 256	2.22 M	0.40	55.1	86.3	55.8	43.8	55.7	60.5
Lite-HRNet	192 × 192	1.76 M	0.30	56.4	87.1	59.2	47.7	56.9	60.9
Lite-HRNet	256 × 256	1.76 M	0.54	60.2	88.7	62.7	50.0	60.8	65.3
HRFormer-Tiny	192 × 192	2.49 M	1.03	60.6	88.8	65.7	48.0	61.3	65.1
HRFormer-Tiny	256 × 256	2.49 M	1.89	61.5	90.0	66.6	47.3	62.2	66.9
DepthFormer	192 × 192	2.27 M	1.50	**62.1**	89.1	67.0	51.9	66.7	66.3
DepthFormer	256 × 256	2.27 M	2.67	**64.1**	89.5	69.3	55.4	64.7	69.0

For example, our approach arrives at an AP score of 62.1 with 1.50 GFLOPs under an input size of 192 × 192, outnumbering both HRFormer-Tiny with 1.89 GFLOPs and SimpleBaseline-18 with 2.37 GFLOPs under an input size of 256 × 256 by 0.6 AP and 1.3 AP, respectively, as reported in Figure 5. Our model could work better than others with low-resolution images captured from the wild environment. At the same time, the parameters of SimpleBaseline and HRFormer are higher than those of our model. When the input size is 256 × 256, our model outperforms HRFormer (which is in second place) by 2.6 AP. Some visual results are shown in Figure 6. DepthFormer could detect the keypoints of animals accurately in most cases, but in the case of occlusion among animals, the detection effect is limited.



(a) Input size of 192 × 192. **(b)** Input size of 256 × 256.

Figure 5. Demonstration of the AP and parameters of the lightweight networks on the AP-10K test set with inputs of different sizes.

Figure 6. Example qualitative results of the AP-10K animal pose estimation test, including multiple animals and changes in viewpoints.

3.3. Ablation Study

Table 4 shows the results of the ablation study under an input size of 192 × 192. We select the PoolFormer block as a basic block of the parallel structure. First of all, we test the performance of the network with only three stages, which is a baseline to test whether the improvements are valid. In addition, the pooling operator is replaced by depthwise convolution, which also exerts a positive influence on performance. These changes brought about a small increase in parameters and GFLOPs, with a 0.4 increase in the AP score.

Table 4. Influence of selecting 3 × 3 depthwise convolution (DWC) as the *TokenMixture*() and representative batch normalization (RBN). We show #param, GFLOPs, AP and AR on AP-10K val.

No.	Baseline	DWC	RBN	2 Blocks	4 Blocks	6 Blocks	#Param.	GFLOPs	*AP*	*AR*
1	√						2.197	1.352	60.2	64.7
2	√	√					2.216	1.363	60.6	65.1
3	√		√				2.197	1.357	60.5	64.9
4	√	√	√				2.216	1.368	61.6	65.9
5	√	√	√	√			2.259	1.466	61.7	65.9
6	√	√	√		√		2.275	1.502	62.1	66.3
7	√	√	√			√	2.292	1.543	61.8	66.1

Then, a layer normalization is changed by representative batch normalization and there is a clear increase in the AP with this change, which brings only a minimal increase in GFLOPs. After that, both depth-wise convolution and representative batch normalization are utilized in this network. The results are better than only using a single one (61.6 AP vs. 60.6 AP and 60.5 AP), illustrating that there is synergy between the two of them. Based on the first two changes, several PoolFormer blocks are added at the tail of the parallel structure. Obviously, it is not the case that the greater the number of blocks, the better. In a series of experiments, we add two, four, and six blocks to conduct experiments to evaluate the best effect of adding several blocks. When adding two blocks, the effect is slightly improved, and the AP is increased by 0.1. When the number of blocks becomes four, it increases the AP to 0.5 compared with not adding blocks. However, when six blocks are added, there is a marginal effect, and the performance does not increase but clearly decreases.

As a result, our network benefits from depthwise convolution and representative batch normalization, and adding four blocks at the tail is the most suitable method for us.

4. Discussion

In this work, to solve a problem of developing an efficient method for animal pose estimation, a lightweight, high-resolution and depthwise convolution network is proposed. The basic block of DepthFormer is designed with depthwise convolution, following the whole structure of Transformer, where there are two similarities between depthwise convolution and self-attention, with less inference time. The results of a series of experiments show that architecture of Transformer has strong power, although *TokenMixer*() is set as *Pooling*. Depthwise convolution could extract feature effectively and bring a great performance, with a few parameters. Moreover, a feature fusion operation is used at the end of a three-stage parallel branch, making good use of the representation with different resolutions. The models that can get rich multi-scale representation such as ours, Lite-HRNet and HRFormer-Tiny show obvious superiority in this task. The RBN is used to maintain instance-specific representations of samples, retaining the other advantages of batch normalization. With the help of the characteristic, our model could get an obvious improvement. It is important that RBN and depthwise convolution have strong synergistic effect.

To make up for the deficiency of the three-stage model in terms of feature extraction, four PoolFormer blocks are added at the end and is demonstrated in ablation experiments to improve performance positively. The performance of our model with a small input size

exceeds that of other models with a large input size. In other words, our DepthFormer could get a better performance with a fewer GFLOPs and parameters than other state-of-the-art lightweight models (HRFormer-Tiny and SimpleBaseline-18). Repeatable experiments with good results show that the proposed lightweight model has strong reliability in this challenging vision task. This research provides effective and reliable technical support for accurately estimating animal poses with limited computing resources.

Although the overall performance of our method is good and far better than that of other models, it requires a little more GFLOPs than MobileNet [32], ShuffleNet [34] and Lite-HRNet [35]. In the future, these problems will be solved by more efficient network structures. Additional development will be launched on edge devices and applied in practice.

5. Conclusions

In this paper, a lightweight, high-resolution and depthwise convolution network is proposed for animal pose estimation. Relying on the well-designed structure of Transformer with strong power, our DepthFormer makes good use of multi-branch architecture to extract high-resolution representations and RBN to obtain instance-specific representations. A series of repeatable and accurate experimental results show that our model could get a greater performance than other lightweight models with a smaller input size. The model will be improved by lighter designs and more efficient structures.

Author Contributions: Conceptualization, S.L. (Sicong Liu) and Q.F.; methodology, S.L. (Sicong Liu) and Q.F; software, Q.F., S.L. (Shanghao Liu) and S.L. (Sicong Liu); validation, Q.F. and S.L. (Sicong Liu); data curation, Q.F.; writing—original draft preparation, S.L. (Sicong Liu), Q.F and S.L. (Shanghao Liu); writing—review and editing, Q.F., S.L. (Shanghao Liu) and S.L. (Sicong Liu); visualization, Q.F.; supervision, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Beijing Natural Science Foundation (4202029).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank Hang Yu of Xidian University in China for opening the AP-10K dataset. This has provided great help for our research.

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

Abbreviations

The following abbreviations are used in this manuscript:

FFN	Feed-Forward Network
ViT	Vision Transformer
MHSA	Multihead Self-Attention
W-MSA	Window-Based Multi-head Self-Attention
FFN-DW	Feed-Forward Network with a 3 × 3 Depth-wise Convolution
HRFormer	High-Resolution Transformer
HRNet	High-Resolution Network
HSC	Horizontal Shortcut Connections

References

1. Arac, A.; Zhao, P.; Dobkin, B.H.; Carmichael, S.T.; Golshani, P. DeepBehavior: A Deep Learning Toolbox for Automated Analysis of Animal and Human Behavior Imaging Data. *Front. Syst. Neurosci.* **2019**, *13*, 20. [CrossRef] [PubMed]
2. Padilla-Coreano, N.; Batra, K.; Patarino, M.; Chen, Z.; Rock, R.R.; Zhang, R.; Hausmann, S.B.; Weddington, J.C.; Patel, R.; Zhang, Y.E.; et al. Cortical ensembles orchestrate social competition through hypothalamic outputs. *Nature* **2022**, *603*, 667–671. [CrossRef] [PubMed]

3. Li, S.; Li, J.; Tang, H.; Qian, R.; Lin, W. ATRW: A Benchmark for Amur Tiger Re-identification in the Wild. In Proceedings of the MM: International Multimedia Conference, Seattle, WA, USA, 12–16 October 2020; pp. 2590–2598. [CrossRef]
4. Harding, E.J.; Paul, E.S.; Mendl, M. Cognitive bias and affective state. *Nature* **2004**, *427*, 312. [CrossRef] [PubMed]
5. Mathis, A.; Mamidanna, P.; Cury, K.M.; Abe, T.; Murthy, V.N.; Mathis, M.W.; Bethge, M. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **2018**, *21*, 1281–1289. [CrossRef] [PubMed]
6. Graving, J.M.; Chae, D.; Naik, H.; Li, L.; Koger, B.; Costelloe, B.R.; Couzin, I.D. DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife* **2019**, *8*, e47994. [CrossRef]
7. Labuguen, R.; Matsumoto, J.; Negrete, S.B.; Nishimaru, H.; Nishijo, H.; Takada, M.; Go, Y.; Inoue, K.i.; Shibata, T. MacaquePose: A Novel "In the Wild" Macaque Monkey Pose Dataset for Markerless Motion Capture. *Front. Behav. Neurosci.* **2021**, *14*, 581154. [CrossRef]
8. Yu, H.; Xu, Y.; Zhang, J.; Zhao, W.; Guan, Z.; Tao, D. AP-10K: A Benchmark for Animal Pose Estimation in the Wild. *arXiv* **2021**. arXiv:2108.12617.
9. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3349–3364. [CrossRef]
10. Xiao, B.; Wu, H.; Wei, Y. Simple Baselines for Human Pose Estimation and Tracking. In *Computer Vision–ECCV 2018*; Lecture Notes in Computer Science; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; Volume 11210, pp. 472–487. [CrossRef]
11. Nie, X.; Feng, J.; Zhang, J.; Yan, S. Single-Stage Multi-Person Pose Machines. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6950–6959. [CrossRef]
12. Kreiss, S.; Bertoni, L.; Alahi, A. PifPaf: Composite Fields for Human Pose Estimation. *arXiv* **2019**, arXiv:1903.06593.
13. Newell, A.; Huang, Z.; Deng, J. Associative Embedding: End-to-End Learning for Joint Detection and Grouping. *arXiv* **2017**, arXiv:1611.05424.
14. 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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5385–5394. [CrossRef]
15. Li, K.; Wang, S.; Zhang, X.; Xu, Y.; Xu, W.; Tu, Z. Pose Recognition with Cascade Transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 1944–1953. [CrossRef]
16. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5686–5696. [CrossRef]
17. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In *Computer Vision–ECCV 2016*; Lecture Notes in Computer Science; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 9912, pp. 483–499. [CrossRef]
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *CoRR* **2017**, *30*. arXiv:1706.03762.
19. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
20. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. *arXiv* **2020**, arXiv:2005.12872.
21. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
22. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. *arXiv* **2022**, arXiv:2107.00652.
23. Yu, W.; Luo, M.; Zhou, P.; Si, C.; Zhou, Y.; Wang, X.; Feng, J.; Yan, S. MetaFormer is Actually What You Need for Vision. *arXiv* **2021**, arXiv:2111.11418.
24. Andreoli, J.M. Convolution, attention and structure embedding. *arXiv* **2020**, arXiv:1905.01289.
25. Cordonnier, J.B.; Loukas, A.; Jaggi, M. On the Relationship between Self-Attention and Convolutional Layers. *arXiv* **2020**, arXiv:1911.03584.
26. Wu, F.; Fan, A.; Baevski, A.; Dauphin, Y.N.; Auli, M. Pay Less Attention with Lightweight and Dynamic Convolutions. *arXiv* **2019**, arXiv:1901.10430.
27. Tay, Y.; Dehghani, M.; Gupta, J.P.; Aribandi, V.; Bahri, D.; Qin, Z.; Metzler, D. Are Pretrained Convolutions Better than Pretrained Transformers? In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (ACL), Bangkok, Thailand, 1–6 August 2021; pp. 4349–4359. [CrossRef]
28. Han, Q.; Fan, Z.; Dai, Q.; Sun, L.; Cheng, M.M.; Liu, J.; Wang, J. On the Connection between Local Attention and Dynamic Depth-wise Convolution. *arXiv* **2022**, arXiv:2106.04263.

29. Yuan, Y.; Fu, R.; Huang, L.; Lin, W.; Zhang, C.; Chen, X.; Wang, J. HRFormer: High-Resolution Vision Transformer for Dense Predict. In Proceedings of the NeurIPS 2021, Virtual, 13 December 2021.
30. Li, K.; Wang, Y.; Zhang, J.; Gao, P.; Song, G.; Liu, Y.; Li, H.; Qiao, Y. UniFormer: Unifying Convolution and Self-attention for Visual Recognition. *arXiv* **2022**, arXiv:2201.09450.
31. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
33. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856. [CrossRef]
34. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *Computer Vision–ECCV 2018*; Lecture Notes in Computer Science; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; Volume 11218, pp. 122–138. [CrossRef]
35. Yu, C.; Xiao, B.; Gao, C.; Yuan, L.; Zhang, L.; Sang, N.; Wang, J. Lite-HRNet: A Lightweight High-Resolution Network. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 10435–10445. [CrossRef]
36. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
37. Luo, Y.; Ou, Z.; Wan, T.; Guo, J.M. FastNet: Fast high-resolution network for human pose estimation. *Image Vis. Comput.* **2022**, *119*, 104390. [CrossRef]
38. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *arXiv* **2017**, arXiv:1612.03144.
39. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. *arXiv* **2018**, arXiv:1808.00897.
40. Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 3686–3693. [CrossRef]
41. Laurent, S. Rigid-Motion Scattering For Image Classification. Ph.D. Thesis, Ecole Polytechnique, Palaiseau, France, 2014.
42. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2017**, arXiv:1610.02357.
43. Zhu, A.; Liu, L.; Hou, W.; Sun, H.; Zheng, N. HSC: Leveraging horizontal shortcut connections for improving accuracy and computational efficiency of lightweight CNN. *Neurocomputing* **2021**, *457*, 141–154. [CrossRef]
44. Gao, S.H.; Han, Q.; Li, D.; Cheng, M.M.; Peng, P. Representative Batch Normalization with Feature Calibration. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8665–8675. [CrossRef]
45. Stoffl, L.; Vidal, M.; Mathis, A. End-to-End Trainable Multi-Instance Pose Estimation with Transformers. *arXiv* **2021**, arXiv:2103.12115.